Zen IoT Register Supplement

Table of Contents

Part I

1 Introduction

This Introduction shows how the different register types are used and arranged for the Zen IoT controller.

See Register Types

See also
Communication Formats

ASCII Mode Format

Macro Compiling & Uploading

Register List

1.1 Register Types

The controller uses 8, 16, 24, and 32-bit signed, unsigned, and floating point registers. There are two types of register used in the controller.

Configuration Register
A configuration register stores signal constants that change only when they are reprogrammed. For example, registers 1129 and 359 store digital counter channel 1 input scale and offset settings.

Working Register
A working register stores signal data that changes regularly due to variations in the input signal, as well as the processes carried out by the controller's functions on the input signal. For example, register 645 stores the processed data for the input signal after it has been processed through the channel 1 functions programmed into the controller.

See also
Intech A16 Compatibility Registers (1 to 127)

32-bit Fixed Point (129 to 1023)

32-bit Floating Point (1025 to 1535)

32-bit Pseudo Floating Point (1537 to 2047)

24-bit Fixed Point (2049 to 3072)

Input Module Registers (3073 to 4096)

16-bit Fixed Point (4097 to 8192)

8-bit Fixed Point (8193 to 16384)

Text Registers (16385 to 20479)

Macro Code Registers (32769 to 65536)

1.1.1 Intech A16 Compatability Registers (1 to 127)

Register addresses 1 to 127 are provided to give backwards compatibility to previous Intech A16 controllers and contain a mixture of 12 & 16 bit fixed point and 32 bit floating point registers. For those registers which are floating point, only odd register addresses are used. Otherwise both odd and even registers addresses are used.

1.1.2 32-bit Fixed Point (129 to 1023)

Register addresses 129 to 1023 are used for 32-bit fixed point addresses. To accommodate for Modbus usage of 32-point registers, only odd register addresses are used, providing a maximum of 447 registers.

1.1.3 32-bit Floating Point (1025 to 1535)

Register addresses 1025 to 1535 are used for 32-bit floating point addresses. All registers in this range are single precision floating point numbers that conform to the IEEE-754 standard format. To accommodate for Modbus usage of 32-point registers, only odd register addresses are used, providing a maximum of 255 registers.

See Also
32-bit Pseudo Floating Point (1537 to 2047)

1.1.4 32-bit Pseudo Floating Point (1537 to 2047)

Register addresses 1537 to 2047 are pseudo 32-bit floating point addresses. To accommodate for Modbus usage of 32-point registers, only odd register addresses are used, providing a maximum of 255 registers.

Pseudo floating point registers are basically floating point images of the 32 bit fixed point registers ranging from register 257 to 767. The float value is created by dividing the original integer value in accordance with it's decimal point selection (see Display Format). Not all 32 bit fixed point registers in the above range have associated user selectable display format registers, and those that don't have preset decimal point settings.

Note: Pseudo floats are only available with Display Format settings from 000 to 006. Anything outside of this range will produce incorrect results. Any rounding applied in the display format setting will be ignored in the pseudo floating point value.

Hint: If you add a register offset of 1280 to any valid 32 bit integer register in the range of 257 to 767, it will address the associated pseudo floating point image of that register.

All registers in this range are single precision floating point numbers that conform to the IEEE-754 standard format. They can be read and written as standard floating point numbers, however they have the following limitations.

Range and Truncation
Because these numbers are derived from an integer value, their range and resolution is limited by how the integer value is configured. For example if the integer register has a display format setting of 1 decimal place, and the value of 0.001234 is written to the pseudo floating point register, the resulting value written to the register will be 0.0.

If the same write is repeated when the display format is set to 6 decimal places then the resulting value written to the register will be 1234 which will be displayed 0.001234.

If the above test is repeated with a display format setting of 4 decimal places, the resulting value written to the register will be 12 which will be displayed 0.0012. The value is truncated and the last 2 decimal places will be lost.

Note: Pseudo floating point registers 1537 to 2047 are only available in firmware version V0.08.01 onwards.

See Also
32-bit Floating Point (1025 to 1535)

1.1.5 24-bit Fixed Point (2049 to 3072)

Register addresses 2049 to 3072 are used for 24 bit fixed point addresses. . To accommodate for Modbus usage of 24 point registers, only odd register addresses are used, giving a maximum of 511 registers.

1.1.6 Input Module Registers (3073 to 4096)

Register addresses 3073 to 4096 are used for Modbus access to input module registers via the index register 8224. Subtracting an offset of 3072 from this register number will give the original register number in the input module. Various data types are used throughout this address range and the user must check the register map for input modules used (contact Define Instruments Ltd. for more information on input module registers and specifications). An absolute maximum of 1023 registers is addressable in this range.

Note: These registers can only be accessed in Modbus RTU mode and only Modbus functions 3 and 16 are supported for accesses within this range. **All other Modbus functions (including function 6 - write single register) are not available when writing to registers 3073 to 4096.** This does not apply to register 8224.

1.1.7 16-bit Fixed Point (4097 to 8192)

Register addresses 4097 to 8192 are used for 16-bit fixed point addresses. Both odd and even addresses in this range are used, providing a maximum of 4096 registers.

1.1.8 8-bit Fixed Point (8193 to 16384)

Register addresses 8193 to 16384 are used for 8-bit fixed point addresses. Both odd and even addresses in this range are used, providing a maximum of 8192 registers.

1.1.9 Text Registers (16385 to 20479)

Register addresses 16385 to 20479 are used for accessing text strings. Only odd addresses in this range are used, providing a maximum of 2047 text strings. Registers 16385 to 16525 are arranged so that they relate to registers numbers 1 to 141 with an offset of 16384 added to them.

See also
ASCII Text Registers

Accessing Text Strings In Modbus

1.1.10 Macro Code Registers (32769 to 65536)

Register addresses 32769 to 65536 are 16-bit unsigned registers used for macro code storage. Both odd and even addresses in this range are used, providing a maximum of 32767 registers.

1.2 Memory Types

Zen IoT series controllers use different types of memory to store register information. In some cases the data is stored in RAM only and is lost at power down (i.e. volatile memory). In other cases the data must be retained at power down so it must be saved in non volatile memory as well. There are also some restrictions on the way some memory types can be used so that their endurance specifications are not exceeded.

The table below shows the different memory types available in the Zen IoT series controllers and the memory characteristics and restrictions which may apply.

Memory Type	Memory Characteristics
RAM	Random Access Memory. This memory is fast to access and is generally used for most working variables. It is volatile memory and it contents are not saved after a power down. Generally this memory is set to zero when the controller is turned on.
EEPROM	Electrically Erasable Programmable Read-Only Memory. This memory is slower to access and usually has a write time of between 5 - 10mS. It also has a limitation of 1x10^6 write cycles which must not be exceeded. There is no limit on the number of read cycles. EEPROM memory is non volatile and it's contents are retained even with no power applied. The controller uses this memory type for non volatile storage of data which is not accessed continuously by the operating system but is needed from time to time.
RAM/EEPROM	This memory type is made up of a combination of the two memory types shown above (i.e. RAM and EEPROM). It is probably the most common memory type used by the controller as it allows fast access and also non volatile storage. When writing to this type of memory from the macro, the RAM value is always updated and the EEPROM value is only updated if the |NON_VOLATILE_WRITE flags is set just prior to the write instruction. This allows the macro to continuously write to a register without exceeding the maximum write cycle limit. When writing to this register via the serial port, both the RAM and EEPROM are updated so care must be taken not to exceed the maximum number of write cycles.
RAM/FLASH	This type of memory is similar to RAM/EEPROM in that it allows fast access and non volatile storage but it uses FLASH memory for the non volatile storage instead of EEPROM. FLASH memory is similar to EEPROM but is usually programmed in larger blocks of memory. This type of memory is used by the controller to store variables which are changing continuously and also need non volatile storage. A write to one of these registers from the macro or the serial port only changes the RAM value. This means that there are no limitations on how many times the register is written. When the power is removed from the controller it senses this and quickly copies the contents of these registers into FLASH memory. When power is restored, the contents of the FLASH memory are copied back into the RAM registers.
RAM/NVRAM	This type of memory uses RAM for fast access and non volatile RAM for data storage. The non volatile RAM is a real time clock device which uses a small battery to retain the contents of the memory during power down. The controller uses this type of memory to store time information.
RAMinputModule	Input modules have an on board microprocessor which contains registers in RAM that can be accessed indirectly by via the index. (See note on Input Module Registers (3073 to 4096))
FLASHinputModule	Input modules have a page of 512 bytes of onboard FLASH memory which holds calibration and setup data. This memory has the similar features and restrictions as the EEPROM listed above. Calibration and setup registers in the input module are written into RAM first via the index register (8224) and then when all data is correct they can be saved to FLASH by setting the save bit (bit 0) in the control byte. FLASH should only be saved in this way when absolutely necessary and care must be taken not to exceed the maximum number of 10^5 write cycles. (See note on Input Module Registers (3073 to 4096))

1.3 Communication Formats

See ASCII Mode

Modbus Mode

Modbus Mode

Character Frame Formats

Command Response Time

1.3.1 ASCII Mode

The ASCII mode is a simple communication protocol using the standard ASCII character set. This mode provides external communication between the controller and a PC allowing remote programming to be carried out. It was designed specifically so that it could be used with standard terminal emulation software allowing the user to communicate with the controller without the need for specialized software. Because of this fact it does not include any error checking or CRC bytes and is intended for configuration of the controller over short distances.

Zen IoT Series controllers use a serial communication channel to transfer data from the controller to another device. With serial communications, data is sent one bit at a time over a single communications line. The voltage is switched between a high and a low level at a predetermined transmission speed (baud rate) using ASCII encoding. Each ASCII character is transmitted individually as a byte of information (eight bits) with a variable idle period between characters. The idle period is the time between the receiving device receiving the stop bit of the last byte sent and the start bit of the next byte. The receiving device (for example a PC) reads the voltage levels at the same interval and then translates the switched levels back to an ASCII character. The voltage levels depend on the interface standard being used.

The following table lists the voltage level conventions used for RS-232 and RS-485. The voltage levels listed are at the receiver.

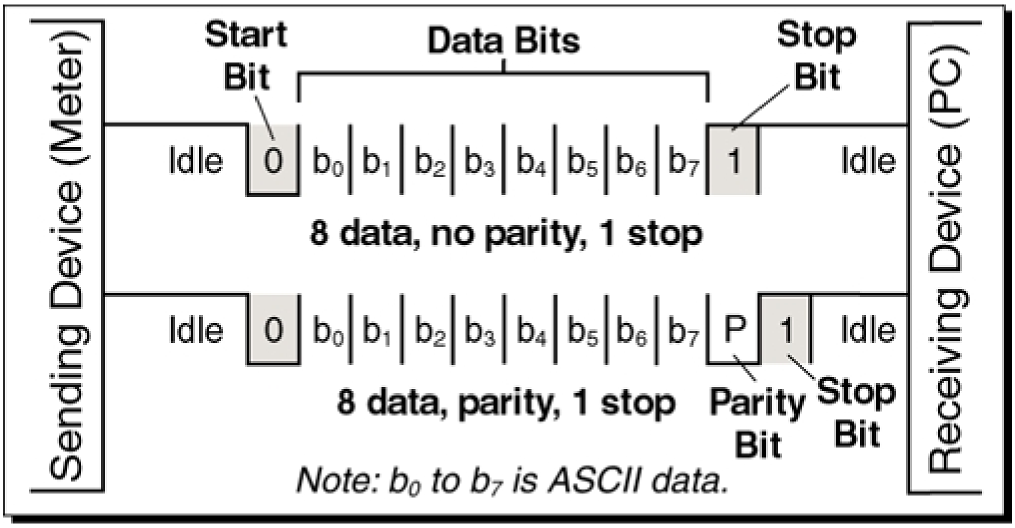
Interface Voltage Level Conventions

Logic	Interface State	RS-232	RS-485
1	Mark (idle)	TXD, RXD: -3 V to -15 V	a+b < -200 mV
0	Space (active)	TXD, RXD: +3 V to +15 V	a-b > +200 mV

Each ASCII character is **framed** with:

- A start bit.
- An optional error detection parity bit.
- And one or more ending stop bits.

For communication to take place, the data format and baud rate (transmission speed) must match that of the other equipment in the communication circuit. The following diagram shows the character frame formats used by the controller.



Character Frame Formats Diagram

See also
Modbus Mode

Character Frame Formats

Command Response Time

1.3.2 Modbus Mode

The Modbus mode uses the Modbus communication protocol to provide external communication between a Zen IoT controller and a process device for monitoring, control, and automation purposes.

Zen IoT controllers use Modbus RTU (Remote Terminal Unit) communication. This is an 8-bit binary transmission mode. The main advantage of this mode is that its greater character density allows better data throughput than ASCII for the same baud rate. Each message must be transmitted in a continuous stream.

Zen IoT controllers can be configured as a Modbus slave device or a Modbus master. In the Modbus slave mode, the controller acts as a slave to a Modbus master (PC or PLC). Data transfers are based on registers and can only be initiated by the Modbus master. The Modbus master must be configured to accept this type of data. Once this is done, seamless communication between the Modbus master and Modbus slave can be initiated.

In Modbus master mode the controller initiates all communications to other Modbus slaves on the bus. On Zen IoT controllers, the Modbus master mode must be used in conjunction with the MODBUS_MASTER_MACRO which defines which slave devices are accessed. (see Modbus Master). In Modbus master mode, Zen IoT Series controllers can only access Modbus Holding registers (in the Modbus 40000 address range) and Input registers (in the Modbus 30000 address range) in external Modbus devices. Coils (20000) are not currently supported.

Modbus Command Summary
All of the registers currently incorporated in Zen IoT Series controllers are accessed as "Holding Registers". Although strictly speaking this means that all of the registers are read/write registers, there are some exceptions to this rule. However the majority of these registers are read/write registers. There are no Discrete input registers, Coils or Input registers available in the Zen IoT.

The following Modbus function codes are supported by Zen IoT controllers in slave mode;

Function Code	Description
1	Read coil status
2	Read input switch status
3	Read holding registers
4	Read input registers
5	Force coils
6	Write single holding register
15	Force multiple coils
16	Write multiple holding registers
23	Read/Write multiple holding registers (V0.08.01 onwards)

NOTE: Access to Modbus addresses 3073 to 4096 are restricted to function codes 3 & 16. (See Input Module Registers (3073 to 4096) for more information).

The following Modbus function codes are supported by Zen IoT controllers in master mode;

Function Code	Description
3	Read holding registers
4	Read input registers
6	Write single holding register
16	Write multiple holding registers

Addressing Convention
All registers numbers contained in this document refer to the original Modbus convention for addressing where register 1 is addressed as 0x0000 in the data packet.

For example, the register number for the Channel 1 processed data register is shown in this document as 9. In Modbus terms this is referred to as 40009. However the actual or direct address contained in the Modbus data packet would be 0x0008 (i.e. 1 count less).

Data Orientation
Zen IoT controllers contain a combination of 8 bit, 16 bit, 24 bit, 32 bit integer, 32 bit floating point registers. The original Modbus protocol only allows for 16bit data registers so to access larger registers, multiple 16 bit registers are accessed. You will notice that all 24 and 32 bit register numbers in the Zen IoT are odd addresses only so that they are spaced 2 register addresses apart from each other. This allows block reads of 32 bit registers to be carried out while still maintaining the correct register addresses.

In Zen IoT controllers the data for 24 and 32 bit registers is transmitted LSW (Least Significant Word) first followed by the MSW (Most Significant Word). In Modbus master mode the user can specify the MB_SWAPPED option to access slave devices which use the alternate format. (See Modbus Master)

For example;
If register 40009 points to a 32 bit long which contains the value 12345678 (0xBC614E hex) then

1st pair of 8 bit bytes transmitted = 0x61 0x4E
2nd pair of 8 bit bytes transmitted = 0x00 0xBC

If the internal register is a 32 bit floating point number then the 1st two 8 bit values transmitted are the least significant 16 bits of the mantissa, while the next two 8 bit values transmitted give the sign, 8 bits of exponent and the most significant 7 bits of the mantissa.

For example;
If register pair 41025 points to a 32 bit float which contains the value –12.5 (0xC1480000 hex) then

1st pair of 8 bit bytes transmitted = 0x00 0x00
2nd pair of 8 bit bytes transmitted = 0XC1 0x48

8 bit Registers
In cases where the internal register is only an 8 bit value, the MSB will be set to zero (if the register is an 8 bit unsigned value) or to the sign (if the register is an 8 bit signed value).

Text String Registers
Zen IoT controllers also contain various text string registers. Text strings vary in maximum length and all text strings must be terminated with an ASCII null (0x00). Most text strings are 14 chars+null (so 15 chars in total) but some are 30chars+null and some are also 62chars+null. (See specific info on each register)

A string can be shorter than the maximum length provided that unused characters are padded with ASCII nulls. Each character in the string is sent in the same order as it appears in the original text string (i.e. 2 characters per 16bit word).

Our addressing of text registers does not strictly adhere to the Modbus spec in that the register number specified for each text string is only used as an entry point into the text string. So for example, register 4016393 is the register number used to access the channel name for input channel 1 which can be up to 14 ASCII characters in length plus a null (ASCII 0x00) terminating character. So the Modbus frame required to read this would be as follows:

Add	Funct	Start Add Hi	Start Add Lo	No. of regs Hi	No. of regs Lo	Chksum
???	0x03	0x40	0x08	0x00	0x08	???

If the channel name was set to "Temp_1" the reply would be as follows:

Add	Funct	Byte Count	Byte1	Byte2	Byte3	Byte4	Byte4	Byte5	Byte6	Byte16	Chksum
???	0x03	0x10	0x54(T)	0x65(e)	0x6D(m)	0x70(p)	0x5F(_)	0x31(1)	0x00	0x00	???

Under standard Modbus addressing another read of register number 4016395 would access byte 5 and 6 (i.e. "_1") of the channel 1 name, but this is not the case in our implementation. Instead register number 4016395 addresses the start of the next text string (i.e. then channel name for input channel 2).

The only limitation with the way we address text string registers is that you can only read/write one complete text string in a single Modbus frame. You cannot access consecutive text string registers in a single long Modbus block read/write. Reading past the maximum size a text string register will give random result values for the unused characters so we recommend that you limit your read/write lengths to those specified for each register.

Data packet Size
Zen IoT controllers can transmit and receive Modbus data packets up to 255 bytes in length.

Modbus/TCP Slave Option
Zen IoT controllers can be supplied with an Ethernet option fitted to serial port 1. When the Ethernet option is fitted, the Zen IoT will automatically switch to Modbus/TCP mode when the Modbus RTU slave protocol is selected for serial port 1. (This also applies to the Intech/Modbus RTU slave protocol. See Intech Mode.) With the Ethernet option fitted the internal serial rate is fixed to 230400 baud, no parity. The Ethernet adapter (Xport device) must be configured with its serial channel set to match. (See ICC402 Ethernet for more information on how to setup the Ethernet port).

NOTE: Later versions of Zen IoT firmware include a Modbus/TCP wrap option which wraps/unwraps TCP packets around a serial frame of data. Its intended for use with some cellular modems and this mode should not be used for standard Modbus/TCP communications.

See also
Character Frame Formats

Modbus Digital Inputs

Modbus Digital Outputs

1.3.3 Intech Mode

The Intech communications mode is designed to allow the Zen IoT series controller to operate with the MicroScan SCADA system developed by Intech Instruments Ltd.

Modbus RTU In Intech Mode
The Intech communications mode also allows Modbus RTU messages to be handled without switching to the standard Modbus mode. In Intech mode, Modbus RTU timing restrictions are slightly relaxed from the Modbus standard with only the inter frame timeout being checked during receive.

1.3.4 MQTT Mode

Firmware V2.2.01 onwards include an MQTT V3.3.1 client operating on port 1. Port 1 can be fitted with a RS232/485 interface, an Ethernet interface or a WiFi module.

The MQTT client must be used in conjunction with a macro or a plugin and cannot operate without this.

See also
MQTT Mode Port 1

1.3.5 Character Frame Formats

Start Bit and Data Bits
Data transmission always begins with the start bit. The start bit signals the receiving device to prepare to receive data. One bit period later, the least significant bit of the ASCII encoded character is transmitted, followed by the remaining data bits. The receiving device then reads each bit position as they are transmitted and, since the sending and receiving devices operate at the same transmission

speed (baud rate), the data is read without timing errors.

Parity Bit
To prevent errors in communication, the sum of data bits in each character (byte) must be the same: either an odd amount or an even amount. The parity bit is used to maintain this similarity for all characters throughout the transmission. It is necessary for the parity protocol of the sending and receiving devices to be set before transmission. There are three options for the parity bit, it can be set to either:

- None – there is no parity.
- Odd – the sum of bits in each byte is odd.
- Even – the sum of bits in each byte is even.

After the start and data bits of the byte have been sent, the parity bit is sent. The transmitter sets the parity bit to 1 or 0 making the sum of the bits of the first character odd or even, depending on the parity protocol set for the sending and receiving devices.
As each subsequent character in the transmission is sent, the transmitter sets the parity bit to a 1 or a 0 so that the protocol of each character is the same as the first character: odd or even.

The parity bit is used by the receiver to detect errors that may occur to an odd number of bits in the transmission. However, a single parity bit cannot detect errors that may occur to an even number of bits. Given this limitation, the parity bit is often ignored by the receiving device. You set the parity bit of incoming data and also set the parity bit of outgoing data to odd, even, or none (mark parity).

Stop Bit
The stop bit is the last character to be transmitted. The stop bit provides a single bit period pause to allow the receiver to prepare to re-synchronize to the start of a new transmission (start bit of next byte). The receiver then continuously looks for the occurrence of the start bit.

See also
Command Response Time

1.3.6 Command Response Time

The controller uses half-duplex operation to send and receive data. This means that it can only send or receive data at any given time. It cannot do both simultaneously. The controller ignores commands while transmitting data, using RXD as a busy signal.

When the controller receives commands and data, after the first command string has been received, timing restrictions are imposed on subsequent commands. This allows enough time for the controller to process the command and prepare for the next command.

See the Timing Diagram below. At the start of the time interval **t1**, the sending device (PC) prints or writes the string to the serial port, initiating a transmission. During **t1** the command characters are under transmission and at the end of this period the controller receives the command terminating character. The time duration of time interval **t1** depends on the number of characters and baud rate of the channel:

```
t1 = (10 * # of characters) / baud rate
```

At the start of time interval **t2**, the controller starts to interpret the command, and when complete performs the command function.

After receiving a valid command string, the controller always indicates to the sending device when it is ready to accept a new command. After a read command, the controller responds with the requested data followed by a carriage return (øDH) and a line feed (øAH) character. After receiving a write command, the controller executes the write command and then responds with a carriage return/line feed.

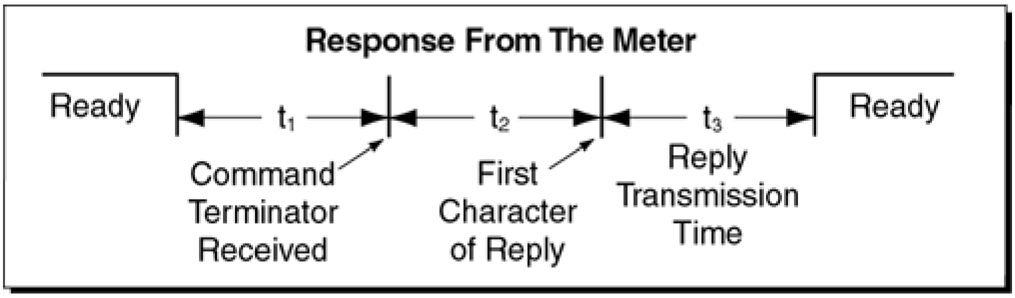
The sending device should wait for the carriage return/line feed characters before sending the next command to the controller.

If the controller is to reply with data, time interval **t2** is controlled by using the command terminating character: **$** or *****. The **$** terminating character results in a response time window of 50 milliseconds minimum and 100 milliseconds maximum. This allows enough time to release the sending driver on the RS-485 bus. Terminating the command line with the ***** symbol, results in a response time window (**t2**) of 2 milliseconds minimum and 50 milliseconds maximum. The faster response time of this terminating character requires that sending drivers release within 2 milliseconds after the terminating character is received.

At the start of time interval **t3**, the controller responds with the first character of the reply. As with **t1**, the time duration of **t3** depends on the number of characters and baud rate of the channel:

```
t3 = (10 * # of characters) / baud rate
```

At the end of **t3** the controller is ready to receive the next command.
The maximum throughput of the controller is limited to the sum of the times: **t1**, **t2**, **t3**.

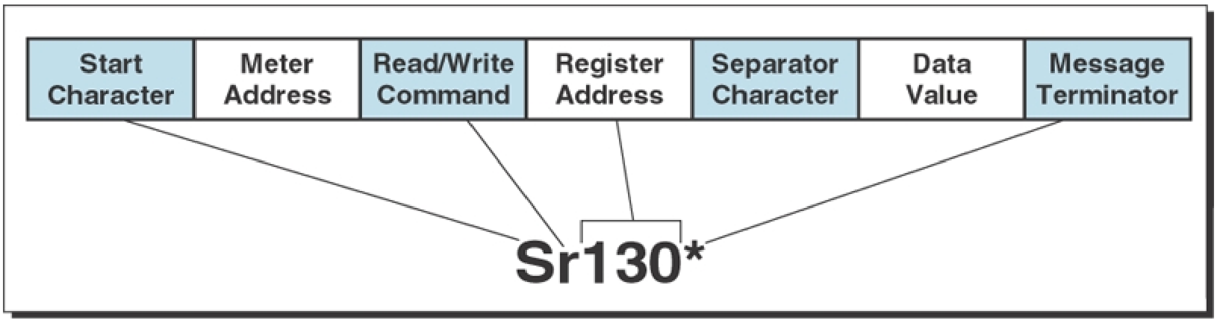


Timing Diagram

See also
ASCII Mode Format

1.4 ASCII Mode Format

Command String Construction
When sending commands to the controller using a Terminal emulation program, a string containing at least one command character must be constructed. A command string consists of the following characters and must be constructed in the order shown:



Command String Construction Diagram

Start Character
Use **S** or **s** for the start character of a command string. This must be the first character in the string.

Controller Address
Use an ASCII number from **0** to **255** for the controller address. If the character following the **start** character is not an ASCII number, then address **0** is assumed. All controllers respond to address **0**.

Read / Write Command

The next character must be an ASCII **R** or **r** for read, an ASCII **U** or **u** for an unformatted read, or an ASCII **W** or **w** for write. Any other character aborts the operation.

Register Address
The register address for the read/write operation is specified next. It can be an ASCII number from **1** to **65535** or, for special text registers, an ASCII letter from **A** to **Z** which is not case sensitive. If the address character is omitted in a read command, the controller always responds with the data value currently on the display. The register address must be specified for a write command.

Separator Character
After the register address in a write command, the next character must be something other than an ASCII number. This is used to separate the register address from the data value. It can be a **space** or a **,** (comma), or any other character except a **$** (dollar) or an * (asterisk).

Data Value
After the separator character, the data value is sent. It must be an ASCII number in the range of **-32766** to **32766**.

Message Terminator
The last character in the message is the message terminator and this must be either a **$** (dollar) or an * (asterisk).

If the **$** is used as a terminator, a minimum delay of 50 milliseconds is inserted before a reply is sent.

If the * is used as a terminator, a minimum delay of 2 milliseconds is inserted before a reply is sent.

NOTE: The **$** and the * characters must not appear anywhere else in the message string.

Controller Response
After the controller has completed a read or write instruction it responds by sending a carriage return/line feed (CR/LF) back to the host. If the instruction was a read command, the CR/LF follows the last character in the ASCII string. If it was a write command, a CR/LF is the only response sent back to the host. The host must wait for this before sending any further commands to the controller.

Unformatted Read
In the ASCII mode data is normally read as formatted data which includes decimal point and any text characters that may be selected to show display units. However it is also possible to read unformatted data (i.e. no decimal point and no text characters) by using a "**U**" or "**u**" in the read command instead of an "**R**" or "**r**". The following command sequence would be used to read unformatted data in channel 4 from controller address 3.

S3U15*

NOTE: There is no unformatted write command. When writing to fixed point registers, any decimal point and text characters are ignored.

See also
ASCII Read/Write Examples

Multiple Write

ASCII Text Registers

1.4.1 ASCII Read/Write Examples

Examples	Description
SR$	Read **display** value, 50 milliseconds delay, all controllers respond.
s15r$	Read **display** value, 50 milliseconds delay, controller address 15 responds.
SR57*	Read **Peak** value, 2 milliseconds delay, all controllers respond.
Sr8194*	Read **Code 1** setting, 2 milliseconds delay, all controllers respond.
s2w1 -10000$	Write **-10000** to the display register of controller address 2, 50 milliseconds delay.
SW16393 Chan_1$	Write ASCII text string **Chan_1** to channel 1 text register, 50 milliseconds.
s10w8206,7*	Change brightness to **7** on controller address 8206, 2 milliseconds delay.

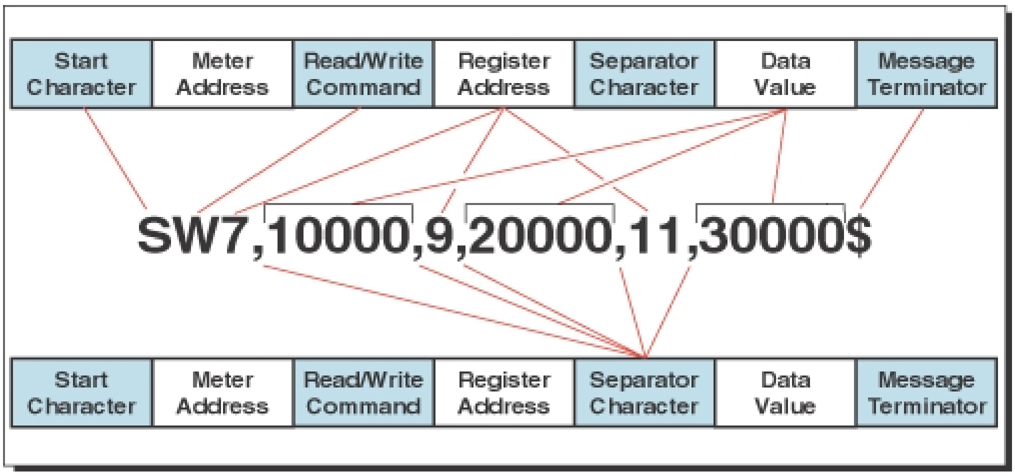
See also
Multiple Write

ASCII Text Registers

1.4.2 Multiple Write

This feature allows multiple registers to be written in a single ASCII command string. It is similar to a normal write command with the following differences:

- After the first data value, a separator character is inserted instead of the message terminator. Then the next register address is specified, followed by another separator character and the next data value. This procedure is repeated for each new register. The message terminator is added after the last data value in the string.
- Any number of registers can be written in the above manner provided the total length of the command string does not exceed 73 ASCII characters, including spaces and message terminator.



Two examples of the multiple write command

See also
ASCII Text Registers

1.5 Macro Compiling & Uploading

A macro is a set of commands that run automatically when the controller is powered up. Define Instruments Ltd. has a growing library of macros to suit a wide range of customer applications. Macros can be installed in the controller at the factory during initial programming or by the customer at some later date. Macros are written by Define Instruments Ltd. or the customer using the DDS, which is available for free download at www.defineinstruments.com.

Part II

2 Register List

The registers described in the topics of this Help are available for controller configuration and macro programming purposes. Each register is identified in four ways, by:

Name
This is the name of the register and relates to its function.

Description
This describes the function of each register.

Symbol Type
Under Symbol Type, the following abbreviations identify the register type:

B_	The symbol B_ is followed by a number from 0 up to 31 and describes the register bit number.
F_32	The symbol F_32 identifies the register as a floating point 32-bit register (IEEE-754). (Modbus word order is Little Endian)
PF_32	The symbol PF_32 identifies the register as a pseudo floating point 32-bit register (IEEE-754). (Modbus word order is Little Endian) (See 32-bit Pseudo Floating Point (1537 to 2047)).
SF_32	The symbol SF_32 identifies the register as a swapped floating point 32-bit register (IEEE-754). When accessing via Modbus word order is Big Endian. This register type was included to maintain backwards compatibility with older Intech products.
_R	The symbol _R identifies the register as a read only register and may be attached to another symbol. For example, B_0_R identifies this as bit 0 read only.
S_	The symbol S_ is followed by either 16, 24, or 32, identifying the register as a 16, 24, or 32-bit signed integer.
U_	The symbol U_ is followed by either 8, 16, or 32 identifying the register as an 8, 16, or 32-bit unsigned integer.
O_	The symbol O_ is followed by either 8, 16, or 32 identifying the register as an 8, 16, or 32-bit unsigned integer which is displayed in an octal format.
_W	The symbol _W identifies the register as a write only register.
L_	The symbol _L identifies the register as a text string that contains printable ASCII characters from 0x20 - 0x7a.

Register Number
This is the number that identifies the register in the controller.

See Also
Memory Types

2.1 ASCII Text Registers

The Zen IoT Series controller incorporates a number of text registers for storage of ASCII text strings. These strings vary in length from 8 to 62 characters depending on the intended function of the text register. The USER_TEXT and TEXT_VARIABLE registers are intended for macro use to store user text data.

Text registers can be accessed in the ASCII serial mode via the serial port or from the Macro.

Name	Description	Symbol Type	Register Number	Memory Type

RESULT_TEXT	Text display for Result	L_14	16391	EEPROM
CHANNEL1_TEXT	Text display for Channel 1	L_14	16393	EEPROM
CHANNEL2_TEXT	Text display for Channel 2	L_14	16395	EEPROM
CHANNEL3_TEXT	Text display for Channel 3	L_14	16397	EEPROM
CHANNEL4_TEXT	Text display for Channel 4	L_14	16399	EEPROM
CHANNEL5_TEXT	Text display for Channel 5	L_14	16401	EEPROM
CHANNEL6_TEXT	Text display for Channel 6	L_14	16403	EEPROM
CHANNEL7_TEXT	Text display for Channel 7	L_14	16405	EEPROM
CHANNEL8_TEXT	Text display for Channel 8	L_14	16407	EEPROM
CHANNEL9_TEXT	Text display for Channel 9	L_14	16409	EEPROM
CHANNEL10_TEXT	Text display for Channel 10	L_14	16411	EEPROM
CHANNEL11_TEXT	Text display for Channel 11	L_14	16413	EEPROM
CHANNEL12_TEXT	Text display for Channel 12	L_14	16415	EEPROM
CHANNEL13_TEXT	Text display for Channel 13	L_14	16417	EEPROM
CHANNEL14_TEXT	Text display for Channel 14	L_14	16419	EEPROM
CHANNEL15_TEXT	Text display for Channel 15	L_14	16421	EEPROM
CHANNEL16_TEXT	Text display for Channel 16	L_14	16423	EEPROM

COUNTER_A_TEXT	Text display for Counter A	L_14	16427	EEPROM
COUNTER_B_TEXT	Text display for Counter B	L_14	16429	EEPROM
COUNTER_C_TEXT	Text display for Counter C	L_14	16431	EEPROM
COUNTER_D_TEXT	Text display for Counter D	L_14	16433	EEPROM
TOTAL1_TEXT	Text display for Totalizer 1	L_14	16437	EEPROM
TOTAL2_TEXT	Text display for Totalizer 2	L_14	16439	EEPROM
TOTAL3_TEXT	Text display for Totalizer 3	L_14	16441	EEPROM
TOTAL4_TEXT	Text display for Totalizer 4	L_14	16443	EEPROM
TOTAL5_TEXT	Text display for Totalizer 5	L_14	16445	EEPROM
TOTAL6_TEXT	Text display for Totalizer 6	L_14	16447	EEPROM
TOTAL7_TEXT	Text display for Totalizer 7	L_14	16449	EEPROM
TOTAL8_TEXT	Text display for Totalizer 8	L_14	16451	EEPROM
TOTAL9_TEXT	Text display for Totalizer 9	L_14	16453	EEPROM
TOTAL10_TEXT	Text display for Totalizer 10	L_14	16455	EEPROM
AUX1_TEXT	Text display for Auxiliary 1	L_14	16463	EEPROM
AUX2_TEXT	Text display for Auxiliary 2	L_14	16465 16463	EEPROM
AUX3_TEXT	Text display for Auxiliary 3	L_14	16467 16463	EEPROM
AUX4_TEXT	Text display for Auxiliary 4	L_14	16469 16463	EEPROM
AUX5_TEXT	Text display for Auxiliary 5	L_14	16471	EEPROM
AUX6_TEXT	Text display for Auxiliary 6	L_14	16473	EEPROM
AUX7_TEXT	Text display for Auxiliary 7	L_14	16475	EEPROM
AUX8_TEXT	Text display for Auxiliary 8	L_14	16477	EEPROM
AUX9_TEXT	Text display for Auxiliary 9	L_14	16479	EEPROM
AUX10_TEXT	Text display for Auxiliary 10	L_14	16481	EEPROM
AUX11_TEXT	Text display for Auxiliary 11	L_14	16483	EEPROM
AUX12_TEXT	Text display for Auxiliary 12	L_14	16485	EEPROM
AUX13_TEXT	Text display for Auxiliary 13	L_14	16487	EEPROM
AUX14_TEXT	Text display for Auxiliary 14	L_14	16489	EEPROM
AUX15_TEXT	Text display for Auxiliary 15	L_14	16491	EEPROM
AUX16_TEXT	Text display for Auxiliary 16	L_14	16493	EEPROM
SETPOINT1_TEXT	Text display for Setpoint 1	L_14	16495	EEPROM
SETPOINT2_TEXT	Text display for Setpoint 2	L_14	16497	EEPROM
SETPOINT3_TEXT	Text display for Setpoint 3	L_14	16499	EEPROM
SETPOINT4_TEXT	Text display for Setpoint 4	L_14	16501	EEPROM
SETPOINT5_TEXT	Text display for Setpoint 5	L_14	16503	EEPROM
SETPOINT6_TEXT	Text display for Setpoint 6	L_14	16505	EEPROM
SETPOINT7_TEXT	Text display for Setpoint 7	L_14	16507	EEPROM
SETPOINT8_TEXT	Text display for Setpoint 8	L_14	16509	EEPROM
SETPOINT9_TEXT	Text display for Setpoint 9	L_14	16511	EEPROM
SETPOINT10_TEXT	Text display for Setpoint 10	L_14	16513	EEPROM
SETPOINT11_TEXT	Text display for Setpoint 11	L_14	16515	EEPROM
SETPOINT12_TEXT	Text display for Setpoint 12	L_14	16517	EEPROM

PEAK1_TEXT	Text display for Peak	L_14	16527	EEPROM
VALLEY1_TEXT	Text display for Valley	L_14	16529	EEPROM
PEAK2_TEXT	Text display for Peak 2	L_14	16531	EEPROM
VALLEY2_TEXT	Text display for Valley 2	L_14	16533	EEPROM
PEAK3_TEXT	Text display for Peak 3	L_14	16535	EEPROM
VALLEY3_TEXT	Text display for Valley 3	L_14	16537	EEPROM
OVER_TEXT	Text display for over range	L_14	16539	EEPROM
UNDER_TEXT	Text display for under range	L_14	16541	EEPROM
PRINT_STRING	Print String	L_62	16543	EEPROM
SINGLE_LOG	This register reads or write a single data log sample if data logging is enabled.	L_14	16553	EEPROM
METER_TYPE		L_14_R	16565	FLASH
USER_TEXT 1 to USER_TEXT64	Are all non-volatile 30 character text strings for user defined text storage, using only odd number register addresses from 16567 to 16693.	L_30	16567 to 16693	EEPROM
TEXT_VARIABLE1	30 character text string variable in RAM.	L_30	16897	RAM
TEXT_VARIABLE2	30 character text string variable in RAM.	L_30	16899	RAM
TEXT_VARIABLE3	30 character text string variable in RAM.	L_30	16901	RAM
TEXT_VARIABLE4	30 character text string variable in RAM.	L_30	16903	RAM
TEXT_VARIABLE5	30 character text string variable in RAM.	L_30	16905	RAM
TEXT_VARIABLE6	30 character text string variable in RAM.	L_30	16907	RAM
TEXT_VARIABLE7	30 character text string variable in RAM.	L_30	16909	RAM
TEXT_VARIABLE8	30 character text string variable in RAM.	L_30	16911	RAM
TEXT_VARIABLE9	30 character text string variable in RAM.	L_30	16913	RAM
TEXT_VARIABLE10	30 character text string variable in RAM.	L_30	16915	RAM
TEXT_VARIABLE11	30 character text string variable in RAM.	L_30	16917	RAM
TEXT_VARIABLE12	30 character text string variable in RAM.	L_30	16919	RAM
TEXT_VARIABLE13	30 character text string variable in RAM.	L_30	16921	RAM
TEXT_VARIABLE14	30 character text string variable in RAM.	L_30	16923	RAM
TEXT_VARIABLE15	30 character text string variable in RAM.	L_30	16925	RAM
TEXT_VARIABLE16	30 character text string variable in RAM.	L_30	16927	RAM

See also
Register 16385

ASCII Characters for 14-segment

Print String - Register 16543

2.1.1 Print String - Register 16543

When setup in the print mode, the controller can print data from any register directly to a serial printer, or to a PC where it can be imported into a spreadsheet.

Register 16543 is a special register that allows you to specify the text and data stored in specific registers to be printed out when a print command is issued by the controller while in the print mode. Through the serial port, register 16543 can be either written to or read from using a terminal program on a PC.

Writing To Register 16543
Writing to register 16543 tells the controller to print the data stored in one or more of the controller's registers when the print command is issued. To get the controller to print, the printer must be connected to the controller through the serial port and the controller must be programmed to run in the

print mode. The data to be printed depends on how the controller has been programmed.

For example, to display a flow rate and total. The total length of a write string can be up to 62 ASCII characters long. See Printing Restrictions.

Reading From Register 16543
Reading from register 16543 allows you to check your settings prior to removing the PC from the serial port and connecting to a printer. Register 16543 can be read in the normal manner: SR16543$.

Example of Writing To Register 16543
The following example shows a write to register 16543 with the controller setup to display flow rate and total flow of channel 1.

swx Rate = ~1 (add carriage return and line feed)
Total = ~37$

The above write to register 16543 means the following:

sw16543: Start writing to register 16543.
Rate =: Tells the controller to print the word Rate =.
~1: Tells the controller to print the current flow rate (display data), held in register 1, after the word Rate =.
Total =: Tells the controller to print the word Total =.
~37: Tells the controller to print the current total flow (stored data), held in register 37, after the word Flow =.

The printer would then print, for example, the following:

Rate = 2000
Total = 25000

This means that the current flow rate is 2000 and the total flow at this point is 25000.

Example of Reading From Register 16543
Having written the above example to the controller, to check the contents of register 16543 using the terminal program through the PC, type the following:

sr16543$

The following is shown on the PC screen:
Rate = ~1
Total = ~37$

Printing Restrictions
When printing, any alphanumeric ASCII character can be used within the following restrictions:

- The **$** and ***** characters are reserved for the terminating character at the end of the string and cannot be used as part of the text string.
- The total string length must be no greater than 62 bytes long. This includes spaces, tabs, carriage returns, line feeds, and the terminating character. There must be a separator space between the register address 16543 and the start of the string. **Note, this separator space does not have to be included in text string length calculations**.
- Any number following a **~** (tilde) character is interpreted as a register address. During a printout the register's current value is printed out in this position.
- The ASCII character **** is treated as a special character in the print string. When a **** is encountered, a ***** is printed in its place (***** is reserved as a terminating character and normally can not appear anywhere in the text string). This allows the print output of one controller to be connected to another controller that is operating in the ASCII mode.

For example, if the print string reads:

swx sw3 ~11\ sw4 ~13\ sw6 ~1\$

The printer prints the following:

sw3 (current register value)* **sw4** (current register value)*
sw6 (current register value)*

Up to seven different registers can be specified in one text string, provided that the total string length is no greater than 62 bytes long and the total length of the resulting printout is less than 100 bytes long (including time stamp if selected).

For example, the following tab delimited output could be specified to input display data, processed result, processed channel 1, processed channel 2, peak, valley, and total, directly into a spreadsheet:

swx ~1(tab)~7(tab)~9(tab)~11(tab)~57(tab)~59(tab)~37$

When calculating the length of the printout, an allowance of 7 bytes for each register address should be used, plus any extra text or separating characters such as tabs or spaces.

NOTE: As a new line is usually represented by a carriage return and a line feed, 2 bytes should be added for each new line in text string length calculations.

2.2 Analog Inputs

The Zen IoT controller can have up to 16 analog input modules fitted. These can be configured for a variety of different input sensors including RTD temperature probes, thermocouple temperature probes, voltage and current measurement and counters. Input modules can be isolated or non-isolated types. Each type of input module will have different setup and configuration requirements which need to be adhered to for correct functionality.

However, regardless of which type of input module is fitted, the Zen IoT main controller will poll all input modules and create an updated copy of the result and status registers of each input module. This data is then scaled and becomes available in the analog result registers shown in this section.

This section also shows the various configuration registers associated with the result data.

NOTE: The configuration of each input module must be done separately via the index register 8224.

2.2.1 Channel 1

Channel 1 registers can be selected as the data source for:

- The primary display.
- The second display, if installed.
- The third display, if installed.
- Trigger for advanced setpoints SP1 to SP16 (integer registers only)
- Analogue output channels (integer registers only).
- Setpoint reset destination (integer registers only).
  The reset destination mode allows you to select a register to be reset using the contents of another register triggered by a setpoint.

Name	Description	Symbol Type	Register Number	Memory Type
CH1	32-bit register that holds the processed data for CH1.	S_32	645	RAM/FLASH
CH1_RAW	32-bit register that holds the raw data for CH1. **Note**: When input module is operating in counter mode, this register shows the raw accumulated count value.	S_32	677	RAM/FLASH
CH1_FLOAT	32-bit register that holds the CH1 data in a floating point format. Scaling and decimal point values are based on those used for the CH1 data shown above.	F_32	1193	RAM
CH1_SWAPPED_FLOAT	32-bit register that holds the CH1 data in a floating point format. Scaling and decimal point values are based on those used for the CH1 data shown above. **Note**: This register is used to maintain backwards compatibility with older Intech products. When reading this register via Modbus the word order is Big Endian.	SF_32	17	RAM
CH1_12	12-bit register that holds the processed data for CH1. (Range from 0 - 4095) This register is used to maintain backwards compatibility with older Intech products.	U_12	1	RAM
IM_STATUS1	16-bit unsigned register that holds the input module status for CH1. (See Input Module Status)	U_16	4592	RAM

The above registers are normally updated by the operating system of the controller after a new input sample is processed. If the channel is disabled or in a counter mode, it is also possible to modify the contents of the register by writing to it from the setpoint reset logic, from the Macro, or via the serial port. A write to these registers in any other operational mode may result in the newly written value being overwritten by the operating system in the controller.

See also
CH1 Setup Registers

2.2.1.1 CH1 Setup Registers

Name	Description	Symbol Type	Register Number	Memory Type
OFFSET_CH1	32-bit register. Holds the calibration offset for CH1 and CH1_FLOAT.	S_32	613	RAM/EEPROM
SCALE_FACTOR_CH1	32-bit floating point register. Holds the calibration scale factor for CH1 and CH1_FLOAT.	F_32	1097	RAM/EEPROM
OFFSET_CH1_12BIT	16-bit register. Holds the calibration offset for CH1_12.	S_16	4576	RAM/EEPROM
SCALE_FACTOR_CH1_12BIT	32-bit floating point register. Holds the calibration scale factor for CH1_12.	F_32	1161	RAM/EEPROM
CHANNEL1_TEXT	Text display for CH1.	L_14	16393	EEROM
UNITS_TEXT_CH1	Units text for CH1. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17409	EEROM
DISPLAY_FORMAT_CH1	8-bit register. Controls the display format settings for CH1 (displayed in octal format).	O_8	8321	RAM/EEPROM
TEXT_CHARACTER_CH1	8-bit register. Holds the ASCII value for the last digit text character for CH1 (0 = no character).	U_8	8375	RAM/EEPROM

See also
Channel 1

2.2.2 Channel 2

Channel 2 registers can be selected as the data source for:

- The primary display.
- The second display, if installed.
- The third display, if installed.
- Trigger for advanced setpoints SP1 to SP16 (integer registers only)
- Analogue output channels (integer registers only).
- Setpoint reset destination (integer registers only).
  The reset destination mode allows you to select a register to be reset using the contents of another register triggered by a setpoint.

Name	Description	Symbol Type	Register Number	Memory Type
CH2	32-bit register that holds the processed data for CH2.	S_32	647	RAM/FLASH
CH2_RAW	32-bit register that holds the raw data for CH2. **Note**: When input module is operating in counter mode, this register shows the raw accumulated count value.	S_32	679	RAM/FLASH
CH2_FLOAT	32-bit register that holds the CH2 data in a floating point format. Scaling and decimal point values are based on those used for the CH2 data shown above.	F_32	1195	RAM
CH2_SWAPPED_FLOAT	32-bit register that holds the CH2 data in a floating point format. Scaling and decimal point values are based on those used for the CH2 data shown above. **Note**:This register is used to maintain backwards compatibility with older Intech products. When reading this register via Modbus the word order is Big Endian.	SF_32	19	RAM
CH2_12	12-bit register that holds the processed data for CH2. (Range from 0 - 4095) This register is used to maintain backwards compatibility with older Intech products.	U_12	2	RAM
IM_STATUS2	16-bit unsigned register that holds the input module status for CH2. (See Input Module Status)	U_16	4593	RAM

The above registers are normally updated by the operating system of the controller after a new input sample is processed. If the channel is disabled or in a counter mode, it is also possible to modify the contents of the register by writing to it from the setpoint reset logic, from the Macro, or via the serial port. A write to these registers in any other operational mode may result in the newly written value being overwritten by the operating system in the controller.

See also
CH2 Setup Registers

2.2.2.1 CH2 Setup Registers

Name	Description	Symbol Type	Register Number	Memory Type
OFFSET_CH2	32-bit register. Holds the calibration offset for CH2 and CH2_FLOAT.	S_32	615	RAM/EEPROM
SCALE_FACTOR_CH2	32-bit floating point register. Holds the calibration scale factor for CH2 and CH2_FLOAT.	F_32	1099	RAM/EEPROM
OFFSET_CH2_12BIT	16-bit register. Holds the calibration offset for CH2_12.	S_16	4577	RAM/EEPROM
SCALE_FACTOR_CH2_12BIT	32-bit floating point register. Holds the calibration scale factor for CH2_12.	F_32	1163	RAM/EEPROM

CHANNEL2_TEXT	Text display for CH2.	L_14	16395	EEROM
UNITS_TEXT_CH2	Units text for CH2. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17411	EEROM
DISPLAY_FORMAT_CH2	8-bit register. Controls the display format settings for CH2 (displayed in octal format).	O_8	8322	RAM/EEPROM
TEXT_CHARACTER_CH2	8-bit register. Holds the ASCII value for the last digit text character for CH2 (0 = no character).	U_8	8376	RAM/EEPROM

See also
Channel 2

2.2.3 Channel 3

Channel 3 registers can be selected as the data source for:

- The primary display.
- The second display, if installed.
- The third display, if installed.
- Trigger for advanced setpoints SP1 to SP16 (integer registers only)
- Analogue output channels (integer registers only).
- Setpoint reset destination (integer registers only).
  The reset destination mode allows you to select a register to be reset using the contents of another register triggered by a setpoint.

Name	Description	Symbol Type	Register Number	Memory Type
CH3	32-bit register that holds the processed data for CH3.	S_32	649	RAM/FLASH
CH3_RAW	32-bit register that holds the raw data for CH3. **Note**: When input module is operating in counter mode, this register shows the raw accumulated count value.	S_32	681	RAM/FLASH
CH3_FLOAT	32-bit register that holds the CH3 data in a floating point format. Scaling and decimal point values are based on those used for the CH3 data shown above.	F_32	1197	RAM
CH3_SWAPPED_FLOAT	32-bit register that holds the CH3 data in a floating point format. Scaling and decimal point values are based on those used for the CH3 data shown above. **Note**: This register is used to maintain backwards compatibility with older Intech products. When reading this register via Modbus the word order is Big Endian.	SF_32	21	RAM
CH3_12	12-bit register that holds the processed data for CH3. (Range from 0 - 4095) This register is used to maintain backwards compatibility with older Intech products.	U_12	3	RAM
IM_STATUS3	16-bit unsigned register that holds the input module status for CH3. (See Input Module Status)	U_16	4594	RAM

The above registers are normally updated by the operating system of the controller after a new input sample is processed. If the channel is disabled or in a counter mode, it is also possible to modify the contents of the register by writing to it from the setpoint reset logic, from the Macro, or via the serial port. A write to these registers in any other operational mode may result in the newly written value being overwritten by the operating system in the controller.

See also
CH3 Setup Registers

2.2.3.1 CH3 Setup Registers

Name	Description	Symbol Type	Register Number	Memory Type
OFFSET_CH3	32-bit register. Holds the calibration offset for CH3 and CH3_FLOAT.	S_32	617	RAM/EEPROM
SCALE_FACTOR_CH3	32-bit floating point register. Holds the calibration scale factor for CH3 and CH3_FLOAT.	F_32	1101	RAM/EEPROM
OFFSET_CH3_12BIT	16-bit register. Holds the calibration offset for CH3_12.	S_16	4578	RAM/EEPROM
SCALE_FACTOR_CH3_12BIT	32-bit floating point register. Holds the calibration scale factor for CH3_12.	F_32	1165	RAM/EEPROM
CHANNEL3_TEXT	Text display for CH3.	L_14	16397	EEROM
UNITS_TEXT_CH3	Units text for CH3. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17413	EEROM
DISPLAY_FORMAT_CH3	8-bit register. Controls the display format settings for CH3 (displayed in octal format).	O_8	8323	RAM/EEPROM
TEXT_CHARACTER_CH3	8-bit register. Holds the ASCII value for the last digit text character for CH3 (0 = no character).	U_8	8377	RAM/EEPROM

See also
Channel 3

2.2.4 Channel 4

Channel 4 registers can be selected as the data source for:

- The primary display.
- The second display, if installed.
- The third display, if installed.
- Trigger for advanced setpoints SP1 to SP16 (integer registers only)
- Analogue output channels (integer registers only).
- Setpoint reset destination (integer registers only).
  The reset destination mode allows you to select a register to be reset using the contents of another register triggered by a setpoint.

Name	Description	Symbol Type	Register Number	Memory Type
CH4	32-bit register that holds the processed data for CH4.	S_32	651	RAM/FLASH
CH4_RAW	32-bit register that holds the raw data for CH4. **Note**: When input module is operating in counter mode, this register shows the raw accumulated count value.	S_32	683	RAM/FLASH
CH4_FLOAT	32-bit register that holds the CH4 data in a floating point format. Scaling and decimal point values are based on those used for the CH4 data shown above.	F_32	1199	RAM

CH4_SWAPPED_FLOAT	32-bit register that holds the CH4 data in a floating point format. Scaling and decimal point values are based on those used for the CH4 data shown above. **Note**:This register is used to maintain backwards compatibility with older Intech products. When reading this register via Modbus the word order is Big Endian.	SF_32	23	RAM
CH4_12	12-bit register that holds the processed data for CH4. (Range from 0 - 4095) This register is used to maintain backwards compatibility with older Intech products.	U_12	4	RAM
IM_STATUS4	16-bit unsigned register that holds the input module status for CH4. (See Input Module Status)	U_16	4595	RAM

The above registers are normally updated by the operating system of the controller after a new input sample is processed. If the channel is disabled or in a counter mode, it is also possible to modify the contents of the register by writing to it from the setpoint reset logic, from the Macro, or via the serial port. A write to these registers in any other operational mode may result in the newly written value being overwritten by the operating system in the controller.

See also
CH4 Setup Registers

2.2.4.1 CH4 Setup Registers

Name	Description	Symbol Type	Register Number	Memory Type
OFFSET_CH4	32-bit register. Holds the calibration offset for CH4 and CH4_FLOAT.	S_32	619	RAM/EEPROM
SCALE_FACTOR_CH4	32-bit floating point register. Holds the calibration scale factor for CH4 and CH4_FLOAT.	F_32	1103	RAM/EEPROM
OFFSET_CH4_12BIT	16-bit register. Holds the calibration offset for CH4_12.	S_16	4579	RAM/EEPROM
SCALE_FACTOR_CH4_12BIT	32-bit floating point register. Holds the calibration scale factor for CH4_12.	F_32	1167	RAM/EEPROM
CHANNEL4_TEXT	Text display for CH4.	L_14	16399	EEROM
UNITS_TEXT_CH4	Units text for CH4. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17415	EEROM
DISPLAY_FORMAT_CH4	8-bit register. Controls the display format settings for CH4 (displayed in octal format).	O_8	8324	RAM/EEPROM
TEXT_CHARACTER_CH4	8-bit register. Holds the ASCII value for the last digit text character for CH4 (0 = no character).	U_8	8378	RAM/EEPROM

See also
Channel 4

2.2.5 Channel 5

Channel 5 registers can be selected as the data source for:

- The primary display.
- The second display, if installed.
- The third display, if installed.
- Trigger for advanced setpoints SP1 to SP16 (integer registers only)
- Analogue output channels (integer registers only).
- Setpoint reset destination (integer registers only).
  The reset destination mode allows you to select a register to be reset using the contents of another register triggered by a setpoint.

Name	Description	Symbol Type	Register Number	Memory Type
CH5	32-bit register that holds the processed data for CH5.	S_32	653	RAM/FLASH
CH5_RAW	32-bit register that holds the raw data for CH5. **Note**: When input module is operating in counter mode, this register shows the raw accumulated count value.	S_32	685	RAM/FLASH
CH5_FLOAT	32-bit register that holds the CH5 data in a floating point format. Scaling and decimal point values are based on those used for the CH5 data shown above.	F_32	1201	RAM
CH5_SWAPPED_FLOAT	32-bit register that holds the CH5 data in a floating point format. Scaling and decimal point values are based on those used for the CH5 data shown above. **Note**: This register is used to maintain backwards compatibility with older Intech products. When reading this register via Modbus the word order is Big Endian.	SF_32	25	
CH5_12	12-bit register that holds the processed data for CH5. (Range from 0 - 4095) This register is used to maintain backwards compatibility with older Intech products.	U_12	5	RAM
IM_STATUS5	16-bit unsigned register that holds the input module status for CH5. (See Input Module Status)	U_16	4596	RAM

The above registers are normally updated by the operating system of the controller after a new input sample is processed. If the channel is disabled or in a counter mode, it is also possible to modify the contents of the register by writing to it from the setpoint reset logic, from the Macro, or via the serial port. A write to these registers in any other operational mode may result in the newly written value being overwritten by the operating system in the controller.

See also
CH5 Setup Registers

2.2.5.1 CH5 Setup Registers

Name	Description	Symbol Type	Register Number	Memory Type
OFFSET_CH5	32-bit register. Holds the calibration offset for CH5 and CH5_FLOAT.	S_32	621	RAM/EEPROM
SCALE_FACTOR_CH5	32-bit floating point register. Holds the calibration scale factor for CH5 and CH5_FLOAT.	F_32	1105	RAM/EEPROM
OFFSET_CH5_12BIT	16-bit register. Holds the calibration offset for CH5_12.	S_16	4580	RAM/EEPROM
SCALE_FACTOR_CH5_12BIT	32-bit floating point register. Holds the calibration scale factor for CH5_12.	F_32	1169	RAM/EEPROM
CHANNEL5_TEXT	Text display for CH5.	L_14	16401	EEROM
UNITS_TEXT_CH5	Units text for CH5. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17417	EEROM
DISPLAY_FORMAT_CH5	8-bit register. Controls the display format settings for CH5 (displayed in octal format).	O_8	8325	RAM/EEPROM
TEXT_CHARACTER_CH5	8-bit register. Holds the ASCII value for the last digit text character for CH5 (0 = no character).	U_8	8379	RAM/EEPROM

See also
Channel 5

2.2.6 Channel 6

Channel 6 registers can be selected as the data source for:

- The primary display.
- The second display, if installed.
- The third display, if installed.
- Trigger for advanced setpoints SP1 to SP16 (integer registers only)
- Analogue output channels (integer registers only).
- Setpoint reset destination (integer registers only).
  The reset destination mode allows you to select a register to be reset using the contents of another register triggered by a setpoint.

Name	Description	Symbol Type	Register Number	Memory Type
CH6	32-bit register that holds the processed data for CH6.	S_32	655	RAM/FLASH
CH6_RAW	32-bit register that holds the raw data for CH6. **Note**: When input module is operating in counter mode, this register shows the raw accumulated count value.	S_32	687	RAM/FLASH
CH6_FLOAT	32-bit register that holds the CH6 data in a floating point format. Scaling and decimal point values are based on those used for the CH6 data shown above.	F_32	1203	RAM
CH6_SWAPPED_FLOAT	32-bit register that holds the CH6 data in a floating point format. Scaling and decimal point values are based on those used for the CH6 data shown above. **Note**:This register is used to maintain backwards compatibility with older Intech products. When reading this register via Modbus the word order is Big Endian.	SF_32	27	RAM
CH6_12	12-bit register that holds the processed data for CH6. (Range from 0 - 4095) This register is used to maintain backwards compatibility with older Intech products.	U_12	6	RAM
IM_STATUS6	16-bit unsigned register that holds the input module status for CH6. (See Input Module Status)	U_16	4597	RAM

The above registers are normally updated by the operating system of the controller after a new input sample is processed. If the channel is disabled or in a counter mode, it is also possible to modify the contents of the register by writing to it from the setpoint reset logic, from the Macro, or via the serial port. A write to these registers in any other operational mode may result in the newly written value being overwritten by the operating system in the controller.

See also
CH6 Setup Registers

2.2.6.1 CH6 Setup Registers

Name	Description	Symbol Type	Register Number	Memory Type
OFFSET_CH6	32-bit register. Holds the calibration offset for CH6 and CH6_FLOAT.	S_32	623	RAM/EEPROM
SCALE_FACTOR_CH6	32-bit floating point register. Holds the calibration scale factor for CH6 and CH6_FLOAT.	F_32	1107	RAM/EEPROM
OFFSET_CH6_12BIT	16-bit register. Holds the calibration offset for CH6_12.	S_16	4581	RAM/EEPROM
SCALE_FACTOR_CH6_12BIT	32-bit floating point register. Holds the calibration scale factor for CH6_12.	F_32	1171	RAM/EEPROM

CHANNEL6_TEXT	Text display for CH6.	L_14	16403	EEROM
UNITS_TEXT_CH6	Units text for CH6. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17419	EEROM
DISPLAY_FORMAT_CH6	8-bit register. Controls the display format settings for CH6 (displayed in octal format).	O_8	8326	RAM/EEPROM
TEXT_CHARACTER_CH6	8-bit register. Holds the ASCII value for the last digit text character for CH6 (0 = no character).	U_8	8380	RAM/EEPROM

See also
Channel 6

2.2.7 Channel 7

Channel 7 registers can be selected as the data source for:

- The primary display.
- The second display, if installed.
- The third display, if installed.
- Trigger for advanced setpoints SP1 to SP16 (integer registers only)
- Analogue output channels (integer registers only).
- Setpoint reset destination (integer registers only).
  The reset destination mode allows you to select a register to be reset using the contents of another register triggered by a setpoint.

Name	Description	Symbol Type	Register Number	Memory Type
CH7	32-bit register that holds the processed data for CH7.	S_32	657	RAM/FLASH
CH7_RAW	32-bit register that holds the raw data for CH7. **Note**: When input module is operating in counter mode, this register shows the raw accumulated count value.	S_32	689	RAM/FLASH
CH7_FLOAT	32-bit register that holds the CH7 data in a floating point format. Scaling and decimal point values are based on those used for the CH7 data shown above.	F_32	1205	RAM
CH7_SWAPPED_FLOAT	32-bit register that holds the CH7 data in a floating point format. Scaling and decimal point values are based on those used for the CH7 data shown above. **Note**: This register is used to maintain backwards compatibility with older Intech products. When reading this register via Modbus the word order is Big Endian.	SF_32	29	RAM
CH7_12	12-bit register that holds the processed data for CH7. (Range from 0 - 4095) This register is used to maintain backwards compatibility with older Intech products.	U_12	7	RAM
IM_STATUS7	16-bit unsigned register that holds the input module status for CH7. (See Input Module Status)	U_16	4598	RAM

The above registers are normally updated by the operating system of the controller after a new input sample is processed. If the channel is disabled or in a counter mode, it is also possible to modify the contents of the register by writing to it from the setpoint reset logic, from the Macro, or via the serial port. A write to these registers in any other operational mode may result in the newly written value being overwritten by the operating system in the controller.

See also
CH7 Setup Registers

2.2.7.1 CH7 Setup Registers

Name	Description	Symbol Type	Register Number	Memory Type
OFFSET_CH7	32-bit register. Holds the calibration offset for CH7 and CH7_FLOAT.	S_32	625	RAM/EEPROM
SCALE_FACTOR_CH7	32-bit floating point register. Holds the calibration scale factor for CH7 and CH7_FLOAT.	F_32	1109	RAM/EEPROM
OFFSET_CH7_12BIT	16-bit register. Holds the calibration offset for CH7_12.	S_16	4582	RAM/EEPROM
SCALE_FACTOR_CH7_12BIT	32-bit floating point register. Holds the calibration scale factor for CH7_12.	F_32	1173	RAM/EEPROM
CHANNEL7_TEXT	Text display for CH7.	L_14	16405	EEROM
UNITS_TEXT_CH7	Units text for CH7. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17421	EEROM
DISPLAY_FORMAT_CH7	8-bit register. Controls the display format settings for CH7 (displayed in octal format).	O_8	8327	RAM/EEPROM
TEXT_CHARACTER_CH7	8-bit register. Holds the ASCII value for the last digit text character for CH7 (0 = no character).	U_8	8381	RAM/EEPROM

See also
Channel 7

2.2.8 Channel 8

Channel 8 registers can be selected as the data source for:

- The primary display.
- The second display, if installed.
- The third display, if installed.
- Trigger for advanced setpoints SP1 to SP16 (integer registers only)
- Analogue output channels (integer registers only).
- Setpoint reset destination (integer registers only).
  The reset destination mode allows you to select a register to be reset using the contents of another register triggered by a setpoint.

Name	Description	Symbol Type	Register Number	Memory Type
CH8	32-bit register that holds the processed data for CH8.	S_32	659	RAM/FLASH
CH8_RAW	32-bit register that holds the raw data for CH8. **Note**: When input module is operating in counter mode, this register shows the raw accumulated count value.	S_32	691	RAM/FLASH
CH8_FLOAT	32-bit register that holds the CH8 data in a floating point format. Scaling and decimal point values are based on those used for the CH8 data shown above.	F_32	1207	RAM

CH8_SWAPPED_FLOAT	32-bit register that holds the CH8 data in a floating point format. Scaling and decimal point values are based on those used for the CH8 data shown above. **Note**: This register is used to maintain backwards compatibility with older Intech products. When reading this register via Modbus the word order is Big Endian.	SF_32	31	RAM
CH8_12	12-bit register that holds the processed data for CH8. (Range from 0 - 4095) This register is used to maintain backwards compatibility with older Intech products.	U_12	8	RAM
IM_STATUS8	16-bit unsigned register that holds the input module status for CH8. (See Input Module Status)	U_16	4599	RAM

The above registers are normally updated by the operating system of the controller after a new input sample is processed. If the channel is disabled or in a counter mode, it is also possible to modify the contents of the register by writing to it from the setpoint reset logic, from the Macro, or via the serial port. A write to these registers in any other operational mode may result in the newly written value being overwritten by the operating system in the controller.

See also
CH8 Setup Registers

2.2.8.1 CH8 Setup Registers

Name	Description	Symbol Type	Register Number	Memory Type
OFFSET_CH8	32-bit register. Holds the calibration offset for CH8 and CH8_FLOAT.	S_32	627	RAM/EEPROM
SCALE_FACTOR_CH8	32-bit floating point register. Holds the calibration scale factor for CH8 and CH8_FLOAT.	F_32	1111	RAM/EEPROM
OFFSET_CH8_12BIT	16-bit register. Holds the calibration offset for CH8_12.	S_16	4583	RAM/EEPROM
SCALE_FACTOR_CH8_12BIT	32-bit floating point register. Holds the calibration scale factor for CH8_12.	F_32	1175	RAM/EEPROM
CHANNEL8_TEXT	Text display for CH8.	L_14	16407	EEROM
UNITS_TEXT_CH8	Units text for CH8. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17423	EEROM
DISPLAY_FORMAT_CH8	8-bit register. Controls the display format settings for CH8 (displayed in octal format).	O_8	8328	RAM/EEPROM
TEXT_CHARACTER_CH8	8-bit register. Holds the ASCII value for the last digit text character for CH8 (0 = no character).	U_8	8382	RAM/EEPROM

See also
Channel 8

2.2.9 Channel 9

Channel 9 registers can be selected as the data source for:

- The primary display.
- The second display, if installed.
- The third display, if installed.
- Trigger for advanced setpoints SP1 to SP16 (integer registers only)
- Analogue output channels (integer registers only).
- Setpoint reset destination (integer registers only).

The reset destination mode allows you to select a register to be reset using the contents of another register triggered by a setpoint.

Name	Description	Symbol Type	Register Number	Memory Type
CH9	32-bit register that holds the processed data for CH9.	S_32	661	RAM/FLASH
CH9_RAW	32-bit register that holds the raw data for CH9. **Note**: When input module is operating in counter mode, this register shows the raw accumulated count value.	S_32	693	RAM/FLASH
CH9_FLOAT	32-bit register that holds the CH9 data in a floating point format. Scaling and decimal point values are based on those used for the CH9 data shown above.	F_32	1209	RAM
CH9_SWAPPED_FLOAT	32-bit register that holds the CH9 data in a floating point format. Scaling and decimal point values are based on those used for the CH9 data shown above. **Note**: This register is used to maintain backwards compatibility with older Intech products. When reading this register via Modbus the word order is Big Endian.	SF_32	33	RAM
CH9_12	12-bit register that holds the processed data for CH9. (Range from 0 - 4095) This register is used to maintain backwards compatibility with older Intech products.	U_12	9	RAM
IM_STATUS9	16-bit unsigned register that holds the input module status for CH9. (See Input Module Status)	U_16	4600	RAM

The above registers are normally updated by the operating system of the controller after a new input sample is processed. If the channel is disabled or in a counter mode, it is also possible to modify the contents of the register by writing to it from the setpoint reset logic, from the Macro, or via the serial port. A write to these registers in any other operational mode may result in the newly written value being overwritten by the operating system in the controller.

See also
CH9 Setup Registers

2.2.9.1 CH9 Setup Registers

Name	Description	Symbol Type	Register Number	Memory Type
OFFSET_CH9	32-bit register. Holds the calibration offset for CH9 and CH9_FLOAT.	S_32	629	RAM/EEPROM
SCALE_FACTOR_CH9	32-bit floating point register. Holds the calibration scale factor for CH9 and CH9_FLOAT.	F_32	1113	RAM/EEPROM
OFFSET_CH9_12BIT	16-bit register. Holds the calibration offset for CH9_12.	S_16	4584	RAM/EEPROM
SCALE_FACTOR_CH9_12BIT	32-bit floating point register. Holds the calibration scale factor for CH9_12.	F_32	1177	RAM/EEPROM
CHANNEL9_TEXT	Text display for CH9.	L_14	16409	EEROM
UNITS_TEXT_CH9	Units text for CH9. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17425	EEROM
DISPLAY_FORMAT_CH9	8-bit register. Controls the display format settings for CH9 (displayed in octal format).	O_8	8329	RAM/EEPROM
TEXT_CHARACTER_CH9	8-bit register. Holds the ASCII value for the last digit text character for CH9 (0 = no character).	U_8	8383	RAM/EEPROM

See also
Channel 9

2.2.10 Channel 10

Channel 10 registers can be selected as the data source for:

- The primary display.
- The second display, if installed.
- The third display, if installed.
- Trigger for advanced setpoints SP1 to SP16 (integer registers only)
- Analogue output channels (integer registers only).
- Setpoint reset destination (integer registers only).
  The reset destination mode allows you to select a register to be reset using the contents of another register triggered by a setpoint.

Name	Description	Symbol Type	Register Number	Memory Type
CH10	32-bit register that holds the processed data for CH10.	S_32	663	RAM/FLASH
CH10_RAW	32-bit register that holds the raw data for CH10. **Note**: When input module is operating in counter mode, this register shows the raw accumulated count value.	S_32	695	RAM/FLASH
CH10_FLOAT	32-bit register that holds the CH10 data in a floating point format. Scaling and decimal point values are based on those used for the CH10 data shown above.	F_32	1211	RAM
CH10_SWAPPED_FLOAT	32-bit register that holds the CH10 data in a floating point format. Scaling and decimal point values are based on those used for the CH10 data shown above. **Note**: This register is used to maintain backwards compatibility with older Intech products. When reading this register via Modbus the word order is Big Endian.	SF_32	35	RAM
CH10_12	12-bit register that holds the processed data for CH10. (Range from 0 - 4095) This register is used to maintain backwards compatibility with older Intech products.	U_12	10	RAM
IM_STATUS10	16-bit unsigned register that holds the input module status for CH10. (See Input Module Status)	U_16	4601	RAM

The above registers are normally updated by the operating system of the controller after a new input sample is processed. If the channel is disabled or in a counter mode, it is also possible to modify the contents of the register by writing to it from the setpoint reset logic, from the Macro, or via the serial port. A write to these registers in any other operational mode may result in the newly written value being overwritten by the operating system in the controller.

See also
CH10 Setup Registers

2.2.10.1 CH10 Setup Registers

Name	Description	Symbol Type	Register Number	Memory Type
OFFSET_CH10	32-bit register. Holds the calibration offset for CH10 and CH10_FLOAT.	S_32	631	RAM/EEPROM
SCALE_FACTOR_CH10	32-bit floating point register. Holds the calibration scale factor for CH10 and CH10_FLOAT.	F_32	1115	RAM/EEPROM
OFFSET_CH10_12BIT	16-bit register. Holds the calibration offset for CH10_12.	S_16	4585	RAM/EEPROM
SCALE_FACTOR_CH10_12BIT	32-bit floating point register. Holds the calibration scale factor for CH10_12.	F_32	1179	RAM/EEPROM
CHANNEL10_TEXT	Text display for CH10.	L_14	16411	EEROM
UNITS_TEXT_CH10	Units text for CH10. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17427	EEROM
DISPLAY_FORMAT_CH10	8-bit register. Controls the display format settings for CH10 (displayed in octal format).	O_8	8330	RAM/EEPROM
TEXT_CHARACTER_CH10	8-bit register. Holds the ASCII value for the last digit text character for CH10 (0 = no character).	U_8	8384	RAM/EEPROM

See also
Channel 10

2.2.11 Channel 11

Channel 11 registers can be selected as the data source for:

- The primary display.
- The second display, if installed.
- The third display, if installed.
- Trigger for advanced setpoints SP1 to SP16 (integer registers only)
- Analogue output channels (integer registers only).
- Setpoint reset destination (integer registers only).
  The reset destination mode allows you to select a register to be reset using the contents of another register triggered by a setpoint.

Name	Description	Symbol Type	Register Number	Memory Type
CH11	32-bit register that holds the processed data for CH11.	S_32	665	RAM/FLASH
CH11_RAW	32-bit register that holds the raw data for CH11. **Note**: When input module is operating in counter mode, this register shows the raw accumulated count value.	S_32	697	RAM/FLASH
CH11_FLOAT	32-bit register that holds the CH11 data in a floating point format. Scaling and decimal point values are based on those used for the CH11 data shown above.	F_32	1213	RAM

CH11_SWAPPED_FLOAT	32-bit register that holds the CH11 data in a floating point format. Scaling and decimal point values are based on those used for the CH11 data shown above. **Note**: This register is used to maintain backwards compatibility with older Intech products. When reading this register via Modbus the word order is Big Endian.	SF_32	37	RAM
CH11_12	12-bit register that holds the processed data for CH11. (Range from 0 - 4095) This register is used to maintain backwards compatibility with older Intech products.	U_12	11	RAM
IM_STATUS11	16-bit unsigned register that holds the input module status for CH11. (See Input Module Status)	U_16	4602	RAM

The above registers are normally updated by the operating system of the controller after a new input sample is processed. If the channel is disabled or in a counter mode, it is also possible to modify the contents of the register by writing to it from the setpoint reset logic, from the Macro, or via the serial port. A write to these registers in any other operational mode may result in the newly written value being overwritten by the operating system in the controller.

See also
CH11 Setup Registers

2.2.11.1 CH11 Setup Registers

Name	Description	Symbol Type	Register Number	Memory Type
OFFSET_CH11	32-bit register. Holds the calibration offset for CH11 and CH11_FLOAT.	S_32	633	RAM/EEPROM
SCALE_FACTOR_CH11	32-bit floating point register. Holds the calibration scale factor for CH11 and CH11_FLOAT.	F_32	1117	RAM/EEPROM
OFFSET_CH11_12BIT	16-bit register. Holds the calibration offset for CH11_12.	S_16	4586	RAM/EEPROM
SCALE_FACTOR_CH11_12BIT	32-bit floating point register. Holds the calibration scale factor for CH11_12.	F_32	1181	RAM/EEPROM
CHANNEL11_TEXT	Text display for CH11.	L_14	16413	EEROM
UNITS_TEXT_CH11	Units text for CH11. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17429	EEROM
DISPLAY_FORMAT_CH11	8-bit register. Controls the display format settings for CH11 (displayed in octal format).	O_8	8331	RAM/EEPROM
TEXT_CHARACTER_CH11	8-bit register. Holds the ASCII value for the last digit text character for CH11 (0 = no character).	U_8	8385	RAM/EEPROM

See also
Channel 11

2.2.12 Channel 12

Channel 12 registers can be selected as the data source for:

- The primary display.
- The second display, if installed.

- The third display, if installed.
- Trigger for advanced setpoints SP1 to SP16 (integer registers only)
- Analogue output channels (integer registers only).
- Setpoint reset destination (integer registers only).
  The reset destination mode allows you to select a register to be reset using the contents of another register triggered by a setpoint.

Name	Description	Symbol Type	Register Number	Memory Type
CH12	32-bit register that holds the processed data for CH12.	S_32	667	RAM/FLASH
CH12_RAW	32-bit register that holds the raw data for CH12. **Note**: When input module is operating in counter mode, this register shows the raw accumulated count value.	S_32	699	RAM/FLASH
CH12_FLOAT	32-bit register that holds the CH12 data in a floating point format. Scaling and decimal point values are based on those used for the CH12 data shown above.	F_32	1215	RAM
CH12_SWAPPED_FLOAT	32-bit register that holds the CH12 data in a floating point format. Scaling and decimal point values are based on those used for the CH12 data shown above. **Note**:This register is used to maintain backwards compatibility with older Intech products. When reading this register via Modbus the word order is Big Endian.	SF_32	39	RAM
CH12_12	12-bit register that holds the processed data for CH12. (Range from 0 - 4095) This register is used to maintain backwards compatibility with older Intech products.	U_12	12	RAM
IM_STATUS12	16-bit unsigned register that holds the input module status for CH12. (See Input Module Status)	U_16	4603	RAM

The above registers are normally updated by the operating system of the controller after a new input sample is processed. If the channel is disabled or in a counter mode, it is also possible to modify the contents of the register by writing to it from the setpoint reset logic, from the Macro, or via the serial port. A write to these registers in any other operational mode may result in the newly written value being overwritten by the operating system in the controller.

See also
CH12 Setup Registers

2.2.12.1 CH12 Setup Registers

Name	Description	Symbol Type	Register Number	Memory Type
OFFSET_CH12	32-bit register. Holds the calibration offset for CH12 and CH12_FLOAT.	S_32	635	RAM/EEPROM
SCALE_FACTOR_CH12	32-bit floating point register. Holds the calibration scale factor for CH12 and CH12_FLOAT.	F_32	1119	RAM/EEPROM
OFFSET_CH12_12BIT	16-bit register. Holds the calibration offset for CH12_12.	S_16	4587	RAM/EEPROM
SCALE_FACTOR_CH12_12BIT	32-bit floating point register. Holds the calibration scale factor for CH12_12.	F_32	1183	RAM/EEPROM

CHANNEL12_TEXT	Text display for CH12.	L_14	16415	EEROM
UNITS_TEXT_CH12	Units text for CH12. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17431	EEROM
DISPLAY_FORMAT_CH12	8-bit register. Controls the display format settings for CH12 (displayed in octal format).	O_8	8332	RAM/EEPROM
TEXT_CHARACTER_CH12	8-bit register. Holds the ASCII value for the last digit text character for CH12 (0 = no character).	U_8	8386	RAM/EEPROM

See also
Channel 12

2.2.13 Channel 13

Channel 13 registers can be selected as the data source for:

- The primary display.
- The second display, if installed.
- The third display, if installed.
- Trigger for advanced setpoints SP1 to SP16 (integer registers only)
- Analogue output channels (integer registers only).
- Setpoint reset destination (integer registers only).
  The reset destination mode allows you to select a register to be reset using the contents of another register triggered by a setpoint.

Name	Description	Symbol Type	Register Number	Memory Type
CH13	32-bit register that holds the processed data for CH13.	S_32	669	RAM/FLASH
CH13_RAW	32-bit register that holds the raw data for CH13. **Note**: When input module is operating in counter mode, this register shows the raw accumulated count value.	S_32	701	RAM/FLASH
CH13_FLOAT	32-bit register that holds the CH13 data in a floating point format. Scaling and decimal point values are based on those used for the CH13 data shown above.	F_32	1217	RAM
CH13_SWAPPED_FLOAT	32-bit register that holds the CH13 data in a floating point format. Scaling and decimal point values are based on those used for the CH13 data shown above. **Note**: This register is used to maintain backwards compatibility with older Intech products. When reading this register via Modbus the word order is Big Endian.	SF_32	41	RAM
CH13_12	12-bit register that holds the processed data for CH13. (Range from 0 - 4095) This register is used to maintain backwards compatibility with older Intech products.	U_12	13	RAM
IM_STATUS13	16-bit unsigned register that holds the input module status for CH13. (See Input Module Status)	U_16	4604	RAM

The above registers are normally updated by the operating system of the controller after a new input sample is processed. If the channel is disabled or in a counter mode, it is also possible to modify the contents of the register by writing to it from the setpoint reset logic, from the Macro, or via the serial port. A write to these registers in any other operational mode may result in the newly written value being overwritten by the operating system in the controller.

See also
CH13 Setup Registers

2.2.13.1 CH13 Setup Registers

Name	Description	Symbol Type	Register Number	Memory Type
OFFSET_CH13	32-bit register. Holds the calibration offset for CH13 and CH13_FLOAT.	S_32	637	RAM/EEPROM
SCALE_FACTOR_CH13	32-bit floating point register. Holds the calibration scale factor for CH13 and CH13_FLOAT.	F_32	1121	RAM/EEPROM
OFFSET_CH13_12BIT	16-bit register. Holds the calibration offset for CH13_12.	S_16	4588	RAM/EEPROM
SCALE_FACTOR_CH13_12BIT	32-bit floating point register. Holds the calibration scale factor for CH13_12.	F_32	1185	RAM/EEPROM
CHANNEL13_TEXT	Text display for CH13.	L_14	16417	EEROM
UNITS_TEXT_CH13	Units text for CH13. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17433	EEROM
DISPLAY_FORMAT_CH13	8-bit register. Controls the display format settings for CH13 (displayed in octal format).	O_8	8333	RAM/EEPROM
TEXT_CHARACTER_CH13	8-bit register. Holds the ASCII value for the last digit text character for CH13 (0 = no character).	U_8	8387	RAM/EEPROM

See also
Channel 13

2.2.14 Channel 14

Channel 14 registers can be selected as the data source for:

- The primary display.
- The second display, if installed.
- The third display, if installed.
- Trigger for advanced setpoints SP1 to SP16 (integer registers only)
- Analogue output channels (integer registers only).
- Setpoint reset destination (integer registers only).
  The reset destination mode allows you to select a register to be reset using the contents of another register triggered by a setpoint.

Name	Description	Symbol Type	Register Number	Memory Type
CH14	32-bit register that holds the processed data for CH14.	S_32	671	RAM/FLASH
CH14_RAW	32-bit register that holds the raw data for CH14. **Note**: When input module is operating in counter mode, this register shows the raw accumulated count value.	S_32	703	RAM/FLASH
CH14_FLOAT	32-bit register that holds the CH14 data in a floating point format. Scaling and decimal point values are based on those used for the CH14 data shown above.	F_32	1219	RAM

CH14_SWAPPED_FLOAT	32-bit register that holds the CH14 data in a floating point format. Scaling and decimal point values are based on those used for the CH14 data shown above. **Note**: This register is used to maintain backwards compatibility with older Intech products. When reading this register via Modbus the word order is Big Endian.	SF_32	43	RAM
CH14_12	12-bit register that holds the processed data for CH14. (Range from 0 - 4095) This register is used to maintain backwards compatibility with older Intech products.	U_12	14	RAM
IM_STATUS14	16-bit unsigned register that holds the input module status for CH14. (See Input Module Status)	U_16	4605	RAM

The above registers are normally updated by the operating system of the controller after a new input sample is processed. If the channel is disabled or in a counter mode, it is also possible to modify the contents of the register by writing to it from the setpoint reset logic, from the Macro, or via the serial port. A write to these registers in any other operational mode may result in the newly written value being overwritten by the operating system in the controller.

See also
CH14 Setup Registers

2.2.14.1 CH14 Setup Registers

Name	Description	Symbol Type	Register Number	Memory Type
OFFSET_CH14	32-bit register. Holds the calibration offset for CH14 and CH14_FLOAT.	S_32	639	RAM/EEPROM
SCALE_FACTOR_CH14	32-bit floating point register. Holds the calibration scale factor for CH14 and CH14_FLOAT.	F_32	1123	RAM/EEPROM
OFFSET_CH14_12BIT	16-bit register. Holds the calibration offset for CH14_12.	S_16	4589	RAM/EEPROM
SCALE_FACTOR_CH14_12BIT	32-bit floating point register. Holds the calibration scale factor for CH14_12.	F_32	1187	RAM/EEPROM
CHANNEL14_TEXT	Text display for CH14.	L_14	16419	EEROM
UNITS_TEXT_CH14	Units text for CH14. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17435	EEROM
DISPLAY_FORMAT_CH14	8-bit register. Controls the display format settings for CH14 (displayed in octal format).	O_8	8334	RAM/EEPROM
TEXT_CHARACTER_CH14	8-bit register. Holds the ASCII value for the last digit text character for CH14 (0 = no character).	U_8	8388	RAM/EEPROM

See also
Channel 14

2.2.15 Channel 15

Channel 15 registers can be selected as the data source for:

- The primary display.
- The second display, if installed.
- The third display, if installed.
- Trigger for advanced setpoints SP1 to SP16 (integer registers only)
- Analogue output channels (integer registers only).
- Setpoint reset destination (integer registers only).

The reset destination mode allows you to select a register to be reset using the contents of another register triggered by a setpoint.

Name	Description	Symbol Type	Register Number	Memory Type
CH15	32-bit register that holds the processed data for CH15.	S_32	673	RAM/FLASH
CH15_RAW	32-bit register that holds the raw data for CH15. **Note**: When input module is operating in counter mode, this register shows the raw accumulated count value.	S_32	705	RAM/FLASH
CH15_FLOAT	32-bit register that holds the CH15 data in a floating point format. Scaling and decimal point values are based on those used for the CH15 data shown above.	F_32	1221	RAM
CH15_SWAPPED_FLOAT	32-bit register that holds the CH15 data in a floating point format. Scaling and decimal point values are based on those used for the CH15 data shown above. **Note**: This register is used to maintain backwards compatibility with older Intech products. When reading this register via Modbus the word order is Big Endian.	SF_32	45	RAM
CH15_12	12-bit register that holds the processed data for CH15. (Range from 0 - 4095) This register is used to maintain backwards compatibility with older Intech products.	U_12	15	RAM
IM_STATUS15	16-bit unsigned register that holds the input module status for CH15. (See Input Module Status)	U_16	4606	RAM

The above registers are normally updated by the operating system of the controller after a new input sample is processed. If the channel is disabled or in a counter mode, it is also possible to modify the contents of the register by writing to it from the setpoint reset logic, from the Macro, or via the serial port. A write to these registers in any other operational mode may result in the newly written value being overwritten by the operating system in the controller.

See also
CH15 Setup Registers

2.2.15.1 CH15 Setup Registers

Name	Description	Symbol Type	Register Number	Memory Type
OFFSET_CH15	32-bit register. Holds the calibration offset for CH15 and CH15_FLOAT.	S_32	641	RAM/EEPROM
SCALE_FACTOR_CH15	32-bit floating point register. Holds the calibration scale factor for CH15 and CH15_FLOAT.	F_32	1125	RAM/EEPROM
OFFSET_CH15_12BIT	16-bit register. Holds the calibration offset for CH15_12.	S_16	4590	RAM/EEPROM
SCALE_FACTOR_CH15_12BIT	32-bit floating point register. Holds the calibration scale factor for CH15_12.	F_32	1189	RAM/EEPROM
CHANNEL15_TEXT	Text display for CH15.	L_14	16421	EEROM
UNITS_TEXT_CH15	Units text for CH15. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17437	EEROM
DISPLAY_FORMAT_CH15	8-bit register. Controls the display format settings for CH15 (displayed in octal format).	O_8	8335	RAM/EEPROM
TEXT_CHARACTER_CH15	8-bit register. Holds the ASCII value for the last digit text character for CH15 (0 = no character).	U_8	8389	RAM/EEPROM

See also
Channel 15

2.2.16 Channel 16

Channel 16 registers can be selected as the data source for:

- The primary display.
- The second display, if installed.
- The third display, if installed.
- Trigger for advanced setpoints SP1 to SP16 (integer registers only)
- Analogue output channels (integer registers only).
- Setpoint reset destination (integer registers only).
  The reset destination mode allows you to select a register to be reset using the contents of another register triggered by a setpoint.

Name	Description	Symbol Type	Register Number	Memory Type
CH16	32-bit register that holds the processed data for CH16.	S_32	675	RAM/FLASH
CH16_RAW	32-bit register that holds the raw data for CH16. **Note**: When input module is operating in counter mode, this register shows the raw accumulated count value.	S_32	707	RAM/FLASH
CH16_FLOAT	32-bit register that holds the CH16 data in a floating point format. Scaling and decimal point values are based on those used for the CH16 data shown above.	F_32	1223	RAM
CH16_SWAPPED_FLOAT	32-bit register that holds the CH16 data in a floating point format. Scaling and decimal point values are based on those used for the CH16 data shown above. **Note**: This register is used to maintain backwards compatibility with older Intech products. When reading this register via Modbus the word order is Big Endian.	SF_32	47	RAM
CH16_12	12-bit register that holds the processed data for CH16. (Range from 0 - 4095) This register is used to maintain backwards compatibility with older Intech products.	U_12	16	RAM
IM_STATUS16	16-bit unsigned register that holds the input module status for CH16. (See Input Module Status)	U_16	4607	RAM

The above registers are normally updated by the operating system of the controller after a new input sample is processed. If the channel is disabled or in a counter mode, it is also possible to modify the contents of the register by writing to it from the setpoint reset logic, from the Macro, or via the serial port. A write to these registers in any other operational mode may result in the newly written value being overwritten by the operating system in the controller.

See also
CH16 Setup Registers

2.2.16.1 CH16 Setup Registers

Name	Description	Symbol Type	Register Number	Memory Type
OFFSET_CH16	32-bit register. Holds the calibration offset for CH16 and CH16_FLOAT.	S_32	643	RAM/EEPROM
SCALE_FACTOR_CH16	32-bit floating point register. Holds the calibration scale factor for CH16 and CH16_FLOAT.	F_32	1127	RAM/EEPROM
OFFSET_CH16_12BIT	16-bit register. Holds the calibration offset for CH16_12.	S_16	4591	RAM/EEPROM
SCALE_FACTOR_CH16_12BIT	32-bit floating point register. Holds the calibration scale factor for CH16_12.	F_32	1191	RAM/EEPROM
CHANNEL16_TEXT	Text display for CH16.	L_14	16423	EEROM
UNITS_TEXT_CH16	Units text for CH16. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17439	EEROM
DISPLAY_FORMAT_CH16	8-bit register. Controls the display format settings for CH16 (displayed in octal format).	O_8	8336	RAM/EEPROM
TEXT_CHARACTER_CH16	8-bit register. Holds the ASCII value for the last digit text character for CH16 (0 = no character).	U_8	8390	RAM/EEPROM

See also
Channel 16

2.2.17 TC Cold Junction Temperature Selection

The Zen IoT controller can be configured to have all of its 16 analogue input channels to work with thermocouple temperature probes. In this mode, cold junction temperature compensation is carried out by measuring the ambient temperature inside the Zen IoT controller at the input terminals.

However in some applications it is desirable to measure the cold junction temperature at an external source. To allow for this the Zen IoT has two cold junction select registers which allow the user to define an input channel to be used as a cold junction temperature reference.

There are two registers associated with this function;

Name	Description	Symbol Type	Register Number	Memory Type
CJC_SELECT_LOW	8 bit register that selects the input channel used for cold junction compensation for input channels 1-8.	U_8	8501	RAM/EEPROM
CJC_SELECT_HIGH	8 bit register that selects the input channel used for cold junction compensation for input channels 9-16.	U_8	8502	RAM/EEPROM

Register 8501 - CJC Select Low
Register 8501 is an 8 bit unsigned register that specifies the input channel to be used with inputs 1-8.

Register 8502 - CJC Select Low
Register 8502 is an 8 bit unsigned register that specifies the input channel to be used with inputs 9-16.

NOTE: The input channel selected to measure the cold junction temperature must be set to operate in **RTD input mode** with a resolution of **0.1 degree F.** Even if the final temperature results for the other TC channels are set to read in degrees C with a different resolution, the cold junction channel must be set as above.

The function of these registers is shown in the table below.

CJC Select Value	Cold Junction Temperature Channel
0 *(default)*	Cold junction temperature is taken from internal sensor in Zen IoT.
1	Cold junction temperature is taken from input channel 1 result.
2	Cold junction temperature is taken from input channel 2 result.
3	Cold junction temperature is taken from input channel 3 result.
4	Cold junction temperature is taken from input channel 4 result.
5	Cold junction temperature is taken from input channel 5 result.
6	Cold junction temperature is taken from input channel 6 result.
7	Cold junction temperature is taken from input channel 7 result.
8	Cold junction temperature is taken from input channel 8 result.
9	Cold junction temperature is taken from input channel 9 result.
10	Cold junction temperature is taken from input channel 10 result.
11	Cold junction temperature is taken from input channel 11 result.
12	Cold junction temperature is taken from input channel 12 result.
13	Cold junction temperature is taken from input channel 13 result.
14	Cold junction temperature is taken from input channel 14 result.
15	Cold junction temperature is taken from input channel 15 result.
16	Cold junction temperature is taken from input channel 16 result.

Note: It is possible for both CJC_SELECT_LOW and CJC_SELECT_HIGH to select the same input channel. This allows 15 thermocouples to be used with only 1 cold junction RTD channel.

2.3 Clock

The following registers are used to hold time and date information from the real-time clock. These read/write registers are continuously updated by the operating system of the controller. If the real-time clock option is installed in the controller, then these registers are maintained even during power down. If the real-time clock option is not installed in the controller then these registers are still updated by the controller, but all values are lost when the power is removed from the controller.

Name	Description	Symbol Type	Register Number	Memory Type
DATE	8-bit register. Holds the real-time clock **date** (range 1 to 31).	U_8	8242	RAM/NVRAM
DAYS	8-bit register. Holds the real-time clock **days of the week** (Sunday = 0, Saturday = 6).	U_8	8241	RAM/NVRAM
HOURS	8-bit register. Holds the real-time clock **hours** count (range 0 to 23).	U_8	8240	RAM/NVRAM
MINUTES	8-bit register. Holds the real-time clock **minutes** count (range 0 to 59).	U_8	8239	RAM/NVRAM
HOURS_MINUTES	16-bit read only register. Holds the real-time clock count in minutes for **hours : minutes** (range 0 to 1439 (00:00 to 23:59)).	U_16_R	4438	RAM
HRS_MIN_SEC	32-bit read only register. Holds the real-time clock count in seconds for **hours : minutes : seconds** (range 0 to 86399 (0:00:00 to 23:59:59)).	U_32_R	151	RAM
MONTH	8-bit register. Holds the real-time clock **month** (range 1 to 12).	U_8	8243	RAM/NVRAM
SECONDS	8-bit register. Holds the real-time clock **seconds** count (range 0 to 59).	U_8	8238	RAM/NVRAM
YEAR	8-bit register. Holds the real-time clock **year** (range 0 to 99).	U_8	8244	RAM/NVRAM

2.3.1 Daylight Saving

From firmware version **V0.08.01 onwards**, Zen IoT controllers support daylight saving correction and Time Zone.

The daylight saving function works by detecting the start and end of daylight saving time as per the configuration specified by the user. If it detects that the current time stamp lies outside of the selected daylight saving period, it reports the current time at the selected time zone. If it finds that the current time stamp lies within the daylight saving period, it then adds the users predefined time offset to the time and also updates a current time zone register to show the adjusted time.

Note 1: In order for daylight saving to work correctly it is important that all clock parameters are correctly synchronized for your local time and your current time zone. This also includes the day of the week. If any clock parameters are not correct, daylight saving adjustments will be incorrect.

Note 2: DS_START_MONTH and DS_END_MONTH must be different. Setting DS_START_MONTH and DS_END_MONTH to the same value will disable the daylight saving function.

The following table shows the registers that are associated with the daylight saving function.

Name	Description	Symbol Type	Register Number	Memory Type
DS_START_MONTH	8-bit register. Holds the **month** when daylight saving starts (range 1 to 12).	U_8	8531	RAM/EEPROM
DS_START_DAY	8-bit register. Holds the **day of the week** that daylight saving starts on (Sunday = 0, Saturday = 6).	U_8	8533	RAM/EEPROM
DS_START_RECURRENCE	8-bit register. Holds the **number of times** that DS_START_DAY must occur before daylight saving time starts (range 1 to 5). Note: selecting 5 is the same as choosing the last occurrence in a month which could be 4 or 5 depending on the month.	U_8	8535	RAM/EEPROM
DS_START_TIME	16-bit register. Holds the daylight saving start time in **minutes** past midnight (range 0 to 1439 (00:00 to 23:59)).	U_16	4659	RAM/EEPROM
DS_END_MONTH	8-bit register. Holds the **month** when daylight saving ends (range 1 to 12).	U_8	8532	RAM/EEPROM
DS_END_DAY	8-bit register. Holds the **day of the week** that daylight saving ends on (Sunday = 0, Saturday = 6).	U_8	8534	RAM/EEPROM
DS_END_RECURRENCE	8-bit register. Holds the **number of times** that DS_END_DAY must occur before daylight saving time ends (range 1 to 5). Note: selecting 5 is the same as choosing the last occurrence in a month which could be 4 or 5 depending on the month.	U_8	8536	RAM/EEPROM
DS_END_TIME	16-bit register. Holds the daylight saving end time in **minutes** past midnight (range 0 to 1439 (00:00 to 23:59)).	U_16	4660	RAM/EEPROM
DS_OFFSET	8-bit signed register. Holds the **daylight saving time offset** that is added to the current time when daylight saving is active (range -128mins to +127mins). Note: typically this value is +60 minutes (+1:00) but it can be a negative value as well.	S_8	8537	RAM/EEPROM

See Also
Clock

Time Zone

2.3.2 Time Zone

From firmware version **V0.08.01 onwards**, Zen IoT controllers support international time zone reporting and daylight saving correction.

An international time zone reference is often needed when communicating with other Internet connected devices. The Zen IoT controller allows the user to specify their time zone in coordinated universal time (UTC) and then provides a register to report the current time zone in UTC which is compensated for daylight saving adjustments.

Note: In order for daylight saving to work correctly it is important that all clock parameters are correctly synchronized for your local time and your current time zone. This also includes the day of the week. If any clock parameters are not correct, daylight saving adjustments will be incorrect.

The following table shows the registers that are associated with the time zone.

Name	Description	Symbol Type	Register Number	Memory Type
TIME_ZONE	16-bit signed register. Holds the **UTC time zone** value in minutes, specified by the user for their particular location. The range of -1439 to +1439 minutes (or -23:59 to +23:59).	S_16	4661	RAM/EEPROM
CURRENT_TIME_ZONE	16-bit signed read only register. This register shows the **current time zone** value in UTC based on the user defined TIME_ZONE and the daylight saving time offset. This value is reported in minutes and has a range of -1439 to +1439 minutes (or -23:59 to +23:59).	S_16_R	4662	RAM/EEPROM

See Also

Clock

Daylight Saving

2.4 Configuration

Configuration Registers

Registers 8193 to 8200 are 8-bit registers used to control the functionality of the controller. When reading or writing to these registers via the serial port in ASCII mode, the data is treated in octal format. The function selected in the 1st digit of each register is stored in bits 6 and 7. The function selected in the 2nd digit of each register is stored in bits 3, 4, and 5. The function selected in the 3rd digit of each register is stored in bits 0, 1, and 2.

For example:

If the manual setup for COUNTER_A_SETUP shows 241 on the display, then reading register 8197 in ASCII mode results in a value of 241. Converting this octal value to a binary equivalent of 10100001 or hexadecimal equivalent of 0A1.

	1st Digit	2nd Digit	3rd Digit
Octal	2	4	1
Binary	10	100	001

Name	Description	Symbol Type	Register Number	Memory Type
CAL	8-bit register. Holds the currently programmed calibration mode settings (Note, the meter display is in octal).	O_8	8193	RAM/EEPROM

ANALOG_MODE	8-bit register. Holds the currently programmed settings for analog mode setup (Note, the display is in octal).	O_8	8195	RAM/EEPROM
COUNTER_A_MODE	8-bit register. Holds the currently programmed settings for Counter A setup (Note, the display is in octal).	O_8	8196	RAM/EEPROM
COUNTER_B_SETUP	8-bit register. Holds the currently programmed settings for Counter B setup (Note, the display is in octal).	O_8	8197	RAM/EEPROM
COUNTER_C_SETUP	8-bit register. Holds the currently programmed settings for Counter C setup (Note, the display is in octal).	O_8	8198	RAM/EEPROM
COUNTER_D_SETUP	8-bit register. Holds the currently programmed settings for Counter D setup (Note, the display is in octal).	O_8	8199	RAM/EEPROM
LOGGING_SETUP	8-bit register. Holds the currently programmed settings for **Code 8** (Note, the display is in octal).	O_8	8200	RAM/EEPROM

2.4.1 Analogue Mode Setup

While programming through the front display, the programming digits of the analog mode setup register provide the settings to select supply rejection, and analogue output modes.

The analog mode setup register is represented in octal format to allow 3 functions to be selected in one digit.

Display Digit	**1st Digit**	**2nd Digit**	**3rd Digit**
Function	Supply Rejection	Analogue O/P Mode	Analogue O/P Options

1st Digit - Supply Rejection
The first digit of the analog mode setup register (bits 6 & 7) are used to select the supply frequency rejection, as shown below:

0XX = 60hz supply rejection.
1XX = 50hz supply rejection.
2XX = Reserved for future development
3XX = Reserved for future development

2nd Digit - Analogue Output Mode
The 2nd digit of the analog mode setup register (bits 3, 4, 5) selects different analogue output modes as per the following options:

X0X = Intech 2100M driver mode.
X1X = Normal mode (SCADA).
X2X = PLC RTX (clk/rst).
X3X = PLC RTX (BCD).
X4X = Reserved for future development.
X5X = Reserved for future development.
X6X = Reserved for future development.
X7X = Reserved for future development.

3rd Digit - Analogue Output Options
The 3rd digit of the analog mode setup register (bits 0,1, 2) has different options depending on the selection of the second digit. The various options are shown below for each relevant setting of the second digit.

2nd digit = 0
If the 2nd digit of the analog mode setup register is set to 0 (2100M driver mode) then the 3rd digit functions as shown below;

X00 = 700mS delay between clock pulses.

X01 = 1 second delay between clock pulses.
X02 = 2 seconds delay between clock pulses.
X03 = 3 seconds delay between clock pulses.
X04 = 4 seconds delay between clock pulses.
X05 = 5 seconds delay between clock pulses.
X06 = 6 seconds delay between clock pulses.
X07 = 7 seconds delay between clock pulses.

2nd digit = 2
If the 2nd digit of the analog mode setup register is set to 2 (PLC RTX (clk/rst) mode) then the 3rd digit functions as a debounce timer for the clock input pin (D2) with the following options;

X20 = No debounce time.
X21 = 2.5mS debounce time.
X22 = 5mS debounce time.
X23 = 10mS debounce time.
X24 = 25mS debounce time.
X25 = 50mS debounce time.
X26 = 100mS debounce time.
X27 = 200mS debounce time.

2nd digit = 3
If the 2nd digit of the analog mode setup register is set to 3 (PLC RTX (BCD) mode) then the 3rd digit gives the following options;

X30 = 12 bit result values output on analogue O/P 1, 12 bit scaled setpoints output on analogue O/P 2. (Intech compatibility mode).
X31 = 32 bit result values output on analogue O/P 1, 32 bit totals 1-10 output on analogue O/P 2.
X32 = Reserved for future development.
X33 = Reserved for future development.
X34 = Reserved for future development.
X35 = Reserved for future development.
X36 = Reserved for future development.
X37 = Reserved for future development.

Note: If the analogue mode is set to X31, BCD input values of 0-9 will cause TOTAL1 - TOTAL10 values to be output on analogue output 2. For BCD input values 10 - 15, analogue output channel 2 will operate in normal mode and output whatever register the DATA_SOURCE_ANALOG2 points to.

2nd digit = 1 or 4-7
If the 2nd digit of the analog mode setup register is set to 1, or 3 to 7 then the 3rd digit has no function. It is recommended that the 3rd digit be set to 0.

NOTE: Some analogue output mode settings shown above also require the use of various digital input pins. Setting the analogue output to these modes will over ride other settings for digital inputs (see Counter Setup)

See also
Counter A Setup

Counter B Setup

Counter C Setup

Counter D Setup

2.4.2 Counter A Mode Setup

While programming through the front display, the programming digits of the counter A mode register allow you to select from various digital input modes associated with the DI A input pin.

The counter 1 mode register is represented in octal format to allow 3 functions to be selected in one digit.

Display Digit	1st Digit	2nd Digit	3rd Digit
Function	Reset/Restore	Digital I/P Mode	Digital I/P Options

1st Digit - Reset/Restore Count A at Power-up
The first digit of the counter A mode register (bits 6 & 7) are used to select the whether the count value for the counter A register is reset to zero at a power up or restored to the last count value before power down. The options are as shown below:

0XX = Restore count A value at power up.
1XX = Reset count A value to zero at power up.
2XX = Apply 32 point linearization to count A and restore count A value at power up.
3XX = Apply 32 point linearization to count A and reset count A value to zero at power up.

Note: Linearization table 1 is used for counter A linearization options 2XX and 3XX. (See Linearization for information).

NOTE: Linearization is applied **AFTER** scaling and offset.

2nd Digit - DI A Digital Input Mode
The 2nd digit of the counter A mode register (bits 3, 4, 5) selects different digital input modes for the DI A pin as per the following options:

X0X = Digital input only.
X1X = Counter input.
X2X = Frequency counter input.
X3X = Reserved for future development.
X4X = Reserved for future development.
X5X = Reserved for future development.
X6X = Reserved for future development.
X7X = Reserved for future development.

3rd Digit - Digital Input Options
The 3rd digit of the counter A mode register (bits 0,1, 2) has different options depending on the selection of the second digit. The various options are shown below for each relevant setting of the second digit.

2nd digit = 0 (digital input)
If the 2nd digit of the counter A mode register is set to 0 (digital input only) then the 3rd digit functions as shown below;

X00 = Digital input only - no other associated functions.
X01 = Digital input which also triggers capture macro on leading edge of pulse.
X02 = Digital input with data log on leading edge of pulse.
X03 = Digital input with gated interval logging and data log on leading edge of pulse. (See Gated Interval Logging).
X04 = Reserved for future development.
X05 = Same as X01 above with 5mS de-bounce applied to leading edge of pulse.
X06 = Same as X02 above with 5mS de-bounce applied to leading edge of pulse.
X07 = Same as X03 above with 5mS de-bounce applied to leading edge of pulse.

(**Note:** options X03 to X07 above are only available on firmware V0.09.04+)

2nd digit = 1 (Counter input)
If the 2nd digit of the counter A mode register is set to 1 (counter input) then the 3rd digit functions as shown below;

X10 = Up counter.
X11 = Up/Down counter (DI B = direction where up=DI B off, down=DI B on).
X12 = Gated up counter (DI B = gate control where count enabled if DI B=on, disabled if DI B=off).
X13 = Reserved for future development.
X14 = De-bounced up counter.
X15 = De-bounced up/down counter (DI B = direction where up=DI B off, down = DI B on).
X16 = De-bounced gated up counter (DI B = gate control, count enabled if DI B=on, disabled if DI B=off).
X17 = Reserved for future development.

Note: In de-bounced count modes a 5mS de-bounce period is applied after the leading edge of a count pulse. The de-bounce logic is only applied to the count input (i.e. no de-bounce applied to DIB in up/down or gated count modes).

2nd digit = 2 (frequency counter input)
If the 2nd digit of the counter A mode register is set to 0 (digital input only) then the 3rd digit functions as shown below;

3rd digit
X20 = Frequency counter (0.10hz - 2500.00hz).
X21 = Reserved for future development.
X22 = Reserved for future development.
X23 = Reserved for future development.
X24 = Reserved for future development.
X25 = Reserved for future development.
X26 = Reserved for future development.
X27 = Reserved for future development.

2nd digit = 3 to 7
If the 2nd digit of the counter A mode register is set from 3 to 7 then the 3rd digit has no function. It is recommended that the 3rd digit be set to 0.

NOTE: Some settings of the analog mode setup register require the use of digital input pins for analogue output modes. These mode will over ride the settings of the digital inputs shown above (see analog mode setup).

Also note that some of the counter options also use the DI B input pins. If these options are selected here they will over ride the setup options for the DI B.

See also
Counter B Mode Setup

Counter C Mode Setup

Counter D Mode Setup

Analog Mode Setup

2.4.3 Counter B Mode Setup

While programming through the front display, the programming digits of the counter B mode register allow you to select from various digital input modes associated with the DI B input pin.

The counter B mode register is represented in octal format to allow 3 functions to be selected in one digit.

Display Digit	1st Digit	2nd Digit	3rd Digit
Function	Reset/Restore	Digital I/P Mode	Digital I/P Options

1st Digit - Reset/Restore Count B at Power-up
The first digit of the counter B mode register (bits 6 & 7) are used to select the whether the count value for the counter 1 register is reset to zero at a power up or restored to the last count value before power down. The options are as shown below:

0XX = Restore count B value at power up.
1XX = Reset count B value to zero at power up.
2XX = Apply 32 point linearization to count B and restore count B value at power up.
3XX = Apply 32 point linearization to count B and reset count B value to zero at power up.

Note: Linearization table 2 is used for counter B linearization options 2XX and 3XX. (See Linearization for information).

NOTE: Linearization is applied **AFTER** scaling and offset.

2nd Digit - DI B Digital Input Mode
The 2nd digit of the counter B mode register (bits 3, 4, 5) selects different digital input modes for the DI B pin as per the following options:

X0X = Digital input only.
X1X = Counter input.
X2X = Frequency counter input.
X3X = Reserved for future development.
X4X = Reserved for future development.
X5X = Reserved for future development.
X6X = Reserved for future development.
X7X = Reserved for future development.

3rd Digit - Digital Input Options
The 3rd digit of the counter B mode register (bits 0,1, 2) has different options depending on the selection of the second digit. The various options are shown below for each relevant setting of the second digit.

2nd digit = 0 (digital input)
If the 2nd digit of the counter B mode register is set to 0 (digital input only) then the 3rd digit functions as shown below;

X00 = Digital input only - no other associated functions.
X01 = Digital input which also triggers capture macro on leading edge of pulse.
X02 = Digital input with data log on leading edge of pulse.
X03 = Digital input with gated interval logging and data log on leading edge of pulse. (See Gated Interval Logging).
X04 = Reserved for future development.
X05 = Same as X01 above with 5mS de-bounce applied to leading edge of pulse.
X06 = Same as X02 above with 5mS de-bounce applied to leading edge of pulse.
X07 = Same as X03 above with 5mS de-bounce applied to leading edge of pulse.

(**Note:** options X03 to X07 above are only available on firmware V0.09.04+)

2nd digit = 1 (Counter input)
If the 2nd digit of the counter B mode register is set to 1 (counter input) then the 3rd digit functions as shown below;

X10 = Up counter.
X11 = Reserved for future development.
X12 = Reserved for future development.
X13 = Reserved for future development.

X14 = De-bounced up counter (5mS de-bounce applied to leading edge of count pulse).
X15 = Reserved for future development.
X16 = Reserved for future development.
X17 = Reserved for future development.

2nd digit = 2 (frequency counter input)
If the 2nd digit of the counter B mode register is set to 0 (digital input only) then the 3rd digit functions as shown below;

3rd digit
X20 = Frequency counter (0.10hz - 2500.00hz).
X21 = Reserved for future development.
X22 = Reserved for future development.
X23 = Reserved for future development.
X24 = Reserved for future development.
X25 = Reserved for future development.
X26 = Reserved for future development.
X27 = Reserved for future development.

2nd digit = 3 to 7
If the 2nd digit of the counter B mode register is set from 3 to 7 then the 3rd digit has no function. It is recommended that the 3rd digit be set to 0.

NOTE: Some settings of the analog mode setup register require the use of digital input pins for analogue output modes. These mode will over ride the settings of the digital inputs shown above (see analog mode setup).

Note also that some of the counter functions for the DI A pin also require the use of the DI B pin and in these modes the above settings for DI B will be overridden. (see Counter A Mode Setup).

See also
Counter A Mode Setup

Counter C Mode Setup

Counter D Mode Setup

Analog Mode Setup

2.4.4 Counter C Mode Setup

While programming through the front display, the programming digits of the counter C mode register allow you to select from various digital input modes associated with the DI C input pin.

The counter C mode register is represented in octal format to allow 3 functions to be selected in one digit.

Display Digit	**1st Digit**	**2nd Digit**	**3rd Digit**
Function	Reset/Restore	Digital I/P Mode	Digital I/P Options

1st Digit - Reset/Restore Count C at Power-up
The first digit of the counter C mode register (bits 6 & 7) are used to select the whether the count value for the counter C register is reset to zero at a power up or restored to the last count value before power down. The options are as shown below:

0XX = Restore count C value at power up.
1XX = Reset count C value to zero at power up.
2XX = Apply 32 point linearization to count C and restore count C value at power up.
3XX = Apply 32 point linearization to count C and reset count C value to zero at power up.

Note: Linearization table 3 is used for counter C linearization options 2XX and 3XX. (See Linearization for information).

NOTE: Linearization is applied **AFTER** scaling and offset.

2nd Digit - DI C Digital Input Mode
The 2nd digit of the counter C mode register (bits 3, 4, 5) selects different digital input modes for the DI C pin as per the following options:

X0X = Digital input only.
X1X = Counter input.
X2X = Frequency counter input.
X3X = Reserved for future development.
X4X = Reserved for future development.
X5X = Reserved for future development.
X6X = Reserved for future development.
X7X = Reserved for future development.

3rd Digit - Digital Input Options
The 3rd digit of the counter C mode register (bits 0,1, 2) has different options depending on the selection of the second digit. The various options are shown below for each relevant setting of the second digit.

2nd digit = 0 (digital input)
If the 2nd digit of the counter C mode register is set to 0 (digital input only) then the 3rd digit functions as shown below;

X00 = Digital input only - no other associated functions.
X01 = Digital input which also triggers capture macro on leading edge of pulse.
X02 = Digital input with data log on leading edge of pulse.
X03 = Digital input with gated interval logging and data log on leading edge of pulse. (See Gated Interval Logging).
X04 = Reserved for future development.
X05 = Same as X01 above with 5mS de-bounce applied to leading edge of pulse.
X06 = Same as X02 above with 5mS de-bounce applied to leading edge of pulse.
X07 = Same as X03 above with 5mS de-bounce applied to leading edge of pulse.

(**Note:** options X03 to X07 above are only available on firmware V0.09.04+)

2nd digit = 1 (Counter input)
If the 2nd digit of the counter C mode register is set to 1 (counter input) then the 3rd digit functions as shown below;

X10 = Up counter.
X11 = Up/Down counter (DI D = direction where up=DI D off, down=DI Don).
X12 = Gated up counter (DI D = gate control where count enabled if DI D=on, disabled DDI D=off).
X13 = Reserved for future development.
X14 = De-bounced up counter.
X15 = De-bounced up/down counter (DI D = direction where up=DI D off, down=DI Don).
X16 = De-bounced gated up counter (DI D = gate control, count enabled if DI D=on, disabled DI D=off).
X17 = Reserved for future development.

Note: In de-bounced count modes a 5mS de-bounce period is applied to the leading edge of a count pulse. The de-bounce logic is only applied to the count input (i.e. no de-bounce applied to DI D in up/down or gated count modes).

2nd digit = 2 (frequency counter input)
If the 2nd digit of the counter C mode register is set to 0 (digital input only) then the 3rd digit functions as shown below;

3rd digit

X20 = Frequency counter (0.10hz - 2500.00hz).
X21 = Reserved for future development.
X22 = Reserved for future development.
X23 = Reserved for future development.
X24 = Reserved for future development.
X25 = Reserved for future development.
X26 = Reserved for future development.
X27 = Reserved for future development.

2nd digit = 3 to 7
If the 2nd digit of the counter C mode register is set from 3 to 7 then the 3rd digit has no function. It is recommended that the 3rd digit be set to 0.

NOTE: Some settings of the analog mode setup register require the use of digital input pins for analogue output modes. These mode will over ride the settings of the digital inputs shown above (see analog mode setup).

Also note that some of the counter options also use the DI D input pins. If these options are selected here they will over ride the setup options for the DI D.

See also
Counter A Mode Setup

Counter B Mode Setup

Counter D Mode Setup

Analog Mode Setup

2.4.5 Counter D Mode Setup

While programming through the front display, the programming digits of the counter D mode register allow you to select from various digital input modes associated with the DI D input pin.

The counter D mode register is represented in octal format to allow 3 functions to be selected in one digit.

Display Digit	**1st Digit**	**2nd Digit**	**3rd Digit**
Function	Reset/Restore	Digital I/P Mode	Digital I/P Options

1st Digit - Reset/Restore Count D at Power-up
The first digit of the counter D mode register (bits 6 & 7) are used to select the whether the count value for the counter D register is reset to zero at a power up or restored to the last count value before power down. The options are as shown below:

0XX = Restore count D value at power up.
1XX = Reset count D value to zero at power up.
2XX = Apply 32 point linearization to count D and restore count D value at power up.
3XX = Apply 32 point linearization to count D and reset count D value to zero at power up.

Note: Linearization table 4 is used for counter 4 linearization options 2XX and 3XX. (See Linearization for information).

NOTE: Linearization is applied **AFTER** scaling and offset.

2nd Digit - DI D Digital Input Mode
The 2nd digit of the counter D mode register (bits 3, 4, 5) selects different digital input modes for the DI D pin as per the following options:

X0X = Digital input only.
X1X = Counter input.
X2X = Frequency counter input.
X3X = Reserved for future development.
X4X = Reserved for future development.
X5X = Reserved for future development.
X6X = Reserved for future development.
X7X = Reserved for future development.

3rd Digit - Digital Input Options
The 3rd digit of the counter D mode register (bits 0,1, 2) has different options depending on the selection of the second digit. The various options are shown below for each relevant setting of the second digit.

2nd digit = 0 (digital input)
If the 2nd digit of the counter D mode register is set to 0 (digital input only) then the 3rd digit functions as shown below;

X00 = Digital input only - no other associated functions.
X01 = Digital input which also triggers capture macro on leading edge of pulse.
X02 = Digital input with data log on leading edge of pulse.
X03 = Digital input with gated interval logging and data log on leading edge of pulse. (See Gated Interval Logging).
X04 = Reserved for future development.
X05 = Same as X01 above with 5mS de-bounce applied to leading edge of pulse.
X06 = Same as X02 above with 5mS de-bounce applied to leading edge of pulse.
X07 = Same as X03 above with 5mS de-bounce applied to leading edge of pulse.

(**Note:** options X03 to X07 above are only available on firmware V0.09.04+)

2nd digit = 1 (Counter input)
If the 2nd digit of the counter D mode register is set to 1 (counter input) then the 3rd digit functions as shown below;

X10 = Up counter.
X11 = Reserved for future development.
X12 = Reserved for future development.
X13 = Reserved for future development.
X14 = De-bounced up counter (5ms de-bounce applied to leading edge of count pulse).
X15 = Reserved for future development.
X16 = Reserved for future development.
X17 = Reserved for future development.

2nd digit = 2 (frequency counter input)
If the 2nd digit of the counter D mode register is set to 0 (digital input only) then the 3rd digit functions as shown below;

3rd digit
X20 = Frequency counter (0.10hz - 2500.00hz).
X21 = Reserved for future development.
X22 = Reserved for future development.
X23 = Reserved for future development.
X24 = Reserved for future development.
X25 = Reserved for future development.
X26 = Reserved for future development.
X27 = Reserved for future development.

2nd digit = 3 to 7
If the 2nd digit of the counter D mode register is set from 3 to 7 then the 3rd digit has no function. It is recommended that the 3rd digit be set to 0.

NOTE: Some settings of the analog mode setup register require the use of digital input pins for analogue output modes. These mode will over ride the settings of the digital inputs shown above (see analog mode setup).

Note also that some of the counter functions for the DI C pin also require the use of the DI D pin and in these modes the above settings for DI D will be overridden. (see Counter C Mode Setup).

See also
Counter A Mode Setup

Counter B Mode Setup

Counter C Mode Setup

Analog Mode Setup

2.4.6 Logging Mode Setup

While programming through the front display, the programming digits of logging mode setup allow you to select data logging and print mode options.

The logging mode setup register is represented in octal format to allow 3 functions to be selected in one digit.

Display Digit	**1st Digit**	**2nd Digit**	**3rd Digit**
Function	Logging/buffer Control	Date/Time/Print Options	Manual Trigger Options

1st Digit - Logging Buffer Control
The first digit of the logging mode setup register (bits 6 & 7) are used to enable data logging and select the type of data logging buffer, as shown below:

0XX = Data logging disabled.
1XX = Data logging enabled - cyclic buffer (wraps around to 1 when it reaches the end of data logging memory).
2XX = Data logging enabled - linear buffer (logging stops when it reaches the end of data logging memory).
3XX = Reserved for future development.

2nd Digit - Date/Time/Print Options
The 2nd digit of the logging mode setup register (bits 3, 4, 5) selects different time stamp and print output options as shown below:

X0X = Printer output - no time stamp.
X1X = Printer output - with time stamp (Month/Day/Year Hrs:Min:Sec).
X2X = Printer output - with time stamp (Day/Month/Year Hrs:Min:Sec).
X3X = Printer output - with time stamp (Hrs:Min:Sec).
X4X = Spreadsheet output - no time stamp.
X5X = Spreadsheet output - with time stamp (Month/Day/Year Hrs:Min:Sec).
X6X = Spreadsheet output - with time stamp (Day/Month/Year Hrs:Min:Sec).
X7X = Spreadsheet output - with time stamp (Hrs:Min:Sec).

3rd Digit - Manual Trigger Options
The 3rd digit of the logging mode setup register (bits 0,1, 2) selects different options to manually trigger a log sample from push button switches. The various options are shown below;

XX0 = No manual trigger.
XX1 = Trigger log sample from Prog button.
XX2 = Trigger log sample from F1 button.

XX3 = Trigger log sample from F2 button.
XX4 = Reserved for future development.
XX5 = Reserved for future development.
XX6 = Reserved for future development.
XX7 = Reserved for future development.

See also
Data Logging

Data Logging Concepts

2.5 Counters

The Zen IoT controller includes 4 digital input pins. These can be configured for a variety of different input functions including standard digital status inputs, various counter modes and frequency counter modes.

The 4 digital input pins **are not isolated from the Zen IoT power supply** so care must be taken when connecting these inputs to sensors etc. All 4 inputs share the same common pin. The maximum frequency of these input pins is limited to approximately 10kHz (or pulse widths > 50uS) and they accept an input voltage range of 2-30V DC with a typical threshold value of 1.2V.

Each of the 4 digital channels has a number of associated registers which hold result and setup data. The result registers are normally updated by the operating system of the controller after each new input sample is processed. The result registers can be read or written to, however the outcome of a write operation to a result register will vary depending on the operational mode of the counter. The following outcomes are possible.

Digital Input Mode
If the counter channel is placed in the digital input only mode (see Counter Mode Setup and Analog Mode Setup) the COUNTER_x register will show a value of "1" or "0" to reflect the current input status of the digital input pin (Note: this only applies to firmware versions 0.09.03 onwards). The result registers COUNTER_x_SCALED, COUNTER_x_RAW and COUNTER_x_16 are totally separate from each other and they are not updated by the operating system. They can be used as storage registers by the serial port or the macro.

Gated Interval Logging
Firmware V0.09.04 onwards includes a gated interval logging option. When this option is selected for one of the digital status input pins, normal data logging will be disabled when the digital input is in-active. Data will only be logged at the rate specified by the LOG_INTERVAL_TIME when the digital input is active (i.e. "ON"). If 2 or more of the digital status inputs are setup in this mode then they form an "AND" function and normal data logging will be activated when all selected inputs are active. The leading edge of a pulse will also be logged with the trigger type for the digital input.

Counter Input Mode
If the counter channel is placed in the counter mode (i.e. up counter or up/down counter) then input counts are applied to the COUNTERx_RAW register, which is then scaled and applied to the COUNTER_x_SCALED register. If 32 point linearization is enabled then the COUNTER_x_SCALED value is taken as an input value and the linearized output value is applied to COUNTER_x. If 32 point linearization is disabled, the COUNTER_x_SCALED value will be copied into the COUNTER_x register directly. So these registers are updated by the operating system after each new input sample. However, a write to these registers is still possible in counter mode to enable the setting or resetting of the count value.

The COUNTER_x_16 register is effectively a copy of the COUNTER_x_RAW register, and is provided to maintain compatibility which older products.
Note: Because COUNTER_x_16 is only a 16 bit register, it will only show the lowest 16 bits of the COUNTER_x_RAW register. For example, if COUNTER_x_RAW equals 65536 counts then COUNTER_x_16 will show a value of 0.

A write to the COUNTER_x_RAW register will effect the COUNTER_x_SCALED, COUNTER_x and COUNTER_x_16 registers. The COUNTER_x register will be updated in accordance with the scale and offset values applied to the counter channel, and any linearization settings. The COUNTER_x_16 register will basically be a copy of the lowest 16 bits of COUNTER_x_RAW.

A write to the COUNTER_x_SCALED register will update the COUNTER_x_RAW register in accordance with the scale and offset values applied to the counter channel. This inturn will cause the COUNTER_x_16 and COUNTER_x registers to be updated with a new value as well. Note:if the value written to COUNTER_x_SCALED causes the COUNTER_x_RAW value to be greater than 65535, then COUNTER_x_16 will only show the lowest 16 bits of COUNTER_x_RAW.

A write to the COUNTERx_16 register will also update the COUNTERx_RAW register with the same value. COUNTERx_SCALED and COUNTERx registers will also be updated on the next sample accordance with the scale and offset values and any linearization applied to the counter channel. .

Frequency Counter Input Mode
If the counter channel is placed in the frequency counter mode then the frequency in Hz is applied to the COUNTER_x_RAW register, which is then scaled and applied to the COUNTER_x_SCALED register. If 32 point linearization is enabled then the COUNTER_x_SCALED value is taken as an input value and the linearized output value is applied to COUNTER_x. If 32 point linearization is disabled, the COUNTER_x_SCALED value will be copied into the COUNTER_x register directly. So these registers are updated by the operating system after each new input sample. Any writes to these register will be lost because the operating system is continuously over writing them with new input samples.

Note: In frequency counter mode COUNTER_x_16 registers are hold a 16 bit copy of COUNTER_x_RAW register. If COUNTER_x_RAW > 65535 then COUNTER_x_16 will display 65535. This is different to straight counter mode - see Counter Input Mode.

See also
Counter A

Counter B

Counter C

Counter D

2.5.1 Counter A

Counter A can be operated in various count and frequency counter modes.

Name	Description	Symbol Type	Register Number	Memory Type
COUNTER_A	32-bit register that holds the processed data for counter A. If 32 point linearization is applied to counter A then this register will hold the linearized output value. If linearization is disabled, this register will be a copy of the COUNTER_A_SCALED register. (Data may be count or frequency data). Scaling and decimal point values are based on those specified in the Counter A Setup Registers)	S_32	525	RAM
COUNTER_A_SCALED	32-bit register that holds the scaled data for counter A. (Data may be count or frequency data). Scaling and decimal point values are based on those specified in the Counter A Setup Registers)	S_32	709	RAM
COUNTER_A_RAW	32-bit register that holds the raw counter value for counter A before scaling is applied. This value is saved in NV memory at power down and can be restored at power up. (see Counter A Mode Setup for more info)	S_32	369	RAM/FLASH
COUNTER_A_16	16-bit register that holds the processed data for COUNTER_A_16. (Range from 0 - 65535) This register is used to maintain backwards compatibility with older Intech products.	U_16	113	RAM
COUNTER_A_FLOAT	32-bit register that holds a pseudo floating point image of processed data for counter A. Scaling and decimal point values are based on those specified in the Counter A Setup Registers). (See 32-bit Pseudo Floating Point).	PF_32	1805	RAM
DI A	1-bit read only flag that indicates the status of the DI A digital input pin. (See Internal Digital Inputs)	B_0_R	4108	RAM

NOTE: Most of the above registers are normally updated by the operating system of the controller after a new input sample is processed. If the channel is disabled or in a counter mode, it is also possible to modify the contents of the register by writing to it from the setpoint reset logic, from the Macro, or via the serial port. A write to these registers in other operational modes may result in the newly written value being overwritten by the operating system in the controller. (See Digital Input Mode, Counter Input Mode and Frequency Counter Input Mode)

See also

Counters

Counter A Setup Registers

Counter A Mode Setup

2.5.1.1 Counter A Setup Registers

Name	Description	Symbol Type	Register Number	Memory Type
OFFSET_COUNTER_A	32-bit register. Holds the calibration offset for COUNTER_A.	S_32	359	RAM/EEPROM
SCALE_FACTOR_COUNTER_A	32-bit floating point register. Holds the calibration scale factor for COUNTER_A.	F_32	1129	RAM/EEPROM

AVERAGING_SAMPLES_COUNTER_A	8-bit register sets the averaging samples for COUNTER_A and COUNTER_A_16. Note: averaging is only applied in frequency counter mode, not in counter mode. (Range 0 to 255, 0=off)..	U_8	8311	RAM/EEPROM
AVERAGING _WINDOW_COUNTER_A	16-bit register sets the averaging window size for COUNTER_A and COUNTER_A_16. Note: averaging is only applied in frequency counter mode, not in counter mode. (Range 0 to 65535, 0=window mode turned off)..	u_16	4419	RAM/EEPROM
COUNTER_A_MODE	8-bit register sets the input mode for D1, COUNTER_A and COUNTER_A_16. (see Counter A Mode Setup)	U_8	8196	RAM/EEPROM
COUNTER_A_TEXT	Text display for COUNTER_A.	L_14	16427	EEROM
UNITS_TEXT_COUNTER_A	Units text for COUNTER_A. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17441	EEROM
DISPLAY_FORMAT_COUNTER_A	8-bit register. Controls the display format settings for COUNTER_A (displayed in octal format).	O_8	8317	RAM/EEPROM
TEXT_CHARACTER_COUNTER_A	8-bit register. Holds the ASCII value for the last digit text character for COUNTER_A (0 = no character).	U_8	8371	RAM/EEPROM

See also

Counter A

Counter A Mode Setup

Counters

2.5.2 Counter B

Counter B can be operated in various count and frequency counter modes.

Name	Description	Symbol Type	Register Number	Memory Type
COUNTER_B	32-bit register that holds the processed data for counter B. If 32 point linearization is applied to counter B then this register will hold the linearized output value. If linearization is disabled, this register will be a copy of the COUNTER_B_SCALED register. (Data may be count or frequency data). Scaling and decimal point values are based on those specified in the Counter B Setup Registers)	S_32	527	RAM
COUNTER_B_SCALED	32-bit register that holds the scaled data for counter B. (Data may be count or frequency data). Scaling and decimal point values are based on those specified in the Counter B Setup Registers)	S_32	711	RAM
COUNTER_B_RAW	32-bit register that holds the raw counter value for counter B before scaling is applied. This value is saved in NV memory at power down and can be restored at power up. (see Counter B Mode Setup for more info)	S_32	371	RAM/FLASH
COUNTER_B_16	16-bit register that holds the processed data for COUNTER_B_16. (Range from 0 - 65535) This register is used to maintain backwards compatibility with older Intech products.	U_16	114	RAM
COUNTER_B_FLOAT	32-bit register that holds a pseudo floating point image of processed data for counter B. Scaling and decimal point values are based on those specified in the Counter B Setup Registers). (See 32-bit Pseudo Floating Point).	PF_32	1807	RAM
DI B	1-bit read only flag that indicates the status of the DI B digital input pin. (See Internal Digital Inputs)	B_1_R	4108	RAM

NOTE: Most of the above registers are normally updated by the operating system of the controller after a new input sample is processed. If the channel is disabled or in a counter mode, it is also possible to modify the contents of the register by writing to it from the setpoint reset logic, from the Macro, or via the serial port. A write to these registers in other operational modes may result in the newly written value being overwritten by the operating system in the controller. (See Digital Input Mode, Counter Input Mode and Frequency Counter Input Mode)

See also
Counters

Counter B Setup Registers

Counter B Mode Setup

2.5.2.1 Counter B Setup Registers

Name	Description	Symbol Type	Register Number	Memory Type
OFFSET_COUNTER_B	32-bit register. Holds the calibration offset for COUNTER_B.	S_32	361	RAM/EEPROM
SCALE_FACTOR_COUNTER_B	32-bit floating point register. Holds the calibration scale factor for COUNTER_B.	F_32	1131	RAM/EEPROM
AVERAGING_SAMPLES_COUNTER_B	8-bit register sets the averaging samples for COUNTER_Band COUNTER_B_16. Note: averaging is only applied in frequency counter mode, not in counter mode. (Range 0 to 255, 0=off)..	U_8	8312	RAM/EEPROM

AVERAGING _WINDOW_COUNTER_B	16-bit register sets the averaging window size for COUNTER_B and COUNTER_B_16. Note: averaging is only applied in frequency counter mode, not in counter mode. (Range 0 to 65535, 0=window mode turned off)..	u_16	4420	RAM/EEPROM
COUNTER_B_MODE	8-bit register sets the input mode for D1, COUNTER_B and COUNTER_B_16. (see Counter B Mode Setup)	U_8	8197	RAM/EEPROM
COUNTER_B_TEXT	Text display for COUNTER_B.	L_14	16429	EEROM
UNITS_TEXT_COUNTER_B	Units text for COUNTER_B. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17443	EEROM
DISPLAY_FORMAT_COUNTER_B	8-bit register. Controls the display format settings for COUNTER_B (displayed in octal format).	O_8	8318	RAM/EEPROM
TEXT_CHARACTER_COUNTER_B	8-bit register. Holds the ASCII value for the last digit text character for COUNTER_B (0 = no character).	U_8	8372	RAM/EEPROM

See also
Counter B

Counter B Mode Setup

Counters

2.5.3 Counter C

Counter C can be operated in various count and frequency counter modes.

Name	Description	Symbol Type	Register Number	Memory Type
COUNTER_C	32-bit register that holds the processed data for counter VC. If 32 point linearization is applied to counter C then this register will hold the linearized output value. If linearization is disabled, this register will be a copy of the COUNTER_C_SCALED register. (Data may be count or frequency data). Scaling and decimal point values are based on those specified in the Counter C Setup Registers)	S_32	529	RAM
COUNTER_C_SCALED	32-bit register that holds the scaled data for counter C. (Data may be count or frequency data). Scaling and decimal point values are based on those specified in the Counter C Setup Registers)	S_32	713	RAM
COUNTER_C_RAW	32-bit register that holds the raw counter value for counter C before scaling is applied. This value is saved in NV memory at power down and can be restored at power up. (see Counter C Mode Setup for more info)	S_32	373	RAM/FLASH
COUNTER_C_16	16-bit register that holds the processed data for COUNTER_C_16. (Range from 0 - 65535) This register is used to maintain backwards compatibility with older Intech products.	U_16	115	RAM
COUNTER_C_FLOAT	32-bit register that holds a pseudo floating point image of the processed data for counter C. Scaling and decimal point values are based on those specified in the Counter C Setup Registers). (See 32-bit Pseudo Floating Point).	PF_32	1809	RAM
DI C	1-bit read only flag that indicates the status of the DI C digital input pin. (See Internal Digital Inputs)	B_2_R	4108	RAM

NOTE: Most of the above registers are normally updated by the operating system of the controller after a new input sample is processed. If the channel is disabled or in a counter mode, it is also possible to modify the contents of the register by writing to it from the setpoint reset logic, from the Macro, or via the serial port. A write to these registers in other operational modes may result in the newly written value being overwritten by the operating system in the controller. (See Digital Input Mode, Counter Input Mode and Frequency Counter Input Mode)

See also
Counters

Counter C Setup Registers

Counter C Mode Setup

2.5.3.1 Counter C Setup Registers

Name	Description	Symbol Type	Register Number	Memory Type
OFFSET_COUNTER_C	32-bit register. Holds the calibration offset for COUNTER_C.	S_32	363	RAM/EEPROM
SCALE_FACTOR_COUNTER_C	32-bit floating point register. Holds the calibration scale factor for COUNTER_C.	F_32	1133	RAM/EEPROM
AVERAGING_SAMPLES_COUNTER_C	8-bit register sets the averaging samples for COUNTER_C and COUNTER_C_16. Note: averaging is only applied in frequency counter mode, not in counter mode. (Range 0 to 255, 0=off)..	U_8	8313	RAM/EEPROM
AVERAGING _WINDOW_COUNTER_C	16-bit register sets the averaging window size for COUNTER_C and COUNTER_C_16. Note: averaging is only applied in frequency counter mode, not in counter mode. (Range 0 to 65535, 0=window mode turned off)..	u_16	4421	RAM/EEPROM
COUNTER_C_MODE	8-bit register sets the input mode for D1, COUNTER_C and COUNTER_C_16. (see Counter C Mode Setup)	U_8	8198	RAM/EEPROM
COUNTER_C_TEXT	Text display for COUNTER_C.	L_14	16431	EEROM
UNITS_TEXT_COUNTER_C	Units text for COUNTER_C. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17445	EEROM
DISPLAY_FORMAT_COUNTER_C	8-bit register. Controls the display format settings for COUNTER_C (displayed in octal format).	O_8	8319	RAM/EEPROM
TEXT_CHARACTER_COUNTER_C	8-bit register. Holds the ASCII value for the last digit text character for COUNTER_C (0 = no character).	U_8	8373	RAM/EEPROM

See also
Counter C

Counter C Mode Setup

Counters

2.5.4 Counter D

Counter D can be operated in various count and frequency counter modes.

Name	Description	Symbol Type	Register Number	Memory Type
COUNTER_D	32-bit register that holds the processed data for counter D. If 32 point linearisation is applied to counter D then this register will hold the linearisaed output value. If linearisation is disabled, this register will be a copy of the COUNTER_D_SCALED register. (Data may be count or frequency data). Scaling and decimal point values are based on those specified in the Counter D Setup Registers)	S_32	531	RAM
COUNTER_D_SCALED	32-bit register that holds the scaled data for counter D. (Data may be count or frequency data). Scaling and decimal point values are based on those specified in the Counter D Setup Registers)	S_32	715	RAM
COUNTER_D_RAW	32-bit register that holds the raw counter value for counter D before scaling is applied. This value is saved in NV memory at power down and can be restored at power up. (see Counter D Mode Setup for more info)	S_32	375	RAM/FLASH
COUNTER_D_16	16-bit register that holds the processed data for COUNTER_D_16. (Range from 0 - 65535) This register is used to maintain backwards compatibility with older Intech products.	U_16	116	RAM
COUNTER_D_FLOAT	32-bit register that holds a pseudo floating point image of the processed data for counter D. Scaling and decimal point values are based on those specified in the Counter D Setup Registers). (See 32-bit Pseudo Floating Point).	PF_32	1811	RAM
DI D	1-bit read only flag that indicates the status of the DI D digital input pin. (See Internal Digital Inputs)	B_3_R	4108	RAM

NOTE: Most of the above registers are normally updated by the operating system of the controller after a new input sample is processed. If the channel is disabled or in a counter mode, it is also possible to modify the contents of the register by writing to it from the setpoint reset logic, from the Macro, or via the serial port. A write to these registers in other operational modes may result in the newly written value being overwritten by the operating system in the controller. (See Digital Input Mode, Counter Input Mode and Frequency Counter Input Mode)

See also

Counters

Counter D Setup Registers

Counter D Mode Setup

2.5.4.1 Counter D Setup Registers

Name	Description	Symbol Type	Register Number	Memory Type
OFFSET_COUNTER_D	32-bit register. Holds the calibration offset for COUNTER_D.	S_32	365	RAM/EEPROM
SCALE_FACTOR_COUNTER_D	32-bit floating point register. Holds the calibration scale factor for COUNTER_D.	F_32	1135	RAM/EEPROM
AVERAGING_SAMPLES_COUNTER_D	8-bit register sets the averaging samples for COUNTER_D and COUNTER_D_16. Note: averaging is only applied in frequency counter mode, not in counter mode. (Range 0 to 255, 0=off)..	U_8	8314	RAM/EEPROM

AVERAGING _WINDOW_COUNTER_D	16-bit register sets the averaging window size for COUNTER_D and COUNTER_D_16. Note: averaging is only applied in frequency counter mode, not in counter mode. (Range 0 to 65535, 0=window mode turned off)..	u_16	4422	RAM/EEPROM
COUNTER_D_MODE	8-bit register sets the input mode for D1, COUNTER_D and COUNTER_D_16. (see Counter D Mode Setup)	U_8	8199	RAM/EEPROM
COUNTER_D_TEXT	Text display for COUNTER_D.	L_14	16433	EEROM
UNITS_TEXT_COUNTER_D	Units text for COUNTER_D. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17447	EEROM
DISPLAY_FORMAT_COUNTER_D	8-bit register. Controls the display format settings for COUNTER_D (displayed in octal format).	O_8	8320	RAM/EEPROM
TEXT_CHARACTER_COUNTER_D	8-bit register. Holds the ASCII value for the last digit text character for COUNTER_D (0 = no character).	U_8	8374	RAM/EEPROM

See also
Counter D

Counter D Mode Setup

Counters

2.6 Data Logging

Most registers from register #1 to register #32765 can be logged. Registers are logged according to what type of register they are, with floating point and text registers also able to be logged. The Zen IoT controllers can log up to 32 different channels in each sample (depending on the data type/size of each channel being logged).

31,774 samples (data records) can be stored (logged) in internal non-volatile memory for before and after analysis of any process condition.

Note: Some models of Zen IoT controllers do come with RTC and data FLASH memory installed. The Zen IoT controller must have RTC/data FLASH memory installed for data logging to function.

Data logging can be triggered (activated) from the logging timer, a setpoint, a front panel button, an external switch, via the serial port or from a macro command. With a real-time clock installed, date and time stamps can be included.

See also
Data Logging Concepts

Name	Description	Symbol Type	Register Number	Memory Type
LOG_READ_COUNT	16-bit register. Sets the number of log samples to read using register 16555 (range 0 to 65535).	U_16	4439	RAM/EEPROM
LOG_WRITE_POINTER	32-bit register. Points to the most recent data log sample number written by the controller. (Pointer is pre-incremented before each new sample is written).	U_32	489	RAM/FLASH/SDcard
LOG_READ_POINTER	32-bit register. Pointer to the most recent data log sample number read by the controller. Pre-incremented before each read of 16553.	U_32	491	RAM/FLASH/SDcard

LOG_REVERSE_READ_POINTER	32-bit register. Pointer to the next data log sample number to be read by the controller when using the	U_32	485	RAM/FLASH
DELIMIT_CHAR	8-bit register. Holds delimiting character for spread sheet output mode. Value is held in volatile RAM which defaults to Horizontal Tab (0x9) at power on.	U_8	8452	RAM
LOG_INTERVAL_TIME	32-bit register. Logging interval time. Specifies the amount of time at which log samples are taken in 0.1 second resolution.	U_32	483	RAM/EEPROM
LOGGING_MODE	8-bit register. Enables data logging and controls buffer type, time stamp and manual trigger modes. (see Logging Mode Setup)	O_8	8200	RAM/EEPROM
LOG_REG1	16-bit register. Contains register number of 1st register logged in sample.	U_16	4275	RAM/EEPROM
LOG_REG2	16-bit register. Contains register number of 2nd register logged in sample.	U_16	4276	RAM/EEPROM
LOG_REG3	16-bit register. Contains register number of 3rd register logged in sample.	U_16	4277	RAM/EEPROM
LOG_REG4	16-bit register. Contains register number of 4th register logged in sample.	U_16	4278	RAM/EEPROM
LOG_REG5	16-bit register. Contains register number of 5th register logged in sample.	U_16	4279	RAM/EEPROM
LOG_REG6	16-bit register. Contains register number of 6th register logged in sample.	U_16	4280	RAM/EEPROM
LOG_REG7	16-bit register. Contains register number of 7th register logged in sample.	U_16	4281	RAM/EEPROM
LOG_REG8	16-bit register. Contains register number of 8th register logged in sample.	U_16	4282	RAM/EEPROM
LOG_REG9	16-bit register. Contains register number of 9th register logged in sample.	U_16	4283	RAM/EEPROM
LOG_REG10	16-bit register. Contains register number of 10th register logged in sample.	U_16	4284	RAM/EEPROM
LOG_REG11	16-bit register. Contains register number of 11th register logged in sample.	U_16	4285	RAM/EEPROM
LOG_REG12	16-bit register. Contains register number of 12th register logged in sample.	U_16	4286	RAM/EEPROM
LOG_REG13	16-bit register. Contains register number of 13th register logged in sample.	U_16	4287	RAM/EEPROM
LOG_REG14	16-bit register. Contains register number of 14th register logged in sample.	U_16	4288	RAM/EEPROM
LOG_REG15	16-bit register. Contains register number of 15th register logged in sample.	U_16	4289	RAM/EEPROM
LOG_REG16	16-bit register. Contains register number of 16th register logged in sample.	U_16	4290	RAM/EEPROM
LOG_REG17	16-bit register. Contains register number of 17th register logged in sample.	U_16	4291	RAM/EEPROM
LOG_REG18	16-bit register. Contains register number of 18th register logged in sample.	U_16	4292	RAM/EEPROM
LOG_REG19	16-bit register. Contains register number of 19th register logged in sample.	U_16	4293	RAM/EEPROM

LOG_REG20	16-bit register. Contains register number of 20th register logged in sample.	U_16	4294	RAM/EEPROM
LOG_REG21	16-bit register. Contains register number of 21th register logged in sample.	U_16	4295	RAM/EEPROM
LOG_REG22	16-bit register. Contains register number of 22th register logged in sample.	U_16	4296	RAM/EEPROM
LOG_REG23	16-bit register. Contains register number of 23th register logged in sample.	U_16	4297	RAM/EEPROM
LOG_REG24	16-bit register. Contains register number of 24th register logged in sample.	U_16	4298	RAM/EEPROM
LOG_REG25	16-bit register. Contains register number of 25th register logged in sample.	U_16	4299	RAM/EEPROM
LOG_REG26	16-bit register. Contains register number of 26th register logged in sample.	U_16	4300	RAM/EEPROM
LOG_REG27	16-bit register. Contains register number of 27th register logged in sample.	U_16	4301	RAM/EEPROM
LOG_REG28	16-bit register. Contains register number of 28th register logged in sample.	U_16	4302	RAM/EEPROM
LOG_REG29	16-bit register. Contains register number of 29th register logged in sample.	U_16	4303	RAM/EEPROM
LOG_REG30	16-bit register. Contains register number of 30th register logged in sample.	U_16	4304	RAM/EEPROM
LOG_REG31	16-bit register. Contains register number of 31st register logged in sample.	U_16	4305	RAM/EEPROM
LOG_REG32	16-bit register. Contains register number of 32nd register logged in sample.	U_16	4306	RAM/EEPROM

Name	Description	Symbol Type	Register Number	Memory Type
MAX_LOG_SAMPLES	This 32 bit unsigned read only register reports how many log samples are available for the current data logging configuration.	U_32_R	487	RAM

See also

Maximum Number Of Log Samples

Log Write Pointer

Log Read Pointer

Numeric Log Sample Values

Log Register Source

Number Of Log Sample Reads

Read Log Sample Data

Read Single Log Data At Log Read Pointer

Read Log Data At Log Read Pointer

Read Only Registers

2.6.1 Data Logging Concepts

The data logging function uses the concept of pointers to control where a sample is to be written to and from where one is to be read. These pointers are referred to as the log write pointer and the log read pointer.

Log Write Pointer
Register 489 is a 32-bit register that points to the most recent log sample written by the controller. It counts up from 0 each time a new sample is logged, with the maximum number of samples being limited by the size of non-volatile memory installed in the controller and also the number/size of registers to be logged. Before a new sample is written, the controller first checks to make sure that it is not overwriting a sample that has not been read. It does this by comparing the write pointer with the read pointer. If they are the same and the Linear logging mode has been selected, data logging is halted until a read is actioned. If this occurs, new samples are lost. If the Cyclic mode has been selected, the oldest sample will be overwritten with new data and the old sample will be lost. When the sample number reaches the maximum count it wraps around to 1.

Register 489 can be read from or written to. Make sure that any values written to this pointer are within the allowable range for the size of the installed memory.

Log Read Pointer
Register 491 is a 32-bit register that points to the most recent log sample read from the controller. It counts up from 0 each time log data is read from the controller, with the maximum number of samples being limited by the size of non-volatile memory installed in the controller and also the number/size of registers to be logged. When it reaches the maximum count it wraps around to 1. When it reaches the write pointer the log buffer is empty and no more data can be read out of the log.

Register 491 can be read from or written to. Make sure that any values written to this pointer are within the allowable range for the size of the installed memory.

Note: Although the log read and write pointers can be reset to zero, sample zero is never used to hold any real sample data. It is only used as "resting point" when the pointers are cleared. This is because the pointers are always pre-incremented before the sample is written. When pointers wrap around at the end of memory they wrap around to the value of 1.

Log Reverse Read Pointer
Register 485 is a 32-bit register that points to the next log sample to be read from the controller using the reverse read register 16549. It is decremented **after** each read from 16549 and works its way down until it reaches the current value of the read pointer. When it reaches the read pointer it stops and no further log samples are sent out. Unlike registers 489 and 491, this register resides in volatile RAM only and must be setup prior to a block read operation. When it reaches the minimum count of 1 it wraps around to the maximum sample number.

Register 485 can be read from or written to. Make sure that any values written to this pointer are within the allowable range for the size of the installed memory.

Buffer Types
The controller has two types of buffer.

Cyclic Buffer. With the **cyclic** buffer selected in the Meter Configuration Utility program, the log write pointer (register 489) increments each time a sample is taken. When it exceeds the maximum sample number (determined by the amount of non-volatile memory installed and the number/size of registers to be logged) it wraps around to zero. If the write pointer equals the read pointer then oldest (unread) data will be overwritten with the new data and old data will lost. This means that when the cyclic buffer is full, the logged data is replaced on a first ON first OFF basis. This means that when the buffer is full, the first logged sample is discarded to make way for a new sample at the end of the logged data string. It then wraps around to sample number 1 again.

See description on Log Read Pointer for information about not overwriting old samples that have not been read.

Linear Buffer. With the **linear** buffer selected in the Meter Configuration Utility program, the log write pointer increments each time a sample is taken until it reaches the read pointer. When it equals the read pointer the controller stops logging data and any new data is lost. If the sample number reaches the maximum sample number (determined by the amount of non-volatile memory installed and the number/size of registers to be logged) it will wrap around to zero. When the linear buffer is full it must either be read or reset to 0. See Reset Buffer.

Reset Buffer

With **reset buffer number to 0** set in the Meter Configuration Utility program, the log write and log read pointers are reset to zero when the PROGRAM button is pressed. The controller then reverts back to the same setting it had before the reset function was executed (either cyclic or linear). Note that when the reset function is executed, the contents of the buffer is not destroyed, only the pointers are changed.

Note: The reset buffer function only works from a display panel. To reset pointers via the serial port they should just be written to individually.

Registers 4275 to 4306

Registers 4275 to 4306 can be read from and written to as normal registers. Registers 4275 to 4306 can only be configured via the serial port or from the macro and are not accessible from the front panel buttons.

Registers 4275 to 4306 are used to specify which registers are to be logged. Register 4275 specifies the first register to be logged, 4276 the second, 4277 the third, and so on. Writing a value of zero to one of these registers disables the register from taking any logs.

Up to 32 registers in total can be logged in each sample however the actual number depends on the size of each register being logged. The overall size of each stored sample is 132 bytes and each sample has a fixed overhead of 8 bytes as shown below.

Each sample will always include:
- 1 byte required for trigger source.
- 3 bytes required for date stamp.
- 3 bytes required for time stamp.
- 1 byte required for checksum (last byte in sample).

This means that all the logged channels must fit into the remaining 124 bytes. The data in each sample can be made of a combination of any of the following data types:
- 8-bit registers use: 1 byte
- 16-bit registers use: 2 bytes
- 24-bit registers use: 3 bytes
- 32-bit registers use: 4 bytes
- Text registers: Number of bytes depends on length of text string.
- (Custom Text strings can be stored but only with special macro command "log_message" or via a serial port write to register 16553. See Read Single Log Data at Log Read Pointer - Register 16553)

Non-volatile Memory Options

When the controller is fitted with the internal data logging memory option, then 32Mbit of non-volatile on-board memory (data FLASH) is installed. This allows up to 31,774 samples to be logged.

NOTE: Altering registers 4275 to 4306 will potentially effect the order of the data within a sample and may also render any previously stored log data as unreadable. All current log data should be read and saved before changes are made to these registers and they should be correctly configured before any new samples are taken.

Special Log Functions

Most data log samples are trigger by the log timer, a setpoint or from an input pin but log samples can also be triggered from other sources such as special macro commands or from the serial port. There are 2 special macro commands which allow log samples to be triggered as shown below.

"force_log" command - This macro command triggers a log sample to be taken in the standard format.

The sample is taken the instant the command is executed.

"log_message" command - This macro command allows a text string to be logged by the data logger. It requires a following text string enclosed in quotation marks (") and allows the logging of custom messages from the macro. The maximum length of the text string is governed by the data logging settings for registers 4275 - 4306.

There are also 2 ways of triggering a data log from the serial port as shown below.

Write to register 8442 - a write to register 8442 via the serial port will trigger a log sample to be taken in the standard format. It allows log samples to be triggered from another serial device.

Write to register 16553 - a write to register 16553 via the serial port allows a text string to be logged in a similar manner to the "log_message" macro command. In this case the text string to be logged is included in the serial command in a similar manner to writing to other text registers.

2.6.2 Read Only Registers

These read only registers are provided to allow the user to selectively read a single parameter from a log sample, instead of reading all parameters in a sample. Only numeric parameters can be read (not text) and they will always be drawn from the sample which the log read pointer is currently pointing to. The user must ensure that the log read pointer is pointing to the correct sample of number before reading these registers. On previous firmware versions (earlier than 4.04.01), reading any of these registers does not alter the log read pointer position. On the Zen IoT an auto increment feature has been added.

Name	Description	Symbol Type	Register Number	Memory Type
LOG_SAMPLE_TRIGGER	Read only register. Trigger source of current log sample.	U_R	8443	EEPROM/SDcard
LOG_SAMPLE_DATE	Read only register. Returns 8-bit value for date of current log sample (range 1 to 31 days).	U_R	8444	EEPROM/SDcard
LOG_SAMPLE_MONTH	Read only register. Returns 8-bit value for months of current log sample (range 1 to 12 months).	U_R	8445	EEPROM/SDcard
LOG_SAMPLE_YEAR	Read only register. Returns 8-bit value for year of current log sample (range 00 to 99 years).	U_R	8446	EEPROM/SDcard
LOG_SAMPLE_HOUR	Read only register. Returns 8-bit value for hours of current log sample (range 0 to 23 hours).	U_R	8447	EEPROM/SDcard
LOG_SAMPLE_MINUTE	Read only register. Returns 8-bit value for minutes of current log sample (range 0 to 59 minutes).	U_R	8448	EEPROM/SDcard
LOG_SAMPLE_SECOND	Read only register. Returns 8-bit value for seconds of current log sample (range 0 to 59 seconds).	U_R	8449	EEPROM/SDcard
LOG_SAMPLE_HUNDREDTHS	Read only register. Returns 8-bit value for 1/100 of a second of current log sample (range 0 to 99 hundredths of a second).	U_R	8450	EEPROM/SDcard

LOG_SAMPLE_REG1	Read only register. Returns 1st data value for logged in current log sample (range depends on size of 1st register).	S_R	493	EEPROM/SDcard
LOG_SAMPLE_REG2	Read only register. Returns 2nd data value for logged in current log sample (range depends on size of 2nd register).	S_R	495	EEPROM/SDcard
LOG_SAMPLE_REG3	Read only register. Returns 3rd data value for logged in current log sample (range depends on size of 3rd register).	S_R	497	EEPROM/SDcard
LOG_SAMPLE_REG4	Read only register. Returns 4th data value for logged in current log sample (range depends on size of 4th register).	S_R	499	EEPROM/SDcard
LOG_SAMPLE_REG5	Read only register. Returns 5th data value for logged in current log sample (range depends on size of 5th register).	S_R	501	EEPROM/SDcard
LOG_SAMPLE_REG6	Read only register. Returns 6th data value for logged in current log sample (range depends on size of 6th register).	S_R	503	EEPROM/SDcard
LOG_SAMPLE_REG7	Read only register. Returns 7th data value for logged in current log sample (range depends on size of 7th register).	S_R	505	EEPROM/SDcard
LOG_SAMPLE_REG8	Read only register. Returns 8th data value for logged in current log sample (range depends on size of 8th register).	S_R	507	EEPROM/SDcard
LOG_SAMPLE_REG9	Read only register. Returns 9th data value for logged in current log sample (range depends on size of 9th register).	S_R	509	EEPROM/SDcard
LOG_SAMPLE_REG10	Read only register. Returns 10th data value for logged in current log sample (range depends on size of 10th register).	S_R	511	EEPROM/SDcard
LOG_SAMPLE_REG11	Read only register. Returns 11th data value for logged in current log sample (range depends on size of 11th register).	S_R	513	EEPROM/SDcard
LOG_SAMPLE_REG12	Read only register. Returns 12th data value for logged in current log sample (range depends on size of 12th register).	S_R	515	EEPROM/SDcard
LOG_SAMPLE_REG13	Read only register. Returns 13th data value for logged in current log sample (range depends on size of 13th register).	S_R	517	EEPROM/SDcard
LOG_SAMPLE_REG14	Read only register. Returns 14th data value for logged in current log sample (range depends on size of 14th register).	S_R	519	EEPROM/SDcard
LOG_SAMPLE_REG15	Read only register. Returns 15th data value for logged in current log sample (range depends on size of 15th register).	S_R	521	EEPROM/SDcard
LOG_SAMPLE_REG16	Read only register. Returns 16th data value for logged in current log sample (range depends on size of 16th register).	S_R	523	EEPROM/SDcard
LOG_SAMPLE_REG17	Read only register. Returns 17th data value for logged in current log sample (range depends on size of 17th register).	S_R	545	EEPROM/SDcard
LOG_SAMPLE_REG18	Read only register. Returns 18th data value for logged in current log sample (range depends on size of 18th register).	S_R	547	EEPROM/SDcard
LOG_SAMPLE_REG19	Read only register. Returns 19th data value for logged in current log sample (range depends on size of 19th register).	S_R	549	EEPROM/SDcard
LOG_SAMPLE_REG20	Read only register. Returns 20th data value for logged in current log sample (range depends on size of 20th register).	S_R	551	EEPROM/SDcard

LOG_SAMPLE_REG21	Read only register. Returns 21th data value for logged in current log sample (range depends on size of 21th register).	S_R	553	EEPROM/SDcard
LOG_SAMPLE_REG22	Read only register. Returns 22th data value for logged in current log sample (range depends on size of 22th register).	S_R	555	EEPROM/SDcard
LOG_SAMPLE_REG23	Read only register. Returns 23th data value for logged in current log sample (range depends on size of 23th register).	S_R	557	EEPROM/SDcard
LOG_SAMPLE_REG24	Read only register. Returns 24th data value for logged in current log sample (range depends on size of 24th register).	S_R	559	EEPROM/SDcard
LOG_SAMPLE_REG25	Read only register. Returns 25th data value for logged in current log sample (range depends on size of 25th register).	S_R	561	EEPROM/SDcard
LOG_SAMPLE_REG26	Read only register. Returns 26th data value for logged in current log sample (range depends on size of 26th register).	S_R	563	EEPROM/SDcard
LOG_SAMPLE_REG27	Read only register. Returns 27th data value for logged in current log sample (range depends on size of 27th register).	S_R	565	EEPROM/SDcard
LOG_SAMPLE_REG28	Read only register. Returns 28th data value for logged in current log sample (range depends on size of 28th register).	S_R	567	EEPROM/SDcard
LOG_SAMPLE_REG29	Read only register. Returns 29th data value for logged in current log sample (range depends on size of 29th register).	S_R	569	EEPROM/SDcard
LOG_SAMPLE_REG30	Read only register. Returns 30th data value for logged in current log sample (range depends on size of 30th register).	S_R	571	EEPROM/SDcard
LOG_SAMPLE_REG31	Read only register. Returns 31st data value for logged in current log sample (range depends on size of 31st register).	S_R	573	EEPROM/SDcard
LOG_SAMPLE_REG32	Read only register. Returns 32nd data value for logged in current log sample (range depends on size of 32nd register).	S_R	575	EEPROM/SDcard

2.6.3 Maximum Number Of Log Samples

Register 487 is a 32-bit read only register that reports the maximum number of log samples available. The maximum number of log samples is defined by the amount of memory installed with the default memory size giving 31,774 samples.

2.6.4 Log Write Pointer

Register 489 is a 32-bit register that points to the most recent log sample written by the controller. It automatically increments by one count just before a new sample is logged and counts up from 0 with the maximum number of samples being limited by the current memory size installed in the controller and the setting of registers 4275 to 4306. When it reaches the maximum count it wraps around to 1. When it catches up to the log read pointer (491) it either overwrites the old unread data or stops logging, depending on which type of buffer has been selected in the Meter Configuration Utility program (i.e. cyclic or linear).

Register 489 can be read or written to. You must make sure that any values written to this pointer are within the allowable range for the installed memory size.

Note: Although the log read and write pointers can be reset to zero, sample zero is never used to hold any real sample data. It is only used as "resting point" when the pointers are cleared. This is because the pointers are always pre-incremented before the sample is written. When pointers wrap around at the end of memory they wrap around to the value of 1.

Note: If a firmware update is performed on the Zen IoT, the contents of the log write and log read pointers will be lost and will be set to 0. Log data will remain intact, but to ensure continuous logging the value of the log read and log write pointers should be recorded first, and then manually restored after the firmware update.

2.6.5 Log Read Pointer

Register 491 is a 32-bit register that points to the most recent log sample read from the controller. It counts up from 0 each time log data is read from the controller, with the maximum number of samples being limited by the current memory size installed in the controller and the setting of registers 4275 to 4306. When it reaches the maximum count it wraps around to 1. When it reaches the log write pointer, the data log buffer is empty and it stops.

Register 491 can be read or written to. You must make sure that any values written to this pointer are within the allowable range for the installed memory size.

Note: Although the log read and write pointers can be reset to zero, sample zero is never used to hold any real sample data. It is only used as "resting point" when the pointers are cleared. This is because the pointers are always pre-incremented before the sample is written. When pointers wrap around at the end of memory they wrap around to the value of 1.

Note: If a firmware update is performed on the Zen IoT, the contents of the log write and log read pointers will be lost and will be set to 0. Log data will remain intact, but to ensure continuous logging the value of the log read and log write pointers should be recorded first, and then manually restored after the firmware update.

2.6.6 Numeric Log Sample Values

Registers 493 to 523 and 545 to 575 provide numeric values for the 1st through to the 32nd register logged in accordance with registers 4275 to 4306. The size and type varies depending on the size and data type of the registers addressed by registers 4275 to 4306. These registers give an unformatted numeric value that can be read in ASCII or any other serial mode.

The user should ensure that the log read pointer is set to the required sample number before accessing any of these registers.

If the sample does not contain the data that is requested, the controller responds by sending a null character in ASCII mode, or a data error in Modbus mode.

2.6.7 Log Register Source

Registers 4275 to 4306 are used to specify which registers the data logger logs. Register 4275 specifies the first register to be logged, 4276 the second, 4277 the third and so on. Setting one of these registers to zero disables that register from logging any data. Changing registers 4275 to 4306 potentially changes the format and order of data within a sample and may also render any previously stored log data as unreadable. All current log data should be read and saved before changes are made to these registers and they should be correctly configured before any new samples are taken.

Registers 4275 to 4306 can be read from and written to as normal registers. Most of the registers in the controller can be logged. This includes floating point and text registers.

NOTE: Text registers 16553, 16555 and 16543 should not be logged.

2.6.8 Number Of Log Sample Reads

Register 4462 defines the maximum number of log samples that will be output when register 16555 is read. This can be set to any number between 1 and 65535 with a default of 100 samples.

Note: This is only relevant in ASCII mode.

2.6.9 Read Log Sample Data

Registers 8443 to 8450 are read only registers used to access data from a log sample. A read of one of these registers reads the appropriate data from the log sample which is addressed by the current value of the log read pointer (register 491). The user must setup the log read pointer to the required sample number before accessing registers 8443 to 8450.

In each case the output is a numeric value only. Registers 8443 to 8450 can be read in ASCII or Modbus modes and can also be read from the macro.

NOTE: The information in registers 8443 to 8450 is logged in every sample, regardless of the settings in the Meter Configuration Utility program.

Register 8443 – Trigger type for sample
This register provides an 8-bit numeric value that defines the trigger point for the sample.

Numeric Value	Function
0	Triggered by reset
1	Triggered by setpoint 1
2	Triggered by setpoint 2
3	Triggered by setpoint 3
4	Triggered by setpoint 4
5	Triggered by setpoint 5
6	Triggered by setpoint 6
7	Triggered by setpoint 7
8	Triggered by setpoint 8
9	Triggered by setpoint 9
10	Triggered by setpoint 10
11	Triggered by setpoint 11
12	Triggered by setpoint 12
13 - 16	Reserved for future development
17	Triggered by Program button
18	Triggered by F1 button
19	Triggered by F2 button
20	Triggered by Hold pin
21	Triggered by Lock pin
22 - 23	Reserved for future development
24	Triggered by executing the macro instruction "force_log" or by a serial port write to register 8442.
25	Triggered by executing the macro instruction "log_message" or by a text string write to register 16553 via the serial port.
26	Triggered from internal logging timer.
27	Triggered by D1 digital input pin.
28	Triggered by D2 digital input pin.
29	Triggered by D3 digital input pin.
30	Triggered by D4 digital input pin.

Register 8444 – Date of sample
This register provides an 8-bit numeric value for the date.

Register 8445 – Month of sample
This register provides an 8-bit numeric value for the month.

Register 8446 – Year of sample
This register provides an 8-bit numeric value for the last 2 digits of the year.

Register 8447 – Hour of sample

This register provides an 8-bit numeric value for the hours.

Register 8448 – Minute of sample
This register provides an 8-bit numeric value for the minutes.

Register 8449 – Second of sample
This register provides an 8-bit numeric value for the seconds.

Register 8450 – 1/100 Second of sample
This register provides an 8-bit numeric value for hundredths of a second.

2.6.10 Read Single Log Data at Log Read Pointer

Register 16553 is used to read the next sample of log data. It does this by comparing the log read pointer (register 491) with the log write pointer (register 489). If they are equal then there has been no new samples logged since the last read and the message **No New Log Data** is sent as a response. If they are not equal, the log read pointer (register 491) is incremented to point to the new sample and the new log data is transmitted. Registers 489 and 491 can be used to control the data logger. To reset the data logger, both register 489 and 491 should be set to 0 (or any other value in the allowable range of memory).

Reading Register 16553 In ASCII Mode
Although register 16553 can be read in ASCII serial mode, it is not recommended if more than 10 channels of data are being logged (see note below on buffer overflow problems). Modbus mode is the recommended mode to read large amounts of data logging memory. To read log data in other serial modes, see **registers 493 to 523 and registers 8443 to 8450**.

Modbus Read Of 16553
Register 16553 can also be read in Modbus mode. Modbus mode should be used if logging more than 10 channels or if using a uSD card with large numbers of samples. It provides a faster and more efficient method of downloading large amounts of data than via the ASCII mode and does not suffer from buffer overflow problems (see note below)

A read of register 16553 in Modbus mode produces a raw output format which needs to be decoded by the user. The data bytes within the Modbus packet is formatted as follows;

Fixed Format Section
Data bytes 1 to 7 of each sample have a fixed format as follows;

1st data byte = Log trigger (see Register 8443 – Trigger type for sample)
2nd data byte = Date (see Register 8444 – Date of sample)
3rd data byte = Month (see Register 8445 – Month of sample)
4th data byte = Year (see Register 8446 – Year of sample)
Data bytes 5-7 = Time stamp as 24 bit unsigned number in 1/100 second resolution.

Variable Format Section
Data bytes 8 and onwards do not have a fixed format. The format of these data bytes is determined by the data logging configuration (i.e. how many registers are being logged and which type of registers are being logged). The length of the data string is also effected by data logging configuration. To determine how many registers need to be read and how to decode them, the user should first read register 16551, the data logging format register.

Writing To Register 16553
A write to register 16553 via the serial port can be used to log a custom text string into data logging memory. The maximum length of the text string is governed by the values written to registers 4417 to 4432 as these effectively control the sample size. In ASCII mode the format would be;

SW16553,Hello World*

This allows text messages from another source to be time stamped and logged along with standard log samples. Data logging must be enabled in the Meter Configuration Utility program for this feature to function. (See Code 8)

A standard log sample can also be trigger from the serial port by writing any value (normally 0) to register #8442. This will cause the registers selected by 4417 - 4432 to be logged and time stamped in the standard format.

Data Log Format - Register 16551
Register 16551 can only be read in Modbus mode. A read of register 16551 will produce the following data.

Byte 1 = length of log format register (number of data bytes in packet)
Byte 2 = Internal sample size for data log in Flash memory.
Byte 3,4 = Controller software version (e.g. 403)
Byte 5 = Not Used in Zen IoT (always read as 0)
Byte 6 = Time format (current configuration of Code 8)

Byte 7,8 = Data log source register |
Byte 9 = Register type | repeated for each register that is logged
Byte 10 = Display format |
Byte 11 = Alpha character |

If the data logging configuration (see Data Logging and Log Register Source - Registers 4275 to 4306) has been set up to log 4 registers per sample, then data bytes 7 - 11 will be repeated for each of the 4 logged registers.

The length byte defines how many bytes of data are contained in the data log format string.
The sample size byte gives the current data log sample size which is determined by the data logging configuration.
The time format byte is basically a copy of Code 8 and specifies how the time/date information is to be displayed.
The two data bytes which form the 16 bit data log source register define the register number of the logged data. The register type byte defines the size and format of the data bytes in the log sample. The different options available are;

0x00 = unsigned char
0x10 = signed char
0x20 = unsigned int
0x30 = signed int
0x40 = unsigned long
0x50 = signed long
0x60 = ASCII TEXT
0x70 = float
0x80 = unsigned 24 bit number(only 3 bytes long)
0x90 = signed 24 bit number(only 3 bytes long)

The display format and alpha character bytes define how the numeric value is to be displayed (i.e. were the decimal point should be and what the trailing text character (if any) should be). (see Display Format Mode for more info on how to interpret the display format byte)

NOTE:
When logging more than 10 registers per sample a buffer overflow can occur when downloading the log data in the printer format via ASCII mode, resulting in truncated output data. If this happens, try switching to spreadsheet format. If you are still experiencing problems then you should consider using Modbus mode as described above.

2.6.11 Read Log Data at Log Read Pointer

Register 16555 is a read only register similar to register 16553, except that it is used to read multiple log samples with a single command. A read of register 16555 outputs log data, starting at the sample pointed to by the read pointer, and continues to output log samples until the read pointer equals the write pointer, or the number of samples specified in register 4462 has been output. Each time it outputs a log sample the read pointer is automatically incremented. At the end of the sequence, the read pointer is equal to the write pointer. This command can also be used to read a selected block of log samples by modifying the read pointer (register 491) and the write pointer (register 489) prior to reading register 16555.

NOTE: Register 16555 can only be read via the serial port in ASCII mode. To read log data in other serial modes or from the macro, see registers 493 to 523 and registers 8443 to 8450.

2.7 Digital I/O

The Zen IoT controller includes 4 digital input pins which can be used as status inputs or in counter/frequency modes.

See Also
Internal Digital Inputs

Counters

Output Masks

2.7.1 Internal Digital Inputs

The Zen IoT controller includes 4 digital input pins. These can be configured for a variety of different input functions including standard digital status inputs, and various counter modes. (For more information on counter modes see Counters)

The 4 digital input pins are isolated from the other Zen IoT pins by opto couplers but all share the same common pin. The maximum frequency of these input pins is limited to approximately 2.5kHz. or pulse widths > 200uS and they are updated in real time when read via the serial port or macro. The status of the 4 digital input pins can be read even if these inputs are configured for other functions (such as counters etc).

Internal Digital Input Register
Register (4108) is a 16 bit register used to show the status of the 4 internal digital input pins.

Name	Description	Symbol Type	Register Number	Memory Type
DIGITAL_INPUT_PINS	16-bit register that shows the status of the internal digital input pins DI A - DI D.	U_16	4108	RAM

The individual bit functions for the DIGITAL_INPUT_PINS register are shown in the table below;

Bit	Description	Function
b0	Status of digital input pin DI A.	1 = DI A on (activated) 0 = DI A off
b1	Status of digital input pin DI B.	1 = DI B on (activated) 0 = DI B off
b2	Status of digital input pin DI C.	1 = DI C on (activated) 0 = DI C off
b3	Status of digital input pin DI D.	1 = DI D on (activated) 0 = DI D off
b3-b15	Reserved for future development	

See Also
Counters

Internal Digital Outputs

Modbus Digital Inputs

2.7.2 Internal Digital Outputs

The Zen IoT controller includes 4 digital outputs in the form of 3 solid state relays (Form A, 0.4A, 30V DC) and 1 latching relay (Form C, 1A, 30V DC). These can be configured for a variety of different output functions including standard outputs, output controllers, serial receive timeout (relay 2 only), and advanced setpoint control.

Note: Relays A - D can be controlled from various sources. See notes on Relay B RX Timeout, Serial Receive Timeout, ASP Control Mode and Controller Mode Registers

Internal Digital Output Register
Register (8205) is a 8 bit register used to control the status of the onboard relays A - D in manual control mode.

Name	Description	Symbol Type	Register Number	Memory Type
RELAY_A_D_CONTROL	8-bit register that controls the output state of onboard relays A - D.	U_8	8205	RAM

The individual bit functions for the DIGITAL_INPUT_PINS register are shown in the table below;

Bit	Description	Function
b0	Output state of relay A.	1 = Relay A on (activated) 0 = Relay A off
b1	Output state of relay B	1 = Relay B on (activated) 0 = Relay B off
b2	Output state of relay C	1 = Relay C on (activated) 0 = Relay C off
b3	Output state of relay D	1 = Relay D on (activated) 0 = Relay D off (**Note**: relay D is a latching relay which will hold its state when power is removed from the controller.)
b4-b7	Reserved for future development	

Zen IoT controllers also support a data source for each relay output. Each register is a 16 bit register which holds a Modbus coil or switch register number (see Modbus Digital Outputs and

Modbus Digital Inputs for Modbus discrete register numbers). This allows you to associate each relay output with virtually any other discrete input or output in the Zen IoT. If the data source value is set to zero, the operation of the relay is identical to older firmware releases (see note above on relay A - D)

Name	Description	Symbol Type	Register Number	Memory Type
DATA_SOURCE_ONBOARD_RELAY_A	16-bit register that points to the data source for relay A. **NOTE**: Only discrete Modbus registers numbers can be used as a data source with a relay output module. (See Modbus Digital Outputs and Modbus Digital Outputs for valid Modbus register numbers).	U_16	4673	RAM/EEPROM
DATA_SOURCE_ONBOARD_RELAY_B	16-bit register that points to the data source for relay B. **NOTE**: Only discrete Modbus registers numbers can be used as a data source with a relay output module. (See Modbus Digital Outputs and Modbus Digital Outputs for valid Modbus register numbers).	U_16	4674	RAM/EEPROM
DATA_SOURCE_ONBOARD_RELAY_C	16-bit register that points to the data source for relay C. **NOTE**: Only discrete Modbus registers numbers can be used as a data source with a relay output module. (See Modbus Digital Outputs and Modbus Digital Outputs for valid Modbus register numbers).	U_16	4675	RAM/EEPROM
DATA_SOURCE_ONBOARD_RELAY_D	16-bit register that points to the data source for relay D. **NOTE**: Only discrete Modbus registers numbers can be used as a data source with a relay output module. (See Modbus Digital Outputs and Modbus Digital Outputs for valid Modbus register numbers).	U_16	4676	RAM/EEPROM

See Also

Relay B RX Timeout

Serial Receive Timeout

ASP Control Mode

Output Masks

Modbus Digital Outputs

2.7.3 Modbus Digital Outputs

The Zen IoT controller supports the following Modbus commands for accessing discrete digital outputs.

Function Code	Description
1	Read coil status
5	Force single coil
15	Force multiple coils

The following table shows a map of all discrete digital outputs that can accessed via the above Modbus commands.

Note: The output numbers shown below are for reference only. These numbers should be decremented by 1 for the coil addresses in the Modbus frame.

Name	Coil Output No.	Description	Type
RA	1	Relay RA (on board Zen IoT).	Read/Write
RB	2	Relay RB (on board Zen IoT).	Read/Write
RC	3	Relay RC (on board Zen IoT).	Read/Write
RD	4	Relay RD (on board Zen IoT).	Read/Write
R5	5	Relay R5 (only available if Zen IoT is fitted with optional relay module in channel 5 slot).	Read/Write
R6	6	Relay R6 (only available if Zen IoT is fitted with optional relay module in channel 6 slot).	Read/Write
R7	7	Relay R7 (only available if Zen IoT is fitted with optional relay module in channel 7 slot).	Read/Write
R8	8	Relay R8 (only available if Zen IoT is fitted with optional relay module in channel 8 slot).	Read/Write
R9	9	Relay R9 (only available if Zen IoT is fitted with optional relay module in channel 9 slot).	Read/Write
R10	10	Relay R10 (only available if Zen IoT is fitted with optional relay module in channel 10 slot).	Read/Write
R11	11	Relay R11 (only available if Zen IoT is fitted with optional relay module in channel 11 slot).	Read/Write
R12	12	Relay R12 (only available if Zen IoT is fitted with optional relay module in channel 12 slot).	Read/Write
R13	13	Relay R13 (only available if Zen IoT is fitted with optional relay module in channel 13 slot).	Read/Write
R14	14	Relay R14 (only available if Zen IoT is fitted with optional relay module in channel 14 slot).	Read/Write
R15	15	Relay R15 (only available if Zen IoT is fitted with optional relay module in channel 15 slot).	Read/Write
R16	16	Relay R16 (only available if Zen IoT is fitted with optional relay module in channel 16 slot).	Read/Write

See Also

Modbus Digital Inputs

Internal Digital Inputs

Internal Digital Outputs

Modbus Mode

2.7.4 Modbus Digital Inputs

The Zen IoT controller supports the following Modbus command for accessing discrete digital inputs.

Function Code	Description
2	Read input switch status

The following table shows a map of all discrete digital outputs that can accessed via the above Modbus commands.

Note: The input numbers shown below are for reference only. These numbers should be decremented by 1 for the switch addresses in the Modbus frame.

Name	Switch Input No.	Description	Type
DI A	1	Digital status input DI A (on board Zen IoT).	Read Only
DI B	2	Digital status input DI B (on board Zen IoT).	Read Only
DI C	3	Digital status input DI C (on board Zen IoT).	Read Only
DI D	4	Digital status input DI D (on board Zen IoT).	Read Only
-	5 - 16	Not used. (Reserved for future development.)	

See Also
Modbus Digital Outputs

Internal Digital Inputs

Internal Digital Outputs

External Digital Inputs

External Digital Outputs

Modbus Mode

2.7.5 Additional Relay Output Modules

Zen IoT controllers support additional relay output modules which can be fitted in the analogue channel slots in place of analogue input/output modules. It is possible to order your Zen IoT in several configurations which may contain combinations of analogue input channels, analogue output channels, and relay output channels (please go to www.defineinstruments.com or contact your Zen IoT distributor for order code options).

When the Zen IoT detects a relay output module in one of the channel slots, the functions for that channel change from the standard analogue input functions to a relay output function and the 3 pin channel connector now provides a single pole double throw (SPDT) relay contact output.

See also
Additional Analogue Output Modules

Status of Analogue O/P Module

Analog Inputs

2.7.5.1 Relay Output Module

When the Zen IoT detects a relay output module in a channel slot, the functions for that channel change from the standard analogue input function to a relay output function and the 3 pin channel connector now provides a single pole double throw (SPDT) relay contact output.

Many of the registers associated with the channel are still valid however for some of them their functionality changes.

The table below shows those existing registers whose functions are changed in analogue output mode.

Name	Description	Symbol Type	Register Numbers	Memory Type
CH1 to CH16	32-bit registers that now hold a status value of 1 or 0 where 1= relay energized and 0=relay de-energized. This status value is sent to the relay output module.	S_32	(Same as input channel registers)	RAM/FLASH
IM_STATUS1 to IM_STATUS16	16-bit unsigned registers that hold the relay output module status for CH1 to CH16. (See Module Status also)	U_16	(Same as input channel registers)	RAM

Note: The information shown above is only valid when the slot for an analogue channel has a relay output module fitted. Although the registers shown above have the same register names and addresses as those shown in the "Analogue Inputs" section, their function changes when a relay output module is fitted in the channel slot.

When an analogue channel slot is fitted with a relay output module, the following registers are used to configure the data source for each relay output. Each register is a 16 bit register which holds a Modbus coil or switch register number. This allows you to associate each relay output with virtually any other discrete input or output in the Zen IoT. The table below shows the register source values for different discrete inputs and outputs and some special logical OR masks.

Discrete I/O Source Table

Name	Source No.	Description
RA	1	Relay A (on board Zen IoT).
RB	2	Relay B (on board Zen IoT).
RC	3	Relay C (on board Zen IoT).
RD	4	Relay D (on board Zen IoT).
R5	5	Relay R5 (only available if Zen IoT is fitted with optional relay module in channel 5 slot).
R6	6	Relay R6 (only available if Zen IoT is fitted with optional relay module in channel 6 slot).
R7	7	Relay R7 (only available if Zen IoT is fitted with optional relay module in channel 7 slot).
R8	8	Relay R8 (only available if Zen IoT is fitted with optional relay module in channel 8 slot).
R9	9	Relay R9 (only available if Zen IoT is fitted with optional relay module in channel 9 slot).
R10	10	Relay R10 (only available if Zen IoT is fitted with optional relay module in channel 10 slot).
R11	11	Relay R11 (only available if Zen IoTis fitted with optional relay module in channel 11 slot).
R12	12	Relay R12 (only available if Zen IoT is fitted with optional relay module in channel 12 slot).
R13	13	Relay R13 (only available if Zen IoT is fitted with optional relay module in channel 13 slot).
R14	14	Relay R14 (only available if Zen IoT is fitted with optional relay module in channel 14 slot).
R15	15	Relay R15 (only available if Zen IoT is fitted with optional relay module in channel 15 slot).
R16	16	Relay R16 (only available if Zen IoT is fitted with optional relay module in channel 16 slot).
ASP_1 to ASP_16	113 to 128	Advanced Setpoint relays 1 to 16 . (note: these bits may be used as flags but actual outputs are only available if selected I/O modules are fitted. See I/O Module Type).
-	129 to 256	Reserved for future development. These should not be used

SP_OR_MASK1	257	Logical OR mask 1 for all 16 advanced setpoint relays 1 to 16.
SP_OR_MASK2	258	Logical OR mask 2 for all 16 advanced setpoint relays 1 to 16.
SP_OR_MASK3	259	Logical OR mask 3 for all 16 advanced setpoint relays 1 to 16.
SP_OR_MASK4	260	Logical OR mask 4 for all 16 advanced setpoint relays 1 to 16.
SP_OR_MASK5	261	Logical OR mask 5 for all 16 advanced setpoint relays 1 to 16.
SP_OR_MASK6	262	Logical OR mask 6 for all 16 advanced setpoint relays 1 to 16.
SP_OR_MASK7	263	Logical OR mask 7 for all 16 advanced setpoint relays 1 to 16.
SP_OR_MASK8	264	Logical OR mask 8 for all 16 advanced setpoint relays 1 to 16.
-	265 to 32768	Reserved for future development. These should not be used
DI A	32769	Digital input A (on board Zen IoT).
DI B	32770	Digital input B (on board Zen IoT).
DI C	32771	Digital input C (on board Zen IoT).
DI_D	32772	Digital input D (on board Zen IoT).
-	32773 to 32784	Reserved for future development. These should not be used

Note: These same 16 registers are also used when analogue output modules are installed in channel slots, however they point to completely different data sources. With relay modules installed they point to discrete digital data flags. With analogue output modules installed they point to analogue registers that contain integer values. It is possible that a Zen IoT may contain a combination of each so it is important check what type of module is occupying a particular channel slot before accessing these registers. The module type can be determined by reading the module ID code.

Name	Description	Symbol Type	Register Number	Memory Type
SLOT1_DATA_SOURCE	16-bit register that points to the data source for CH1 relay output module. **NOTE**: Only discrete Modbus registers numbers can be used as a data source with a relay output module. (See Discrete I/O Source Table.)	U_16	(same as for analogue output)	RAM/EEPROM
SLOT2_DATA_SOURCE	16-bit register that points to the data source for CH2 relay output module. **NOTE**: Only discrete Modbus registers numbers can be used as a data source with a relay output module. (See Discrete I/O Source Table.)	U_16	(same as for analogue output)	RAM/EEPROM
SLOT3_DATA_SOURCE	16-bit register that points to the data source for CH3 relay output module. **NOTE**: Only discrete Modbus registers numbers can be used as a data source with a relay output module. (See Discrete I/O Source Table.)	U_16	(same as for analogue output)	RAM/EEPROM

SLOT4_DATA_SOURCE	16-bit register that points to the data source for CH4 relay output module. **NOTE**: Only discrete Modbus registers numbers can be used as a data source with a relay output module. (See Discrete I/O Source Table.)	U_16	(same as for analogue output)	RAM/EEPROM
SLOT5_DATA_SOURCE	16-bit register that points to the data source for CH5 relay output module. **NOTE**: Only discrete Modbus registers numbers can be used as a data source with a relay output module. (See Discrete I/O Source Table.)	U_16	(same as for analogue output)	RAM/EEPROM
SLOT6_DATA_SOURCE	16-bit register that points to the data source for CH6 relay output module. **NOTE**: Only discrete Modbus registers numbers can be used as a data source with a relay output module. (See Discrete I/O Source Table.)	U_16	(same as for analogue output)	RAM/EEPROM
SLOT7_DATA_SOURCE	16-bit register that points to the data source for CH7 relay output module. **NOTE**: Only discrete Modbus registers numbers can be used as a data source with a relay output module. (See Discrete I/O Source Table.)	U_16	(same as for analogue output)	RAM/EEPROM
SLOT8_DATA_SOURCE	16-bit register that points to the data source for CH8 relay output module. **NOTE**: Only discrete Modbus registers numbers can be used as a data source with a relay output module. (See Discrete I/O Source Table.)	U_16	(same as for analogue output)	RAM/EEPROM
SLOT9_DATA_SOURCE	16-bit register that points to the data source for CH9 relay output module. **NOTE**: Only discrete Modbus registers numbers can be used as a data source with a relay output module. (See Discrete I/O Source Table.)	U_16	(same as for analogue output)	RAM/EEPROM
SLOT10_DATA_SOURCE	16-bit register that points to the data source for CH10 relay output module. **NOTE**: Only discrete Modbus registers numbers can be used as a data source with a relay output module. (See Discrete I/O Source Table.)	U_16	(same as for analogue output)	RAM/EEPROM
SLOT11_DATA_SOURCE	16-bit register that points to the data source for CH11 relay output module. **NOTE**: Only discrete Modbus registers numbers can be used as a data source with a relay output module. (See Discrete I/O Source Table.)	U_16	(same as for analogue output)	RAM/EEPROM
SLOT12_DATA_SOURCE	16-bit register that points to the data source for CH12 relay output module. **NOTE**: Only discrete Modbus registers numbers can be used as a data source with a relay output module. (See Discrete I/O Source Table.)	U_16	(same as for analogue output)	RAM/EEPROM
SLOT13_DATA_SOURCE	16-bit register that points to the data source for CH13 relay output module. **NOTE**: Only discrete Modbus registers numbers can be used as a data source with a relay output module. (See Discrete I/O Source Table.)	U_16	(same as for analogue output)	RAM/EEPROM
SLOT14_DATA_SOURCE	16-bit register that points to the data source for CH14 relay output module. **NOTE**: Only discrete Modbus registers numbers can be used as a data source with a relay output module. (See Discrete I/O Source Table.)	U_16	(same as for analogue output)	RAM/EEPROM
SLOT15_DATA_SOURCE	16-bit register that points to the data source for CH15 relay output module. **NOTE**: Only discrete Modbus registers numbers can be used as a data source with a relay output module. (See Discrete I/O Source Table.)	U_16	(same as for analogue output)	RAM/EEPROM
SLOT16_DATA_SOURCE	16-bit register that points to the data source for CH16 relay output module. **NOTE**: Only discrete Modbus registers numbers can be used as a data source with a relay output module. (See Discrete I/O Source Table.)	U_16	(same as for analogue output)	RAM/EEPROM

See also
Additional Analogue Output Modules

Status of Analogue O/P Module

Analogue Inputs

2.8 Linearization

Linearization registers contain the input and output data for the 32 input and 32 output points of each linearization table, as well as the table's date and serial number.

Name	Description	Symbol Type	Register Number	Memory Type
TABLE1_DATE	16-bit register. Date (Year/Week) when linearization table 1 was last modified (range 0000 - 9952).	U_16	4249	EEPROM
TABLE1_SERIAL_NO	16-bit register. Serial number of linearization table 1 (range 0-65535).	U_16	4253	EEPROM
TABLE2_DATE	16-bit register. Date (Year/Week) when linearization table 2 was last modified (range 0000 - 9952).	U_16	4250	EEPROM
TABLE2_SERIAL_NO	16-bit register. Serial number of linearization table 2 (range 0-65535).	U_16	4254	EEPROM
TABLE3_DATE	16-bit register. Date (Year/Week) when linearization table 3 was last modified (range 0000 - 9952).	U_16	4251	EEPROM
TABLE3_SERIAL_NO	16-bit register. Serial number of linearization table 3 (range 0-65535).	U_16	4255	EEPROM
TABLE4_DATE	16-bit register. Date (Year/Week) when linearization table 4 was last modified (range 0000 - 9952).	U_16	4252	EEPROM
TABLE4_SERIAL_NO	16-bit register. Serial number of linearization table 4 (range 0-65535).	U_16	4256	EEPROM

See Also
Linearization Table 1

Linearization Table 2

Linearization Table 3

Linearization Table 4

2.8.1 Linearization Table 1

Name	Description	Symbol Type	Register Number	Memory Type
TABLE1_INPUT1	24-bit register. Input point 1, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2049	RAM/EEPROM
TABLE1_INPUT10	24-bit register. Input point 10, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2067	RAM/EEPROM
TABLE1_INPUT11	24-bit register. Input point 11, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2069	RAM/EEPROM
TABLE1_INPUT12	24-bit register. Input point 12, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2071	RAM/EEPROM
TABLE1_INPUT13	24-bit register. Input point 13, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2073	RAM/EEPROM

TABLE1_INPUT14	24-bit register. Input point 14, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2075	RAM/EEPROM
TABLE1_INPUT15	24-bit register. Input point 15, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2077	RAM/EEPROM
TABLE1_INPUT16	24-bit register. Input point 16, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2079	RAM/EEPROM
TABLE1_INPUT17	24-bit register. Input point 17, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2081	RAM/EEPROM
TABLE1_INPUT18	24-bit register. Input point 18, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2083	RAM/EEPROM
TABLE1_INPUT19	24-bit register. Input point 19, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2085	RAM/EEPROM
TABLE1_INPUT2	24-bit register. Input point 2, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2051	RAM/EEPROM
TABLE1_INPUT20	24-bit register. Input point 20, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2087	RAM/EEPROM
TABLE1_INPUT21	24-bit register. Input point 21, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2089	RAM/EEPROM
TABLE1_INPUT22	24-bit register. Input point 22, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2091	RAM/EEPROM
TABLE1_INPUT23	24-bit register. Input point 23, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2093	RAM/EEPROM
TABLE1_INPUT24	24-bit register. Input point 24, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2095	RAM/EEPROM
TABLE1_INPUT25	24-bit register. Input point 25, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2097	RAM/EEPROM
TABLE1_INPUT26	24-bit register. Input point 26, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2099	RAM/EEPROM
TABLE1_INPUT27	24-bit register. Input point 27, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2101	RAM/EEPROM
TABLE1_INPUT28	24-bit register. Input point 28, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2103	RAM/EEPROM
TABLE1_INPUT29	24-bit register. Input point 29, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2105	RAM/EEPROM
TABLE1_INPUT3	24-bit register. Input point 3, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2053	RAM/EEPROM
TABLE1_INPUT30	24-bit register. Input point 30, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2107	RAM/EEPROM
TABLE1_INPUT31	24-bit register. Input point 31, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2109	RAM/EEPROM
TABLE1_INPUT32	24-bit register. Input point 32, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2111	RAM/EEPROM
TABLE1_INPUT4	24-bit register. Input point 4, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2055	RAM/EEPROM
TABLE1_INPUT5	24-bit register. Input point 5, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2057	RAM/EEPROM
TABLE1_INPUT6	24-bit register. Input point 6, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2059	RAM/EEPROM
TABLE1_INPUT7	24-bit register. Input point 7, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2061	RAM/EEPROM
TABLE1_INPUT8	24-bit register. Input point 8, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2063	RAM/EEPROM
TABLE1_INPUT9	24-bit register. Input point 9, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2065	RAM/EEPROM

TABLE1_OUTPUT1	24-bit register. Output point 1, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2113	RAM/EEPROM
TABLE1_OUTPUT10	24-bit register. Output point 10, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2131	RAM/EEPROM
TABLE1_OUTPUT11	24-bit register. Output point 11, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2133	RAM/EEPROM
TABLE1_OUTPUT12	24-bit register. Output point 12, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2135	RAM/EEPROM
TABLE1_OUTPUT13	24-bit register. Output point 13, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2137	RAM/EEPROM
TABLE1_OUTPUT14	24-bit register. Output point 14, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2139	RAM/EEPROM
TABLE1_OUTPUT15	24-bit register. Output point 15, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2141	RAM/EEPROM
TABLE1_OUTPUT16	24-bit register. Output point 16, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2143	RAM/EEPROM
TABLE1_OUTPUT17	24-bit register. Output point 17, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2145	RAM/EEPROM
TABLE1_OUTPUT18	24-bit register. Output point 18, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2147	RAM/EEPROM
TABLE1_OUTPUT19	24-bit register. Output point 19, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2149	RAM/EEPROM
TABLE1_OUTPUT2	24-bit register. Output point 2, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2115	RAM/EEPROM
TABLE1_OUTPUT20	24-bit register. Output point 20, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2151	RAM/EEPROM
TABLE1_OUTPUT21	24-bit register. Output point 21, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2153	RAM/EEPROM
TABLE1_OUTPUT22	24-bit register. Output point 22, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2155	RAM/EEPROM
TABLE1_OUTPUT23	24-bit register. Output point 23, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2157	RAM/EEPROM
TABLE1_OUTPUT24	24-bit register. Output point 24, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2159	RAM/EEPROM
TABLE1_OUTPUT25	24-bit register. Output point 25, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2161	RAM/EEPROM
TABLE1_OUTPUT26	24-bit register. Output point 26, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2163	RAM/EEPROM
TABLE1_OUTPUT27	24-bit register. Output point 27, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2165	RAM/EEPROM
TABLE1_OUTPUT28	24-bit register. Output point 28, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2167	RAM/EEPROM
TABLE1_OUTPUT29	24-bit register. Output point 29, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2169	RAM/EEPROM
TABLE1_OUTPUT3	24-bit register. Output point 3, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2117	RAM/EEPROM
TABLE1_OUTPUT30	24-bit register. Output point 30, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2171	RAM/EEPROM
TABLE1_OUTPUT31	24-bit register. Output point 31, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2173	RAM/EEPROM

TABLE1_OUTPUT32	24-bit register. Output point 32, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2175	RAM/EEPROM
TABLE1_OUTPUT4	24-bit register. Output point 4, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2119	RAM/EEPROM
TABLE1_OUTPUT5	24-bit register. Output point 5, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2121	RAM/EEPROM
TABLE1_OUTPUT6	24-bit register. Output point 6, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2123	RAM/EEPROM
TABLE1_OUTPUT7	24-bit register. Output point 7, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2125	RAM/EEPROM
TABLE1_OUTPUT8	24-bit register. Output point 8, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2127	RAM/EEPROM
TABLE1_OUTPUT9	24-bit register. Output point 9, 32-point linearization table 1 (range -8388607 - +8388607).	S_24	2129	RAM/EEPROM

2.8.2 Linearization Table 2

Name	Description	Symbol Type	Register Number	Memory Type
TABLE2_INPUT1	24-bit register. Input point 1, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2177	RAM/EEPROM
TABLE2_INPUT10	24-bit register. Input point 10, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2195	RAM/EEPROM
TABLE2_INPUT11	24-bit register. Input point 11, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2197	RAM/EEPROM
TABLE2_INPUT12	24-bit register. Input point 12, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2199	RAM/EEPROM
TABLE2_INPUT13	24-bit register. Input point 13, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2201	RAM/EEPROM
TABLE2_INPUT14	24-bit register. Input point 14, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2203	RAM/EEPROM
TABLE2_INPUT15	24-bit register. Input point 15, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2205	RAM/EEPROM
TABLE2_INPUT16	24-bit register. Input point 16, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2207	RAM/EEPROM
TABLE2_INPUT17	24-bit register. Input point 17, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2209	RAM/EEPROM
TABLE2_INPUT18	24-bit register. Input point 18, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2211	RAM/EEPROM
TABLE2_INPUT19	24-bit register. Input point 19, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2213	RAM/EEPROM

TABLE2_INPUT2	24-bit register. Input point 2, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2179	RAM/EEPROM
TABLE2_INPUT20	24-bit register. Input point 20, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2215	RAM/EEPROM
TABLE2_INPUT21	24-bit register. Input point 21, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2217	RAM/EEPROM
TABLE2_INPUT22	24-bit register. Input point 22, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2219	RAM/EEPROM
TABLE2_INPUT23	24-bit register. Input point 23, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2221	RAM/EEPROM
TABLE2_INPUT24	24-bit register. Input point 24, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2223	RAM/EEPROM
TABLE2_INPUT25	24-bit register. Input point 25, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2225	RAM/EEPROM
TABLE2_INPUT26	24-bit register. Input point 26, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2227	RAM/EEPROM
TABLE2_INPUT27	24-bit register. Input point 27, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2229	RAM/EEPROM
TABLE2_INPUT28	24-bit register. Input point 28, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2231	RAM/EEPROM
TABLE2_INPUT29	24-bit register. Input point 29, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2233	RAM/EEPROM
TABLE2_INPUT3	24-bit register. Input point 3, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2181	RAM/EEPROM
TABLE2_INPUT30	24-bit register. Input point 30, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2235	RAM/EEPROM
TABLE2_INPUT31	24-bit register. Input point 31, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2237	RAM/EEPROM
TABLE2_INPUT32	24-bit register. Input point 32, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2239	RAM/EEPROM
TABLE2_INPUT4	24-bit register. Input point 4, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2183	RAM/EEPROM
TABLE2_INPUT5	24-bit register. Input point 5, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2185	RAM/EEPROM
TABLE2_INPUT6	24-bit register. Input point 6, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2187	RAM/EEPROM
TABLE2_INPUT7	24-bit register. Input point 7, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2189	RAM/EEPROM
TABLE2_INPUT8	24-bit register. Input point 8, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2191	RAM/EEPROM
TABLE2_INPUT9	24-bit register. Input point 9, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2193	RAM/EEPROM
TABLE2_OUTPUT1	24-bit register. Output point 1, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2241	RAM/EEPROM
TABLE2_OUTPUT10	24-bit register. Output point 10, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2259	RAM/EEPROM
TABLE2_OUTPUT11	24-bit register. Output point 11, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2261	RAM/EEPROM
TABLE2_OUTPUT12	24-bit register. Output point 12, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2263	RAM/EEPROM
TABLE2_OUTPUT13	24-bit register. Output point 13, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2265	RAM/EEPROM
TABLE2_OUTPUT14	24-bit register. Output point 14, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2267	RAM/EEPROM

TABLE2_OUTPUT15	24-bit register. Output point 15, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2269	RAM/EEPROM
TABLE2_OUTPUT16	24-bit register. Output point 16, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2271	RAM/EEPROM
TABLE2_OUTPUT17	24-bit register. Output point 17, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2273	RAM/EEPROM
TABLE2_OUTPUT18	24-bit register. Output point 18, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2275	RAM/EEPROM
TABLE2_OUTPUT19	24-bit register. Output point 19, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2277	RAM/EEPROM
TABLE2_OUTPUT2	24-bit register. Output point 2, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2243	RAM/EEPROM
TABLE2_OUTPUT20	24-bit register. Output point 20, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2279	RAM/EEPROM
TABLE2_OUTPUT21	24-bit register. Output point 21, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2281	RAM/EEPROM
TABLE2_OUTPUT22	24-bit register. Output point 22, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2283	RAM/EEPROM
TABLE2_OUTPUT23	24-bit register. Output point 23, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2285	RAM/EEPROM
TABLE2_OUTPUT24	24-bit register. Output point 24, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2287	RAM/EEPROM
TABLE2_OUTPUT25	24-bit register. Output point 25, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2289	RAM/EEPROM
TABLE2_OUTPUT26	24-bit register. Output point 26, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2291	RAM/EEPROM
TABLE2_OUTPUT27	24-bit register. Output point 27, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2293	RAM/EEPROM
TABLE2_OUTPUT28	24-bit register. Output point 28, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2295	RAM/EEPROM
TABLE2_OUTPUT29	24-bit register. Output point 29, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2297	RAM/EEPROM
TABLE2_OUTPUT3	24-bit register. Output point 3, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2245	RAM/EEPROM
TABLE2_OUTPUT30	24-bit register. Output point 30, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2299	RAM/EEPROM
TABLE2_OUTPUT31	24-bit register. Output point 31, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2301	RAM/EEPROM
TABLE2_OUTPUT32	24-bit register. Output point 32, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2303	RAM/EEPROM
TABLE2_OUTPUT4	24-bit register. Output point 4, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2247	RAM/EEPROM
TABLE2_OUTPUT5	24-bit register. Output point 5, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2249	RAM/EEPROM
TABLE2_OUTPUT6	24-bit register. Output point 6, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2251	RAM/EEPROM
TABLE2_OUTPUT7	24-bit register. Output point 7, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2253	RAM/EEPROM
TABLE2_OUTPUT8	24-bit register. Output point 8, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2255	RAM/EEPROM
TABLE2_OUTPUT9	24-bit register. Output point 9, 32-point linearization table 2 (range -8388607 - +8388607).	S_24	2257	RAM/EEPROM

2.8.3 Linearization Table 3

Name	Description	Symbol Type	Register Number	Memory Type
TABLE3_INPUT1	24-bit register. Input point 1, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2305	RAM/EEPROM

TABLE3_INPUT11	24-bit register. Input point 11, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2325	RAM/EEPROM
TABLE3_INPUT12	24-bit register. Input point 12, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2327	RAM/EEPROM
TABLE3_INPUT13	24-bit register. Input point 13, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2329	RAM/EEPROM
TABLE3_INPUT14	24-bit register. Input point 14, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2331	RAM/EEPROM
TABLE3_INPUT15	24-bit register. Input point 15, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2333	RAM/EEPROM
TABLE3_INPUT16	24-bit register. Input point 16, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2335	RAM/EEPROM
TABLE3_INPUT17	24-bit register. Input point 17, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2337	RAM/EEPROM
TABLE3_INPUT18	24-bit register. Input point 18, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2339	RAM/EEPROM
TABLE3_INPUT19	24-bit register. Input point 19, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2341	RAM/EEPROM
TABLE3_INPUT2	24-bit register. Input point 2, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2307	RAM/EEPROM
TABLE3_INPUT20	24-bit register. Input point 20, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2343	RAM/EEPROM
TABLE3_INPUT21	24-bit register. Input point 21, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2345	RAM/EEPROM
TABLE3_INPUT22	24-bit register. Input point 22, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2347	RAM/EEPROM
TABLE3_INPUT23	24-bit register. Input point 23, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2349	RAM/EEPROM
TABLE3_INPUT24	24-bit register. Input point 24, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2351	RAM/EEPROM
TABLE3_INPUT25	24-bit register. Input point 25, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2353	RAM/EEPROM
TABLE3_INPUT26	24-bit register. Input point 26, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2355	RAM/EEPROM
TABLE3_INPUT27	24-bit register. Input point 27, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2357	RAM/EEPROM
TABLE3_INPUT28	24-bit register. Input point 28, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2359	RAM/EEPROM
TABLE3_INPUT29	24-bit register. Input point 29, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2361	RAM/EEPROM
TABLE3_INPUT3	24-bit register. Input point 3, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2309	RAM/EEPROM
TABLE3_INPUT30	24-bit register. Input point 30, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2363	RAM/EEPROM
TABLE3_INPUT31	24-bit register. Input point 31, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2365	RAM/EEPROM
TABLE3_INPUT32	24-bit register. Input point 32, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2367	RAM/EEPROM
TABLE3_INPUT4	24-bit register. Input point 4, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2311	RAM/EEPROM

TABLE3_INPUT5	24-bit register. Input point 5, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2313	RAM/EEPROM
TABLE3_INPUT6	24-bit register. Input point 6, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2315	RAM/EEPROM
TABLE3_INPUT7	24-bit register. Input point 7, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2317	RAM/EEPROM
TABLE3_INPUT8	24-bit register. Input point 8, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2319	RAM/EEPROM
TABLE3_INPUT9	24-bit register. Input point 9, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2321	RAM/EEPROM
TABLE3_OUTPUT1	24-bit register. Output point 1, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2369	RAM/EEPROM
TABLE3_OUTPUT10	24-bit register. Output point 10, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2387	RAM/EEPROM
TABLE3_OUTPUT11	24-bit register. Output point 11, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2389	RAM/EEPROM
TABLE3_OUTPUT12	24-bit register. Output point 12, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2391	RAM/EEPROM
TABLE3_OUTPUT13	24-bit register. Output point 13, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2393	RAM/EEPROM
TABLE3_OUTPUT14	24-bit register. Output point 14, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2395	RAM/EEPROM
TABLE3_OUTPUT15	24-bit register. Output point 15, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2397	RAM/EEPROM
TABLE3_OUTPUT16	24-bit register. Output point 16, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2399	RAM/EEPROM
TABLE3_OUTPUT17	24-bit register. Output point 17, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2401	RAM/EEPROM
TABLE3_OUTPUT18	24-bit register. Output point 18, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2403	RAM/EEPROM
TABLE3_OUTPUT19	24-bit register. Output point 19, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2405	RAM/EEPROM
TABLE3_OUTPUT2	24-bit register. Output point 2, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2371	RAM/EEPROM
TABLE3_OUTPUT20	24-bit register. Output point 20, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2407	RAM/EEPROM
TABLE3_OUTPUT21	24-bit register. Output point 21, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2409	RAM/EEPROM

TABLE3_OUTPUT22	24-bit register. Output point 22, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2411	RAM/EEPROM
TABLE3_OUTPUT23	24-bit register. Output point 23, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2413	RAM/EEPROM
TABLE3_OUTPUT24	24-bit register. Output point 24, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2415	RAM/EEPROM
TABLE3_OUTPUT25	24-bit register. Output point 25, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2417	RAM/EEPROM
TABLE3_OUTPUT26	24-bit register. Output point 26, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2419	RAM/EEPROM
TABLE3_OUTPUT27	24-bit register. Output point 27, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2421	RAM/EEPROM
TABLE3_OUTPUT28	24-bit register. Output point 28, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2423	RAM/EEPROM
TABLE3_OUTPUT29	24-bit register. Output point 29, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2425	RAM/EEPROM
TABLE3_OUTPUT3	24-bit register. Output point 3, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2373	RAM/EEPROM
TABLE3_OUTPUT30	24-bit register. Output point 30, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2427	RAM/EEPROM
TABLE3_OUTPUT31	24-bit register. Output point 31, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2429	RAM/EEPROM
TABLE3_OUTPUT32	24-bit register. Output point 32, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2431	RAM/EEPROM
TABLE3_OUTPUT4	24-bit register. Output point 4, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2375	RAM/EEPROM
TABLE3_OUTPUT5	24-bit register. Output point 5, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2377	RAM/EEPROM
TABLE3_OUTPUT6	24-bit register. Output point 6, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2379	RAM/EEPROM
TABLE3_OUTPUT7	24-bit register. Output point 7, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2381	RAM/EEPROM
TABLE3_OUTPUT8	24-bit register. Output point 8, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2383	RAM/EEPROM
TABLE3_OUTPUT9	24-bit register. Output point 9, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2385	RAM/EEPROM

2.8.4 Linearization Table 4

Name	Description	Symbol Type	Register Number	Memory Type
TABLE4_INPUT1	24-bit register. Input point 1, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2433	RAM/EEPROM
TABLE4_INPUT10	24-bit register. Input point 10, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2451	RAM/EEPROM
TABLE4_INPUT11	24-bit register. Input point 11, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2453	RAM/EEPROM
TABLE4_INPUT12	24-bit register. Input point 12, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2455	RAM/EEPROM
TABLE4_INPUT13	24-bit register. Input point 13, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2457	RAM/EEPROM
TABLE4_INPUT14	24-bit register. Input point 14, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2459	RAM/EEPROM
TABLE4_INPUT15	24-bit register. Input point 15, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2461	RAM/EEPROM
TABLE4_INPUT16	24-bit register. Input point 16, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2463	RAM/EEPROM
TABLE4_INPUT17	24-bit register. Input point 17, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2465	RAM/EEPROM

TABLE4_INPUT2	24-bit register. Input point 2, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2435	RAM/EEPROM
TABLE4_INPUT20	24-bit register. Input point 20, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2471	RAM/EEPROM
TABLE4_INPUT21	24-bit register. Input point 21, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2473	RAM/EEPROM
TABLE4_INPUT22	24-bit register. Input point 22, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2475	RAM/EEPROM
TABLE4_INPUT23	24-bit register. Input point 23, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2477	RAM/EEPROM
TABLE4_INPUT24	24-bit register. Input point 24, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2479	RAM/EEPROM
TABLE4_INPUT25	24-bit register. Input point 25, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2481	RAM/EEPROM
TABLE4_INPUT26	24-bit register. Input point 26, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2483	RAM/EEPROM
TABLE4_INPUT27	24-bit register. Input point 27, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2485	RAM/EEPROM
TABLE4_INPUT28	24-bit register. Input point 28, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2487	RAM/EEPROM
TABLE4_INPUT29	24-bit register. Input point 29, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2489	RAM/EEPROM
TABLE4_INPUT3	24-bit register. Input point 3, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2437	RAM/EEPROM
TABLE4_INPUT30	24-bit register. Input point 30, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2491	RAM/EEPROM
TABLE4_INPUT31	24-bit register. Input point 31, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2493	RAM/EEPROM
TABLE4_INPUT32	24-bit register. Input point 32, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2495	RAM/EEPROM
TABLE4_INPUT4	24-bit register. Input point 4, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2439	RAM/EEPROM
TABLE4_INPUT5	24-bit register. Input point 5, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2441	RAM/EEPROM
TABLE4_INPUT6	24-bit register. Input point 6, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2443	RAM/EEPROM
TABLE4_INPUT7	24-bit register. Input point 7, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2445	RAM/EEPROM
TABLE4_INPUT8	24-bit register. Input point 8, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2447	RAM/EEPROM
TABLE4_INPUT9	24-bit register. Input point 9, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2449	RAM/EEPROM
TABLE4_OUTPUT1	24-bit register. Output point 1, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2497	RAM/EEPROM
TABLE4_OUTPUT10	24-bit register. Output point 10, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2515	RAM/EEPROM
TABLE4_OUTPUT11	24-bit register. Output point 11, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2517	RAM/EEPROM
TABLE4_OUTPUT12	24-bit register. Output point 12, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2519	RAM/EEPROM
TABLE4_OUTPUT13	24-bit register. Output point 13, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2521	RAM/EEPROM
TABLE4_OUTPUT14	24-bit register. Output point 14, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2523	RAM/EEPROM

TABLE4_OUTPUT15	24-bit register. Output point 15, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2525	RAM/EEPROM
TABLE4_OUTPUT16	24-bit register. Output point 16, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2527	RAM/EEPROM
TABLE4_OUTPUT17	24-bit register. Output point 17, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2529	RAM/EEPROM
TABLE4_OUTPUT18	24-bit register. Output point 18, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2531	RAM/EEPROM
TABLE4_OUTPUT19	24-bit register. Output point 19, 32-point linearization table 3 (range -8388607 - +8388607).	S_24	2533	RAM/EEPROM
TABLE4_OUTPUT2	24-bit register. Output point 2, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2499	RAM/EEPROM
TABLE4_OUTPUT20	24-bit register. Output point 20, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2535	RAM/EEPROM
TABLE4_OUTPUT21	24-bit register. Output point 21, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2537	RAM/EEPROM
TABLE4_OUTPUT22	24-bit register. Output point 22, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2539	RAM/EEPROM
TABLE4_OUTPUT23	24-bit register. Output point 23, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2541	RAM/EEPROM
TABLE4_OUTPUT24	24-bit register. Output point 24, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2543	RAM/EEPROM
TABLE4_OUTPUT25	24-bit register. Output point 25, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2545	RAM/EEPROM
TABLE4_OUTPUT26	24-bit register. Output point 26, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2547	RAM/EEPROM
TABLE4_OUTPUT27	24-bit register. Output point 27, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2549	RAM/EEPROM
TABLE4_OUTPUT28	24-bit register. Output point 28, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2551	RAM/EEPROM
TABLE4_OUTPUT29	24-bit register. Output point 29, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2553	RAM/EEPROM
TABLE4_OUTPUT3	24-bit register. Output point 3, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2501	RAM/EEPROM
TABLE4_OUTPUT30	24-bit register. Output point 30, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2555	RAM/EEPROM
TABLE4_OUTPUT31	24-bit register. Output point 31, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2557	RAM/EEPROM
TABLE4_OUTPUT32	24-bit register. Output point 32, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2559	RAM/EEPROM
TABLE4_OUTPUT4	24-bit register. Output point 4, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2503	RAM/EEPROM
TABLE4_OUTPUT5	24-bit register. Output point 5, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2505	RAM/EEPROM
TABLE4_OUTPUT6	24-bit register. Output point 6, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2507	RAM/EEPROM
TABLE4_OUTPUT7	24-bit register. Output point 7, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2509	RAM/EEPROM
TABLE4_OUTPUT8	24-bit register. Output point 8, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2511	RAM/EEPROM
TABLE4_OUTPUT9	24-bit register. Output point 9, 32-point linearization table 4 (range -8388607 - +8388607).	S_24	2513	RAM/EEPROM

2.9 MicroScan

The Zen IoT controller has been designed to work with MicroScan, a Windows based Supervisory Control and Data Acquisition software product produced by Intech Instruments Ltd. To allow MicroScan to work efficiently, a number of registers have been reserved in the Zen IoT memory map exclusively for MicroScan use. These are described in the following sections.

2.9.1 16-bit Scratchpad Memory

The Zen IoT controller has 256 x 16 bit non-volatile memory locations reserved exclusively for the use of the MicroScan SCADA system. It is not recommended that these registers be used for any other purposes.

Name	Description	Symbol Type	Register Number	Memory Type
MICROSCAN_MEMORY_1	16 bit signed register reserved for MicroScan use.	S_16	7169	EEPROM
MICROSCAN_MEMORY_2 to MICROSCAN_MEMORY_255	16 bit signed registers reserved for MicroScan use.	S_16	7170 to 7423	EEPROM
MICROSCAN_MEMORY_256	16 bit signed register reserved for MicroScan use.	S_16	7424	EEPROM

NOTE: All of these registers use EEPROM memory. (See EEPROM write limitations)

See Also
Station Name

Macro Name

Intech Scratchpad Text

2.9.2 Intech Scratchpad Text

The Zen IoT controller has two special 30 character text string registers reserved exclusively for the use of the MicroScan SCADA system. It is not recommended that these registers be used for any other purposes. These strings are stored in non-volatile EEPROM memory which has certain write restrictions (see EEPROM write restrictions)

Name	Description	Symbol Type	Register Number	Memory Type
MICROSCAN_TEXT_1	Non-volatile 30 character text string which is reserved for MicroScan use.	L_30	16827	EEPROM
MICROSCAN_TEXT_2	Non-volatile 30 character text string which is reserved for MicroScan use.	L_30	16829	EEPROM

NOTE: All of these registers use EEPROM memory. (See EEPROM write limitations)

See Also
Station Name

Macro Name

16-bit Scratchpad Memory

2.10 Output Controllers

The Zen IoT provides various relay output options, depending on the configuration being used. It has 3 solid state relay outputs and 1 SPDT relay output.

Solid State Relays
The SSR outputs (Form A) are rated at 30V d.c. and can handle up to 400mA. Relays 2 & 3 share the same common while relay 1 has its own common which is isolated from relays 2 and 3. All SSR outputs are normally open contacts.

Mechanical Latching Relay
Relay 4 is a single pole double throw relay (Form C) which is rated at 1A 30V DC which has a latching function which allows it to hold it's state even when the power is removed from the Zen IoT controller.

Relay B RX Timeout
If the Zen IoT on board relay's A & B are not configured as controller outputs, then relay B has an additional receiver time out feature. When enabled, this feature turns on relay B at power up and holds relay B on until it detects a specified period of inactivity on various serial ports. It then turns off relay B and keeps it off until serial activity resumes or the Zen IoT is re-powered. This feature is enabled by setting the associated COMS_Timeout register for each serial port to a value > 0.

Note: Relays B is configured to function as part of a control loop then the RX Timeout feature will be disabled.

See also
Controller Setpoints

Controller Mode Registers

Controller Cooling Differential

Controller Heating Differential

Controller Deadband

Output Masks

Internal Digital Outputs

2.10.1 Controller Mode Registers

The Zen IoT has up to 16 output controllers to control relay functions for heating, cooling or manual on/off control. The functionality of each controller is defined by one of the controller mode registers which contains various flags governing its operation. There is also a global mode control register which overrides specific functions of all controller mode registers (see

The controller mode registers are 16 bit unsigned registers with bit functions as shown below;

Bit	Function	SA/DA Mode	Value=0	Value=1
b0	Enable	Both	Controller Disabled.	Controller Enabled.
b1	Manual Override	Both	Auto Mode.	Manual Mode. Output State Specified by Bit 2 (SA) or bits 5,6 (DA).
b2	Manual State	Single Action	Output Off.	Output On.
b3	Reverse Action	Both	Cooling Action (Relay is on when cooling is required i.e when temp is above setpoint).	Heating Action (Relay is on when heating required i.e when temp is below setpoint).
b4	Heat Cool Mode	Dual Action	Single action control with either one of Heating or Cooling control specified by Reverse Action Bit.	Dual Action mode with both Heat and Cool relays working.
b5	Manual Heat On	Dual Action	Heat Relay Off when manual override set.	Heat Relay On when manual override set. (Only one of bit 5 and bit 6 should be set at one time).
b6	Manual Cool On	Dual Action	Cool Relay Off when manual override set.	Cool Relay On when manual override set. (Only one of bit 5 and bit 6 should be set at one time)
b7	Scaled Integer	Dual Action	Settings are float values (2 floats per message).	Settings are DA Scaled Integers (4 words per message).
b8 - b15	Undefined		Should always to be set to 0.	

If a controller is active and then deactivated by clearing the controllers enable bit, the controllers relay state will be left in the last remaining state. It is up to the software to turn the relay to the desired state.

Controller Modes
Single Action/Dual Action.
The Single Action/Dual Action mode is specified in the Station's Advanced dialog box. Details of relay allocations are specified in the manual supplied with the station. In the Single Action Mode the behavior of the relays is defined by the control action chosen. In the Dual Action mode, the functions of the relays are fixed, but will only operate according to the control action chosen.

Dual Action Mode Relays

Relay	Function	Heating	Cooling	Heat/Cool
Relay 1	Heat Action	Active	Off	Active
Relay 2	Cool Action	Off	Active	Active

This is chosen so the user can switch between Heat/Cool and Heat only or Cool only and have the relay numbers stay the same.

A16
For the Single Action control mode there are 16 controllers.
For the Dual Action mode, there are 8 controllers. With the setup for the extra parameters needed for the controller coming from the controller index+8.

Name	Description	Symbol Type	Register Number	Memory Type
GLOBAL_CONTROLLER_MODE	16-bit register which acts as a global mode control for all controller registers. Setting flags in this register globally sets single or dual action.	U_16	4354	RAM/EEPROM
CONTROLLER1_MODE	16-bit register which defines the operation of controller 1.	U_16	4355	RAM/EEPROM
CONTROLLER2_MODE	16-bit register which defines the operation of controller 2.	U_16	4356	RAM/EEPROM
CONTROLLER3_MODE	16-bit register which defines the operation of controller 3.	U_16	4357	RAM/EEPROM
CONTROLLER4_MODE	16-bit register which defines the operation of controller 4.	U_16	4358	RAM/EEPROM
CONTROLLER5_MODE	16-bit register which defines the operation of controller 5.	U_16	4359	RAM/EEPROM
CONTROLLER6_MODE	16-bit register which defines the operation of controller 6.	U_16	4360	RAM/EEPROM
CONTROLLER7_MODE	16-bit register which defines the operation of controller 7.	U_16	4361	RAM/EEPROM
CONTROLLER8_MODE	16-bit register which defines the operation of controller 8.	U_16	4362	RAM/EEPROM
CONTROLLER9_MODE	16-bit register which defines the operation of controller 9.	U_16	4363	RAM/EEPROM
CONTROLLER10_MODE	16-bit register which defines the operation of controller 10.	U_16	4364	RAM/EEPROM
CONTROLLER11_MODE	16-bit register which defines the operation of controller 11.	U_16	4365	RAM/EEPROM
CONTROLLER12_MODE	16-bit register which defines the operation of controller 12.	U_16	4366	RAM/EEPROM
CONTROLLER13_MODE	16-bit register which defines the operation of controller 13.	U_16	4367	RAM/EEPROM
CONTROLLER14_MODE	16-bit register which defines the operation of controller 14.	U_16	4368	RAM/EEPROM
CONTROLLER15_MODE	16-bit register which defines the operation of controller 15.	U_16	4369	RAM/EEPROM
CONTROLLER16_MODE	16-bit register which defines the operation of controller 16.	U_16	4370	RAM/EEPROM

See also

Controller Setpoints

Controller Cooling Differential

Controller Heating Differential

Controller Deadband

Controller Outputs

Output Masks

Internal Digital Outputs

2.10.2 Controller Setpoints

For more info on the operation of controller setpoints contact Define Instruments Ltd..

Name	Description	Symbol Type	Register Number	Memory Type
CONTROLLER1_SP	16-bit register that holds the setpoint value for controller 1.	S_16	4387	RAM/EEPROM
CONTROLLER2_SP	16-bit register that holds the setpoint value for controller 2.	S_16	4388	RAM/EEPROM
CONTROLLER3_SP	16-bit register that holds the setpoint value for controller 3.	S_16	4389	RAM/EEPROM
CONTROLLER4_SP	16-bit register that holds the setpoint value for controller 4.	S_16	4390	RAM/EEPROM
CONTROLLER5_SP	16-bit register that holds the setpoint value for controller 5.	S_16	4391	RAM/EEPROM
CONTROLLER6_SP	16-bit register that holds the setpoint value for controller 6.	S_16	4392	RAM/EEPROM
CONTROLLER7_SP	16-bit register that holds the setpoint value for controller 7.	S_16	4393	RAM/EEPROM
CONTROLLER8_SP	16-bit register that holds the setpoint value for controller 8.	S_16	4394	RAM/EEPROM
CONTROLLER9_SP	16-bit register that holds the setpoint value for controller 9.	S_16	4395	RAM/EEPROM
CONTROLLER10_SP	16-bit register that holds the setpoint value for controller 10.	S_16	4396	RAM/EEPROM
CONTROLLER11_SP	16-bit register that holds the setpoint value for controller 11.	S_16	4397	RAM/EEPROM
CONTROLLER12_SP	16-bit register that holds the setpoint value for controller 12.	S_16	4398	RAM/EEPROM
CONTROLLER13_SP	16-bit register that holds the setpoint value for controller 13.	S_16	4399	RAM/EEPROM
CONTROLLER14_SP	16-bit register that holds the setpoint value for controller 14.	S_16	4400	RAM/EEPROM
CONTROLLER15_SP	16-bit register that holds the setpoint value for controller 15.	S_16	4401	RAM/EEPROM
CONTROLLER16_SP	16-bit register that holds the setpoint value for controller 16.	S_16	4402	RAM/EEPROM

See also

Controller Mode

Controller Cooling Differential

Controller Heating Differential

Controller Deadband

Controller Outputs

Output Masks

2.10.3 Controller Cooling Differential

For more info on the operation of controller cooling differential contact Define Instruments Ltd..

Need diagram showing operation of cooling differential.

Name	Description	Symbol Type	Register Number	Memory Type
COOL_DIFF1	16-bit register that holds the cooling differential value for controller 1.	U_16	4472	RAM/EEPROM
COOL_DIFF2	16-bit register that holds the cooling differential value for controller 2.	U_16	4473	RAM/EEPROM
COOL_DIFF3	16-bit register that holds the cooling differential value for controller 3.	U_16	4474	RAM/EEPROM
COOL_DIFF4	16-bit register that holds the cooling differential value for controller 4.	U_16	4475	RAM/EEPROM
COOL_DIFF5	16-bit register that holds the cooling differential value for controller 5.	U_16	4476	RAM/EEPROM
COOL_DIFF6	16-bit register that holds the cooling differential value for controller 6.	U_16	4477	RAM/EEPROM
COOL_DIFF7	16-bit register that holds the cooling differential value for controller 7.	U_16	4478	RAM/EEPROM
COOL_DIFF8	16-bit register that holds the cooling differential value for controller 8.	U_16	4479	RAM/EEPROM
COOL_DIFF9	16-bit register that holds the cooling differential value for controller 9.	U_16	4480	RAM/EEPROM
COOL_DIFF10	16-bit register that holds the cooling differential value for controller 10.	U_16	4481	RAM/EEPROM
COOL_DIFF11	16-bit register that holds the cooling differential value for controller 11.	U_16	4482	RAM/EEPROM
COOL_DIFF12	16-bit register that holds the cooling differential value for controller 12.	U_16	4483	RAM/EEPROM
COOL_DIFF13	16-bit register that holds the cooling differential value for controller 13.	U_16	4484	RAM/EEPROM
COOL_DIFF14	16-bit register that holds the cooling differential value for controller 14.	U_16	4485	RAM/EEPROM
COOL_DIFF15	16-bit register that holds the cooling differential value for controller 15.	U_16	4486	RAM/EEPROM
COOL_DIFF16	16-bit register that holds the cooling differential value for controller 16.	U_16	4487	RAM/EEPROM

See also

Controller Heating Differential

Controller Deadband

Controller Outputs

Controller Setpoints

Controller Mode

Output Masks

2.10.4 Controller Heating Differential

For more info on the operation of controller heating differential contact Define Instruments Ltd..

Need diagram showing operation of heating differential.

Name	Description	Symbol Type	Register Number	Memory Type
HEAT_DIFF1	16-bit register that holds the heating differential value for controller 1.	U_16	4504	RAM/EEPROM
HEAT_DIFF2	16-bit register that holds the heating differential value for controller 2.	U_16	4505	RAM/EEPROM
HEAT_DIFF3	16-bit register that holds the heating differential value for controller 3.	U_16	4506	RAM/EEPROM
HEAT_DIFF4	16-bit register that holds the heating differential value for controller 4.	U_16	4507	RAM/EEPROM
HEAT_DIFF5	16-bit register that holds the heating differential value for controller 5.	U_16	4508	RAM/EEPROM
HEAT_DIFF6	16-bit register that holds the heating differential value for controller 6.	U_16	4509	RAM/EEPROM
HEAT_DIFF7	16-bit register that holds the heating differential value for controller 7.	U_16	4510	RAM/EEPROM
HEAT_DIFF8	16-bit register that holds the heating differential value for controller 8.	U_16	4511	RAM/EEPROM
HEAT_DIFF9	16-bit register that holds the heating differential value for controller 9.	U_16	4512	RAM/EEPROM
HEAT_DIFF10	16-bit register that holds the heating differential value for controller 10.	U_16	4513	RAM/EEPROM
HEAT_DIFF11	16-bit register that holds the heating differential value for controller 11.	U_16	4514	RAM/EEPROM
HEAT_DIFF12	16-bit register that holds the heating differential value for controller 12.	U_16	4515	RAM/EEPROM
HEAT_DIFF13	16-bit register that holds the heating differential value for controller 13.	U_16	4516	RAM/EEPROM
HEAT_DIFF14	16-bit register that holds the heating differential value for controller 14.	U_16	4517	RAM/EEPROM
HEAT_DIFF15	16-bit register that holds the heating differential value for controller 15.	U_16	4518	RAM/EEPROM
HEAT_DIFF16	16-bit register that holds the heating differential value for controller 16.	U_16	4519	RAM/EEPROM

See also

Controller Cooling Differential

Controller Deadband

Controller Setpoints

Controller Mode

Controller Outputs

Output Masks

2.10.5 Controller Deadband

For more info on the operation of controller Deadband contact Define Instruments Ltd..

Need diagram showing Deadband operation.

Name	Description	Symbol Type	Register Number	Memory Type
DEADBAND1	16-bit register that holds the Deadband value for controller 1.	U_16	4536	RAM/EEPROM
DEADBAND2	16-bit register that holds the Deadband value for controller 2.	U_16	4537	RAM/EEPROM
DEADBAND3	16-bit register that holds the Deadband value for controller 3.	U_16	4538	RAM/EEPROM
DEADBAND4	16-bit register that holds the Deadband value for controller 4.	U_16	4539	RAM/EEPROM
DEADBAND5	16-bit register that holds the Deadband value for controller 5.	U_16	4540	RAM/EEPROM
DEADBAND6	16-bit register that holds the Deadband value for controller 6.	U_16	4541	RAM/EEPROM
DEADBAND7	16-bit register that holds the Deadband value for controller 7.	U_16	4542	RAM/EEPROM
DEADBAND8	16-bit register that holds the Deadband value for controller 8.	U_16	4543	RAM/EEPROM
DEADBAND9	16-bit register that holds the Deadband value for controller 9.	U_16	4544	RAM/EEPROM
DEADBAND10	16-bit register that holds the Deadband value for controller 10.	U_16	4545	RAM/EEPROM
DEADBAND11	16-bit register that holds the Deadband value for controller 11.	U_16	4546	RAM/EEPROM
DEADBAND12	16-bit register that holds the Deadband value for controller 12.	U_16	4547	RAM/EEPROM
DEADBAND13	16-bit register that holds the Deadband value for controller 13.	U_16	4548	RAM/EEPROM
DEADBAND14	16-bit register that holds the Deadband value for controller 14.	U_16	4549	RAM/EEPROM
DEADBAND15	16-bit register that holds the Deadband value for controller 15.	U_16	4550	RAM/EEPROM
DEADBAND16	16-bit register that holds the Deadband value for controller 16.	U_16	4551	RAM/EEPROM

See also

Controller Cooling Differential

Controller Heating Differential

Controller Mode

Controller Setpoints

Controller Outputs

Output Masks

2.10.6 Output Masks

The Zen IoT also has the ability to invert the state of its controller outputs using output masks. These output masks perform an exclusive "OR" (XOR) function on the output relay state so that if the mask bit is a 1 then the output state will be inverted, and if it is a 0 then the output state will be unchanged. This register is stored in RAM and EEPROM so it's state is restored at power-up.

The following output masks are available for this purpose.

Name	Description	Symbol Type	Register Number	Memory Type
OUTPUT_MASK_RELAY_A_D	8-bit register that holds an XOR mask which is used to invert the state of the on board relays A - D.	U_8	4424	RAM/EEPROM

Output Mask Relay A - D
Register (4424) is a 8 bit register used to invert the relay output states of on board relays A to D.

The individual bit functions are shown in the table below;

Bit	Description	Function
b0	Inverts the state of onboard relay A.	1 = Invert State of Relay A 0 = Relay A State Unchanged
b1	Inverts the state of onboard relay B.	1 = Invert State of Relay B 0 = Relay B State Unchanged
b2	Inverts the state of onboard relay C.	1 = Invert State of Relay C 0 = Relay C State Unchanged
b3	Inverts the state of onboard relay D.	1 = Invert State of Relay D 0 = Relay D State Unchanged
b4-b7	Not used. Reserved for future development.	No Function

2.11 Serial Port

The Zen IoT controller has 3 independent serial communications channels, each with different hardware interface options and baud rate options as shown below.

Serial Port Number	Hardware Interface Options	Baud Rate Options
1	RS232/RS485 or Ethernet or WiFi (options must be specified at time of order)	Adjustable from 2400-230400 baud (See Port 1)
2	RS232/RS485 (automatic detection of RS232 input)	Adjustable from 2400-230400 baud. (See Port 2)
3	5v TTL levels via a quad 2.5mm jack socket. USB adapter available from Define Instruments Ltd.. (Bluetooth option also available).	Adjustable from 2400-19200 baud. (See Port 3)

Serial Port Modes - Registers 8215 - 8217
Registers 8215 to 8217 are 8 bit registers which control the functionality of serial ports 1 to 3 respectively. The following table shows the register value for currently available serial port protocols.

Value	Mode
0	ASCII
1	Modbus RTU slave
2	Macro Master mode
3	Printer
4	Modbus/TCP wrap.
5	Intech/Modbus RTU slave.
6	LCD touch panel
7	Modbus RTU Master
8	Bridge to port 1.
9	Bridge to port 2.
10	Bridge to port 3.
11	Ethernet IP (firmware V0.08.01 onwards)
12	MQTT (Port 1 only from firmware V2.2.01 onwards)

Note: Some of the above protocols shown above are specific to different serial ports and some require special hardware support. Please check with modes for Port 1, Port 2, and Port 3.

Register 8219 – Serial Port In Use
Register 8219 is an 8-bit register that reports which serial port is currently in use. Because the Zen IoT Series controllers have multiple serial ports it may be necessary for an external device to know which one it is currently using. For example, a read of 8219 via serial port 1 results in a number 1 being returned.

Name	Description	Symbol Type	Register Number
SERIAL_PORT_NO	8-bit read only register shows which serial port is being accessed.	U_8_R	8219

See also
ASCII Mode Format

ASCII Text Registers

Modbus Master

Port 1

Port 2

Port 3

2.11.1 Serial Port Settings

Registers 8207 to 8209 are 8-bit registers used to store the serial port settings for serial ports 1 to 3. Bits 0 to 2 are used to hold the baud rate information. Bit 3 is used to select between 7 or 8 data bits. Bits 4 and 5 are used to select the parity type. Bits 6 and 7 allow different transmit delay times to be selected. The various options available are shown as follows:

Bits 7, 6 **Transmit delay**
00 = 2 milliseconds
01 = 20 milliseconds
10 = 50 milliseconds
11 = 100 milliseconds

Bits 5, 4 **Parity**
00 = no parity
01 = odd parity
10 = even parity

Bit 3 **Data bits**
0 = 8 data bits
1 = 7 data bits

Bits 2, 1, 0 **Baud rate**
000 = 2400 baud
001 = 4800 baud
010 = 9600 baud
011 = 19200 baud
100 = 38400 baud
101 = 57600 baud
110 = 115200 baud
111 = 230400 baud

NOTE: The baud rates shown above are not available on all ports. Please check bauds rates for Port 1, Port 2 and Port 3

NOTE: If these registers are modified via the serial port, the controller response (and any subsequent communications) is issued at the new modified baud rate/parity settings and may result in a communications error at the master device.

2.11.2 Serial Address

These are 8-bit registers that set the serial address for serial ports 1 - 3 respectively. The controller address can be set from 1 to 255. The controller address should not be set to 0 as this address is reserved (all controllers respond to a request at address zero).

2.11.3 Serial Strings In Macro Master Mode

The serial receive protocol for macro master mode is not set in firmware, it is user defined in the macro. For each serial channel, the start of string character, end of string character, and string length can be defined. The following table shows which register groups are used for this purpose.

Register Group	Serial Port	Memory Type
8509/8226/8230	Port 1	RAM/EEPROM
8510/8227/8231	Port 2	RAM/EEPROM
8511/8228/8232	Port 3	RAM/EEPROM

Start Of String Character - (Registers 8509 - 8511)
Registers 8509 to 8511 are 8 bit unsigned registers that define the start of string characters used in macro master mode for serial ports 1 -3 respectively.

Start of string character **not zero** - Setting one of these registers to a any value other than zero will cause the respective serial port to search incoming serial data for a character that matches the start of string character. If an incoming character does the match the start of string character, the new character is stored in the first byte of the serial receive buffer and the serial port will continue receiving data until string length and/or end of string conditions are met (see details below). If the incoming character does not match the start of string it is ignored and the serial port continues searching for a correct start of string character.

Start of string character **zero** - Setting one of these registers to a value of zero prevents the respective serial port from searching for a start of string character. In this mode each new byte of data received is stored in the serial receive buffer until string length and/or end of string conditions are met (see details below).

String Length Character - (Registers 8230- 8232)
Registers 8230 to 8232 are 8 bit unsigned registers that define the string length used in macro master mode for serial ports 1 -3 respectively.

Note: the following logic for the string length character will only become active when the conditions specified by the start of string character have been met (see details above).

String length **not zero** - Setting the string length register to a value other than zero will cause the respective serial port to keep adding incoming serial data to it's serial receive buffer until the number of bytes in the buffer is equal to the string length value. Once it has received the specified number of bytes, it will then do one of two options.

1) If the end of string character for the respective serial port is disabled (i.e. set to a value of zero), the receive ready flag will be set and further reception will be disabled.

2) If the end of string character for the respective serial port is enabled (i.e. set to a non zero value), the serial port will continue receiving data into it's serial receive buffer until it receives the specified end of string character. At this point the receive ready flag will be set and further reception will be disabled.

String length **zero** - Setting the string length register to a value of zero will cause the respective serial port to keep adding incoming serial data into the serial receive buffer until one of the following two conditions has been satisfied.

1) If the end of string character for the respective serial port is disabled (i.e. set to a value of zero), the serial port will continue receiving data into it's serial receive buffer until it's serial receive buffer is full (i.e. 255 bytes have been received). At this point the receive buffer overflows and it is reset to start looking for a new string.

2) If the end of string character for the respective serial port is enabled (i.e. set to a non zero value), the serial port will continue receiving data into it's serial receive buffer until it receives the specified end of string character. At this point the receive ready flag will be set and further reception will be disabled.

End Of String Character - (Registers 8226 - 8228)
Registers 8226 to 8228 are 8 bit unsigned registers that define the end of string characters used in macro master mode for serial ports 1 -3 respectively.

Note: the following logic for the end of string character will only become active when the conditions specified by the start of string character and the string length character have been met (see details above).

End of string character **not zero** - Setting the end if string register to a value other than zero will enable the detection of an end of string character. If the conditions specified for the start of string character and the string length character have been satisfied, then the serial port will continue receiving data into it's serial receive buffer until it receives the specified end of string character. At this point the receive ready flag will be set and further reception will be disabled.

End of string character **zero** - Setting the end if string register to a value of zero will disable the detection of an end of string character. If the conditions specified for the start of string character and the string length character have been satisfied, then the serial port will continue receiving data into it's serial receive buffer until it's serial receive buffer is full (i.e. 255 bytes have been received). At this point the receive buffer overflows and it is reset to start looking for a new string.

Note: Although it is possible to disable all of the above registers (i.e. all set to zero), this is not the recommended mode of operation as it will eventually cause the receive buffer to over flow, which in turn will cause the buffer to be flushed and the serial port to be reset. If you want to receive a large number of bytes without specifying a start or end character, then you should set only the string length to a large number (< 255).

2.11.4 Serial Receive Count

Registers 8454 to 8456 are 8-bit registers that show the received message length for serial ports 1 - 3 respectively (i.e. how many bytes have been received by the serial port in a message). Although these registers can be read in all serial modes, their main purpose is for use in master mode under macro control of the serial ports.

2.11.5 Serial Transmit Count

Registers 8465 to 8467 are 8-bit unsigned registers that relate to the number of bytes to be transmitted by serial ports 1 - 3 respectively. They are only intended for use with the serial port set in Macro master mode.

Writing To The Transmit Count Register
Normally the 'PRINT' command is used in a macro to send an ASCII string out via one of the serial ports. However sometimes it may be necessary to send non ASCII strings or complicated strings which need extra processing for checksums etc. In this case the macro would write directly to the serial buffer and load the outgoing string byte by byte. When this process has been completed, the string is transmitted by writing the length of the string to the appropriate TRANSMIT_COUNT register.

NOTE: The serial port must be operating in Macro Master mode for correct operation.

Reading The Transmit Count Register
Reading the transmit count register while a string is being transmitted will show the progress of the transmit process by pointing to the next byte to be transmitted.

2.11.6 Serial Receive Timeout

The serial receive time out registers (COMS_TIMEOUT1 - COMS_TIMEOUT3) work in conjunction with the RECEIVE_IDLE_TIMEx registers to detect a break or malfunction on a particular serial communications port. Once the timeout period is exceeded, the Zen IoT on board relay 2 will turn off to signal an error or reset any external com's equipment such as modems etc.

NOTE: Relay 2 will only operate in this manner if it is not configured for any other control mode. (See Relay 2 RX Timeout for more information on this feature).

Each serial port has an associated COMS_TIMEOUTx register and a RECEIVE_IDLE_TIMEx register. The RECEIVE_IDLE_TIMEx register is reset to zero whenever activity is detected on the related serial port. When no activity is present, RECEIVE_IDLE_TIMEx counts up in 1 second intervals. When the value of RECEIVE_IDLE_TIMEx exceeds the limit set by COMS_TIMEOUTx, then relay 2 is turned off and held off until serial activity resumes or the Zen IoT is re-powered.

This feature can be individually enabled for each serial port by setting the value of its related COMS_TIMEOUTx register to greater than 0. If COMS_TIMEOUTx equals 0 then the feature is disabled for that particular serial port, but it can still be enabled for other serial ports. If multiple serial ports are enabled, then the first one to exceed its timeout value will turn off relay 2.

See also
Relay 2 RX Timeout

2.11.7 ModBus Master

The Modbus master macro is a special macro area which can be used to configure the Zen IoT controller as a Modbus master which is capable of reading and writing to other Modbus slave devices. Several special Modbus macro commands are included which can only be used in the MODBUS_MASTER_MACRO. In addition, all other macro commands can be used as well. (Note: Although the MODBUS_MASTER_MACRO is primarily designed for the purpose of providing a Modbus master function, it can be used for other purposes and does not strictly require the inclusion of any Modbus macro commands).

When implementing a Modbus master function the desired serial port(s) must also be set to Modbus master mode. See serial port modes.

Dual Master/Slave Function.
Zen IoT firmware revisions V0.07.07 onwards support dual Modbus Master/Slave mode. This was designed for applications where the Zen IoT is operating as the primary Modbus master, but may need to accessed for maintenance and configuration from time to time. (See note below on dual mode with Ethernet operation).

The operation of the dual master/slave function is as follows. Normally the Zen IoT will be operating in Modbus master mode, and each time it receives an incoming reply it compares the slave address (unit ID) to ensure it matches the slave it tried to access. If it finds the unit ID is different to what it expected, it then checks to see if the incoming slave address matches the its own address. If it does then it switches out of Modbus master mode temporarily and starts operating as a Modbus slave. It will stay in Modbus slave mode as long as it continues to receive incoming messages within the time period specified by RESPONSE_TIME. If there are no messages received within this time it then switches back to Modbus master mode.

NOTE: Modbus was not designed to be operated with more than 1 master on the bus so using the dual mode function does not comply with the standard. It may take several attempts for a secondary master to connect with the Zen IoT when its operating in Modbus Master mode and the integrity of data on the bus may be compromised during this time. This is particularly true when POLL_TIME is set to very small values or when there is a large amount of Modbus Master traffic. Care should be taken when using this mode. It is recommended that POLL_TIME be set to value which provides sufficient idle periods on the bus and any secondary slave should time its connection attempts to occur during these idle periods.

Note: To use this feature the address (unit ID) of the Zen IoT acting as the primary master should be set to a unique address which is not used by any other slaves on the bus.

Modbus Master Registers
The Modbus Master mode can only be used under the control of the Modbus Master macro. The registers shown below are only intended for this use.

Name	Description	Symbol Type	Register Number
CRC_ERROR	Read only flag - response from slave received with CRC checksum error.	B_5	8464
DATA_ERROR	Read only flag - Modbus attempted to read/write incorrect data type to slave.	B_6	8464
MESSAGE_COMPLETE	Read only flag - previous message transaction is completed correctly.	B_7	8464
MESSAGE_TIMEOUT	Read only flag - no response received from slave.	B_4	8464
MODBUS_MASTER_FLAGS	8-bit read only register which contains status flags for the Modbus master macro.	U_8_R	8464
POLL_TIME	8-bit register which sets the polling time for the Modbus master macro (1 count = 0.01 seconds).	U_8	8462
RESPONSE_TIME	8-bit register which sets the message response timeout for the Modbus master macro (1 count = 0.1 seconds).	U_8	8463

Modbus/TCP Master Mode
Serial port 1 can be configured to provide Modbus/TCP Master operation with specific hardware fitted. You will need to contact Define Instruments Ltd. and specify this at the time of ordering as this option is not upgradable in the field.

NOTE: Once the hardware Modbus/TCP Ethernet option is fitted to port 1, the port is restricted to operate only in Modbus/TCP Master mode or Modbus/TCP slave mode depending on how the Ethernet adapter has been configured. (Dual master/slave mode is not available with the standard Ethernet option. If you require this functionality in Ethernet mode contact your Zen IoT distributor for more information.)

Simple Modbus Master Example
The following example shows a typical Modbus master implementation in the MODBUS_MASTER_MACRO. This example shows the simplest form of the command which allows one register to be read or written in each Modbus command. (See the section below on Zen IoT enhancements)

```
Modbus_Master_Macro:
  &POLL_TIME =10                          //100mS (1 COUNT = 10mS)
  &RESPONSE_TIME=5                        //0.5S (1 count = 0.1S)
  MODBUS_READ 1 (1,40004,&CH3,MB_LONG)           //
  GOSUB CHECK_MESSAGE
  MODBUS_WRITE 1 (1,&CH4,40005,MB_LONG)
  GOSUB CHECK_MESSAGE
  MODBUS_READ 1 (1,30005,&CH2,MB_LONG)
  GOSUB CHECK_MESSAGE
RETURN

CHECK_MESSAGE:
  IF (&MODBUS_MASTER_FLAGS AND 0x7F) != 0 THEN
    WRITE "   ___ERROR - "+&MODBUS_MASTER_FLAGS+"      "
  ENDIF
RETURN
```

As shown in the above example, the MODBUS_READ and MODBUS_WRITE commands have a similar format. The number following the command (i.e. outside the brackets) specifies the serial port number to be used for the Modbus master mode. Then inside the brackets the format is as follows;

([slave device address],[Modbus source register],[Modbus destination register],[register type])

[slave device address] = controller address of the Modbus slave can be a number from 0 to 247.

[Modbus source register] = in a read command this can be a number from 30001 to 49999. For a write command you can specify the register name and the compiler while calculate the register number.

[Modbus destination register] = in a write command this can be a number from 30001 to 49999. For a read command you can specify the register name and the compiler while calculate the register number.

[register type] = This specifies the size and type of the register being accessed with the following options being available.

MB_BIT	Not supported at present
MB_BYTE	8 bit register
MB_SHORT	16 bit register
MB_24	24 bit register
MB_24_SWAPPED	24 bit register with MSW and LSW swapped
MB_LONG	32 bit register
MB_LONG_SWAPPED	32 bit register with MSW and LSW swapped
MB_FLOAT	32 bit single precision floating point register
MB_FLOAT_SWAPPED	32 bit single precision floating point register with MSW and LSW swapped
MB_STRING	Text string register (only supported with enhanced command format - see note below)

NOTE:
If the register types for the source and destination registers in the Modbus command do not match, the Zen IoT will attempt to correct this if possible. In the case of a float/fixed point mismatch, the Zen IoT will attempt to type cast the value into the different format, but this will not always be possible in the case of large floating point numbers so the user should be careful if using this feature to ensure that range problems do not occur.

Currently Supported Functions
At present input registers (30000 range) and holding registers (40000 range) are supported in the Modbus master mode.

Modbus Master Flags
Register 8464 (MODBUS_MASTER_FLAGS) can be used in Modbus Master mode to determine if a transmission error occurred in the previous communication. The following errors are possible.

Bits 0 - 3 (Standard Modbus Exception Error Codes)
1 = Illegal function call (function call not supported by slave)
2 = Illegal data address (the data address specified in the command is not available in the slave)
3 = Illegal data value (a data value specified in the command is not in the acceptable range)
4 = Slave device failure
5 = Acknowledge
6 = Slave device busy
7 = Negative acknowledge
8 = Memory parity errors

Bit 4 = Message timeout
Bit 5 = CRC receive error
Bit 6 = data type error
Bit 7 = Reception complete and ready for new command

Poll Time
Register 8462 is an 8 bit register that defines the rate at which the Modbus master macro is executed. Each count of the POLL_TIME register represents a time interval of 10mS so a value of 100 would result in the Modbus master macro being executed once a second. If a value of 0 is written to POLL_TIME the Modbus master macro will execute as fast as the operating system will allow.

Register 8462 defaults to a value of 10 (i.e. 0.1S) each time the controller is powered up and any writes to this register are stored in volatile memory which is lost at power down. For this reason the register &POLL_TIME should be written in either the RESET_MACRO or the MODBUS_MASTER_MACRO.

Response Time
Register 8463 is an 8 bit register that defines the maximum time the Modbus master will wait for a slave to respond. Each count of the RESPONSE_TIME register represents a time interval of 100mS so a value of 10 would result in the Modbus master waiting for up to a second for a slave response. If the slave device fails to respond within the set time, bit 4 of the MODBUS_MASTER_FLAGS register is set and the Modbus master macro continues execution at the next line of macro code.

Register 8463 defaults to a value of 10 (i.e. 1S) each time the controller is powered up and any writes to this register are stored in volatile memory which is lost at power down. For this reason the register &RESPONSE_TIME should be written in either the RESET_MACRO or the MODBUS_MASTER_MACRO.

Zen IoT Modbus Master Enhancements
The Modbus master example shown above is the simplest from of Modbus master command. The Zen IoT has been enhanced to allow block reads and writes of registers and also to allow variable expressions in the Modbus master command. This allows the implementation of more efficient and more compact Modbus master macros, saving processing time and macro memory space. The enhanced form of the Modbus master commands are shown below.

MODBUS_READ [serial port number] ([slave address],[remote source register],[local destination register],[register type],[number of registers to be read])
MODBUS_WRITE [serial port number] ([slave address],[local source register],[remote destination register],[register type],[number of registers to be written])

In the enhanced command;
[slave device address] = controller address of the Modbus slave. Can either be a number from 0 to 247 or can also be a variable register which holds a number from 1 to 247.

[remote source register] = this can be a constant (i.e. number from 1 to 19999) or it can be a variable register which holds a number from 1 to 19999 or it can be an expression.

[remote destination register] = this can be a constant (i.e. number from 1 to 19999) or it can be a variable register which holds a number from 1 to 19999 or it can be an expression.

Note: With the enhanced command format you no longer need to specify the address in the 30000 or 40000 format for remote registers. You should now just specify the register number only without the 30000 or 40000 offset. The [register type] field now specifies whether the register is an input or holding register.

[local source register] = **this must be specified as a register**, with or without array index.

[local destination register] = **this must be specified as a register**, with or without array index.

[register type] = this is similar to the simple example above but it must also have either +MB_HOLD or +MB_INPUT added to the end to specify whether the register is a holding register or an input register. So a typical example would be MB_LONG+MB_HOLD. The enhanced command also includes the new register type of MB_STRING which allows the reading and writing to text string registers via the Modbus master mode. See note below on Text Strings.

[number of registers to be read] = This field is mandatory for the enhanced Modbus master command and must be a constant greater than zero. The following range is allowed for different commands with different register types. (See note below regarding use of this field for with the register type MB_STRING)

Read Commands
1 to 250 for MB_TEXT
1 to 125 for MB_BYTE , MB_SHORT
1 to 62 for MB_24, MB_24_SWAPPED, MB_LONG ,MB_LONG_SWAPPED, MB_FLOAT, MB_FLOAT_SWAPPED

Write Commands
1 to 246 for MB_TEXT
1 to 123 for MB_BYTE , MB_SHORT
1 to 61 for MB_24, MB_24_SWAPPED, MB_LONG ,MB_LONG_SWAPPED, MB_FLOAT, MB_FLOAT_SWAPPED

Block Reads/Writes In Modbus Master Commands
The first Modbus master example shown above only allows 1 register to be read or written to in each Modbus master command line. The Zen IoT controller now allows blocks of registers to be read or written to using a single Modbus read or write command. The following example shows the new form.

```
MODBUS_READ 2 (1, 111, &AUX1, MB_LONG+MB_HOLD, 6)
MODBUS_WRITE 1 (1, &SETPOINT1, 111, MB_LONG+MB_HOLD, 6)
```

Using Variables In Modbus Master Commands
With the simplest form of the Modbus master command the parameters are specified directly in the command (i.e. with fixed values). The Zen IoT controller has some new enhancements which allow variables and expressions to be used in the Modbus master command. This allows for next loops to be used with Modbus master commands and greatly reduces the amount of macro code needed. The following example shows how variables and expressions can be used with the new Modbus master command.

```
#src = addr(&SETPOINT1)
for #address = 1 to 6
  MODBUS_READ 2 (#address, #src, &AUX1[#address-1], MB_LONG+MB_HOLD, 1)
  #src = #src + 2
next #address
```

The example above could be used to read data from 6 other Define Instruments Ltd. controllers with addresses 1 - 6. It would read the value of setpoint1 in controller 1 and store it in &AUX1. Then it would read the value of setpoint2 in controller 2 and store it in &AUX2 and so on.

Text Strings In Modbus Master Mode
The simple form of the Modbus master command did not allow text strings to be accessed via the Modbus master command but the enhanced command does. To read or write to a text string register the register type should be specified as MB_STRING. In this mode the field [number of registers to be read] is interpreted as [number of characters to be read] and any where from 1 to 250 characters can be read. Only one text string register can be read with each command - reading of successive text registers is supported.

Note: If [number of characters to be read] is set to a value which is greater than the actual size of the text string in the remote register, the excess characters are padded out with ASCII nulls (0x00). Although this condition is acceptable it is not very efficient as extends the length of the string by adding unnecessary bytes. For this reason the [number of characters to be read] should match the size of the remote text register.

2.11.8 Bridging Modes

The Zen IoT controller also supports a transparent bi-directional serial bridge mode between any two serial ports. This mode allows serial traffic from one serial port to be bridged to another port with differing baud rate and parity settings on each port.

Note: both ports must be set to bridging mode for correct operation. So if for example you wanted to bridge port 1 and port 2, you would setup port 1 serial mode to be "Bridge to port 2" and setup port 2 serial mode to be "Bridge to port 1".

Serial Buffering
In bridge mode incoming serial data is buffered until the receiver encounters a gap in the transmission of greater than 3.5 character spaces. This then triggers the outgoing transmission on the bridged port so that all data bytes are sent contiguously in a single frame. This allows for protocols such as Modbus to be used with different baud rates without violating timing requirements.

Note: serial buffering will cause a data delay across the bridge which may need to be compensated for in some systems. Serial buffering was only applied to firmware version V0.07.08 onwards so older firmware does not include buffering and may not be suitable for some protocols.

Serial Escape Sequence
Once the Zen IoT has been set to bridge mode, the serial ports involved can only pass data between each other in a transparent bridge. Neither serial port can access the Zen IoT's internal registers in this mode. However an escape sequence can be sent to one of the bridged ports to force it out of bridge mode and cause it to enter Modbus slave mode.

The escape sequence consists of sending a separate 2 byte frame consisting of the ASCII characters <DLE><EOT>. The DLE (Data Link Escape) character is an 0x10 hex followed by the EOT (End Of Transmission) character 0x04 hex. The escape sequence will not work if it is found in frame greater than 2 bytes so it must be sent with the 3.5 character gaps before and after it.

Note: when an escape sequence is received in bridge mode, only the receiving port is set to Modbus slave mode. The other bridged serial port will still be set to bridge mode. Provided that there is no traffic on the other bridged serial port, the first serial port will still function correctly in Modbus slave mode, allowing the user to change the configuration of the other bridged serial port to the desired mode.

Note: the serial escape sequence was only applied to firmware version V0.07.08 onwards. Older firmware requires the user to manually change the serial mode out of bridging mode.

Ethernet To Serial Bridging
If port 1 is fitted with an Xport Ethernet adapter, bridging is also possible from Ethernet to port 2 (RS232/485) or to port 3 (USB adapter). In this case care must be taken to specify the correct baud rate for the Ethernet adapter being used. The Modbus/TCP version of the Xport communicates to the Zen IoT at 115200 baud while the ASCII version of the Xport communicates at 230400 baud. The user must select the appropriate baud rate for port 1 depending on the adapter used.

See also

Port 1

Port 2

Port 3

2.11.9 Port 1

Serial port 1 can be supplied with a RS422/RS485 option or an Ethernet option. (Options must be specified at time of order - not upgradable in field)

The following table shows all registers associated with serial port 1.

Name	Description	Symbol Type	Register Number	Memory Type
BAUDRATE1	8-bit register sets the serial port 1 baud rate (0 = 2400, 1 = 4800, 2 = 9600, 3 = 19200, 4 = 38.4 k, 5 = 57.6 k, 6 = 115.2 k, 7 = 230.4 k).	U_8	8207	RAM/EEPROM
SERIAL_BUFFER1	Start of serial transmit/receive buffer for port 1 (255 bytes long).	U_8	12289	RAM
RECEIVE_COUNT1	16-bit register which shows how many characters have been received by the serial port 1.	U_16	8454	RAM
TRANSMIT_COUNT1	16-bit register which sets how many characters are to transmitted by the serial port 1.	U_16	8465	RAM
RECEIVE_FLAGS1	8-bit register. Serial receive flags. Used in master mode.	U_8	8234	RAM
RECEIVE_READY1	This flag shows that a new message string has been received on port 1 in master mode.	B_0	8234	RAM
RECEIVE_RESULT1	32-bit register holds the 1st numeric value received in a string via serial port 1.	S_32	349	RAM
SERIAL_ADDRESS1	8-bit register holds the serial address of the controller.	U_8	8211	RAM/EEPROM
SERIAL_MODE1	8-bit register sets the serial mode for port 1.	U_8	8215	RAM/EEPROM
SERIAL_POINTER1	16-bit pointer used for string compare commands with serial port 1.	U_16	8458	RAM
START_OF_STRING_CHARACTER1	8-bit register. Sets ASCII character for the start of serial receive string in master mode for port 1.	U_8	8509	RAM/EEPROM
END_OF_STRING_CHARACTER1	8-bit register. Sets ASCII character for the end of serial receive string in master mode for port 1.	U_8	8226	RAM/EEPROM
STRING_LENGTH1	8-bit register. Sets string length of serial receive string in master mode for port 1.	U_8	8230	RAM/EEPROM
RECEIVE_IDLE_TIME1	16-bit register. Shows seconds of inactivity on serial port 1. (max. count = 65535 seconds)	U_16	4572	RAM
COMS_TIMEOUT1	16-bit register that specifies the timeout interval in seconds of inactivity on serial port 1. (Range 0 - 65535 seconds, 0=disabled). See Relay 2 RX Timeout)	U_16	4568	RAM/EEPROM
ETHERNET_ADAPTOR	1 bit read only flag that indicates the presence of an Ethernet adaptor on serial port 1.	B_1_R	8222	RAM

Serial Port 1 Modes - Registers 8215
Registers 8215 is an 8 bit register which controls the serial protocol used by serial port 1. The following table shows the register value for currently available serial protocols.

Value	Mode
0	ASCII
1	Modbus RTU slave (or ModbusTCP slave with Ethernet option fitted).
2	Macro Master mode.(See Modbus/TCP Slave Mode)
3	Printer
4	Modbus/TCP wrap.
5	Intech/Modbus RTU slave. (default)
6	LCD touch panel
7	Modbus RTU Master. (See Modbus/TCP Master Mode)
8	N/A - (bridge to port 1 for other serial ports)
9	Bridge to port 2.
10	Bridge to port 3.
11	Ethernet IP (firmware V0.08.01 onwards)
12	MQTT (firmware V2.2.01 onwards)

Note: When set to Modbus RTU slave mode, any packets that do not comply with the Modbus format will be interrogated as possible ASCII packets (i.e. mode 0 above - this should not be confused with

Modbus ASCII packets).

Baud Rates - Port 1
The baud rates for port 1 are controlled by bits 0 to 2 of register 8207 and the available options are shown below.

Bits 2, 1, 0	Baud rate
	000 = 2400 baud 001 = 4800 baud 010 = 9600 baud (default) 011 = 19200 baud 100 = 38400 baud 101 = 57600 baud 110 = 115200 baud 111 = 230400 baud.

The Zen IoT controller can be supplied with either an Ethernet or RS422/RS485 option fitted to serial port 1. If the RS422/RS485 option is detected then the above baud rates are available. If an Ethernet adapter is detected, the internal serial baud rate for port 1 will be set according to the serial mode being used. If Modbus master or slave modes are being used then the baud rate will be preset to 115200,8,N,1 baud for an Xport Modbus/TCP adapter. For all other modes (except bridging modes) the baud rate will be preset to 230400,8,N,1 baud for an Xport ASCII adapter. Bridging modes can be used with either type of Xport Ethernet adapter so for these modes the user must select the appropriate baud rate setting.

Number of Data Bits - Port 1
Bit 3 of register 8207 allows the user to select whether they require 7 or 8 bit data. This option is only available when the Macro Master mode protocol is selected.

Bit 3	Data bits
	0 = 8 data bits (default) 1 = 7 data bits

Parity Setting - Port 1
Bits 4 & 5 of register 8207 allow the user to select a parity setting as per the options below.

Bits 5, 4	Parity
	00 = no parity (default) 01 = odd parity 10 = even parity

Transmit Delay - Port 1
To allow for slower devices, a transmit delay is available for most serial protocols. Bits 6 and 7 of register 8207 allow different transmit delay times to be selected. The various options available are shown as follows:

Bits 7, 6	Transmit delay
	00 = 2 milliseconds (default) 01 = 20 milliseconds 10 = 50 milliseconds 11 = 100 milliseconds

2.11.9.1 Serial Buffer Port 1

Registers 12289 to 12544 are all 8-bit unsigned registers that are used as a buffer for serial port 1 received and transmitted data. They are used in all serial port modes, but their intended use is in master mode under macro control. By accessing these registers individually, a message string can be built up or interrogated, byte by byte.

NOTE: Although registers 12289 to 12544 can be written to, it is not recommended unless you have a thorough knowledge of how the serial port operates. Writing the wrong value to these registers could cause the serial port to lock up.

See Also
Serial Transmit Count - Registers 8465 - 8467

Serial Receive Count- Registers 8454 - 8456

2.11.9.2 MQTT Mode Port 1

Port 1 includes a MQTT V3.1.1 client mode which is used in conjunction with a macro.

The table below shows the various MQTT related registers.

Name	Description	Symbol Type	Register Number	Memory Type
MQTT_CONNECT_FLAGS	8-bit register used in the MQTT CONNECT command to specify the behavior of the MQTT connection. (see MQTT_CONNECT_FLAGS below for details).	U_8	8542	RAM/EEPROM
MQTT_CLIENT_ID_SOURCE	16-bit register which points to any string register containing the client ID string that is sent in the MQTT CONNECT command.	U_16	4693	RAM/EEPROM
MQTT_USERNAME_SOURCE	16-bit register which points to any string register containing the username string that is sent in the MQTT CONNECT command.	U_16	4696	RAM/EEPROM
MQTT_USERNAME_SOURCE	16-bit register which points to any string register containing the password string that is sent in the MQTT CONNECT command.	U_16	4697	RAM/EEPROM
MQTT_WILL_TOPIC_SOURCE	16-bit register which points to any string register containing the will topic string that is sent in the MQTT CONNECT command.	U_16	4694	RAM/EEPROM
MQTT_WILL_MESSAGE_SOURCE	16-bit register which points to any string register containing the will message string that is sent in the MQTT CONNECT command.	U_16	4695	RAM/EEPROM
MQTT_STATUS_FLAGS	16-bit register which contains various status flags associated with MQTT mode. (see MQTT_STATUS_FLAGS below for details).	U_16	4691	RAM
MQTT_KEEP_ALIVE	16-bit register which contains MQTT connection keep alive time in seconds. An MQTT broker will close the connection if it does not receive any data from the client within this time period. (valid data range is 1 - 65535 seconds)	U_16	4692	RAM/EEPROM
MQTT_PACKET_CONTROL	8-bit register which controls which MQTT packet is sent. This register is written to by the macro to trigger the various MQTT packets being sent. (see MQTT_PACKET_CONTROL below for details).	U_8	8540	RAM
MQTT_RESONSE_CODES	8-bit register which shows the response code (if any) from the previous MQTT command. (see MQTT_RESONSE_CODES below for details).	U_8	8541	RAM

MQTT Connect Flags
The 8th byte of an MQTT connect packet contains connection flags which control the behavior of the new connection. The flags are defined by the MQTT V3.1.1 standard and are shown below.

Bit	Name	Description
B_0	Reserved	Always set to zero.
B_1	Clean Session	This bit is used to control the lifetime of the Session state.
B_2	Will Flag	If the Will Flag is set to 1 this indicates that, if the Connect request is accepted, a Will Message MUST be 483 stored on the Server and associated with the Network Connection.
B_3 & B_4	Will QoS	These two bits specify the QoS level to be used when publishing the Will Message. (Valid values are 00=QoS0, 01=QoS1, 10=QoS2).
B_5	Will Retain	This bit specifies if the Will message is to be retained when it is published by the server.
B_6	Password Flag	This bit specifies if a password is included in the connect packet..
B_7	Username Flag	This bit specifies if a username is included in the connect packet..

The macro code is responsible for setting the connect flag value before the connection packet is sent.

MQTT Status Flags

The MQTT status flags show the state of the MQTT connection and confirm that various commands have been sent or received correctly. The table below shows the various status bits in the register.

Bit	Name	Description
B_0	Reserved	Always set to zero.
B_1	MQTT_CONNECT_OK	This bit is shows that an MQTT connection with a broker is currently active.
B_2	MQTT_PUBLISH_OK	This bit is shows that the previous publish command was sent ok. If QoS >=1 then this bit shows that all the required acknowledgments were received ok.
B_3	MQTT_SUBSCRIBE_OK	This bit is shows that the previous subscribe command was sent ok. If QoS >=1 then this bit shows that all the required acknowledgments were received ok and there were no subscription errors int he servers response.
B_4	MQTT_SUBSCRIBE_ERROR	This bit is shows that the response to the previous subscribe command failed and came back with subscribe errors in the response. See MQTT Response Codes below
B_5	MQTT_UNSUBSCRIBE_OK	This bit is shows that the previous unsubscribe command was sent ok. If QoS >=1 then this bit shows that all the required acknowledgments were received ok and there were no subscription errors int he servers response.
B_6	MQTT_PING_OK	This bit is shows that the previous ping command was acknowledged successfully.
B_7	MQTT_PACKET_ERROR	This bit specifies that the length of a received MQTT packet' does not match the length bytes contained in the packet.

MQTT Response Codes

The MQTT_RESONSE_CODES register contains the response code byte of commands such as CONACK and PUBACK. After sending a CONNECT or PUBLISH command the MQTT_RESONSE_CODES register can be read to check for any errors.

MQTT Packet Control

The MQTT packet control register translates to the first byte of the fixed header in every MQTT packet.

Bits 0 -3 are the MQTT flags specific to each type of MQTT Control packet while bits 4 - 7 define the Control packet type.

The table below shows the different control packet types for bits 4 - 7 that can be sent from the Zen IoT to the MQTT message broker and also shows the meaning of bits 0-3 for each control packet type.

(**Note:** The MQTT V3.1.1 standard contains more control packet types than shown below. These control packet types are generated by the Zen IoT automatically, depending on QoS requirements).

Name	Value	Description	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	0	Reserved	0	0	0	0
CONNECT	0x10	Client request to connect to server	0	0	0	0
PUBLISH	0x30	Client publishes message to server	DUP	QoS	QoS	RETAIN
SUBSCRIBE	0x80	Client subscribes to a topic on the server	0	0	1	0
UNSUBSCRIBE	0xA0	Client unsubscribes from a topic on the server	0	0	1	0
PINGREQ	0xC0	Client pings the server	0	0	0	0
DISCONNECT	0xE0	Client disconnects from the server	0	0	0	0

See Also
MQTT Macro Example

2.11.9.2.1 MQTT Example Macro

The following lines show the source code of an example macro demonstrating how to use MQTT mode in conjunction with the macro. (Note: this macro was written for a Zen controller using a FM1602 LCD with 3 function buttons to trigger various MQTT commands for simplicity).

Note: MQTT mode is only available on port 1 which has a serial buffer of 512 bytes.

```
//==========================================================================
=
//              MQTT Example Macro for Zen/Zen Mini
//
//  26/7/2016
//  V1 R.Mulder
//
//  This macro was created as an example only to show the user how to
//  implement an MQTT mode on the Zen/Zen Mini controllers.
//
//
//
//
//
// Definitions for &MQTT_PACKET_CONTROL
//=====================================
//  This value is the first byte of every MQTT packet.
const connect=0x10
const publish=0x30
const pubAck=0x40
const subscribe=0x80
const unsubscribe=0xA0
const ping=0xC0
const disconnect=0xE0

// Definitions for PUBLISH packet
//===================================
// These flags are sent as part of the publish packet.
```

```
const retain=0x01                // instructs the server to retain data in between sessions
const qos0=0x00
const qos1=0x02
const qos2=0x04
const duplicate=0x08

// Print Destinations
//====================
const pubVarHdr=0x30
const pubTopic=0x34
const subQos0=0x80
const subQos1=0x84
const subQos2=0x88
const unsub=0xA0

//===========================================================================
=
//  Connection Requirements
//
//  The MQTT connect packet contains the following parts:
//    Protocol Name & level - set by firmware to MQTT V3.1.1
//    Connect Flags (specifying what type of parameters are included in the frame)
//    Keep Alive time
//    Client ID
//    Will Topic
//    Will message
//    Username
//    Password
//
//  Before attempting to connect to a message broker you first need to setup the
//  MQTT connection flags that will be used to determine what sort of connection
//  you require.
//
//  The flags are as follows:
//
//  bit 0 = reserved
const cleanSession=0x02          //  bit 1 = Clean Session
const willFlag=0x04              //  bit 2 = Will flag
//  bit 3-4 = Quality of Service
const willQos0=0x00              //          QoS0 = 00
const willQos1=0x08              //          QoS1 = 01
const willQos2=0x10              //          QoS2 = 10
const willRetain=0x20            //  bit 5 = Will Retain flag
const enablePassword=0x40        //  bit 6 = Password flag
const enableUsername=0x80        //  bit 7 = Username flag
//
//  The &MQTT_CONNECT_FLAGS register is stored in non-volatile memory so you
//  can set it up with a mem statement or from the macro.

mem &MQTT_CONNECT_FLAGS =cleanSession

//  The connect packet also specifies the keep alive time in seconds. If the
//  server does not receive a control packet from the client with in this
//  time it will close the connection. You should either publish data within
//  this time frame or send a PING request with int this time frame. The keep
//  alive value is a non-volatile register so it can be setup with a mem
//  statement as below or directly from the macro.

mem &MQTT_KEEP_ALIVE=60*10       // 10 minutes

//  If you enable the will flag, password flag or username flag above, then what
//  ever is enabled will printed from a text register.
```

```
//
// Note: Client ID, Will Topic, Will message, Username and  Password, all
// come from user definable text registers. You must set up the appropriate
// source registers for each of the ones you plan to use. See below.
//

mem &MQTT_CLIENT_ID_SOURCE=addr(&USER_TEXT1)
mem &USER_TEXT1="username"

mem &MQTT_WILL_TOPIC_SOURCE=addr(&USER_TEXT2)
mem &USER_TEXT2="myWillTopic"

mem &MQTT_WILL_MESSAGE_SOURCE=addr(&USER_TEXT3)
mem &USER_TEXT3="myWillMessage"

mem &MQTT_PASSWORD_SOURCE=addr(&PASSWORD1)
mem &USER_TEXT4="password"

mem &MQTT_USERNAME_SOURCE=addr(&USER_TEXT5)
mem &USER_TEXT5="username"

reg &topic1=&USER_TEXT6
mem &topic1="Topicone"

reg &topic2=&USER_TEXT7
mem &topic2="Topictwo"

reg &topicName= &USER_TEXT8
mem &topicName="topicname"

reg &topicMessage= &USER_TEXT9
mem &topicMessage="HelloWorld"

mem &SERIAL_MODE1=12    //MQTT mode


//=============================================================================
=
// Simple main macro to allow you to issue various commands and check an incoming publish
command from a
// subscribed topic.
main_macro:
  if |RECEIVE_READY1 =true then
    if serial_input 1 ="Topictwo" then
      &TEXT_VARIABLE1 = serial_pointer 1
      write "  __Topic2="+&TEXT_VARIABLE1+"             "
      &MQTT_PACKET_CONTROL =pubAck
    endif
    |RECEIVE_READY1=false
  endif
end

f1_button_macro:
  gosub connectToBroker
end

f2_button_macro:
  gosub pusblishData
end

f3_button_macro:
  gosub subscribeToTopic
```

```
end


//================================================================================
=
//  Connect to the message broker
//
connectToBroker:
//  Before attempting to connect to a message broker you first need to ensure all
//  MQTT connection parameters (i.e. connection flags, keep alive time,
//  Client ID, Will Topic, Will message, Username and  Password registers) are
//  setup correctly. See definitions above.

//  To send the connection request just load &MQTT_PACKET_CONTROL as below.
//  The firmware will send the MQTT connect packet and await the reply.

  &MQTT_PACKET_CONTROL =connect

//  You can test the |MQTT_CONNECT_OK flag to ensure the connection request is
//  successful.
return
//================================================================================
=

//================================================================================
=
//  Publish Data
//
pusblishData:
//  Before we publish any data we first need to load up DUP flag, QoS, and
//  RETAIN flag with our required settings.

    &MQTT_PACKET_CONTROL =qos2

//  The variable header for the publish command includes the topic name and the
//  packet ID. The topic name needs to be printed in one single print line with
//  the port destination set to "pubVarHdr". It can have constant text added
//  anywhere as shown below. The packet ID is added automatically by firmware at
//  at the end of the print.

    print pubVarHdr &topicName

//  Then the topic message can be printed. This can be printed with individual
//  print commands as shown below provided the total length of the packet does
//  not exceed 511 bytes.

    print pubTopic &topicMessage
    print pubTopic "End of topic message"

//  After all of the topic payload has been added to the buffer as shown above, now
//  we trigger the transmission of entire MQTT packet by loading the &MQTT_PACKET_CONTROL
//  register as shown below.

    &MQTT_PACKET_CONTROL =publish+qos2
return
//================================================================================
=


//================================================================================
=
```

```
// Subscribe to Topic
//
subscribeToTopic:
// When subscribing to a topic you must specify the QoS you require for that
// by printing to either subQos0, subQos1 ro subQos2 as shown below.
   print subQos1 &topic1

// You can subscribe to multiple topics at the same time provided the
// total length of the packet does not exceed 511 bytes. To subscribe to a
// second topic in the same packet add another topic as shown below.
   print subQos2 &topic2

// Load &MQTT_PACKET_CONTROL with "subscribe" to send the packet to the server.\

   &MQTT_PACKET_CONTROL =subscribe
return
//===============================================================================
=



//===============================================================================
=
// Unsubscribe from a Topic
//
unsubscribeFromTopic:
// To unsubscribe from a topic you just print to "unsub" and specify the
// topic name you want to unsubscribe from as shown below.
   print unsub &topic1

// You can unsubscribe from multiple topics at the same time provided the
// total length of the packet does not exceed 511 bytes. To unsubscribe from a
// second topic in the same packet add another command as shown below.
   print unsub &topic2

// Load &MQTT_PACKET_CONTROL with "subscribe" to send the packet to the server.\

   &MQTT_PACKET_CONTROL =unsubscribe
return
//===============================================================================
=



//===============================================================================
=
// Ping MQTT Server
//
pingServer:
   &MQTT_PACKET_CONTROL =ping
return
//===============================================================================
=



//===============================================================================
=
// Disconnect from MQTT Server
//
serverDisconnect:
   &MQTT_PACKET_CONTROL =disconnect
```

return
//===
=

2.11.10 Port 2

Name	Description	Symbol Type	Register Number	Memory Type
BAUDRATE2	8-bit register sets the serial port 2 baud rate (0-2 not available, 3 = 9600, 4 = 19.2 k, 5 = 38.4 k, 6 = 57.6 k, 7 = 115.2 k).	U_8	8208	RAM/EEPROM
SERIAL_BUFFER2	Start of serial transmit/receive buffer for port 2 (255 bytes long).	U_8	12545	RAM
RECEIVE_COUNT2	16-bit register which shows how many characters have been received by the serial port 2.	U_16	8455	RAM
TRANSMIT_COUNT2	8-bit register which sets how many characters are to transmitted by the serial port 2.	U_8	8466	RAM
RECEIVE_FLAGS2	8-bit register. Serial receive flags. Used in master mode.	U_8	8235	RAM
RECEIVE_READY2	This flag shows that a new message string has been received on port 2 in master mode.	B_0	8235	RAM
RECEIVE_RESULT2	32-bit register holds the 1st numeric value received in a string via serial port 2.	S_32	351	RAM
SERIAL_ADDRESS2	8-bit register holds the serial address of the controller.	U_8	8212	RAM/EEPROM
SERIAL_MODE2	8-bit register sets the serial mode for port 2.	U_8	8216	RAM/EEPROM
SERIAL_POINTER2	16-bit pointer used for string compare commands with serial port 2.	U_16	8459	RAM
START_OF_STRING_CHARACTER2	8-bit register. Sets ASCII character for the start of serial receive string in master mode for port 2.	U_8	8510	RAM/EEPROM
END_OF_STRING_CHARACTER2	8-bit register. Sets ASCII character for the end of serial receive string in master mode for port 2.	U_8	8227	RAM/EEPROM
STRING_LENGTH2	8-bit register. Sets string length of serial receive string in master mode for port 2.	U_8	8231	RAM/EEPROM
RECEIVE_IDLE_TIME2	16-bit register. Shows seconds of inactivity on serial port 2. (max. count = 255 seconds	U_16	4573	RAM
COMS_TIMEOUT2	16-bit register that specifies the timeout interval in seconds of inactivity on serial port 2. (Range 0 - 65535 seconds, 0=disabled). See Relay 2 RX Timeout)	U_16	4569	RAM/EEPROM

Serial Port 2 Modes - Registers 8216
Registers 8216 is an 8 bit register which controls the functionality of serial port 2. The following table shows the register value for currently available serial port protocols.

Value	Mode
0	ASCII
1	Modbus RTU slave
2	Macro Master mode.
3	Printer
4	Modbus/TCP wrap.
5	Intech/Modbus RTU slave. (default)
6	LCD touch panel
7	Modbus RTU Master
8	Bridge to port 1.
9	N/A - (bridge to port 2 for other serial ports)

10	Bridge to port 3.
11	N/A - (Ethernet IP on port 1 only - V0.08.01+)

Note: When set to Modbus RTU slave mode, any packets that do not comply with the Modbus format will be interrogated as possible ASCII packets (i.e. mode 0 above - this should not be confused with Modbus ASCII packets).

Baud Rates - Port 2
The baud rates for port 2 are controlled by bits 0 to 2 of register 8208 and the available options are shown below.

Bits 2, 1, 0	Baud rate
	000 = 2400 baud
	001 = 4800 baud
	010 = 9600 baud
	011 = 19200 baud
	100 = 38400 baud
	101 = 57600 baud
	110 = 115200 baud (default)
	111 = 230400 baud.

Number of Data Bits - Port 2
Bit 3 of register 8208 allows the user to select whether they require 7 or 8 bit data. This option is only available when the Macro Master mode protocol is selected.

Bit 3	Data bits
	0 = 8 data bits (default)
	1 = 7 data bits

Parity Setting - Port 2
Bits 4 & 5 of register 8208 allow the user to select a parity setting as per the options below.

Bits 5, 4	Parity
	00 = no parity (default)
	01 = odd parity
	10 = even parity

Transmit Delay - Port 2
To allow for slower devices, a transmit delay is available for most serial protocols. Bits 6 and 7 of register 8208 allow different transmit delay times to be selected. The various options available are shown as follows:

Bits 7, 6	Transmit delay
	00 = 2 milliseconds (default)
	01 = 20 milliseconds
	10 = 50 milliseconds
	11 = 100 milliseconds

2.11.10.1 Serial Buffer Port 2

Registers 12545 to 12800 are all 8-bit unsigned registers that are used as a buffer for serial port 2 received and transmitted data. They are used in all serial port modes, but their intended use is in master mode under macro control. By accessing these registers individually, a message string can be built up or interrogated, byte by byte.

NOTE: Although registers 12545 to 12800 can be written to, it is not recommended unless you have a thorough knowledge of how the serial port operates. Writing the wrong value to these registers could cause the serial port to lock up.

See Also
Serial Transmit Count - Registers 8465 - 8467

Serial Receive Count- Registers 8454 - 8456

2.11.11 Port 3

The serial port 3 output on the Zen IoT controller is via a quad 2.5mm jack socket or via the Bluetooth (BLE) module (if fitted). The output from the 2.5mm jack socket is just normal TTL levels (0 - 5volts). This is designed to work with the Define Instruments Ltd. USB to serial adapter to provide a low speed USB connection option. If the Bluetooth option is fitted to the Zen IoT then Bluetooth connectivity is broken whenever a 2.5 jack plug is inserted in the socket. If the 2.5mm jack plug is removed from the socket, Bluetooth functionality is resumed. Although port 3 is intended for device configuration and setup, all serial protocols are available for port 3. At power up port 3 always defaults to Modbus RTU slave mode at 19200,8,n,1. (See **Serial Port Modes** below).

Name	Description	Symbol Type	Register Number	Memory Type
BAUDRATE3	8-bit register sets the serial port 3 baud rate (0 = 1200, 1 = 2400, 2 = 4800, 3 = 9600, 4 = 19.2 k, 5 - 7 not available).	U_8	8209	RAM/EEPROM
SERIAL_BUFFER3	Start of serial transmit/receive buffer for port 3 (255 bytes long).	U_8	12801	RAM
RECEIVE_COUNT3	16-bit register which shows how many characters have been received by the serial port 3.	U_16	8456	RAM
TRANSMIT_COUNT3	8-bit register which sets how many characters are to transmitted by the serial port 3.	U_8	8467	RAM
RECEIVE_FLAGS3	8-bit register. Serial receive flags. Used in master mode.	U_8	8236	RAM
RECEIVE_READY3	This flag shows that a new message string has been received on port 3 in master mode.	B_0	8236	RAM
RECEIVE_RESULT3	32-bit register holds the 1st numeric value received in a string via serial port 3.	S_32	353	RAM
SERIAL_ADDRESS3	8-bit register holds the serial address of the controller.	U_8	8213	RAM/EEPROM
SERIAL_MODE3	8-bit register sets the serial mode for port 3.	U_8	8217	RAM/EEPROM
SERIAL_POINTER3	16-bit pointer used for string compare commands with serial port 3.	U_16	8460	RAM
START_OF_STRING_CHARACTER3	8-bit register. Sets ASCII character for the start of serial receive string in master mode for port 3.	U_8	8511	RAM/EEPROM
END_OF_STRING_CHARACTER3	8-bit register. Sets ASCII character for the end of serial receive string in master mode for port 3.	U_8	8228	RAM/EEPROM
STRING_LENGTH3	8-bit register. Sets string length of serial receive string in master mode for port 3.	U_8	8232	RAM/EEPROM
RECEIVE_IDLE_TIME3	16-bit register. Shows seconds of inactivity on serial port 3. (max. count = 255 seconds	U_16	4574	RAM
COMS_TIMEOUT3	16-bit register that specifies the timeout interval in seconds of inactivity on serial port 3. (Range 0 - 65535 seconds, 0=disabled). See Relay 2 RX Timeout)	U_16	4570	RAM/EEPROM

Serial Port Modes - Registers 8217
Registers 8217 is an 8 bit register which controls the functionality of serial port 3. The following table shows the register value for currently available serial port protocols.

Value	Mode
0	ASCII
1	Modbus RTU slave
2	Macro Master mode.
3	Printer
4	Modbus/TCP wrap.
5	Intech/Modbus RTU slave. (default)
6	LCD touch panel
7	Modbus RTU Master
8	Bridge to port 1.
9	Bridge to port 2.
10	N/A - (bridge to port 3 for other serial ports)
11	N/A - (Ethernet IP on port 1 only - V0.08.01+)

Note: When set to Modbus RTU slave mode, any packets that do not comply with the Modbus format will be interrogated as possible ASCII packets (i.e. mode 0 above - this should not be confused with Modbus ASCII packets).

Baud Rates - Port 3
The baud rates for port 3 are controlled by bits 0 to 2 of register 8209 and the available options are shown below.

Bits 2, 1, 0	Baud rate
	000 = 2400 baud
	001 = 4800 baud
	010 = 9600 baud (default)
	011 = 19200 baud
	100 = Not available (19200 baud)
	101 = Not available (19200 baud)
	110 = Not available (19200 baud)
	111 = Not available (19200 baud).

Number of Data Bits - Port 3
Bit 3 of register 8209 allows the user to select whether they require 7 or 8 bit data. This option is only available when the Macro Master mode protocol is selected.

Bit 3	Data bits
	0 = 8 data bits (default)
	1 = 7 data bits

Parity Setting - Port 3
Bits 4 & 5 of register 8209 allow the user to select a parity setting as per the options below.

Bits 5, 4	Parity
	00 = no parity (default)
	01 = odd parity
	10 = even parity

Transmit Delay - Port 3
To allow for slower devices, a transmit delay is available for most serial protocols. Bits 6 and 7 of register 8209 allow different transmit delay times to be selected. The various options available are shown as follows:

Bits 7, 6	Transmit delay
	00 = 2 milliseconds (default)
	01 = 20 milliseconds
	10 = 50 milliseconds
	11 = 100 milliseconds

2.11.11.1 Serial Buffer Port 3

Registers 12801 to 13056 are all 8-bit unsigned registers that are used as a buffer for serial port 3 received and transmitted data. They are used in all serial port modes, but their intended use is in master mode under macro control. By accessing these registers individually, a message string can be built up or interrogated, byte by byte.

NOTE: Although registers 12801 to 13056 can be written to, it is not recommended unless you have a thorough knowledge of how the serial port operates. Writing the wrong value to these registers could cause the serial port to lock up.

See Also
Serial Transmit Count - Registers 8465 - 8467

Serial Receive Count- Registers 8454 - 8456

2.12 Advanced Setpoints

Apart from the Controller Outputs, the Zen IoT controller also has 16 advanced setpoints. Each advanced setpoint includes the following features:

- Delay timers from setpoint activation - for automated make & break delays.
- Variable hysteresis operating in either control mode or alarm mode.
- Deviation mode.
- Relay output options including normal, 1 shot, pulse, repeat, -ve 1 shot, -ve pulse, -ve repeat.
- Power on inhibit modes.
- Relay latching and de-energizing options.
- PID.
- Register reset logic - allows automatic modification of register contents when a relay activates.
- Trigger logic (make, break, level) to activate a print or log sample.
- Setpoint tracking.

Note: Some of the features of the advanced setpoints can be used without relays being connected, however when relays are required, the relay IO module must be correctly configured to respond to the advanced setpoints. Please inform Define Instruments Ltd. when placing order.

Setpoint registers contain all individual setpoint activation, control, and setup information for the 6 advanced setpoints that are available for macro and front panel programming.

See
Setpoint 1
Setpoint 2
Setpoint 3
Setpoint 4
Setpoint 5
Setpoint 6
Setpoint 7
Setpoint 8
Setpoint 9
Setpoint 10
Setpoint 11
Setpoint 12
Setpoint 13
Setpoint 14
Setpoint 15
Setpoint 16
PID

See also
Setpoint Control Registers
Setpoint Latch Mask
Relay De-energize Mask
Setpoint Reset Delay (Power-On Inhibit)
Reset Destination
Setpoint Data Source Selection
Setpoint Tracking
Delay Type
Setpoint Trigger Functions
Setpoint Status Flags
Setpoint Trigger Flags
Setpoint Blanking

See Setpoints & Relays Supplement (NZ201) for a detailed description of the functionality of setpoints and relays.

2.12.1 Setpoint Control Registers

Registers 8245 to 8260
These are 8-bit registers used to control setpoint functionality. When reading or writing to these registers from the macro or via the serial port, the data is treated in octal format so that it is identical to the value shown on the display of the controller when setting these codes up manually. This allows 3 function groups to be controlled with one 3-digit number. The functional groups for the setpoint control registers are:

Display Digit	1st Digit	2nd Digit	3rd Digit
Function	Relay Energize Function	SP Activation Source	SP Functions

See graphic of setpoint 3-digit settings.

1st Digit - Relay Energize Function
The 1st digit of a setpoint control register (bits 6 & 7) controls when the relays energize in response to the input condition. The options available are:

0 = Energizes at or above setpoint value
1 = Energizes below setpoint value
2 = Energizes at or above setpoint value with falling input signal initial startup inhibit
3 = Energizes below setpoint value with rising input signal initial startup inhibit.

See detailed description of 1st digit Relay Energize Functions options.

2nd Digit - SP Activation Source
The 2nd digit of a setpoint control register (bits 3, 4, 5) selects the data source for the setpoint control logic. The options available are:

0 = Activate setpoint from selected source register
1 = Select source register for setpoint
2 = Activate setpoint from digital input source - DI_A Pin
3 = Activate setpoint from digital input source - DI_B Pin
4 = Activate setpoint from digital input source - DI_C Pin
5 = Activate setpoint from digital input source - DI_D Pin
6 = Reserved for future development.
7 = Reserved for future development.

See list of most commonly used named registers for the Setpoint Activation Source.

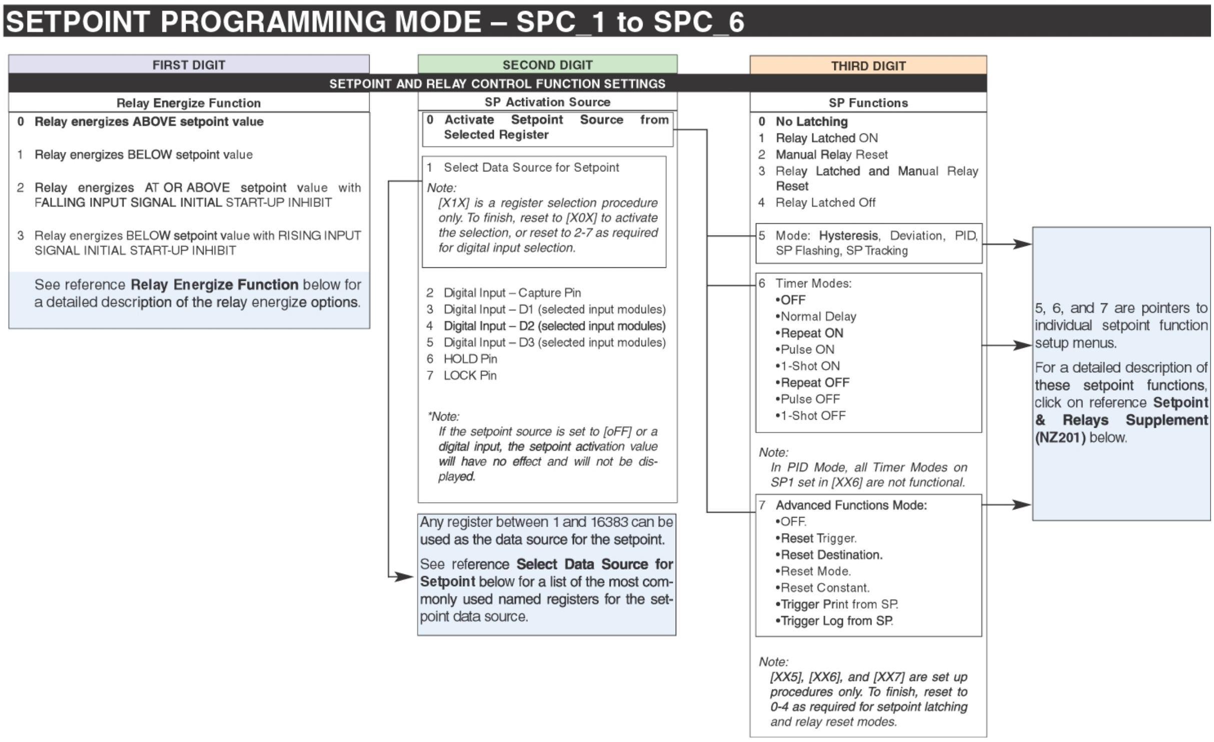
3rd Digit - SP Functions
The 3rd digit of a setpoint control register (bits 0,1,2) selects special setpoint functions and gives access to higher level setpoint functions from the display panel of the controller. The options available are:

0 = No latching
1 = Relay latched ON
2 = Manual relay reset
3 = Relay latched ON and manual relay reset
4 = Relay latched OFF
5 = Entry into Hysteresis, deviation and PID menus
6 = Entry into Timer menu
7 = Entry into trigger menu

See Setpoints & Relays Supplement (NZ201) for a detailed description of the functionality of setpoints and relays, including timer modes; reset and trigger modes; hysteresis, deviation, and PID modes; setpoint tracking and more.

See also
Octal Format

2.12.1.1 Setpoint 3-digit Graphic



2.12.1.1.1 Setpoint Latch Mask

This is a 16-bit register in RAM that controls the latching feature for the setpoints. If latching is selected for a setpoint and the appropriate bit of register 4100 is set, then the setpoint is latched (either above or below the setpoint value as selected). The setpoint can be unlatched by clearing the appropriate bit to zero. This holds true regardless of whether the latching is in the ON state or the OFF state. Unlatching the controller from the front panel buttons or from the LOCK or HOLD pin does exactly the same thing.

The function of each bit is as follows:

Bit	Name	Description	Function
0	SP1_LATCH	Flag shows/controls the latch status of setpoint 1.	0 = Setpoint unlatched 1 = Setpoint Latched
1	SP2_LATCH	Flag shows/controls the latch status of setpoint 2.	0 = Setpoint unlatched 1 = Setpoint Latched
2	SP3_LATCH	Flag shows/controls the latch status of setpoint 3.	0 = Setpoint unlatched 1 = Setpoint Latched
3	SP4_LATCH	Flag shows/controls the latch status of setpoint 4.	0 = Setpoint unlatched 1 = Setpoint Latched
4	SP5_LATCH	Flag shows/controls the latch status of setpoint 5.	0 = Setpoint unlatched 1 = Setpoint Latched
5	SP6_LATCH	Flag shows/controls the latch status of setpoint 6.	0 = Setpoint unlatched 1 = Setpoint Latched
6	SP7_LATCH	Flag shows/controls the latch status of setpoint 7.	0 = Setpoint unlatched 1 = Setpoint Latched
7	SP8_LATCH	Flag shows/controls the latch status of setpoint 8.	0 = Setpoint unlatched 1 = Setpoint Latched
8	SP9_LATCH	Flag shows/controls the latch status of setpoint 9.	0 = Setpoint unlatched 1 = Setpoint Latched
9	SP10_LATCH	Flag shows/controls the latch status of setpoint 10.	0 = Setpoint unlatched 1 = Setpoint Latched
10	SP11_LATCH	Flag shows/controls the latch status of setpoint 11.	0 = Setpoint unlatched 1 = Setpoint Latched
11	SP12_LATCH	Flag shows/controls the latch status of setpoint 12.	0 = Setpoint unlatched 1 = Setpoint Latched
12	SP13_LATCH	Flag shows/controls the latch status of setpoint 13.	0 = Setpoint unlatched 1 = Setpoint Latched
13	SP14_LATCH	Flag shows/controls the latch status of setpoint 14.	0 = Setpoint unlatched 1 = Setpoint Latched
14	SP15_LATCH	Flag shows/controls the latch status of setpoint 15.	0 = Setpoint unlatched 1 = Setpoint Latched
15	SP16_LATCH	Flag shows/controls the latch status of setpoint 16.	0 = Setpoint unlatched 1 = Setpoint Latched

2.12.1.2 Relay Energize Functions

Following is a detailed description of the options available on the 1st digit of the setpoint programming mode's setpoint control settings. Each description shows how the relay energize function operates when the setpoint has been set up for either hysteresis, deviation, or PID modes.

1st Digit Of Setpoint Control	Setpoint Mode	Hysteresis Type	Description
0	Normal	n/a	Relay energizes at or above the setpoint value.
	Hysteresis	Temperature	Cooling mode - Relay de-energizes at or below the setpoint value and energizes above (setpoint value + hysteresis value).
		Alarm	Relay energizes at or above the setpoint value and de-energizes below (setpoint value - hysteresis value).
	Deviation	n/a	Relay energizes inside the deviation band (setpoint value +/- deviation counts) and de-energizes outside the deviation band.
	PID	n/a	Controls above the setpoint value.
1	Normal	n/a	Relay energizes below the setpoint value.
	Hysteresis	Temperature	Heating mode - Relay de-energizes at or above the setpoint value and energizes below (setpoint value - hysteresis value).
		Alarm	Relay energizes at or below the setpoint value and de-energizes above (setpoint value + hysteresis value).
	Deviation	n/a	Relay energizes outside the deviation band (setpoint value +/- deviation counts) and de-energizes inside the deviation band.
	PID	n/a	Controls below the setpoint value.
2	Normal	n/a	Relay energizes at or above the setpoint value with falling input startup inhibit. (see note below on Falling Input Startup Inhibit)
	Hysteresis	Temperature	Cooling mode - Relay de-energizes at or below the setpoint value and energizes above (setpoint value + hysteresis value) with falling input startup inhibit. (see note below on Falling Input Startup Inhibit)
		Alarm	Relay energizes at or above the setpoint value and de-energizes below (setpoint value - hysteresis value) with falling input startup inhibit.
	Deviation	n/a	Relay energizes inside the deviation band (setpoint value +/- deviation counts) and de-energizes outside the deviation band with falling input startup inhibit. (see note below on Falling Input Startup Inhibit)
	PID	n/a	Controls above the setpoint value.
3	Normal	n/a	Relay energizes below the setpoint value with rising input startup inhibit.
	Hysteresis	Temperature	Heating mode - Relay de-energizes at or above the setpoint value and energizes below (setpoint value - hysteresis value) with rising input startup inhibit. (see note below on Rising Input Startup Inhibit)
		Alarm	Relay energizes at or below the setpoint value and de-energizes above (setpoint value + hysteresis value) with rising input startup inhibit. (see note below on Rising Input Startup Inhibit)
	Deviation	n/a	Relay energizes outside the deviation band (setpoint value +/- deviation counts) and de-energizes inside the deviation band with rising input startup inhibit. (see note below on Rising Input Startup Inhibit)
	PID	n/a	Controls below the setpoint value.

Falling Input Startup Inhibit

Falling input startup inhibit means that if the input signal is above the setpoint value at power up the relay will not be energized. The input signal must first fall below the setpoint value and rise again before the relay will be energized.

Rising Input Startup Inhibit

Rising input startup inhibit means that if the input signal is below the setpoint value at power up the relay will not be energized. The input signal must first rise above the setpoint value and fall again before the relay will be energized.

2.12.2 Relay De-energize Mask

This is a 16-bit register in RAM that controls the de-energizing feature for the relays. If the de-energize feature is selected for a setpoint, then when that setpoint is in its inactive state, the appropriate bit of register 4101 is set by the software. When the setpoint becomes active, register 4101 is used as a mask and the appropriate bit is ANDed with the relay output state. If the result is a **1**, then the relay is energized. If the appropriate bit of register 4101 is cleared to a **0** (while the setpoint is active), the relay is de-energized. As soon as the setpoint returns to its inactive state the appropriate bit of register 4101 is set to a **1** again. If a relay is de-energized by the front panel buttons, register 4101 is modified in the same way.

The function of each bit is shown as follows:

Bit	Name	Description	Function
0	RLY1_DE_ENERGISE	Flag shows/controls the de-energized status of relay 1.	0 = Relay De-energize (Activate) 1 = Relay De-energize (Inactive)
1	RLY2_DE_ENERGISE	Flag shows/controls the de-energized status of relay 2.	0 = Relay De-energize (Activate) 1 = Relay De-energize (Inactive)
2	RLY3_DE_ENERGISE	Flag shows/controls the de-energized status of relay 3.	0 = Relay De-energize (Activate) 1 = Relay De-energize (Inactive)
3	RLY4_DE_ENERGISE	Flag shows/controls the de-energized status of relay 4.	0 = Relay De-energize (Activate) 1 = Relay De-energize (Inactive)
4	RLY5_DE_ENERGISE	Flag shows/controls the de-energized status of relay 5.	0 = Relay De-energize (Activate) 1 = Relay De-energize (Inactive)
5	RLY6_DE_ENERGISE	Flag shows/controls the de-energized status of relay 6.	0 = Relay De-energize (Activate) 1 = Relay De-energize (Inactive)
6	RLY6_DE_ENERGISE	Flag shows/controls the de-energized status of relay 7.	0 = Relay De-energize (Activate) 1 = Relay De-energize (Inactive)
7	RLY6_DE_ENERGISE	Flag shows/controls the de-energized status of relay 8.	0 = Relay De-energize (Activate) 1 = Relay De-energize (Inactive)
8	RLY6_DE_ENERGISE	Flag shows/controls the de-energized status of relay 9.	0 = Relay De-energize (Activate) 1 = Relay De-energize (Inactive)
9	RLY6_DE_ENERGISE	Flag shows/controls the de-energized status of relay 10.	0 = Relay De-energize (Activate) 1 = Relay De-energize (Inactive)
10	RLY6_DE_ENERGISE	Flag shows/controls the de-energized status of relay 11.	0 = Relay De-energize (Activate) 1 = Relay De-energize (Inactive)
11	RLY6_DE_ENERGISE	Flag shows/controls the de-energized status of relay 12.	0 = Relay De-energize (Activate) 1 = Relay De-energize (Inactive)
12	RLY6_DE_ENERGISE	Flag shows/controls the de-energized status of relay 13.	0 = Relay De-energize (Activate) 1 = Relay De-energize (Inactive)
13	RLY6_DE_ENERGISE	Flag shows/controls the de-energized status of relay 14.	0 = Relay De-energize (Activate) 1 = Relay De-energize (Inactive)
14	RLY6_DE_ENERGISE	Flag shows/controls the de-energized status of relay 15.	0 = Relay De-energize (Activate) 1 = Relay De-energize (Inactive)
15	RLY6_DE_ENERGISE	Flag shows/controls the de-energized status of relay 16.	0 = Relay De-energize (Activate) 1 = Relay De-energize (Inactive)

2.12.3 Setpoint Reset Delay (Power-On Inhibit)

Register 4102 is a 16-bit register in RAM that contains flags for the reset delay function of the setpoints.

The function of each bit is as follows:

Bit	Name	Description	Function
0	POWERON_INHIBIT_SP1	Bit flag shows that setpoint 1 has been in-active since power-on.	0 = power-on inhibit ACTIVE 1 = power-on inhibit INACTIVE
1	POWERON_INHIBIT_SP2	Bit flag shows that setpoint 2 has been in-active since power-on.	0 = power-on inhibit ACTIVE 1 = power-on inhibit INACTIVE
2	POWERON_INHIBIT_SP3	Bit flag shows that setpoint 3 has been in-active since power-on.	0 = power-on inhibit ACTIVE 1 = power-on inhibit INACTIVE
3	POWERON_INHIBIT_SP4	Bit flag shows that setpoint 4 has been in-active since power-on.	0 = power-on inhibit ACTIVE 1 = power-on inhibit INACTIVE
4	POWERON_INHIBIT_SP5	Bit flag shows that setpoint 5 has been in-active since power-on.	0 = power-on inhibit ACTIVE 1 = power-on inhibit INACTIVE
5	POWERON_INHIBIT_SP6	Bit flag shows that setpoint 6 has been in-active since power-on.	0 = power-on inhibit ACTIVE 1 = power-on inhibit INACTIVE
6	POWERON_INHIBIT_SP7	Bit flag shows that setpoint 7 has been in-active since power-on.	0 = power-on inhibit ACTIVE 1 = power-on inhibit INACTIVE
7	POWERON_INHIBIT_SP8	Bit flag shows that setpoint 8 has been in-active since power-on.	0 = power-on inhibit ACTIVE 1 = power-on inhibit INACTIVE
8	POWERON_INHIBIT_SP9	Bit flag shows that setpoint 9 has been in-active since power-on.	0 = power-on inhibit ACTIVE 1 = power-on inhibit INACTIVE
9	POWERON_INHIBIT_SP10	Bit flag shows that setpoint 10 has been in-active since power-on.	0 = power-on inhibit ACTIVE 1 = power-on inhibit INACTIVE
10	POWERON_INHIBIT_SP11	Bit flag shows that setpoint 11 has been in-active since power-on.	0 = power-on inhibit ACTIVE 1 = power-on inhibit INACTIVE
11	POWERON_INHIBIT_SP12	Bit flag shows that setpoint 12 has been in-active since power-on.	0 = power-on inhibit ACTIVE 1 = power-on inhibit INACTIVE
12	POWERON_INHIBIT_SP13	Bit flag shows that setpoint 13 has been in-active since power-on.	0 = power-on inhibit ACTIVE 1 = power-on inhibit INACTIVE
13	POWERON_INHIBIT_SP14	Bit flag shows that setpoint 14 has been in-active since power-on.	0 = power-on inhibit ACTIVE 1 = power-on inhibit INACTIVE
14	POWERON_INHIBIT_SP15	Bit flag shows that setpoint 15 has been in-active since power-on.	0 = power-on inhibit ACTIVE 1 = power-on inhibit INACTIVE
15	POWERON_INHIBIT_SP16	Bit flag shows that setpoint 16 has been in-active since power-on.	0 = power-on inhibit ACTIVE 1 = power-on inhibit INACTIVE

After power on, register 4102 is initially cleared to zero. As each setpoint is examined, the appropriate bit of register 4102 is set only if the setpoint is inactive. A setpoint that has the reset delay feature selected can only activate the relay if the appropriate power-on inhibit bit is set. This means that after reset, the setpoint must first enter the in-active state before it can be activated.

This register can be read or written to.

2.12.4 Reset Destination

Registers 4191 to 4206 are 16-bit registers that specify the destination register that will be modified by each setpoint reset function.

See Also
Common Reset Destination Registers

2.12.5 Setpoint Data Source Selection

Registers 4338 to 4353 are 16-bit registers that specify the data source for the setpoints. The number they contain is the ASCII/Modbus register number for the required data source.

NOTE: Only registers that hold integer values can be used as a data source for the display. Floating point and text registers can not be used.

See Also
Common Data Source Registers

2.12.6 Setpoint Tracking

Registers 8261 to 8276 are 8-bit registers used for selecting setpoint tracking. The function of each bit is as follows:

Bit Position								Description
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
			0	0	0	0	0	Tracking disabled
			0	0	0	0	1	Setpoint tracks SP1
			0	0	0	1	0	Setpoint tracks SP2
			0	0	0	1	1	Setpoint tracks SP3
			0	0	1	0	0	Setpoint tracks SP4
			0	0	1	0	1	Setpoint tracks SP5
			0	0	1	1	0	Setpoint tracks SP6
			0	0	1	1	1	Setpoint tracks SP7
			0	1	0	0	0	Setpoint tracks SP8
			0	1	0	0	1	Setpoint tracks SP9
			0	1	0	1	0	Setpoint tracks SP10
			0	1	0	1	1	Setpoint tracks SP11
			0	1	1	0	0	Setpoint tracks SP12
			0	1	1	0	1	Setpoint tracks SP13
			0	1	1	1	0	Setpoint tracks SP14
			0	1	1	1	1	Setpoint tracks SP15
			1	0	0	0	0	Setpoint tracks SP16

2.12.7 Delay Type

Registers 8277 to 8292 are 8-bit registers used to control the delay type, display flashing, and mode of each setpoint SP1 to SP16.

The function of each bit is as follows:

Bits 0 to 2: Delay type

Bit Position								Description
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
					0	0	0	OFF
					0	0	1	Normal
					0	1	0	1-Shot
					0	1	1	Pulse
					1	0	0	Repeat
					1	0	1	Negative 1-Shot
					1	1	0	Negative Pulse
					1	1	1	Negative Repeat

Bit 3: Display flash on setpoint
0 = no flash
1 = flash on setpoint active

Bit 4, 5: Hysteresis / Deviation / PID mode

Bit Position								Description
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
		0	0					OFF
		0	1					Hysteresis
		1	0					Deviation
		1	1					PID

Bit 6 **Delay resolution**
0 = 0.1 second resolution
1 = 1 millisecond resolution

Bit 7 **Hysteresis Type**
0 = Temperature control
1 = Alarm
(Note: see Hysteresis Type for a full explanation of options)

2.12.8 Hysteresis Type

When a setpoint is operated in Hysteresis mode, two types of hysteresis action can be selected depending on the application. The hysteresis can be set to operate in a manner suitable for temperature control applications or it can be set to operate for use with alarms. A more detailed explanation of each mode is given below.

1st Digit Of Setpoint Control	Hysteresis Type	Description
0, 2	Control	Cooling mode - Relay de-energises at or below the setpoint value and energises above (setpoint value + hysteresis value).
	Alarm	Relay energises at or above the setpoint value and de-energises below (setpoint value - hysteresis value).
1, 3	Control	Heating mode - Relay de-energises at or above the setpoint value and energises below (setpoint value - hysteresis value).
	Alarm	Relay energises at or below the setpoint value and de-energises above (setpoint value + hysteresis value).

See Also
Delay Type - Register 8277 to 8292

2.12.9 Setpoint Trigger Functions

Registers 8293 to 8308 are 8-bit registers used for selecting the setpoint trigger functions on the following setpoints.

The function of each bit is as follows:

Bits 0 to 2: Trigger functions

Bit Position								Description
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
					0	0	0	All trigger functions disabled
					0	0	1	Trigger on make edge
					0	1	0	Trigger on break edge
					0	1	1	Trigger on both make & break edge
					1	0	0	Trigger when energized

Bit 3 & 4: Reset mode

Bit Position								Description
7	6	5	4	3	2	1	0	
			0	0				Destination register = user defined constant
			0	1				Destination register = Input_data - Setpoint_value+Constant
			1	0				Destination register = Destination + Constant
			1	1				Destination register = Source register

Bit 5: **Reserved for future development**

Bit 6: **Log on selected edge**
0 = no log
1 = log

Bit 7: **Print on selected edge**
0 = no print
1 = print

2.12.10 Setpoint Status Flags

Register 4097 is a single 16-bit read only register that contains 16 flags showing the status of setpoints 1 to 16 in normal and remote mode. It differs from the ALARM_STATUS register (239) which allows the setpoints to be remotely controlled as well. If a setpoint is forced into remote mode by a write to register 239, the setpoint will no longer respond to input changes based on the setpoint logic but will now only change state when 239 is written. However, the setpoint logic (i.e. comparison between the setpoint activation value and the data input value) is still operational in the background even though it is not used. Register 4097 displays the status of the comparison at this point before the final control is diverted to remote mode. It includes features such hysteresis, deviation, and setpoint tracking.

This can be useful in a macro that needs to control the relay in a special way and still use one of the above standard features.

The function of each bit is as follows:

Bit	Name	Description	Function
0	SP1_STATUS	Read only flag shows the status of setpoint 1 in normal & remote mode.	(1 = setpoint activated)
1	SP2_STATUS	Read only flag shows the status of setpoint 2 in normal & remote mode.	(1 = setpoint activated)
2	SP3_STATUS	Read only flag shows the status of setpoint 3 in normal & remote mode.	(1 = setpoint activated)
3	SP4_STATUS	Read only flag shows the status of setpoint 4 in normal & remote mode.	(1 = setpoint activated)
4	SP5_STATUS	Read only flag shows the status of setpoint 5 in normal & remote mode.	(1 = setpoint activated)
5	SP6_STATUS	Read only flag shows the status of setpoint 6 in normal & remote mode..	(1 = setpoint activated)
6	SP7_STATUS	Read only flag shows the status of setpoint 7 in normal & remote mode..	(1 = setpoint activated)
7	SP8_STATUS	Read only flag shows the status of setpoint 8 in normal & remote mode..	(1 = setpoint activated)
8	SP9_STATUS	Read only flag shows the status of setpoint 9 in normal & remote mode..	(1 = setpoint activated)
9	SP10_STATUS	Read only flag shows the status of setpoint 10 in normal & remote mode..	(1 = setpoint activated)
10	SP11_STATUS	Read only flag shows the status of setpoint 11 in normal & remote mode..	(1 = setpoint activated)
11	SP12_STATUS	Read only flag shows the status of setpoint 12 in normal & remote mode..	(1 = setpoint activated)
12	SP13_STATUS	Read only flag shows the status of setpoint 13 in normal & remote mode..	(1 = setpoint activated)
13	SP14_STATUS	Read only flag shows the status of setpoint 14 in normal & remote mode..	(1 = setpoint activated)
14	SP15_STATUS	Read only flag shows the status of setpoint 15 in normal & remote mode..	(1 = setpoint activated)
15	SP16_STATUS	Read only flag shows the status of setpoint 16 in normal & remote mode..	(1 = setpoint activated)

See also
ALARM_STATUS

2.12.11 Setpoint Trigger Flags

Register 4098 is a single 16-bit read only register that contains 16 flags showing the trigger status for each of the 16 setpoints. Each flag is set if the trigger condition selected for that setpoint (i.e. make, break, both, level) is satisfied, and cleared if the trigger condition is false.

The function of each bit is as follows:

Bit	Name	Description	Function
0	TRIGGER1	Read only flag shows the trigger status for setpoint 1.	0 = Trigger INACTIVE 1 = Trigger ACTIVATED
1	TRIGGER2	Read only flag shows the trigger status for setpoint 2.	0 = Trigger INACTIVE 1 = Trigger ACTIVATED
2	TRIGGER3	Read only flag shows the trigger status for setpoint 3.	0 = Trigger INACTIVE 1 = Trigger ACTIVATED
3	TRIGGER4	Read only flag shows the trigger status for setpoint 4.	0 = Trigger INACTIVE 1 = Trigger ACTIVATED
4	TRIGGER5	Read only flag shows the trigger status for setpoint 5.	0 = Trigger INACTIVE 1 = Trigger ACTIVATED
5	TRIGGER6	Read only flag shows the trigger status for setpoint 6.	0 = Trigger INACTIVE 1 = Trigger ACTIVATED
6	TRIGGER7	Read only flag shows the trigger status for setpoint 7.	0 = Trigger INACTIVE 1 = Trigger ACTIVATED
7	TRIGGER8	Read only flag shows the trigger status for setpoint 8.	0 = Trigger INACTIVE 1 = Trigger ACTIVATED
8	TRIGGER9	Read only flag shows the trigger status for setpoint 9.	0 = Trigger INACTIVE 1 = Trigger ACTIVATED
9	TRIGGER10	Read only flag shows the trigger status for setpoint 10.	0 = Trigger INACTIVE 1 = Trigger ACTIVATED
10	TRIGGER11	Read only flag shows the trigger status for setpoint 11.	0 = Trigger INACTIVE 1 = Trigger ACTIVATED
11	TRIGGER12	Read only flag shows the trigger status for setpoint 12.	0 = Trigger INACTIVE 1 = Trigger ACTIVATED
12	TRIGGER13	Read only flag shows the trigger status for setpoint 13.	0 = Trigger INACTIVE 1 = Trigger ACTIVATED
13	TRIGGER14	Read only flag shows the trigger status for setpoint 14.	0 = Trigger INACTIVE 1 = Trigger ACTIVATED
14	TRIGGER15	Read only flag shows the trigger status for setpoint 15.	0 = Trigger INACTIVE 1 = Trigger ACTIVATED
15	TRIGGER16	Read only flag shows the trigger status for setpoint 16.	0 = Trigger INACTIVE 1 = Trigger ACTIVATED

NOTE: These flags only remain **alive** for one output cycle following the active edge which caused the trigger. They are intended to be used by the macro.

2.12.12 Setpoint 1

Name	Description	Symbol Type	Register Number	Memory Type
SETPOINT1	32-bit register for setpoint 1 value.	S_32	257	RAM/EEPROM
SETPOINT1_FLOAT	32-bit pseudo floating point register for setpoint 1 value. (See 32-bit Pseudo Floating Point).	PF_32	1537	RAM/EEPROM
SP1	This flag shows/controls the status of setpoint 1 (ON = setpoint activated).	B_0	239	RAM
SP1_REMOTE	Setting this bit to ON places setpoint 1 in remote mode.	B_16	239	RAM
SP1_STATUS	Read only flag shows the status of setpoint 1 in normal & remote mode.	B_0_R	4097	RAM
TRIGGER1	8-bit read only register which contains status flags for the Modbus master macro.	B_0_R	4098	RAM
RELAY1	Flag which shows/controls the instantaneous state of relay 1 (ON=energized).	B_0	4099	RAM

See also

Setpoint Status Flags - Register 4097

Setpoint Trigger Flags - Register 4098

Register 239 - Alarm Status

SP1 Setup

2.12.12.1 SP1 Setup

Name	Description	Symbol Type	Register Number	Memory Type
POWERON_INHIBIT_SP1	Bit flag shows that setpoint 1 has been in-active since power-on.	B_0	4102	RAM
RLY1_DE_ENERGISE	Bit flag shows that setpoint 1 has been in-active since power-on.	B_0	4101	RAM
SETPOINT1_TEXT	Text display for setpoint 1.	L_14_T	16495	EEPROM
SP1_BREAK_DELAY	16-bit register holds the break delay time for setpoint 1 (0.1s or 0.001s resolution).	U_16	4175	RAM/EEPROM
SP1_CONTROL	8-bit register holds the setpoint & relay control setting for setpoint 1(note: controller display is in octal).	O_8	8245	RAM/EEPROM
SP1_DATA_SOURCE	16-bit register holds the register number of the data source for setpoint 1.	U_16	4338	RAM/EEPROM
SP1_DELAY_TYPE	8-bit register controls the delay type settings for setpoint 1.	U_8	8277	RAM/EEPROM
SP1_HYST	16-bit register holds the hysteresis/passband value for setpoint 1.	U_16	4143	RAM/EEPROM
SP1_LATCH	Flag shows/controls the latch status of setpoint 1(ON = setpoint latched).	B_0	4100	RAM
SP1_MAKE_DELAY	16-bit register holds the make delay time for setpoint 1 (0.1s or 0.001s resolution).	U_16	4159	RAM/EEPROM
SP1_RESET_DESTINATION	8-bit register holds the destination register number for setpoint 1 trigger functions.	U_8	4191	RAM/EEPROM
SP1_RESET_VALUE	32-bit register holds the reset value used with setpoint 1 trigger functions.	S_32	429	RAM/EEPROM
SP1_TRACKING	8-bit register controls the setpoint tracking for setpoint 1.	U_8	8261	RAM/EEPROM
SP1_TRIGGER	8-bit register. Controls trigger functions of setpoint 1.	U_8	8293	RAM/EEPROM

See also
Setpoint Latch Mask - Register 4100

Relay De-energize Mask - Register 4101

Setpoint Reset Delay (Power-On Inhibit) - Register 4102

PID

PID 1

Octal Format

2.12.13 Setpoint 2

Name	Description	Symbol Type	Register Number	Memory Type
SETPOINT2	32-bit register for setpoint 2 value.	S_32	259	RAM/EEPROM
SETPOINT2_FLOAT	32-bit pseudo floating point register for setpoint 2 value. (See 32-bit Pseudo Floating Point).	PF_32	1539	RAM/EEPROM
SP2	This flag shows/controls the status of setpoint 2 (ON = setpoint activated).	B_1	239	RAM
SP2_REMOTE	Setting this bit to ON places setpoint 2 in remote mode.	B_17	239	RAM
SP2_STATUS	Read only flag shows the status of setpoint 2 in normal & remote mode.	B_1_R	4097	RAM
TRIGGER2	8-bit read only register which contains status flags for the Modbus master macro.	B_1_R	4098	RAM
RELAY2	Flag which shows/controls the instantaneous state of relay 2 (ON=energized).	B_1	4099	RAM

See also

Setpoint Status Flags - Register 4097

Setpoint Trigger Flags - Register 4098

Register 239 - Alarm Status

SP2 Setup

2.12.13.1 SP2 Setup

Name	Description	Symbol Type	Register Number	Memory Type
POWERON_INHIBIT_SP2	Bit flag shows that setpoint 2 has been in-active since power-on.	B_1	4102	RAM
RLY2_DE_ENERGISE	Bit flag shows that setpoint 2 has been in-active since power-on.	B_1	4101	RAM
SETPOINT2_TEXT	Text display for setpoint 2.	L_14_T	16497	EEPROM
SP2_BREAK_DELAY	16-bit register holds the break delay time for setpoint 2 (0.1s or 0.001s resolution).	U_16	4176	RAM/EEPROM
SP2_CONTROL	8-bit register holds the setpoint & relay control setting for setpoint 2 (note: controller display is in octal).	O_8	8246	RAM/EEPROM
SP2_DATA_SOURCE	16-bit register holds the register number of the data source for setpoint 2.	U_16	4339	RAM/EEPROM
SP2_DELAY_TYPE	8-bit register controls the delay type settings for setpoint 2.	U_8	8278	RAM/EEPROM
SP2_HYST	16-bit register holds the hysteresis/passband value for setpoint 2.	U_16	4144	RAM/EEPROM
SP2_LATCH	Flag shows/controls the latch status of setpoint 2 (ON = setpoint latched).	B_1	4100	RAM
SP2_MAKE_DELAY	16-bit register holds the make delay time for setpoint 2 (0.1s or 0.001s resolution).	U_16	4160	RAM/EEPROM

SP2_RESET_DESTINATION	8-bit register holds the destination register number for setpoint 2 trigger functions.	U_8	4192	RAM/EEPROM
SP2_RESET_VALUE	32-bit register holds the reset value used with setpoint 2 trigger functions.	S_32	431	RAM/EEPROM
SP2_TRACKING	8-bit register controls the setpoint tracking for setpoint 2.	U_8	8262	RAM/EEPROM
SP2_TRIGGER	8-bit register. Controls trigger functions of setpoint 2.	U_8	8294	RAM/EEPROM

See also

Setpoint Latch Mask - Register 4100

Relay De-energize Mask - Register 4101

Setpoint Reset Delay (Power-On Inhibit) - Register 4102

PID

PID2

Octal Format

2.12.14 Setpoint 3

Name	Description	Symbol Type	Register Number	Memory Type
SETPOINT3	32-bit register for setpoint 3 value.	S_32	261	RAM/EEPROM
SETPOINT3_FLOAT	32-bit pseudo floating point register for setpoint 3 value. (See 32-bit Pseudo Floating Point).	PF_32	1541	RAM/EEPROM
SP3	This flag shows/controls the status of setpoint 3 (ON = setpoint activated).	B_2	239	RAM
SP3_REMOTE	Setting this bit to ON places setpoint 3 in remote mode.	B_18	239	RAM
SP3_STATUS	Read only flag shows the status of setpoint 3 in normal & remote mode.	B_2_R	4097	RAM
TRIGGER3	8-bit read only register which contains status flags for the Modbus master macro.	B_2_R	4098	RAM
RELAY3	Flag which shows/controls the instantaneous state of relay 3 (ON=energized).	B_2	4099	RAM

See also

Setpoint Status Flags - Register 4097

Setpoint Trigger Flags - Register 4098

Register 239 - Alarm Status

SP3 Setup

2.12.14.1 SP3 Setup

Name	Description	Symbol Type	Register Number	Memory Type
POWERON_INHIBIT_SP3	Bit flag shows that setpoint 3 has been in-active since power-on.	B_2	4102	RAM
RLY3_DE_ENERGISE	Bit flag shows that setpoint 3 has been in-active since power-on.	B_2	4101	RAM
SETPOINT3_TEXT	Text display for setpoint 3.	L_14_T	16499	EEPROM
SP3_BREAK_DELAY	16-bit register holds the break delay time for setpoint 3 (0.1s or 0.001s resolution).	U_16	4177	RAM/EEPROM
SP3_CONTROL	8-bit register holds the setpoint & relay control setting for setpoint 3 (note: controller display is in octal).	O_8	8247	RAM/EEPROM
SP3_DATA_SOURCE	16-bit register holds the register number of the data source for setpoint 3.	U_16	4340	RAM/EEPROM
SP3_DELAY_TYPE	8-bit register controls the delay type settings for setpoint 3.	U_8	8279	RAM/EEPROM
SP3_HYST	16-bit register holds the hysteresis/passband value for setpoint 3.	U_16	4145	RAM/EEPROM
SP3_LATCH	Flag shows/controls the latch status of setpoint 3 (ON = setpoint latched).	B_2	4100	RAM
SP3_MAKE_DELAY	16-bit register holds the make delay time for setpoint 3 (0.1s or 0.001s resolution).	U_16	4161	RAM/EEPROM
SP3_RESET_DESTINATION	8-bit register holds the destination register number for setpoint 3 trigger functions.	U_8	4193	RAM/EEPROM
SP3_RESET_VALUE	32-bit register holds the reset value used with setpoint 3 trigger functions.	S_32	433	RAM/EEPROM
SP3_TRACKING	8-bit register controls the setpoint tracking for setpoint 3.	U_8	8263	RAM/EEPROM
SP3_TRIGGER	8-bit register. Controls trigger functions of setpoint 3.	U_8	8295	RAM/EEPROM

See also

Setpoint Latch Mask - Register 4100

Relay De-energize Mask - Register 4101

Setpoint Reset Delay (Power-On Inhibit) - Register 4102

PID

PID 3

Octal Format

2.12.15 Setpoint 4

Name	Description	Symbol Type	Register Number	Memory Type
SETPOINT4	32-bit register for setpoint 4 value.	S_32	263	RAM/EEPROM
SETPOINT4_FLOAT	32-bit pseudo floating point register for setpoint 4 value. (See 32-bit Pseudo Floating Point).	PF_32	1543	RAM/EEPROM
SP4	This flag shows/controls the status of setpoint 4 (ON = setpoint activated).	B_3	239	RAM
SP4_REMOTE	Setting this bit to ON places setpoint 4 in remote mode.	B_19	239	RAM
SP4_STATUS	Read only flag shows the status of setpoint 4 in normal & remote mode.	B_3_R	4097	RAM
TRIGGER4	8-bit read only register which contains status flags for the Modbus master macro.	B_3_R	4098	RAM
RELAY4	Flag which shows/controls the instantaneous state of relay 4 (ON=energized).	B_3	4099	RAM

See also

Setpoint Status Flags - Register 4097

Setpoint Trigger Flags - Register 4098

Register 239 - Alarm Status

SP4 Setup

2.12.15.1 SP4 Setup

Name	Description	Symbol Type	Register Number	Memory Type
POWERON_INHIBIT_SP4	Bit flag shows that setpoint 4 has been in-active since power-on.	B_3	4102	RAM
RLY4_DE_ENERGISE	Bit flag shows that setpoint 4 has been in-active since power-on.	B_3	4101	RAM
SETPOINT4_TEXT	Text display for setpoint 4.	L_14_T	16501	EEPROM
SP4_BREAK_DELAY	16-bit register holds the break delay time for setpoint 4 (0.1s or 0.001s resolution).	U_16	4178	RAM/EEPROM
SP4_CONTROL	8-bit register holds the setpoint & relay control setting for setpoint 4 (note: controller display is in octal).	O_8	8248	RAM/EEPROM
SP4_DATA_SOURCE	16-bit register holds the register number of the data source for setpoint 4.	U_16	4341	RAM/EEPROM
SP4_DELAY_TYPE	8-bit register controls the delay type settings for setpoint 4.	U_8	8280	RAM/EEPROM
SP4_HYST	16-bit register holds the hysteresis/passband value for setpoint 4.	U_16	4146	RAM/EEPROM
SP4_LATCH	Flag shows/controls the latch status of setpoint 4 (ON = setpoint latched).	B_3	4100	RAM
SP4_MAKE_DELAY	16-bit register holds the make delay time for setpoint 4 (0.1s or 0.001s resolution).	U_16	4162	RAM/EEPROM

SP4_RESET_DESTINATION	8-bit register holds the destination register number for setpoint 4 trigger functions.	U_8	4194	RAM/EEPROM
SP4_RESET_VALUE	32-bit register holds the reset value used with setpoint 4 trigger functions.	S_32	435	RAM/EEPROM
SP4_TRACKING	8-bit register controls the setpoint tracking for setpoint 4.	U_8	8264	RAM/EEPROM
SP4_TRIGGER	8-bit register. Controls trigger functions of setpoint 4.	U_8	8296	RAM/EEPROM

See also

Setpoint Latch Mask - Register 4100

Relay De-energize Mask - Register 4101

Setpoint Reset Delay (Power-On Inhibit) - Register 4102

PID

PID 4

Octal Format

2.12.16 Setpoint 5

Name	Description	Symbol Type	Register Number	Memory Type
SETPOINT5	32-bit register for setpoint 5 value.	S_32	265	RAM/EEPROM
SETPOINT5_FLOAT	32-bit pseudo floating point register for setpoint 5 value. (See 32-bit Pseudo Floating Point).	PF_32	1545	RAM/EEPROM
SP5	This flag shows/controls the status of setpoint 5 (ON = setpoint activated).	B_4	239	RAM
SP5_REMOTE	Setting this bit to ON places setpoint 5 in remote mode.	B_20	239	RAM
SP5_STATUS	Read only flag shows the status of setpoint 5 in normal & remote mode.	B_4_R	4097	RAM
TRIGGER5	8-bit read only register which contains status flags for the Modbus master macro.	B_4_R	4098	RAM
RELAY5	Flag which shows/controls the instantaneous state of relay 5 (ON=energized).	B_4	4099	RAM

See also

Setpoint Status Flags - Register 4097

Setpoint Trigger Flags - Register 4098

Register 239 - Alarm Status

SP5 Setup

2.12.16.1 SP5 Setup

Name	Description	Symbol Type	Register Number	Memory Type
POWERON_INHIBIT_SP5	Bit flag shows that setpoint 5 has been in-active since power-on.	B_4	4102	RAM
RLY5_DE_ENERGISE	Bit flag shows that setpoint 5 has been in-active since power-on.	B_4	4101	RAM
SETPOINT5_TEXT	Text display for setpoint 5.	L_14_T	16503	EEPROM
SP5_BREAK_DELAY	16-bit register holds the break delay time for setpoint 5 (0.1s or 0.001s resolution).	U_16	4179	RAM/EEPROM
SP5_CONTROL	8-bit register holds the setpoint & relay control setting for setpoint 5 (note: controller display is in octal).	O_8	8249	RAM/EEPROM
SP5_DATA_SOURCE	16-bit register holds the register number of the data source for setpoint 5.	U_16	4342	RAM/EEPROM
SP5_DELAY_TYPE	8-bit register controls the delay type settings for setpoint 5.	U_8	8281	RAM/EEPROM
SP5_HYST	16-bit register holds the hysteresis/passband value for setpoint 5.	U_16	4147	RAM/EEPROM
SP5_LATCH	Flag shows/controls the latch status of setpoint 5 (ON = setpoint latched).	B_4	4100	RAM
SP5_MAKE_DELAY	16-bit register holds the make delay time for setpoint 5 (0.1s or 0.001s resolution).	U_16	4163	RAM/EEPROM
SP5_RESET_DESTINATION	8-bit register holds the destination register number for setpoint 5 trigger functions.	U_8	4195	RAM/EEPROM
SP5_RESET_VALUE	32-bit register holds the reset value used with setpoint 5 trigger functions.	S_32	437	RAM/EEPROM
SP5_TRACKING	8-bit register controls the setpoint tracking for setpoint 5.	U_8	8265	RAM/EEPROM
SP5_TRIGGER	8-bit register. Controls trigger functions of setpoint 5.	U_8	8297	RAM/EEPROM

See also

Setpoint Latch Mask - Register 4100

Relay De-energize Mask - Register 4101

Setpoint Reset Delay (Power-On Inhibit) - Register 4102

PID

PID 5

Octal Format

2.12.17 Setpoint 6

Name	Description	Symbol Type	Register Number	Memory Type
SETPOINT6	32-bit register for setpoint 6 value.	S_32	267	RAM/EEPROM
SETPOINT6_FLOAT	32-bit pseudo floating point register for setpoint 6 value. (See 32-bit Pseudo Floating Point).	PF_32	1547	RAM/EEPROM
SP6	This flag shows/controls the status of setpoint 6 (ON = setpoint activated).	B_5	239	RAM
SP6_REMOTE	Setting this bit to ON places setpoint 6 in remote mode.	B_21	239	RAM
SP6_STATUS	Read only flag shows the status of setpoint 6 in normal & remote mode.	B_5_R	4097	RAM
TRIGGER6	8-bit read only register which contains status flags for the Modbus master macro.	B_5_R	4098	RAM
RELAY6	Flag which shows/controls the instantaneous state of relay 6 (ON=energized).	B_5	4099	RAM

See also

Setpoint Status Flags - Register 4097

Setpoint Trigger Flags - Register 4098

Register 239 - Alarm Status

SP6 Setup

Register 4099 - Relay output

2.12.17.1 SP6 Setup

Name	Description	Symbol Type	Register Number	Memory Type
POWERON_INHIBIT_SP6	Bit flag shows that setpoint 6 has been in-active since power-on.	B_5	4102	RAM
RLY6_DE_ENERGISE	Bit flag shows that setpoint 6 has been in-active since power-on.	B_5	4101	RAM
SETPOINT6_TEXT	Text display for setpoint 6.	L_14_T	16505	EEPROM
SP6_BREAK_DELAY	16-bit register holds the break delay time for setpoint 6 (0.1s or 0.001s resolution).	U_16	4180	RAM/EEPROM
SP6_CONTROL	8-bit register holds the setpoint & relay control setting for setpoint 6 (note: controller display is in octal).	O_8	8250	RAM/EEPROM
SP6_DATA_SOURCE	16-bit register holds the register number of the data source for setpoint 6.	U_16	4343	RAM/EEPROM
SP6_DELAY_TYPE	8-bit register controls the delay type settings for setpoint 6.	U_8	8282	RAM/EEPROM
SP6_HYST	16-bit register holds the hysteresis/passband value for setpoint 6.	U_16	4148	RAM/EEPROM
SP6_LATCH	Flag shows/controls the latch status of setpoint 6 (ON = setpoint latched).	B_5	4100	RAM
SP6_MAKE_DELAY	16-bit register holds the make delay time for setpoint 6 (0.1s or 0.001s resolution).	U_16	4164	RAM/EEPROM

SP6_RESET_DESTINATION	8-bit register holds the destination register number for setpoint 6 trigger functions.	U_8	4196	RAM/EEPROM
SP6_RESET_VALUE	32-bit register holds the reset value used with setpoint 6 trigger functions.	S_32	439	RAM/EEPROM
SP6_TRACKING	8-bit register controls the setpoint tracking for setpoint 6.	U_8	8266	RAM/EEPROM
SP6_TRIGGER	8-bit register. Controls trigger functions of setpoint 6.	U_8	8298	RAM/EEPROM

See also
Setpoint Latch Mask - Register 4100

Relay De-energize Mask - Register 4101

Setpoint Reset Delay (Power-On Inhibit) - Register 4102

Octal Format

2.12.18 Setpoint 7

Name	Description	Symbol Type	Register Number	Memory Type
SETPOINT7	32-bit register for setpoint 7 value.	S_32	269	RAM/EEPROM
SETPOINT7_FLOAT	32-bit pseudo floating point register for setpoint 7 value. (See 32-bit Pseudo Floating Point).	PF_32	1549	RAM/EEPROM
SP7	This flag shows/controls the status of setpoint 7 (ON = setpoint activated).	B_6	239	RAM
SP7_REMOTE	Setting this bit to ON places setpoint 7 in remote mode.	B_22	239	RAM
SP7_STATUS	Read only flag shows the status of setpoint 7 in normal & remote mode.	B_6_R	4097	RAM
TRIGGER7	8-bit read only register which contains status flags for the Modbus master macro.	B_6_R	4098	RAM
RELAY7	Flag which shows/controls the instantaneous state of relay 7 (ON=energized).	B_6	4099	RAM

See also
Setpoint Status Flags - Register 4097

Setpoint Trigger Flags - Register 4098

Register 239 - Alarm Status

SP7 Setup

Register 4099 - Relay output

2.12.18.1 SP7 Setup

Name	Description	Symbol Type	Register Number	Memory Type
POWERON_INHIBIT_SP7	Bit flag shows that setpoint 7 has been in-active since power-on.	B_6	4102	RAM
RLY7_DE_ENERGISE	Bit flag shows that setpoint 7 has been in-active since power-on.	B_6	4101	RAM
SETPOINT7_TEXT	Text display for setpoint 7.	L_14_T	16507	EEPROM
SP7_BREAK_DELAY	16-bit register holds the break delay time for setpoint 7 (0.1s or 0.001s resolution).	U_16	4181	RAM/EEPROM
SP7_CONTROL	8-bit register holds the setpoint & relay control setting for setpoint 7 (note: controller display is in octal).	O_8	8251	RAM/EEPROM
SP7_DATA_SOURCE	16-bit register holds the register number of the data source for setpoint 7.	U_16	4344	RAM/EEPROM
SP7_DELAY_TYPE	8-bit register controls the delay type settings for setpoint 7.	U_8	8283	RAM/EEPROM
SP7_HYST	16-bit register holds the hysteresis/passband value for setpoint 7.	U_16	4149	RAM/EEPROM
SP7_LATCH	Flag shows/controls the latch status of setpoint 7 (ON = setpoint latched).	B_6	4100	RAM
SP7_MAKE_DELAY	16-bit register holds the make delay time for setpoint 7 (0.1s or 0.001s resolution).	U_16	4165	RAM/EEPROM
SP7_RESET_DESTINATION	8-bit register holds the destination register number for setpoint 7 trigger functions.	U_8	4197	RAM/EEPROM
SP7_RESET_VALUE	32-bit register holds the reset value used with setpoint 7 trigger functions.	S_32	441	RAM/EEPROM
SP7_TRACKING	8-bit register controls the setpoint tracking for setpoint 7.	U_8	8267	RAM/EEPROM
SP7_TRIGGER	8-bit register. Controls trigger functions of setpoint 7.	U_8	8299	RAM/EEPROM

See also

Setpoint Latch Mask - Register 4100

Relay De-energize Mask - Register 4101

Setpoint Reset Delay (Power-On Inhibit) - Register 4102

Octal Format

2.12.19 Setpoint 8

Name	Description	Symbol Type	Register Number	Memory Type
SETPOINT8	32-bit register for setpoint 8 value.	S_32	271	RAM/EEPROM
SETPOINT8_FLOAT	32-bit pseudo floating point register for setpoint 8 value. (See 32-bit Pseudo Floating Point).	PF_32	1551	RAM/EEPROM
SP8	This flag shows/controls the status of setpoint 8 (ON = setpoint activated).	B_7	239	RAM
SP8_REMOTE	Setting this bit to ON places setpoint 8 in remote mode.	B_23	239	RAM
SP8_STATUS	Read only flag shows the status of setpoint 8 in normal & remote mode.	B_7_R	4097	RAM
TRIGGER8	8-bit read only register which contains status flags for the Modbus master macro.	B_7_R	4098	RAM
RELAY8	Flag which shows/controls the instantaneous state of relay 8 (ON=energized).	B_7	4099	RAM

See also

Setpoint Status Flags - Register 4097

Setpoint Trigger Flags - Register 4098

Register 239 - Alarm Status

SP8 Setup

Register 4099 - Relay output

2.12.19.1 SP8 Setup

Name	Description	Symbol Type	Register Number	Memory Type
POWERON_INHIBIT_SP8	Bit flag shows that setpoint 8 has been in-active since power-on.	B_7	4102	RAM
RLY8_DE_ENERGISE	Bit flag shows that setpoint 8 has been in-active since power-on.	B_7	4101	RAM
SETPOINT8_TEXT	Text display for setpoint 8.	L_14_T	16509	EEPROM
SP8_BREAK_DELAY	16-bit register holds the break delay time for setpoint 8 (0.1s or 0.001s resolution).	U_16	4182	RAM/EEPROM
SP8_CONTROL	8-bit register holds the setpoint & relay control setting for setpoint 8 (note: controller display is in octal).	O_8	8252	RAM/EEPROM
SP8_DATA_SOURCE	16-bit register holds the register number of the data source for setpoint 8.	U_16	4345	RAM/EEPROM
SP8_DELAY_TYPE	8-bit register controls the delay type settings for setpoint 8.	U_8	8284	RAM/EEPROM
SP8_HYST	16-bit register holds the hysteresis/passband value for setpoint 8.	U_16	4150	RAM/EEPROM
SP8_LATCH	Flag shows/controls the latch status of setpoint 8 (ON = setpoint latched).	B_7	4100	RAM
SP8_MAKE_DELAY	16-bit register holds the make delay time for setpoint 8 (0.1s or 0.001s resolution).	U_16	4166	RAM/EEPROM

SP8_RESET_DESTINATION	8-bit register holds the destination register number for setpoint 8 trigger functions.	U_8	4198	RAM/EEPROM
SP8_RESET_VALUE	32-bit register holds the reset value used with setpoint 8 trigger functions.	S_32	443	RAM/EEPROM
SP8_TRACKING	8-bit register controls the setpoint tracking for setpoint 8.	U_8	8268	RAM/EEPROM
SP8_TRIGGER	8-bit register. Controls trigger functions of setpoint 8.	U_8	8300	RAM/EEPROM

See also

Setpoint Latch Mask - Register 4100

Relay De-energize Mask - Register 4101

Setpoint Reset Delay (Power-On Inhibit) - Register 4102

Octal Format

2.12.20 Setpoint 9

Name	Description	Symbol Type	Register Number	Memory Type
SETPOINT9	32-bit register for setpoint 9 value.	S_32	273	RAM/EEPROM
SETPOINT9_FLOAT	32-bit pseudo floating point register for setpoint 9 value. (See 32-bit Pseudo Floating Point).	PF_32	1553	RAM/EEPROM
SP9	This flag shows/controls the status of setpoint 9 (ON = setpoint activated).	B_8	239	RAM
SP9_REMOTE	Setting this bit to ON places setpoint 9 in remote mode.	B_24	239	RAM
SP9_STATUS	Read only flag shows the status of setpoint 9 in normal & remote mode.	B_8_R	4097	RAM
TRIGGER9	8-bit read only register which contains status flags for the Modbus master macro.	B_8_R	4098	RAM
RELAY9	Flag which shows/controls the instantaneous state of relay 9 (ON=energized).	B_8	4099	RAM

See also

Setpoint Status Flags - Register 4097

Setpoint Trigger Flags - Register 4098

Register 239 - Alarm Status

SP9 Setup

Register 4099 - Relay output

2.12.20.1 SP9 Setup

Name	Description	Symbol Type	Register Number	Memory Type
POWERON_INHIBIT_SP9	Bit flag shows that setpoint 9 has been in-active since power-on.	B_8	4102	RAM
RLY9_DE_ENERGISE	Bit flag shows that setpoint 9 has been in-active since power-on.	B_8	4101	RAM
SETPOINT9_TEXT	Text display for setpoint 9.	L_14_T	16511	EEPROM
SP9_BREAK_DELAY	16-bit register holds the break delay time for setpoint 9 (0.1s or 0.001s resolution).	U_16	4183	RAM/EEPROM
SP9_CONTROL	8-bit register holds the setpoint & relay control setting for setpoint 9 (note: controller display is in octal).	O_8	8253	RAM/EEPROM
SP9_DATA_SOURCE	16-bit register holds the register number of the data source for setpoint 9.	U_16	4346	RAM/EEPROM
SP9_DELAY_TYPE	8-bit register controls the delay type settings for setpoint 9.	U_8	8285	RAM/EEPROM
SP9_HYST	16-bit register holds the hysteresis/passband value for setpoint 9.	U_16	4151	RAM/EEPROM
SP9_LATCH	Flag shows/controls the latch status of setpoint 9 (ON = setpoint latched).	B_8	4100	RAM
SP9_MAKE_DELAY	16-bit register holds the make delay time for setpoint 9 (0.1s or 0.001s resolution).	U_16	4167	RAM/EEPROM
SP9_RESET_DESTINATION	8-bit register holds the destination register number for setpoint 9 trigger functions.	U_8	4199	RAM/EEPROM
SP9_RESET_VALUE	32-bit register holds the reset value used with setpoint 9 trigger functions.	S_32	445	RAM/EEPROM
SP9_TRACKING	8-bit register controls the setpoint tracking for setpoint 9.	U_8	8269	RAM/EEPROM
SP9_TRIGGER	8-bit register. Controls trigger functions of setpoint 9.	U_8	8301	RAM/EEPROM

See also

Setpoint Latch Mask - Register 4100

Relay De-energize Mask - Register 4101

Setpoint Reset Delay (Power-On Inhibit) - Register 4102

Octal Format

2.12.21 Setpoint 10

Name	Description	Symbol Type	Register Number	Memory Type
SETPOINT10	32-bit register for setpoint 10 value.	S_32	275	RAM/EEPROM
SETPOINT10_FLOAT	32-bit pseudo floating point register for setpoint 10 value. (See 32-bit Pseudo Floating Point).	PF_32	1555	RAM/EEPROM
SP10	This flag shows/controls the status of setpoint 10 (ON = setpoint activated).	B_9	239	RAM
SP10_REMOTE	Setting this bit to ON places setpoint 10 in remote mode.	B_25	239	RAM
SP10_STATUS	Read only flag shows the status of setpoint 10 in normal & remote mode.	B_9_R	4097	RAM
TRIGGER10	8-bit read only register which contains status flags for the Modbus master macro.	B_9_R	4098	RAM
RELAY10	Flag which shows/controls the instantaneous state of relay 10 (ON=energized).	B_9	4099	RAM

See also

Setpoint Status Flags - Register 4097

Setpoint Trigger Flags - Register 4098

Register 239 - Alarm Status

SP10 Setup

Register 4099 - Relay output

2.12.21.1 SP10 Setup

Name	Description	Symbol Type	Register Number	Memory Type
POWERON_INHIBIT_SP10	Bit flag shows that setpoint 10 has been in-active since power-on.	B_9	4102	RAM
RLY10_DE_ENERGISE	Bit flag shows that setpoint 10 has been in-active since power-on.	B_9	4101	RAM
SETPOINT10_TEXT	Text display for setpoint 10.	L_14_T	16513	EEPROM
SP10_BREAK_DELAY	16-bit register holds the break delay time for setpoint 10 (0.1s or 0.001s resolution).	U_16	4184	RAM/EEPROM
SP10_CONTROL	8-bit register holds the setpoint & relay control setting for setpoint 10 (note: controller display is in octal).	O_8	8254	RAM/EEPROM
SP10_DATA_SOURCE	16-bit register holds the register number of the data source for setpoint 10.	U_16	4347	RAM/EEPROM
SP10_DELAY_TYPE	8-bit register controls the delay type settings for setpoint 10.	U_8	8286	RAM/EEPROM
SP10_HYST	16-bit register holds the hysteresis/passband value for setpoint 10.	U_16	4152	RAM/EEPROM
SP10_LATCH	Flag shows/controls the latch status of setpoint 10 (ON = setpoint latched).	B_9	4100	RAM
SP10_MAKE_DELAY	16-bit register holds the make delay time for setpoint 10 (0.1s or 0.001s resolution).	U_16	4168	RAM/EEPROM

SP10_RESET_DESTINATION	8-bit register holds the destination register number for setpoint 10 trigger functions.	U_8	4200	RAM/EEPROM
SP10_RESET_VALUE	32-bit register holds the reset value used with setpoint 10 trigger functions.	S_32	447	RAM/EEPROM
SP10_TRACKING	8-bit register controls the setpoint tracking for setpoint 10.	U_8	8270	RAM/EEPROM
SP10_TRIGGER	8-bit register. Controls trigger functions of setpoint 10.	U_8	8302	RAM/EEPROM

See also

Setpoint Latch Mask - Register 4100

Relay De-energize Mask - Register 4101

Setpoint Reset Delay (Power-On Inhibit) - Register 4102

Octal Format

2.12.22 Setpoint 11

Name	Description	Symbol Type	Register Number	Memory Type
SETPOINT11	32-bit register for setpoint 11 value.	S_32	277	RAM/EEPROM
SETPOINT11_FLOAT	32-bit pseudo floating point register for setpoint 11 value. (See 32-bit Pseudo Floating Point).	PF_32	1557	RAM/EEPROM
SP11	This flag shows/controls the status of setpoint 11 (ON = setpoint activated).	B_10	239	RAM
SP11_REMOTE	Setting this bit to ON places setpoint 11 in remote mode.	B_26	239	RAM
SP11_STATUS	Read only flag shows the status of setpoint 11 in normal & remote mode.	B_10_R	4097	RAM
TRIGGER11	8-bit read only register which contains status flags for the Modbus master macro.	B_10_R	4098	RAM
RELAY11	Flag which shows/controls the instantaneous state of relay 11 (ON=energized).	B_10	4099	RAM

See also

Setpoint Status Flags - Register 4097

Setpoint Trigger Flags - Register 4098

Register 239 - Alarm Status

SP11 Setup

Register 4099 - Relay output

2.12.22.1 SP11 Setup

Name	Description	Symbol Type	Register Number	Memory Type
POWERON_INHIBIT_SP11	Bit flag shows that setpoint 11 has been in-active since power-on.	B_10	4102	RAM
RLY11_DE_ENERGISE	Bit flag shows that setpoint 11 has been in-active since power-on.	B_10	4101	RAM
SETPOINT11_TEXT	Text display for setpoint 11.	L_14_T	16515	EEPROM
SP11_BREAK_DELAY	16-bit register holds the break delay time for setpoint 11 (0.1s or 0.001s resolution).	U_16	4185	RAM/EEPROM
SP11_CONTROL	8-bit register holds the setpoint & relay control setting for setpoint 11 (note: controller display is in octal).	O_8	8255	RAM/EEPROM
SP11_DATA_SOURCE	16-bit register holds the register number of the data source for setpoint 11.	U_16	4348	RAM/EEPROM
SP11_DELAY_TYPE	8-bit register controls the delay type settings for setpoint 11.	U_8	8287	RAM/EEPROM
SP11_HYST	16-bit register holds the hysteresis/passband value for setpoint 11.	U_16	4153	RAM/EEPROM
SP11_LATCH	Flag shows/controls the latch status of setpoint 11 (ON = setpoint latched).	B_10	4100	RAM
SP11_MAKE_DELAY	16-bit register holds the make delay time for setpoint 11 (0.1s or 0.001s resolution).	U_16	4169	RAM/EEPROM
SP11_RESET_DESTINATION	8-bit register holds the destination register number for setpoint 11 trigger functions.	U_8	4201	RAM/EEPROM
SP11_RESET_VALUE	32-bit register holds the reset value used with setpoint 11 trigger functions.	S_32	449	RAM/EEPROM
SP11_TRACKING	8-bit register controls the setpoint tracking for setpoint 11.	U_8	8271	RAM/EEPROM
SP11_TRIGGER	8-bit register. Controls trigger functions of setpoint 11.	U_8	8303	RAM/EEPROM

See also

Setpoint Latch Mask - Register 4100

Relay De-energize Mask - Register 4101

Setpoint Reset Delay (Power-On Inhibit) - Register 4102

Octal Format

2.12.23 Setpoint 12

Name	Description	Symbol Type	Register Number	Memory Type
SETPOINT12	32-bit register for setpoint 12 value.	S_32	279	RAM/EEPROM
SETPOINT12_FLOAT	32-bit pseudo floating point register for setpoint 12 value. (See 32-bit Pseudo Floating Point).	PF_32	1559	RAM/EEPROM
SP12	This flag shows/controls the status of setpoint 12 (ON = setpoint activated).	B_11	239	RAM
SP12_REMOTE	Setting this bit to ON places setpoint 12 in remote mode.	B_27	239	RAM
SP12_STATUS	Read only flag shows the status of setpoint 12 in normal & remote mode.	B_11_R	4097	RAM
TRIGGER12	8-bit read only register which contains status flags for the Modbus master macro.	B_11_R	4098	RAM
RELAY12	Flag which shows/controls the instantaneous state of relay 12 (ON=energized).	B_11	4099	RAM

See also

Setpoint Status Flags - Register 4097

Setpoint Trigger Flags - Register 4098

Register 239 - Alarm Status

SP12 Setup

Register 4099 - Relay output

2.12.23.1 SP12 Setup

Name	Description	Symbol Type	Register Number	Memory Type
POWERON_INHIBIT_SP12	Bit flag shows that setpoint 12 has been in-active since power-on.	B_11	4102	RAM
RLY12_DE_ENERGISE	Bit flag shows that setpoint 12 has been in-active since power-on.	B_11	4101	RAM
SETPOINT12_TEXT	Text display for setpoint 12.	L_14_T	16517	EEPROM
SP12_BREAK_DELAY	16-bit register holds the break delay time for setpoint 12 (0.1s or 0.001s resolution).	U_16	4186	RAM/EEPROM
SP12_CONTROL	8-bit register holds the setpoint & relay control setting for setpoint 12 (note: controller display is in octal).	O_8	8256	RAM/EEPROM
SP12_DATA_SOURCE	16-bit register holds the register number of the data source for setpoint 12.	U_16	4349	RAM/EEPROM
SP12_DELAY_TYPE	8-bit register controls the delay type settings for setpoint 12.	U_8	8288	RAM/EEPROM
SP12_HYST	16-bit register holds the hysteresis/passband value for setpoint 12.	U_16	4154	RAM/EEPROM
SP12_LATCH	Flag shows/controls the latch status of setpoint 12 (ON = setpoint latched).	B_11	4100	RAM
SP12_MAKE_DELAY	16-bit register holds the make delay time for setpoint 12 (0.1s or 0.001s resolution).	U_16	4170	RAM/EEPROM

SP12_RESET_DESTINATION	8-bit register holds the destination register number for setpoint 12 trigger functions.	U_8	4202	RAM/EEPROM
SP12_RESET_VALUE	32-bit register holds the reset value used with setpoint 12 trigger functions.	S_32	451	RAM/EEPROM
SP12_TRACKING	8-bit register controls the setpoint tracking for setpoint 12.	U_8	8272	RAM/EEPROM
SP12_TRIGGER	8-bit register. Controls trigger functions of setpoint 12.	U_8	8304	RAM/EEPROM

See also
Setpoint Latch Mask - Register 4100

Relay De-energize Mask - Register 4101

Setpoint Reset Delay (Power-On Inhibit) - Register 4102

Octal Format

2.12.24 Setpoint 13

Name	Description	Symbol Type	Register Number	Memory Type
SETPOINT13	32-bit register for setpoint 13 value.	S_32	281	RAM/EEPROM
SETPOINT13_FLOAT	32-bit pseudo floating point register for setpoint 13 value. (See 32-bit Pseudo Floating Point).	PF_32	1561	RAM/EEPROM
SP13	This flag shows/controls the status of setpoint 13 (ON = setpoint activated).	B_12	239	RAM
SP13_REMOTE	Setting this bit to ON places setpoint 13 in remote mode.	B_28	239	RAM
SP13_STATUS	Read only flag shows the status of setpoint 13 in normal & remote mode.	B_12_R	4097	RAM
TRIGGER13	8-bit read only register which contains status flags for the Modbus master macro.	B_12_R	4098	RAM
RELAY13	Flag which shows/controls the instantaneous state of relay 13 (ON=energized).	B_12	4099	RAM

See also
Setpoint Status Flags - Register 4097

Setpoint Trigger Flags - Register 4098

Register 239 - Alarm Status

SP13 Setup

Register 4099 - Relay output

2.12.24.1 SP13 Setup

Name	Description	Symbol Type	Register Number	Memory Type
POWERON_INHIBIT_SP13	Bit flag shows that setpoint 13 has been in-active since power-on.	B_12	4102	RAM
RLY13_DE_ENERGISE	Bit flag shows that setpoint 13 has been in-active since power-on.	B_12	4101	RAM
SETPOINT13_TEXT	Text display for setpoint 13.	L_14_T	16519	EEPROM
SP13_BREAK_DELAY	16-bit register holds the break delay time for setpoint 13 (0.1s or 0.001s resolution).	U_16	4187	RAM/EEPROM
SP13_CONTROL	8-bit register holds the setpoint & relay control setting for setpoint 13 (note: controller display is in octal).	O_8	8257	RAM/EEPROM
SP13_DATA_SOURCE	16-bit register holds the register number of the data source for setpoint 13.	U_16	4350	RAM/EEPROM
SP13_DELAY_TYPE	8-bit register controls the delay type settings for setpoint 13.	U_8	8289	RAM/EEPROM
SP13_HYST	16-bit register holds the hysteresis/passband value for setpoint 13.	U_16	4155	RAM/EEPROM
SP13_LATCH	Flag shows/controls the latch status of setpoint 13 (ON = setpoint latched).	B_12	4100	RAM
SP13_MAKE_DELAY	16-bit register holds the make delay time for setpoint 13 (0.1s or 0.001s resolution).	U_16	4171	RAM/EEPROM
SP13_RESET_DESTINATION	8-bit register holds the destination register number for setpoint 13 trigger functions.	U_8	4203	RAM/EEPROM
SP13_RESET_VALUE	32-bit register holds the reset value used with setpoint 13 trigger functions.	S_32	453	RAM/EEPROM
SP13_TRACKING	8-bit register controls the setpoint tracking for setpoint 13.	U_8	8273	RAM/EEPROM
SP13_TRIGGER	8-bit register. Controls trigger functions of setpoint 13.	U_8	8305	RAM/EEPROM

See also

Setpoint Latch Mask - Register 4100

Relay De-energize Mask - Register 4101

Setpoint Reset Delay (Power-On Inhibit) - Register 4102

Octal Format

2.12.25 Setpoint 14

Name	Description	Symbol Type	Register Number	Memory Type
SETPOINT14	32-bit register for setpoint 14 value.	S_32	283	RAM/EEPROM
SETPOINT14_FLOAT	32-bit pseudo floating point register for setpoint 14 value. (See 32-bit Pseudo Floating Point).	PF_32	1563	RAM/EEPROM
SP14	This flag shows/controls the status of setpoint 14 (ON = setpoint activated).	B_13	239	RAM
SP14_REMOTE	Setting this bit to ON places setpoint 14 in remote mode.	B_29	239	RAM
SP14_STATUS	Read only flag shows the status of setpoint 14 in normal & remote mode.	B_13_R	4097	RAM
TRIGGER14	8-bit read only register which contains status flags for the Modbus master macro.	B_13_R	4098	RAM
RELAY14	Flag which shows/controls the instantaneous state of relay 14 (ON=energized).	B_13	4099	RAM

See also

Setpoint Status Flags - Register 4097

Setpoint Trigger Flags - Register 4098

Register 239 - Alarm Status

SP14 Setup

Register 4099 - Relay output

2.12.25.1 SP14 Setup

Name	Description	Symbol Type	Register Number	Memory Type
POWERON_INHIBIT_SP14	Bit flag shows that setpoint 14 has been in-active since power-on.	B_13	4102	RAM
RLY14_DE_ENERGISE	Bit flag shows that setpoint 14 has been in-active since power-on.	B_13	4101	RAM
SETPOINT14_TEXT	Text display for setpoint 14.	L_14_T	16521	EEPROM
SP14_BREAK_DELAY	16-bit register holds the break delay time for setpoint 14 (0.1s or 0.001s resolution).	U_16	4188	RAM/EEPROM
SP14_CONTROL	8-bit register holds the setpoint & relay control setting for setpoint 14 (note: controller display is in octal).	O_8	8258	RAM/EEPROM
SP14_DATA_SOURCE	16-bit register holds the register number of the data source for setpoint 14.	U_16	4351	RAM/EEPROM
SP14_DELAY_TYPE	8-bit register controls the delay type settings for setpoint 14.	U_8	8290	RAM/EEPROM
SP14_HYST	16-bit register holds the hysteresis/passband value for setpoint 14.	U_16	4156	RAM/EEPROM
SP14_LATCH	Flag shows/controls the latch status of setpoint 14 (ON = setpoint latched).	B_13	4100	RAM
SP14_MAKE_DELAY	16-bit register holds the make delay time for setpoint 14 (0.1s or 0.001s resolution).	U_16	4172	RAM/EEPROM

SP14_RESET_DESTINATION	8-bit register holds the destination register number for setpoint 14 trigger functions.	U_8	4204	RAM/EEPROM
SP14_RESET_VALUE	32-bit register holds the reset value used with setpoint 14 trigger functions.	S_32	455	RAM/EEPROM
SP14_TRACKING	8-bit register controls the setpoint tracking for setpoint 14.	U_8	8274	RAM/EEPROM
SP14_TRIGGER	8-bit register. Controls trigger functions of setpoint 14.	U_8	8306	RAM/EEPROM

See also

Setpoint Latch Mask - Register 4100

Relay De-energize Mask - Register 4101

Setpoint Reset Delay (Power-On Inhibit) - Register 4102

Octal Format

2.12.26 Setpoint 15

Name	Description	Symbol Type	Register Number	Memory Type
SETPOINT15	32-bit register for setpoint 15 value.	S_32	285	RAM/EEPROM
SETPOINT15_FLOAT	32-bit pseudo floating point register for setpoint 15 value. (See 32-bit Pseudo Floating Point).	PF_32	1565	RAM/EEPROM
SP15	This flag shows/controls the status of setpoint 15 (ON = setpoint activated).	B_14	239	RAM
SP15_REMOTE	Setting this bit to ON places setpoint 15 in remote mode.	B_30	239	RAM
SP15_STATUS	Read only flag shows the status of setpoint 15 in normal & remote mode.	B_14_R	4097	RAM
TRIGGER15	8-bit read only register which contains status flags for the Modbus master macro.	B_14_R	4098	RAM
RELAY15	Flag which shows/controls the instantaneous state of relay 15 (ON=energized).	B_14	4099	RAM

See also

Setpoint Status Flags - Register 4097

Setpoint Trigger Flags - Register 4098

Register 239 - Alarm Status

SP15 Setup

Register 4099 - Relay output

2.12.26.1 SP15 Setup

Name	Description	Symbol Type	Register Number	Memory Type
POWERON_INHIBIT_SP15	Bit flag shows that setpoint 15 has been in-active since power-on.	B_14	4102	RAM
RLY15_DE_ENERGISE	Bit flag shows that setpoint 15 has been in-active since power-on.	B_14	4101	RAM
SETPOINT15_TEXT	Text display for setpoint 15.	L_14_T	16523	EEPROM
SP15_BREAK_DELAY	16-bit register holds the break delay time for setpoint 15 (0.1s or 0.001s resolution).	U_16	4189	RAM/EEPROM
SP15_CONTROL	8-bit register holds the setpoint & relay control setting for setpoint 15 (note: controller display is in octal).	O_8	8259	RAM/EEPROM
SP15_DATA_SOURCE	16-bit register holds the register number of the data source for setpoint 15.	U_16	4352	RAM/EEPROM
SP15_DELAY_TYPE	8-bit register controls the delay type settings for setpoint 15.	U_8	8291	RAM/EEPROM
SP15_HYST	16-bit register holds the hysteresis/passband value for setpoint 15.	U_16	4157	RAM/EEPROM
SP15_LATCH	Flag shows/controls the latch status of setpoint 15 (ON = setpoint latched).	B_14	4100	RAM
SP15_MAKE_DELAY	16-bit register holds the make delay time for setpoint 15 (0.1s or 0.001s resolution).	U_16	4173	RAM/EEPROM
SP15_RESET_DESTINATION	8-bit register holds the destination register number for setpoint 15 trigger functions.	U_8	4205	RAM/EEPROM
SP15_RESET_VALUE	32-bit register holds the reset value used with setpoint 15 trigger functions.	S_32	457	RAM/EEPROM
SP15_TRACKING	8-bit register controls the setpoint tracking for setpoint 15.	U_8	8275	RAM/EEPROM
SP15_TRIGGER	8-bit register. Controls trigger functions of setpoint 15.	U_8	8307	RAM/EEPROM

See also

Setpoint Latch Mask - Register 4100

Relay De-energize Mask - Register 4101

Setpoint Reset Delay (Power-On Inhibit) - Register 4102

Octal Format

2.12.27 Setpoint 16

Name	Description	Symbol Type	Register Number	Memory Type
SETPOINT16	32-bit register for setpoint 16 value.	S_32	287	RAM/EEPROM
SETPOINT16_FLOAT	32-bit pseudo floating point register for setpoint 16 value. (See 32-bit Pseudo Floating Point).	PF_32	1567	RAM/EEPROM
SP16	This flag shows/controls the status of setpoint 16 (ON = setpoint activated).	B_15	239	RAM
SP16_REMOTE	Setting this bit to ON places setpoint 16 in remote mode.	B_31	239	RAM
SP16_STATUS	Read only flag shows the status of setpoint 16 in normal & remote mode.	B_15_R	4097	RAM
TRIGGER16	8-bit read only register which contains status flags for the Modbus master macro.	B_15_R	4098	RAM
RELAY16	Flag which shows/controls the instantaneous state of relay 16 (ON=energized).	B_15	4099	RAM

See also

Setpoint Status Flags - Register 4097

Setpoint Trigger Flags - Register 4098

Register 239 - Alarm Status

SP16 Setup

Register 4099 - Relay output

2.12.27.1 SP16 Setup

Name	Description	Symbol Type	Register Number	Memory Type
POWERON_INHIBIT_SP16	Bit flag shows that setpoint 16 has been in-active since power-on.	B_15	4102	RAM
RLY16_DE_ENERGISE	Bit flag shows that setpoint 16 has been in-active since power-on.	B_15	4101	RAM
SETPOINT16_TEXT	Text display for setpoint 16.	L_14_T	16525	EEPROM
SP16_BREAK_DELAY	16-bit register holds the break delay time for setpoint 16 (0.1s or 0.001s resolution).	U_16	4190	RAM/EEPROM
SP16_CONTROL	8-bit register holds the setpoint & relay control setting for setpoint 16 (note: controller display is in octal).	O_8	8260	RAM/EEPROM
SP16_DATA_SOURCE	16-bit register holds the register number of the data source for setpoint 16.	U_16	4353	RAM/EEPROM
SP16_DELAY_TYPE	8-bit register controls the delay type settings for setpoint 16.	U_8	8292	RAM/EEPROM
SP16_HYST	16-bit register holds the hysteresis/passband value for setpoint 16.	U_16	4158	RAM/EEPROM
SP16_LATCH	Flag shows/controls the latch status of setpoint 16 (ON = setpoint latched).	B_15	4100	RAM
SP16_MAKE_DELAY	16-bit register holds the make delay time for setpoint 16 (0.1s or 0.001s resolution).	U_16	4174	RAM/EEPROM

SP16_RESET_DESTINATION	8-bit register holds the destination register number for setpoint 16 trigger functions.	U_8	4206	RAM/EEPROM
SP16_RESET_VALUE	32-bit register holds the reset value used with setpoint 16 trigger functions.	S_32	459	RAM/EEPROM
SP16_TRACKING	8-bit register controls the setpoint tracking for setpoint 16.	U_8	8276	RAM/EEPROM
SP16_TRIGGER	8-bit register. Controls trigger functions of setpoint 16.	U_8	8308	RAM/EEPROM

See also

Setpoint Latch Mask - Register 4100

Relay De-energize Mask - Register 4101

Setpoint Reset Delay (Power-On Inhibit) - Register 4102

Octal Format

2.13 Status Registers

Apart from configuration and working registers the controller also contains various status registers which contain flags relating to key functions in the controllers operation. Some flags are read only while others also allow the user to take control of functions remotely or via the macro.

The table below shows status registers available in the Zen IoT.

Name	Description	Symbol Type	Register Number	Memory Type
ALARM_STATUS	32 bit value showing status and allowing control of setpoints.	U_32	239	RAM
AMBIENT_TEMP	32 bit read only float showing ambient temperature.	F_32_R	1225	RAM
AMBIENT_TEMP_SWAPPED	32 bit read only float showing ambient temperature. **Note**: This register is used to maintain backwards compatibility with older Intech products. When reading this register via Modbus the word order is Big Endian.	SF_32_R	121	RAM
EEPROM_MEMORY_SIZE	16-bit register shows how much EEPROM memory is fitted in the controller (value in Kbytes).	U_16_R	4437	RAM
RELAY_DE_ENERGIZE_FLAGS	16-bit register holds the de-energized status for relays.	U_16	4101	RAM
SETPOINT_STATUS_FLAGS	16 bit read only register shows the status of the setpoints	U_16_R	4097	RAM
DIGITAL_PINS	16-bit register contains flags for the digital inputs on the top of the controller.	U_16	4108	RAM
STATE	16-bit register. Used to control the state number of the macro (cleared to 0 when returning to operational display).	U_16	4109	RAM

LAST_ERROR	8 bit register shows the last error encountered by the controller during power up.	U_8	8431	EEPROM
ERROR_STATUS	8 bit register shows the current error status of the controller during power up (0=no errors).	U_8	8435	RAM
ERROR_THRESHOLD	8 bit register sets the threshold at which errors will stop operation. Errors at or below the threshold will halt normal operation while errors above the threshold value will be automatically corrected (by loading default values).	U_8	8436	RAM/EEPROM
SOFTWARE_VERSION_NO	16-bit register which displays the software version number currently operating the controller.	U_16_R	4106	RAM
DEVICE_TYPE	Text register which displays the device type	L_14_T	16565	RAM
SERIAL_NO	32 bit read only register. Contains product serial number (V4.02a onwards)	U_32_R	541	EEPROM
CAL_DATE	32 bit register that contains the last calibration date. (MSW=Year, LSW=(MSB=Month, LSB=Date)).	U_32	543	EEPROM

Register 4099 Relay Output Image

This 16-bit register shows the current status of the relays after setpoint processing has been done. The difference between this register and register 239 (alarm status) is that the alarm status register shows the current status of the setpoints as opposed to the relays. For example a setpoint may have been activated but the relay may not yet be turned on because of a 10 second delay on make. In this case, reading alarm status register 239 shows the setpoint as active, but reading 4099 would show that the relay had not yet turned ON.

Writing To Register 4099

Register 4099 is normally used to read the current status of a relay as it is controlled by the setpoint logic. However it is possible to write directly to these flags as well, but the user should first understand the operation of the setpoint logic thoroughly before doing so. Under normal situations the setpoint logic calculates the new status of the relays at the control output rate (10mS or 100mS) and writes a new value to register 4099. This means that writes to this register from any other sources (macro or serial ports) will be overwritten by the setpoint logic. If you wish to write directly to a relay you should ensure that the setpoint logic for that relay is disabled by setting the setpoint source to "OFF". This means that all of the setpoint functions will be disabled and the relay will be totally under your control.

Speed Of Write To 4099

Under normal conditions (excluding PID and high resolution modes) the relay states are only updated at the control output rate. However, for setpoints 1 - 6, a write from the macro or serial port to register 4099 will cause the new value to appear directly on the relay output pins. (This only applies to code versions 4.01e and later).

Name	Description	Symbol Type	Register Number	Memory Type
RELAY_STATUS	16-bit register contains flags showing the instantaneous status of each relay (Note, this may be different to the setpoint status).	U_16	4099	RAM

The function of each bit of register 4099 is shown as follows.

Bit	Name	Description	Function
0	RELAY1	Flag shows the instantaneous state of relay 1 (ON = energized).	0 = OFF 1 = ON
1	RELAY2	Flag shows the instantaneous state of relay 2 (ON = energized).	0 = OFF 1 = ON
2	RELAY3	Flag shows the instantaneous state of relay 3 (ON = energized).	0 = OFF 1 = ON
3	RELAY4	Flag shows the instantaneous state of relay 4 (ON = energized).	0 = OFF 1 = ON
4	RELAY5	Flag shows the instantaneous state of relay 5 (ON = energized).	0 = OFF 1 = ON
5	RELAY6	Flag shows the instantaneous state of relay 6 (ON = energized).	0 = OFF 1 = ON
6	RELAY7	Flag shows the instantaneous state of relay 7 (ON = energized).	0 = OFF 1 = ON
7	RELAY8	Flag shows the instantaneous state of relay 8 (ON = energized).	0 = OFF 1 = ON
8	RELAY9	Flag shows the instantaneous state of relay 9 (ON = energized).	0 = OFF 1 = ON
9	RELAY10	Flag shows the instantaneous state of relay 10 (ON = energized).	0 = OFF 1 = ON
10	RELAY11	Flag shows the instantaneous state of relay 11 (ON = energized).	0 = OFF 1 = ON
11	RELAY12	Flag shows the instantaneous state of relay 12 (ON = energized).	0 = OFF 1 = ON
12	RELAY13	Flag shows the instantaneous state of relay 13 (ON = energized).	0 = OFF 1 = ON
13	RELAY14	Flag shows the instantaneous state of relay 14 (ON = energized).	0 = OFF 1 = ON
14	RELAY15	Flag shows the instantaneous state of relay 15 (ON = energized).	0 = OFF 1 = ON
15	RELAY16	Flag shows the instantaneous state of relay 16 (ON = energized).	0 = OFF 1 = ON

See also
Status Switches

2.13.1 Input Module Status

The Zen IoT has 16 analog inputs and each input channels has an associated status register which shows the current operating state of that channel. Registers 4592 to 4607 are 16 bit unsigned registers that hold the current status of each input channel. Most flags in the status registers are read only, however there is a special write function associated with each of these registers. See Writing To Input Module Status Registers below.

Each input module status register contains up to 16 bit flags which define different status functions. The table below shows the meaning of each status bit.

NOTE: The status bits shown below relate to the standard isolated input module. In the future these functions could change as new input modules are released or new functions are introduced.

Bit	Name	Description	Function
0	Not Initialized	This flag shows that the input module is not initialized. This flag is normally set at initial power on or when the input mode is changed.	0 = OK 1 = Not Initialized
1	Busy	This flag shows that the input module is busy doing some internal function and may not be able to execute normal input sampling.	0 = OK 1 = Busy
2	No Response	This flag indicates that the input module is not responding to the Zen IoT's repeated attempts to communicate with it. After 5 consecutive bad or no responses, the Zen IoT will stop trying and assume the module is no longer operational. (See Writing To Input Module Status Registers below to reset this flag)	0 = OK 1 = Not responding
3	Flash Memory Error	This flag indicates that a Flash memory error has occurred in the input module. This means that the configuration information in the input module has been corrupted and it can no longer operate correctly. Contact Define Instruments Ltd. for service advice.	0 = OK 1 = Flash Memory Error
4	Sensor Error	This flag indicates that a sensor error has been detected. This is used in RTD or TC modes to show that the sensor is open or short circuit. If this flag is set, the result data for this channel should be ignored.	0 = OK 1 = Sensor Error
5	Over Range	This flag indicates that input channel is in over range and the result data for the channel is not valid.	0 = OK 1 = Over Range
6	Under Range	This flag indicates that input channel is in under range and the result data for the channel is not valid.	0 = OK 1 = Under Range
7	32 Bit Result	For the isolated analog input module this bit will always be 0, but in future designs this bit could be set to 1 to instruct the Zen IoT main controller to do a 32 bit read of the result register.	0 = Normal 16 bit Result 1 = 32 bit Result
8	Incremental Read	This flag tells the Zen IoT main controller that value in the result register is an incremental value which shows the difference between the previous read. When this bit is set, the result register will be automatically reset to zero after each read of the result register. This mode is used when the input module is set to counter mode.	0 = Normal Read 1 = Incremental Read
9-15	Reserved	These flags are not used at present and are reserved for future functions.	0 = OK 1 = Don't Care

Writing To Input Module Status Registers
Although the flags contained in each of these register are normally read only flags, it is possible to write to these registers. A write to one of these status registers will cause the Zen IoT main controller to clear the No Response flag (bit 2) and will reset its internal retry count, causing the Zen IoT to resume polling this input module again.

2.13.2 Module ID

Zen IoT controllers support various input and output modules which can be fitted in the analogue channel slots. The Zen IoT detects which type of module is fitted in each of the channel slots, alters the functions for that channel appropriately.

As from firmware version **V0.08.01 onwards**, Zen IoT controllers support the following module ID types.

Module ID Codes

Module ID	Description	Module Type	Module Function
0	No module detected in channel slot.	-	-
1 - 4	Not allocated. Reserved for future development.	Analogue	Input
5	Isolated multi-input module (default).	Analogue	Input
6	Non-isolated RTD input module.	Analogue	Input
7 - 63	Not allocated. Reserved for future development.	Analogue	Input
64	Isolated passive analogue output module.	Analogue	Output
65 - 127	Not allocated. Reserved for future development.	Analogue	Output
128	SPDT relay output module.	Digital	Input/Output
129 - 191	Not allocated. Reserved for future development.	Digital	Input/Output
192 - 255	Not allocated. Reserved for future development.	Reserved	Reserved

The type of module fitted in each slot can be read from the 16 Module ID registers shown in the table below.

Name	Description	Symbol Type	Register Number	Memory Type
MODULE_ID1	8-bit read only register that reports the module ID code for the module inserted in the CH1 slot. (See module ID table).	U_8_R	8515	RAM
MODULE_ID2	8-bit read only register that reports the module ID code for the module inserted in the CH1 slot. (See module ID table).	U_8_R	8516	RAM
MODULE_ID3	8-bit read only register that reports the module ID code for the module inserted in the CH1 slot. (See module ID table).	U_8_R	8517	RAM
MODULE_ID4	8-bit read only register that reports the module ID code for the module inserted in the CH1 slot. (See module ID table).	U_8_R	8518	RAM
MODULE_ID5	8-bit read only register that reports the module ID code for the module inserted in the CH1 slot. (See module ID table).	U_8_R	8519	RAM
MODULE_ID6	8-bit read only register that reports the module ID code for the module inserted in the CH1 slot. (See module ID table).	U_8_R	8520	RAM
MODULE_ID7	8-bit read only register that reports the module ID code for the module inserted in the CH1 slot. (See module ID table).	U_8_R	8521	RAM
MODULE_ID8	8-bit read only register that reports the module ID code for the module inserted in the CH1 slot. (See module ID table).	U_8_R	8522	RAM

MODULE_ID9	8-bit read only register that reports the module ID code for the module inserted in the CH1 slot. (See module ID table).	U_8_R	8523	RAM
MODULE_ID10	8-bit read only register that reports the module ID code for the module inserted in the CH1 slot. (See module ID table).	U_8_R	8524	RAM
MODULE_ID11	8-bit read only register that reports the module ID code for the module inserted in the CH1 slot. (See module ID table).	U_8_R	8525	RAM
MODULE_ID12	8-bit read only register that reports the module ID code for the module inserted in the CH1 slot. (See module ID table).	U_8_R	8526	RAM
MODULE_ID13	8-bit read only register that reports the module ID code for the module inserted in the CH1 slot. (See module ID table).	U_8_R	8527	RAM
MODULE_ID14	8-bit read only register that reports the module ID code for the module inserted in the CH1 slot. (See module ID table).	U_8_R	8528	RAM
MODULE_ID15	8-bit read only register that reports the module ID code for the module inserted in the CH1 slot. (See module ID table).	U_8_R	8529	RAM
MODULE_ID16	8-bit read only register that reports the module ID code for the module inserted in the CH1 slot. (See module ID table).	U_8_R	8530	RAM

2.13.3 Error Status

Last Error - Register 8431

Register 8431 is an 8 bit unsigned register which records the first error encountered after power up. Register 8431 is also stored in non volatile memory so that it can be viewed even after a power down. Register 8431 can be read or written to for clearing an error number if required, but unlike register 8435 below, clearing register 8431 has no other function. It will not remove or fix any error condition or enable the controller to continue operating. It's purpose is purely for a diagnostics tool. See the Error table below for an explanation of the error codes.

Current Error Status at Power up - Register 8435

Register 8435 is an 8 bit unsigned register which reports the current status of any errors at power up. It is useful for controller models which do not have a display or are run with a different display option. After power up, the Zen IoT will start loading configuration data from non-volatile memory and as it does so, it checks the data for errors. If errors are found, register 8435 is loaded with the appropriate error number as described in the Error table below. Error numbers are arranged in order of priority (1= highest priority, 255=lowest). Higher priority errors will be reported in preference to lower priority errors. If the Zen IoT powers up successfully with no errors, register 8435 will be loaded with a value of zero.

Note: Multiple errors are not reported. If more than 1 error is discovered, only the error with the highest priority is reported.

Writing to Register 8435

Writing any value to register 8435 will attempt to clear the current error condition by reprogramming the effected block of data back to the factory defaults. If when attempting to repair an error, the controller finds there are more than **10** errors, the Zen IoT will initialize all of its NV memory to factory defaults. This does not effect the calibration or configuration of any input modules and it will not effect the calibration of the analog output (unless it detects an error in the analog output configuration data).

Note: Writing to register 8435 will always cause the Zen IoT to reset and re-power again. The will result in no reply being sent via the serial port so management software must expect a timeout condition after writing to this register.

Note: While clearing an error may remove the error condition, it cannot return the data to it's original state prior to the error. It is highly likely that the controller will need to be re-configured or even re-calibrated in some cases.

The remedy for each error condition will vary widely. If you encounter one of these errors you should contact your supplier for advice, quoting the error code.

Error Threshold - Register 8436
Register 8436 is an 8 bit unsigned register which controls the operation of the Zen IoT when it encounters an error. The Error table below shows different errors in order of priority. If the final error has a lower priority (higher number) than that stored in register 8436, the Zen IoT will ignore the error and continue with normal operation. If the error has the same or higher priority than register 8436 (i.e. reg 8435 <= 8436), the Zen IoT will halt normal operation and wait for some intervention, either via a display panel or via the serial port.

The factory default setting for register 8436 is 20.

Error Table
The following table shows the possible error conditions and suggested causes. The last column shows an error group which is defined in the Error Group Table below.

Error #	Error Description	Group #
0	Successful power up with no errors.	
1	The controller has attempted to read it's onboard EEPROM memory but either the SCL or SDA line has been held low. This could be a device busy but is more likely to be a hardware fault. (check for shorts and check pull up resistor on SCL.) This could also be caused by a fault in cabling or external equipment connected to the Zen IoT's expansion output socket.	1
2	The controller has attempted to read the EEPROM but didn't receive any acknowledgement that it was there. Either there is no EEPROM installed in the required position or the wrong type of EEPROM has been installed (must be 24xC128 or greater)	1
3	During a read attempt of EEPROM the controller received a negative acknowledge error.	1
4	The controller managed to read data out of the EEPROM but the checksum for the data was incorrect.	1
5	Memory size error – checksum is ok but the memory size is either 0 or some other value which is not a valid size for this controller.	1
6	An internal RAM error was detected. This means that some or all of the microprocessors internal RAM is faulty and the controller will have to be returned to Define Instruments Ltd. for service.	1
7	Checksum error when reading setups Cal – Code10.	1
8	Checksum error when reading baud rate & serial address.	1
9	Brown out trigger activated at power on. Check power supply voltages.	1
12	Checksum error when reading input module 1-8 scaling data for 12 bit registers.	2
13	Checksum error when reading input module 9-16 scaling data for 12 bit registers.	2
14	Checksum error when reading input module 1-8 offset data for 12 bit registers.	2
15	Checksum error when reading input module 9-16 offset data for 12 bit registers.	2
20	Checksum error when reading display format data. (This effects scale factor for floating point values.)	2
21	Checksum error when reading MUX source data.	2
22	Checksum error when reading input module 1-8 scaling data.	2
23	Checksum error when reading input module 9-16 scaling data.	2
24	Checksum error when reading input module 1-8 offset data.	2
25	Checksum error when reading input module 9-16 offset data.	2
26	Checksum error when reading config data for cold junction channel selection.	2

30	Checksum error when reading Intech controller flag data.	3
31	Checksum error when reading Intech controller setpoint data.	3
32	Checksum error when reading XOR mask data for Intech controller outputs.	3
33	Checksum error when reading Intech controller cooling differential data.	3
34	Checksum error when reading Intech controller deadband data.	3
35	Checksum error when reading Intech controller heating differential data.	3
36	Checksum error when reading I/O module type. Default will be set 0 for auto detect.	3
40	Checksum error when reading D/A calibration low data.	4
41	Checksum error when reading D/A calibration high data.	4
42	Checksum error when reading analog output source values.	4
43	Checksum error when reading 16 analog output source values for input module slots.	4
50	Checksum error when reading scaling data for frequency/counters (DI1 - DI4).	5
51	Checksum error when reading offset data for frequency/counters (DI1 - DI4).	5
52	Checksum error when reading counter A memory from on board FLASH	5
53	Checksum error when reading counter B memory from on board FLASH	5
54	Checksum error when reading counter C memory from on board FLASH	5
55	Checksum error when reading counter D memory from on board FLASH	5
56	Checksum error when reading raw result memory from on board FLASH	5
57	Checksum error when reading averaging window values for frequency (DI1 - DI4).	5
58	Checksum error when reading averaging sample values for frequency (DI1 - DI4).	5
60	Checksum error when reading configuration data for totalizers.	6
61	Checksum error when reading totalisator source values.	6
62	Checksum error when reading final total configuration data.	6
63	Checksum error when reading totalizer memory from on board FLASH	6
70	Checksum error when reading data log write pointer from on board FLASH	7
71	Checksum error when reading data log read pointer from on board FLASH	7
72	Checksum error when reading data log register pointers 1-16	7
73	Checksum error when reading data log register pointers 17-32	7
74	Checksum error when reading data log read size	7
75	EEPROM memory size has changed from the size originally installed in the controller. It now has no internal data logging memory installed. To accept this change, press the Prog button and the new memory size will be saved.	7
80	Checksum error when reading display source values.	8
81	Checksum error when reading display text character data.	8
82	Checksum error when reading peak/valley source values.	8
90	Checksum error when reading Auxiliary and Floating point variables from on board FLASH	9
91	Checksum error when reading user memory from on board FLASH	9
92	Checksum error when reading user memory band parameters	9
93	Checksum error when reading Model selection byte.	9
94	Checksum error when reading serial configuration for macro master mode.	9

100	Checksum error when reading advanced setpoint source values.	10
101	Checksum error when reading advanced setpoint control.	10
102	Checksum error when reading advanced setpoint data.	10
103	Checksum error when reading advanced setpoint hysteresis data.	10
104	Checksum error when reading advanced setpoint delay_on_make data.	10
105	Checksum error when reading advanced setpoint delay_on_break data.	10
106	Checksum error when reading advanced setpoint delay mode data.	10
107	Checksum error when reading advanced setpoint tracking data.	10
108	Checksum error when reading advanced setpoint trigger/reset values.	10
109	Checksum error when reading advanced setpoint reset values.	10
110	Checksum error when reading advanced setpoint reset destination values.	10
120	Checksum error when reading PID span data	12
121	Checksum error when reading PID gain data	12
122	Checksum error when reading PID integral time data	12
123	Checksum error when reading PID derivative time data	12
124	Checksum error when reading PID anti reset windup data.	12
125	Checksum error when reading PID cycle time data.	12
126	Checksum error when reading PID saturation high data	12
127	Checksum error when reading PID saturation low data	12
128	Checksum error when reading PID alpha constant data	12
129	Checksum error when reading PID setpoint weighting B data	12
130	Checksum error when reading PID setpoint weighting C data	12
131	Checksum error when reading PID N derivative data	12
150	Checksum error when reading com's timeout data	15
151	Checksum error when reading linearization table data	15
152	Checksum error when reading brightness.	15
153	Checksum error when reading manual display memory from on board FLASH	15
200	No real time clock device detected but internal data memory has been installed. This combination is not a standard option and suggests that either the RTC device is installed but not operating correctly, or there is no RTC device installed (but data logging memory has been installed).	20
201	Oscillator error on internal real time clock. This means that the crystal oscillator on the real time clock chip is either not currently going, or (more likely) it has stopped at some stage since last powered up. The most common cause of this fault is that the small button cell battery on the main board is flat. Call Define Instruments Ltd. to find out how to replace the battery.	20

Error Groups

To help identify what effect a certain error will have on the operation of the Zen IoT, all errors have been categorized into different error groups. These are general groups and the user should still look firstly at the specific error number, as explained in the Error table above.

Error Group	Description of Error Group
1	These are critical errors of the highest priority. In some cases these errors may be hardware errors which need to be returned to Define Instruments Ltd. for service. In other cases they are primary configuration registers which effect the entire operation of the controller.
2	This group of errors relates to the configuration of primary analogue input functions. An error in these parameters will potentially effect the scaling of the 16 analogue input channels so the channel results can no longer be trusted and controller should be re-configured by the user.
3	This group of errors relates to the configuration of the output controllers. An error in these parameters will potentially effect the operation of relays and other outputs. The output controllers should be re-configured by the user.
4	This group of errors relates to the configuration of the analogue output channels. An error in these parameters will potentially effect the operation of the analogue outputs so the analogue outputs settings should be checked by the user.
5	This group of errors relates to the configuration of the 4 digital input channels. An error in these parameters could potentially change the scaling of these channels or in counter mode the count value at power-up could be lost.
6	This group of errors relates to the configuration of the totalizer channels. An error in these parameters could potentially change the operation of the totalizers or the total value at power-up could be lost.
7	This group of errors relates to the configuration of the data logging section. An error in these parameters could potentially change the operation of the data logger and may also cause data corruption in the data log memory.
8	This group of errors relates to the configuration of the display. An error in these parameters could potentially change the parameters on the display. It should not effect the main operation of the controller but may create confusion if data is being retrieved in some serial modes.
9	This group of errors relates to parameters which are normally used with a Macro. An error in these parameters could potentially change the way a macro operates or cause loss of stored parameters used by the macro. To recover from these faults it is highly recommended that the macro be re-loaded.
10	This group of errors relates to the configuration of the advanced setpoints. An error in these parameters will potentially effect the operation of advanced relays and other advanced relay functions. The configuration of the advanced setpoints should be re-configured by the user.
12	This group of errors relates to the configuration of the PID parameters. An error in these parameters will potentially effect the operation of the PID loops. The configuration of the PID should be re-configured by the user.
15	This group of errors relates to the configuration of the various miscellaneous functions. If post linearization tables are being used then these should be re-loaded. In general these errors are probably not critical to the operation of the controller but it is recommended that they are checked.
20	This group of errors relates to errors in the real time clock. In general operation of the Zen IoT can continue but all time and date values will be incorrect. This is generally only a problem when using data logging.

2.13.4 Register 239 - Alarm Status

Register 239 is a 32-bit register that contains flags to indicate the status of the 16 advanced setpoints. In the normal mode of operation, the status of each setpoint is controlled by the controller, based on a comparison between the input value and the setpoint value. Each setpoint can be individually placed into remote mode by setting the appropriate mode control bit (bits 16 to 31).

In remote mode, the input value and setpoint values have no effect on the setpoint status. Instead, the status of the setpoint is controlled directly by setting or clearing the appropriate status bit. This can be done from the serial port or from a macro.

Note: See Alarm Status 16 bit for information about accessing these status bits via two 16 bit registers

instead of one 32 bit register.

See also
Alarm Status Read

Alarm Status Write

Setpoint Status Flags

2.13.4.1 Alarm Status Read

The following table shows the function of each bit when **reading** the alarm status.

Bit	Name	Description	Function
0	SP1	This flag shows/controls the status of setpoint 1	0 = OFF 1 = ON
1	SP2	This flag shows/controls the status of setpoint 2	0 = OFF 1 = ON
2	SP3	This flag shows/controls the status of setpoint 3	0 = OFF 1 = ON
3	SP4	This flag shows/controls the status of setpoint 4	0 = OFF 1 = ON
4	SP5	This flag shows/controls the status of setpoint 5	0 = OFF 1 = ON
5	SP6	This flag shows/controls the status of setpoint 6	0 = OFF 1 = ON
6	SP7	This flag shows/controls the status of setpoint 7	0 = OFF 1 = ON
7	SP8	This flag shows/controls the status of setpoint 8	0 = OFF 1 = ON
8	SP9	This flag shows/controls the status of setpoint 9	0 = OFF 1 = ON
9	SP10	This flag shows/controls the status of setpoint 10	0 = OFF 1 = ON
10	SP11	This flag shows/controls the status of setpoint 11	0 = OFF 1 = ON
11	SP12	This flag shows/controls the status of setpoint 12	0 = OFF 1 = ON
12	SP13	This flag shows/controls the status of setpoint 13	0 = OFF 1 = ON
13	SP14	This flag shows/controls the status of setpoint 14	0 = OFF 1 = ON
14	SP15	This flag shows/controls the status of setpoint 15	0 = OFF 1 = ON
15	SP16	This flag shows/controls the status of setpoint 16	0 = OFF 1 = ON
16	SP1_REMOTE	When this bit is ON setpoint 1 is in remote mode. (If setpoint 1 is operating in PID mode then PID 1 is in manual mode)	0 = Normal Mode (auto mode for PID operation) 1 = Remote Mode (manual or zero output mode for PID operation)
17	SP2_REMOTE.	When this bit is ON setpoint 2 is in remote mode. (If setpoint 1 is operating in PID mode then PID 2 is in manual mode)	0 = Normal Mode (auto mode for PID operation) 1 = Remote Mode (manual or zero output mode for PID operation)

18	SP3_REMOTE.	When this bit is ON setpoint 3 is in remote mode. (If setpoint 1 is operating in PID mode then PID 3 is in manual mode)	0 = Normal Mode (auto mode for PID operation) 1 = Remote Mode (manual or zero output mode for PID operation)
19	SP4_REMOTE	When this bit is ON setpoint 4 is in remote mode. (If setpoint 1 is operating in PID mode then PID 4 is in manual mode)	0 = Normal Mode (auto mode for PID operation) 1 = Remote Mode (manual or zero output mode for PID operation)
20	SP5_REMOTE	When this bit is ON setpoint 5 is in remote mode. (If setpoint 1 is operating in PID mode then PID 5 is in manual mode)	0 = Normal Mode (auto mode for PID operation) 1 = Remote Mode (manual or zero output mode for PID operation)
21	SP6_REMOTE	When this bit is ON setpoint 6 is in remote mode.	0 = Normal Mode 1 = Remote Mode
22	SP7_REMOTE	When this bit is ON setpoint 7 is in remote mode.	0 = Normal Mode 1 = Remote Mode
23	SP8_REMOTE	When this bit is ON setpoint 8 is in remote mode.	0 = Normal Mode 1 = Remote Mode
24	SP9_REMOTE	When this bit is ON setpoint 9 is in remote mode.	0 = Normal Mode 1 = Remote Mode
25	SP10_REMOTE	When this bit is ON setpoint 10 is in remote mode.	0 = Normal Mode 1 = Remote Mode
26	SP11_REMOTE	When this bit is ON setpoint 11 is in remote mode.	0 = Normal Mode 1 = Remote Mode
27	SP12_REMOTE	When this bit is ON setpoint 12 is in remote mode.	0 = Normal Mode 1 = Remote Mode
28	SP13_REMOTE	When this bit is ON setpoint 13 is in remote mode.	0 = Normal Mode 1 = Remote Mode
29	SP14_REMOTE	When this bit is ON setpoint 14 is in remote mode.	0 = Normal Mode 1 = Remote Mode
30	SP15_REMOTE	When this bit is ON setpoint 15 is in remote mode.	0 = Normal Mode 1 = Remote Mode
31	SP16_REMOTE	When this bit is ON setpoint 16 is in remote mode.	0 = Normal Mode 1 = Remote Mode

NOTE: Bits 0 to 15 indicate the setpoint status only, not the relay status. Setpoint timer and manual reset settings could cause the relay status to be different from the setpoint status. For relay status, see register number 4099.

See also
Alarm Status Write

2.13.4.2 Alarm Status Write

The following table shows the function of each bit when **writing** to the alarm status.

Bit	Name	Description	Function
0	SP1	This flag controls the status of setpoint 1	Setpoint Mode (with SP1_REMOTE on) 0 = OFF 1 = ON PID Mode (with SP1_REMOTE on) 0 = Output off 1 = Manual output
1	SP2	This flag controls the status of setpoint 2	Setpoint Mode (with SP2_REMOTE on) 0 = OFF 1 = ON PID Mode (with SP2_REMOTE on) 0 = Output off 1 = Manual output
2	SP3	This flag controls the status of setpoint 3	Setpoint Mode (with SP3_REMOTE on) 0 = OFF 1 = ON PID Mode (with SP3_REMOTE on) 0 = Output off 1 = Manual output
3	SP4	This flag controls the status of setpoint 4	Setpoint Mode (with SP4_REMOTE on) 0 = OFF 1 = ON PID Mode (with SP4_REMOTE on) 0 = Output off 1 = Manual output
4	SP5	This flag controls the status of setpoint 5	Setpoint Mode (with SP5_REMOTE on) 0 = OFF 1 = ON PID Mode (with SP5_REMOTE on) 0 = Output off 1 = Manual output
5	SP6	This flag controls the status of setpoint 6	0 = OFF 1 = ON
6	SP7	This flag controls the status of setpoint 7	0 = OFF 1 = ON
7	SP8	This flag controls the status of setpoint 8	0 = OFF 1 = ON
8	SP9	This flag controls the status of setpoint 9	0 = OFF 1 = ON
9	SP10	This flag controls the status of setpoint 10	0 = OFF 1 = ON
10	SP11	This flag controls the status of setpoint 11	0 = OFF 1 = ON
11	SP12	This flag controls the status of setpoint 12	0 = OFF 1 = ON
12	SP13	This flag controls the status of setpoint 13	0 = OFF 1 = ON
13	SP14	This flag controls the status of setpoint 14	0 = OFF 1 = ON
14	SP15	This flag controls the status of setpoint 15	0 = OFF 1 = ON
15	SP16	This flag controls the status of setpoint 16	0 = OFF 1 = ON

16	SP1_REMOTE	Setting this bit to ON places setpoint 1 in remote mode. (If SP1 is operating in PID mode then setting this bit causes the PID1 to operate in manual or off mode and clearing this bit causes PID1 to revert back to auto mode)	0 = Normal Mode (auto mode for PID operation) 1 = Remote Mode (manual or zero output mode for PID operation)
17	SP2_REMOTE.	Setting this bit to ON places setpoint 2 in remote mode. (If SP2 is operating in PID mode then setting this bit causes the PID2 to operate in manual or off mode and clearing this bit causes PID2 to revert back to auto mode)	0 = Normal Mode (auto mode for PID operation) 1 = Remote Mode (manual or zero output mode for PID operation)
18	SP3_REMOTE.	Setting this bit to ON places setpoint 3 in remote mode. (If SP3 is operating in PID mode then setting this bit causes the PID3 to operate in manual or off mode and clearing this bit causes PID3 to revert back to auto mode)	0 = Normal Mode (auto mode for PID operation) 1 = Remote Mode (manual or zero output mode for PID operation)
19	SP4_REMOTE	Setting this bit to ON places setpoint 4 in remote mode. (If SP4 is operating in PID mode then setting this bit causes the PID4 to operate in manual or off mode and clearing this bit causes PID4 to revert back to auto mode)	0 = Normal Mode (auto mode for PID operation) 1 = Remote Mode (manual or zero output mode for PID operation)
20	SP5_REMOTE	Setting this bit to ON places setpoint 5 in remote mode. (If SP5 is operating in PID mode then setting this bit causes the PID5 to operate in manual or off mode and clearing this bit causes PID5 to revert back to auto mode)	0 = Normal Mode (auto mode for PID operation) 1 = Remote Mode (manual or zero output mode for PID operation)
21	SP6_REMOTE	Setting this bit to ON places setpoint 6 in remote mode.	0 = Normal Mode 1 = Remote Mode
22	SP7_REMOTE	Setting this bit to ON places setpoint 7 in remote mode.	0 = Normal Mode 1 = Remote Mode
23	SP8_REMOTE	Setting this bit to ON places setpoint 8 in remote mode.	0 = Normal Mode 1 = Remote Mode
24	SP9_REMOTE	Setting this bit to ON places setpoint 9 in remote mode.	0 = Normal Mode 1 = Remote Mode
25	SP10_REMOTE	Setting this bit to ON places setpoint 10 in remote mode.	0 = Normal Mode 1 = Remote Mode
26	SP11_REMOTE	Setting this bit to ON places setpoint 11 in remote mode.	0 = Normal Mode 1 = Remote Mode
27	SP12_REMOTE	Setting this bit to ON places setpoint 12 in remote mode.	0 = Normal Mode 1 = Remote Mode
28	SP13_REMOTE	Setting this bit to ON places setpoint 13 in remote mode.	0 = Normal Mode 1 = Remote Mode
29	SP14_REMOTE	Setting this bit to ON places setpoint 14 in remote mode.	0 = Normal Mode 1 = Remote Mode
30	SP15_REMOTE	Setting this bit to ON places setpoint 15 in remote mode.	0 = Normal Mode 1 = Remote Mode
31	SP16_REMOTE	Setting this bit to ON places setpoint 16 in remote mode.	0 = Normal Mode 1 = Remote Mode

NOTE: Bits 0 to 15 indicate the setpoint status only, not the relay status. Setpoint timer and manual reset settings could cause the relay status to be different from the setpoint status. For relay status, see register number 4099.

If setpoints are configured to operate in PID mode (setpoints 1 - 5 only) they will start up in PID auto mode at power on. Each PID loop can then be switched to operate in manual mode or PID off mode by setting the appropriate |SPx_REMOTE bit and |SPx bits in the ALARM_STATUS register.

The following table shows the different options available.

|SPx_REMOTE bit	|SPx bit	PID function
Off (0)	Don't care	Auto Mode - normal mode of operation where PID output is under control of the PID algorithm. (Default mode at power on)
On(1)	On(1)	Manual Output Mode - manual mode of operation where PID output is controlled by manually writing to PIDx_MAN_OUT register.
On(1)	Off (0)	Off Mode - PID algorithm and output are turned off, PID registers (PIDx_ERRD_OLD, PIDx_INTEGRAL_TERM, PIDx_OUTPUT) are reset to zero.

See also
Alarm Status Read

2.13.4.3 Alarm Status 16 bit

When using Modbus communications it is sometimes difficult to access 32 bit registers so the Alarm Status register can also be accessed by two 16 bit registers. Registers 4507 and 4508 duplicate the functions of the Alarm Status register 239 but allow it to be accessed in two 16 bit words instead of one 32 bit register.

Register 4507 - Alarm Status Low
Register 4507 is a 16 bit register which allows access to status/control flags for setpoints 1 - 8. The bit functions for register 4507 are shown below.

Bit	Name	Description	Function
0	SP1	This flag shows/controls the status of setpoint 1	0 = OFF 1 = ON
1	SP2	This flag shows/controls the status of setpoint 2	0 = OFF 1 = ON
2	SP3	This flag shows/controls the status of setpoint 3	0 = OFF 1 = ON
3	SP4	This flag shows/controls the status of setpoint 4	0 = OFF 1 = ON
4	SP5	This flag shows/controls the status of setpoint 5	0 = OFF 1 = ON
5	SP6	This flag shows/controls the status of setpoint 6	0 = OFF 1 = ON

6	SP7	This flag shows/controls the status of setpoint 7	0 = OFF 1 = ON
7	SP8	This flag shows/controls the status of setpoint 8	0 = OFF 1 = ON
8	SP1_REMOTE	Setting this bit to ON places setpoint 1 in remote mode. (If SP1 is operating in PID mode then setting this bit causes the PID1 to operate in manual mode and clearing this bit causes PID1 to revert back to auto mode)	0 = Normal Mode (auto mode for PID operation) 1 = Remote Mode (manual mode for PID operation)
9	SP2_REMOTE.	Setting this bit to ON places setpoint 2 in remote mode. (If SP2 is operating in PID mode then setting this bit causes the PID2 to operate in manual mode and clearing this bit causes PID2 to revert back to auto mode)	0 = Normal Mode (auto mode for PID operation) 1 = Remote Mode (manual mode for PID operation)
10	SP3_REMOTE.	Setting this bit to ON places setpoint 3 in remote mode. (If SP3 is operating in PID mode then setting this bit causes the PID3 to operate in manual mode and clearing this bit causes PID3 to revert back to auto mode)	0 = Normal Mode (auto mode for PID operation) 1 = Remote Mode (manual mode for PID operation)
11	SP4_REMOTE	Setting this bit to ON places setpoint 4 in remote mode. (If SP4 is operating in PID mode then setting this bit causes the PID4 to operate in manual mode and clearing this bit causes PID4 to revert back to auto mode)	0 = Normal Mode (auto mode for PID operation) 1 = Remote Mode (manual mode for PID operation)
12	SP5_REMOTE	Setting this bit to ON places setpoint 5 in remote mode. (If SP5 is operating in PID mode then setting this bit causes the PID5 to operate in manual mode and clearing this bit causes PID5 to revert back to auto mode)	0 = Normal Mode (auto mode for PID operation) 1 = Remote Mode (manual mode for PID operation)
13	SP6_REMOTE	Setting this bit to ON places setpoint 6 in remote mode.	0 = Normal Mode 1 = Remote Mode
14	SP7_REMOTE	Setting this bit to ON places setpoint 7 in remote mode.	0 = Normal Mode 1 = Remote Mode
15	SP8_REMOTE	Setting this bit to ON places setpoint 8 in remote mode.	0 = Normal Mode 1 = Remote Mode

Register 4508 - Alarm Status High

Register 4508 is a 16 bit register which allows access to status/control flags for setpoints 9-16. The bit functions for register 4508 are shown below.

Bit	Name	Description	Function
0	SP9	This flag shows/controls the status of setpoint 9	0 = OFF 1 = ON
1	SP10	This flag shows/controls the status of setpoint 10	0 = OFF 1 = ON
2	SP11	This flag shows/controls the status of setpoint 11	0 = OFF 1 = ON
3	SP12	This flag shows/controls the status of setpoint 12	0 = OFF 1 = ON
4	SP13	This flag shows/controls the status of setpoint 13	0 = OFF 1 = ON
5	SP14	This flag shows/controls the status of setpoint 14	0 = OFF 1 = ON
6	SP15	This flag shows/controls the status of setpoint 15	0 = OFF 1 = ON
7	SP16	This flag shows/controls the status of setpoint 16	0 = OFF 1 = ON

8	SP9_REMOTE	Setting this bit to ON places setpoint 9 in remote mode.	0 = Normal Mode 1 = Remote Mode
9	SP10_REMOTE	Setting this bit to ON places setpoint 10 in remote mode.	0 = Normal Mode 1 = Remote Mode
10	SP11_REMOTE	Setting this bit to ON places setpoint 11 in remote mode.	0 = Normal Mode 1 = Remote Mode
11	SP12_REMOTE	Setting this bit to ON places setpoint 12 in remote mode.	0 = Normal Mode 1 = Remote Mode
12	SP13_REMOTE	Setting this bit to ON places setpoint 13 in remote mode.	0 = Normal Mode 1 = Remote Mode
13	SP14_REMOTE	Setting this bit to ON places setpoint 14 in remote mode.	0 = Normal Mode 1 = Remote Mode
14	SP15_REMOTE	Setting this bit to ON places setpoint 15 in remote mode.	0 = Normal Mode 1 = Remote Mode
15	SP16_REMOTE	Setting this bit to ON places setpoint 16 in remote mode.	0 = Normal Mode 1 = Remote Mode

2.14 Timers

The timer registers shown below are software timers that are managed by the operating system of the controller and run continuously in the background with no user intervention required. Apart from being automatically incremented, they are not used by the operating system or any other standard functions in the controller, so you have complete freedom to use them as required.

The timer count is incremented by the operating system at set intervals (usually 0.1 seconds). Timers can be read or written to from the macro or via the serial port. Timer values will not change within a macro loop (with the exception of the Modbus master macro).

Name	Description	Symbol Type	Register Number	Memory Type
SHORT_TIMER1	16-bit timer counts up every 0.1 seconds (109 minutes). It can be set/reset by macro.	U_16	4111	RAM
SHORT_TIMER2	16-bit timer counts up every 0.1 seconds (109 minutes). It can be set/reset by macro.	U_16	4112	RAM
SHORT_TIMER3	16-bit timer counts up every 0.1 seconds (109 minutes). It can be set/reset by macro.	U_16	4113	RAM
SHORT_TIMER4	16-bit timer counts up every 0.1 seconds (109 minutes). It can be set/reset by macro.	U_16	4114	RAM
FAST_TIMER1	16-bit timer counts up every 0.01 seconds (10.9 minutes). It can be set/reset by macro.	U_16	4308	RAM
FAST_TIMER2	16-bit timer counts up every 0.01 seconds (10.9 minutes). It can be set/reset by macro.	U_16	4309	RAM
FAST_TIMER3	16-bit timer counts up every 0.01 seconds (10.9 minutes). It can be set/reset by macro.	U_16	4310	RAM
FAST_TIMER4	16-bit timer counts up every 0.01 seconds (10.9 minutes). It can be set/reset by macro.	U_16	4311	RAM
TIMER1	32-bit timer counts up every 0.1 seconds (13.6 years) can be set/reset by macro.	U_32	143	RAM
TIMER2	32-bit timer counts up every 0.1 seconds (13.6 years) can be set/reset by macro.	U_32	145	RAM
TIMER3	32-bit timer counts up every 0.1 seconds (13.6 years) can be set/reset by macro.	U_32	147	RAM
TIMER4	32-bit timer counts up every 0.1 seconds (13.6 years) can be set/reset by macro.	U_32	149	RAM

NOTE: If a timer reaches its maximum count it overflows to 0 without any warning so you must ensure that you use the correct timer (long or short) for the task.

2.15 Totalizers

A totalizer is a user selectable software function of the controller that converts an input rate to an input total over time. Each Zen IoT controller has 10 independent totalizers suitable for a wide variety of totaling applications. Totals can be reset using one of a number of methods. Setpoints can be used to reset a sub-total and increment a grand total.

For example:
You have a settling tank being filled with water. The flow rate is metered and input to a Zen IoT controller. The flow rate indicates the speed at which the volume of water travels past a set point, but not the total volume accumulated in the tank. The controller's totalizer performs this function and provides you with the total amount of water currently in the tank. This then allows you to make control decisions, such as when to turn the tap off before the tank overflows!

To set up a totalizer channel, you need to do the following;
1) Load the DATA_SOURCE_TOTALx register with the address (i.e. register number) of the register which holds the flow rate value. For example if your flow rate was coming from the COUNTER_a register (525) then you would set DATA_SOURCE_TOTALx to 525.

2) Setup the DISPLAY_FORMAT_TOTALx with the decimal point format that want the final total to be displayed with. For example if you want the total to be displayed with 1 decimal points (i.e. X.X), you would set DISPLAY_FORMAT_TOTALx to a value of 1. If you don't want any decimal points then set DISPLAY_FORMAT_TOTALx to 0.(See Display Format for more info on decimal points)

3) You can also load TEXT_CHARACTER_TOTALx with ASCII value from 30 to 127 if you want the final total to be displayed with a trailing text character to display units. For example if you wanted the total to be displayed as "X.XXXL" you would set TEXT_CHARACTER_TOTALx to 76 (the ASCII value for "L" is 76). If you don't want any trailing text character then set TEXT_CHARACTER_TOTALx to 0.

4) Next you need to tell the totalizer how to calculate your total based on a theoretical input flow rate, and the resulting total you would expect to see after a certain amount of time has elapsed. This is done using the registers INPUT_RATEx, FINAL_TOTALx and TOTALx_TIME. The example below shows how this is done.

Lets assume our flow rate is in liters per minute, and we want our final total value to display in thousands of liters with 1 decimal place. That means:

With a flow rate of 1000 liters, our total should read 1.0 after 1 minute.
=> INPUT_RATEx = 1000
=> FINAL_TOTALx = 10 (this is only a 16 bit integer so *just ignore the decimal point for now and enter the digits you expect in your final total*)

To setup the TOTALx_TIME value, see Totalizer Time Period and Rollover. This register controls the time period used for the calculation of K factor and also allows you to select if you want the total to rollover when it exceeds its maximum value. So assuming we want rollover to be active:
=> TOTALx_TIME = 18 (or 0x12 in hex)

You could setup the above registers in different ways and still get the same result. For example we could have said;
With a flow rate of 10000 liters, our total should read 100.0 after 10 minutes.
=> INPUT_RATEx = 10000
=> FINAL_TOTALx = 1000
=> TOTALx_TIME = 19 (or 0x13 in hex)

The result would be exactly the same so provided the relationships between the 3 parameters are correct the calculated K factor will be correct. However, there are some limitations that have to be observed when choosing values for these parameters. See Totalizer Limitations below.

5) Next you can specify the minimum flow rate you want the totalizer to work with by entering a value in CUTOFFx. Whenever the input flow rate falls below the value of CUTOFFx, the totalizer will stop totalizing.

6) Finally, you can enter a text string up to 8 characters long into TOTALx_TEXT to identify the what the total represents. This last step is optional and if you don't do this the default text will be "Total x".

Totalizer Limitations
Due to limitations in the math's calculations, some combinations of INPUT_RATEx, FINAL_TOTALx and TOTALx_TIME may cause errors. So the following limitations apply;

INPUT_RATEx must be > 0 (range 1 - 65535)
FINAL_TOTALx must be > 0 (range 1 - 32767)

With time = 1 week,
INPUT_RATEx/FINAL_TOTALx must be < 3550

With time = 1 day,
INPUT_RATEx/FINAL_TOTALx must be < 24855

With time = 10 hours, no extra conditions apply.

With time = 1 hour,
FINAL_TOTALx/INPUT_RATEx must be < 3600

With time = 10 minutes,
FINAL_TOTALx/INPUT_RATEx must be < 600

With time = 1 minute,
FINAL_TOTALx/INPUT_RATEx must be < 60

With time = 10 seconds,
FINAL_TOTALx/INPUT_RATEx must be < 10

With time = 1 second,
INPUT_RATEx must be >= FINAL_TOTALx

See also
Total 1

Total 2

Total 3

Total 4

Total 5

Total 6

Total 7

Total 8

Total 9

Total 10

Final Total Value

Input Rate Value

Totalizer Time Period and Rollover

2.15.1 Total 1

Name	Description	Symbol Type	Register Number	Memory Type
TOTAL1	Non-volatile 32-bit register for Totalizer 1 value.	S_32	289	RAM/FLASH
TOTAL1_FLOAT	Non-volatile 32-bit pseudo floating point register for Totalizer 1 value. (See 32-bit Pseudo Floating Point).	PF_32	1569	RAM/FLASH
CUTOFF1	16-bit register. Cutoff value for totalizer 1 (range -32768 to 32767).	S_16	4618	RAM/EEPROM
DATA_SOURCE_TOTAL1	16-bit register holds the register number of the data source for totalizer 1.	U_16	4328	RAM/EEPROM
DISPLAY_FORMAT_TOTAL1	8-bit register controls the display format settings for totalizer 1 (display is in octal format).	O_8	8337	RAM/EEPROM
INPUT_RATE1	16-bit unsigned register. Input rate for K factor calculation totalizer 1. (See Totalizer Limitations)	U_16	4257	RAM/EEPROM
FINAL_TOTAL1	16-bit register. Expected final total value for totalizer 1 after selected time period with specified input rate. (See Totalizer Limitations)	S_16	4608	RAM/EEPROM
TOTAL1_TIME	8-bit register for totalizer 1 time calculation & rollover. (See Totalizer Limitations)	U_8	8473	RAM/EEPROM
TEXT_CHARACTER_TOTAL1	8-bit register holds the ASCII value for the last digit text character for totalizer 1 (0 = no character).	U_8	8391	RAM/EEPROM
TOTAL1_TEXT	Text display for total 1.	L_14	16437	EEPROM
UNITS_TEXT_TOTAL1	Units text for TOTAL1. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17449	EEPROM

See also

K Factor

Input Rate Value

Totalizer Data Source Selection

Totalizer Time Period and Rollover

2.15.2 Total 2

Name	Description	Symbol Type	Register Number	Memory Type
TOTAL2	Non-volatile 32-bit register for totalizer 2 value.	S_32	291	RAM/FLASH
TOTAL2_FLOAT	Non-volatile 32-bit pseudo floating point register for totalizer 2 value. (See 32-bit Pseudo Floating Point).	PF_32	1571	RAM/FLASH
CUTOFF2	16-bit register. Cutoff value for totalizer 2 (range -32768 to 32767).	S_16	4619	RAM/EEPROM
DATA_SOURCE_TOTAL2	16-bit register holds the register number of the data source for totalizer 2.	U_16	4329	RAM/EEPROM
DISPLAY_FORMAT_TOTAL2	8-bit register controls the display format settings for totalizer 2 (display is in octal format).	O_8	8338	RAM/EEPROM
INPUT_RATE2	16-bit unsigned register. Input rate for K factor calculation totalizer 2. (See Totalizer Limitations)	U_16	4258	RAM/EEPROM
FINAL_TOTAL2	16-bit register. Expected final total value for totalizer 2 after selected time period with specified input rate. (See Totalizer Limitations)	S_16	4609	RAM/EEPROM
TOTAL2_TIME	8-bit register for totalizer 2 time calculation & rollover. (See Totalizer Limitations)	U_8	8474	RAM/EEPROM
TEXT_CHARACTER_TOTAL2	8-bit register holds the ASCII value for the last digit text character for totalizer 2 (0 = no character).	U_8	8392	RAM/EEPROM
TOTAL2_TEXT	Text display for total 2.	L_14	16439	EEPROM
UNITS_TEXT_TOTAL2	Units text for TOTAL2. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17451	EEPROM

See also

K Factor

Input Rate Value

Totalizer Data Source Selection

Totalizer Time Period and Rollover

2.15.3 Total 3

Name	Description	Symbol Type	Register Number	Memory Type
TOTAL3	Non-volatile 32-bit register for totalizer 3 value.	S_32	293	RAM/FLASH
TOTAL3_FLOAT	Non-volatile 32-bit pseudo floating point register for totalizer 3 value. (See 32-bit Pseudo Floating Point).	PF_32	1573	RAM/FLASH
CUTOFF3	16-bit register. Cutoff value for totalizer 3 (range -32768 to 32767).	S_16	4620	RAM/EEPROM
DATA_SOURCE_TOTAL3	16-bit register holds the register number of the data source for totalizer 3.	U_16	4330	RAM/EEPROM
DISPLAY_FORMAT_TOTAL3	8-bit register controls the display format settings for totalizer 3 (display is in octal format).	O_8	8339	RAM/EEPROM

INPUT_RATE3	16-bit unsigned register. Input rate for K factor calculation totalizer 3. (See Totalizer Limitations)	U_16	4259	RAM/EEPROM
FINAL_TOTAL3	16-bit register. Expected final total value for totalizer 3 after selected time period with specified input rate. (See Totalizer Limitations)	S_16	4610	RAM/EEPROM
TOTAL3_TIME	8-bit register for totalizer 3 time calculation & rollover. (See Totalizer Limitations)	U_8	8475	RAM/EEPROM
TEXT_CHARACTER_TOTAL3	8-bit register holds the ASCII value for the last digit text character for totalizer 3 (0 = no character).	U_8	8393	RAM/EEPROM
TOTAL3_TEXT	Text display for total 3.	L_14	16441	EEPROM
UNITS_TEXT_TOTAL3	Units text for TOTAL3. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17453	EEPROM

See also

K Factor

Input Rate Value

Totalizer Data Source Selection

Totalizer Time Period and Rollover

2.15.4 Total 4

Name	Description	Symbol Type	Register Number	Memory Type
TOTAL4	Non-volatile 32-bit register for totalizer 4 value.	S_32	295	RAM/FLASH
TOTAL4_FLOAT	Non-volatile 32-bit pseudo floating point register for totalizer 4 value. (see 32-bit Pseudo Floating Point).	PF_32	1575	RAM/FLASH
CUTOFF4	16-bit register. Cutoff value for totalizer 4 (range -32768 to 32767).	S_16	4621	RAM/EEPROM
DATA_SOURCE_TOTAL4	16-bit register holds the register number of the data source for totalizer 4.	U_16	4331	RAM/EEPROM
DISPLAY_FORMAT_TOTAL4	8-bit register controls the display format settings for totalizer 4 (display is in octal format).	O_8	8340	RAM/EEPROM
INPUT_RATE4	16-bit unsigned register. Input rate for K factor calculation totalizer 4. (See Totalizer Limitations)	U_16	4260	RAM/EEPROM
FINAL_TOTAL4	16-bit register. Expected final total value for totalizer 4 after selected time period with specified input rate. (See Totalizer Limitations)	S_16	4611	RAM/EEPROM
TOTAL4_TIME	8-bit register for totalizer 4 time calculation & rollover. (See Totalizer Limitations)	U_8	8476	RAM/EEPROM
TEXT_CHARACTER_TOTAL4	8-bit register holds the ASCII value for the last digit text character for totalizer 4 (0 = no character).	U_8	8394	RAM/EEPROM
TOTAL4_TEXT	Text display for total 4.	L_14	16443	EEPROM
UNITS_TEXT_TOTAL4	Units text for TOTAL4. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17455	EEPROM

See also

K Factor

Input Rate Value

Totalizer Data Source Selection

Totalizer Time Period and Rollover

2.15.5 Total 5

Name	Description	Symbol Type	Register Number	Memory Type
TOTAL5	Non-volatile 32-bit register for totalizer 5 value.	S_32	297	RAM/FLASH
TOTAL5_FLOAT	Non-volatile 32-bit pseudo floating point register for totalizer 5 value. (See 32-bit Pseudo Floating Point).	PF_32	1577	RAM/FLASH
CUTOFF5	16-bit register. Cutoff value for totalizer 5 (range -32768 to 32767).	S_16	4622	RAM/EEPROM
DATA_SOURCE_TOTAL5	16-bit register holds the register number of the data source for totalizer 5.	U_16	4332	RAM/EEPROM
DISPLAY_FORMAT_TOTAL5	8-bit register controls the display format settings for totalizer 5 (display is in octal format).	O_8	8341	RAM/EEPROM
INPUT_RATE5	16-bit unsigned register. Input rate for K factor calculation totalizer 5. (See Totalizer Limitations)	U_16	4261	RAM/EEPROM
FINAL_TOTAL5	16-bit register. Expected final total value for totalizer 5 after selected time period with specified input rate. (See Totalizer Limitations)	S_16	4612	RAM/EEPROM
TOTAL5_TIME	8-bit register for totalizer 5 time calculation & rollover. (See Totalizer Limitations)	U_8	8477	RAM/EEPROM
TEXT_CHARACTER_TOTAL5	8-bit register holds the ASCII value for the last digit text character for totalizer 5 (0 = no character).	U_8	8395	RAM/EEPROM
TOTAL5_TEXT	Text display for total 5.	L_14	16445	EEPROM
UNITS_TEXT_TOTAL5	Units text for TOTAL5. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17457	EEPROM

See also
K Factor

Input Rate Value

Totalizer Data Source Selection

Totalizer Time Period and Rollover

2.15.6 Total 6

Name	Description	Symbol Type	Register Number	Memory Type
TOTAL6	Non-volatile 32-bit register for totalizer 6 value.	S_32	299	RAM/FLASH
TOTAL6_FLOAT	Non-volatile 32-bit pseudo floating point register for totalizer 6 value. (See 32-bit Pseudo Floating Point).	PF_32	1579	RAM/FLASH
CUTOFF6	16-bit register. Cutoff value for totalizer 6 (range -32768 to 32767).	S_16	4623	RAM/EEPROM
DATA_SOURCE_TOTAL6	16-bit register holds the register number of the data source for totalizer 6.	U_16	4333	RAM/EEPROM
DISPLAY_FORMAT_TOTAL6	8-bit register controls the display format settings for totalizer 6 (display is in octal format).	O_8	8342	RAM/EEPROM
INPUT_RATE6	16-bit unsigned register. Input rate for K factor calculation totalizer 6. (See Totalizer Limitations)	U_16	4262	RAM/EEPROM
FINAL_TOTAL6	16-bit register. Expected final total value for totalizer 6 after selected time period with specified input rate. (See Totalizer Limitations)	S_16	4613	RAM/EEPROM
TOTAL6_TIME	8-bit register for totalizer 6 time calculation & rollover. (See Totalizer Limitations)	U_8	8478	RAM/EEPROM
TEXT_CHARACTER_TOTAL6	8-bit register holds the ASCII value for the last digit text character for totalizer 6 (0 = no character).	U_8	8396	RAM/EEPROM
TOTAL6_TEXT	Text display for total 6.	L_14	16447	EEPROM
UNITS_TEXT_TOTAL6	Units text for TOTAL6. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17459	EEPROM

See also

K Factor

Input Rate Value

Totalizer Data Source Selection

Totalizer Time Period and Rollover

2.15.7 Total 7

Name	Description	Symbol Type	Register Number	Memory Type
TOTAL7	Non-volatile 32-bit register for totalizer 7 value.	S_32	301	RAM/FLASH
TOTAL7_FLOAT	Non-volatile 32-bit pseudo floating point register for totalizer 7 value. (See 32-bit Pseudo Floating Point).	PF_32	1581	RAM/FLASH
CUTOFF7	16-bit register. Cutoff value for totalizer 7 (range -32768 to 32767).	S_16	4624	RAM/EEPROM
DATA_SOURCE_TOTAL7	16-bit register holds the register number of the data source for totalizer 7.	U_16	4334	RAM/EEPROM
DISPLAY_FORMAT_TOTAL7	8-bit register controls the display format settings for totalizer 7 (display is in octal format).	O_8	8343	RAM/EEPROM

INPUT_RATE7	16-bit unsigned register. Input rate for K factor calculation totalizer 7. (See Totalizer Limitations)	U_16	4263	RAM/EEPROM
FINAL_TOTAL7	16-bit register. Expected final total value for totalizer 7 after selected time period with specified input rate. (See Totalizer Limitations)	S_16	4614	RAM/EEPROM
TOTAL7_TIME	8-bit register for totalizer 7 time calculation & rollover. (See Totalizer Limitations)	U_8	8479	RAM/EEPROM
TEXT_CHARACTER_TOTAL7	8-bit register holds the ASCII value for the last digit text character for totalizer 7 (0 = no character).	U_8	8397	RAM/EEPROM
TOTAL7_TEXT	Text display for total 7.	L_14	16449	EEPROM
UNITS_TEXT_TOTAL7	Units text for TOTAL7. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17461	EEPROM

See also

K Factor

Input Rate Value

Totalizer Data Source Selection

Totalizer Time Period and Rollover

2.15.8 Total 8

Name	Description	Symbol Type	Register Number	Memory Type
TOTAL8	Non-volatile 32-bit register for totalizer 8 value.	S_32	303	RAM/FLASH
TOTAL8_FLOAT	Non-volatile 32-bit pseudo floating point register for totalizer 8 value. (See 32-bit Pseudo Floating Point).	PF_32	1583	RAM/FLASH
CUTOFF8	16-bit register. Cutoff value for totalizer 8 (range -32768 to 32767).	S_16	4625	RAM/EEPROM
DATA_SOURCE_TOTAL8	16-bit register holds the register number of the data source for totalizer 8.	U_16	4335	RAM/EEPROM
DISPLAY_FORMAT_TOTAL8	8-bit register controls the display format settings for totalizer 8 (display is in octal format).	O_8	8344	RAM/EEPROM
INPUT_RATE8	16-bit unsigned register. Input rate for K factor calculation totalizer 8. (See Totalizer Limitations)	U_16	4264	RAM/EEPROM
FINAL_TOTAL8	16-bit register. Expected final total value for totalizer 8 after selected time period with specified input rate. (See Totalizer Limitations)	S_16	4615	RAM/EEPROM
TOTAL8_TIME	8-bit register for totalizer 8 time calculation & rollover. (See Totalizer Limitations)	U_8	8480	RAM/EEPROM
TEXT_CHARACTER_TOTAL8	8-bit register holds the ASCII value for the last digit text character for totalizer 8 (0 = no character).	U_8	8398	RAM/EEPROM
TOTAL8_TEXT	Text display for total 8.	L_14	16451	EEPROM
UNITS_TEXT_TOTAL8	Units text for TOTAL8. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17463	EEPROM

See also

K Factor

Input Rate Value

Totalizer Data Source Selection

Totalizer Time Period and Rollover

2.15.9 Total 9

Name	Description	Symbol Type	Register Number	Memory Type
TOTAL9	Non-volatile 32-bit register for totalizer 9 value.	S_32	305	RAM/FLASH
TOTAL9_FLOAT	Non-volatile 32-bit pseudo floating point register for totalizer 9 value. (See 32-bit Pseudo Floating Point).	PF_32	1585	RAM/FLASH
CUTOFF9	16-bit register. Cutoff value for totalizer 9 (range -32768 to 32767).	S_16	4626	RAM/EEPROM
DATA_SOURCE_TOTAL9	16-bit register holds the register number of the data source for totalizer 9.	U_16	4336	RAM/EEPROM
DISPLAY_FORMAT_TOTAL9	8-bit register controls the display format settings for totalizer 9 (display is in octal format).	O_8	8345	RAM/EEPROM
INPUT_RATE9	16-bit unsigned register. Input rate for K factor calculation totalizer 9. (See Totalizer Limitations)	U_16	4265	RAM/EEPROM
FINAL_TOTAL9	16-bit register. Expected final total value for totalizer 9 after selected time period with specified input rate. (See Totalizer Limitations)	S_16	4616	RAM/EEPROM
TOTAL9_TIME	8-bit register for totalizer 9 time calculation & rollover. (See Totalizer Limitations)	U_8	8481	RAM/EEPROM
TEXT_CHARACTER_TOTAL9	8-bit register holds the ASCII value for the last digit text character for totalizer 9 (0 = no character).	U_8	8399	RAM/EEPROM
TOTAL9_TEXT	Text display for total 9.	L_14	16453	EEPROM
UNITS_TEXT_TOTAL9	Units text for TOTAL9. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17465	EEPROM

See also
K Factor

Input Rate Value

Totalizer Data Source Selection

Totalizer Time Period and Rollover

2.15.10 Total 10

Name	Description	Symbol Type	Register Number	Memory Type
TOTAL10	Non-volatile 32-bit register for totalizer 10 value.	S_32	307	RAM/FLASH
TOTAL10_FLOAT	Non-volatile 32-bit pseudo floating point register for totalizer 10 value. (See 32-bit Pseudo Floating Point).	PF_32	1587	RAM/FLASH
CUTOFF10	16-bit register. Cutoff value for totalizer 10 (range -32768 to 32767).	S_16	4627	RAM/EEPROM
DATA_SOURCE_TOTAL10	16-bit register holds the register number of the data source for totalizer 10.	U_16	4337	RAM/EEPROM
DISPLAY_FORMAT_TOTAL10	8-bit register controls the display format settings for totalizer 10 (display is in octal format).	O_8	8346	RAM/EEPROM
INPUT_RATE10	16-bit unsigned register. Input rate for K factor calculation totalizer 10. (See Totalizer Limitations)	U_16	4266	RAM/EEPROM
FINAL_TOTAL10	16-bit register. Expected final total value for totalizer 10 after selected time period with specified input rate. (See Totalizer Limitations)	S_16	4617	RAM/EEPROM
TOTAL10_TIME	8-bit register for totalizer 10 time calculation & rollover. (See Totalizer Limitations)	U_8	8482	RAM/EEPROM
TEXT_CHARACTER_TOTAL10	8-bit register holds the ASCII value for the last digit text character for totalizer 10 (0 = no character).	U_8	8400	RAM/EEPROM
TOTAL10_TEXT	Text display for total 10.	L_14	16455	EEPROM
UNITS_TEXT_TOTAL10	Units text for TOTAL10. (Note: this is a storage register used by external applications. It is not shown on the standard display.)	L_14	17467	EEPROM

See also
K Factor

Input Rate Value

Totalizer Data Source Selection

Totalizer Time Period and Rollover

2.15.11 Final Total Vaue

The registers 4608 to 4617 are used to calculate the totalizer K factor. They are used by the controller, in conjunction with Input rate and Totalizer Time to calculate the true K factor. These registers are loaded with the desired value of the totalizer after the selected rate time has elapsed with the specified input rate value.

NOTE: Registers 4608 to 4617 are not the true K factor value but are only used in the calculation of the K factor. The actual value of the K factor is calculated by the controller.

NOTE: Due to constraints in the math's calculations, limits on INPUT_RATEx and FINAL_TOTALx values will vary with the time selection chosen for TOTALx_TIME. See Totalizer Limitations for more information).

See Also
Input Rate Value

Totalizer Time Period and Rollover

2.15.12 Input Rate Value

Registers 4257 to 4266 are 16-bit unsigned registers that hold the numeric value for the input rate used during the totalizer calibration procedure.

NOTE: Due to constraints in the math's calculations, limits on INPUT_RATEx and FINAL_TOTALx values will vary with the time selection chosen for TOTALx_TIME. See Totalizer Limitations for more information).

See Also
Final Total Value

Totalizer Time Period and Rollover

2.15.13 Totalizer Data Source Selection

Registers 4328 to 4337 are 16-bit registers that specify the data source for the totalizer channels. The number they contain is the ASCII/Modbus register number for the required data source.

NOTE: Only registers that hold integer values can be used as a data source for the display. Floating point and text registers can not be used.

See Also
Common Data Source Registers

2.15.14 Totalizer Time Period and Rollover

The TOTALx_TIME registers 8473 to 8482 are 8-bit registers that control the time period for K factor calculation and rollover features for totalizers 1 to 10 respectively.

The function of each bit is shown as follows:

Bits 0 to 3

Bit Position								Description
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
				0	0	0	0	1 second period for K factor calculations
				0	0	0	1	10 seconds period for K factor calculations
				0	0	1	0	1 minute period for K factor calculations
				0	0	1	1	10 minute period for K factor calculations
				0	1	0	0	1 hour period for K factor calculations
				0	1	0	1	10 hours period for K factor calculations
				0	1	1	0	1 day period for K factor calculations
				0	1	1	1	1 week period for K factor calculations

NOTE: Due to constraints in the math's calculations, limits on INPUT_RATEx and FINAL_TOTALx values will vary with the time selection chosen for TOTALx_TIME. See Totalizer Limitations for more information).

Bit 4 **Totalizer rollover**
0 = inactive
1 = rollover active

Bit 5 to 7 **Unused at present**

See Also
Final Total Value

Input Rate Value

2.16 User

The controller includes a section of memory reserved for the storage of user data. This memory is not used by the operating system or any of the other standard functions in the controller, so you are free to allocate this as required. It is normally used in conjunction with a macro to store data tables or other setup parameters for the macro, but it could equally be used via the serial port.

NOTE 1: Some of this memory is non volatile EEPROM and write restrictions apply. See user memory for more details.

NOTE 2: If your Zen IoT is running with a plugin, then some of these registers maybe be used by the plugin. You should check with Define Instruments Ltd. if you intend to use any of these registers for other purposes. Overwriting registers may cause the plugin to stop functioning correctly.

See also
Auxiliary

Memory

Text Memory

Variables

2.16.1 Auxiliary

Registers 315 to 345 are signed 32-bit auxiliary registers that are intended for use with the macro. They can be used to hold calculated result values that can then be displayed or saved in the data logger. Each auxiliary register also has a pseudo floating point register and a user definable 8 character text string that can be used as a description of the function. Also the Display Format and Text Character for each auxiliary register can be independently setup (see Setup (Auxiliary)).

Auxiliary registers are stored in RAM and also stored in non volatile FLASH memory at power down.

Auxiliary registers are not used by the operating system of the controller or by any other function so they can be used freely in the macro for any purpose.

Name	Description	Symbol Type	Register Number	Memory Type
AUX1	Non-volatile 32-bit auxiliary register.	S_32	315	RAM/FLASH
AUX1_FLOAT	Non-volatile 32-bit pseudo floating point value for auxiliary register 1. (See 32-bit Pseudo Floating Point).	PF_32	1595	RAM/FLASH
AUX2	Non-volatile 32-bit auxiliary register.	S_32	317	RAM/FLASH
AUX2_FLOAT	Non-volatile 32-bit pseudo floating point value for auxiliary register 2. (See 32-bit Pseudo Floating Point).	PF_32	1597	RAM/FLASH
AUX3	Non-volatile 32-bit auxiliary register.	S_32	319	RAM/FLASH
AUX3_FLOAT	Non-volatile 32-bit pseudo floating point value for auxiliary register 3. (See 32-bit Pseudo Floating Point).	PF_32	1599	RAM/FLASH
AUX4	Non-volatile 32-bit auxiliary register.	S_32	321	RAM/FLASH
AUX4_FLOAT	Non-volatile 32-bit pseudo floating point value for auxiliary register 4. (See 32-bit Pseudo Floating Point).	PF_32	1601	RAM/FLASH
AUX5	Non-volatile 32-bit auxiliary register.	S_32	323	RAM/FLASH
AUX5_FLOAT	Non-volatile 32-bit pseudo floating point value for auxiliary register 5. (See 32-bit Pseudo Floating Point).	PF_32	1603	RAM/FLASH
AUX6	Non-volatile 32-bit auxiliary register.	S_32	325	RAM/FLASH
AUX6_FLOAT	Non-volatile 32-bit pseudo floating point value for auxiliary register 6. (See 32-bit Pseudo Floating Point).	PF_32	1605	RAM/FLASH
AUX7	Non-volatile 32-bit auxiliary register.	S_32	327	RAM/FLASH
AUX7_FLOAT	Non-volatile 32-bit pseudo floating point value for auxiliary register 7. (See 32-bit Pseudo Floating Point).	PF_32	1607	RAM/FLASH
AUX8	Non-volatile 32-bit auxiliary register.	S_32	329	RAM/FLASH
AUX8_FLOAT	Non-volatile 32-bit pseudo floating point value for auxiliary register 8. (See 32-bit Pseudo Floating Point).	PF_32	1609	RAM/FLASH
AUX9	Non-volatile 32-bit auxiliary register.	S_32	331	RAM/FLASH
AUX9_FLOAT	Non-volatile 32-bit pseudo floating point value for auxiliary register 9. (See 32-bit Pseudo Floating Point).	PF_32	1611	RAM/FLASH
AUX10	Non-volatile 32-bit auxiliary register.	S_32	333	RAM/FLASH
AUX10_FLOAT	Non-volatile 32-bit pseudo floating point value for auxiliary register 10. (See 32-bit Pseudo Floating Point).	PF_32	1613	RAM/FLASH
AUX11	Non-volatile 32-bit auxiliary register.	S_32	335	RAM/FLASH
AUX11_FLOAT	Non-volatile 32-bit pseudo floating point value for auxiliary register 11. (See 32-bit Pseudo Floating Point).	PF_32	1615	RAM/FLASH
AUX12	Non-volatile 32-bit auxiliary register.	S_32	337	RAM/FLASH
AUX12_FLOAT	Non-volatile 32-bit pseudo floating point value for auxiliary register 12. (See 32-bit Pseudo Floating Point).	PF_32	1617	RAM/FLASH

AUX13	Non-volatile 32-bit auxiliary register.	S_32	339	RAM/FLASH
AUX13_FLOAT	Non-volatile 32-bit pseudo floating point value for auxiliary register 13. (See 32-bit Pseudo Floating Point).	PF_32	1619	RAM/FLASH
AUX14	Non-volatile 32-bit auxiliary register.	S_32	341	RAM/FLASH
AUX14_FLOAT	Non-volatile 32-bit pseudo floating point value for auxiliary register 14. (See 32-bit Pseudo Floating Point).	PF_32	1621	RAM/FLASH
AUX15	Non-volatile 32-bit auxiliary register.	S_32	343	RAM/FLASH
AUX15_FLOAT	Non-volatile 32-bit pseudo floating point value for auxiliary register 15. (See 32-bit Pseudo Floating Point).	PF_32	1623	RAM/FLASH
AUX16	Non-volatile 32-bit auxiliary register.	S_32	345	RAM/FLASH
AUX16_FLOAT	Non-volatile 32-bit pseudo floating point value for auxiliary register 16. (See 32-bit Pseudo Floating Point).	PF_32	1625	RAM/FLASH

See also
Setup (Auxiliary)

Display

2.16.1.1 Setup (Auxiliary)

Each auxiliary register can be formatted to display its own unique decimal point, rounding and trailing text character. Register functions and addresses are shown in the table below.

Name	Description	Symbol Type	Register Number	Memory Type
DISPLAY_FORMAT_AUX1	8-bit register controls the display format settings for auxiliary 1 (displayed in octal format).	O_8	8347	RAM/EEPROM
DISPLAY_FORMAT_AUX2	8-bit register controls the display format settings for auxiliary 2 (displayed in octal format).	O_8	8348	RAM/EEPROM
DISPLAY_FORMAT_AUX3	8-bit register controls the display format settings for auxiliary 3 (displayed in octal format).	O_8	8349	RAM/EEPROM
DISPLAY_FORMAT_AUX4	8-bit register controls the display format settings for auxiliary 4 (displayed in octal format).	O_8	8350	RAM/EEPROM
DISPLAY_FORMAT_AUX5	8-bit register controls the display format settings for auxiliary 5 (displayed in octal format).	O_8	8351	RAM/EEPROM
DISPLAY_FORMAT_AUX6	8-bit register controls the display format settings for auxiliary 6 (displayed in octal format).	O_8	8352	RAM/EEPROM
DISPLAY_FORMAT_AUX7	8-bit register controls the display format settings for auxiliary 7 (displayed in octal format).	O_8	8353	RAM/EEPROM
DISPLAY_FORMAT_AUX8	8-bit register controls the display format settings for auxiliary 8 (displayed in octal format).	O_8	8354	RAM/EEPROM
DISPLAY_FORMAT_AUX9	8-bit register controls the display format settings for auxiliary 9 (displayed in octal format).	O_8	8355	RAM/EEPROM

DISPLAY_FORMAT_AUX10	8-bit register controls the display format settings for auxiliary 10 (displayed in octal format).	O_8	8356	RAM/EEPROM
DISPLAY_FORMAT_AUX11	8-bit register controls the display format settings for auxiliary 11 (displayed in octal format).	O_8	8357	RAM/EEPROM
DISPLAY_FORMAT_AUX12	8-bit register controls the display format settings for auxiliary 12 (displayed in octal format).	O_8	8358	RAM/EEPROM
DISPLAY_FORMAT_AUX13	8-bit register controls the display format settings for auxiliary 13 (displayed in octal format).	O_8	8359	RAM/EEPROM
DISPLAY_FORMAT_AUX14	8-bit register controls the display format settings for auxiliary 14 (displayed in octal format).	O_8	8360	RAM/EEPROM
DISPLAY_FORMAT_AUX15	8-bit register controls the display format settings for auxiliary 15 (displayed in octal format).	O_8	8361	RAM/EEPROM
DISPLAY_FORMAT_AUX16	8-bit register controls the display format settings for auxiliary 16 (displayed in octal format).	O_8	8362	RAM/EEPROM
TEXT_CHARACTER_AUX1	8-bit register holds the ASCII value for the last digit text character for auxiliary 1 (0= no character).	U_8	8401	RAM/EEPROM
TEXT_CHARACTER_AUX2	8-bit register holds the ASCII value for the last digit text character for auxiliary 2 (0= no character).	U_8	8402	RAM/EEPROM
TEXT_CHARACTER_AUX3	8-bit register holds the ASCII value for the last digit text character for auxiliary 3 (0= no character).	U_8	8403	RAM/EEPROM
TEXT_CHARACTER_AUX4	8-bit register holds the ASCII value for the last digit text character for auxiliary 4 (0= no character).	U_8	8404	RAM/EEPROM
TEXT_CHARACTER_AUX5	8-bit register holds the ASCII value for the last digit text character for auxiliary 5 (0= no character).	U_8	8405	RAM/EEPROM
TEXT_CHARACTER_AUX6	8-bit register holds the ASCII value for the last digit text character for auxiliary 6 (0= no character).	U_8	8406	RAM/EEPROM
TEXT_CHARACTER_AUX7	8-bit register holds the ASCII value for the last digit text character for auxiliary 7 (0= no character).	U_8	8407	RAM/EEPROM
TEXT_CHARACTER_AUX8	8-bit register holds the ASCII value for the last digit text character for auxiliary 8 (0= no character).	U_8	8408	RAM/EEPROM
TEXT_CHARACTER_AUX9	8-bit register holds the ASCII value for the last digit text character for auxiliary 9 (0= no character).	U_8	8409	RAM/EEPROM
TEXT_CHARACTER_AUX10	8-bit register holds the ASCII value for the last digit text character for auxiliary 10 (0= no character).	U_8	8410	RAM/EEPROM
TEXT_CHARACTER_AUX11	8-bit register holds the ASCII value for the last digit text character for auxiliary 11 (0= no character).	U_8	8411	RAM/EEPROM
TEXT_CHARACTER_AUX12	8-bit register holds the ASCII value for the last digit text character for auxiliary 12 (0= no character).	U_8	8412	RAM/EEPROM

TEXT_CHARACTER_AUX13	8-bit register holds the ASCII value for the last digit text character for auxiliary 13 (0= no character).	U_8	8413	RAM/EEPROM
TEXT_CHARACTER_AUX14	8-bit register holds the ASCII value for the last digit text character for auxiliary 14 (0= no character).	U_8	8414	RAM/EEPROM
TEXT_CHARACTER_AUX15	8-bit register holds the ASCII value for the last digit text character for auxiliary 15 (0= no character).	U_8	8415	RAM/EEPROM
TEXT_CHARACTER_AUX16	8-bit register holds the ASCII value for the last digit text character for auxiliary 16 (0= no character).	U_8	8416	RAM/EEPROM

Each auxiliary register has its own text string register which lets users add a meaningful identification name to the register. Each string can be up to 8 characters long and should be terminated with an ASCII null (0x00) if less than 8 characters long.

The table below shows the auxiliary text registers available.

Name	Description	Symbol Type	Register Number	Memory Type
AUX1_TEXT	Text display for Auxiliary 1.	L_14	16463	EEPROM
AUX2_TEXT	Text display for Auxiliary 2.	L_14	16465	EEPROM
AUX3_TEXT	Text display for Auxiliary 3.	L_14	16467	EEPROM
AUX4_TEXT	Text display for Auxiliary 4.	L_14	16469	EEPROM
AUX5_TEXT	Text display for Auxiliary 5.	L_14	16471	EEPROM
AUX6_TEXT	Text display for Auxiliary 6.	L_14	16473	EEPROM
AUX7_TEXT	Text display for Auxiliary 7.	L_14	16475	EEPROM
AUX8_TEXT	Text display for Auxiliary 8.	L_14	16477	EEPROM
AUX9_TEXT	Text display for Auxiliary 9.	L_14	16479	EEPROM
AUX10_TEXT	Text display for Auxiliary 10.	L_14	16481	EEPROM
AUX11_TEXT	Text display for Auxiliary 11.	L_14	16483	EEPROM
AUX12_TEXT	Text display for Auxiliary 12.	L_14	16485	EEPROM
AUX13_TEXT	Text display for Auxiliary 13.	L_14	16487	EEPROM
AUX14_TEXT	Text display for Auxiliary 14.	L_14	16489	EEPROM
AUX15_TEXT	Text display for Auxiliary 15.	L_14	16491	EEPROM
AUX16_TEXT	Text display for Auxiliary 16.	L_14	16493	EEPROM

2.16.2 Memory

User memories are provided for non volatile storage of user data or look up tables, etc. These registers can be accessed either by the macro or via the serial port. They are not used by the operating system or any other functions of the controller so they can be used freely for any purpose required.

User memories are stored in RAM to give fast access and are backed up to non volatile FLASH memory at power down so that data is retained even after the power to the controller has been disconnected. There are no restrictions on the number of writes to user memory with the Zen IoT.

User memories can be addressed as either 8 bit unsigned registers or as 16 bit signed registers, however they share the same physical memory area and overlap each other. When using both types in the same application care should be taken avoid using the memory area for different variables.

User Memory Display Format/Text Character
By default, all user memory is displayed without any decimal point or additional text character. However, the user memory area can be divided up into 3 different bands, each with different decimal

point/text character configurations.

This is achieved by programming the user band registers USER_MEMORY16_BANDx (or USER_MEMORY8_BANDx for 8 bit user memories) with a register number which effectively defines the end of a group of registers which share the same display format/text settings. The display format settings for each band (or group) are specified in DISPLAY_FORMAT_USER16_BANDx (or DISPLAY_FORMAT_USER8_BANDx) while the text character selection for each band are is specified in TEXT_CHARACTER_USER16_BANDx (or TEXT_CHARACTER_USER8_BANDx).

Band 1 starts from the beginning of user memory (register 5121 for 16 bit user memories and 10241 for 8 bit user memories) and finishes after the register specified for band 1.

Band 2 starts with the next register after that specified for band 1 and finishes after the register specified for band 2.

Band 3 starts with the next register after that specified for band 2 and finishes after the register specified for band 3.

Any user memory after band 3 will be displayed without any decimal point or text character.

When using these features you should always begin by using band 1 first. The factory default setting for these bands is zero which effectively means that the whole of user memory is displayed without any decimal point or text character.

16-bit User Memory: Registers 5121 to 6144

Name	Description	Symbol Type	Register Number	Memory Type
USER_MEMORY1	signed 16-bit non-volatile memory for user defined data/tables (range -32768 to 32767).	S_16	5121	RAM/FLASH
User memory 2 through to 1023	User memories 2 to 1023 are signed 16-bit non-volatile memory for user defined data/tables (range -32768 to 32767). The register numbers begin at 5122 for user memory 2 and end at 6143 for user memory 1023, increasing by 1 register number each time.	S_16	5122 to 6143	RAM/FLASH
USER_MEMORY1024	signed 16-bit non-volatile memory for user defined data/tables (range -32768 to 32767).	S_16	6144	RAM/FLASH
DISPLAY_FORMAT_USER16_BAND1	8-bit register. Controls the display format settings for 16 bit user memory band 1 (displayed in octal format).	O_8	8363	RAM/EEPROM
DISPLAY_FORMAT_USER16_BAND2	8-bit register. Controls the display format settings for 16 bit user memory band 1 (displayed in octal format).	O_8	8364	RAM/EEPROM
DISPLAY_FORMAT_USER16_BAND3	8-bit register. Controls the display format settings for 16 bit user memory band 1 (displayed in octal format).	O_8	8365	RAM/EEPROM
TEXT_CHARACTER_USER16_BAND1	8-bit register. Holds the ASCII value for the last digit text character for 16 bit user memory band 1 (0 = no character).	U_8	8417	RAM/EEPROM

TEXT_CHARACTER_USER16_BAND2	8-bit register. Holds the ASCII value for the last digit text character for 16 bit user memory band 2 (0 = no character).	U_8	8418	RAM/EEPROM
TEXT_CHARACTER_USER16_BAND3	8-bit register. Holds the ASCII value for the last digit text character for 16 bit user memory band 3 (0 = no character).	U_8	8419	RAM/EEPROM
USER_MEMORY16_BAND1	Unsigned 16-bit register that defines the last (highest) register number for band 1 of 16bit user memory.	U_16	4464	RAM/EEPROM
USER_MEMORY16_BAND2	Unsigned 16-bit register that defines the last (highest) register number for band 2 of 16bit user memory.	U_16	4465	RAM/EEPROM
USER_MEMORY16_BAND3	Unsigned 16-bit register that defines the last (highest) register number for band 3 of 16bit user memory.	U_16	4466	RAM/EEPROM

CAUTION **16-bit user memories overlap 8-bit user memories.**

8-bit User Memory Bytes: Registers 10241 to 12288

Name	Description	Symbol Type	Register Number	Memory Type
USER_MEMORY_BYTE_1	unsigned 8-bit non-volatile memory for macro use (range 0 to 255).	U_8	10241	RAM/FLASH
User memory bytes 2 through to 2047	User memory bytes 2 to 2047 are unsigned 8-bit non-volatile memory for macro use. The register numbers begin at 10241 for user memory byte 1 and end at 12288 for user memory byte 2048, increasing by 1 register number for each byte.	U_8	10242 to 12287	RAM/FLASH
USER_MEMORY_BYTE_2048	unsigned 8-bit non-volatile memory for macro use (range 0 to 255).	U_8	12288	RAM/FLASH
DISPLAY_FORMAT_USER8_BAND1	8-bit register. Controls the display format settings for 8 bit user memory band 1 (displayed in octal format).	O_8	8366	RAM/EEPROM
DISPLAY_FORMAT_USER8_BAND2	8-bit register. Controls the display format settings for 8 bit user memory band 2 (displayed in octal format).	O_8	8367	RAM/EEPROM
DISPLAY_FORMAT_USER8_BAND3	8-bit register. Controls the display format settings for 8 bit user memory band 3 (displayed in octal format).	O_8	8368	RAM/EEPROM
TEXT_CHARACTER_USER8_BAND 1	8-bit register. Holds the ASCII value for the last digit text character for 8 bit user memory band 1 (0 = no character).	U_8	8420	RAM/EEPROM

TEXT_CHARACTER_USER8_BAND 2	8-bit register. Holds the ASCII value for the last digit text character for 8 bit user memory band 2 (0 = no character).	U_8	8421	RAM/EEPROM
TEXT_CHARACTER_USER8_BAND 3	8-bit register. Holds the ASCII value for the last digit text character for 8 bit user memory band 3 (0 = no character).	U_8	8422	RAM/EEPROM
USER_MEMORY8_BAND1	Unsigned 16-bit register that defines the last (highest) register number for band 1 of 8 bit user memory.	U_16	4467	RAM/EEPROM
USER_MEMORY8_BAND2	Unsigned 16-bit register that defines the last (highest) register number for band 2 of 8 bit user memory.	U_16	4468	RAM/EEPROM
USER_MEMORY8_BAND3	Unsigned 16-bit register that defines the last (highest) register number for band 3 of 8 bit user memory.	U_16	4469	RAM/EEPROM

CAUTION 8-bit user memory bytes overlap 16-bit user memories.

2.16.2.1 16-bit User Memory

Registers 5121 to 6144 are 16-bit signed registers that can be used for non volatile storage of user data or look up tables, etc. These registers can be accessed either by the macro or via the serial port. They are not used by the operating system or any other functions of the controller so they can be used freely for any purpose required.

User memories are stored in RAM to give fast access and are backed up to non volatile FLASH memory at power down so that data is retained even after the power to the controller has been disconnected. There are no restrictions on the number of writes to user memory with the Zen IoT.

CAUTION: These registers overlap the 8-bit User Memories and share the same physical memory space. The only difference is that registers 5121 to 6144 address 1024 user memories, each 16 bits wide. For example;

Register 10241 occupies the same memory space as the most significant byte of register 5121
Register 10242 occupies the same memory space as the least significant byte of register 5121

Care should be taken when using both types of user memories to ensure that different memory areas are used.

For information on how to set the display format and text character of user memories see;
User Memory Display Format/Text Characters

2.16.2.2 8-bit User Memories

Registers 10241 to 12288 are 8-bit unsigned registers that can be used for non volatile storage of user data or look up tables, etc. These registers can be accessed either by the macro or via the serial port. They are not used by the operating system or any other functions of the controller so they can be used freely for any purpose required.

User memories are stored in RAM to give fast access and are backed up to non volatile FLASH memory at power down so that data is retained even after the power to the controller has been disconnected. There are no restrictions on the number of writes to user memory with the Zen IoT.

CAUTION: These registers overlap the 16-bit User Memory and share the same physical memory space. The only difference is that registers 10241 to 12288 address 2048 user memories, each 8 bits wide. For example;

Register 10241 occupies the same memory space as the most significant byte of register 5121

Register 10242 occupies the same memory space as the least significant byte of register 5121

Care should be taken when using both types of user memories to ensure that different memory areas are used.

For information on how to set the display format and text character of user memories see;
User Memory Display Format/Text Characters

2.16.3 Text Memory

Registers 16567 to 16693 are used to store user defined text strings of up to 30 characters long. Text strings are stored in EEPROM non volatile memory and are retained at power down. They can be used in a macro to store text which may need to changed by the end user. A good example of this would be to store a company name or phone number or maybe an email address.

Text strings which are shorter that 30 characters should be terminated with an ASCII null (i.e. should have an extra character of 0 added to the end of the string).

Name	Description	Symbol Type	Register Number	Memory Type
USER_TEXT1	Non-volatile 30 character text string for user defined text storage.	L_30	16567	EEPROM
User text 1 through to 64	User text strings 1 to 64 are non-volatile 30 character text strings for user defined text storage. The register numbers begin at 16567 for text string 1 and end at 16693 for text string 64, increasing by 2 register numbers for each text string.	L_30	16567 to 16693	EEPROM
USER_TEXT64	Non volatile 30 character text string for user defined text storage.	L_30	16693	EEPROM
USER_LONG_TEXT1	Non-volatile 80 character text string for user defined text storage.	L_80	16831	EEPROM
User text 1 through to 25	User text strings 1 to 25 are non-volatile 80 character text strings for user defined text storage. The register numbers begin at 16831 for long text string 1 and end at 16879 for long text string 25, increasing by 2 register numbers for each text string.	L_80	16831 to 16879	EEPROM
USER_LONG_TEXT25	Non-volatile 80 character text string for user defined text storage.	L_80	16879	EEPROM

Password Storage
Registers 16881 to 16895 give 8 80 character non-volatile registers for password storage. These registers can be written to in the same way as other text registers but they cannot be read from an external source. Any attempt to read these registers via a serial port will result in each character being replaced with "*" character instead of the original character. These registers can however be accessed from macro commands so it is up to the users discretion to ensure that passwords are not send to the display or serial port via the macro.

Note: All of these 8 password registers are erased whenever the macro is erased or over written.

PASSWORD1	Non-volatile 80 character text string for user defined password storage.	L_80	16881	EEPROM
User text 1 through to 25	User text strings 1 to 25 are non-volatile 80 character text strings for user defined text storage. The register numbers begin at 16881 for long text string 1 and end at 16895 for long text string 25, increasing by 2 register numbers for each text string.	L_80	16881 to 16895	EEPROM
USER_LONG_TEXT25	Non-volatile 80 character text string for user defined password storage.	L_80	16895	EEPROM

NOTE: Because user text memories are stored in EEPROM there is a limitation of 1x10^6 writes allowed to these registers (see Memory Types for more information on maximum write limitation). If you need to write continuously to a text register then you should use Text Variables instead)

See also
ASCII Text Registers

Text Variables

Startup Text

2.16.3.1 Station Name

Register 16823 is a special 30 character text string register which contains the station name. This is a user defined string which can be used to identify a particular device on a network. It is normally written and read by an external device via the serial port. It is stored in non-volatile EEPROM memory which has certain write restrictions (see EEPROM write restrictions)

Name	Description	Symbol Type	Register Number	Memory Type
STATION_NAME	Non-volatile 30 character text string which can be used to identify a particular device on a network.	L_30	16823	EEPROM

See Also
Macro Name

Startup Text

Intech Scratchpad Text

2.16.3.2 Macro Name

Register 16825 is a special 30 character text string register which contains the macro name. This is a user defined string which can be used to identify a which macro is installed in the controller and may include a macro version number as well. It is normally written and read by an external device via the serial port. It is stored in non-volatile EEPROM memory which has certain write restrictions (see EEPROM write restrictions)

Name	Description	Symbol Type	Register Number	Memory Type
MACRO_NAME	Non-volatile 30 character text string which can be used to identify which macro is currently installed in the controller.	L_30	16825	EEPROM

See Also

Station Name

Startup Text

Intech Scratchpad Text

2.16.4 Variables

Variable registers are provided as a means of storing temporary data in a macro application. Variable registers are not used by the operating system or any other standard controller functions. However some variable registers are assigned by the compiler. The different types of variable registers are shown below.

See Bit Flags

Floating Point

Integers

Text

See also
ASCII Text Registers

2.16.4.1 Bit Flags

Register 241 is a special purpose register which contains 32 bit flags which can be set or cleared in the macro.

These flags are stored in RAM and are lost at power down.

Name	Description	Symbol Type	Register Number	Memory Type
GPF1	General purpose bit flag for macro use. Also settable via an LCD panel using the RPC (101) command.	B_0	241	RAM
General purpose bit flags 1 through to 32	All general purpose bit flags from 1 to 32 are for macro use and are also settable via an LCD panel using the RPC (101) command. The general purpose bit flags fall under register number 241 and are identified by their bit number: GPF1 = B_0 through to GPF32 = B_31.	B_0 to B_31	241	RAM
GPF32	General purpose bit flag for macro use. Also settable via an LCD panel using the RPC (101) command.	B_31	241	RAM

2.16.4.2 Floating Point

Registers 1025 to 1055 are non volatile 32-bit floating point variables that can be used in the macro to save floating point parameters. They hold a single precision floating point number that is formatted according to the IEEE-745 standard. They are stored in RAM for fast access and then saved to non volatile FLASH memory at power down.

Registers 1025 to 1055 are not used by the operating system or any other standard functions in the controller. However, they are assigned as floating point variables by the compiler when a new variable is declared using the % symbol. For example, if the following macro code is compiled:

```
RESET_MACRO:
%TEMP1 = 1.5
%TEMP2 = 1.234e-2
%TEMP3 = 1.0
END
```

The variable TEMP1 would be assigned to register FLOAT_VARIABLE1 (1025), TEMP2 would be assigned to register FLOAT_VARIABLE2 (1027), and TEMP3 would be assigned to register

FLOAT_VARIABLE3 (1029). In this case the same data can be accessed by either register name. You need to ensure that the same memory area is not used for different variable functions.

Register 1095 is a 32-bit floating point register that can also be accessed as a 32-bit fixed point number via register 479. This can be a useful feature in some macros that decode an incoming serial string by reading data in a fixed point number but then need to treat it as a floating point number. See also Miscellaneous Registers

Name	Description	Symbol Type	Register Number	Memory Type
FLOAT_VARIABLE1	32-bit floating point register used by the macro for variable space.	F_32	1025	RAM/FLASH
Floating point variables 1 through to 16	Floating point variables 1 to 16 are 32-bit for macro use. The register numbers begin at 1025 for floating point register 1 and end at 1055 for floating point register 16, increasing by 2 register numbers each time.	F_32	1025 to 1055	RAM/FLASH
FLOAT_VARIABLE16	32-bit floating point register used by the macro for variable space.	F_32	1055	RAM/FLASH
VARIABLE_A_FP	32-bit register for variable A, accessed in floating point format.	F_32	1095	RAM

2.16.4.3 Integers

Registers 155 to 217 are 32-bit signed variables which can be used by in the macro to temporarily store parameters. They are stored in RAM for fast access and their contents is lost at power down. They default to 0 when the controller is turned ON.

Name	Description	Symbol Type	Register Number	Memory Type
INTEGER_VARIABLE1	32-bit integer used for macro variable space.	S_32	155	RAM
Integer variables 1 through to 32	Integer variables 1 to 32 are 32-bit for macro variable space. The register numbers begin at 155 for integer variable 1 and end at 217 for integer variable 32, increasing by 2 register numbers each time.	S_32	155 to 217	RAM
INTEGER_VARIABLE32	32-bit integer used for macro variable space.	S_32	217	RAM
VARIABLE_A_INT	32-bit register for variable A, accessed in fixed point format.	S_32	479	RAM

Registers 155 to 217 are not used by the operating system or any other standard functions in the controller. However, they are assigned as variables by the DDS when a new variable is declared using the # symbol. For example, if the following macro code is compiled:

```
RESET_MACRO:
#TEMP1 = 15
#TEMP2 = 12345
#TEMP3 = 1
END
```

The variable TEMP1 would be assigned to register INTEGER_VARIABLE1 (155), TEMP2 would be assigned to register INTEGER_VARIABLE2 (157), and TEMP3 would be assigned to register INTEGER_VARIABLE3 (159). In this case the same data can be accessed by either register name.

Bit Variables
When using the DDS, bit variables can also be declared in a macro in a similar method as above. The following macro code shows an example of how to declare bit variables in your macro source code.

```
RESET_MACRO:
 |MY_FLAG1 = FALSE
 |MY_FLAG2 = FALSE
```

|MY_FLAG3 = FALSE
END

When bit variables are declared as shown above, the DDS will take the highest unused integer variable (usually INTEGER_VARIABLE32) and allocate this as a temporary bit flag register. It will then assign |MY_FLAG1 to be bit 0 of that register, MY_FLAG2 to be bit 1, MY_FLAG3 to be bit 2 and so on. If more than 32 bit variables are declared the DDS will start using the next integer variable down and assign bit flags there.

Note:
You need to ensure that the same memory area is not used for different variable functions. If you are allowing the compiler to allocate variables by declaring them with the '#' symbol or bit flags with the '|' symbol and also referencing the same variable by it's predefined variable name (INTEGER_VARIABLEx) then the compiler will issue a warning to make you aware that you could be overwriting the same variable space.

Register 479 is a 32-bit fixed point register that can also be accessed as a 32-bit floating point number via register 1095. This can be a useful feature in some macros that decode an incoming serial string by reading data in a fixed point number but then need to treat it as a floating point number. See also Miscellaneous Registers

2.16.4.4 Text Variables

Registers 16897 to 16927 are used to store user defined text strings of up to 30 characters long. Text strings are stored in RAM for fast access and are lost at power down. They can be used in a macro to store temporary text which is received via the serial port.

Name	Description	Symbol Type	Register Number	Memory Type
TEXT_VARIABLE1	30 character text string variable in RAM.	L_30	16897	RAM
Text variables 1 through to 16	Text variables 1 to 16 are 30 character text string variables in RAM. The register numbers begin at 16897 for text variable 1 and end at 16927 for text variable 16, increasing by 2 register numbers each time.	L_30	16897 to 16927	RAM
TEXT_VARIABLE16	30 character text string variable in RAM.	L_30	16927	RAM

NOTE: Text strings which are shorter than 30 characters long should be terminated with an ASCII null character (0x00).

See also
Text Memory

2.17 Miscellaneous Registers

Variable A Register – 479 and 1095
Registers 479 and 1095 both address the same physical 32-bit register in memory (Variable A), but differ in the way the register is interpreted. Accessing Variable A through register 1095 assumes the contents have been stored in a 32-bit single precision floating point format. Accessing Variable A through register 479 assumes the contents have been stored in a 32-bit fixed point long format.

This pair of registers is only intended for use with the macro.

Power-up Reset Counter – Register 4307
Register 4307 is a 16-bit read/write register that is incremented each time the controller is powered up or reset from the rear test pin. It is used for diagnostic purposes only.

Memory Size for External Data Logger – Register 8432
Register 8432 is an 8-bit unsigned register that shows if an external data logger module is connected to the controller and how much memory is installed. (See External Data Logger for more information

on how to interpret this register).

CPU Loading – Register 8434
This register is an 8-bit read only unsigned register that shows the current processing load on the CPU in the controller from 1 to 100%. A value over 90% indicates that the controller is running out of processing time to complete all of the required functions within the selected update time. As a result input samples or output functions may be skipped and software timers or totalizers may become inaccurate. The solution is to set to OFF all unused functions and setpoints, reduce the size of any macros that are currently running, and select an update rate of 0.1 seconds.

OEM Control Register - Register 8513
On the Zen IoT series of controllers the macro space can be split to allow an open area of macro space for the end user application and a closed (or locked) area of macro space for an OEM macro. Register 8513 is an 8 bit unsigned register which controls the size and lock status of the OEM macro area. Bits 0 to 4 select the size of the OEM macro in 1k byte blocks so that a size of 0 to 31k bytes can be selected. Bits 0 - 4 can only be written when their original value is 0. If any value other than 0 has been written to these bytes they cannot be overwritten until the OEM macro is erased. The OEM macro can be erased from the macro development system.

Name	Description	Symbol Type	Register Number	Memory Type
OEM_CONTROL	8 bit register that shows the size and lock status of the OEM macro area.	U_8	8513	FLASH

Bit 7 locks all read/write access to the OEM macro area and also locks write access to the OEM control byte. Once bit 7 is set the OEM control byte cannot be modified until the OEM macro is erased.

NOTE:
1) The actual size of an OEM macro is limited by the amount of macro code space available in the controller. So the maximum OEM macro size of 31k in the OEM control byte is a theoretical maximum only and allows for future development. (See note below on Macro Size)

Macro Size – Register 4433
This register is a 16-bit read only register that defines the amount of macro code space available in the controller. Reading this register produces a number from 1 to 65535 that relates to the number of ASCII/Modbus registers allocated for macro code storage. This may change with model or version number.

Software Version Number - Register 4106
This is a 16 bit read only register that defines the currently installed software version number.

Device Type - Register 16565
This is a read only text register that defines the model or controller type.

Product Serial Number - Register 541
This is a 32 bit read only register that holds the product serial number.

Cal Date - Register 543
This is a 32 bit register that holds the calibration date for the product. The format is as follows;

MSW = Year
LSW = (MSB = Month, LSB = Date)

Memory Reset - Register 16559
This is write only text register which can be used to reset the controller or parts of the controllers memory. All text written to this register must be in upper case. The following functions can be accessed via this register;

Factory defaults - The controller configuration can be returned to factory defaults by writing "INIT" to register 16559.

Erase Macro - All programmed macros can be erased by writing "ERASE MACRO" to register 16559.

Controller Reset - The controller can be totally reset by writing "RESET" to register 16559.

WARNING: Using this register could erase any currently installed macro or cause the loss of custom configuration data by returning the meter to factory defaults. We recommend care when using this register!

Index

- 1 -

- 2 -

- 3 -

- 4 -

- 8 -

- A -

- B -

- C -

- D -

- E -

- F -

- H -

- I -

- L -

- M -

- N -

- O -

- P -

- R -

- S -

- T -

- U -

- V -