

# IO Module

User Manual



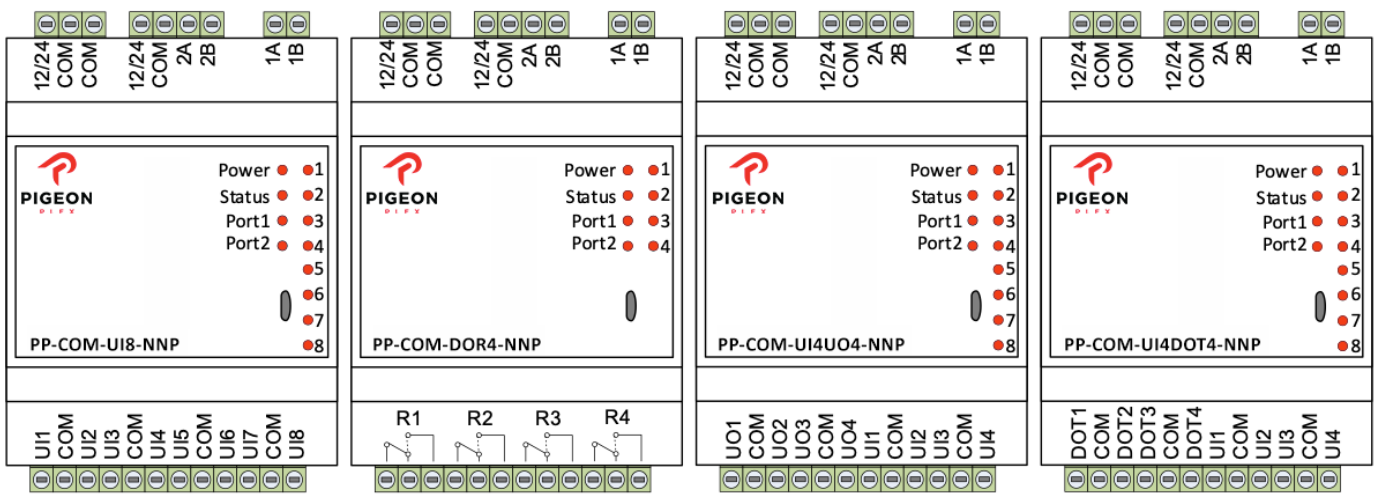
# PP-COM-IO INPUT / OUTPUT MODULES

Edge Input and Output controller for data gathering and control. Models are available with different Input/Output (IO) configurations matching controller to use case. Each controller has hardware as well as licensed feature options to effectively match use case to price.

Non-Volatile memory - FLASH and EERAM for pulse counting, runtime and user defined variables make these IO modules suitable for a wide variety of applications.

Optional protocols such as BACnet on Port1, Modbus Client on Port2 and user programmable control blocks are available. Refer to the [PP-COM-IO-Roadmap.pdf](#) for more information.

Hardware options and license features are different part numbers – ensure the correct hardware is matched to the license features required. As an example, if a protocol is selected for Port2 then ensure the hardware selected has 2 Ports. An [PP-COM-IO-SelectionTool.xlsx](#) is available to simplify this process.



## Revision Control

Revision	Description	User	Date
1.0	Release for review. Note that Rev1-1-x must be used for any devices on rev 1-1-x. Changes to register table structure, IO handling, control blocks plus the addition of BACnet protocol necessitated a minor revision change. Always check the firmware revision available on the console and use the correct revision user manual.	I Potter	24 October 2022
1.1	Always check the firmware revision available on the console and use the correct revision user manual. Unit tests complete.	I Potter	6 February 2024

Revision numbering has 3 digits. Major Revision – Minor Revision – Bug Fix

- Major Revision – hardware change with associated firmware improvements
- Minor Revision – firmware improvements
- Bug Fix – Correction of identified bugs

## Index

<b>Overview</b> .....	4
<b>Wiring</b> .....	5
Power and RS485 .....	5
.....	5
<b>Terminals and IO</b> .....	5
<b>Licensing</b> .....	5
<b>Memory for User Applications</b> .....	6
<b>Register Table</b> .....	7
Register Table - Logical Areas .....	7
Register table. VOLATILE, FLASH and EERAM .....	7
<b>Data Type</b> .....	9
<b>Filing System</b> .....	9
<b>EERAM</b> .....	9
EERAM Rules .....	9
<b>FLASH</b> .....	9
<b>USB Logs – Log Verbosity. Register 551</b> .....	10
IO Definition .....	10
AI Threshold and Response.....	11
DI Threshold and Debounce .....	11
AI Calibration (Associated Value) .....	11
AO Calibration (Associated Value).....	11
<b>IO CONFIGURATION</b> .....	11
DI.....	11
DIVIDER/PULSE.....	11
DO/DOT/DOR .....	12
UO/DOT/DOR PWM or PULSE Period .....	12
<b>IO Values</b> .....	13
<b>SPECIAL SYSTEM Register 698 and 699</b> .....	13
Reboot.....	13
Factory Reset.....	13
Configuration Reset .....	13
Write Configuration to File. Register 407 .....	13
<b>Boot Delay. Register 465</b> .....	14
<b>Factory Test. Register 404</b> .....	14
<b>Port1 (405) and Port2 (406) Communication quality</b> .....	14
<b>Health</b> .....	14
<b>1 Second Pulse Train</b> .....	14
<b>Register Initialization and Reboot Process</b> .....	15
<b>LED Indication</b> .....	16
Boot.....	16
Reboot.....	16
Factory Reset.....	16
Configuration Reset .....	16
Normal Operation.....	16

Input and Output LED's.....	17
<b>Personality Text Files.....</b>	<b>18</b>
JSON Strings .....	19
<file> .....	19
<lic>.....	19
<cb> .....	20
<reg>.....	20
<riv> .....	20
<dev> .....	21
<ai> <ao> <av> .....	21
<bi> <bo> <bv> .....	21
<msv> .....	22
<b>Console .....</b>	<b>22</b>
Console Commands.....	22
<b>Ports - UART – Communications Parameters .....</b>	<b>23</b>
<b>Ports and Protocols .....</b>	<b>23</b>
Modbus Server – Port1 .....	24
Modbus Client – Port2.....	24
Communication Blocks – Modbus client – Port 2.....	24
Modbus Scan cycle and Trigger.....	24
Client Queue Management .....	25
Modbus Register Sequence.....	27
Endianness and Byte Swap .....	28
Modbus Data Formats.....	28
BACnet – Port1 .....	28
BACnet Capabilities Statement.....	28
Special Functions and Interfaces.....	32
BACnet Manual Override .....	33
<b>IO Manual Override.....</b>	<b>33</b>
AI/DI Manual Processing.....	33
AO/DO Manual Processing .....	33
<b>Control Blocks.....</b>	<b>34</b>
<b>Register Table.....</b>	<b>34</b>
USER Volatile - Vuser .....	34
User Flash - Fuser .....	34
User EERAM - Euser .....	34
System Volatile - Vsys .....	35
System Flash - Fsys.....	84
System EERAM – Esys.....	85
Special .....	89
<b>Tutorials .....</b>	<b>90</b>

## Overview

---

PP-COM-IO hardware and all communication channels are interfaced through a common register table. This approach allows the controller flexibility in communications and execution by changing registers – a simple method with immense power and flexibility.

The Inputs transduce real world values and scale them to the defined requirements then place the values in registers. The outputs take register value's and transduce them to match the application – whether this be 0-10v or a relay – in each case, the register determines the output value. The power of control blocks is deployed between the input and output registers to perform the control logic required.

Data types define how the information is represented – an example for the technical experts. A data point can be uint16 or float. The power of this is evident once the control blocks are used. To simplify data type casting, all registers are always available as all data types. As mentioned – this is for the technical boffins – don't worry if this is not understood, the controller handles the tasks without programmer intervention.

Controller configuration determines how the inputs and outputs respond and what they are connected to. This is kept in text files and sent to the controller using the [edgeUP-App](#). Make sure to install the correct version (in this case 1-1-x).

The same app is used for downloading control blocks.

The following resources are available:

1. [PP-COM-IO-Roadmap.pdf](#)
2. [PP-COM-IO-SelectionTool.xlsx](#) Remember to save a copy of this file to your device to enable the drop-down lists to operate.
3. [PP-COM-IO-Datasheet.pdf](#)
4. [PP-COM-IO-UserManual.pdf](#)
5. [PP-COM-IO-UserManual-ControlBlocks.pdf](#) this manual
6. [edgeUP-App](#)

## Wiring

The modules have the same form factor and terminal arrangement which allows for standardized wiring. The following IO types are supported.

- UI User selectable for 0-10V, 10k resistor, 10k thermistor and digital input.
- SI User selectable 10k resistor, 10k thermistor and digital input.
- DI Digital input only.
- UO User selectable 0-10V dc, 0-10V PWM or 12V DC on/off.
- DOC Digital Output Open Collector.
- DOT Digital Output Triac
- DOR Digital Output Relay

Power supply is 12 to 26v AC or DC. 3 Wire - if the power supply is AC, the AC neutral is connected to the COM rail. Used mainly in HVAC applications.

## Power and RS485

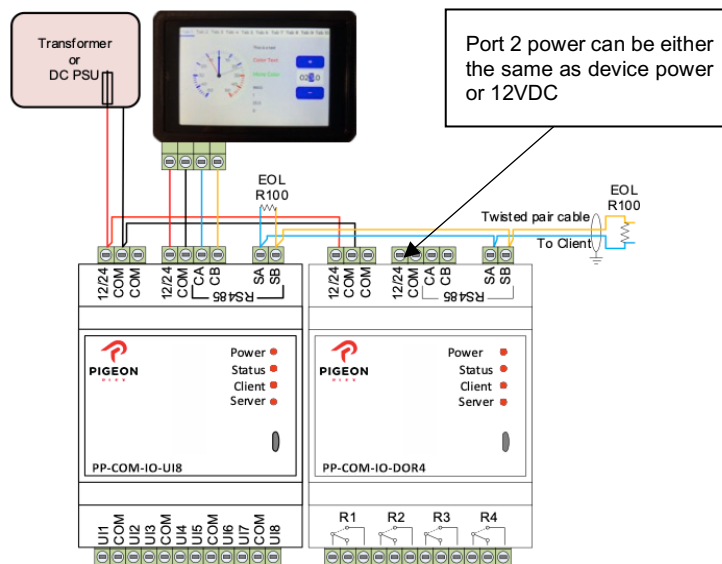


Figure 1

## Terminals and IO

No special requirements for wiring. Follow standard practice for screening of inputs. Note the I/O are not isolated therefore ground rail is susceptible to induced voltages if field wiring is not installed as per standard practice. Special care to be given to preventing induced voltages by using screen cable, terminating screen at one end only and not running cables next to higher voltages. RS485 is not optically isolated on the N models. Terminals are numbered from left to right and are allocated as per the legend which is model specific. For UI8 terminals follow the UI number. For UI4UO4 or UI4DOT4 the terminals are 1 to 4 for UO/DOT 1 to 4 the 5 to 8 for UI 1 to 4. Terminal numbers are used to identify the input when viewing console logs and at BACnet point creation (if BACnet is licensed).

## Licensing

Licensing is used to enable additional features. An unlicensed device will operate in IO mode with Modbus Server on Port1. No support for Port2 or control blocks.

The license is contained in line 2 of file system.txt and is ordered as a separate part number when ordering the IO module or as an upgrade. Orders can be sent to PigeonPlex (email [info@digitaltwin.digital](mailto:info@digitaltwin.digital)). The device MAC and features required are to be emailed. Review the [PP-COM-IO-Roadmap.pdf](#) document for available options.

On receipt of the license code, paste this into system.txt file on a new line anywhere after the file name line and save the file to the device. Refer to the lessons on licensing for more detail.

The License flag (refer to system volatile register table) indicates the state of registration.

Bit interpretation of the License flags.

Value	Descriptor	Define
0 (1)	Registration File Found	h_LIC_f
1 (2)	Registration file name check passed	h_LIC_n
2 (4)	License read OK	h_LIC_r
3 (8)	License code read from file	h_LIC_c
4 (16)	License code decoded ready for check	h_LIC_i
5 (32)	License code passed serial number match	h_LIC_s

License features are embedded in the license and determine the personality of the device.

## Memory for User Applications

Approx 20k bytes of memory are available for user defined applications. This includes control blocks and communication objects. The following table defines the number of bytes used per object. Use this to determine the available space for each application. Refer to the Free Memory register in System Volatile space for the runtime memory available.

Function Communication	Bytes	Function Control	Bytes
Modbus Client	20	Logic	16
BACnet ai	40	Invert	8
BACnet ao	90	AnalogSelect	16
BACnet av	30	Delay	15
BACnet bi	60	Runtime	15
BACnet bo	60	Totalizer	10
BACnet bv	40	MinOnOff	20
BACnet msv	100	AV-MSV	22
		DataType	8
		Transfer	8
		OnOff	12
		PID	30
		3Point	18
		FAN	28
		Latch	40
		Math	22
		Limit	10
		Linear	36

## Register Table

A common register table is used to provide a flexible modular product that is easily adapted to each use case. The diagram below shows the register table positioned as the central repository for data – all operations read from this table and write the result to the table. The table definition per module type details each register and is divided into user and system areas. These are further divided into volatile, flash and eeram.

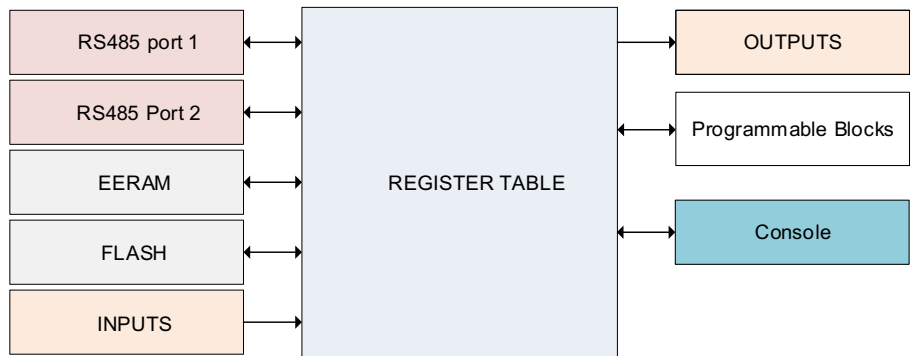


Figure 2

### Register Table - Logical Areas

The register table is divided into *user* and *system* logical areas. Each area is further divided by memory type into volatile and nonvolatile with non-volatile divided into flash and eeram.

Each register is 16 bits and can be assigned a data type to identify how it is represented. Each register therefore has 2 tables, the *register table* and the *register type table*. **Data Type**

Register Section	Note	Address from	Address to
Volatile User	Use for variables display variables such as IO	0	199
Flash User	Use for variables display variables such as setpoints. These values are read from Flash on restart and NOT written to flash on change. This allows for default values. If the value is to be saved on write then use EERAM.	200	249
EERAM User	Use for changing variables such as counters and runtime. These values are not saved in flash so have no initialization. Can be initialized via file and will retain last value between power cycles.	250	349
Volatile system	Predefined use. Refer to User Selected System Values.	350	449
Flash System	Predefined use. Refer to User Selected System Values. Flash	450	459
EERAM System	Predefined use. Refer to User Selected System Values. EERAM	550	697
Special Registers	Reboot – Write 0x55AA. Not written to NV	698	
	Factory Reset – Write 0x1928. Not written to NV	699	

### Register table. VOLATILE, FLASH and EERAM

Refer to **Figure 3** and **Register Table**

Volatile memory does not retain its value when power cycled or rebooted. To maintain values through power cycles non-volatile memory is used. There are 2 different memory types – each performs a specific function. Non-volatile FLASH is located on a flash chip using SPIFFS filing system and is used for low frequency writes such as fall-back values and configuration settings. An EERAM chip is used to store high frequency writes such as runtime or pulse counters – this type of ram only writes to the non-volatile section of the chip on loss of power – an internal capacitor keeps the device alive on power failure for long enough to write the data.

All registers are in volatile ram –this implies that if the device power cycles or reboots the values will revert to 0. To retain the values, the register must be copied from or copied to the non-volatile memory on change or power cycle. There are 3 possible channels that change a registers value in the register table.

1. Internal process – such as boot health – this is part of the device operating system.
2. Console command such as riv.
3. Communication channel – either channel can write directly. Example Modbus write single register.
4. Control Block result. Registers hold the result of control blocks.

FLASH and EERAM are handled as follows:

1. FLASH
  - a. Registers changed in the flash section are not automatically written to FLASH (SPIFFS)
  - b. This allows values to be changed and then returned to a fall-back value on power cycle or reboot.
2. EERAM
  - a. System register changes are written to EERAM chip every second only on value change. Runtime and Pulse counters are examples.
  - b. Console or comms channel changes are flagged and written every second on change.

SPIFFS is a filing system located on the flash chip. Text files with JSON strings record the device and register configurations as well as control block definitions. The files related to flash and eeram are shown in the diagram below. The files can either be loaded by the user from an external file, created by the system at boot if missing or written on user command. Refer to **Write Configuration to File. Register 407** for how to create the files from the existing memory values or use edgeUP-App to write and download files.

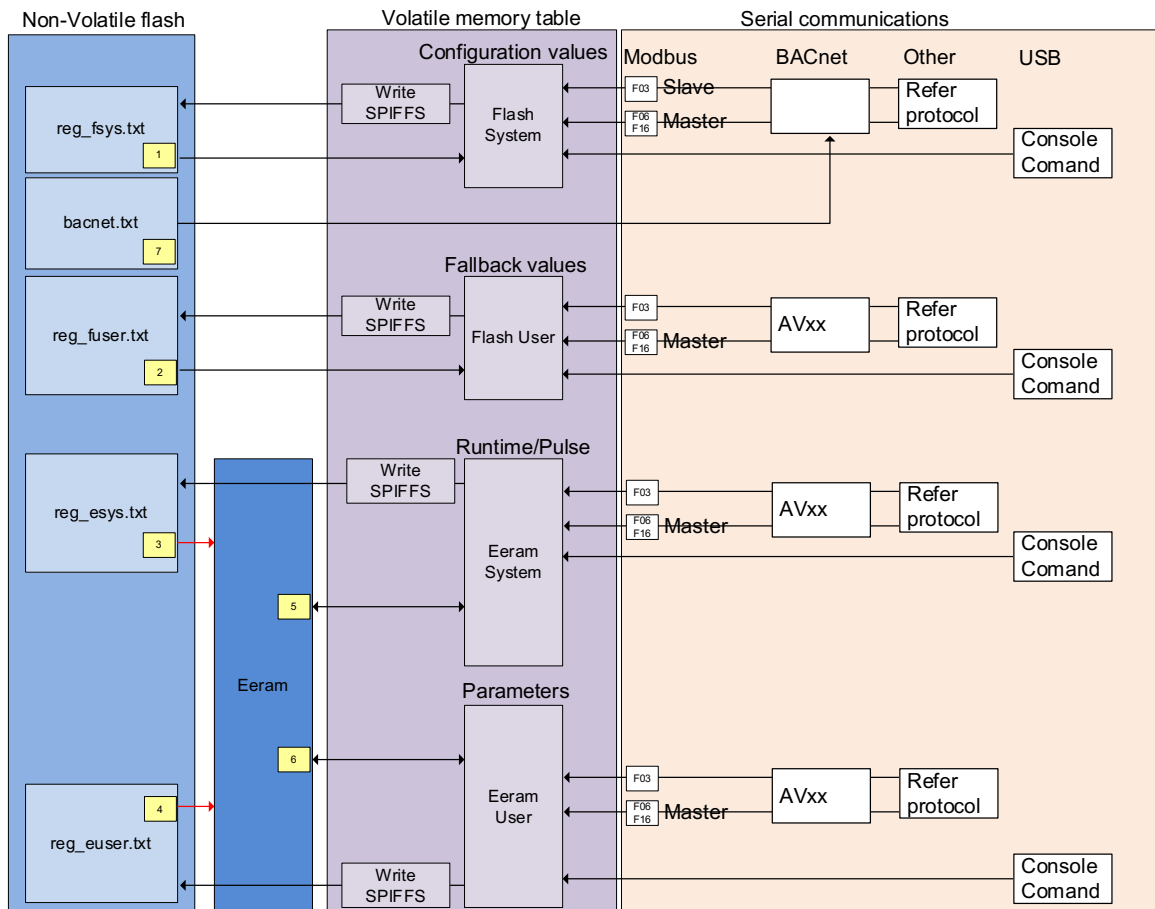


Figure 3

1. Loaded at reboot. If file does not exist, firmware defaults are loaded for both value and type. These values are device dependent. Refer to device register table default values.
2. Loaded at reboot. If file does not exist, values are set to 0 and no register types set (0).
3. Loaded when special register eeram system is set.
4. Loaded when special register eeram user is set.
5. If an IO is configured as runtime or pulse count, these values are written every second.
6. Eeram updated on each non-volatile register change in eeram space within 1 second.
7. Loaded at reboot. If file does not exist, BACnet objects use default values. Only valid for BACnet protocol

## Data Type

---

The register table is made up of 16-bit registers. The following data types are supported by using 16 bit registers as a base.

Data Type	Data Type Enum	Number of Registers	Note
uint8	1	1	0 to 255
uint16	2	1	0 to 65 535
uint32	3	2	0 to 4 294 967 295
int8	4	1	-127 to +127
int16	5	1	-32 767 to +32 767
int32	6	2	-2 147 483 647 to + 2 147 483 647
float	7	2	-3.4E+38 to +3.4E+38. Little Endian byte swap
char	8	Depends on char application	String. Char length is dependent on application

Register allocation and data types are set up as default – no user intervention is required. For advanced users, remember to allocate the correct number of registers when creating the Index. If a data type uses 2 registers, then allocate register x for the data type and register x+2 for the next.

When writing to registers where the value exceeds the max value of a data type, the results will be as follows.

Example write value 300 to an uint8. The result will be 44. This is explained by looking at the bytes.

300 decimal = 0x12C. If we look at the size of an uint8 it is 1 byte. take the 0x2C and convert to decimal. The result is 44. This means that type casting shows the bytes available in the registers at the correct byte position and does not do range checking.

## Filing System

---

A SPIFFS filing system to store non-volatile data – SPIFFS is an acronym for Serial Peripheral Interface Flash File System. The filing system contains ASCII text files to save the various user defined options. SPIFFS is not a memory type – it is a file system.

The files are used for configuration, default values, personalities, and control loops. Each file has a defined format and syntax and is described in the relevant sections. The files are loaded using edgeUP-App or a 3<sup>rd</sup> party serial port utility – edgeUP-App is recommended for reliable trouble-free transfer of text files.

## EERAM

---

Memory that retains data values after power cycle or power fail. Eeram user section holds user non-volatile values where a pre-determined fallback value is not required. Eeram system holds runtime and pulse counts. The data types are pre-defined as detailed in the register table (refer [User EERAM](#) and [System EERAM](#)). There are register commands to write the values in EERAM to flash [Write Configuration to File. Register 407](#) and to read from flash into EERAM [SPECIAL SYSTEM REGISTER 698 AND 699](#).

Writing from EERAM to flash allows for devices to be configured with default values and the defaults saved to flash. The file contents can then be copied into the project folder and saved as a default for configuring other devices. This improves commissioning time where multiple controllers have the same configuration.

## EERAM Rules

---

User values are saved on change (either comms or console).

System values are saved on change every second. If there is no change, no registers are saved.

## FLASH

---

Memory that retains data values after power cycle or power fail. These registers are read from a text file onto SPIFFS. Flash registers are read after power cycle or reboot.

When a write is received in the user or system section of flash, it must be written to SPIFFS if the value is to become nonvolatile. This is done by using the write configuration register.

## USB Logs – Log Verbosity. Register 551

A log bitmap determines the Log Verbosity – or what modules will be logged on the USB console as defined in table below. This print out of additional log information can be used to assist in error or fault finding. If the bit is set in the Log Mask then the log of that type will print on the USB port. Use the console log command to change the log for the current session (value not saved in eeram) until reboot or use the console riv command to set the eeram log value.

Bit Position	Log Identifier
0 (1)	TRACE
1 (2)	INFO (default)
2 (4)	WARNING (default)
3 (8)	ERROR (default)
4 (16)	CONSOLE
5 (32)	PORT1
6 (64)	PORT22
7 (128)	EERAM
8 (256)	FLASH
9 (512)	IO
10 (1024)	CNTL
11(2048)	IO Expander

When IO is enabled the IO log shows all IO including on board temperature every 1 second. When a file write is in progress, the IO log is temporarily disabled to improve readability of the write progress.

## IO Definition

Value	Type	Pull Up and Pull Down
Inputs		
0	DISABLED	None
1	DIGITAL	Pull up 10k no pulldown
2	PULSE	Pull up 10k no pulldown
3	RUNTIME	Pull up 10k no pulldown
10	ANALOG_RAW	No pull up no pulldown
11	ANALOG_0_TO_10_VOLT	No pull up. Vin through 22k then 10k pulldown
12	ANALOG_PERCENT	No pull up. Vin through 22k then 10k pulldown
13	RESISTANCE_THERMISTOR_10K_DEGC. Default	Pull up 10k no pulldown
14	RESISTANCE_THERMISTOR_10K_DEGF	Pull up 10k no pulldown
15	RESISTANCE_OHMS	Pull up 10k no pulldown
Outputs		
30	DIGITAL (0-11V). Default for DOT/DOR	
32	PWM	
33	PULSE	
40	ANALOG_0_TO_10_VOLT Percent. Default for UO	
41	ANALOG_2_TO_10_VOLT Percent	
42	ANALOG_0_TO_5_VOLT Percent	
43	ANALOG_0_TO_10_VOLT Volts	

## AI Threshold and Response

---

Refer to **System Flash**. This is a set of registers that are applicable to all Universal Inputs to determine how the controller filters and cleans up the measured value. The new measured value is subtracted from the old measured value and the *difference* is calculated.

A threshold counter determines the number of consecutive 10mSec cycles the difference exists. The difference must be in the same direction for *threshold* counts. On reaching this, the lowest of the *difference* or *response* is added/subtracted to/from the current value. Threshold default 20. Response default 8.

## DI Threshold and Debounce

---

Threshold: Raw A/D value that determines the switch point. Default 1024. This value measured can be set in volts as  $3.3 \times (\text{value}/4096)$ . Example 1024 gives a switch point when measured on the input terminals of 0.825V

Debounce: Number of consecutive 1mSec scan cycles where the input must be at the same value. For fast scan points (pulse counting at >10hz) do not set this above 3. Default is 3. Maximum value 32000 = 32 seconds.

## AI Calibration (Associated Value)

---

For Analog input types this is added to the final calculated value as an offset. The value is the same type as the input.

This is in the flash section as the calibration value corrects for wiring resistance and thermistor variations. A small part is to calibrate out the electronics in the controller. To ensure that the calibration values carry over if a controller is replaced, they are in the flash and not eeprom.

## AO Calibration (Associated Value)

---

Allows for adding/subtracting an offset to the Analog output voltage. The float value is added/subtracted to the voltage driver. Default 0.0.

Example – if an output is 2 to 10V Percent then adding 10% will add 10% of 8V = 0.8V.

## IO CONFIGURATION

---

IO operation is set by the IO Type Definition register for each IO. This determines what the IO is used for and matches the internal processing to the field device – refer to **System Flash**. Additional configuration is set in the Associated Configuration register such invert or which edge to operate on. Divider/Pulse register sets the Runtime divider, PWM period and Pulse time. Refer to **System EERAM – Esys**.

Some IO types do not require additional configuration – such as temperature input. In this case, these registers are ignored.

### DI

---

Sets the value the input will be set to if the input is at 0 volts. This normally equates to the contact closed state. The default is 1 meaning the register value will be 1 when the input has a closed volt free contact and a measured voltage across the terminals of less than :  $\text{DI Threshold value} / 4096 \times 3.3$ . As a default  $1024/4096 \times 3.3 = 0.825\text{v}$ . Note – the value must be 0 or < 255.

## DIVIDER/PULSE

---

### Runtime Divider

---

If a DI is configured as Runtime, the input is scanned every 1Sec and processed against the runtime divider. The updated runtime value is written to EERAM according to the EERAM rules. Example – change the runtime from seconds to minutes by setting the divider to 60. This means the runtime divided value is saved and not the raw count. Allows the user to extend the range of the uint32 counter. Default 1

## Pulse Divider

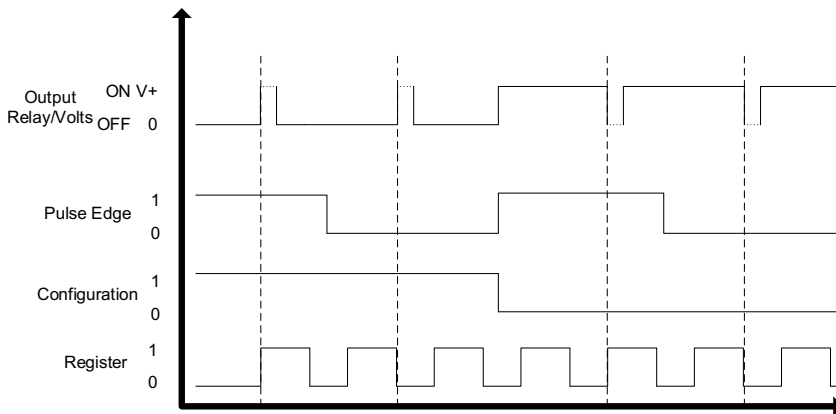
If a DI is configured as Pulse, the input is scanned every 10mSec and processed against the pulse divider. Example - pulse divider is set to 10. EERAM pulse count will increment for every 10 pulse inputs. The updated pulse count value is written to EERAM according to the EERAM rules. if pulse divider value is 0 or 1 then no pulse divider is applied. A value above 1 will divide the pulse by the divider value. Default 1

## DO/DOT/DOR

The lower order byte (bits 0-7) determines what the digital output register displays when the output is on. For DO/DOT, on means the voltage is present. For DOR, on means the relay is energized.

- Configuration 0. Output register will be 0 for off and 1 for on. Default
- Configuration 1 (or non-zero). Output register will be 0 for on and 1 for off.
- Configuration does not apply to a PWM output.

The higher order byte (bits 8-15) are the pulse edge - applicable to pulse type outputs only. This allows the pulse to be generated at the leading or trailing edge of the register change. Refer to the diagram below to determine the relationship between Configuration, Pulse edge and output. Example Low order byte 1 and high order byte 1 gives a decimal value of 257. This will give a pulse voltage pulse for the duration of the set time.



## UO/DOT/DOR PWM or PULSE Period

**PWM:** This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.

**Note1** – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)

**Note2** – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.

**Note3** - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.

**PULSE:** This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.

**Note1** – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.

**Note2** - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is > 0. The output will then go off after the pulse time – because the output register is > 0 the LED will be on and the output off. USB logs show both the register and physical output.

## IO Values

Input and Output values are in the Volatile System area. Additional IO information are in the EERAM System Area. For AI/AO/PWM the calibration value. For DI Runtime, Pulse Count. For all IO types, Manual Override and Manual Override Value.

### SPECIAL SYSTEM Register 698 and 699

Special Registers			
	Reboot – Write 0x55AA (21 930). Device will reboot	698	
	Factory Reset – Write 0x1928 (6 440). Eeram is written with 0 and eeram initialization flag cleared. SPIFFS erased. Reboot initiated	699	
	Configuration Reset – Write 0xAA55 (43 605). Firmware defaults written to reg fsys.txt and to flash system register logical area.		
	Eeram System Reset – Write 0xAB47 (43 847). Values read from flash file reg_esys.txt to eeram system register logical area. Only successful if the file exists on SPIFFS		
	Eeram User Reset – Write 0x12FE (4 862). Values read from flash file mb_obj_euser.txt to eeram user register logical area. Only successful if the file exists on SPIFFS		

#### Reboot

On receipt of reboot. Power LED 100msec blink for 5 seconds. Device reboots and normal reboot LED's continue.

#### Factory Reset

On receipt of factory reset. All LED's ON for 1 second. All LED's 100msec blink while reboot. The EERAM is erased and all files erased on flash. LED's after boot will show this.

#### Configuration Reset

On receipt of configuration. Firmware defaults written to reg fsys.txt and to flash system register logical area. Both Port1 and Port2 LEDs 500msec blink for 2 seconds.

#### Write Configuration to File. Register 407

User command to write configuration file images of FLASH or EERAM for both system and User spaces to SPIFFS. This allows FLASH default values such as fallback or configuration parameters to be saved once the device is configured and provides a default EERAM set of values to be loaded.

Value	Function
1	Flash system write next register received to SPIFFS obj_mb_fsys.txt
2	Flash system write entire flash system section to SPIFFS obj_mb_fsys.txt
3	Flash user write next register received to SPIFFS obj_mb_fuser.txt
4	Flash user write entire flash system section to SPIFFS obj_mb_fuser.txt
5	Eeram system write next register received to SPIFFS obj_mb_esys.txt
6	Eeram system write entire eeram system section to SPIFFS obj_mb_esys.txt
7	Eeram user write next register received to SPIFFS obj_mb_euser.txt
8	Eeram user write entire eeram user section to SPIFFS obj_mb_euser.txt

To prevent a write configuration from remaining in the system, the register table write command must be sent within 15 seconds of the write configuration command. This prevents the scenario where a user sets the configuration then is interrupted and forgets it is set.

To read the eeram files into eeram and register table, use **SPECIAL SYSTEM Register 698 and 699**.

## Boot Delay. Register 465

---

Register 465 defines the number of seconds to delay after reboot before executing control blocks. This allows the Inputs to settle through the A/D converters and be filtered to represent the correct measured value before applying control logic.

## Factory Test. Register 404

---

Executes the factory test procedure. This is not a user feature and is intended for production testing. Do not use unless the correct training is received. The factory test requires a code to be entered to put the device into factory test mode. Rebooting the device or clearing the code from the register will return the controller to normal operation.

## Port1 (405) and Port2 (406) Communication quality

---

For each valid communication frame this value is incremented to a maximum of 100. When there is an invalid frame or timeout, this value is decremented to a minimum of 0. This represents the communications quality over the last 100 data packets expressed as a % where 0% comms indicates fail and 100% comms excellent.

Note: If there is no communications (as opposed to corrupt communications) for 10 seconds consecutively, the value will decrement by 10%. This means that the comms value will go from 100% to 0% over 100 Seconds.

## Health

---

Health flag (refer to register table) indicates state of health as the device progresses through initialization of the hardware components. If the LED's indicate a boot error or to check if there are any components not booting correctly, look at the health register using a console rir command. A value of 0 indicates good health. Refer to the device register table for the health flag index.

Bit interpretation of the health flags.

Value	Descriptor	Define
0 (1)	EERAM failed initialization	h_EERAM_i
1 (2)	EERAM failed last read	h_EERAM_r
2 (4)	EERAM failed last write	h_EERAM_w
3 (8)	EERAM – not previously used	h_EERAM_d
4 (16)	EERAM – default loaded	h_EERAM_f
5 (32)	FLASH failed initialization	h_FLASH_i
6 (64)	FLASH failed last read	h_FLASH_r
7 (128)	FLASH failed last write	h_FLASH_w
8 (256)	FLASH reg_fsfs.txt default created	h_FLASH_d
9 (512)	FLASH default file load failed	h_FLASH_f
10 (1024)	FLASH SPIFFS mount failed	h_FLASH_m
11 (2048)	IO Expander Failed	h_IO_f
12 (4096)	Port1 config error. Defaults used	h_UART1_error
13 (8192)	Port2 config error. Defaults used	h_UART2_error
14 (16384)	Boot Error	h_boot_error
15 (32768)	Boot in progress	h_boot

## 1 Second Pulse Train

---

Register 399 is a 1 second pulse train. 1 Second ON and 1 Second OFF. This can be used to create flashers or to trigger time-based events.

## Register Initialization and Reboot Process

---

1. Factory reset, first boot after manufacture. (Eeram not initialized)
  - a. EERAM system and user registers set as per register table type defaults with value 0.
  - b. Flash system and user registers set as per register table defaults with value 0.
  - c. Volatile system and user registers set as per register table defaults with value 0.
  - d. If reg\_fsys.txt does not exist create with system defaults. Model dependent.
  - e. Read reg\_fsys.reg type and value to register table.
  - f. If reg\_fuser.txt exists read type and value to register table.
2. Power cycle or reboot (eeram initialized)
  - a. EERAM system and user registers set as per register table type defaults and values populated from EERAM.
  - b. Flash system and user registers set as per register table defaults with value 0.
  - c. Volatile system and user registers set as per register table defaults with value 0.
  - d. If reg\_fsys.txt does not exist create with system defaults. Model dependent.
  - e. Read reg\_fsys.reg type and value to register table.
  - f. If reg\_fuser.txt exists read type and value to register table
3. Eeram user reset (eeram initialized)
  - a. Files reg\_fsys.txt, reg\_fuser.txt read from SPIFFS and written to respective registers.
  - b. Files reg\_euser.txt read from SPIFFS and written to respective registers – eeram user is also updated. The only other activity that will trigger reg\_euser.txt to update user eeram is by writing a new or overwriting an existing reg\_euser.txt file from the console.

## LED Indication

---

### Boot

---

1. On boot all Power/Status/Port1/Port2 on for 2 seconds. OK
2. Status/Port1/Port2 LED Off.
3. Port2 LED ON for 2 to seconds while checking for EERAM chip.
4. Port2 LED 100msec blink for 5 seconds if EERAM chip not present. Go to Exit EERAM init.
5. Port2 LED 500msec blink for 2 seconds if EERAM need initialising. (Never used before or on factory reset)
6. Port2 LED 100msec blink for 5 seconds if EERAM error.
7. Exit EERAM init.
8. Port2 LED OFF.
9. Port1 LED ON for 2 to seconds while checking for FLASH chip.
10. Port1 LED 100msec blink for 5 seconds if FLASH chip not present. Go to Exit FLASH init.
11. Port1 LED 100msec blink for 3 seconds if no default files on FLASH.
12. Port1 LED 500msec blink for 5 seconds if writing defaults FLASH error. Go to Exit FLASH init.
13. Exit EERAM init.
14. Port1 LED OFF.
15. Status LED ON.
16. Power LED FLASH100 if boot error. This will remain till boot error cleared.

### Reboot

---

Refer to Special Registers: On receipt of reboot. Power LED 100msec blink for 5 seconds. Device reboots and normal reboot LED's continue.

### Factory Reset

---

Refer to Special Registers: On receipt of factory reset. All LED's ON for 1 second. All LED's 100msec blink while reboot. The EERAM is erased and all files erased on flash. LED's after boot will show this.

### Configuration Reset

---

Refer to Special Registers: On receipt of configuration. Firmware defaults written to reg\_fsys.txt and to flash system register logical area. Both Port1 and Port2 LEDS 500msec blink for 2 seconds.

### Normal Operation

---

<b>Power LED (Blue)</b>	
ON	Boot OK.
Flash 0.5 seconds	Boot Error. Requires reboot to clear. Check Health Flag for cause of error.
OFF	No power or device in abnormal state
<b>Status LED (Blue)</b>	
ON	ALL OK
Flash 1 Seconds	USB Connected
Flash 0.5 seconds	Read/Write EERAM or Flash
OFF	Error. Check log and reboot device
<b>Port1 and Port2 LED (Blue)</b>	
Flash	Tx/Rx data

## Input and Output LED's

Each IO has an associated terminal number with an IO LED that for visual indication. The LED conveys information based on the IO configuration as follows:

Type	LED
DISABLED	Off
DIGITAL	Follows the IO register. 0 = LED Off. >0 = LED On.
PULSE WIDTH	If PWM <2% LED Off else LED On.
RUNTIME	Follows the IO register. 0 = LED Off. >0 = LED On.
<b>UI</b>	
ANALOG_RAW	If IO <0.2V LED Off else LED On. 0 to 4096 = 0 to 10.6V
ANALOG_0_TO_10_VOLT	If IO <0.2V LED Off else LED On
ANALOG_PERCENT	If IO <0.2V LED Off else LED On. 0-10V as 0-100%. >100% means >10v
RESISTANCE_THERMISTOR_10K_DEGC. Default	If IO <9.8C or Rin > 59.8 LED Off else LED On
RESISTANCE_THERMISTOR_10K_DEGF	If IO <9.8F or Rin > 139F LED Off else LED On
RESISTANCE_OHMS	If IO Rin < 100 $\Omega$ or Rin > 52k $\Omega$ LED Off else LED On. 100k is open circuit
<b>UO</b>	
ANALOG_0_TO_10_PERCENT	If IO <0.2V LED Off else LED On
ANALOG_2_TO_10_PERCENT	If IO <2.2V LED Off else LED On
ANALOG__0_to_5_PERCENT	If IO <0.2V LED Off else LED On
ANALOG_PERCENT	If IO <0.2V LED Off else LED On. 0-10V as 0-100%. >100% means >10v
ANALOG_0_TO_10_volt	If IO <0.2V LED Off else LED On

## Personality Text Files

The personality – or how the device has been configured is determined by text files with very specific names and functions as defined below. The text files are loaded using the desktop application [edgeUP-App](#).

Text files contain ASCII JSON strings that embed configuration information. There are multiple files each with a specific file name that identifies the use case.

### Restrictions on files

1. The maximum number of lines is 300.
2. Maximum file name length including .txt extension 20
3. Maximum file line length 200
4. Maximum number of digits in a number 12

The files are shown in **Figure 4**.

Some of the files are license dependent. As an example, if the device is not licensed for control blocks, the control.txt file will be ignored at reboot. The same goes for bacnet.txt if the device is not registered for BACnet.

Files reg\_euser.txt and reg\_esys.txt are only read by special register command. Refer **Write Configuration to File. Register 407**

reg\_esys.txt is loaded at reboot if the system detects that the EERAM is uninitialized. Refer **Health**. If no file is found then the system writes the default values as per the register table.

If reg\_fsys.txt is not found at reboot, the system creates the file and writes the defaults as per the register table. Both reg\_fsys.txt and reg\_fuser.txt are loaded at reboot.

Fixed Application Controllers (FAC) have a special file that is loaded at factory. fac.txt lists the control logic for fixed application controllers and does not require a license to run. These files are only read and executed by fixed application controllers. User modification of the file will result in the FAC failing to execute the control application.

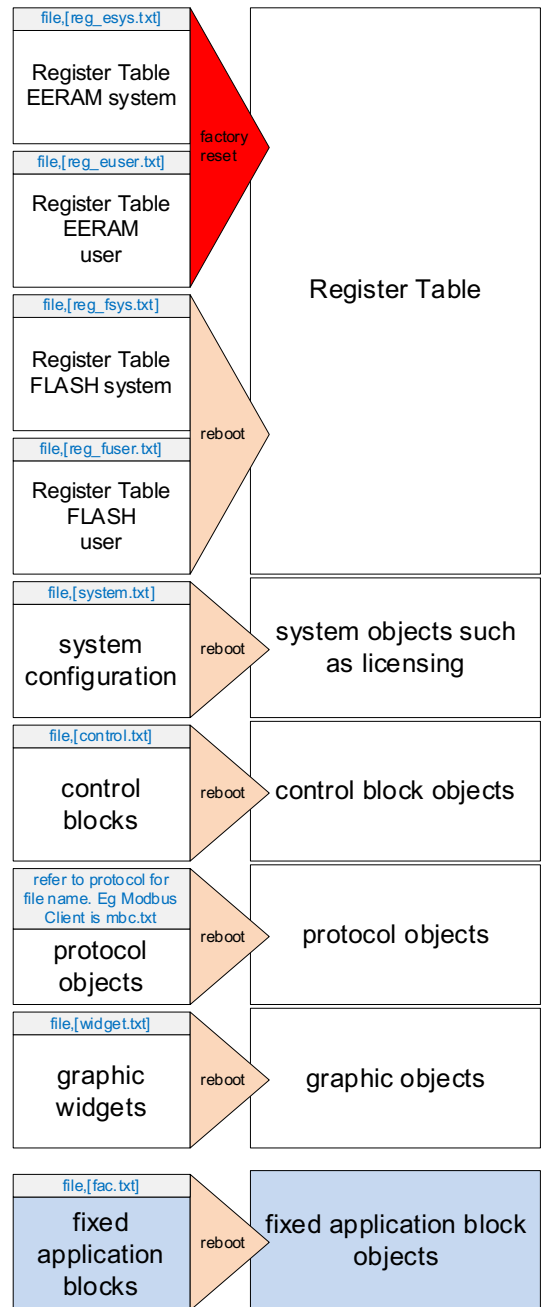


Figure 4

## JSON Strings

---

Each line of the text file follows a common format.

<param0>,[<param1>,<param2>,...<paramX>]

line is terminated with CR or LF or CRLF. No other control characters allowed.

<param0> defines the use, from <param1>..<paramX> define the use case.

Only JSON strings relevant to the file will be processed. As an example, a cb function in the reg\_esys.txt will not be processed.

<param0> Each file is listed with the allowable param0 types.

1. system.txt
  - file,[] (must be on line 1)
  - lic,[] (must be on line 2)
2. control.txt
  - file,[] (must be on line 1)
  - cb,[] where cb refers to the control block number
3. reg\_esys.txt, reg\_euser.txt
  - file,[] (must be on line 1)
  - riv,[]
4. reg\_fsys.txt, reg\_fuser.txt
  - file,[] (must be on line 1)
  - reg,[]
  - riv,[]
5. mbc\_txt
  - file,[] (must be on line 1)
  - mbc01,[]
  - mbc02,[]
  - reg,[]
6. bacnet.txt
  - file,[] (must be on line 1)
  - dev,[]
  - ai,[]
  - ao,[]
  - av,[]
  - bi,[]
  - bo,[]
  - bv,[]
  - msv,[]
7. fac.txt
  - file,[] (must be on line 1)
  - cb,[] where cb refers to the control block number
  - reg,[]
8. widget.txt
  - Not used on PP-COM-IO. Refer to PP-View for graphical widgets

<file>

---

file,[filename.txt]

The first line in system.txt identifies the file name to provide a check that the correct file is processed. No white spaces are allowed in the file name and will be removed. Example file,[file name] is not allowed as there is a white space and the extension .txt is left out.

<lic>

---

lic,[license number]

The second line in system.txt is the supplier issued license number. This license number identifies the additional features licensed. If this line is excluded then the device will operate in default mode with no additional features such as control block capability.

**<cb>**

cb,[Refer to Control Block user manual for parameter interpretation.]

Each control block is identified by a control block number and has a unique syntax. The control block identifier determines the control block function. Each control block has numbered parameters. These must be entered in the sequence described for the control block.

**Example**

1. 80,[13,1,10.0,3]
  - a. cb - Control Block type – 80 is a Math1
  - b. param1 – 13 Divide. Operand 1 register Operand 2 constant value
  - c. param2 – 0 is Operand 1 register table address. Note float uses 2 register spaces 0 and 1
  - d. param3 – 10.0 is operand 2 – the divisor as a constant.
  - e. param4 – is for the result register 2. Note float uses 2 register spaces 2 and 3

**<reg>**

reg,[<register table index>,<register table type>,<register table default value>]

Defines a register in the register table.

<register table index>. Refer to Register Table. [Register Table](#)

<register table type>. Refer to [Data Type](#)

<register table default value>. The value written to the register table. Must be consistent with the Data Type range.

**Examples**

1. reg,[1,2,50]
  - a. param1 – register table index (register 1)
  - b. param2 – register number type (uint16)
  - c. param3 – value. 50
2. reg,[2,7,50.0]
  - a. param1 – register table index (register 3)
  - b. param2 – register number type (float)
  - c. param3 – value. 50.0
  - d. NOTE this register is 32 bits wide. Takes 2 register locations so the next available register is 4

**<riv>**

riv,[<register table index>, register table default value>]

Defines a register in the register table.

<register table index>. Refer to register table. [Register Table](#)

<register table value>. The value written to the register table.

**Examples**

1. reg,[1,50]
  - a. param1 – register table index (register 1)
  - b. param2 – value. 50
2. reg,[2,50.0]
  - a. param1 – register table index (register 3)
  - b. param2 – value. 50.0
  - c. NOTE this register is 32 bits wide. Takes 2 register locations so the next available register is 4

<dev>

---

dev,[name, description, location]

<param1>. Name (max 32 characters). If not set then shows "Name Not Defined"

<param2>.Description (max 32 characters). If not set then shows "Description Not Defined"

<param3>. Location (max 32 characters). If not set then shows "Location Not Defined"

If text is in "" then white spaces and printable characters are retained. If not, all white spaces and non-printable characters are removed.

<ai> <ao> <av>

---

<ai>,[instance, name, engineering units]

<ao>,[instance, name, engineering units]

<av>,[instance, name, engineering units]

<param1>Instance of object. For AI and AO this is the sequential number starting from 1. For AV, his is the register table address. Register table points must be defined and the register type set for the point to be available to BACnet. Refer to reg\_xxxx.txt for setting up the register table. For 32 bit registers, the first (lowest) address is used. For physical points (ai,ao) the instance is the same as the index. For av, the index is created as the points are read from the text file so the instance (register) is different to the index (pointer into av array).

<param2>.Name. Max 16 characters. White spaces are allowed in text strings between the [" "]. If "" not used, the white spaces will be removed.

<param3>. Engineering Units (from BACnet Enum)

For clarity, the difference between index and instance are explained. Use a list of people that must be assigned a name as an example. As the list is read, the names are written down. 1<sup>st</sup> name on line 1 and so on until all ten names are written down. The line number is the index and the name the instance. Give me the name of the person on line 1 uses the index to give the name or instance. Conversely you can ask for the name (instance) of line 1. From this example we can see that the index is the line number or index into an array and the instance are the contents.

<bi> <bo> <bv>

---

<bi>,[instance, name, state0,state1]

<bo>,[ instance, name, state0,state1]

<bv>,[ instance, name, state0,state1]

<param1>Instance of object. For BI and BO this a sequential number starting from 1. For BV Instance of physical point or register table point. Register table points must be defined and the register type set for the point to be available to BACnet. Refer to reg\_xxxx.txt for setting up the register table. For 32 bit registers, the first (lowest) address is used. For physical points (bi,bo) the instance is the same as the index. For bv, the index is created as the points are read from the text file so the instance (register) is different to the index (pointer into av array).

<param2>.Name. Max 16 characters. White spaces are allowed in text strings between the [""]. If "" not used the white spaces will be removed.

<param3>. State0 text. Max 8 characters

<param4>. State1 text. Max 8 characters

## <msv>

<msv>,[instance, name, state0,state1,state2,state3,state4,state5,state6,state7]

<param1>Instance of object. Instance of physical point or register table point. Register table points must be defined and the register type set for the point to be available to BACnet. Refer to reg\_xxxx.txt for setting up the register table. For 32 bit registers, the first (lowest) address is used. The index is created as the points are read from the text file so the instance (register) is different to the index (pointer into msv array).

<param2>. Name. Max 16 characters. White spaces are allowed in text strings between the [""]. If "" not used the white spaces will be removed.

<param3>. State0 text. Max 8 characters

<param4>. State1 text. Max 8 characters

<param5>. State2 text. Max 8 characters

<param6>. State3 text. Max 8 characters

<param7>. State4 text. Max 8 characters

<param8>. State5 text. Max 8 characters

<param9>. State6 text. Max 8 characters

<param10>. State7 text. Max 8 characters

## Console

A text console for direct commands is available on the USB port. Command echo is on meaning the character typed is echoed back. If a command is not completed. The console will time out after 5 seconds. To check if the console is connected, press enter – reply will be error if USB connected and device communicating. Console commands are used to communicate with the edgeUP-App or any terminal program.

## Console Commands

Command	Description	Implemented
?	Help - prints out the list of commands.	Y
dir	SPIFFS directory.	Y
log	Log verbosity to USB. Refer <b>USB Logs – Log Verbosity. Register 551</b> . Not written to flash	Y
printn	print no line numbers - then the file index as listed by SPIFFS directory.	Y
printl	print with line numbers - then the file index as listed by SPIFFS directory.	Y
reboot	reboot the device - useful after loading personalities.	Y
writet	write a text file - replies ok. Send the file name followed by each text line. All lines are ack'd (0x06).	Y
del	delete - then the file index as listed by SPIFFS directory.	
info	Product Family, model, revision, serial number, licence and personality	Y
format	format SPIFFS.	Y
rit	register index type. Used to set the <b>Data Type</b> of the register. rit index type. Example rit 20 7 will set register 20 to float	Y
riv	register index value. Used to set the register value. riv index value. Example riv 20 12.5 will set register 20 to 12.5	Y
rir	register index read. rir 20 will reply <i>rir Register index:20: Type:7: Value:12.50000: ok&gt;</i>	Y

## Ports - UART – Communications Parameters

---

RS485 Port 1 is available on all models and operates a Modbus Server on unlicensed models. An optional second RS485 port is available on models with 2 ports populated. The second port has additional RS485 hardware and must therefore be ordered as a 2 port device. Refer to the part numbers for ordering.

The communication parameters for these ports are stored in non-volatile flash system register table (Refer **System Flash**).

Default values for both are 9600 baud with address 1 for the server. The data bits are fixed at 8 and stop bit.

Baud Rate options are 1200 to 57600. No range checking is done between these 2 values. Ensure full baud number is entered Do not use 96 for 9600 as an example.

The address range must be between 1 and 255. Values outside this range will not be accepted.

Each serial port has 2 timing parameters – the first is Timeout and second is Turnaround.

1. Timeout is the time from the start of a packet to when the reply is complete and the UART released. This includes the Rx time, turnaround time and Tx time.
2. Turnaround is the time allowed from Transmit/Receive of last byte to Transmit/Receive of first byte.

The default times in mSec are in the table below – these can be adjusted by entering an offset value (-5000 to +5000 mSecs). The offset is added to the default.

Baud Rate	Port1 Timeout	Port1 Turnaround	Port2 Timeout	Port2 Turnaround
1200	1000	1	1000	1
2400	500	1	500	1
4800	250	1	250	1
9600	125	1	125	1
14400	83	1	83	1
19200	62	1	62	1
38400	31	1	31	1
57600	20	1	20	1

## Ports and Protocols

---

Ports depend on the model and the part number (Refer to [PP-COM-IO-SelectionTool.xlsx](#)). Remember to save a copy of this file to your device to enable the drop-down lists to operate.

Example for IO-UI8 with Port1 and Port2

- PP-COM-NNP-UI8-MSMP

NNP means both ports are non-isolated and the aux power is the same as the device power. MS indicates Port1 is Modbus Server and Port2 is Modbus Client.

Protocols supported on each port differ so be sure to use the selection tool to ensure the correct hardware model is selected to match the required protocol.

## Modbus Server – Port1

---

Modbus server is supported on Port1 pins A and B. Default baud rate is 9600 with address 1 as per the Flash system register table. The number of controllers on the Modbus network must be limited to the RS485 driver load and required Modbus network throughput. Recommended to not exceed 32 devices per Modbus network. Maximum PP-COM-IO devices allowed is 64 (RS485 chip is ¼ load).

Modbus commands are:

- 0x03 Read Holding registers (Max 16 registers or 32 bytes)
- 0x06 Write single register.
- 0x10 Write multiple registers (Max 16 registers or 32 bytes)

Refer to writing system flash and eeprom for sequence.

## Modbus Client – Port2

---

Modbus client is supported on Port2 terminals A and B.

Modbus commands are:

- 0x03 Read Holding registers (Max 16 registers or 32 bytes)
- 0x06 Write single register.
- 0x10 Write multiple registers (Max 16 registers or 32 bytes)

Multiple baud rates are supported where the baud rate is set according to the protocol block settings. The default baud rate is set according to the register table. Refer to **System Flash**.

Modbus Client requires an optional license. This allows the client Modbus blocks to be created to match the address, baud rate and register type/address.

The following Modbus Client functions are supported

## Communication Blocks – Modbus client – Port 2

---

A set of control blocks to perform protocol specific communication tasks. These blocks are only available if the protocol is supported on the controller hardware with the defined port available. This is an addition to Control Blocks and will not operate unless the firmware supports Control Blocks. Please confirm the part number to verify this.

Communication blocks are used to setup the register mapping – Refer to Register Mapping in the User Manual. Each control block is executed each cycle period (in seconds). The blocks are defined in the control.txt file which is loaded on boot.

## Modbus Scan cycle and Trigger

---

The scan cycle and trigger provide a method of reducing Modbus traffic.

Each trigger as defined in the trigger table below will execute the block and be put into the Client queue on the next 1 second scan cycle for transmission.

Scan Cycle - If the scan cycle is 0, the block is not processed in the 1 second scan cycle 1. Any non-zero scan cycle value executes the block each *scan cycle* number of seconds. By combining the trigger and a long scan cycle – such as 30m minutes (1800 seconds), a reliable method of triggering the block as well as periodic updates is achieved to ensure the Server data is always valid.

Trigger	Action
0	No Trigger
1	Leading Edge. From 0 to 1
2	Trailing Edge. From 1 to 0
3	Both Edges. From 0 to 1 and 1 to 0
4	Hi. If trigger > 0
5	Lo. If trigger = 0

## Client Queue Management

---

Modbus Client blocks are scanned every second and sent to the Client queue on for transmission. The client queue sequentially sends out the messages FIFO (first in first out) until the queue is cleared. This ensures sequential handling of messages and prevents offline or error transmissions from blocking up the queue. Each block is guaranteed to be sent before the next round of transmissions. This means that for heavily loaded systems or where there are multiple offline devices, the scan cycle will not be adhered to.

Note – the client queue limit is 20 messages – any system with more than 20 client blocks can expect lost packets in large payload packets or with offline devices.

mbc01 - Read Holding (F03)
 

---

Read 1 or more registers with data format from the server address at the baud rate. Convert to little endian byte swap and save at the address in the register table. The block can be time or trigger initiated.

Name	Param Index	Entry Data Type	Default Register Type	Description
device address	1	uint8	constant	Address of device
device baud	2	uint16	constant	Baud rate from available baud rates. 0 to use the default baud rate.
device register address	3	uint16	constant	Destination register address according to the manufacturers addressing scheme
device register quantity	4	uint8	constant.	Number of registers to read. Max quantity 16
device data format	5	uint8	constant	The registers will be read in the data type defined for the register table at the specified address and converted to the device data format. Refer <a href="#">Error! Reference source not found.</a>
register table address	6	uint16	as per data type	register table destination address – must be in user address range. If value is to be read outside this range use a transfer block to duplicate the register into the system address range.
scan cycle	7	uint16	constant	Number of seconds between consecutive communications. Scan cycles up to 65 536 seconds. 0 for no scanning. <b>Modbus Scan Cycle and Trigger</b>
trigger	8	uint8	constant	0 for no trigger. 1 For leading edge trigger. 2 for trailing edge trigger
trigger register	9	uint16	uint16	if trigger > 0 then any valid register table address in the user space.
trigger retries	10	uint8	constant	if trigger > 0 then the number of retries when a trigger is activated. The retries will be sent irrespective if the previous transmit was successful.

**Initialization error codes**

- 1 Invalid data type
- 2 no index found
- 5 Register out of table range
- 6 Register + qty out of table range
- 7 Reg Qty > max
- 8 Trigger value out of range
- 9 Data format out of range

mbc02 - Write Holding (F06 F16)

Write 1 or more registers from the register table to the server address at the baud rate. Convert from little endian byte swap to data format of the device and save at the address in the device. The block can be time or trigger initiated.

Name	Param Index	Entry Data Type	Default Register Type	Description
device address	1	uint8	constant	Address of device
device baud	2	uint16	constant	Baud rate from available baud rates. 0 to use the default baud rate.
device register address	3	uint16	constant	Destination register address according to the manufacturers addressing scheme
device register quantity	4	uint8	constant.	Number of registers to write. 0 means use F06 to write 1 register. A value >0 <=16 means write using F16. Max quantity 16. Remember that all 32 bit registers use 2 x registers.
device data format	5	uint8	constant	The registers will be read in the data type defined for the register table at the specified address and converted to the device data format. Refer <a href="#">Error! Reference source not found.</a>
register table address	6	uint16	as per data type	register table source address – must be in user address range. If value is to be written outside this range use a transfer block to duplicate the register into the system address range.
scan cycle	7	uint16	constant	Number of seconds between consecutive communications. Scan cycles up to 65 536 seconds. 0 for no scanning. <b>Modbus Scan Cycle and Trigger</b>
trigger	8	uint8	constant	0 for no trigger. 1 For leading edge trigger. 2 for trailing edge trigger
trigger register	9	uint16	uint16	if trigger > 0 then any valid register table address in the user space.
trigger retries	10	uint8	constant	if trigger > 0 then the number of retries when a trigger is activated. The retries will be sent irrespective if the previous transmit was successful.

Initialization error codes

- 1 Invalid data type
- 2 no index found
- 5 Register out of table range
- 6 Register + qty out of table range
- 7 Reg Qty > max
- 8 Trigger value out of range
- 9 Data format out of range

Modbus Register Sequence

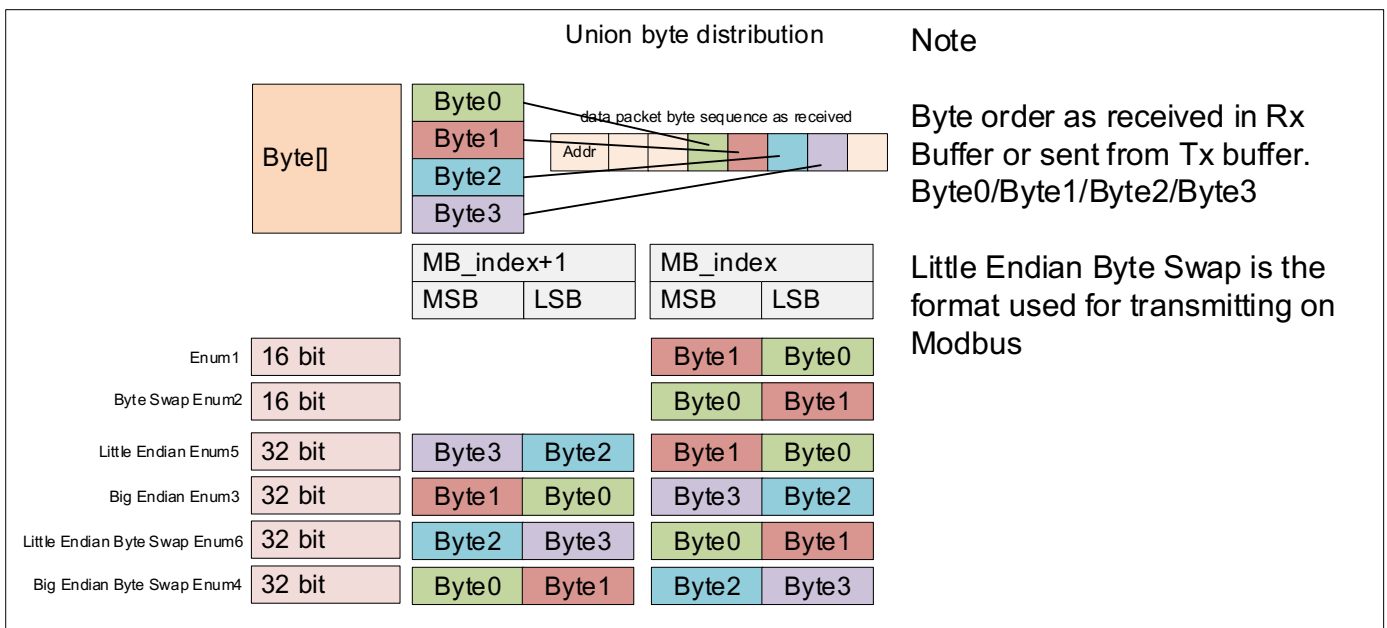


Figure 5

## Endianness and Byte Swap

Endianness refers to the order in which bytes (8 bits) are arranged in a multi-byte data value. There are two types of endianness: big-endian and little-endian.

In big-endian (also known as network byte order), the most significant byte (MSB) is stored first in memory (MB\_index), followed by the less significant byte (LSB) (MB\_Index+1). This means that the MSB has the lowest memory address, while the LSB has the highest memory address.

In little-endian the least significant byte (LSB) is stored first in memory (MB\_index), followed by the most significant byte (MSB) (MB\_Index+1). This means that the LSB has the highest memory address, while the MSB has the lowest memory address.

Byte swap refers to the order of the bytes when transmitted in a word. For a 16 bit word made of 2 x bytes, the normal order is Byte 0 occupies the lower position and Byte 1 the higher. Example the value of a 16 bit word is 0x1234. If the MSByte is transmitted first 0x12 followed by 0x34 this is the normal order. In byte swap the bytes are transmitted 0x34 then 0x12. Following the diagram above, if the bytes are transmitted from the lowest MB\_index starting at the lowest byte then the data will be transmitted as little endian byte swap.

As an example, the float value of 25.2 (0x41C9 999A) when stored in the register table will have register address [x] the 16 bit word 999A and register [x+1] 41C9. This is little endian.

An online converter [Floating Point to Hex Converter \(gregstoll.com\)](http://gregstoll.com)

## Modbus Data Formats

data format value	data format type
1	16 bit
2	16 bit byte swap
3	32 bit big endian
4	32 bit big endian byte swap
5	32 bit little endian
6	32 bit little endian byte swap

## BACnet – Port1

BACnet MSTP is supported on Port1 pins A and B as a licensed option. Default baud rate is 9600 with MAC address 1 as per the Flash system register table. Note – the MAC address cannot be changed via BACnet. The console must be used to change the register value and save to flash or reg\_fsys.txt must be changed and save on the device.

Additional BACnet related settings are detailed in the Flash system register table.

MAC and Instance number are configurable. MAC address or the address on the MS/TP network can be between 1 and MAX Master (usually 127). This is the DIP switch setting on some controllers. The number of controllers on the MSTP network must be limited to the RS485 driver load and required MSTP network throughput. Recommended to not exceed 32 devices per MSTP bus. PP-COM-IO devices use RS485 ¼ load devices.

The device instance number is used for higher level communications and is set as a default to the hardware MAC address of the device. This can be changed as per rules define in the Flash system register table.

File bacnet.txt is used to define the BACnet object descriptors and user settings. The IO is created based on the IO configuration set in the Flash system register table. If the IO configurations are changed, reboot to re-configure BACnet to match.

## BACnet Capabilities Statement

### BACnet Controller

Controller Model: PP-COM-IO-UI8, PP-COM-IO-UI4UO4, PP-COM-IO-UI4DOT4, PP-COM-IO-DOR4  
 Manufacturer: Digital Twin (Pty) Ltd

Firmware/Software Version: >1.0.

BACnet Protocol Version Supported: BACnet Protocol Revision: 14

## Capabilities and Features

---

### Communication Protocols

---

#### BACnet MS/TP

#### Supported BACnet Objects and Services

---

##### Analog Input Objects

Property	Read	Write
PROP_OBJECT_IDENTIFIER	X	
PROP_OBJECT_NAME	X	
PROP_DESCRIPTION	X	
PROP_OBJECT_TYPE	X	
PROP_PRESENT_VALUE	X	X
PROP_UNITS	X	
PROP_STATUS_FLAGS	X	
PROP_EVENT_STATE	X	
PROP_OUT_OF_SERVICE	X	X
PROP_PROPERTY_LIST	X	
PROP_RELIABILITY	X	

Note the PROP\_UNITS refers to a list of BACnet Engineering Units. ([BACnet Engineering Unit Conversion-Software Toolbox](#))

##### Analog Output Objects

Property	Read	Write
PROP_OBJECT_IDENTIFIER	X	
PROP_OBJECT_NAME	X	
PROP_DESCRIPTION	X	
PROP_OBJECT_TYPE	X	
PROP_PRESENT_VALUE	X	X
PROP_UNITS	X	
PROP_STATUS_FLAGS	X	
PROP_EVENT_STATE	X	
PROP_OUT_OF_SERVICE	X	X
PROP_PROPERTY_LIST	X	
PROP_PRIORITY	X	
PROP_PRIORITY_ARRAY	X	
PROP_RELINQUISH_DEFAULT	X	
PROP_RESOLUTION	X	
PROP_RELIABILITY	X	
PROP_MIN_PRES_VALUE	X	
PROP_MAX_PRES_VALUE	X	

## Binary Input Objects

Property	Read	Write
PROP_OBJECT_IDENTIFIER	X	
PROP_OBJECT_NAME	X	
PROP_DESCRIPTION	X	
PROP_OBJECT_TYPE	X	
PROP_PRESENT_VALUE	X	X
PROP_ACTIVE_TEXT	X	
PROP_INACTIVE_TEXT	X	
PROP_STATE_TEXT	X	
PROP_STATUS_FLAGS	X	
PROP_EVENT_STATE	X	
PROP_OUT_OF_SERVICE	X	X
PROP_PROPERTY_LIST	X	
PROP_RELIABILITY	X	

## Binary Output Objects

Property	Read	Write
PROP_OBJECT_IDENTIFIER	X	
PROP_OBJECT_NAME	X	
PROP_DESCRIPTION	X	
PROP_OBJECT_TYPE	X	
PROP_PRESENT_VALUE	X	X
PROP_ACTIVE_TEXT	X	
PROP_INACTIVE_TEXT	X	
PROP_STATE_TEXT	X	
PROP_STATUS_FLAGS	X	
PROP_EVENT_STATE	X	
PROP_OUT_OF_SERVICE	X	X
PROP_PROPERTY_LIST	X	
PROP_PRIORITY	X	
PROP_PRIORITY_ARRAY	X	
PROP_RELINQUISH_DEFAULT	X	
PROP_RELIABILITY	X	

## Analog Value Objects

Property	Read	Write
PROP_OBJECT_IDENTIFIER	X	
PROP_OBJECT_NAME	X	
PROP_DESCRIPTION	X	
PROP_OBJECT_TYPE	X	
PROP_PRESENT_VALUE	X	X
PROP_UNITS	X	
PROP_STATUS_FLAGS	X	
PROP_PROPERTY_LIST	X	

## Binary Value Objects

Property	Read	Write
PROP_OBJECT_IDENTIFIER	X	
PROP_OBJECT_NAME	X	
PROP_DESCRIPTION	X	
PROP_OBJECT_TYPE	X	
PROP_PRESENT_VALUE	X	X
PROP_ACTIVE_TEXT	X	
PROP_INACTIVE_TEXT	X	
PROP_STATE_TEXT	X	
PROP_STATUS_FLAGS	X	
PROP_PROPERTY_LIST	X	

## Multistate Value Objects

Property	Read	Write
PROP_OBJECT_IDENTIFIER	X	
PROP_OBJECT_NAME	X	
PROP_DESCRIPTION	X	
PROP_OBJECT_TYPE	X	
PROP_PRESENT_VALUE	X	X
PROP_STATE_TEXT	X	
PROP_NUMBER_OF_STATES	X	
PROP_STATUS_FLAGS	X	
PROP_PROPERTY_LIST	X	

## BACnet Compliance

BACnet Standard Compliance: BACnet Terminal Equipment Controller [B-TEC]

### Supported Data Types

- BACnetNULL
- BACnetBOOLEAN
- BACnetUNSIGNED
- BACnetSIGNED
- BACnetREAL
- BACnetDOUBLE
- BACnetOCTET\_STRING
- BACnetCHARACTER\_STRING
- BACnetBIT\_STRING
- BACnetENUMERATED
- BACnetDATE
- BACnetTIME
- BACnetOBJECT\_ID
- BACnetDEVICE\_OBJECT\_PROP\_REF

### Special Functions and Interfaces

---

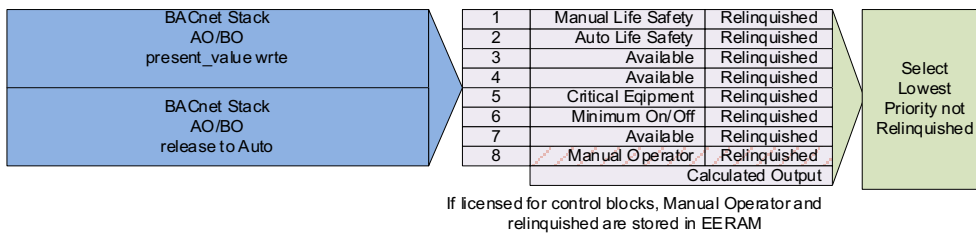
BACnet objects are created based on IO configuration and parameters contained in bacnet.txt file on flash. If the device is not licensed for control, manual override of Inputs and Outputs does not function. The out of service flag and priority arrays have no operation.

## BACnet Manual Override

Manual Override is supported on devices licensed for control. To allow Niagara and other systems to use this, the out of service flag as well as priority array are implemented as follows:

**Analog and Binary Outputs:** If the out of service flag is set or priority array 8 written with a manual override value, the manual override flag and value will be saved in EERAM and the output will be driven to the manual override value. **Analog and Binary Inputs:** Set the Out of Service flag as a writable point then use this to manually override the input. The measured value is replaced by the manual value. The Out of service flag and value are saved in EERAM. Refer to the section on IO Manual Override for more information.

Priority array 8 is used for manual override. If the manual override flag is set, this sets the out of service flag and clears the relinquished flag in priority array 8 to indicate that manual override is currently active. The EERAM override flag and manual override value are both saved to EERAM. On release to Auto the manual override flag is cleared in EERAM and the out of service flag cleared.



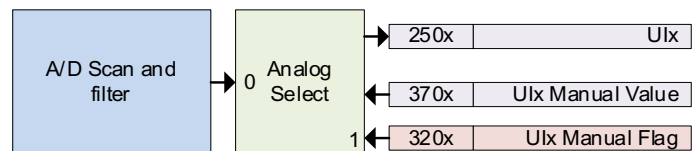
Note: Niagara maintains a local version of the override flag and only updates this from the controller at manual override or release from Niagara. This is fine if only Niagara BACnet is on the bus. To have a view of whether the point is overridden by the local terminal, or another communications channel add a binary readable point of the out of service flag for the point. If Yabe is used, the points are scanned and the flags updated if the point is dragged into the Periodic Polling area.

## IO Manual Override

Inputs and outputs have manual override registers only when licensed for Control Blocks. There are 2 registers that are available, one is the manual override flag and the other is the manual override value. Both registers are in EERAM so will retain the values after power fail or reboot. If the Manual register is set to 1 then the input or output will use the override value. Note that the inputs will show the manual value only when in override. The output will show both calculated and manual values but drive outputs from the override value. The IO log shows manually overridden points with an M when in Manual override mode. Figure 6 below shows how the override registers are processed.

## AI/DI Manual Processing

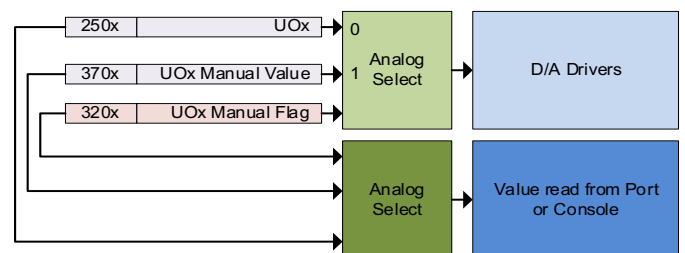
Note that the UI shows the real UI value from A/D converter or the over-riden value if override is on. The diagram shows that the AI/BI value is selected from the A/D converter if the manual override value is 0 and the Manual override value is selected if the manual override value is 1. This allows for full override control to test and force inputs to a fixed value. Values are EERAM based so remain after power fail or reboot.



## AO/DO Manual Processing

Input values are replaced in override mode. Output values are selected – the difference is that both the calculated output value and the manual value are accessible.

The diagram shows that the AO/BO value is selected from the output register if the manual override value is 0. The manual override value is selected if the manual override value is 1. This allows for full override control with view of both values to test and force outputs to a fixed value. Values are EERAM based so remain after power fail or reboot.



## Control Blocks

---

Control blocks are a licensed option. Refer to the Control Block User Manual. [PP-IO-COM-UserManual-ControlBlocks.pdf](#)

## Register Table

---

User register types are pre-set as per the table below. This can be changed in the respective user.txt file. If the defaults are kept then there is no need to set up the register types for control blocks unless the type is not float or uint16.

### USER Volatile - Vuser

---

User				Volatile			
Index	Description	Default Type	Reference	8UI	4UI4UO	4UI4DOT	4DOR
0-119	User defined register	float		0	0	0	0
120-199	User defined register	uint16		0	0	0	0

### User Flash - Fuser

---

User				Flash			
Index	Description	Default Type	Reference	8UI	4UI4UO	4UI4DOT	4DOR
200-229	User defined register	float		0	0	0	0
230-249	User defined register	uint16		0	0	0	0

### User EERAM - Euser

---

User				Eeram			
Index	Description	Default Type	Reference	8UI	4UI4UO	4UI4DOT	4DOR
250-319	User defined register	float		0	0	0	0
320-349	User defined register	uint16		0	0	0	0

System Volatile - Vsys

System				Volatile			
Index	Description	Default Type	Reference	8UI	4UI4UO	4UI4DOT	4DOR
350	Terminal 1 Value.	float for AI/AO/PWM. uint16 for DI/DO	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

		<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--

			<p>both the register and physical output.</p> <p>IO Values</p>				
352	Terminal 2 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

		<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--

			both the register and physical output. IO Values				
354	Terminal 3 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

			<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--	--

			<p>both the register and physical output.</p> <p>IO Values</p>				
356	Terminal 4 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

		<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--

			both the register and physical output. IO Values				
358	Terminal 5 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

		<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--

			both the register and physical output. IO Values				
360	Terminal 6 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

		<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--

			<p>both the register and physical output.</p> <p>IO Values</p>				
362	Terminal 7 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

			<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--	--

			both the register and physical output. IO Values				
364	Terminal 8 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

		<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--

			<p>both the register and physical output.</p> <p>IO Values</p>				
366	Terminal 9 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

		<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--

			both the register and physical output. IO Values				
368	Terminal 10 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

			<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--	--

			both the register and physical output. IO Values				
370	Terminal 11 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

		<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--

			both the register and physical output. IO Values				
372	Terminal 12 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

		<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--

			both the register and physical output. IO Values				
374	Terminal 13 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

			<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>			
--	--	--	--	--	--	--

			<p>both the register and physical output.</p> <p>IO Values</p>				
376	Terminal 14 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

			<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--	--

			<p>both the register and physical output.</p> <p>IO Values</p>				
378	Terminal 15 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

			<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--	--

			<p>both the register and physical output.</p> <p>IO Values</p>				
380	Terminal 16 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

			<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--	--

			both the register and physical output. IO Values				
382	Terminal 17 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

		<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--

			both the register and physical output. IO Values				
384	Terminal 18 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

		<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--

			both the register and physical output. IO Values				
386	Terminal 19 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

		<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--

			<p>both the register and physical output.</p> <p>IO Values</p>				
388	Terminal 20 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

		<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--

			<p>both the register and physical output.</p> <p>IO Values</p>				
390	Terminal 21 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

		<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--

			<p>both the register and physical output.</p> <p>IO Values</p>				
392	Terminal 22 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

		<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--

			<p>both the register and physical output.</p> <p>IO Values</p>				
394	Terminal 23 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

		<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--

			both the register and physical output. IO Values				
396	Terminal 24 Value.	Refer Terminal 1	<p><b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.</p> <p><b>Note1</b> – the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)</p> <p><b>Note2</b> – For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.</p> <p><b>Note3</b> - The relationship between the output register and the physical output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To</p>				

		<p>provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 – the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is &gt; 0. The output will then go off after the pulse time – because the output register is &gt; 0 the LED will be on and the output off. USB logs show</p>				
--	--	--	--	--	--	--

			both the register and physical output. IO Values				
398		uint16					
399	1 Second Pulse Train	uint16	<b>1 Second Pulse Train</b>				
400	Onboard temperature	float					
402	Health	unit16	<b>Health</b>				
403	License	unit16	<p><b>No</b> special requirements for wiring. Follow standard practice for screening of inputs. Note the I/O are not isolated therefore ground rail is susceptible to induced voltages if field wiring is not installed as per standard practice. Special care to be given to preventing induced voltages by using screen cable, terminating screen at one end only and not running cables next to higher voltages. RS485 is not optically isolated on the N models. Terminals are numbered from left to right and are allocated as per the legend which is model specific. For UI8 terminals follow the UI number. For UI4UO4 or UI4DOT4 the terminals are 1 to 4 for UO/DOT 1 to 4 the 5 to 8 for UI 1 to 4. Terminal numbers are used to identify the input when viewing console logs and at BACnet point creation (if BACnet is licensed).</p> <p>Licensing</p>				
404	Factory Test	unit16	<b>Factory Test</b>				
405	Port 1 Error Counter	unit16	<b>Port 1 (405) and Port2 (406)</b>				
406	Port 2 Error Counter	unit16					
407	Write to Non-Volatile Memory.	unit16	<b>Write Configuration to File.</b>				
408	Free Memory	uint32	Memory				
410	Spare	uint16					
..449	Spare						

## System Flash - Fsys

System				Flash			
Index	Description	Default Type	Reference	8UI	4UI4UO	4UI4DOT	4DOR
450	Port1 Baud Rate.	uint16	<b>Ports - UART – Communications Parameters</b>	9600	9600	9600	9600
451	Port1 Address	uint16		1	1	1	1
452	Port1 Turnaround Time offset.	int16		0	0	0	0
453	Port1 Timeout Time offset.	int16		0	0	0	0
454	Port1 BACnet instance number	uint16		1000	1000	1000	1000
455	Port1 BACnet Maximum Master.	uint16		127	127	127	127
456							
457							
458							
459							
460	Port2 Baud Rate	uint16	<b>Ports - UART – Communications Parameters</b>	9600	9600	9600	9600
461	Port2 Address	uint16		1	1	1	1
462	Port2 Turnaround Time offset.	int16		0	0	0	0
463	Port2 Timeout Time offset.	int16		0	0	0	0
464							
465	Boot Delay	uint16	<b>Boot Delay in seconds</b>	10	10	10	10
466	AI threshold	uint16	<b>AI Threshold and Response</b>	20	20	20	
467	AI response	uint16		8	8	8	
468	DI threshold.	uint16	<b>DI Threshold and Debounce</b>	1024	1024	1024	
469	DI debounce.	uint16		3	3	3	3
470	Terminal 1 Type Definition	uint16	<b>IO Definition</b>	13	40	30	30
471	Terminal 1 Associated Configuration.	uint16	<b>IO Configuration</b>	1	0	0	0
472	Terminal 1 Runtime/Pulse Divider or PWM time	uint16		1	1	0	0
473	Terminal 2 Type Definition	uint16		13	40	30	30
474	Terminal 2 Associated Configuration	uint16		1	0	0	0
475	Terminal 2 Runtime/Pulse Divider or PWM time	uint16		1	1	0	0
476	Terminal 3 Type Definition	uint16		13	40	30	30
477	Terminal 3 Associated Configuration	uint16		1	0	0	0
478	Terminal 3 Runtime/Pulse Divider or PWM time	uint16		1	1	0	0
479	Terminal 4 Type Definition	uint16		13	40	30	30
480	Terminal 4 Associated Configuration	uint16		1	0	0	0
481	Terminal 4 Runtime/Pulse Divider or PWM time	uint16		1	1	0	0
482	Terminal 5 Type Definition	uint16		13	13	13	
483	Terminal 5 Associated Configuration	uint16		1	1	1	
484	Terminal 5 Runtime/Pulse Divider or PWM time	uint16		1	1	1	
485	Terminal 6 Type Definition	uint16		13	13	13	
486	Terminal 6 Associated Configuration	uint16		1	1	1	
487	Terminal 6 Runtime/Pulse Divider or PWM time	uint16		1	1	1	
488	Terminal 7 Type Definition	uint16		13	13	13	
489	Terminal 7 Associated Configuration	uint16		1	1	1	
490	Terminal 7 Runtime/Pulse Divider or PWM time	uint16		1	1	1	
491	Terminal 8 Type Definition	uint16		13	13	13	
492	Terminal 8 Associated Configuration	uint16		1	1	1	
493	Terminal 8 Runtime/Pulse Divider or PWM time	uint16		1	1	1	

...549							
--------	--	--	--	--	--	--	--

## System EERAM – Esys

System				Eeram			
Index	Description	Default Type	Reference	8UI	4UI4UO	4UI4DOT	4DOR
550	Reboot Analysis	uint16	Not implemented				
551	Log Mask. Bit table for USB Log output.	uint16	<b>USB Logs – Log Verbosity</b>	14	14	14	14
552	Terminal 1 Associated Value.	float for AI/AO/PWM. uint32 for DI/DO	<b>PWM:</b> This defines the PWM period in 10mSec ticks. A minimum period of 1 second of 100 ticks is allowed. If a value of below 100 is used, then the controller will assume 100 as the count. A max count of 65 535 gives a PWM period of 655 seconds.  <b>Note1 –</b> the circuitry for an Analog Output on the UO models uses a capacitor filter. When using a UO in PWM mode use a period value of above 10 seconds. (1000)  <b>Note2 –</b> For DOR remember the relay is a mechanical device with a maximum number of operations. Keep the PWM period in the order of 2000 plus. Default PWM period is 100.  <b>Note3 –</b> The relationship between the output register and the physical	0	0	0	0
554	Terminal 1 Manual Over-ride flag.	uint16		0	0	0	0
555	Unused	uint16		0	0	0	0
556	Terminal 1 Override Value	float for AI/AO/PWM. uint16 for DI/DO	0	0	0	0	

		<p>output for PWM is defined by a calculation. By looking at the PWM output register, the output state (On/Off) cannot be definitively determined (unless the output is 0% for off or 100% for on). To provide visual indication, the output LED's will follow the output register and the USB Log will show both the register value (0-100%) and the physical output.</p> <p>PULSE: This defines the PULSE period in 10mSec ticks. A max count of 65 535 gives a PULSE period of 655 seconds.</p> <p>Note1 - the circuitry for an Analog Output on the UO models uses a capacitor filter. This means that the UO when used in PULSE mode is not a true switched signal.</p> <p>Note2 - The relationship between the output register and the physical output for Pulse is defined by a calculation. By</p>				
--	--	--	--	--	--	--

			looking at the Pulse output register, the output state (On/Off) cannot be definitively determined. The LED will be ON if the output register is > 0. The output will then go off after the pulse time – because the output register is > 0 the LED will be on and the output off. USB logs show both the register and physical output.				
			IO Values				
558	Terminal 2 Associated Value.	Refer Terminal1		0	0	0	0
560	Terminal 2 Manual Over-ride flag.	Refer Terminal1		0	0	0	0
555	Unused	Refer Terminal1		0	0	0	0
562	Terminal 2 Override Value	Refer Terminal1		0	0	0	0
564	Terminal 3 Associated Value.	Refer Terminal1		0	0	0	0
566	Terminal 3 Manual Over-ride flag.	Refer Terminal1		0	0	0	0
555	Unused	Refer Terminal1		0	0	0	0
568	Terminal 3 Override Value	Refer Terminal1		0	0	0	0
570	Terminal 4 Associated Value.	Refer Terminal1		0	0	0	0
572	Terminal 4 Manual Over-ride flag.	Refer Terminal1		0	0	0	0
555	Unused	Refer Terminal1		0	0	0	0
574	Terminal 4 Override Value	Refer Terminal1		0	0	0	0
576	Terminal 5 Associated Value.	Refer Terminal1		0	0	0	
578	Terminal 5 Manual Over-ride flag.	Refer Terminal1		0	0	0	
555	Unused	Refer Terminal1		0	0	0	
580	Terminal 5 Override Value	Refer Terminal1		0	0	0	
582	Terminal 6 Associated Value.	Refer Terminal1		0	0	0	
584	Terminal 6 Manual Over-ride flag.	Refer Terminal1		0	0	0	
555	Unused	Refer Terminal1		0	0	0	
586	Terminal 6 Override Value	Refer Terminal1		0	0	0	
588	Terminal 7 Associated Value.	Refer Terminal1		0	0	0	
590	Terminal 7 Manual Over-ride flag.	Refer Terminal1		0	0	0	
555	Unused	Refer Terminal1		0	0	0	
592	Terminal 7 Override Value	Refer Terminal1		0	0	0	
594	Terminal 8 Associated Value.	Refer Terminal1		0	0	0	
596	Terminal 8 Manual Over-ride flag.	Refer Terminal1		0	0	0	
597	Unused	Refer Terminal1		0	0	0	
598	Terminal 8 Override Value	Refer Terminal1		0	0	0	

..697							

## Special

Special				Eeram			
Index	Description	Default Type	Reference	8UI	4UI4UO	4UI4DOT	4DOR
698	Reboot – Write 0x55AA or 21930 decimal. Not written to NV	uint16	<b>SPECIAL SYSTEM Register</b>	0	0	0	0
699	Factory Reset – Write 0x1928 or 6440 decimal. Not written to NV	uint16		0	0	0	0

## Tutorials

---

Tutorials are available as part of the edgeUP-App user manual. Use edgeUP-App lessons to learn the concepts of PP-COM-IO.

edgeUP-App provides a unified interface to configure, read and write registers plus knowledge base. To get a copy of the software email [support@digitaltwin.digital](mailto:support@digitaltwin.digital). Only available for Windows or download the correct revision from [edgeUP-App](#).

The software is supplied with a user manual detailing the installation procedure. Once edgeUP-App is installed then continue with the lessons.

Hint: in Adobe PDF if you select a reference link – use Alt left arrow to return to the last view.