Notes from addressing ethernet IO Manually

Ethernet remote IO Modbus addressing

Inputs start at address 0 **(0x000)**

Outputs start at address 2048 **(0x0800)**

Data is shared in a series of 8bit Bytes starting from first module in the chain.

Analogue modules will use a single UINT16 register per analogue channel.

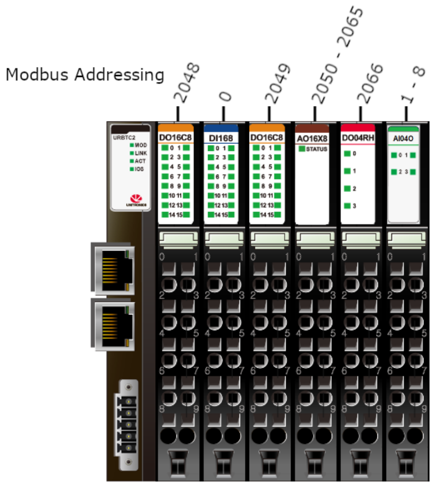
(So, if an analogue output module with 16 analogue channels is the first output module in the configuration, then this could be configured for address '**2048 – 2063**'. 1 address per analogue channel.)

Digital modules in banks of 16 will use a single UINT16 register to cover all 16 digital IO.

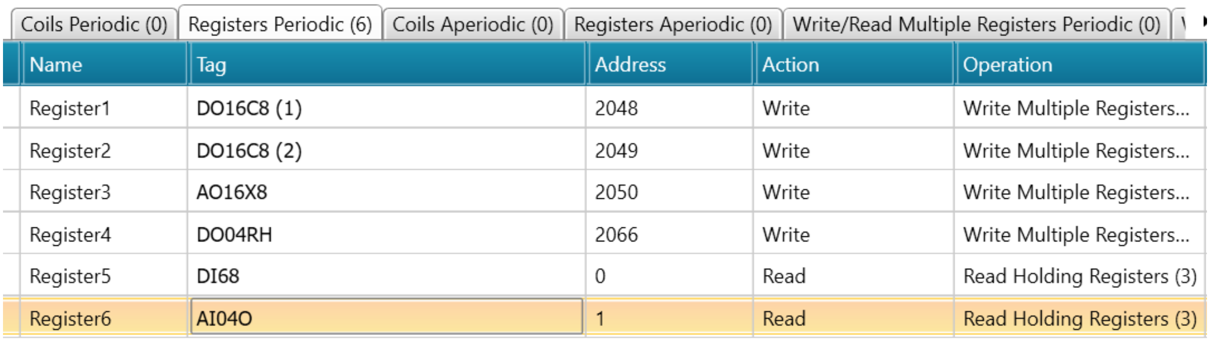
(So, if a digital output module with 16 digital channels is the next output module in the configuration after the above analogue output module, then this would be assigned to address 2064. 1 address for the whole module of 16 Digital outputs)

Digital modules in banks of 4 only use 8 bits so if this is assigned a 16 bit register the last 8 bits will spill over into the next module in the configuration. For this reason, 4 channel modules are advised to be placed at the end of the configuration to avoid spill over. Because of this you could use 1 address to control two 4 channel digital modules - 8 bits for each module controlled via UINT16 register for example.

The same is true for digital and analogue outputs except the addressing starts at 0.



Notes from addressing ethernet IO Manually



Name	Tag	Address	Action	Operation
Register1	DO16C8 (1)	2048	Write	Write Multiple Registers...
Register2	DO16C8 (2)	2049	Write	Write Multiple Registers...
Register3	AO16X8	2050	Write	Write Multiple Registers...
Register4	DO04RH	2066	Write	Write Multiple Registers...
Register5	DI68	0	Read	Read Holding Registers (3)
Register6	AI04O	1	Read	Read Holding Registers (3)

Addressing using buffer variables-

1 address can be used as starting point for reading and another for writing.

For this you can use a buffer with array length to match amount of data to be transmitted – so for example –

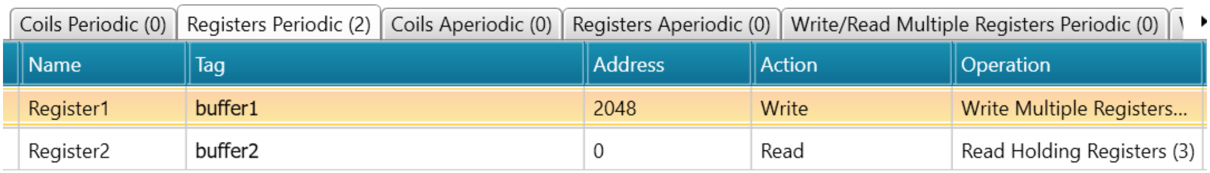
DO16C8 will use a buffer with array length of 2.

DO04RH will use a buffer with array length of 1.

AO04W will use buffer with array length of 2 per channel (8).

If all three of these modules are configured in this order, then you would need a buffer array of 11 starting at address 2048. Array 1 & 2 controlling DO16C8 outputs, array 3 controlling DO04RH outputs. Array 4 & 5 will be for the first Analogue output of AO04W, 6 & 7 the second and so on.

The same is true for the inputs but starting at address 0 rather than 2048

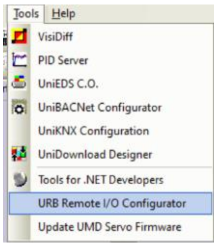


Name	Tag	Address	Action	Operation
Register1	buffer1	2048	Write	Write Multiple Registers...
Register2	buffer2	0	Read	Read Holding Registers (3)

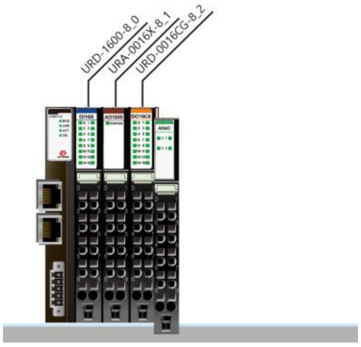
Remote IO with Vision PLC

Open remote IO configurator

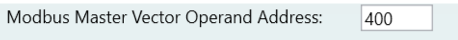
VisiLogic -> Tools -> URB Remote IO Configurator



Add IO modules used in your configuration



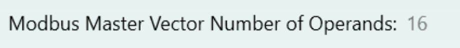
Go to '**Addressing**' tab and select a '**Modbus Master Vector Operand Type'** (XB, XI, MB, MI) starting '**Modbus Master Vector Operand Address**' which is the start point of the vector to write to in the PLC before sending down or receiving data from the URB adapter.



Modbus Master Vector Operand Address: 400

*Further configuration can be achieved quickly for inputs & outputs using the '**Address Configurator**' from the ribbon at the top of the screen.*

Make sure the operands for the length of your vector of 'Modbus Master Vector Number of Operands' are available addresses in your PLC program.



Modbus Master Vector Number of Operands: 16

Notes from addressing ethernet IO Manually

For example -

Start address = MI400

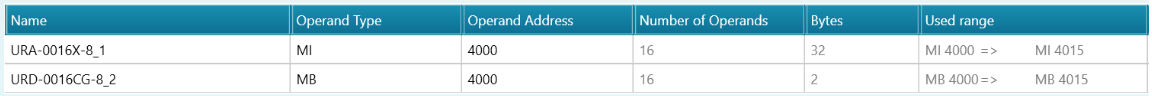
Number of operands = 16

Operands that need to be available in PLC = MI400 – MI415

The above addressing can be thought of as a buffer register for sending the data between the remote IO & PLC via P.H.R or R.H.R commands.

For Digital Output channels - the operands which you change in the PLC to trigger the corresponding output is defined in the table (below) by selecting '**Operand Type**' (XB or MB) and selecting the start '**Operand Address**' of the vector to be used in the PLC. '**Used range**' column will then be populated with which operands are used for this module. (These operands will be labelled in your program when the STRUCT is imported to VisiLogic - these are the values which you change in the PLC to control the outputs)

For Analogue Output channels - the operands which you change in the PLC to change the corresponding output is defined in the table (below) by selecting '**Operand Type**' (XI or MI) and selecting the start '**Operand Address**' of the vector to be used in the PLC. '**Used range**' column will then be populated with which operands are used for this module. (These operands will be labelled in your program when the STRUCT is imported to VisiLogic - these are the values which you change in the PLC to control the outputs)



Name	Operand Type	Operand Address	Number of Operands	Bytes	Used range	
URA-0016X-8_1	MI	4000	16	32	MI 4000 =>	MI 4015
URD-0016CG-8_2	MB	4000	16	2	MB 4000=>	MB 4015

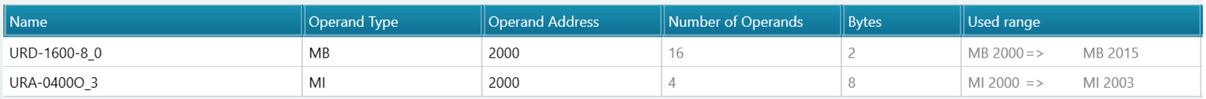
i.e. To control digital output 2 of the ***URD-0016CG-8_2*** *module you will switch MB4001 in the PLC. Output 5 = MB4004 etc.*

To control analogue output 3 of the ***URA-0016X-8_1*** *module you will write the value to MI4002. Output 15 = MI4014 etc.*

Notes from addressing ethernet IO Manually

For Digital Input channels - the operands which are switched in the PLC when the corresponding input gets triggered is defined in the table (below) by selecting '**Operand Type**' (XB or MB) and selecting the start '**Operand Address**' of the vector to be used in the PLC. '**Used range**' column will then be populated with which operands are used for this module. (These operands will be labelled in your program when the STRUCT is imported to VisiLogic - these are the values which will change in the PLC when the input is triggered)

For Analogue Input channels - the operands which are switched in the PLC when the corresponding input gets changed is defined in the table (below) by selecting '**Operand Type**' (XI or MI) and selecting the start '**Operand Address**' of the vector to be used in the PLC. '**Used range**' column will then be populated with which operands are used for this module. (These operands will be labelled in your program when the STRUCT is imported to VisiLogic - these are the values which will change in the PLC when the input is changed)



Name	Operand Type	Operand Address	Number of Operands	Bytes	Used range	
URD-1600-8_0	MB	2000	16	2	MB 2000=>	MB 2015
URA-0400O_3	MI	2000	4	8	MI 2000 =>	MI 2003

*i.e. To read digital input 2 of the **URD-1600-8_0** module you will monitor MB2001 in the PLC. Input 5 = MB2004 etc.*

*To read analogue input 3 of the **URA-0400O_3** module you will monitor the value in MI2002. Input 4 = MI2003 etc.*

Once the configuration has been configured within the URB Configurator this should be downloaded to the adapter via ethernet using the pre-determined IP address of the adapter (192.168.100.100 as default) or the address as set in the BootP server. (*See '**Setting the adapter's IP**' section of help file in URB Configurator*)

Select '**Export Structs VisiLogic**' and choose save location folder.

Setup Modbus network in VisiLogic program as per accompanying demo application *(Default settings below)*

Network ID: 255

Remote Port: 502

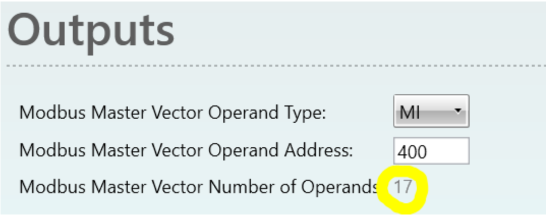
URB IP Address: 192.168.100.100

Once Modbus network is setup as shown in the demo application you can import the .ulri '**Structs**' files exported from the Remote IO configurator into the appropriate STRUCT function block. *[Inputs = STRUCT (extract), Outputs = STRUCT (Collect)]*

From here you can simply change the state of the outputs by changing the previously assigned output operands or retrieve the state of the input by monitoring the previously assigned input operands.

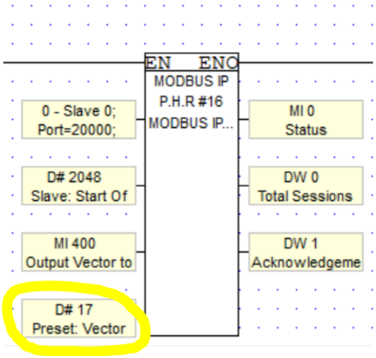
The address for the output vector starts from #2048 and the address for the input vector starts from #0.

The '**Preset: Vector Length**' in the P.H.R function block must be equal to the '**Modbus Master Vector Number of Operands**' value in the '**Outputs**' section or comms will not work (Modbus Status 6 or 7)



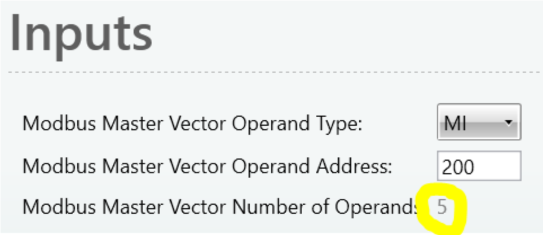
Outputs

Modbus Master Vector Operand Type: MI
Modbus Master Vector Operand Address: 400
Modbus Master Vector Number of Operands 17



Notes from addressing ethernet IO Manually

The '**Read: Vector Length**' in the R.H.R function block must be equal to the '**Modbus Master Vector Number of Operands**' value in the '**Inputs**' section or comms will not work (Modbus Status 6 or 7)

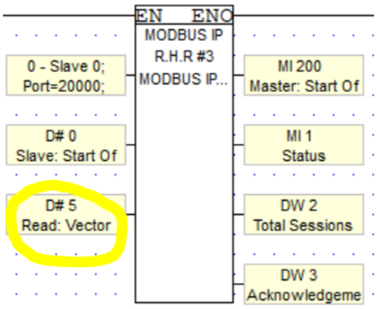


Inputs

Modbus Master Vector Operand Type: MI

Modbus Master Vector Operand Address: 200

Modbus Master Vector Number of Operands: 5



Note- currently after importing a STRUCT into your project you will be unable to change the value of the linked operand using online test in VisiLogic. The values can still be changed from the HMI or ladder.

Tom Johnson

January 2023