Technical Note

QuickStart Guide

Configuring a RedLion HMI to communicate with an MPiec Controller via MODBUS/TCP using Crimson 2

Applicable Product: MPiec, RedLion G310 HMI

Table of Contents

1. Application Overview:

This document is intended to guide a user through the steps of configuring a RedLion HMI as a MODBUS master communicating with the MPiec controller as a slave. This application note was created with a G310 series HMI with Crimson 2.0 software. The MPiec controller has firmware version 1.0.0.1 and MotionWorks IEC Express software version 1.0.0.1.

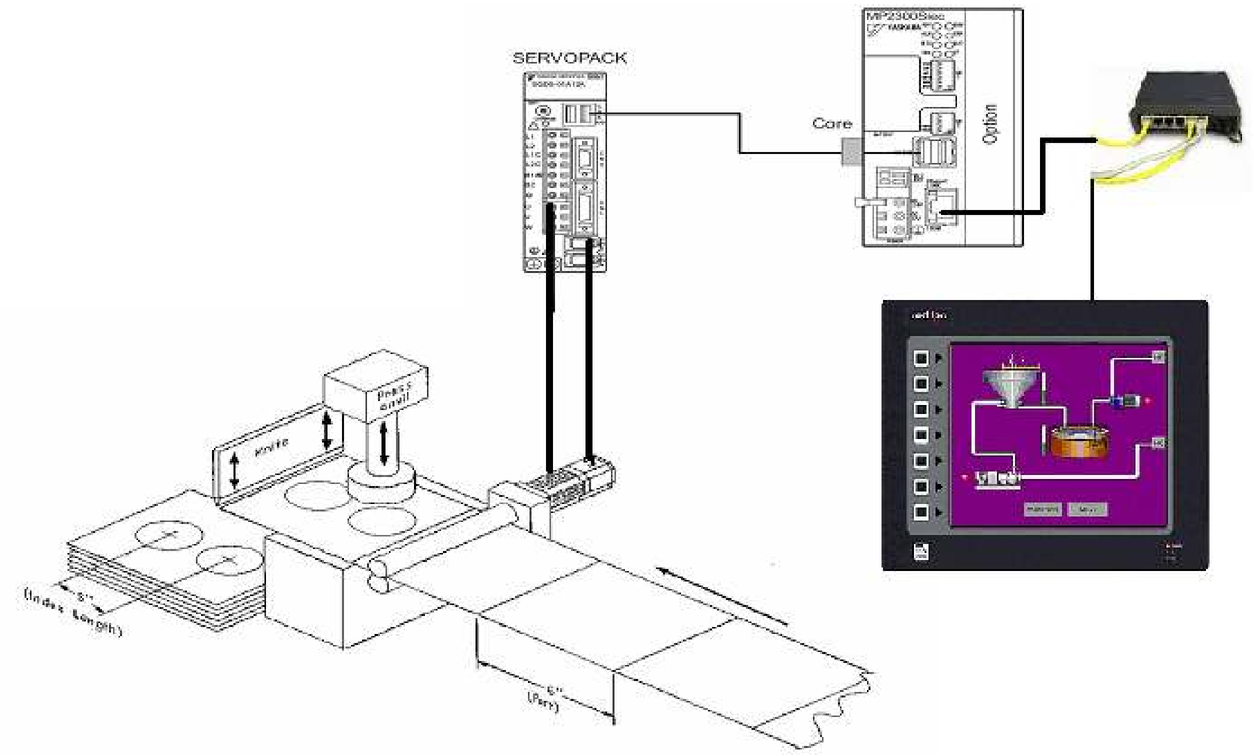


Figure1: Overview of test set up

Products Used:

Controller	Yaskawa MPiec controller
Hardware	Sigma series servopack and motor, RedLion G310 HMI, Ethernet switch
Software	MotionWorksIEC, Crimson 2.0

2. Configuration Procedure

I. Set the IP addresses of the devices (HMI, controller, PC with programs). In the test set up, the IP addresses were: Controller: 192.168.1.1, HMI: 192.168.1.2, PC: 192.168.1.3
The default IP address on the MPiec controller is 192.168.1.1 unless changed by a user.

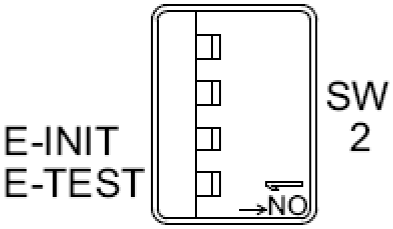
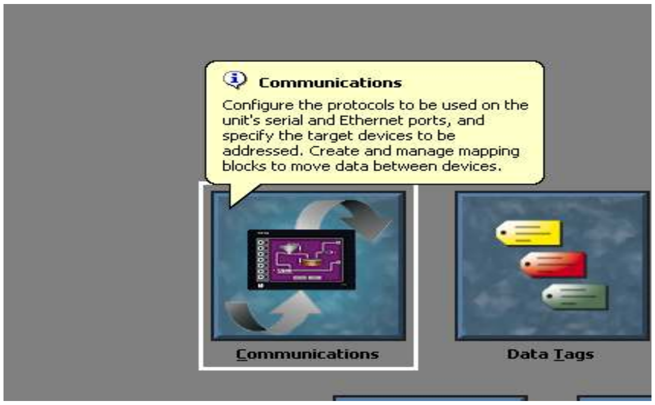


Figure 2: Set IP address of controller

II. To set the IP address of the RedLion G310: On the opening page of Crimson 2.0, double click on the communications tab.

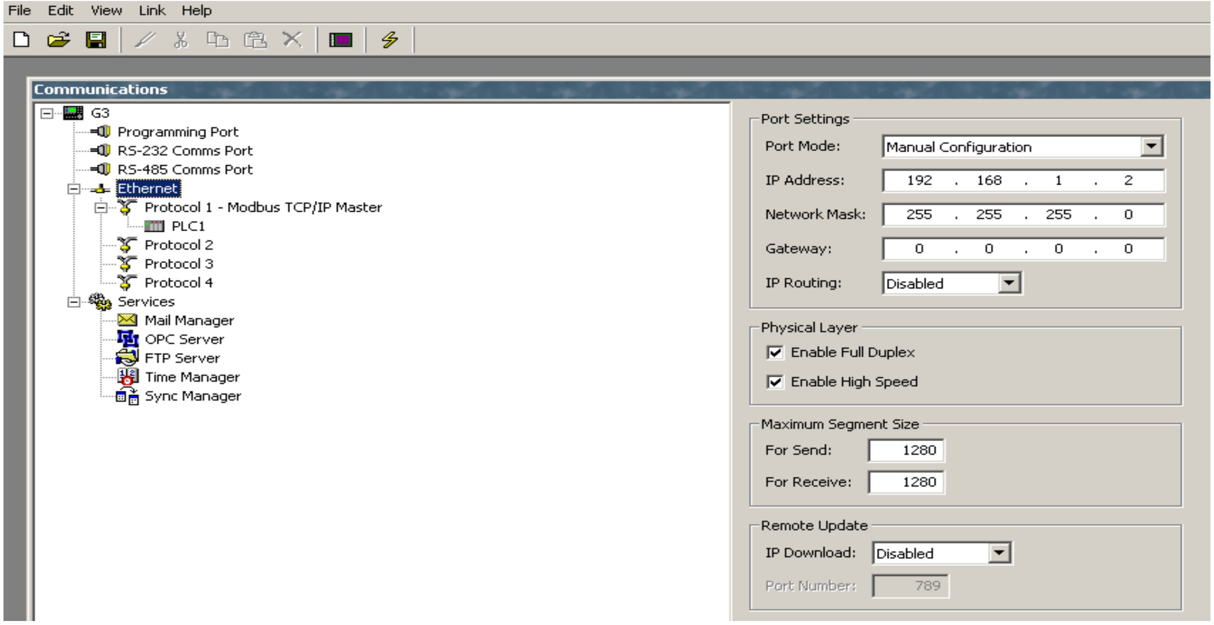


Communications

Configure the protocols to be used on the unit's serial and Ethernet ports, and specify the target devices to be addressed. Create and manage mapping blocks to move data between devices.

Fugure3: To set the HMI IP address

Double click on Ethernet and set the IP address of the HMI as shown in Figure 4.



- G3
  - Programming Port
  - RS-232 Comms Port
  - RS-485 Comms Port
  - Ethernet
    - Protocol 1 - Modbus TCP/IP Master
      - PLC1
    - Protocol 2
    - Protocol 3
    - Protocol 4
  - Services
    - Mail Manager
    - OPC Server
    - FTP Server
    - Time Manager
    - Sync Manager

Port Settings

Port Mode: Manual Configuration

IP Address: 192 . 168 . 1 . 2

Network Mask: 255 . 255 . 255 . 0

Gateway: 0 . 0 . 0 . 0

IP Routing: Disabled

Physical Layer

☑ Enable Full Duplex

☑ Enable High Speed

Maximum Segment Size

For Send: 1280

For Receive: 1280

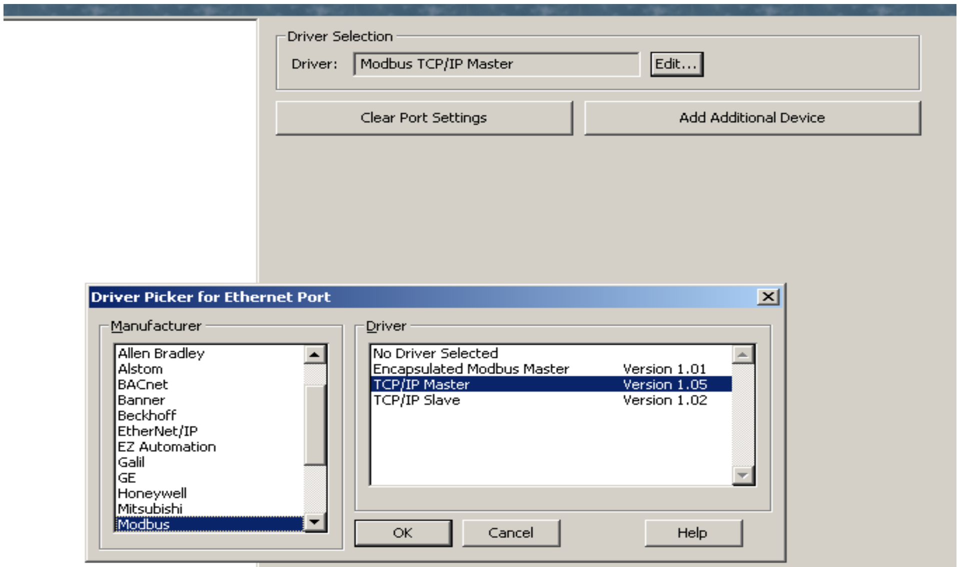
Remote Update

IP Download: Disabled

Port Number: 789

Figure 4: HMI IP address

Click on Protocol 1. Edit the driver selection to select MODBUS as manufacturer and TCP/IP master as driver. No special Yaskawa driver is necessary.



Driver Selection

Driver: Modbus TCP/IP Master

Driver Picker for Ethernet Port

Manufacturer

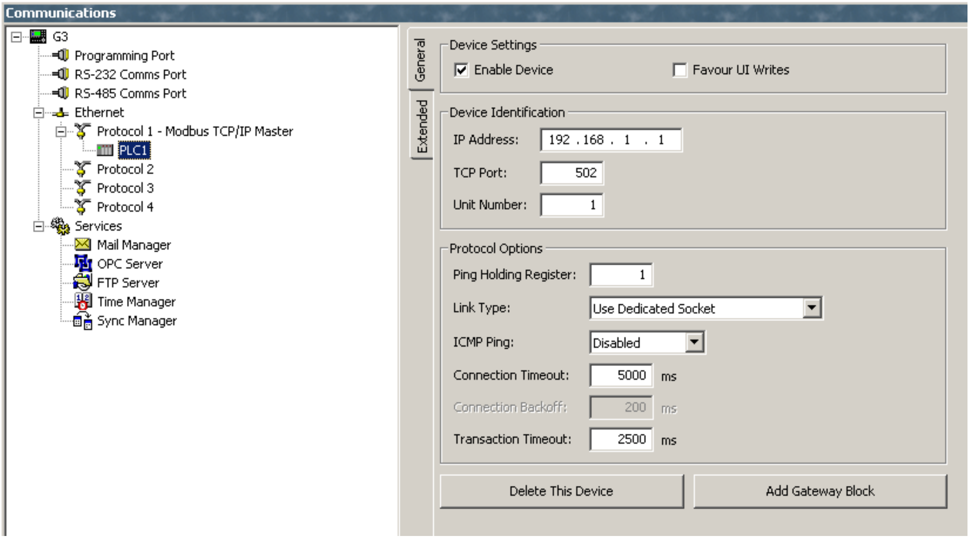
- Allen Bradley
- Alstom
- BACnet
- Banner
- Beckhoff
- EtherNet/IP
- EZ Automation
- Galil
- GE
- Honeywell
- Mitsubishi
- Modbus

Driver

No Driver Selected	
Encapsulated Modbus Master	Version 1.01
TCP/IP Master	Version 1.05
TCP/IP Slave	Version 1.02

Figure 5: Communication Protocol

This will create the slave configuration under protocol 1. When double clicking on the slave device created, the IP address of the slave in this communication process can be specified. (Figure 6).



Communications

- G3
  - Programming Port
  - RS-232 Comms Port
  - RS-485 Comms Port
  - Ethernet
    - Protocol 1 - Modbus TCP/IP Master
      - PLC1
    - Protocol 2
    - Protocol 3
    - Protocol 4
  - Services
    - Mail Manager
    - OPC Server
    - FTP Server
    - Time Manager
    - Sync Manager

Device Settings

☑ Enable Device ☐ Favour UI Writes

Device Identification

IP Address: 192 . 168 . 1 . 1

TCP Port: 502

Unit Number: 1

Protocol Options

Ping Holding Register: 1

Link Type: Use Dedicated Socket

ICMP Ping: Disabled

Connection Timeout: 5000 ms

Connection Backoff: 200 ms

Transaction Timeout: 2500 ms

Figure 6: IP address of slave specified

III. Using an HMI (Master) to communicate to the controller using Modbus TCP

Figure 7 shows the Modbus memory map, and how it relates to the Global Variables in MotionWorks IEC. Note that function codes 01 and 03 return data that was sent to the controller from the master and do not reflect data from the Global Variables in the application program.

The Modbus data memory is copied to the Global Variables at the task update rate.

Modbus coil 0 equates to the Global Variable at %IX24560. 128 coils are available.

Modbus register 40000 equates to the Global Variable at %IW28672. 1024 registers are available.

Modbus input 10000 equates to the Global Variable at %QX24560. 128 inputs are available.

Modbus register 30000 equates to the Global Variable at %QW28672. 1024 registers are available.

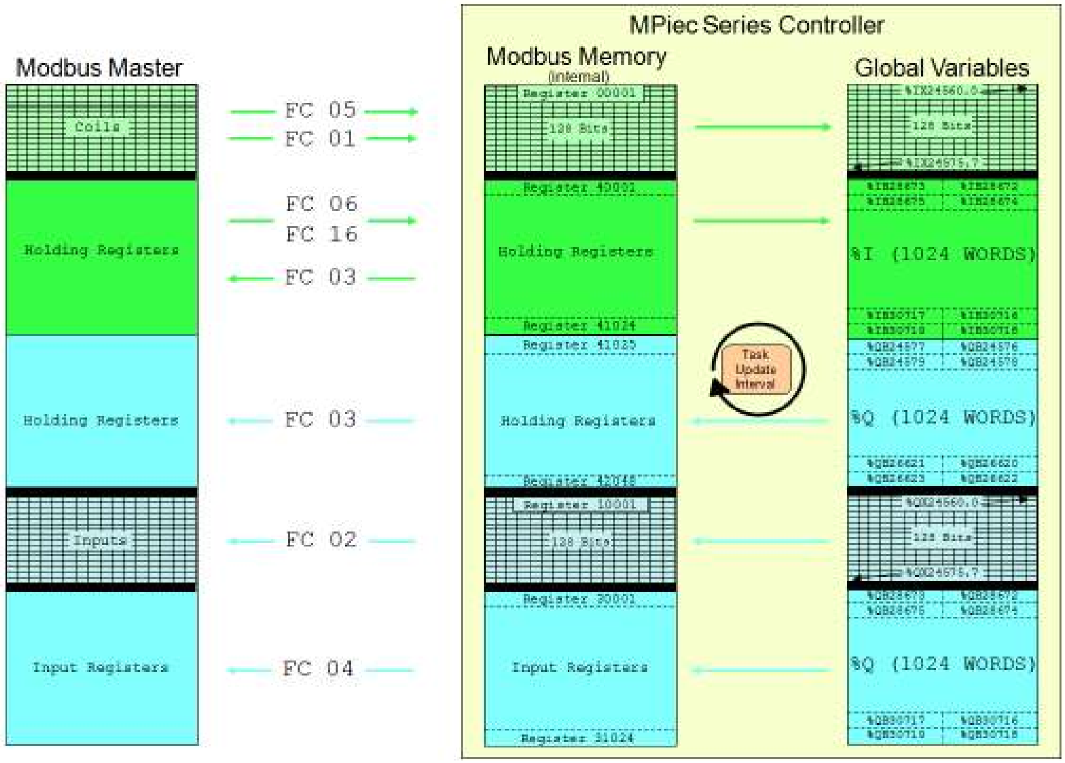
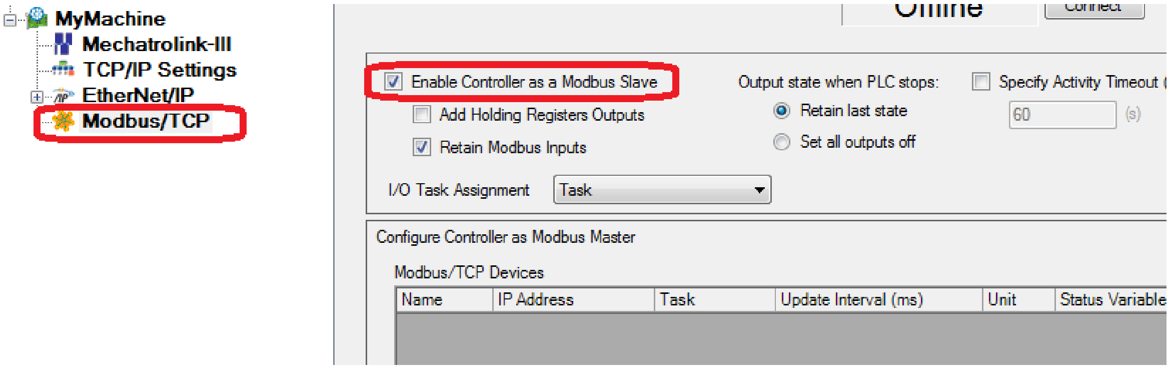


Figure 7a: MODBUS mapping

As shown in Figure 7a below, the input and output memory is in a different location even though they have the same numerical addressing.

IV. Launch the Hardware Configuration in MotionWorks IEC. Connect to the controller through the hardware configuration. Select the MODBUS TCP option and enable the controller as a MODBUS slave. Save this configuration and reboot the controller. These steps will activate the MODBUS communication driver on the MPiec controller.



☑ Enable Controller as a Modbus Slave

□ Add Holding Registers Outputs

☑ Retain Modbus Inputs

I/O Task Assignment: Task

Output state when PLC stops:

● Retain last state

○ Set all outputs off

□ Specify Activity Timeout

60 (s)

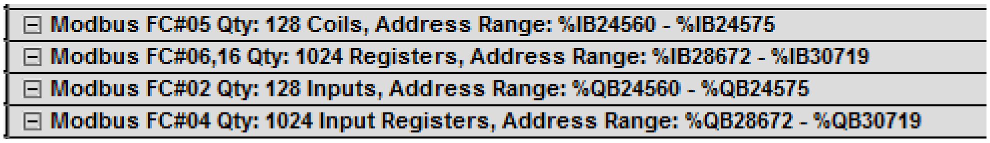
Configure Controller as Modbus Master

Modbus/TCP Devices

Name	IP Address	Task	Update Interval (ms)	Unit	Status Variable

Figure 7b: Enabling the MPiec controller as a MODBUS TCP slave (server)

V. The global variables worksheet in MotionWorks IEC will now have MODBUS TCP groups which will have to be populated by the user. These memory areas are generic: no variables are automatically created, the user can decide on the arrangement and type of data to populate within the communications memory region



⊟ Modbus FC#05 Qty: 128 Coils, Address Range: %IB24560 - %IB24575
⊟ Modbus FC#06,16 Qty: 1024 Registers, Address Range: %IB28672 - %IB30719
⊟ Modbus FC#02 Qty: 128 Inputs, Address Range: %QB24560 - %QB24575
⊟ Modbus FC#04 Qty: 1024 Input Registers, Address Range: %QB28672 - %QB30719

Figure 8: Variable groups created in template

A word coming from the HMI will be mapped from register 40001 in the RedLion to %IB28672. The user should verify the data type used for data transfer. Bits written from the HMI to the controller are mapped from 00001 to %IX24560.0. Outputs from the controller are mapped from %QB28672 ->300001 and %QX24560.0 -> 10001

3. HMI variables

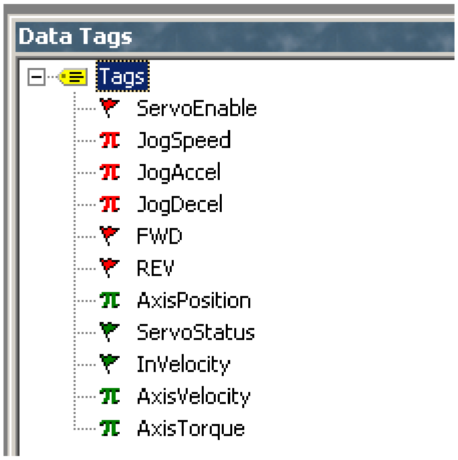
Create output bits/words on the HMI.

1. Servo Enable (bit)
2. Forward direction
3. Reverse direction
4. Jog Speed
5. Jog acceleration
6. Jog deceleration

Create input bits/words on the HMI.

1. ServoStatus
2. InVelocity
3. Actual Position
4. Actual Velocity
5. Actual Torque

The variable tags have been defined in the data tags tab in the RedLion project.

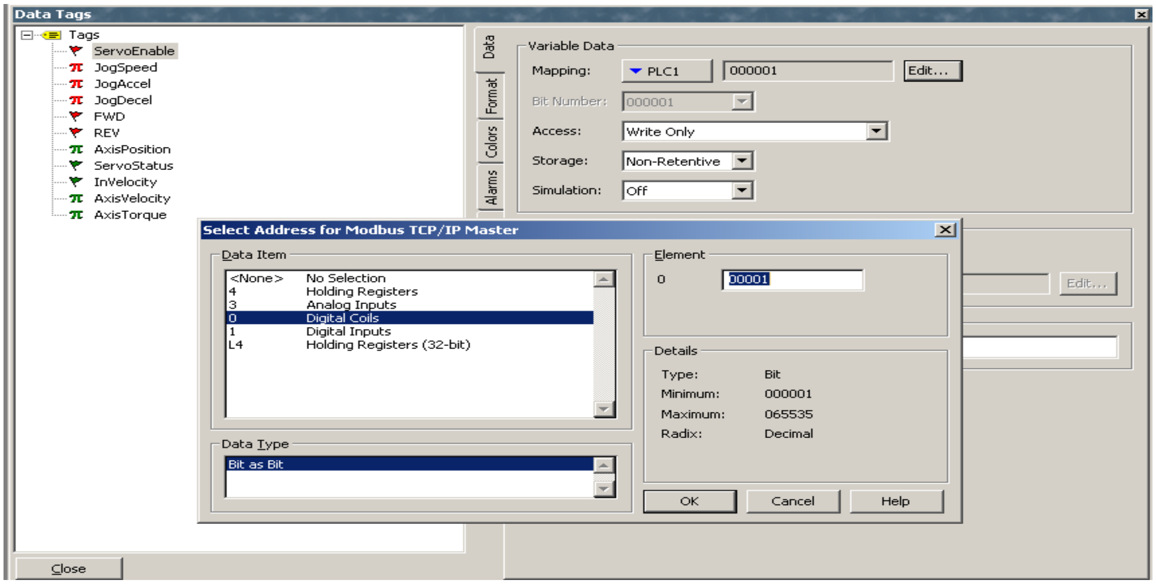


Data Tags

- Tags
  - ServoEnable
  - JogSpeed
  - JogAccel
  - JogDecel
  - FWD
  - REV
  - AxisPosition
  - ServoStatus
  - InVelocity
  - AxisVelocity
  - AxisTorque

Figure 9: Data tags

When creating a new tag, the user can specify the data type of the tag. The user can also map the tag and specify what register location that tag should be associated with. For example, the tag ‘ServoEnable’ is mapped to a location in PLC1, which is the slave IEC controller. On clicking the edit button, the register and data type can be seen as shown in Figure 10. It can be seen that a bit was chosen as the data type. The register to which that tag is mapped is 00001 and it is a coil. This bit will be mapped to the first bit mapped under Function Code (FC) 5. The access to this tag is set as ‘Write Only’ as this variable is used for writing to the controller. FWD and REV are similar bit tags. The ‘Jog speed’ tag mapping is shown in Figure 11. A ‘word swap’ transformation is carried out for words. The register used for the word is 40001. Figure 12 illustrates the mapping of a tag that is a read access variable that reads the servo status from the controller to display at the HMI panel. Figure 13 illustrates a controller output real variable displayed at the HMI.



Data Tags

Tags
- ServoEnable
- JogSpeed
- JogAccel
- JogDecel
- FWD
- REV
- AxisPosition
- ServoStatus
- InVelocity
- AxisVelocity
- AxisTorque

Variable Data

Mapping: PLC1 000001 Edit...

Bit Number: 000001

Access: Write Only

Storage: Non-Retentive

Simulation: Off

Select Address for Modbus TCP/IP Master

Data Item

<None>	No Selection
4	Holding Registers
3	Analog Inputs
0	Digital Coils
1	Digital Inputs
L4	Holding Registers (32-bit)

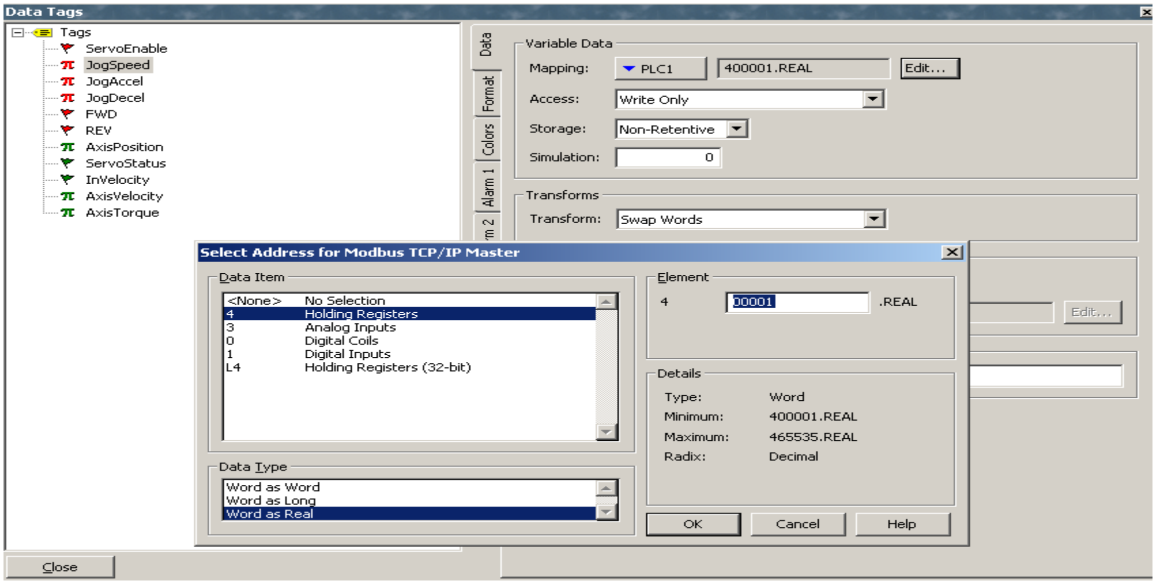
Data Type: Bit as Bit

Element: 0 00001

Details

Type:	Bit
Minimum:	000001
Maximum:	065535
Radix:	Decimal

Figure 10: Mapping bit (FC: 5)



Data Tags

Tags
- ServoEnable
- JogSpeed
- JogAccel
- JogDecel
- FWD
- REV
- AxisPosition
- ServoStatus
- InVelocity
- AxisVelocity
- AxisTorque

Variable Data

Mapping: PLC1 400001.REAL

Access: Write Only

Storage: Non-Retentive

Simulation: 0

Transforms

Transform: Swap Words

Select Address for Modbus TCP/IP Master

Data Item

<None>	No Selection
4	Holding Registers
3	Analog Inputs
0	Digital Coils
1	Digital Inputs
L4	Holding Registers (32-bit)

Element

4 00001 .REAL

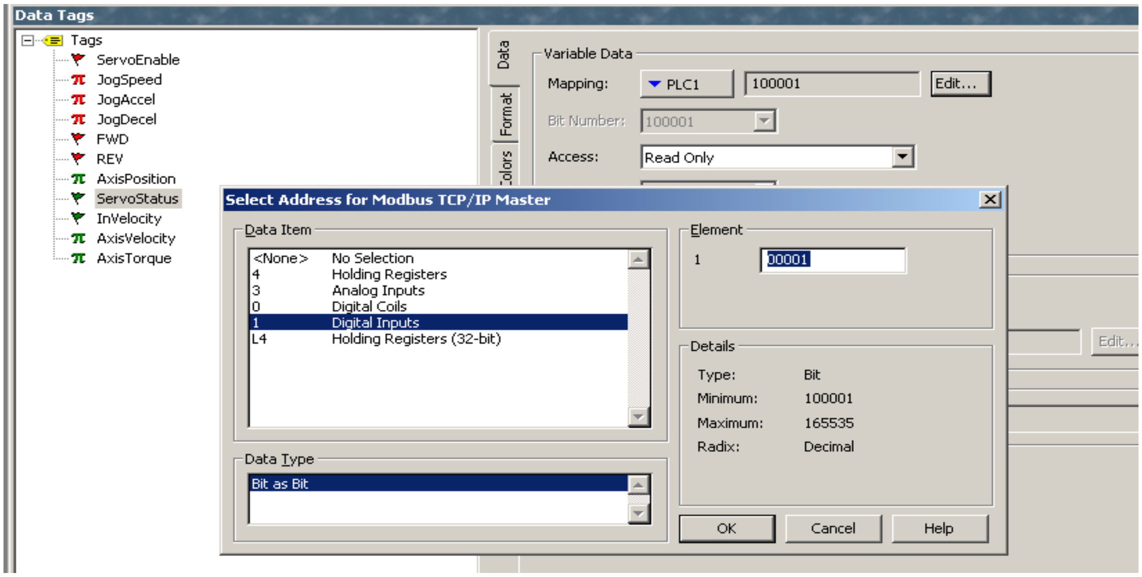
Details

Type:	Word
Minimum:	400001.REAL
Maximum:	465535.REAL
Radix:	Decimal

Data Type

- Word as Word
- Word as Long
- Word as Real

Figure 11: Mapping word (FC: 16)



Data Tags

Tags
- ServoEnable
- JogSpeed
- JogAccel
- JogDecel
- FWD
- REV
- AxisPosition
- ServoStatus
- InVelocity
- AxisVelocity
- AxisTorque

Variable Data

Mapping: PLC1 100001

Bit Number: 100001

Access: Read Only

Select Address for Modbus TCP/IP Master

Data Item

<None>	No Selection
4	Holding Registers
3	Analog Inputs
0	Digital Coils
1	Digital Inputs
L4	Holding Registers (32-bit)

Element

1 00001

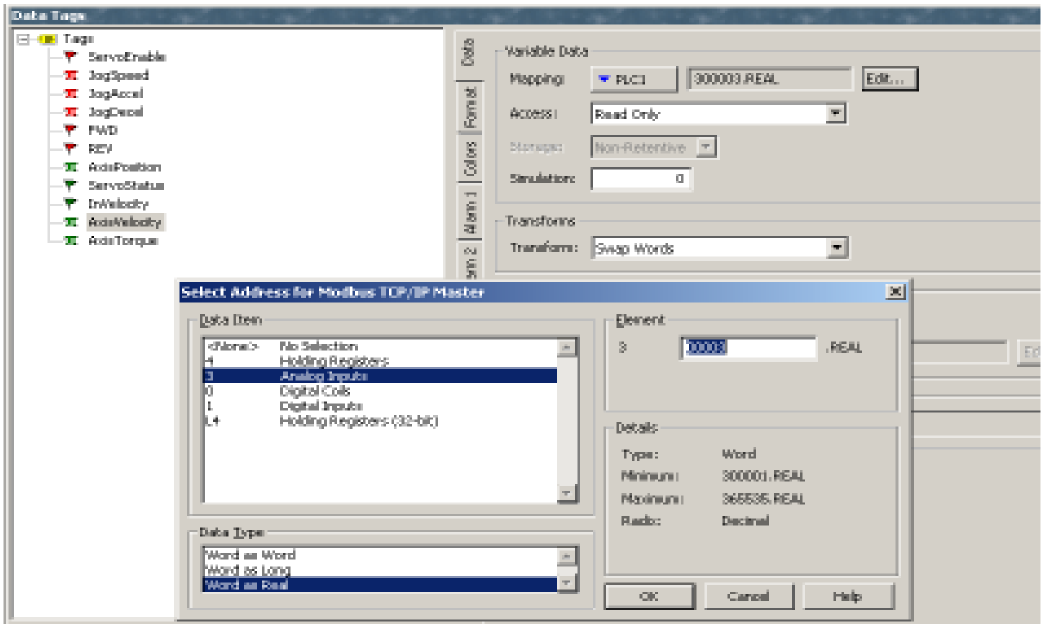
Details

Type:	Bit
Minimum:	100001
Maximum:	165535
Radix:	Decimal

Data Type

- Bit as Bit

Figure 12: Bit Mapping (FC: 2)



Data Tags

Tags
- ServoEnable
- JogSpeed
- JogAccel
- JogDecel
- FWD
- REV
- AxisPosition
- ServoStatus
- InVelocity
- AxisVelocity
- AxisTorque

Variable Data
Mapping: PLC1 300003.REAL
Access: Read Only
Storage: Non-Retentive
Simulation: 0

Transforms
Transform: Swap Words

Select Address for Modbus TCP/IP Master

Data Item

<None>	No Selection
4	Holding Registers
3	Analog Inputs
0	Digital Coils
1	Digital Inputs
L4	Holding Registers (32-bit)

Element: 3 00003 .REAL

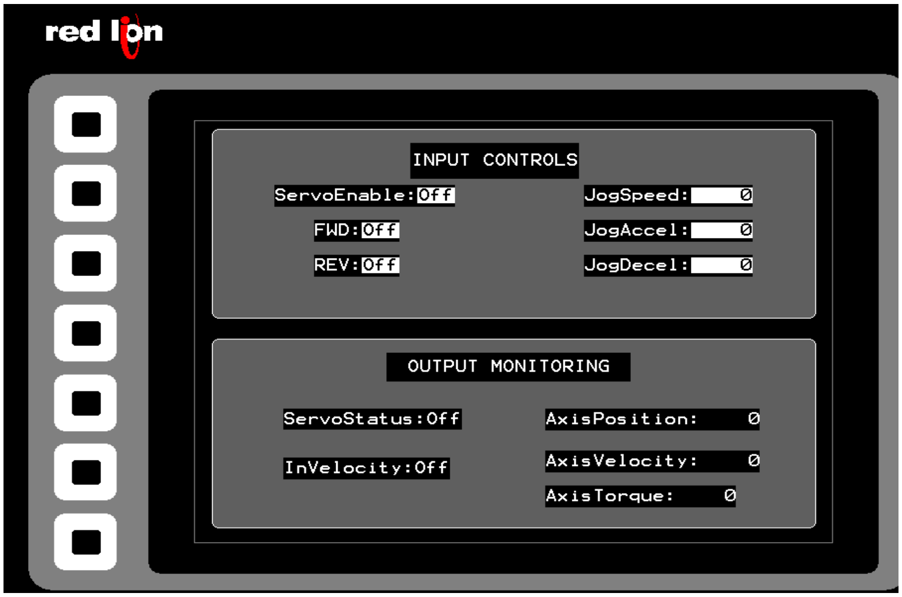
Details

Type:	Word
Minimum:	300001.REAL
Maximum:	365535.REAL
Radix:	Decimal

Data Type
- Word as Word
- Word as Long
- Word as Real

Figure 13: Word mapping (FC: 4)

The user interface on the HMI screen is as shown in Figure 14.



INPUT CONTROLS

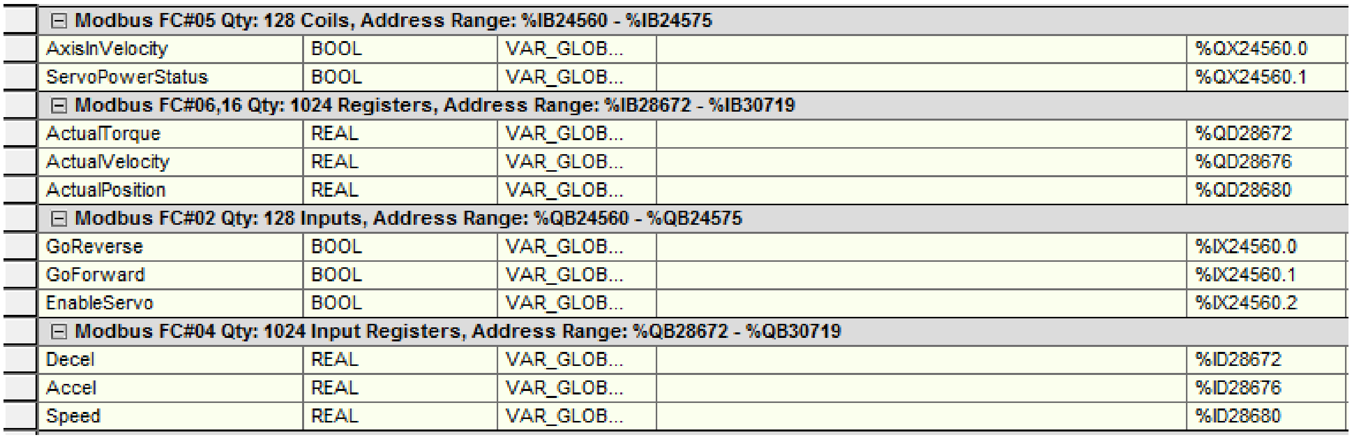
ServoEnable: Off
FWD: Off
REV: Off
JogSpeed: 0
JogAccel: 0
JogDecel: 0

OUTPUT MONITORING

ServoStatus: Off
InVelocity: Off
AxisPosition: 0
AxisVelocity: 0
AxisTorque: 0

Figure 14: HMI user interface screen shot

4. MotionWorks IEC variable declaration



⊟ Modbus FC#05 Qty: 128 Coils, Address Range: %IB24560 - %IB24575				
AxisInVelocity	BOOL	VAR_GLOB...		%QX24560.0
ServoPowerStatus	BOOL	VAR_GLOB...		%QX24560.1
⊟ Modbus FC#06,16 Qty: 1024 Registers, Address Range: %IB28672 - %IB30719				
ActualTorque	REAL	VAR_GLOB...		%QD28672
ActualVelocity	REAL	VAR_GLOB...		%QD28676
ActualPosition	REAL	VAR_GLOB...		%QD28680
⊟ Modbus FC#02 Qty: 128 Inputs, Address Range: %QB24560 - %QB24575				
GoReverse	BOOL	VAR_GLOB...		%IX24560.0
GoForward	BOOL	VAR_GLOB...		%IX24560.1
EnableServo	BOOL	VAR_GLOB...		%IX24560.2
⊟ Modbus FC#04 Qty: 1024 Input Registers, Address Range: %QB28672 - %QB30719				
Decel	REAL	VAR_GLOB...		%ID28672
Accel	REAL	VAR_GLOB...		%ID28676
Speed	REAL	VAR_GLOB...		%ID28680

Figure 14: Variables in the MPiec controller (Slave)

The global variable list with the MODBUS variables being used is shown in detail in Figure 14. The first two groups corresponding to function codes 05 and 16 are the input variables written from the HMI to the controller. The groups under function codes 02 and 04 are used to transfer variables from the controller to the HMI. They are denoted by %QX and %QB. Table 1 shows the variable mapping in this sample project.

Table 1: variable mapping

FC#05: From master to slave (coils)				
EnableServo	%IX24560.0	←	00001	ServoEnable
GoForward	%IX24560.1	←	00002	FWD
GoReverse	%IX24560.2	←	00003	REV
FC#06: From master to slave (registers)				
Speed	%ID28672	←	400001	JogSpeed
Accel	%ID28676	←	400003	JogAccel
Decel	%ID28680	←	400005	JogDecel
FC#02: From slave to master (Inputs)				
ServoPowerStatus	%QX24560.0	→	100001	ServoStatus
AxisInVelocity	%QX24560.1	→	100002	InVelocity

FC#04: From slave to master (Registers)				
ActualPosition	%QD28672	→	300001	AxisPosition
ActualVelocity	%QD28676	→	300003	AxisVelocity
ActualTorque	%QD28680	→	300005	AxisTorque

5. Execution

Communication between the Red Lion HMI and the MPiec controller should commence as soon as the two projects start running on the two devices.