

Application Note

Using Mitsubishi FX brand PLCs with Qlarity-based terminals

Introduction

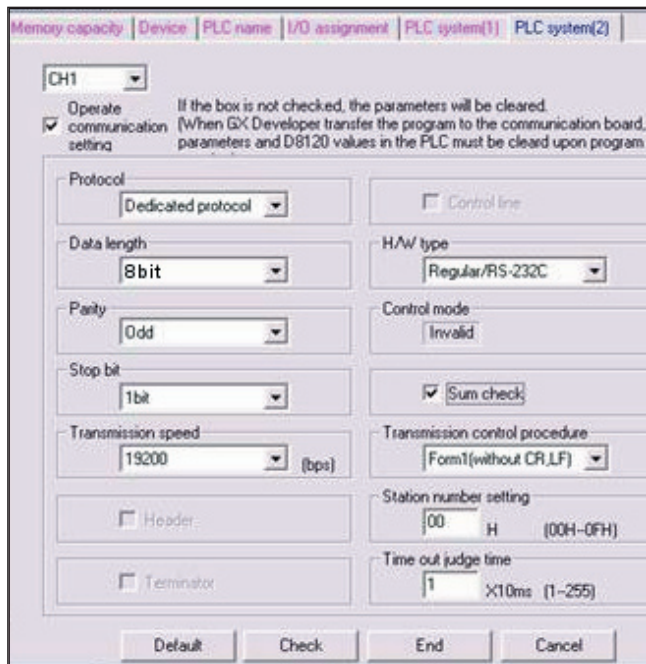
A Qlarity™ library of objects was created to communicate with Mitsubishi FX-brand PLC controllers with serial communication modules. Modules that use the FX communication protocol include: FX3U-232-BD, FX2N-232-BD, FX1N-232-BD. The MitsubishiFX library uses the FX communication protocol commands: Batch Read of Bit Device (BR), Batch Read of Word Device (WR), Batch Write of Bit Device (BW), and Batch Write of Word Device (WR).

Configuring the Controller

Your controller must be configured using Mitsubishi's GX Developer before it can communicate with the serial module using the FX communication protocol.

Here are the steps to do this in GX Developer:

1. Open an existing project or start a new one in GX Developer.



2. Go to the parameter settings, in the project data list.

3. Select the “PLC system(2) tab and set the communication settings similar to those in the screen capture above.

NOTE: Make sure the Qlarity terminal or Qlarity Foundry Simulation serial port settings such as data length, parity, stop bit and baud rate match the communication settings you select. Contact Mitsubishi for further help if unable to configure the module.

Configuring Qlarity Foundry™

If you are not familiar with Qlarity Foundry, you may find it helpful to complete the first Qlarity Tutorial. The Qlarity Foundry software and tutorials are available free at www.qlarity.com.

After downloading and installing Qlarity Foundry, start a new workspace.

If the MitsubishiFX library is not in the library list, it may need to be downloaded from our website (www.qlarity.com). To add a library to your workspace, select “Edit Libraries” from the “Edit” menu. Then select “Add existing library” and “Add Other library” to select to the library file.

Once you have added the MitsubishiFX library, create an instance of the *FXComm* and *FXLabel* objects. You should see these two objects listed in the object tree as *FXComm_1* and *FXLabel_1*.

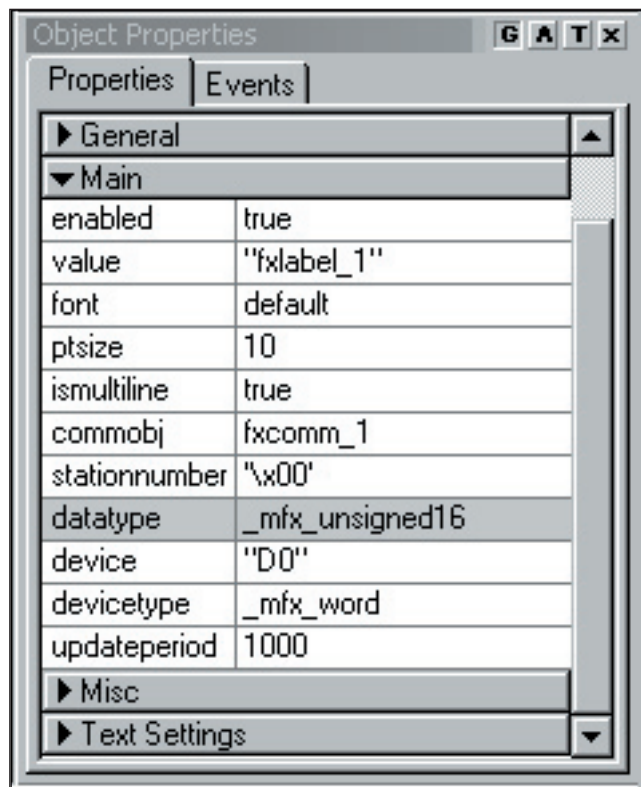


NOTE: You should only have one FXComm object in your workspace for this tutorial.

Select the object *FXComm_1* in the object tree so that it is highlighted and look at the Object Properties window in the lower right. Make sure the *serialport* property is set to the serial port number that you will be using with the terminal (e.g. COM1).

Select the *FXLabel_1* object in the Object Tree window so that it is highlighted. Look at the Object Properties window in the lower right of the screen. In the Main category, make sure that the *commobj* property is set to the *FXComm* object.

The *stationnumber* property identifies the controller, while the *device* and *devicetype* properties specify the address of the data to be read.



- Typically the *stationnumber* property will stay at 0 unless you are talking to multiple devices over 485 serial.
- The *datatype* property specifies what format the data is to be converted to after a successful query.
- The *updateperiod* property specifies how often to send a query to the controller.

You may want to start out by reading a 'D' address word which is common among FX Controllers.

Simulation

Before simulating your application, make sure that the Qlarity Foundry and the GX Developer serial port properties are set correctly. In Qlarity Foundry, you can configure serial ports in the Tools, Settings menu.

In Qlarity Foundry, press the Simulation view tab to run this program on your PC. The MitsubishiFX library will periodically communicate with the Mitsubishi controller to assure continued communications.

If there is trouble communicating with the controller and no response is received, you will see a message displayed at the top of the screen.

fxcomm_1: Sending packet to verify connection with controller

If you see this message, please check the wiring to the controller and check the serial settings for the controller and Qlarity Foundry.

NOTE: You may need a null modem cable for the connection to the PC from the FX controller (if using Qlarity Foundry in simulation mode).

The configuration shown in the screen shot at the left:

- The *FXLabel* will read a 16-bit word at address 'D0' from a single Mitsubishi FX controller at *stationnumber* 0.
- The *FXLabel* will query the device once per second. After a query, the received 16-bit word will be converted to an unsigned integer and displayed in the label.

NOTE: Not all device addresses are readable from the FXcontroller with the FXcommunication module. If your having trouble reading a device address, try setting the devicetype property to _mfx_word and _mfx_bit. If you are still unable to read the device address it is most likely not supported in the protocol and may need to be copied into a known available device address such as 'D' before it can be queried.