

FXTION

Broadcast Control Engineering

www.fxtion.com

FXDrvGPI

Version 1.00.23 – 22/12/11

Authors : Simon Dowson & Adam Brocklesby

Introduction

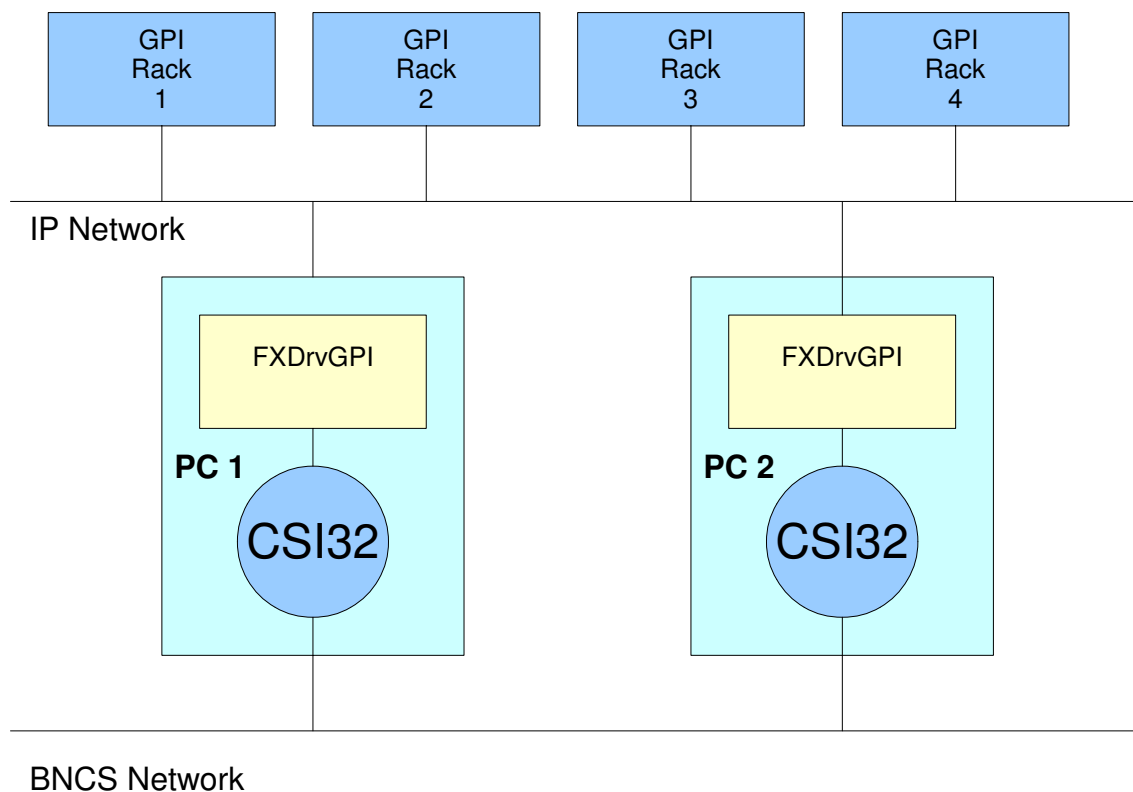
This document describes an FXtion GPI driver ***FxDrvGPI.exe***, which is primarily designed to control the ***R2X GPI Rack***. This driver is also able to interface to ***R3-32*** and ***R2-576*** controllers running firmware version ***2.00.xx***.

Documentation

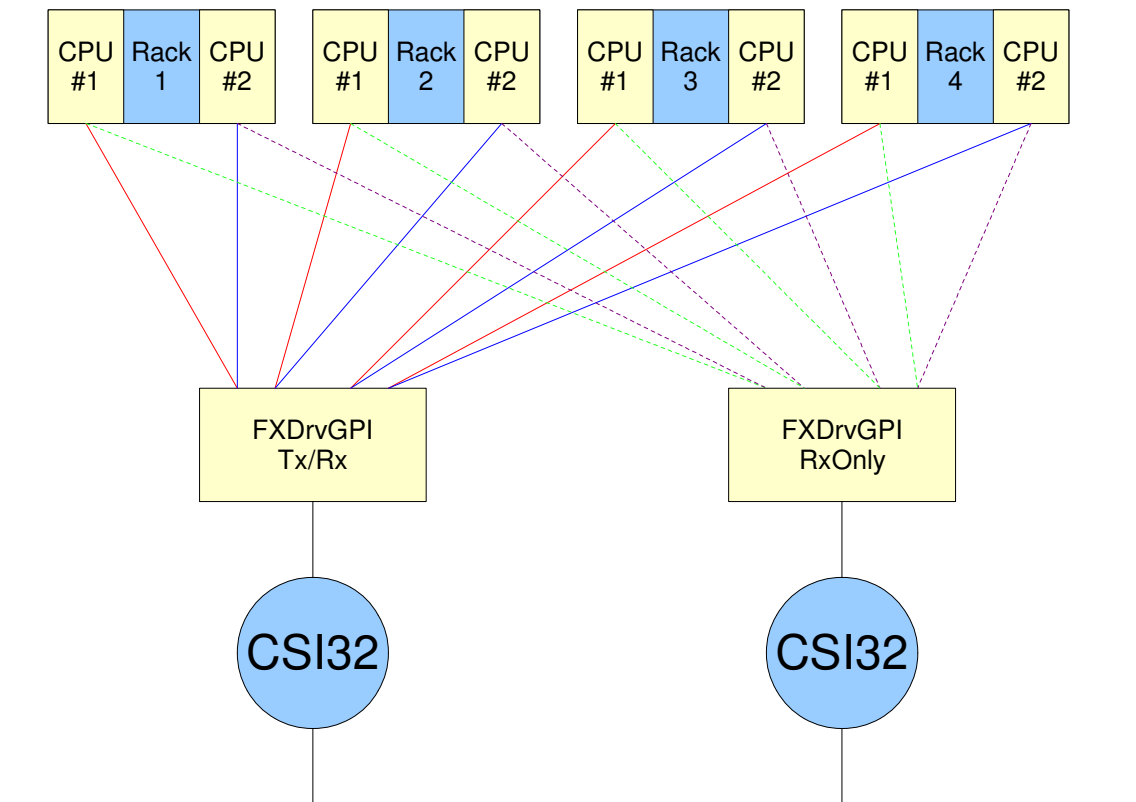
Important: This document is being written during the development of the driver application and only refers to the version number of the application specified at the end of this document.

Overview

The driver can provide redundant control of up to 32 R2X GPI racks, where each rack is fitted with 2 CPU cards. A pair of drivers can provide complete fail over protection. The basic topology is shown below :-



Each driver maintains connections to both controllers in a rack. This protects against a coincident failure of both a driver, or its host PC, and R2X CPU. The interconnection between the drivers and the R2X racks is shown below :-



The coloured lines represent individual session based connections between the drivers and the R2X racks.

The driver communicates with each CPU card via TCP which guarantees delivery of both commands and reverts. The TCP connection is in the form of a pseudo TelNet session with messages being in ASCII text. This improves diagnostics without compromising on speed.

Each CPU card can support 3 simultaneous TCP sessions. Two are for control by the dual redundant drivers and one is for maintenance and diagnostics.

Each CPU beacons its status every 20 seconds, by way of a broadcast UDP message on port 40200, with its configuration parameters such as name, location, IP addresses and port numbers.

Inputs and outputs are referenced via their rack, card and index and this is reflected in the configuration of the driver which will be covered later.

The I/O and invert states are always controllable. The mask states can only be changed dynamically on an R2-576 where each I/O direction bit is configurable.

R3-32 Control

When an **R3-32** is running **V2.00.xx** firmware the driver is capable of controlling up to 32 R3-32 units or a combination of R3-32s and other units. The R3-32 appears to the **FxDrvGPI** driver as **8** cards each with **8** I/O. The **first 4** cards are inputs and the **second 4** cards are outputs.

Inversion control of each I/O index is available but I/O direction is fixed due to the nature of the hardware.

R2- 576 Control

When an **R2-576** is running **V2.00.xx** firmware the driver is capable of controlling up to **8** R2-576 units or a combination of R2-576s and other units. Each R2-576 appears to the **FxDrvGPI** driver as **20** cards each with **24** I/O.

When using an **R2-576** with **SMART** connectors the control of inversion and direction for each I/O index is available as far as the **SMART** connectors themselves, although the availability of direction control is ultimately determined by the hardware the **SMART** connectors are plugged into.

Regardless of how many **SMART** connectors are connected the R2-576 will always appear to the **FxDrvGPI** driver as a full set of **20 SMARTS** (Cards) x **24** I/O, which equals **480** indexes.

Beacon Messages

The primary means of control is via **TCP** to a **TelNet** style interface. This offers guaranteed message delivery as well as a reliable means of detecting the presence or absence of a rack CPU.

In addition to TCP control the R2X controller CPUs also issue a regular **UDP** message as a status beacon. This message is optionally displayed by the driver in the scrolling diagnostic window as an aid to configuration. To view the status beacon messages for all racks select the Status tab on the grid. The display will look similar to the picture below:-

BNCS Network
Tx 00000000 Rx 00000001

GPI Network
Tx 00000000 Rx 00000000

Logging mode OFF
Logging mode OFF
Show Beacon Messages is OFF
View mode is I/O STATES
Opening connection to 192.168.1.201
Opening connection to 192.168.1.202
Ready
CONNECT : Now connected to CPU 1 on Rack 2
CONNECT : Now connected to CPU 1 on Rack 1

Status: **CPU1** 192.168.1.201 **CPU2** 192.168.1.202

	Rack Id	CPU Id	CPU Pos.	Rack Name	Rack Location	MAC Address	IP Address	IP Port	Up Time	Temp	PSU Volts	Master Slave
1	0	1001	Left	Default System	Location Not Set	0.8.215.0.111.200	192.168.1.201	23	0004 - 01:47:47	24	7.7	Master
2	0	68	Left	Default System	Location Not Set	0.8.215.0.111.202	192.168.1.202	23	0000 - 04:59:15			Master
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												
21												
22												
23												
24												

Status : Ready

Rack Id

This is an optional value assigned during configuration as a unique numeric identifier in the system.

CPU Id

This is a 2 byte read only value assigned to the CPU controller card when it was programmed. No two cards will have the same Id, therefore this value uniquely identifies the CPU card.

CPU Pos(ition)

This is an optional value assigned during configuration and relates to the position of the CPU card in an R2X frame. CPU cards are normally fitted to the far left or right of a rack and labeled accordingly.

Rack Name

This is a value assigned during configuration. It can be used to identify the owner or purpose of the rack.

Rack Location

This is a value assigned during configuration and can be used to identify where the rack is installed, such as building, room, bay number etc.

MAC Address

This hardware address is assigned during initial configuration at the time of manufacture. The last two bytes of the MAC address are the same as the CPU Id. This has been chosen for convenience, since the CPU Id must be unique, therefore the MAC address is guaranteed to be unique between GPI frames.

IP Address

The IP address to be used for control connections.

IP Port

The IP port to be used for control connections. The default is 23.

Temp(erature)

This is a value in centigrade sampled every 10 seconds, although the beacon will only report every 20 seconds.

PSU Volts.

This value shows the unregulated DC volts applied to the CPU card.

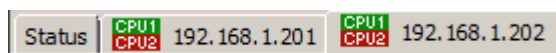
Master / Slave

In any rack only one CPU can be master at any one time. The driver will always select the left to be master if available. If the right appears as master then a changeover has occurred.

Local Diagnostics and Control

The status of up to 32 configured racks can be viewed and changed from the driver. A separate tab appears for each. If the **name** is configured it will appear in the tab, otherwise the primary IP address will be shown.

Also on the tab a bitmap shows the current connection status to the two CPU cards.



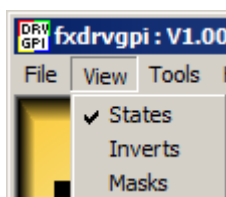
If the background is green then there is a connection to that CPU card. If either or both legends are green then control is available. If both are red then no control is available from this driver.

Each rack is shown as a number of columns with each column representing a card in the rack. The row indicates the I/O index on each card. The contents of the cell in the grid shows the BNCS index value.

The screenshot shows the 'fxdrvapi : V1.00.6.0 : WS 995 : DEV 63 : TxRx' window. It has a menu bar (File, View, Tools, Help) and a toolbar. The main area is divided into two panes. The left pane shows 'BNCS Network' and 'GPI Network' settings. The right pane shows 'Ready', 'CONNECT : Now connected to CPU 1 on Rack 1', 'View mode is I/O STATES', 'Local control disabled - Check the Tools menu option', 'Database edit mode OFF', and 'Output control via grid is ON'. Below the panes is a status bar showing 'Rack 1, Card 7, Index 3 : Input 51' and buttons for 'On', 'Off', and 'Toggle'. The main grid shows 24 rows (State 1 to State 24) and 20 columns (Card 1 to Card 20). Each cell contains a BNCS index value. The grid is color-coded: green for 'On', red for 'Off', and grey for 'Toggle'. The 'On' button is highlighted in the status bar.

If the card is an output card, and local control has been enabled via the Tools menu, an output can be toggled by left clicking a cell to select it then pressing one of the On, Off or Toggle buttons. If the card is an input card then it will display the input status of the card.

Menu Options



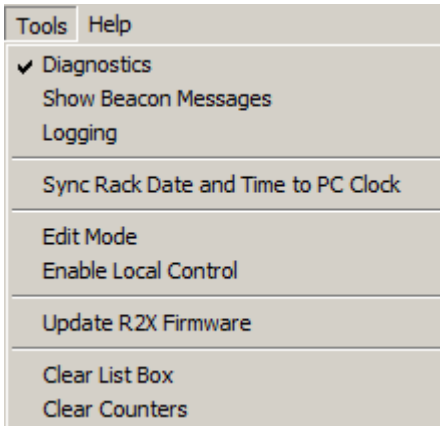
The **View** menu gives three options. Selection of **States**, **Inverts** or **Masks** will be reflected in the grid on the currently selected Rack tab. The rows will be labeled accordingly. Each index can be toggled by clicking on the required cell and then clicking the buttons above the grid on the right hand side.

Cell colours are :-

States : Off = Green, On = Red

Inverts : Normal =Cyan, Inverted = Purple

Masks : Output = Blue, Input = Yellow



The **Tools** option on the menu bar gives access to the diagnostic and logging features. If **Diagnostics** is checked then status information is presented in the scrolling window.

If the **Show Beacon Messages** option is checked then the regular messages from the CPU cards are sent to the scrolling window.

If the **Logging** option is checked any output to the list box is also written to a file. The entries in the log file are time and date stamped. The log file is appended to rather than overwritten and includes entries that show when the file was opened and closed.



The **Sync Rack Date and Time to PC Clock** option is enabled when a connected rack tab is selected. Clicking it will set the time and date of connected CPU cards to that of the controlling PC.

When a **Rack Tab** is selected the **Edit Mode** option on the menu becomes available. When selected the logo changes as shown to the left. Clicking on a cell causes the current database name to be presented on a line above the grid. In Edit Mode this line can be edited. Pressing the Enter key will commit the new name to the database via CSI. Deselecting **Edit Mode** will restore the original logo and the ability to control the I/O states via the grid cells.

When the **Status** tab is selected the **Edit Mode** option is disabled. Edit mode is not persistent and will be turned off if the driver is closed and restarted.

Selecting **Enable Local Control** will allow outputs to be toggled by selecting an output from the grid and then clicking the buttons above. The logo changes to reflect the mode.

This setting is not persistent and will be off by default whenever the driver starts. Additionally, if no outputs are toggled during the timeout period configured in the INI file then the local control mode is automatically disabled again.

The **Update R2X Firmware** enables the remote updating of FXOS firmware in R2X racks and is described in the **Firmware Updating** section later in this document.

The remaining two menu options clear the list box and reset the message counters to zero respectively.

Configuration

Configuration is achieved through the **DEV_XYZ.INI** file associated with the device Id supplied as a command line parameter for the driver. The exact location of the INI file is dependent on the type of BNCS system in use. When the driver is first run it will create default entries for all parameters.

Overall configuration information for the driver is shown in the **[fxdrvmpi]** section as shown below :-

[fxdrvghi]	
DebugMode=0	If set to '1' then any status messages will be output to the scrolling listbox. This flag can be toggled from the application menu.
LoggingMode=0	If set to '1' then any status messages sent to the scrolling listbox will also be sent to a log file. This flag can be toggled from the menu.
BeaconMode=0	If set to '1' then any beacon messages will be output to list box. This flag can be toggled from the application menu.
AuxiliaryMode=0	If set to '0', which is the default, then this parameter has no effect. If set to '1' then the driver will not influence the Master/Slave condition of the connected CPUs unless both report that they are slave, in which case one will be asserted to be Master. This mode is intended to provide auxiliary control of a rack in case both the primary dual redundant pair of drivers fails.
LocalControlTimeout=300	When Enable Local Control is enabled from the Tools menu this value determines the period of inactivity before local control is disabled again. If set to zero then local control never times out.
RUTDelay=5	The time delay in seconds between sending 'aRe yoU There' messages from the driver to each CPU card.
MaxRUTUnAcks=2	The number of RUT messages that can go unacknowledged before the driver closes the connection to the CPU card.
RelayBeaconIndex=0	If set to a valid BNCS device index number the driver will produce an InfoDriver revertive from this slot, containing beacon data, every time a message is received from a CPU card. When set to zero, which is the default, then no beacon messages are relayed.

Configuration of each individual rack and how its I/O is presented within the **FXDrvGPI** driver is achieved through individual sections from **[Rack_01]** to **[Rack_32]**.

Note : Only when at least one IP address has been configured for a rack will a tab appear in the drivers grid.

[Rack_01]	
UserName=admin	The user name required to log in in to the CPU card in the rack. Both CPU cards should be configured to accept the same username and password.
Password=pw	The password required to log in in to the CPU card in the rack. Both CPU cards should be configured to accept the same username and password.
ControlPassword=changeit	The password required to allow write access. Without this only status information will be available.
Location=- - -	The location of the rack. This will appear in beacon messages displayed on the status grid.
IPAddress_1	The IP Address of the primary CPU in the rack. This should be the left hand card. This is only a recommended convention.
IPAddress_2	The IP Address of the secondary CPU in the rack. This should be the right hand card. This is only a

	recommended convention.
Port_1=23	The IP port of the primary CPU card
Port_2=23	The IP port of the secondary CPU card
Card_01=Offset,Indices ... upto MaxCards entries .. Card_XX=Y,Z	Where 'Offset' is the index in BNCS where control of this card should begin and 'Indices' is the number of I/O. For example Card_01=201,16 would configure the driver to present card 1 from this rack as BNCS indices 201 through to 216.
Card_01_Inverts=000000000000000000000000	For each I/O on each card the bit inversion can be set. A zero means not inverted, whilst a 1 means inverted. The leftmost digit is the first I/O on the card and the rightmost digit is the last.
IPAlarmIndexCPU1=0	If set to a valid GPI index number the driver will produce a GPI revertive to indicate the connection state to CPU 1. The default index is zero indicating no revertive will be generated. A revertive will be zero for comms fail and 1 for a successful connection.
IPAlarmIndexCPU2=0	If set to a valid GPI index number the driver will produce a GPI revertive indicating to indicate the connection state to CPU 2. The default index is zero indicating no revertive will be generated. A revertive will be zero for comms fail and 1 for a successful connection.

Databases

The driver uses database 2 for the names that appear in the drivers rack grids. Since all racks controlled by the driver share a common database the name index corresponds to the I/O index. So for example if you configure a card to have outputs from 1001 to 1016 then you should add names in the database file for indexes 1001 to 1016. The database entries can be up to 256 characters in length and can be edited within the driver, via the Edit Box above the grid, by selecting **Tools->Edit Mode**.

I/O Inversion

The CPU card in the R2X rack is capable of handling I/O inversion, but for BNCS purposes it is easier to leave the CPUs configured not to invert and for the driver to apply inversion based on settings in the INI file.

For each configured card in a rack there is a line in the INI file to enable the inversion of 1 or more I/O to be set. For example:-

[Rack_03]

Card_07_Inverts=010100000000000000000000

On rack 3 card 7 is configured to have I/O 2 and 4 inverted, all other I/O is not inverted. Since the main and reserve drivers will share the same INI file they will also pick up the same I/O inversion tables. Replacing the CPU, or I/O card, or re-powering the rack will not affect GPI inversion as it is all handled in the driver.

The I/O presented on the driver GUI and to BNCS via the network is before inversion is applied. For example if an output is to to invert and then BNCS sets that output high it will appear high on

the driver GUI, but the switch command sent to the R2X rack and its subsequent state will be low. Similarly a received high input that is configured in the driver to invert will present a low on the driver GUI and to BNCS.

Comms Fail

The driver is able to optionally notify the connection state for each CPU card via a GPI index allocated to the task. The indexes are set in the INI file via the **IPAlarmIndexCPU1** and **IPAlarmIndexCPU2** parameters for each rack. When a successful connection is made a revertive of **1** will be generated for that index. A revertive of **zero** will be generated if comms fails and the connection lost.

If a poll is issued for a range of indexes then revertives will only be generated for the indexes that land on racks with which the driver has an active connection.

TCP Control - TelNet

The GPI Racks are controlled via one or more **TCP** connections. A secondary **UDP** connection is catered for within the software architecture, but currently only used for beacon messages.

The **TCP** connection is **ASCII** and based on three letter commands, such as **GRS** for example, meaning **Get Rack States**

The three letters signify **Major** Group, **Sub** Group and **Parameter**.

The Major Groups are :-

- **Get** – for getting values
- **Set** – for setting values
- **Response** – for revertives and general return values

The Sub Groups are:-

- **GPI** – for manipulating individual input output indices
- **Card** - for manipulating an entire card of I/O
- **Debug** – for obtaining status information from the CPU FXOS operating system
- **Help** – for accessing help information
- **Rack** - for manipulating an entire rack of I/O
- **System** – for system management functions
- **User** – for managing connection timeouts and admin

A full table of all commands, their parameters and meanings can be found towards the end of this document.

Opening a TelNet Connection.

Each R2X CPU card is able to support three simultaneous TCP connections for control and interrogation purposes. Two are intended for use by main and reserve **FXDrvGPI** drivers. The third is for diagnostic use.

Open a TelNet connection to the IP Address of one of the CPU cards. The default port is 23. If successful a banner and prompt similar to the one below will be displayed.

```
*****
*      Fxtion FXOS Interface Server V1.00.00 : Apr 6 2011 - 20:28:22
*****
```

Connection #3
Ready...

Under normal operational circumstances the 'Connection' is likely to be **#3**, since the main and reserve drivers should already have been allocated the first **#1** and **#2** connections.

If the rack is configured to require a user name and password the prompt will say :-

Ready for login...

If logging in is required then use the following **Set User Authentication** command :-

SUA <Username> <Password> [<Control Password>]

eg. **SUA admin pw virax** followed by the **<Enter>** key.

If authentication fails the response will be :-

RUA Login failed.
Ready for login...

If authentication is successful the rack will respond with :-

RUA Login successful.
Ready ...

If the control password is correct then the connection will be allowed write access. If the control password is incorrect or omitted then only read access is available. You can now use any of the commands in the Command Reference. Making state changes via the driver will reflect back over the TelNet link on your session. Similarly, any changes you make over the TelNet session will be reflected by the drivers and within control system network. This makes the TelNet facility a useful configuration tool and debugging aid.

The entire I/O status of a rack can be obtained simply by sending the **GRS** (Get Rack States). The rack will reply with an RRS (Response Rack States) message of the form :-

```
RRS 1201FA FA0011 A587DE FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF  
FFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF
```

This shows 16 blocks of 3 hex bytes (24 bits), one block for each card, representing the I/O state for the card. **FFFFFF** is returned for card slots that are not populated.

The Invert and Mask states can be obtained using **GRI** and **GRM** commands respectively. For an input toggling an invert bit will always cause the reported I/O state to change. For an output toggling an invert bit will always cause both the reported I/O state and the output state to change.

A full table of available commands can be found on the following pages. The number and scope of these will certainly increase with each version of the FXOS firmware in the GPI racks.

To close the TelNet session the client application can be told to disconnect or the user can issue a **SUQ** (Set User Quit) command which will force the GPI rack to close the connection. Using the

SUD (Set User Disconnect) command it is possible to forcibly close other sessions.

Some commands are for development purposes, such as **SGD** (Set Graphics Data), which can be used to talk directly with the OLED graphics processor on the CPU card.

The **SPS** command enables the configuration of various parameters to be changed. In all cases the change only takes place after a restart of the firmware.

The **FXOS Command Reference** can be found towards the end of this document.

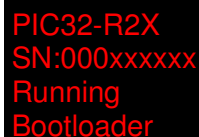
Firmware Updating

The main processor firmware in the CPU cards can be updated over the network from the driver. The processor contains a bootloader that can be primed to activate when the CPU card is restarted.

When restarted in bootloader mode the CPU card will passively wait until it receives a message from the driver, if no message is received after 20 seconds then the existing firmware will be executed. If the CPU card is subsequently restarted, without again priming the bootloader to activate, the firmware will be executed immediately.

The bootloader can be primed in two ways. Either via the front panel of the CPU card or, if card already has existing firmware, via a **TelNet** control session. The latter method is especially useful if the GPI rack to be reprogrammed is remote or physically inaccessible.

Press and hold the joystick control on the front of the CPU card, then press and release the reset switch. Continue to hold in the joystick control until the OLED display shows :-

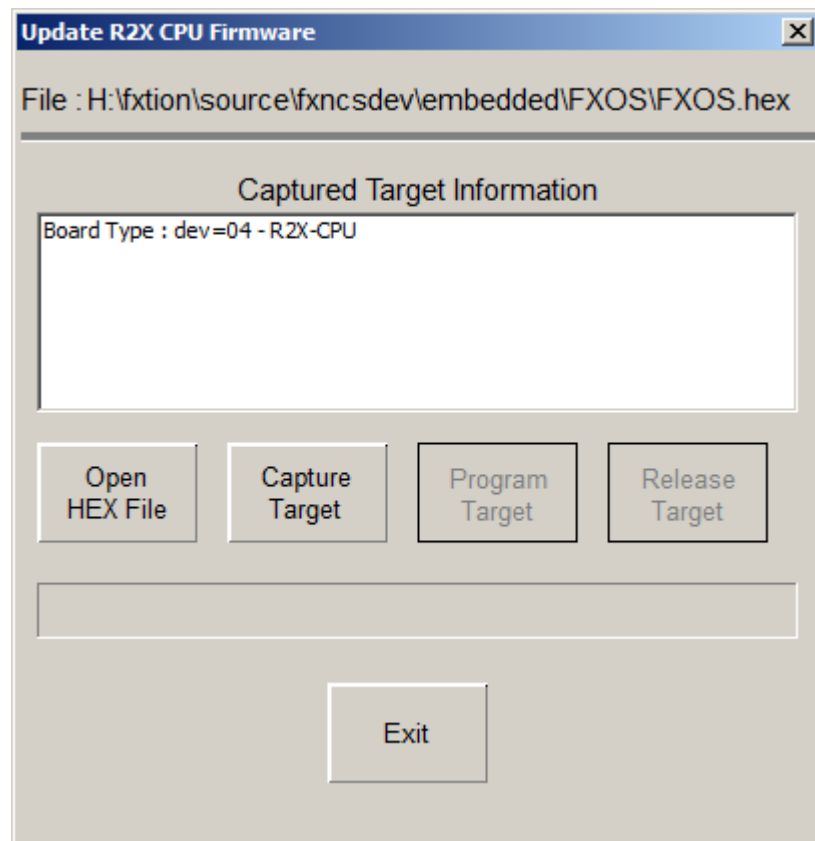


PIC32-R2X
SN:000xxxxxx
Running
Bootloader

The alternative way is to connect to the CPU card over a **TelNet** session and issue a **Set System Bootloader** command with a parameter of 1, followed by a **Set System Restart** command with a parameter of **RESTART**. For example :-

SSB 1
SSR RESTART

The firmware is updated using the driver via the tools menu option. Select **Update R2X Firmware**. The dialog box below will appear. Click the **Open Hex File** button and select the firmware file you wish to upload. Note that firmware files for the R2X CPUs are prefixed **FXOS** and have a suffix of **.HEX**. The selected file path should appear at the top of the dialog box.

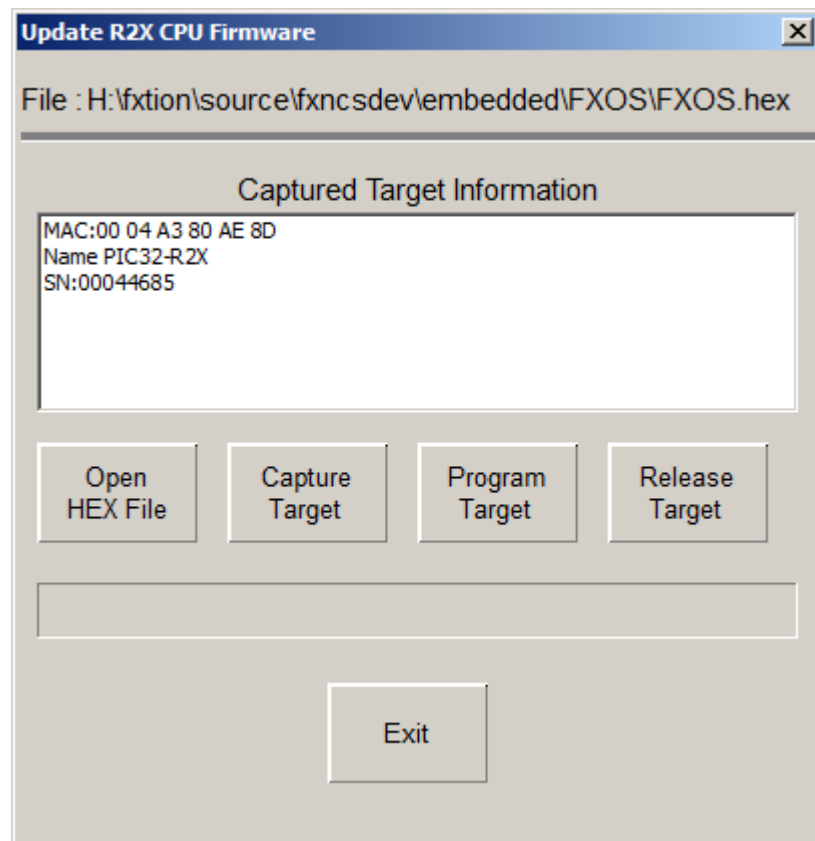


The CPU that you wish to reprogram is referred to as the target because it will be the target for the firmware being uploaded. In the ***Captured Target Information*** window a line will appear confirming that the loaded file is indeed intended for an R2X CPU.

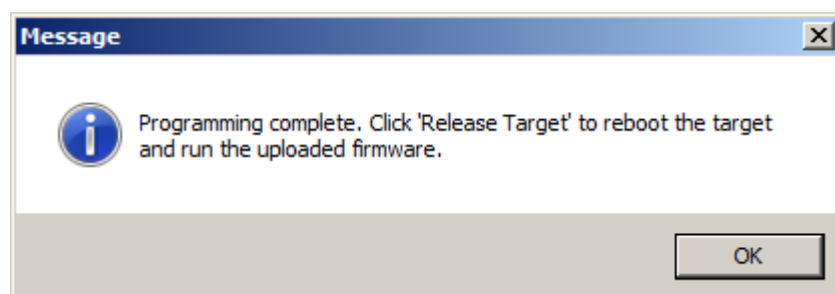
The ***Program Target*** and ***Release Target*** buttons will be disabled and remain so until a CPU card has been successfully captured.

Using one of the methods previously described, arm the bootloader and restart the CPU in bootloader mode. The CPU card will sit passively for up to 20 seconds waiting for a capture message from the driver. Now press ***Capture Target***. The driver will send a broadcast message to the network, to which the CPU card will respond.

When a CPU target is successfully captured the dialog box will show additional information in the ***Captured Target Information*** window and the ***Program Target*** and ***Release Target*** will be enabled.

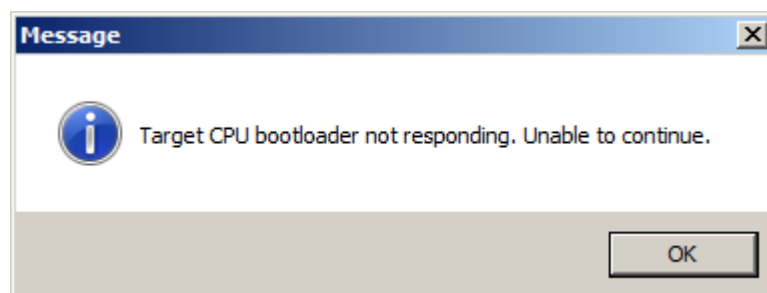


Press the **Program Target** button. The progress bar just above the **Exit** button will advance as the target is programmed. If programming is successful then the following message will be displayed



Press OK and then the **Release Target** button on the dialog. The CPU card will be restarted and the new firmware executed.

If for any reason programming is interrupted then the following message box will be displayed.



If this should happen then re-arm the bootloader and try again.

FXOS Command Reference - V1.00.01 - 01/07/11

GET Commands

<u>Command</u>	<u>Meaning</u>	<u>Function</u>
GCI <Card #>	Get Card Inverts	Gets the I/O invert flags for the specified card
GCM <Card #>	Get Card Masks	Gets the I/O mask flags for the specified card
GCS <Card #>	Get Card States	Gets the I/O states for the specified card
GDI <Option>	Get Debug Information	Returns hex dumps of internal arrays
GPS <Section> <Parameter>	Get Profile String	Returns the value of the configuration entry
GGI <Card> <Index>	Get GPI Invert	Gets the invert flag of the specified GPI index
GGM <Card> <Index>	Get GPI Mask	Gets the mask flag of the specified GPI index
GGs <Card> <Index>	Get GPI State	Gets the state of the specified GPI index
GRE	Get RUT Enable	Gets the RUT mechanism active state
GRI	Get Rack Inverts	Gets the I/O invert flags for the entire rack
GRM	Get Rack Masks	Gets the I/O mask flags for the entire rack
GRS	Get Rack States	Gets the I/O states for the entire rack
GSC	Get System Connections	Get a list of logged on IP addresses and other info
GSD	Get System Date&Time	Gets the real time clock date and time
GSM	Get System Temperature	Gets the temperature from the on board sensor
GSU	Get System Up-Time	Gets the elapsed time since the CPU was reset
GSV	Get System Volts	Gets the two system input voltage values

SET Commands

<u>Command</u>	<u>Meaning</u>	<u>Function</u>
SCI <Card #>	Set Card Inverts	Sets the invert flags for the specified card
SCM <Card #>	Set Card Masks	Sets the mask flags for the specified card
SCS <Card #>	Set Card States	Sets the state flags for the specified card
SGD <Hex Bytes x N>	Set Graphics Data	Sends bytes of graphics data to the OLED display
SGL <Card> <Index> <State>	Set GPI Invert	Sets the invert of the specified GPI index to <State>
SGM <Card> <Index> <State>	Set GPI Mask	Sets the mask of the specified GPI index to <State>
SGS <Card> <Index> <State>	Set GPI State	Sets the bit of the specified GPI index to <State>
SPS <Section> <Parameter> <Value>	Set Profile String	Sets the value of the configuration entry. Please note the <Value> string cannot have embedded spaces.
SRE <0/1>	Set RUT Enable	Enables (1) or disables (0) the RUT mechanism
SSB <0/1>	Set System Bootloader	If set to 1 arms the boot loader for 20 seconds
SSD <D> <M> <Y> <H> <M> <S>	Set System Date & Time	Sets the real time clock
SSM <0> or <1>	Set System Master	Sets this CPU to be master (1) or slave (0)

SSR <RESTART>	Set System Restart	Signals to the system it should restart
SUA <Name> <Password> <Control>	Set User Authentication	Allows a user to login.
SUD <Connection Id>	Set User Disconnect	Disconnects a session
SUQ	Set User Quit	Closes the TCP connection

Response Commands

<u>Response</u>	<u>Meaning</u>	<u>Function</u>
RCI <123456>	Response Card Inverts	Response to GCI
RCM <123456>	Response Card Masks	Response to GMI
RCS	Response Card States	Response to GCS
RGI	Response GPI Inverts	Response to GGI
RGM	Response GPI Masks	Response to GGM
RGS	Response GPI States	Response to GGS
RPS	Response Profile String	Response to SPS and GPS
RRI	Response Rack Inverts	Response to GRI
RRM	Response Rack Masks	Response to GRM
RRS	Response to States	Response to GRS
RSC	Response System Connections	Response to GSC
RSD	Response System Date	Response to SSD and GSD
RSH	Response System Help	Response to GSH
RSM	Response System Master	Response to SSM and GSM
RSR	Response System Restart	Response to SSR
RST	Response System Temperature	Response to GST
RSU	Response System Uptime	Response to GSU
RSV	Response System Volts	Response to GSV
RUA	Response User Authentication	Response to SUA
RUR	Response User Reply	Response to RUT
RUT	<u>Are You There</u>	Responds with RUR

Credits

Many thanks to Brian Walton for proof reading, corrections and suggestions. Thanks also to Robert Brown and Rob Staton for testing and feedback.

Version History

1.00.00	15/04/11	First release
1.00.01	22/04/11	Added SSM and SSB commands Added firmware updater to driver and documentation
1.00.02	05/05/11	Added 'Sync Rack Date and Time to PC Clock' to the 'Tools' menu
1.00.03	11-06-16	Added Enable Local Control to menu to manage ability to toggle outputs from the driver. Added timeout in INI file to configure period of inactivity before control mode disabled automatically No longer reverts when State/Mask/Invert view is changed
1.00.04	11-06-21	Added database section to documentation Fix to beacon reception
1.00.05	11-06-22	Added RUTDelay and MaxRUTUnAcks as configurable options for CPU fail detection Reduced unacknowledged RUTs from 3 to 2 and RUT delay from 20 to 5 seconds Increased cards to 20 to accommodate R2-576 driving a fully populated R2X Rack of 20 cards Fix to revertive mapping of masks and inverts Now detects if CSI is not enabled in driver mode and closes down if not.
1.00.06	11-06-27	Fix to message counters Added buttons to toggle the outputs instead of just the grid cells Added database name line and moved editing function to it
1.00.07	11-06-28	Now uses database 2 so that names greater than 16 characters can be used
1.00.08	11-06-29	Using the cursor keys to move around the grid changes the selection Selecting an unconfigured cell clears the database line. Added configuration option to generate GPI revertives on change in CPU connection status Added configuration option to relay beacon messages as an InfoDriver revertive.
1.00.09	11-06-30	Documents new GRE & SRE commands in firmware 1.00.01 Canceling the Open File in the firmware upload dialog no long brings up error. Now sets date and time on CPU 2 even if there is a connection to CPU 1 Masks and Inverts can now be toggled if action supported by type of rack. Now processes invert and mask tables from CPU 2 when CPU 1 not connected Now creates a rack tab if either IP address is configured
1.00.10	11-07-03	Numerous corrections to documentation Local control is now still available if the driver is in RxOnly Now uses the presence of a '.' in an IPAddress to determine whether to show a tab Fixed positioning of card information field in database change line above grid Disabled editing of I/O cells in grids Added SRE & GRE commands
1.00.11	11-07-06	Now forces none active CPU card to be slave.
1.00.12	11-07-10	Upper boundary fix to rack arrays intialisation No longer polls when connecting if entering slave mode Changed Notebook page to rack mapping to cope with none contiguous rack configuration
1.00.13	11-07-11	Added fifo buffering to beacon messaging to prevent IAC thread crashing wxWidgets redraws Removed debug code limiting status grid to only 6 beacons and causing crash
1.00.14	11-07-11	Modified method by which CPU connection notifications are sent to BNCS Bug fix to correctly apply inversion when switches received via BNCS
1.00.15	11-07-20	Added sentence to documentation about a restart being required after an SPS command in order for new setting to take effect. Unsolicited revertives now generated with a workstion number of zero.

1.00.16	11-09-10	DAT file now saved to config directory
1.00.17	19/09/11 15/11/11	Added information on inversion Added extra information to beacon message logging
1.00.18	11-11-17	RxOnly driver will no longer send switch commands to CPU cards
1.00.19	11-11-19	Reworked CPU card connection handling not to force CPU 1 to be the master Fix to remove filter that prevented revertives from CPU 2 if CPU connected Added RSM handler to ensure one CPU card is always Master by adding GSM command to the RUT routine. Added check to prevent drivers fighting to assert System Master state if badly misconfigured. Added countdown timeout column to Status Grid which removes beacon if not heard from within timeout period, currently 60 seconds
1.00.20	11-12-03	Added forward buffering to prevent TCP layer from splitting commands. The beacon timeout now reflect a green cell for normal operation and a red cell if there is a timeout. Fix to initial 'Open Connection' message Now creates daily log files Fix to set logging mode check mark on startup Beacon status revertives now buffered Reinstated the ability to poll status indices
1.00.21	11-12-11	Added Rack number before diagnostic information on Master/Slave status Completed forward buffering with fifo 2 Added Auxiliary mode for emergency GPI control Put in protection for TCP input buffer overflow Trapped for error message from FXOS 1.00.06 that does not end in a 0x0D
1.00.22	21/12/11	A test version adding the FXOS 1.00.06 error message trap to 1.00.19
1.00.23	22/12/11	Based on 1.00.21 this removes the fifo and uses the direct TCP mode used in 1.00.19