

IQSPI00 Serial Port Interface with RollNet



Table of Contents

Module Description	3
Rear Panel Views	3
Product Comparison.....	4
Block Diagram.....	5
Features	5
Technical Profile	5
Connections	6
Serial Connections	6
Configure Serial Ports 1 and 2	6
Pin Connections for 9 way D-Type Connectors.....	7
RollNet.....	7
Card Edge Functions.....	8
IQSPI00 Menu System.....	9
Operation from an Active Control Panel	10
Main Menu.....	10
Custom 1 to Custom 4 Menu Options	10
Display Menu.....	11
Setup Menu	12
Logging Menu.....	13
IQSPI00 Configuration File Editor	14
Introduction.....	14
Functions within the IQSPI00 Configuration Editor	14
Installation	14
Toolbar and Menus.....	15
Tools Menu.....	16
Help Menu	16
Starting a New File	17
Uploading a File.....	17
Downloading a File	18
Editing the Configuration File	19
Editing the File in Treeview.....	19
Editing in Output File mode	19
Editing Ports	20
Adding a Command.....	21
Deleting a Command.....	21
Editing a Command's Properties	23
The Command Tab	23
The Parameters Tab	23
The Data Tab.....	24
The Tally Tab.....	27
The Preset Tab.....	27
The Tracks Tab	28

Track Parameter window.....	29
The Logging Tab	30
File Header	30
Port Settings	31
Swapping Ports	32
Cut, Copy & Paste	32
Prompt to Save.....	32
Communication Between IQSPI and a Remote Device.....	33
Transaction Types	33
Command and Reply.....	33
Request and RequestReply.....	33
Tally and TallyReply.....	33
Transaction Type Example.....	33
Syntax of the IQSPI Configuration File.....	33
[IQSPI] File Header	33
[Port N] Sections.....	33
Command Definitions	35
[Strings] Section	44

Module Description

The IQSPI00 provides a programmable serial port interface for external devices and RollCall compatible products.

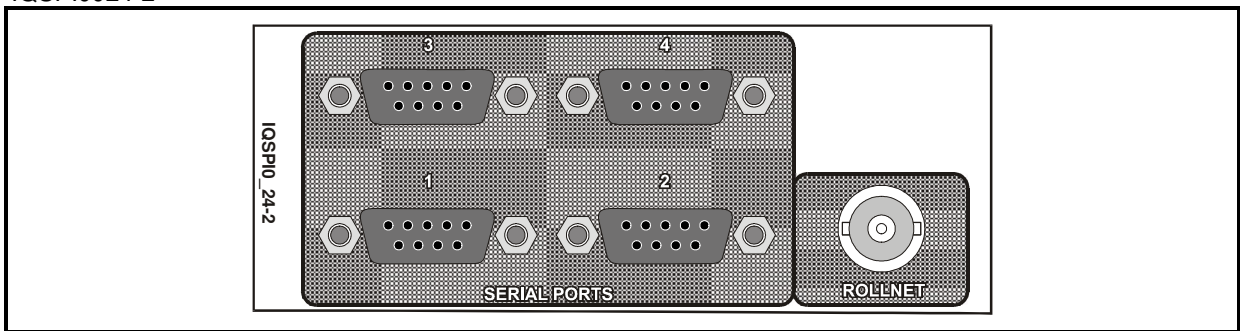
The IQSPI00 can control many types of devices that have serial ports, including tape machines, routers, and disk stores. In addition, the IQSPI00 can receive commands from an intelligent control panel with a serial port, providing a customized user interface.

Each button could activate pre-defined commands, such as router switching or frame freezing. Multiple actions from one input can be generated, producing many pre-defined commands via RollTrack. The IQSPI can communicate with other IQSPI and GPI (General Purpose Interface) products. Thus, pre-defined commands can be generated by external devices and sent to other external devices via the RollCall network.

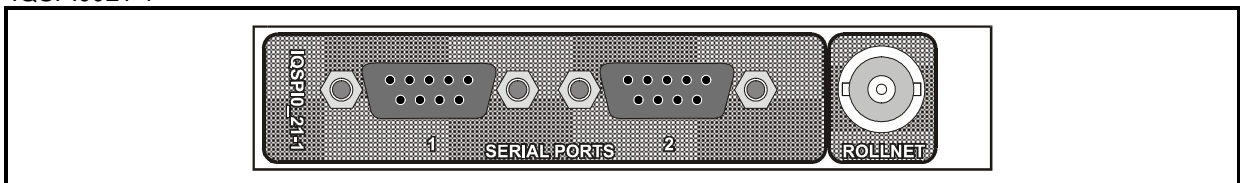
A Configuration Editor is provided for editing the file, uploading configurations from the IQSPI card, and downloading configurations to the IQSPI card.

Rear Panel Views

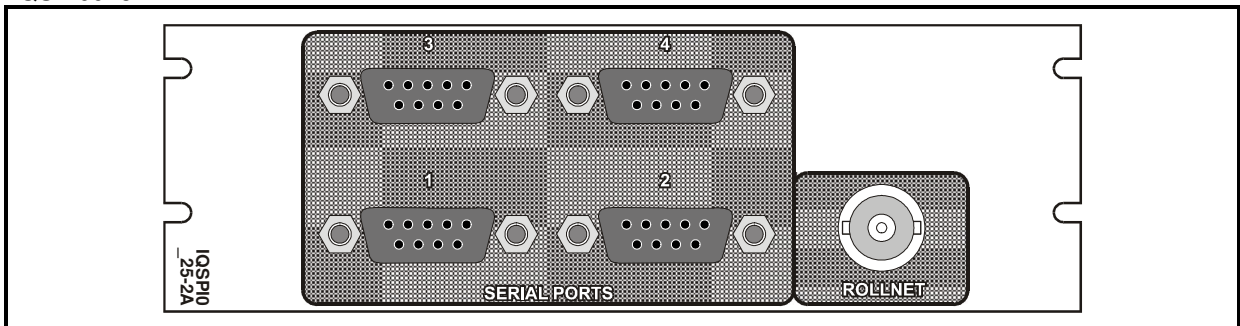
IQSPI0024-2



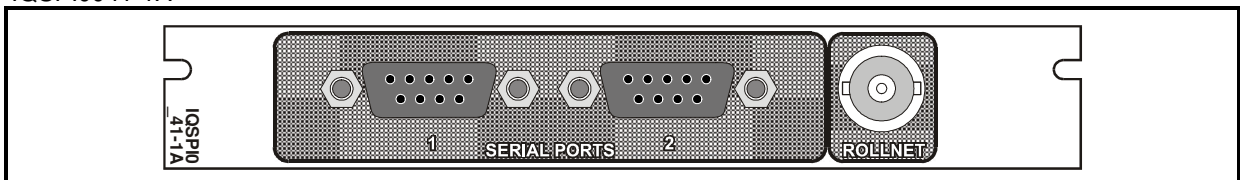
IQSPI0021-1



IQSPI0025-2A



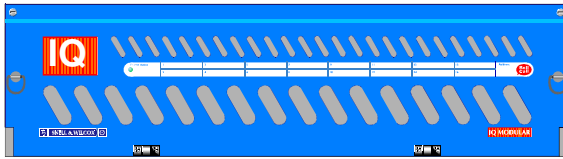
IQSPI0041-1A



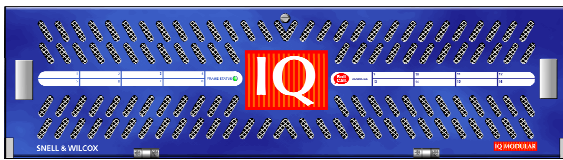
Note that there are two styles of rear panels available. They are not interchangeable between the two styles of enclosures. However, the cards may be fitted into any style of enclosure.

‘A’ Style Enclosure

Rear panels **with** the suffix A may only be fitted into the ‘A’ style enclosure shown below.



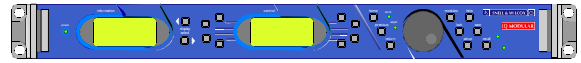
(Enclosure order codes IQH3A-E-0, IQH3A-E-P, IQH3A-0-0, IQH3A-0-P)



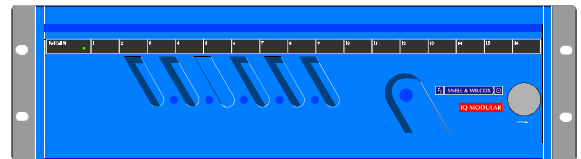
(Enclosure order codes IQH3A-S-0, IQH3A-S-P)

‘0’ Style Enclosures

Rear panels **without** the suffix A may only be fitted into the ‘0’ style enclosures shown below.



(Enclosure order codes IQH1S-RC-0, IQH1S-RC-AP, IQH1U-RC-0, IQH1U-RC-AP, Kudos Plus Products)



(Enclosure order codes IQH3N-0, IQH3N-P)



(Enclosure order codes IQH3U-RC-0, IQH3U-RC-P)

Product Comparison

Product	Serial Ports	RollNet Connection	Width & Style
IQSPI0021-1	2 via D-type	1 via BNC	Single 0
IQSPI0041-1A	2 via D-type	1 via BNC	Single A
IQSPI0024-2	4 via D-type	1 via BNC	Double 0
IQSPI0025-2A	4 via D-type	1 via BNC	Double A

This manual covers the following products:

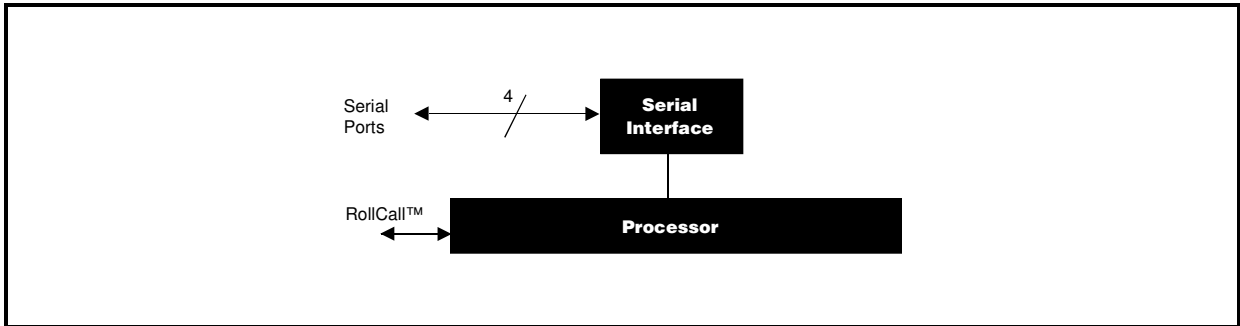
IQSPI0021-1 Serial Port Interface. 2xD9. 2 x Serial Ports, 1 x RollNet BNC

IQSPI0041-1A Serial Port Interface. 2xD9. 2 x Serial Ports, 1 x RollNet BNC

IQSPI0024-2 Serial Port Interface. 4xD9. 4 x Serial Ports, 1 x RollNet BNC

IQSPI0025-2A Serial Port Interface. 4xD9. 4 x Serial Ports, 1 x RollNet BNC

Block Diagram



Features

- Enables control of products on the RollCall network via external serial events
- Interfaces external devices to RollCall i.e. tape machines, routers and disk stores
- Two RS232/RS422 user-configurable ports
- Two further RS422 serial ports
- Multiple actions from one serial message with RollTrack
- External serial events produce RollCall logging messages
- Windows software program for function set-up
- Note: Contact sales office for a list of currently available interfaces to 3rd party equipment

Technical Profile

Features

Serial Ports

Ports 1 & 2 RS232/422 selectable connection via 9 way D-Type
 Ports 3 & 4 RS422 connection only via 9 way D-Type

Control Interface

RollCall 1 x RollNet Interface via BNC/75 ohm connector
 Format: 2.5 Mbit/s
 Indicators
 Data Sent For 4 interfaces
 Data Received For 4 interfaces
 RS232 mode
 RollCall network activity and status

Specifications

All Ports Speed..... 1200 – 115200 bit/s

Peak Mains Inrush Current following a 5 second mains interruption
 No mains input

Power Consumption

Module Power Consumption
 5.9 W max

Performance Information ... No performance degradations or cable length limitations

EMC Performance Information

Environment Commercial and light industrial E2

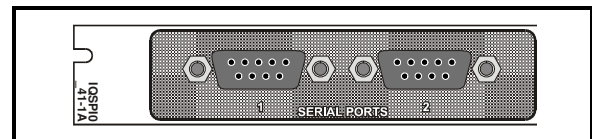
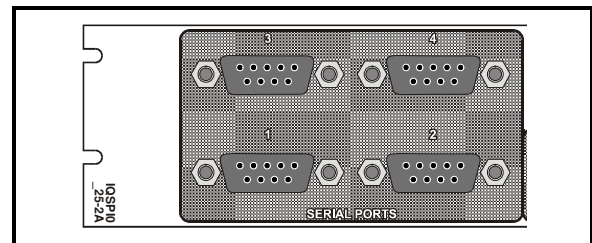
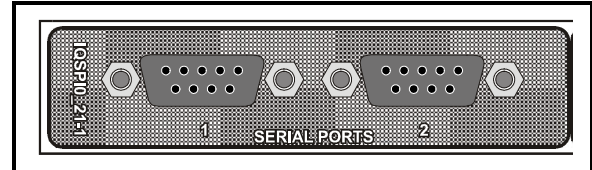
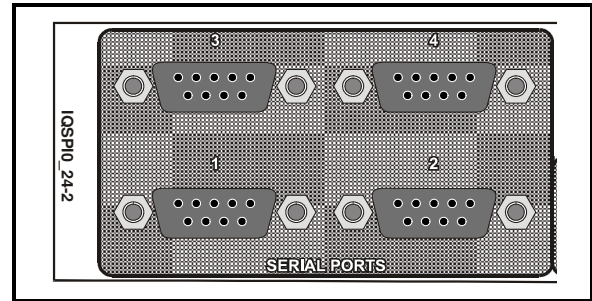
Connections

Serial Connections

The IQSPI00 is an interface between Serial Data and RollCall™.

All serial connections are via 9 way D type connectors and all interfaces are bi-directional.

Ports 1 and 2 may be configured as either an RS422 or a RS232 interface. Ports 3 and 4 are RS422 interfaces.



Configure Serial Ports 1 and 2

Serial ports 1 and 2 may be configured as either RS422 or RS232 via the RS232 Mode control in the [Port] section of the configuration file. See *Syntax of the IQSPI Configuration File* for more information.

Pin Connections for 9 way D-Type Connectors

As RS422 (ports 1 to 4)				
IQSPI00			Connect to	Remote Unit
9 Way D-Type Pin Number	Ribbon Cable Strand Number	Description		
1	1	Ground	↔	Ground
6	2	Not Used		
2	3	Received Data A (RxDA)(Rx-ve)	↔	Transmitted Data A (TxDA)(Tx-ve)
7	4	Received Data B (RxDB)(Rx+ve)	↔	Transmitted Data B (TxDB)(Tx+ve)
3	5	Transmitted Data B (TxDB)(Tx+ve)	↔	Received Data B (RxDB)(Rx+ve)
8	6	Transmitted Data A (TxDA)(Tx-ve)	↔	Received Data A (RxDA)(Rx-ve)
4	7	Not Used		
9	8	Not Used		
5	9	Not used		

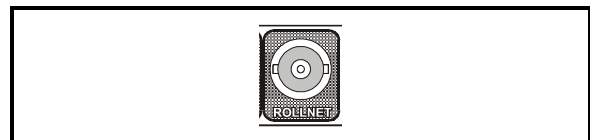
For reference, the A signal is at 0V at line idle, and the B signal at +5V.

As RS232 (only applies to ports 1 and 2 when configured as RS232)				
IQSPI00			Connect to	Remote Unit
9 Way D-Type Pin Number	Ribbon Cable Strand Number	Description		
1	1	Ground	↔	Ground
6	2	Not Used		
2	3	Received Data (Rx D)	↔	Transmitted Data (Tx D)
7	4	+10 V		
3	5	Transmitted Data (Tx D)	↔	Received Data (Rx D)
8	6	Not Used		
4	7	+10 V		
9	8	Not Used		
5	9	Ground	↔	Ground

RollNet

This BNC connector allows the unit to be connected to the RollCall network communications system.

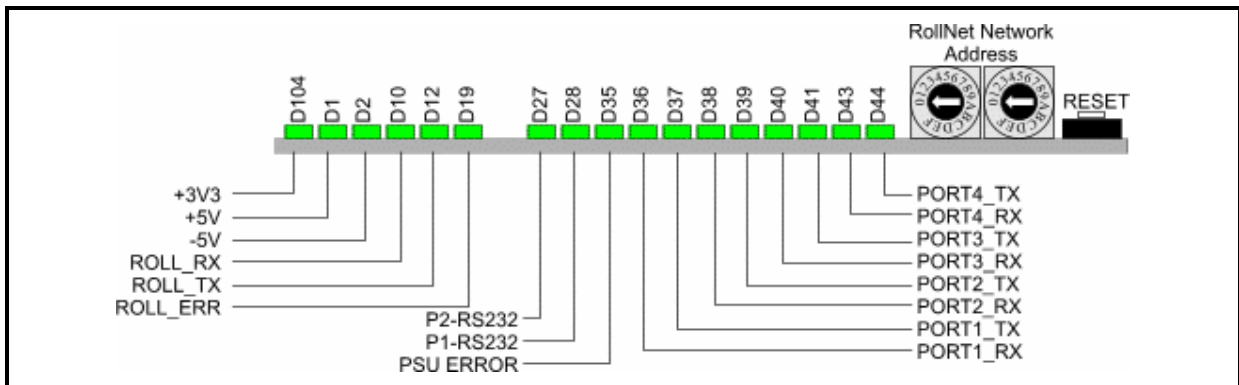
Connect the RollCall system using 75 ohm "T" pieces in a similar manner to a coaxial "Ethernet" system. Both extremities of the system must be terminated in 75 ohms.



Notes:

- RollNet will only be active if it is enabled by means of the UseRollNet control in the [ISQPI] section of the configuration file.
- The coaxial link is bi-directional and therefore must not be passed through signal switching networks.
- In a RollCall™ segment, all units must have different unit address codes.

Card Edge Functions



D104

Indicates that a 3.3V power supply is present.

D1

Indicates that the positive power supply is present.

D2

Indicates that the negative power supply is present.

D10

Indicates that RollNet data is being received.

D12

Indicates that RollNet data is being transmitted.

D19

Indicates that a RollNet error has occurred.

D27

When illuminated, indicates that serial port 2 is configured to be RS422. When not illuminated, indicates that serial port 2 is configured to be RS232,

D28

When illuminated, indicates that serial port 1 is configured to be RS422. When not illuminated, indicates that serial port 1 is configured to be RS232.

D35

Indicates that a PSU Error has occurred.

D36

Indicates that Port 1 is receiving serial data.

D37

Indicates that Port 1 is transmitting serial data.

D38

Indicates that Port 2 is receiving serial data.

D39

Indicates that Port 2 is transmitting serial data.

D40

Indicates that Port 3 is receiving serial data.

D41

Indicates that Port 3 is transmitting serial data.

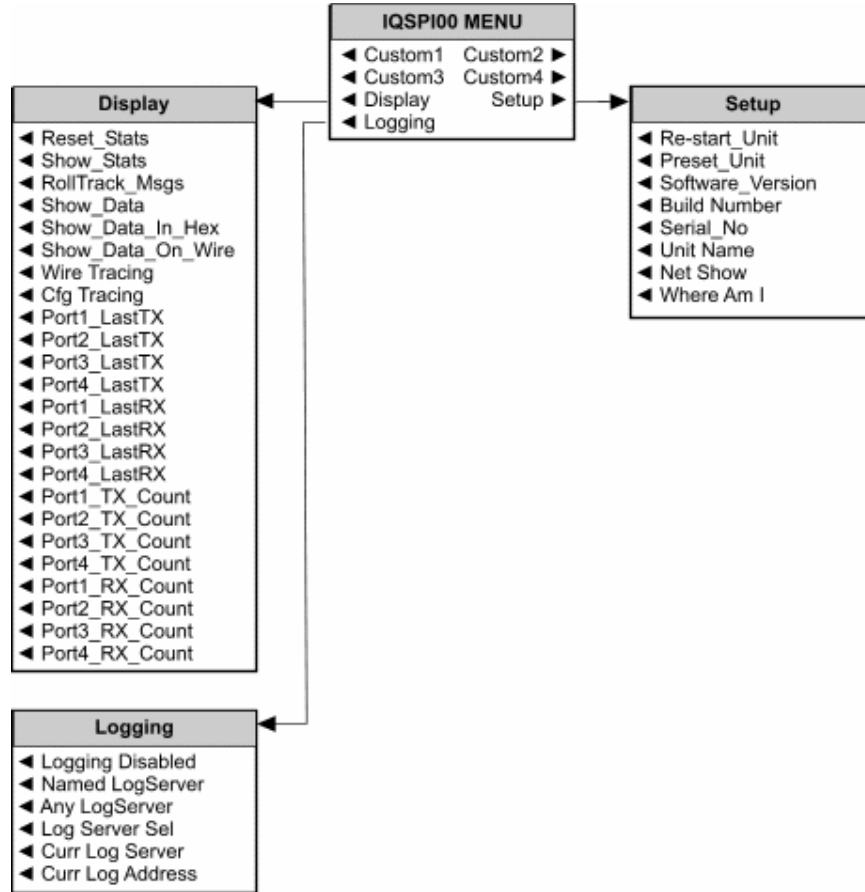
43

Indicates that Port 4 is receiving serial data.

D44

Indicates that Port 4 is transmitting serial data.

IQSPI00 Menu System



Operation from an Active Control Panel

The card may be operated with an active control panel via the RollCall™ network.

The menus available for this card are shown on the previous page and will appear in the Control display window.

Operational details for the remote control panel will be found in SECTION 1 of the Modular System Operator's Manual.

Main Menu

The main menu enables you to select various sub-menus by pressing the button adjacent to the required text line.

Whenever a menu item is selected, the parameters of that selection are displayed in the Information window of the front panel. Where the selection is purely a mode selection and does not enable a sub-menu, the text will become reversed (white-on-black) indicating that the mode is active. If the mode is not available for selection the text will remain normal.

IQSPI00 MENU	
◀ Custom1	Custom2 ▶
◀ Custom3	Custom4 ▶
◀ Display	Setup ▶
◀ Logging	

Custom 1 to Custom 4 Menu Options

These functions reflect the four-customizable serial ports, used to connect to serial devices. They are configured by downloading a configuration file to the unit.

For information about configuring these custom options, see the *IQSPI00 Configuration File Editor* section.

Display Menu

Information about the operational state of the IQSPI00 can be monitored and/or reset here.

The following functions may be selected:

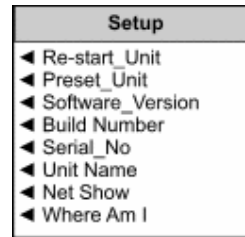
- **Reset_Stats:** Resets statistical data.
- **Show_Stats:** Displays statistical data
- **RollTrack_Msgs:** Selecting this item shows acknowledgement messages from RollTrack units, in the information window and enables RollTrack failure reporting to be displayed.
- **Show_Data:** This enables the display of the most recent send and received data strings on each port.
- **Show_Data_In_Hex:** Controls whether data is shown in hexadecimal or ACSII format.
- **Show_Data_On_Wire:** Shows data strings as they occur on the wire.
- **Wire Tracing:** Traces data strings, in real time, as they appear on the wire, into the circular RAM buffer for debug purposes.
- **Cfg Tracing:** Traces data strings, in real time, as they appear from the configuration file, into the circular RAM buffer for debug purposes.
- **Port1_LastTX to Port4_LastTX:** For each port, displays the most recent messages sent by the IQSPI00.
- **Port1_LastRX to Port4LastRX:** For each port, displays the most recent messages received by the IQSPI00.
- **Port1_TX_Count to Port4_TX_Count:** For each port, displays the count of messages sent by the IQSPI00.
- **Port1_RX_Count to Port4_RX_Count:** For each port, displays the count of messages received by the IQSPI.

Display
◀ Reset_Stats
◀ Show_Stats
◀ RollTrack_Msgs
◀ Show_Data
◀ Show_Data_In_Hex
◀ Show_Data_On_Wire
◀ Wire Tracing
◀ Cfg Tracing
◀ Port1_LastTX
◀ Port2_LastTX
◀ Port3_LastTX
◀ Port4_LastTX
◀ Port1_LastRX
◀ Port2_LastRX
◀ Port3_LastRX
◀ Port4_LastRX
◀ Port1_TX_Count
◀ Port2_TX_Count
◀ Port3_TX_Count
◀ Port4_TX_Count
◀ Port1_RX_Count
◀ Port2_RX_Count
◀ Port3_RX_Count
◀ Port4_RX_Count

Setup Menu

This selection reveals a sub-menu that allows the following functions to be enabled:

- **Re-start_Unit:** This function restarts the unit. It has the same action as a power-up power-down operation.
- **Preset_Unit:** Sets all adjustment functions that include a preset facility, to their preset value. *Note that this is a momentary action and the text will not become reversed.*
- **Software_Version:** Displays the version of software fitted in the module. Select OK to return to the Setup Menu.
- **Build Number:** Displays the build number of the unit. Select OK to return to the Setup menu.
- **Serial_No:** Display the serial number of the unit. Select OK to return to the Setup menu.



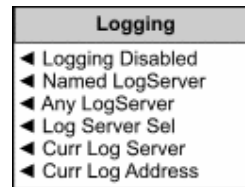
The following options are only available when the UseRollNet control is enabled in the [ISQPI] section of the configuration file.

- **Unit Name:** Displays the unit name. Select Ok to return to the Setup menu.
- **Net Show:** This function allows a unit to be 'hidden' from the network system. When netshow is active the unit broadcasts its presence.
- **Where Am I:** This allows a unit to be physically located in a large system. When this function is selected, LED indicators on the front of the card will flash.

Logging Menu

This menu enables you to configure the logging characteristics of the unit. These options are only available when the UseRollNet control is enabled in the [ISQP] section of the configuration file.

- **Logging Disabled:** This option allows logging to be disabled.
- **Named LogServer:** If this option is selected, logged information will only be sent to the selected log server.
- **Any LogServer:** If this option is selected, log information will be sent to any Logger on the system. *If there is only one log server on the system, it is recommended that you use this option.*
- **Log Server Sel:** This option enables you to select a specific log server.
- **Curr Log Server:** This option displays the name of the current log server.
- **Curr Log Server:** This option displays the address of the current log server.



IQSPI00 Configuration File Editor

Introduction

The IQSPI00 Configuration Editor is designed for use with a Serial Port Interface (SPI) card. The purpose of the program is to provide easy creation and editing of an SPI card's configuration file.

Functions within the IQSPI00 Configuration Editor

Some of the functions of the IQSPI00 Configuration File Editor are as follows:

- Output File View – The complete file within a text editor.
- Port View - A graphical (treeview) representation of how the menus will look if viewed on an IQ Modular control panel.
- Track Parameters – Constructs a RollTrack of a selected command from a device via RollCall.
- Read/Write of SPI configuration file – Transfer of the configuration file between a drive and the SPI card.

Installation

The following pre-installation requirements exist:

- Win95 (with Internet explorer version 4.0 or later), or later.
- One of the suite of RollCall software version 3.3 or later
(Note: If RollCall software is not present then install the free copy of RollCall IQSPCE, which is included with this package).

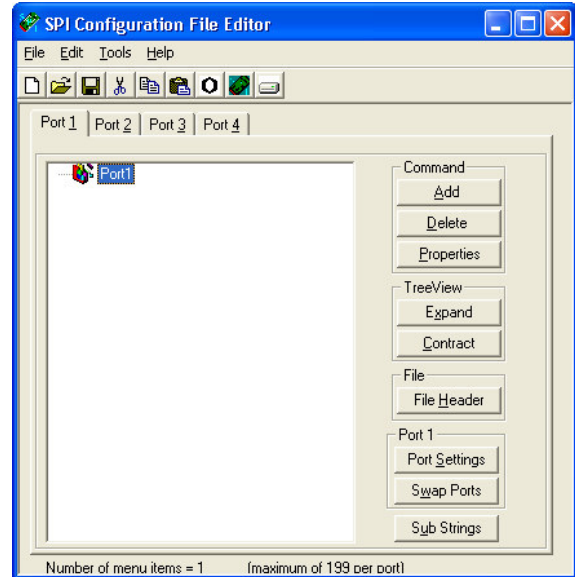
Once the above pre-installation requirements have been met, the IQSPI Configuration Editor may be installed.

Toolbar and Menus

Toolbar

Starting from the left, the icons are:

- **New:** Resets to default contents.
- **Open:** Opens a configuration file.
- **Save:** Saves the current file.
- **Cut:** Cut text to Clipboard (or else Commands if in a treeview).
- **Copy:** Copy text to Clipboard (or else Commands if in a treeview).
- **Paste:** Paste text from Clipboard (or else Commands if in a treeview).
- **Toggle Port View On/Off:** Switches the editing mode between the Output File and Commands within a Port View.
- **Download to SPI:** Transfers current configuration file to selected SPI card.
- **Upload from SPI:** Transfers configuration file from an SPI card to a selected file (path\name.cfg).



File Menu

In addition to the options included in the toolbar, the File menu provides access to the following functions.

- **SaveAs:** Saves the current file under a different name.
- **WriteHardwareFile**
- **Exit:** Terminates the SPI Configuration File Editor.
- A list of the recently opened files is appended to the bottom of this menu. Clicking on one will open that file.

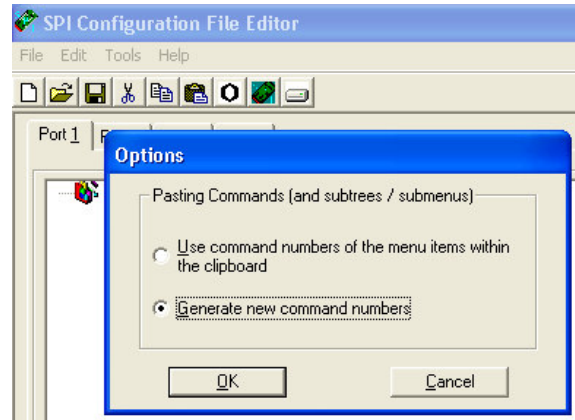
Edit Menu

In addition to the options included in the toolbar, the Edit menu provides access to the following functions.

- **Versions:** Displays the current file versions and allows them to be edited.
- **Output File:** This will switch to editing the actual output file in a text editor. (When in that mode the menu item Ports will appear at the bottom of the Edit menu, selecting that will switch to editing in Port View mode.)

Tools Menu

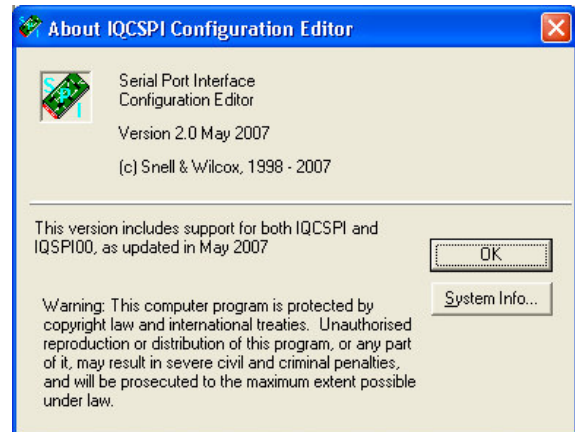
The Tools menu provides access to only one item, Options. The Options window allows the choice of whether to generate new command numbers when performing Cut and Paste of commands within a tree view.



Help Menu

The Help menu provides access to two items:

- Snell & Wilcox Website: If the computer is connected to the Internet, the Snell & Wilcox Website loads when you select this option.
- About SPI Configuration File Editor: Displays information about the Configuration Editor. This contains the version number of your installation of the IQSPI00 Configuration Editor.



Starting a New File

To start a new file, from the File menu, select New. Alternatively, click the New button in the toolbar.

This creates a new file with default contents as shown on the right. You can now proceed to edit the file as required. For more information, see *Editing the Configuration File* on page 19

Note: This new file is held only within the IQSPI00 Configuration Editor application, so it will need to be saved when you close the application, or your changes will be lost.



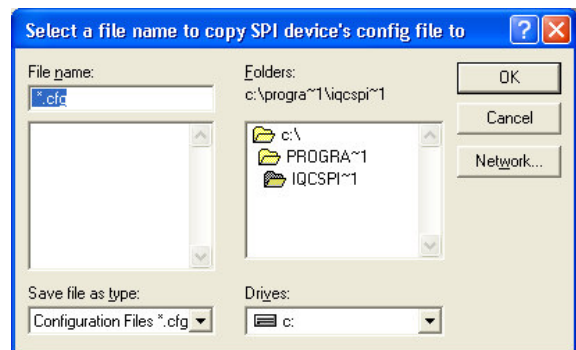
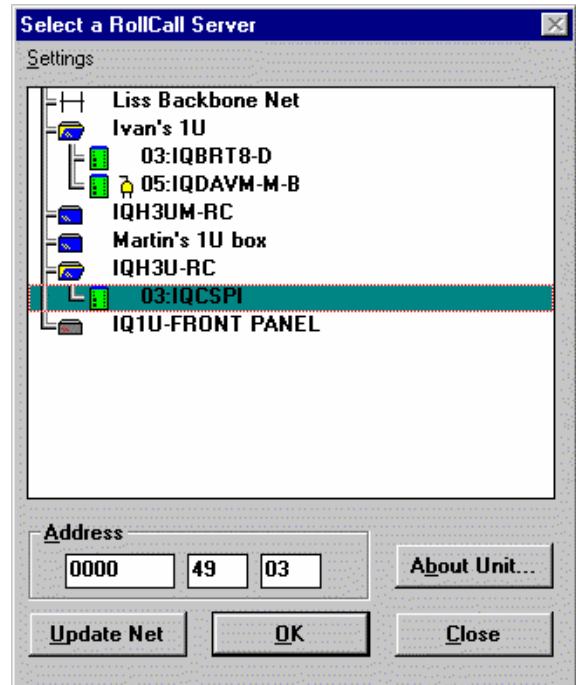
Uploading a File

To edit a file existing within an SPI card, first it must be transferred to a drive that the computer can access.

To upload a file from an SPI card:

1. From the File menu, select Upload from SPI. The Select a RollCall Server window appears, displaying the RollCall network.
2. Select the SPI card containing the file to upload, and click OK.
3. In the window that appears, provide a file name (and path) to copy the SPI configuration file to, and click OK. A wait message appears followed by another indicating if the upload was successful.

The file is now open for you to edit.



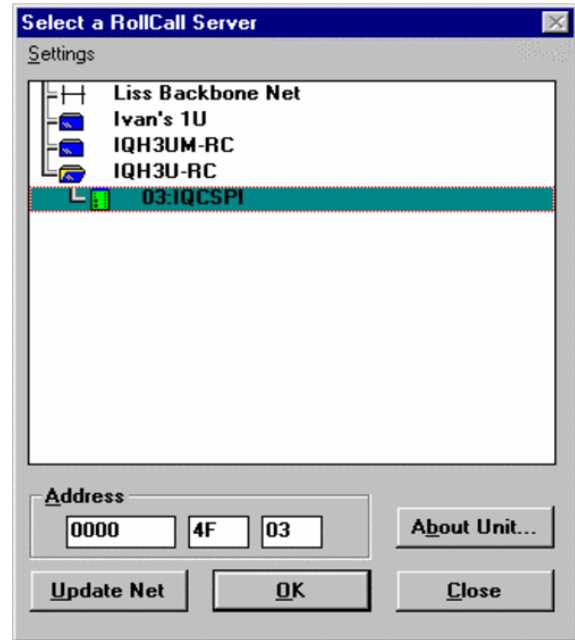
Downloading a File

After the configuration file is completed it can be transferred to an SPI card.

To download a file to an SPI card:

1. From the File menu, select Download to SPI. A window appears showing the RollCall network.
2. Select the SPI card to which you want to download the file, and click OK.

The currently opened file is transferred to the SPI card. After a wait, a message appears indicating if the download was successful. The SPI card now contains this configuration file.

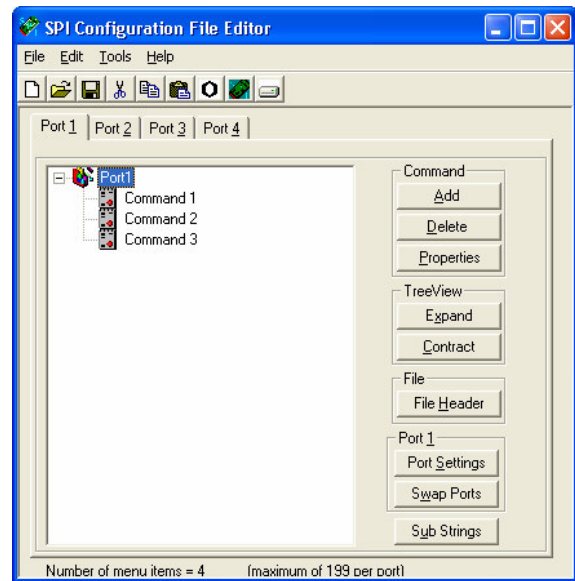


Editing the Configuration File

There are two ways to edit the configuration file. The entire output file may be edited, or more commonly the file is edited using a graphical treeview of the commands (menu) for each port

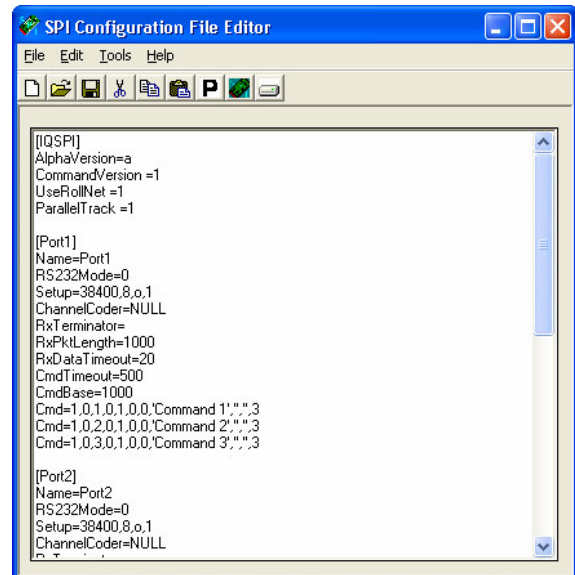
Editing the File in Treeview

When you edit the file in the graphical treeview, the commands for each port are displayed in a tree structure. In this mode, a series of dialogs enable you to create or edit the configuration file.



Editing in Output File mode

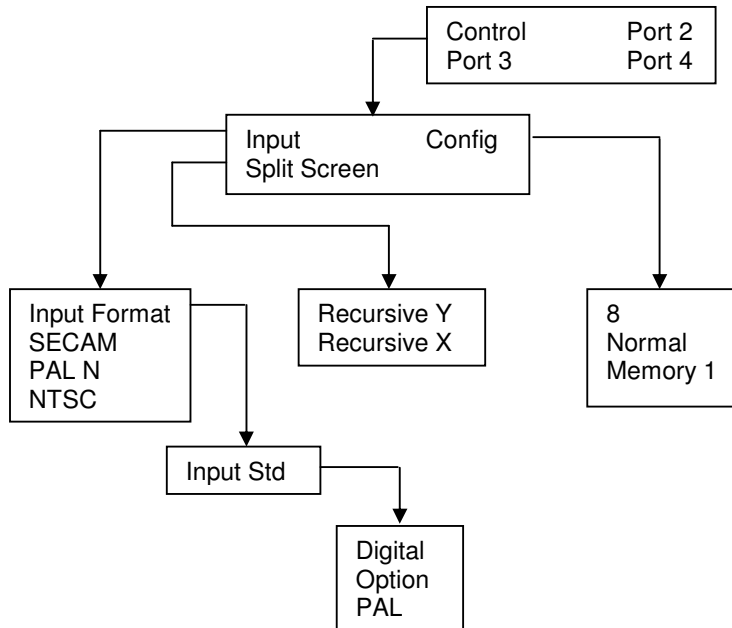
When you edit in output file mode, the file content appears in a text editor. For more information about the syntax of the configuration file, see the *Syntax of the IQSPI Configuration File* section.



Editing Ports

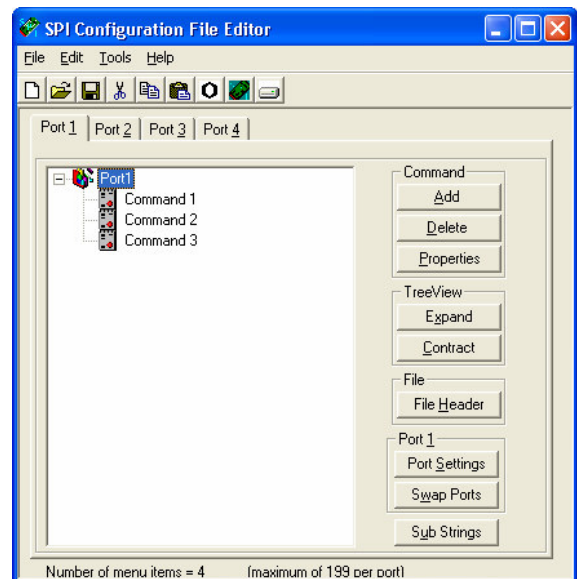
There are four ports, whose commands are shown in a treeview structure.

In this example, Port 1 has been configured, and Ports 2 to 4 have not been. Viewing the SPI card's commands from a front panel would show:



Selecting *Edit, Ports* (or the respective toolbar icon) will produce the following display:

By clicking on a tab labeled Port1, Port2, Port3 or Port4 the respective port's menu is shown:



Adding a Command

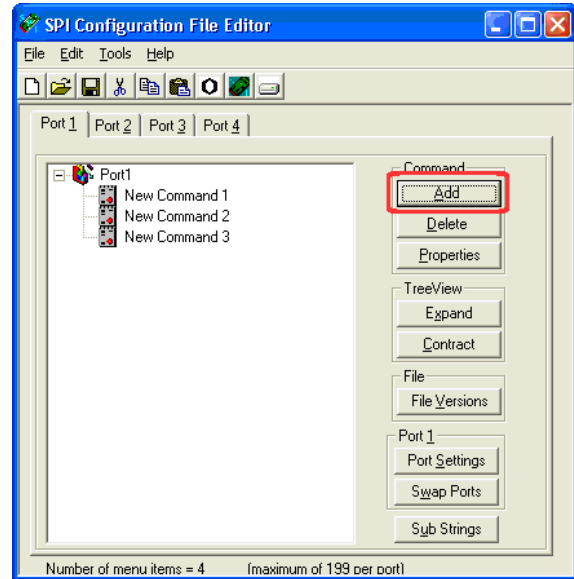
To add a command, select the command you wish to appear above it and then click Add.

Alternatively, right click in the tree view and select Add Menu Item.

A new command appears with a default set of parameters

Notes:

- Only styles of List and Tiled may have children.
- Select the command's Style before amending other properties.



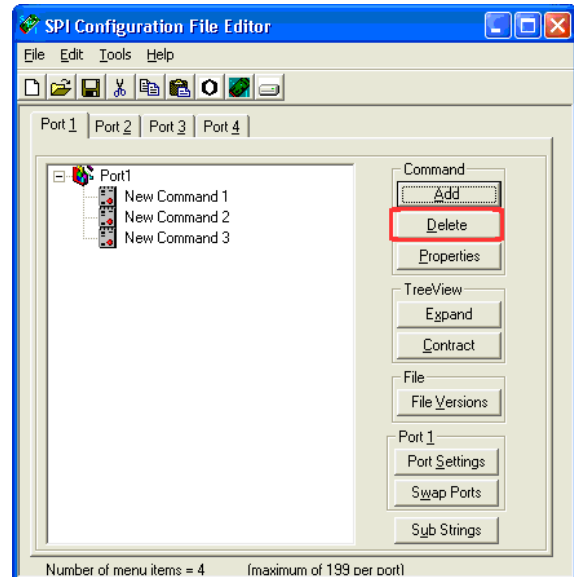
Deleting a Command

To delete a command and all of its children, select the command, and click Delete.

Alternatively, right-click the command and select Remove Menu Item.

If the command (is a list or tiled style and) has children then these will also be deleted.

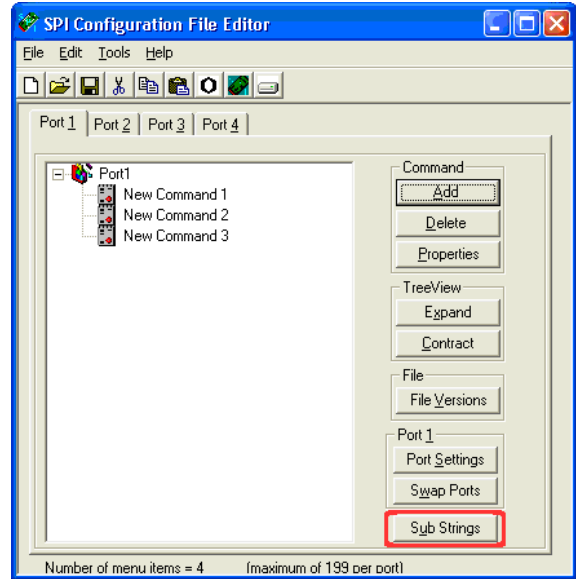
Note: If the command has children, these are also deleted.



Substitution Strings

The IQSPI00 can perform string substitution. That is, insert user-meaningful name strings into a serial message or log message according to the command's numeric value.

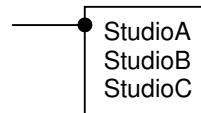
Click Sub Strings to add, edit, or remove substitution strings.



Example

If \String%ld appears within an outgoing payload (or the 'ParamStg' field) the string 'StudioB' will be substituted when the command's numeric value is 1.

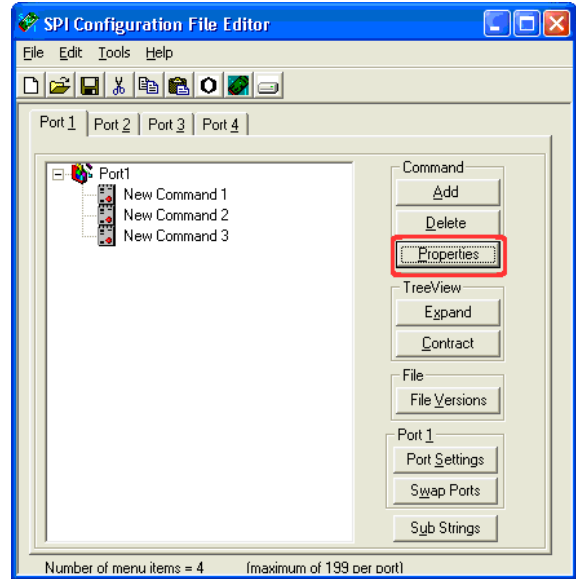
Text of
Substitution
Strings



Editing a Command's Properties

Selecting a command and clicking the Properties button (or double clicking on a command) will display a properties window. If the properties window is shown and a command clicked on then the properties of that command will be displayed.

Once the Properties box is visible, selecting (clicking on) a command within the treeview will allow its properties to be edited.



The Command Tab

On the command tab you can specify Style, Number, and ValueOffset.

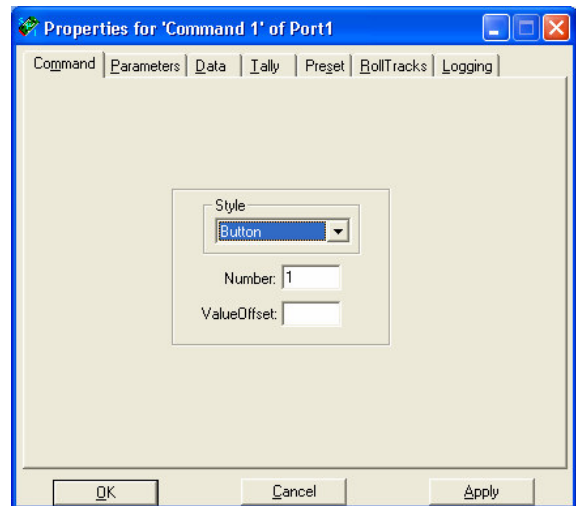
Style

The style can be one of the following: Tiled, List, Display, Button, CheckBox, Number, Vertical Graph, Horizontal Graph, Edit String, Vertical Level, Horizontal Level, Data.

The Tiled or List styles may have children (menu options below this level). The default style for new commands is Tiled.

There are two settings on the Command tab:

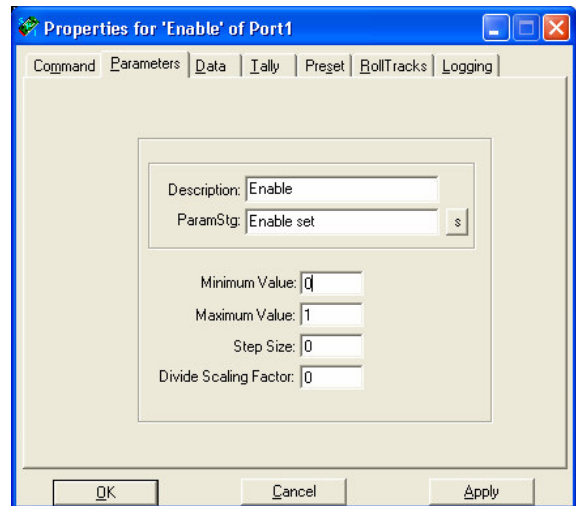
- **Number:** The Number field specifies the RollCall command number of this command.
- **Value Offset:** If a menu item has, for example, RollCall values of min = -20 to max = 20, but the device connected via the IQSPI00's serial link requires values within the range 0 → 40 then this can be achieved by having a ValueOffset = 20.



The Parameters Tab

The available parameters vary according to the style. Depending on the selected style, they include:

- Description
- ParamStg
- Minimum Value
- Maximum Value
- Step Size
- Divide
- Scaling Factor.



The Data Tab

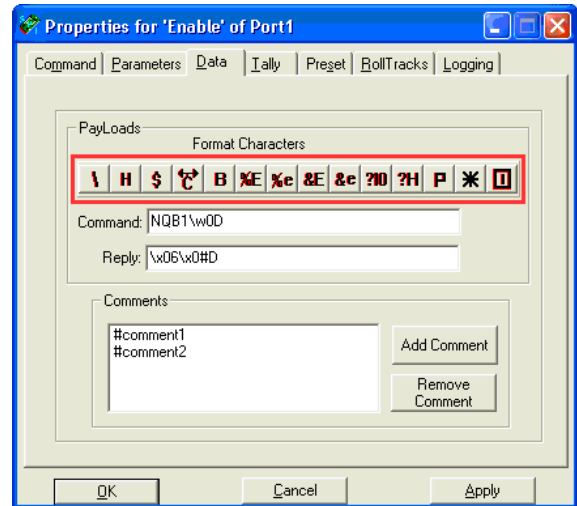
This contains the Command & Reply payloads and the Comments.

There are a number of format characters, each of which will be applied to the active textbox. The buttons shown on the right enable you to insert format characters into the payloads of the Command, Reply, Tally, TallyReply, Request, and RequestReply fields. Outgoing payloads are Command, TallyReply, and Request. Incoming payloads are Reply, Tally, and RequestReply.

When setting something on the device that the SPI card is connected, a Command is sent from the SPI card from the device. The Reply is then sent from the device to the SPI card in response to the Command.



You can add, edit, and delete configuration file comments:

- To add a comment, click Add Comment.
- To edit an existing comment, double click on it.
- To delete a comment, click on it to select it, and then click Remove Comment.



Payload Format Characters		
Button	Character	Description
	Single Back Slash	This button inserts two backslashes, the first of which is ignored. For example: <ul style="list-style-type: none"> • Payload = \\A • Sends = 92 65 (which is the ASCII decimal value of \A)
	Hex Value (Two Characters)	This button inserts \x followed by two Hex characters. For example: <ul style="list-style-type: none"> • Payload = \xFF • Sends = 255 (which is the decimal value of FF)
	String Value	This button inserts \s. The string parameter of the command, up to a maximum of 20 characters is sent. For example: <ul style="list-style-type: none"> • Command's string value = On • Payload = \s • Sends = 79 78 (which is the decimal ASCII value of On)
	Offset Character	This button inserts \c followed by two Hex characters. The Hex characters identify a particular character from within the command's string. For example: <ul style="list-style-type: none"> • Command's String value = EFGHIJKLMNOPQ • Payload = \c0A • Sends = 78 (which is the decimal ASCII value of N)
	Byte Value	This button inserts \vb. The byte value of the commands numeric value is sent. For example: <ul style="list-style-type: none"> • Command's numeric value = 2B 71 FF A3 • Payload = \vb • Sends = 163 (which is the decimal value of A3)

Payload Format Characters		
Button	Character	Description
	Integer (Big Endian)	This button inserts \vI. The integer value of the command's numeric value is sent in big endian format. For example: <ul style="list-style-type: none"> Command's numeric value = 2B 71 FF A3 Payload = \vI Sends = 255 163 (which is the decimal value of FF A3)
	Integer (Little Endian)	This button inserts \vi. The integer value of the command's numeric value is sent in little endian format. For example: <ul style="list-style-type: none"> Command's numeric value = 2B 71 FF A3 Payload = \vi Sends = 163 255 (which is the decimal value of A3 FF)
	Long (Big Endian)	This button inserts \vL. The long value of the command's numeric value is sent in big endian format. For example: <ul style="list-style-type: none"> Command's numeric value = 2B 71 FF A3 Payload = \vL Sends = 43 113 255 163 (which is the decimal value of 2B 71 FF A3)
	Long (Little Endian)	This button inserts \vl. The long value of the command's numeric value is sent in little endian format. For example: <ul style="list-style-type: none"> Command's numeric value = 2B 71 FF A3 Payload = \vl Sends = 163 255 113 43 (which is the decimal value of A3 FF 71 2B)
	Decimal ASCII String	This button inserts \a followed by a number 0-9. The number indicates the minimum number of digits sent from the command's numeric value. For example: <ul style="list-style-type: none"> Command's numeric value = 255 Payload = \a4 Sends the string = '0' '2' '5' '5'
	Hex ASCII String	<i>This button inserts \h followed by a number 0-9. The number indicates the minimum number of digits sent.</i> <i>For example:</i> <ul style="list-style-type: none"> Command's numeric value = FF Payload = \h5 Sends the string = '00255' <p><i>Note that for incoming ASCII payloads of unknown size '\a0' and '\h0' may be used.</i></p>
	Pro-Bel Two's Complement	This button inserts \P. This only applies to outgoing payloads. It calculates the two's complement (for Pre-Bel's SW-P-02 General Switcher Communication Protocol) upon everything which appears to its left within the payload.

Payload Format Characters		
Button	Character	Description
	Wildcard	<p>This button inserts *.</p> <p>This applies to incoming payloads only.</p> <p>For example, ****/x0D within a payload will match any four characters followed by /x0D.</p>
	Substitution Strings	<p>This button inserts \String%ld\$.</p> <p>This may apply to outgoing payloads, log messages and the 'ParamStg' field.</p> <p>Where this exists it will be substituted with a string corresponding to the commands numeric value</p>
<p>All other characters are sent as their decimal ASCII equivalent.</p> <p>For example:</p> <ul style="list-style-type: none"> • Payload = BRT7 • Sends = 66 82 84 55 <p>Note: The string value and numeric value of a command is set by a front panel or by RollCall control panel or by an incoming stream on the serial port.</p>		

The Tally Tab

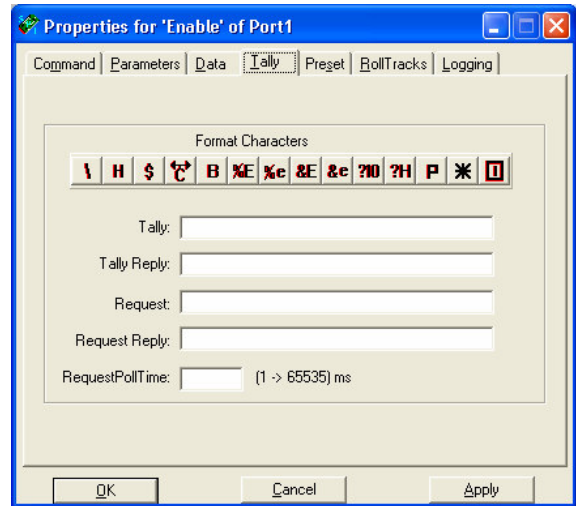
This contains the payloads for the Tally and the Tally Reply.

For information on the Format Characters buttons, see the section on the Data tab on page 24.

The SPI card sends a Request to the device to which it is connected to obtain the current status of the device. A RequestReply is then sent from the device to the SPI card in response to the Request.

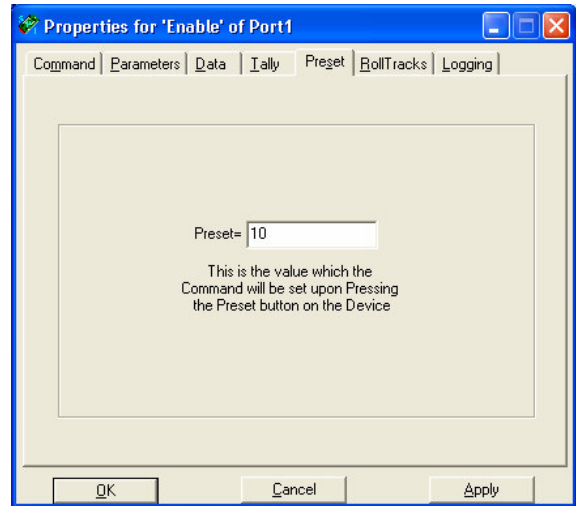
A Tally is initiated by the device and sent to the SPI card. If for example the device is a VCR and the play button is pressed on it, then the device will send a Tally to the SPI card (indicating that it is now playing). The TallyReply is sent from the SPI card to the device in response to the Tally.

Request Poll Time is the time (in milliseconds) between each Request payload being sent.



The Preset Tab

This contains the value, which Number and String style commands will be set to upon pressing the preset button on the SPI card.



The Tracks Tab

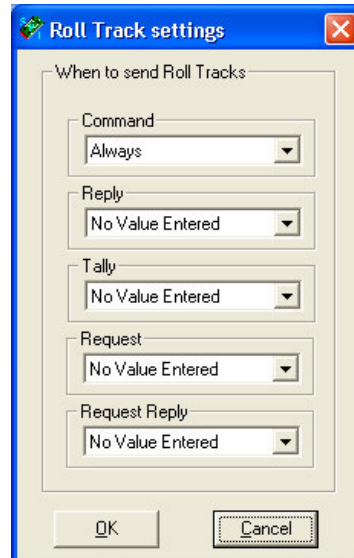
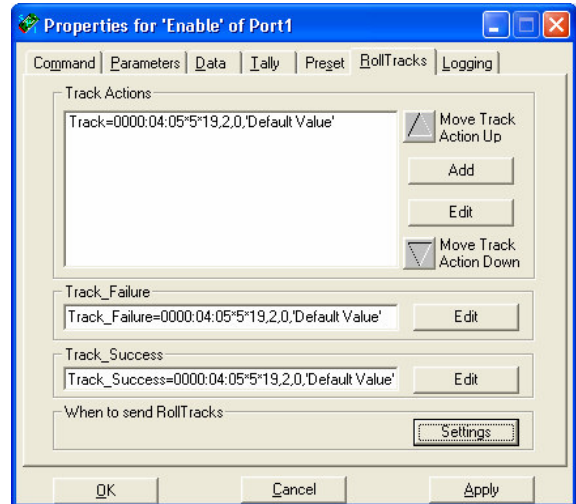
This contains the Track Actions and the Track confirmations.

The order in which the Track Actions appear represent the order in which they will be executed. To amend the order, select a Track Action and then click on either the Move Track Action Up button or the Move Track Action Down button (these will move the selected Track Action one position up or down).

This view displays the Tracks Actions and confirmation (Track_Failure, Track_Success). To remove Track Actions, select one and press the Delete key on the keyboard. To remove Track_Failure or Track_Success go to the respective text box and delete its contents.

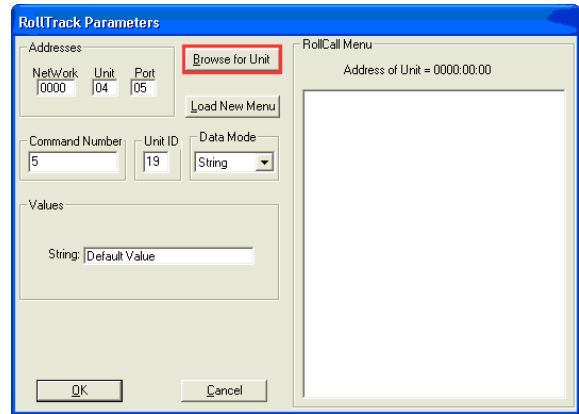
Clicking on the Edit or Add button opens a Track Parameter window, where each track is constructed.

For each command, you can select when RollTracks will be sent.



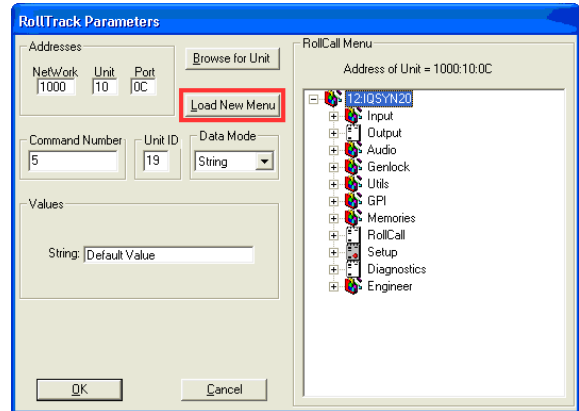
Track Parameter window

Clicking the Browse for Unit button allows the selection of a RollCall device on the network.



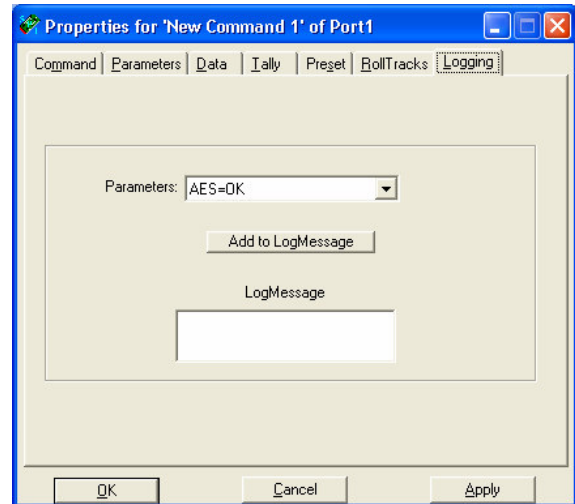
Once a device is selected, clicking on the Load New Menu button will load the selected device's menu. Selecting a command from within the treeview representation of the menu loads the command's parameters. Upon selecting OK a RollTrack is created and placed in the respective area on the Tracks Tab.

Upon clicking the Load New Menu button, there will be a pause while the menu loads. If it is not possible to load a menu then a message will appear to this effect.



The Logging Tab

This contains the LogMessage. There are a number of default messages, which can be selected and then edited.



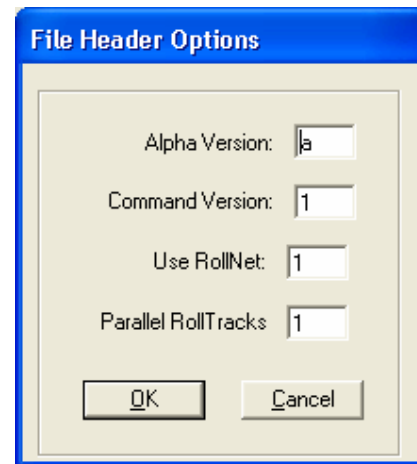
File Header

To edit the configuration file header information, click File Header.

Alternatively, from the Edit menu, select Versions.

In the File Header Options dialog, you can specify the following information:

- **Alpha Version:** Defines the RollCall alphanumeric version.
- **Command Version:** Defines the RollCall command set version number.
- **Use RollNet:** Defines whether the card should use a direct RollNet connection.
- **Parallel RollTracks:** Defines whether the card should send RollTracks in parallel whenever possible.



Note: The File Header Options dialog box can only be accessed when editing in Port View mode.

Port Settings

To specify the settings for each port, click on the tab for the port, and then click Port Settings.

Name

The name used in menus for the port.

Interface

Defines whether the port operates in RS232 or RS422 mode.

Comms Settings

The settings in this section enable you to specify the port's baud rate, parity, data bits, and stop bits settings.

Protocol Settings

- Channel Coder. Enables you to select which of the built-in channel coders (if any) should be used for serial communications on this port.
- Receive Terminators. Enables you to specify up to four characters to be interpreted as end of message markers. Specify the characters as comma separated values.
- Receive Packet Length. The module declares a message complete when the number of characters specified in this field is received.
- Receive Char Time Out (ms). Defines the timeout period, in milliseconds, for received serial characters.
- Command Time Out (ms). Defines how long the module waits for a reply to commands sent over the serial port.
- Base Command Number. Defines the base number to be used for RollCall commands associated with this port.

Port 1

Port Name:

Interface
 RS232 RS422

Comms Settings
 Baud Rate:

Parity
 Even Odd None

Data Bits
 7 8

Stop Bits
 1 2

Protocol Settings
 Channel Coder:

Receive Terminator(s)

Receive Packet Length

Receive Char Time Out (ms)

Command Time Out (ms)

Base Command Number

Swapping Ports

To exchange the commands and settings of two ports:

1. Click Swap Ports.
2. In the dialog that appears, select the ports to exchange.
3. Click OK.

**Cut, Copy & Paste**

When using the tree structure representation of each port cut, copy and paste work as follows:

- Cut & Copy will place a copy of the selected command (and its sub menu) in the clipboard. Cut will also remove the selected command.
- Paste will place a copy of the command (which is held in the clipboard) underneath the selected command.

Prompt to Save

If you have made changes to the configuration file, you will be prompted to save when you:

- Create a new file.
- Open a new file.
- Upload a file to the SPI.
- Download a file from the SPI.
- Exit the Configuration File Editor.

Communication Between IQSPI and a Remote Device

Transaction Types

There are three basic transaction types supported by the SPI.

By analogy with RollCall, transactions initiated by the SPI may be considered the Front Channel, whereas transactions initiated by the controlled device, for example, a router, may be considered the Back Channel. Both Front and Back Channel transactions are carried via the one serial link.

Command and Reply

This is a Front Channel transaction, where the SPI sends a status request to the device, and the device replies. This is analogous to an SP_SETPARAM, and is triggered by the SPI receiving an SP_SETPARAM over the RollCall network. The SPI would normally be configured to wait and check for the correct reply.

Request and RequestReply

This is a Front Channel transaction, where the SPI sends a status request to the device and the device replies. The SPI can be configured to poll the device at regular intervals to ensure that the SPI (and hence the RollCall network) always reflects the current state of the controlled device (this is important for devices that do not have the concept of a back channel update).

Tally and TallyReply

This is a Back Channel transaction, where the controlled device spontaneously sends a Tally to the SPI, and the SPI (optionally) replies. Typically, the SPI will issue one or more RollTracks to other modules, so that the RollCall network remains in step with the external device state.

Transaction Type Example

As a simple example, consider an SPI card connected to a VCR:

Command and Reply

- If a RollCall control is activated on the SPI card to control the VCR, a Command is sent from the SPI card to the VCR.
- A Reply is sent from the VCR to the SPI card in response to the Command.

Request and RequestReply

- For the SPI card to get the current status of the VCR, a Request is sent from the SPI card to the VCR every 100 ms.
- A RequestReply is sent from the VCR to the SPI card in response to the Request.

Tally and TallyReply

- If a button (for example, the Play button) is pressed on the VCR, then a Tally is sent from the VCR to the SPI card, which updates the relevant RollCall control.
- A TallyReply is sent from the SPI card to the VCR in response to the Tally.

Syntax of the IQSPI Configuration File

An IQSPI configuration file consists of:

- [IQSPI] file header
- [PortN] sections – Each port section (1-4) comprises a series of Command definitions for each Port. Most of the configuration file consists of this set of Command definitions, which together define the operation of the unit.
- [Strings] section

To designate a configuration line as a comment, prefix it with a hash “#” symbol.

Where settings are optional, the default value shown below will be used if the setting is absent from the configuration file.

[IQSPI] File Header

Every configuration file must start with an [IQSPI] file header section, which must be on the first line of the file.

The file header contains settings that apply to the whole product, as follow:

AlphaVersion	Defines the RollCall alphanumeric version (a single ASCII character). The default setting is <code>AlphaVersion=a</code>
CommandVersion	Defines the RollCall command set version number (an unassigned number)

between 0 and 255).

The default setting is `CommandVersion=1`

UseRollNet

Defines whether card should use direct RollNet connection (IQSPI00 only).

The default setting is `UseRollNet=1`

ParallelTrack

Defines whether the card should send RollTracks in parallel whenever possible (IQSPI00 only).

The default setting is `ParallelTrack=1`

[Port N] Sections

The configuration file is split into four sections, one section for each port. Within each section are all the commands and settings belonging to that port.

Name

This is the name used in the menus for the port.

For example, `Name=Port1`

RS232Mode

Defines whether the port operates in RS232 or RS422.

The default setting is `RS232Mode=0`

On the IQSPI00 (type ID 477):

- Ports 1 and 2 can be set to operate in either RS232 or RS422.
- Ports 3 and 4 can only operate in RS422.

On the IQCSPI (type ID 127):

- Port 1 can be set to operate in RS232 or RS422 via links on the board.
- Ports 2, 3, and 4 can only operate in RS422.

Setup

Defines the serial port settings, in the form

`Setup = BaudRate, DataBits, Parity, StopBits`

The default setting is `Setup=38400,8,o,1`

- The baud rate is selected from a predefined list: 600, 1200, 2400, 4800, 9600, 19200, 38400 bps. The IQSPI00 also supports 57600 and 115200 bps.
- The number of Data Bits is selected from a predefined list: 5, 6, 7, 8.
- The Parity can be set to Odd "o", Even "e" Or None "n".
- This can be set to either 1 or 2.

ChannelCoder

Defines which of the built-in channel coders should be used for serial communications on this port.

The default is `ChannelCode=NULL`

The function of a channel coder is to perform data link level processing of serial messages for those protocols that cannot be accommodated by the simple string manipulation rules provided as part of the normal command definitions (see below).

Typically, the types of operation that require a channel coder are:

- Computing checksums.
- Implementing character escape sequences.
- Handling non-standard bit-packing of values.

The set of built-in channel coders currently consists of:

- Stagetec Nexus protocol.
- GVG Series 7000 Native Protocol.
- Probel SW-P-88 General Switcher. Communication Protocol.
- Probel SW-P-08 General Remote Control Communication Protocol.
- Null coder (transparent).

RxTerminator	<p>This can be used to define a set of up to four characters that will be interpreted as end of message markers. The characters are specified as comma separated, decimal ASCII values. For example, <code>RxTerminator=65,66</code> would set 'A' and 'B' as terminator characters.</p> <p>By default, no characters are interpreted as <code>RxTerminator</code>.</p>
RxPktLength	<p>This optional control applies to the IQSPI00 only. It can be used to configure the module to declare a received message as complete whenever <code>RxPktLength</code> characters have been received. In other words, this defines the maximum number of characters expected in any received message.</p> <p>This mechanism can be used alongside <code>RxDataTimeout</code>, <code>RxTerminator</code>, or both.</p> <p>The <code>RxPktLength</code> control is particularly useful for protocols where all (or the majority) of the messages are of the same fixed length, since these will be recognized as complete as soon as the last character arrives.</p> <p>The default setting is <code>RxPktLength=1000</code>.</p>
RxDataTimeout	<p>Defines the timeout period in milliseconds for received serial characters. That is, for the gap between characters. This can be used to detect that a complete message has been received and should be processed.</p> <p>The timeout mechanism can be used in parallel with the <code>RxPktLength</code> and <code>RxTerminator</code> mechanisms. In other words, a message will be considered complete if any of the conditions is triggered.</p> <p>The <code>RxDataTimeout</code> value for a particular interface should be chosen such that the gap between characters within a message never exceeds the value. At the same time, choosing too large a value will result in extra delay between the actual message being sent and the module starting to process it (unless message completion is detected via packet length).</p> <p>The default setting is <code>RxDataTimeout=20</code>.</p>
CmdTimeout	<p>Defines how long the module waits for a reply to commands sent over the serial port, with the delay set in milliseconds.</p> <p>The default setting is <code>CmdTimeout=500</code>.</p>
CmdBase	<p>This defines the base number to be used for RollCall commands associated with this port. The usual convention (and default) is to allocate <code>CmdBase</code> as follows:</p> <ul style="list-style-type: none">• Port 1 has <code>CmdBase=1000</code>• Port 2 has <code>CmdBase=2000</code>• Port 3 has <code>CmdBase=3000</code>• Port 4 has <code>CmdBase=4000</code> <p>The RollCall command number for a particular command is defined by adding the <code>RCCmd</code> value for that <code>Cmd</code> definition (see below) to the <code>CmdBase</code>.</p>

Command Definitions

Command definitions comprise most of the configuration file. Each command definition is associated with one port, and consists of several optional parameters.

Each command definition line in the configuration file starts with `Cmd=` and continues until the next `Cmd=` line.

Although referred to here as SPI configuration file Commands, each `Cmd=` line actually signifies the creation of a new Menu item in the RollCall menu set.

The list of Command parameters is as follows. All except `Cmd` are optional.

- `Cmd`
- `Reply`
- `Tally`
- `TallyReply`

- Request
- RequestReply
- RequestPollTime
- Preset
- ValueMultiplier
- ValueDivisor
- ValueOffset
- LogMessage
- Track
- Track_Delay
- Track_Agent
- Track_Success
- Track_Failure
- MultiMatch

Cmd

This parameter denotes the start of a new Command definition, and consists of the following comma separated fields:

Cmd=Dir, Style, Num, Min, Max, Step, Div, Txt, ParamStg, PayLoad, RollTrackCtrl

For example, `Cmd=1, 48, 1, 1, 0, 0, 0, 'PLAY', '', '\x20\x01\x21', 3`

Typically, each `Cmd` line is used to define the serial string to send to the remote device when a particular RollCall command is changed. However, `Cmd` lines can also be used simply to add appropriate menu lines into the product menu set, for example, to group together related controls into a page. In this case, the serial payload is not used.

Text is enclosed within single quotes. Any string field that is not used in a command must have two single quotes placed at its position within the `Cmd` statement.

This definition can be broken down as follows:

- **Dir** defines the direction of this command (always set to 1).
- The set of fields “**Style, Num, Min, Max, Step, Div, Txt, ParamStg**” effectively comprises a standard RollCall menu line definition (a `FUNC_STR` structure).
- **PayLoad** defines the serial string corresponding to this command.
- **RollTrackCtrl** defines when RollTracks defined for this command (if any) are sent.

Taking each field individually, the usage is as follows:

- **Dir**: Should be set to 1
- **Style**: The style is one of a set of predefined values, corresponding to the different RollCall menu styles as follows:

0	Tiled	96	Vgraph
16	List	112	Hgraph
32	Display	128	EditString
48	Button	144	Vlevel
64	Checkbox	160	Hlevel - horizontal level meter. No control
80	Number	192	Data – binary data block

- **Num:** This is the number given to this command. This will be added to the Port `CmdBase` number to form the final RollCall command number.
- **Min:** This number is defined as a signed long integer. It represents the Minimum value allowed. It is not required for all menu styles.
- **Max:** This number is defined as a signed long integer. It represents the Maximum value allowed. It is not required for all menu styles.
- **Step:** This is a number, which represents increment/decrement steps. For List and Tiled styles, the step value is equal to the total number of descendants (i.e. total number of commands which appear under the list, not just the immediate children)
- **Div:** This is a number that represents the divide-scaling factor that is applied to the command's numeric value for display. It is not required for all menu styles.
- **Txt:** This string represents the command text, which appears as the label in the menus
- **ParamStg:** This string is the printf string for parameter display.
- **Payload:** This string defines the serial message (if any) that is sent from the SPI card to the device when the Command is activated.
- **RollTrackCtrl:** This field controls the sending of RollTracks associated with this command (if any), as follows
 - 0 = Never
 - 1 or 5 = On Change
 - 3 or 7 = Always

The usage of each of these fields varies by Style, as shown in this table:

(R=Required, N=Not Used)

Style	Dir	Style	RCCmd	Min	Max	Step	Div	Txt	ParamStg
Tiled	R	R	R	N	N	R	N	R	R
List	R	R	R	N	N	R	N	R	R
Display	R	R	R	N	N	N	N	R	R
Button	R	R	R	R	N	N	N	R	R
CheckBox	R	R	R	R	N	N	N	R	R
Number	R	R	R	R	R	R	R	R	R
Vgraph	R	R	R	R	R	R	N	R	N
Hgraph	R	R	R	R	R	R	N	R	N
EditString	R	R	R	R	R	N	N	R	R
Vlevel	R	R	R	R	R	R	N	R	N
Hlevel	R	R	R	R	R	R	N	R	N

Payload Settings

A number of special format characters and escape sequences are supported in the payload string controls. There is a slight variation according to the message type, as follows:

- **Tx:** messages sent by IQSPI (i.e. Command, TallyReply, Request)
- **Rx:** messages received by IQSPI (i.e. Reply, Tally, RequestReply)

Token in Payload String	Resultant String Sent / Matched	Tx	Rx
\\	Single Back Slash	✓	✓

<code>\x</code> (followed by 2 hex chars)	Unsigned Hex Value Two Characters	✓	✓
<code>\s</code>	String Value (current RollCall string value)	✓	*
<code>\c</code> (followed by 2 hex chars)	Offset Character (character at specified index into current RollCall string value)	✓	✓
<code>\vb</code>	8-bit Signed Byte Value	✓	✓
<code>\vI</code>	16-bit Signed Integer – Big Endian	✓	✓
<code>\vi</code>	16-bit Signed Integer – Little Endian	✓	✓
<code>\vL</code>	32-bit Signed Long – Big Endian	✓	✓
<code>\vI</code>	32-bit Signed Long – Little Endian	✓	✓
<code>\a</code> (followed by single decimal digit N)	Decimal ASCII String. For N = 1 to 9, this is a string of up to N decimal ASCII chars. For N=0 (on Rx only), this matches up to the first non-decimal ASCII character	✓	✓
<code>\h</code> (followed by single decimal digit N)	Hex ASCII String. For N = 1 to 9, this is a string of up to N hex ASCII chars. For N=0 (on Rx only), this matches up to the first non-hex ASCII character.	✓	✓
<code>\\$SubStr\$</code> or <code>\\$SubStr%ld\$</code>	String lookup from [Strings] section. SubStr is used as sprintf format string to generate key string for string lookup.	✓	*
<code>\P</code> (all to the left of P)	Pro-Bel Checksum (as per General Switcher Protocol SW-P-02)	✓	*
*	WildCard (matches any single character)	*	✓
!	Matches zero or more characters. Used to terminate further processing and declare incoming string successfully matched.	*	✓
Any other character	Literal Character	✓	✓

Reply

This is an optional control that can be used to define the expected response from the remote device when the relevant `Cmd` is sent. For most – but not all - device configurations, checking for the correct reply is useful.

If present, there can be only one `Reply` control per `Cmd`, and it should usually be on the line immediately following the `Cmd` statement.

The `Reply` control has the format:

```
Reply=Dir:Payload, RollTrackCtrl
```

Taking each field individually, the usage is as follows:

- **Dir**: Should be set to 0
- **Payload**: This string (enclosed within single quotes) defines the serial message that the IQSPI expects to receive from the device in response to the command being sent.
- **RollTrackCtrl**: This optional field controls the sending of RollTracks associated with this reply being received, as follows:
 - 0 = Never
 - 1 or 5 = On Change
 - 3 or 7 = Always

Tally

This is an optional control that can be used to define an unsolicited serial message that may be received from the remote device, for example, as a result of some external action such as a crosspoint switch.

The `Tally` control has the format:

```
Tally=Payload, RollTrackCtrl
```

Taking each field individually, the usage is as follows:

- **Payload**: This string (enclosed within single quotes) defines the unsolicited

serial message that the IQSPI may receive from the device.

- **RollTrackCtrl**: This optional field controls the sending of RollTracks associated with this Tally being received, as follows:
 - 0 = Never
 - 1 or 5 = On Change
 - 3 or 7 = Always

On the IQSPI00 only, multiple `Tally` (and appropriate `TallyReply`) controls can be defined for each command.

TallyReply

This optional control is used to define the serial message the IQSPI should send in response to a received `Tally` message. Hence, a `TallyReply` makes sense only where a `Tally` has already been defined.

There can be only one `TallyReply` control per `Tally`, and it should usually be on the line immediately following the `Tally` statement. Where a `Cmd` has multiple `Tally` statements, then any `TallyReply` statement is assumed to relate to the `Tally` statement that immediately precedes it.

The `TallyReply` statement has the format:

```
TallyReply=PayloadString, RollTrackCtrl
```

Taking each field individually, the usage is as follows:

- **Payload**: This string (enclosed within single quotes) defines the serial message that the IQSPI will send when it receives the relevant tally from the device.
- **RollTrackCtrl**: This optional field controls the sending of RollTracks associated with this `TallyReply` being sent as follows:
 - 0 = Never
 - 3 or 7 = Always

Request

This is an optional control, which can be used in conjunction with a `RequestPollTime`, to instruct the IQSPI to send regular status polling messages to the remote device. There can be only one `Request` control per command.

The `Request` control has the format :

```
Request=Payload, RollTrackCtrl
```

Taking each field individually, the usage is as follows:

- **Payload**: This string (enclosed within single quotes) defines the serial message that the IQSPI will send to the remote device every `RequestPollTime` milliseconds.
- **RollTrackCtrl**: This optional field controls the sending of RollTracks associated with the `Request` being sent as follows:
 - 0 = Never
 - 3 or 7 = Always

RequestReply

This is an optional control, which can be used to define the expected response from the remote device when the `Request` message has been sent. If present, the `RequestReply` should usually be on the line immediately following the `Request` statement.

The `RequestReply` control has the format:

```
RequestReply=Payload, RollTrackCtrl
```

Taking each field individually, the usage is as follows:

- **Payload**: This string (enclosed within single quotes) defines the serial message that the IQSPI expects to receive from the remote device in response to the `Request`.
- **RollTrackCtrl**: This optional field controls the sending of RollTracks associated with the `RequestReply` being received as follows:
 - 0 = Never
 - 1 or 5 = On Change

3 or 7 = Always

RequestPollTime	<p>This control sets the polling time in milliseconds for the relevant <code>Request</code> message.</p> <p>If no <code>RequestPollTime</code> is set, then the <code>Request</code> will never be sent.</p>
Preset	<p>Each command can have an optional <code>Preset</code> statement, which serves two purposes:</p> <ul style="list-style-type: none"> • At power-up, the value of the <code>RollCall</code> command is set to this <code>Preset</code> value. • When set to <code>Preset</code> via <code>RollCall</code>, the command's value is set to this preset value. <p>For example, <code>Preset=5</code></p>
ValueMultiplier	<p>The IQSPI can perform a simple numeric scaling conversion between the <code>RollCall</code> command value and the value sent or received in the serial strings.</p> <p>The conversion from <code>RollCall</code> to serial string value is as follows: $\text{SerialStringValue} = ((\text{RollCallValue} / \text{ValueDivisor}) * \text{ValueMultiplier}) + \text{ValueOffset}$</p> <p>The conversion from serial string to <code>RollCall</code> value is as follows: $\text{RollCallValue} = ((\text{SerialStringValue} - \text{ValueOffset}) / \text{ValueMultiplier}) * \text{ValueDivisor}$</p> <p>This conversion is performed in terms of signed 32-bit integers, and in the order set out above (which may be significant in some cases at the limits of the range).</p> <p><code>ValueMultiplier</code> is interpreted as a signed 16-bit integer, with default value of 1.</p>
ValueDivisor	<p>See description of <code>ValueMultiplier</code>.</p> <p><code>ValueDivisor</code> is interpreted as a signed 16-bit integer, with default value of 1.</p>
ValueOffset	<p>See description of <code>ValueMultiplier</code>.</p> <p><code>ValueOffset</code> is interpreted as a signed 32-bit integer, with default value of 0.</p>
LogMessage	<p>A log message can be associated with each <code>Cmd</code> definition. This log message will then be sent whenever the control value changes, whether as a result of a <code>RollCall SP_SETPARAM</code> message or an incoming <code>Tally</code> message on the serial interface.</p> <p>The log message is specified using the same format as the serial payload field for <code>Cmd</code> or <code>Request</code> controls. Only one log message can be defined for each <code>Cmd</code>.</p>
Track	<p>This control can be used to define one or more <code>RollTracks</code> to be sent for this command, when the specified events occur (see the <code>RollTrackCtrl</code> fields for the various message types above).</p> <p>Any number of <code>Track</code> entries can appear for each command; each has the format, <code>Track=nnnn:uu:pp*cmd*id,model,value1,string1,offset1,mode2,value2,string2,offset2</code></p> <p>This breaks down as follows:</p> <ul style="list-style-type: none"> • nnnn:uu:pp: defines the <code>RollCall</code> address of the target destination • cmd*id: defines the command number and type ID at the target destination • model,value1,string1,offset1: defines the control value to be sent, effectively filling in the fields of a <code>RollCall FUNCSTATUS_STR</code> • mode2,value2,string2,offset2: optionally defines the control value expected in reply (where this differs from the value sent) <p>Taking each field individually, the usage is as follows:</p> <ul style="list-style-type: none"> • nnnn: <code>RollCall</code> network address, consisting of four hex digits. For units on the same <code>RollNet</code> segment, this is simply 0000.

- **uu**: RollCall unit address, consisting of two hex digits. For example, hex switches in an IQ frame gateway card.
- **pp**: RollCall port address, consisting of two hex digits. For example, physical slot number in an IQ frame.
- **cmd**: RollCall command number in decimal for the particular function/device.
- **id**: RollCall unit ID in decimal, as found in the product manual or RollCall "About unit" box.
- **mode1**: RollCall data mode to be sent, (1=numeric value, 2=string value, 4=Data, 16=Preset). The usage of `value1` and `string1` depends on the setting of `mode1` as follows:
 - If value, then numeric `value1` contains the current value for the command.
 - If string, then string value `string1` contains a string.
 - If data, then data follows within the string value `string1`, with the `LengthOfData` value set in `value1`.
 - If preset, then the device being addressed will set the menu item identified by the command number to its preset / default value.
- **value1**: RollCall numeric value in decimal to be sent, depending on `mode1` as above.
- **string1**: RollCall string value field to be sent, depending on `mode1` as above.
- **offset1**: Numeric offset value (optional) in decimal, added to RollCall value prior to sending in RollTrack.

Where the response value is expected to differ from the sent value, the following optional controls can be appended:

- **mode2**: RollCall data mode expected in reply. Setting this to the value 65536 instructs the IQSPI to accept any value response as acceptable.
- **value2**: RollCall numeric value field expected in reply, depending on `mode1` as above.
- **string2**: RollCall string value field expected in reply, depending on `mode1` as above.
- **offset2**: Numeric offset value (optional), added to RollCall value prior to comparing with received RollTrack response value.

There are in effect three options for RollTrack reply checking:

- **specify mode 2 = 65536**: check that a reply is received but allow any value
- **specify mode 2 etc**: check that reply matches specified return value in every respect
- **omit mode 2 etc**: check that reply matches value sent in every respect

If `Track_Success` and/or `Track_Failure` are defined, then the results of the RollTrack reply checking (across the complete set of RollTracks specified for a Command) are used to send the appropriate final RollTrack (as defined by `Track_Success` and/or `Track_Failure`).

Some examples of valid forms of the Track line are:

- `Track=nnnn:uu:pp*cmd*id,mode1,value1,string1`
This sends the value formed by `mode1`, `value1`, and `string1` to command `cmd` on the unit `nnnn:uu:pp` with unit type `id`, and expects to receive a reply with the same data mode and value.
- `Track=nnnn:uu:pp*cmd*id,mode1,value1,string1,offset1`
This sends the value formed by `mode1`, `value1+ offset1` and `string1` to command `cmd` on the unit `nnnn:uu:pp` with unit type `id`, and expects to receive a reply with the same data mode and value.
- `Track=nnnn:uu:pp*cmd*id,mode1,value1,string1,offset1,65536`
This sends the value formed by `mode1`, `value1+ offset1` and `string1` to command `cmd` on the unit `nnnn:uu:pp` with unit type `id`. It expects to receive a reply, but ignores the value.

- Track=nnnn:uu:pp*cmd*id,mode1,value1,string1,offset1,mode2,value2,string2,offset2

This sends the value formed by mode1, value1 and string1 to command cmd on the unit nnnn:uu:pp with unit type id, and expects to receive a reply with data mode mode2 and value specified by value2, string2 and offset2.

The IQSPI00 supports all of the available addressing modes for a modular product. For RollTracks addressed outside the frame, the physical interface chosen will also depend on whether RollNet is enabled (see UseRollNet control in File Header section).

The set of addressing modes supported is as follows (where the SPI itself is in slot “mm” within a frame located at address “gg”, “ss” means any other slot, and “uu” means any other frame address):

RollTrack Addressing Modes

Address	Description	Comments on message transmission
FFFF:00:00	Full loopback	Completely local to IQSPI – message never leaves card. Physical interface in message trace shown as Loopback.
0000:00:mm	Relative, to self	Completely local to IQSPI – message never leaves card. Physical interface in message trace shown as I2C. Would be better to use Full Loopback instead.
0000:00:ss	Relative, to other module in same frame	Transfers over I2c via gateway to other module and back.
0000:00:00	Relative, to this frame's gateway	Transfers over I2c to own gateway.
0000:gg:00	Absolute, to this frame's gateway	If RollNet enabled, then sent via RollNet to own gateway. Otherwise, sent via I2c. May make sense if RollNet enabled, for speed.
0000:gg:mm	Absolute, to self	If RollNet enabled, then sent via RollNet to own gateway, and thence to self. Otherwise, sent via I2c to own gateway and back to self. Would be better to use Full Loopback instead.
0000:gg:ss	Absolute, to other module in same frame	If RollNet enabled, then sent via RollNet to own gateway, and hence to other module. Otherwise, sent via I2c to own gateway and on to other module. May make sense if RollNet enabled, for speed.
nnnn:uu:pp	Absolute, to some other unit	If RollNet enabled, then sent via RollNet to other unit. Otherwise, sent via I2c to own gateway and hence out on RollNet to other unit.

Track Delay

This optional control applies for the IQSPI00 only, and can be used to delay the execution of the various RollTracks defined for the command by a set number of milliseconds. This can be useful if precise co-ordination of automation system and RollCall network is required.

The delay specified in `Track_Delay` occurs whenever the RollTracks have

	been triggered prior to any RollTracks being sent. For example, due to RollCall control or serial Tally being received.
Track_Agent	<p>This optional control applies for the IQSPI00 only, and allows the user to maximize the performance of the RollTrack sending, in conjunction with the <code>ParallelTrack</code> control. If present, the <code>Track_Agent</code> value must be in the range 1 to 5.</p> <p>Assuming that parallel RollTrack sending is enabled via <code>ParallelTrack=1</code>, then the IQSPI00 will attempt to execute the list of RollTracks associated with a particular command as quickly and efficiently as possible.</p> <p>Within the same command:</p> <ul style="list-style-type: none">• Multiple RollTracks to different targets will be sent in parallel.• Multiple RollTracks to the same unit will be sent in series (waiting for a response to one before sending the next). <p>By default, when multiple commands are triggered, the IQSPI00 will process these separate commands in series. But by specifying different <code>Track_Agents</code> for different commands, the user can achieve parallel operation across commands as well as for the RollTracks within an individual command.</p> <p>To ensure consistent response, the user needs to ensure that all commands with RollTracks to a particular target destination are assigned to the same RollTrack agent; otherwise that target might end up with multiple RollTracks in-flight to it at the same time, breaking the end-to-end acknowledgement rule that is fundamental to RollCall.</p>
Track_Success	<p>This optional control can be used to indicate whether the set of RollTracks for a particular command were sent successfully or not, by sending a final RollTrack on success as defined by <code>Track_Success</code>.</p> <p>The format of the RollTrack definition is the same as for <code>Track</code> (see above).</p>
Track_Failure	<p>This optional control can be used to indicate whether the set of RollTracks for a particular command were sent successfully or not, by sending a final RollTrack on failure as defined by <code>Track_Failure</code>.</p> <p>The format of the RollTrack definition is the same as for <code>Track</code> (see above).</p>
MultiMatch	<p>This optional control applies to the IQSPI00 only, and controls the behavior of Tally matching when the IQSPI00 receives an unsolicited serial string.</p> <p>On the IQSPI00, it is possible to define multiple Tally strings per command, and of course there are usually multiple command definitions per port.</p> <p>By default, the IQSPI00 will search its complete set of Tally strings (in the order that they appear in the configuration file), and will stop searching as soon as it finds a match. In other words, a received serial string will match to at most one Tally string.</p> <p>The <code>MultiMatch</code> control can be used to change this behavior on an individual command basis as follows. If the IQSPI00 matches the incoming serial string with (one of) the Tally strings for a command and the <code>MultiMatch</code> control for this command is set to 1, then the IQSPI00 will not stop searching, but will continue to look for further matches starting with the next command.</p> <p>This allows a single received serial string to be matched to multiple command lines.</p>

[Strings] Section

This section provides support for string substitution by the IQSPI.

Each line is of the format: `KeyStr=FullString`

Typically, this is used to associate meaningful user names to router source numbers. The individual command payload strings refer to key strings. For example, `src1`, `src2`, etc..., which are then looked up to user names, for example, `OffAir` or `Preview`. These user names then appear in the serial control packets or log messages.

The ISQPI00 Configuration Editor inserts logs strings of the form `String1`, `String2`, `String3`, etc... However, the file format and the ISQPI00 itself support key strings in any format.

