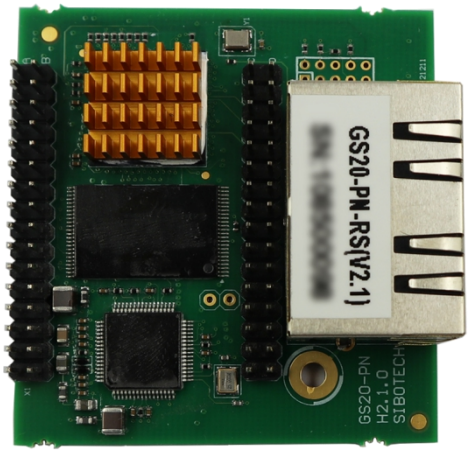
Embedded PROFINET IO RT Interface Module
GS20-PN-RS

User Manual

V2.1

Rev B



SST Automation

E-mail: SUPPORT@SSTCOMM.COM
WWW.SSTCOMM.COM

Important Information

Warning

The data and examples in this manual cannot be copied without authorization. SSTCOMM reserves the right to upgrade the product without notifying users.

The product has many applications. The users must make sure that all operations and results are in accordance with the safety of relevant fields, and the safety includes laws, rules, codes and standards.

Copyright

Trademark

SSTCOMM is the registered trade mark of SST Automation.

Technical Support Contact Information

www.sstcomm.com

E-mail: support@sstcomm.com

Catalog

1 Product Overview

1.1 Product Function

The GS20-PN-RS provides the PROFINET slave interface for developing products PROFINET communication through UART.

1.2 Product Features

- Quick developing: Without the details of PROFINET protocol or PROFINET development system, you can develop the PROFINET product quickly and easily, just edit the configuration files and GSDML files according to actual applications.
- Easy to use: The host processor can read / write input / output data to the GS20-PN-RS through serial interface (UART) , and the GS20-PN-RS will automatically convert data into PROFINET data and transfer data to PROFINET PLC.
- Typical Application: It can be widely used in various products, such as frequency converter, motor start protection device, intelligent high and low voltage electrical apparatus, electric quantity measuring device, various transmitters, intelligent field measuring equipment and meters, etc.

1.3 Technical Specifications

[1] Supports PROFINET IO RT V2.2 Protocol.

[2] PROFINET input-byte and output-byte number are configurable, up to 384 input bytes and384 output bytes.

[3] Supports serial interface (UART).

[4] GS20-PN-RS to baseboard interface is a CMOS TTL asynchronous serial port, baud rate: Fixed to 115.2K, data bit: 8 bit, even check, 1 stop bit.

[5] Supports even check of characters and CRC check of frames to ensure data security.

[6] Supports TCP and UDP.

[7] Power: 3.3VDC, 500mA (suggested 2W / 3.3VDC).

[8] Operation temperature: -4℉~122℉ (-20℃~50℃), Humidity: 5%~ 95% (non-condensing).

[9] External dimension (W*H*D):1.85 in*0.67 in*2.22 in (47mm*17mm*56.3mm).

1.4 Revision History

Revision	Date	Chapter	Description
V2.1, Rev A	6/22/2015	All	First release V2.1 manual
V2.1, Rev B	10/26/2020	All	Revision

2 Product Development Process

Start

⇩

Step 1: Hardware Design

According to the chapter 3 *Hardware Description* to complete the hardware schematic diagram and PCB diagram design.

⇩

Step 2: Software Design

According to the chapter 4 *Software Description* to program the configuration for the GS20-PN-RS and the baseboard.

⇩

Step 3: Data Communication Debugging

Preliminary debugging, mainly to make sure the UART communication between the baseboard and the GS20-PN-RS is normal. The debug information refers to the user manual chapter 4 *Software Description*.

⇩

Step 4: Edit the GSDML File

Edit the GS20-PN-RS.GSDML file according to the configuration of the GS20-PN-RS. Please see "GS20-PN-RS Design Guide" for details (downloaded at www.sstcomm.com).

⇩

Step 5: Design and Set up a Debugging System

⇩

Step 6: Install, Configure and Debug the Product

⇩

END

3 Hardware Description

3.1 Hardware Dimensions

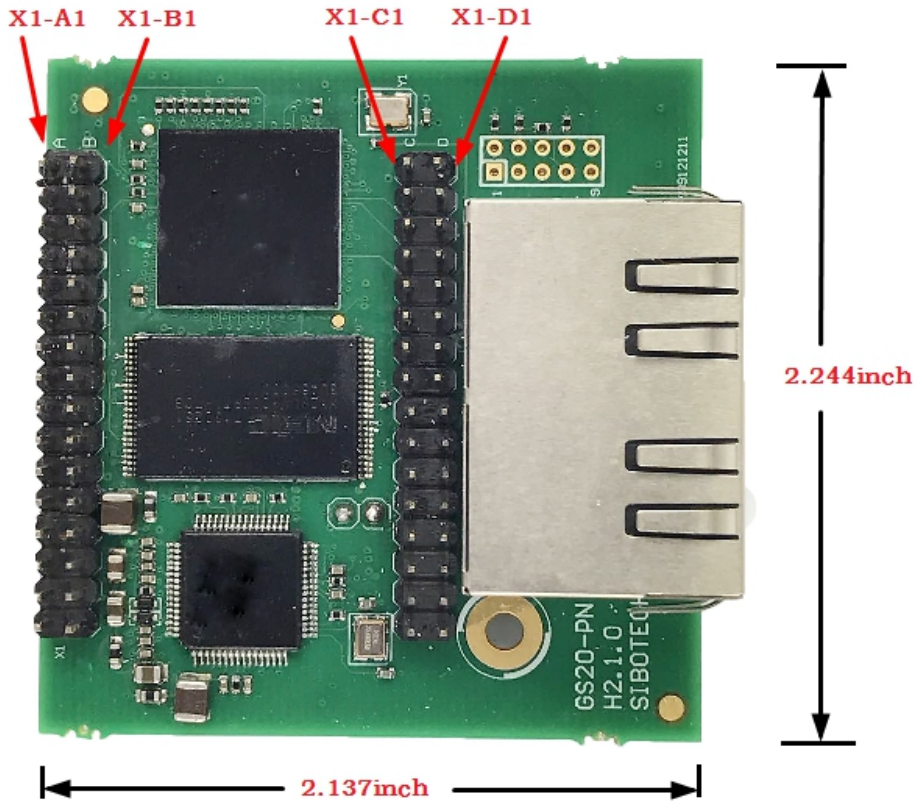


Figure 1 - GS20-PN-RS Dimensions (Plane View)

Notes: Length: 2.137 inch (54.3mm). Width: 2.244 inch (57mm).

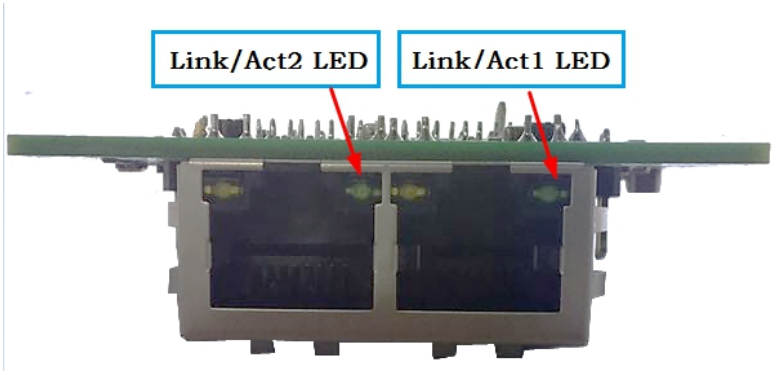


Figure 2 - Ethernet Interfaces of GS20-PN-RS

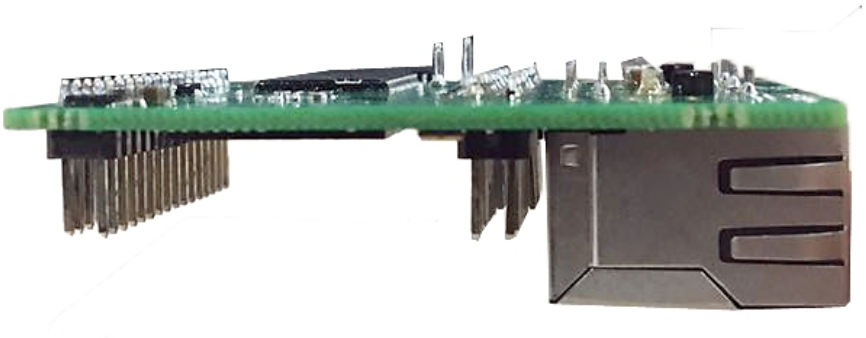


Figure 3 - GS20-PN-RS Side View

3.2 Socket Dimensions of Network Interface Module

The following dimensions are in inch.

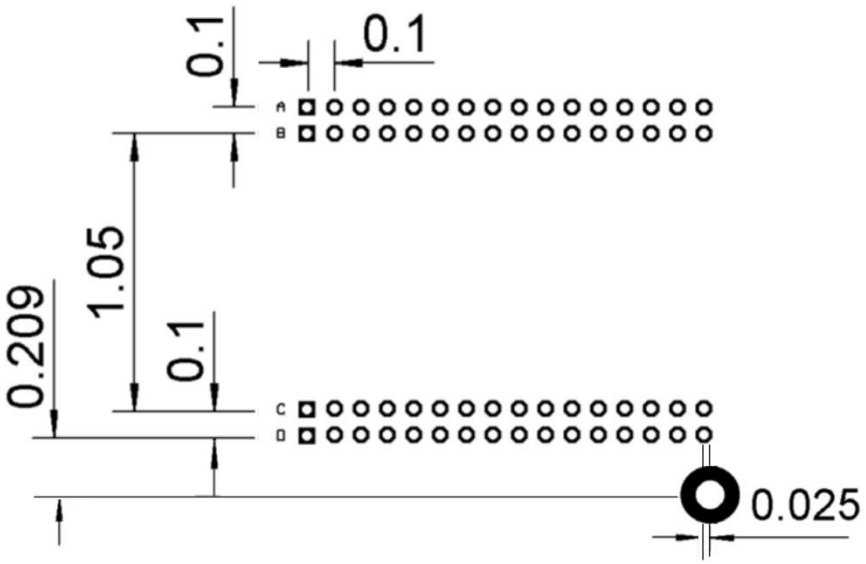


Figure 4 - Socket Dimensions of the Network Interface Module

3.3 Connector Pin Definitions

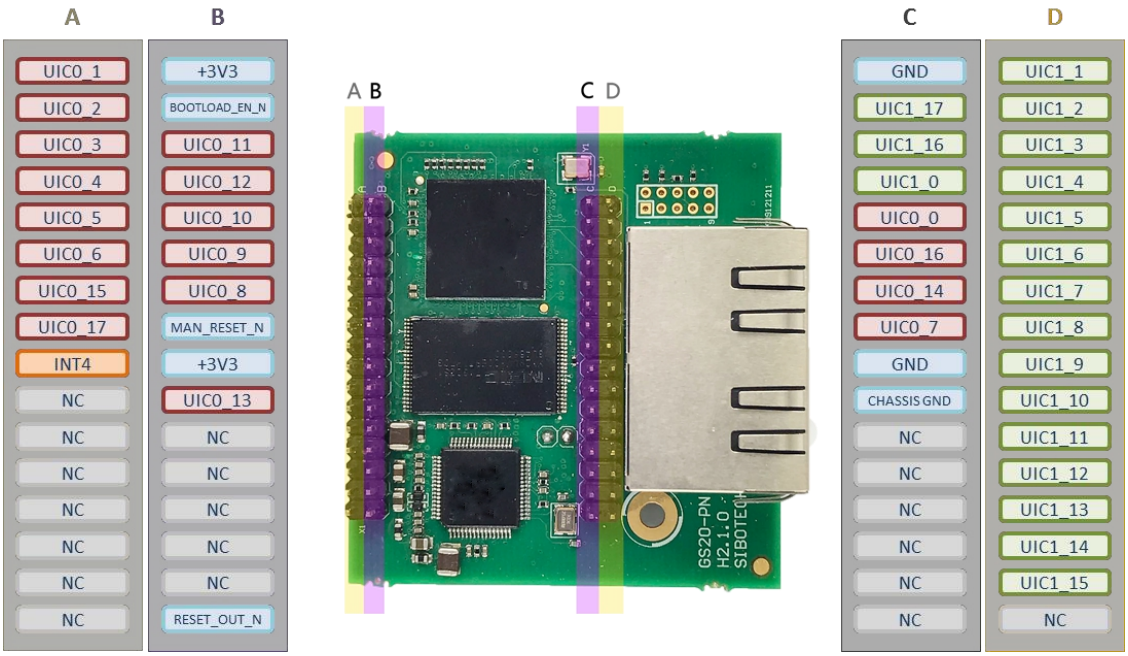


Figure 5 - Connector Pin of the Network Interface Module

Notes: The GS20-PN-RS supports UART Host Interfaces only.

Table 1 - Connector Pin Definitions of X1-A

X1 Pin	Signal Name	Description	Function
A1	UIC0_1	Input	Write Data Bit 1 for Parallel Host Interface
A2	UIC0_2	Input	Write Data Bit 2 for Parallel Host Interface
A3	UIC0_3	Input	Write Data Bit 3 for Parallel Host Interface
A4	UIC0_4	Input	Write Data Bit 4 for Parallel Host Interface
A5	UIC0_5	Input	Write Data Bit 5 for Parallel Host Interface
A6	UIC0_6	Input	Write Data Bit 6 for Parallel Host Interface
A7	UIC0_15	Input	Write Data Bit 15 for Parallel Host Interface
A8	UIC0_17	Output	“INT4” connection for Parallel Host Interface
A9	INT4	Input	Interrupt 4, has 10K pull-up resistor on module; “USED” for Parallel Host Interface
A10	NC	Not connected	Do not connect
A11	NC	Not connected	Do not connect
A12	NC	Not connected	Spare pin
A13	NC	Not connected	Spare pin
A14	NC	Not connected	Spare pin
A15	NC	Not connected	Spare pin
A16	NC	Not connected	Spare pin

Table 2 - Connector Pin Definitions of X1-B

X1 Pin	Signal Name	Description	Function
B1	+3V3	Power	+3.3V Power Supply Input
B2	BOOTLOAD_EN_N	Input	Bootload Enable (active low), has 10K pull-up resistor on module
B3	UIC0_11	Input	Write Data Bit 11 for Parallel Host Interface
B4	UIC0_12	Input	Write Data Bit 12 for Parallel Host Interface
B5	UIC0_10	Input	Write Data Bit 10 for Parallel Host Interface
B6	UIC0_9	Input	Write Data Bit 9 for Parallel Host Interface
B7	UIC0_8	Input	Write Data Bit 8 for Parallel Host Interface
B8	MAN_RESET_N	Input	Manual Reset Input to FIDO's supervisor IC (active low); **Not 5V Tolerant**
B9	+3V3	Power	+3.3V Power Supply Input, **must be connected to the same Power with X1-B1**
B10	UIC0_13	Input	Write Data Bit 13 for Parallel Host Interface
B11	NC	Not connected	Do not connect
B12	NC	Not connected	Do not connect
B13	NC	Not connected	Do not connect
B14	NC	Not connected	Do not connect
B15	NC	Not connected	Do not connect
B16	RESET_OUT_N	Output	FIDO's reset output

Table 3 - Connector Pin Definitions of X1-C

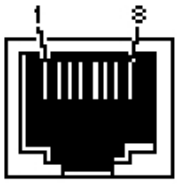
X1 Pin	Signal name	Description	Function
C1	GND	GND	Signal Ground
C2	UIC1_17	Output	"FIFONOTEMPTY_N" for Parallel Host Interface
C3	UIC1_16	Input	TX for UART Host Interface (Connect to Host RX); Or "RD_SEL_N" for Parallel Host Interface
C4	UIC1_0	Output	RX for UART Host Interface (Connect to Host TX); Or Read Data Bit 0 for Parallel Host Interface
C5	UIC0_0	Input	Write Data Bit 0 for Parallel Host Interface
C6	UIC0_16	Input	"WR_SEL_N" for Parallel Host Interface
C7	UIC0_14	Output	Write Data Bit 14 for Parallel Host Interface
C8	UIC0_7	Output	Write Data Bit 7 for Parallel Host Interface
C9	GND	GND	Signal Ground
C10	CHASSIS GND	Chassis GND	Ethernet line-side isolated ground
C11	NC	Not connected	Do not connect
C12	NC	Not connected	Do not connect
C13	NC	Not connected	Do not connect
C14	NC	Not connected	Do not connect
C15	NC	Not connected	Do not connect
C16	NC	Not connected	Spare pin

Table 4 - Connector Pin Definitions of X1-D

X1 Pin	Signal name	Description	Function
D1	UIC1_1	Output	Read Data Bit 1 for Parallel Host Interface
D2	UIC1_2	Output	Read Data Bit 2 for Parallel Host Interface
D3	UIC1_3	Output	Read Data Bit 3 for Parallel Host Interface
D4	UIC1_4	Output	Read Data Bit 4 for Parallel Host Interface
D5	UIC1_5	Output	Read Data Bit 5 for Parallel Host Interface
D6	UIC1_6	Output	Read Data Bit 6 for Parallel Host Interface
D7	UIC1_7	Output	Read Data Bit 7 for Parallel Host Interface
D8	UIC1_8	Output	Read Data Bit 8 for Parallel Host Interface
D9	UIC1_9	Output	Read Data Bit 9 for Parallel Host Interface
D10	UIC1_10	Output	Read Data Bit 10 for Parallel Host Interface
D11	UIC1_11	Output	Read Data Bit 11 for Parallel Host Interface
D12	UIC1_12	Output	Read Data Bit 12 for Parallel Host Interface
D13	UIC1_13	Output	Read Data Bit 13 for Parallel Host Interface
D14	UIC1_14	Output	Read Data Bit 14 for Parallel Host Interface
D15	UIC1_15	Output	Read Data Bit 15 for Parallel Host Interface
D16	NC	Not connected	Spare pin

3.3 Ethernet Interface

Each Ethernet interface of the GS20-PN-RS is a RJ-45 connector.



RJ-45 port

Figure 6 - RJ-45 Connector

Table 5 - Pin Definitions of RJ-45 Port

Pin	Signal Description
S1	TXD+, Transceiver Data+, Output
S2	TXD-, Transceiver Data-, Output
S3	RXD+, Receive Data+, Input
S4	Bi-directional Data+
S5	Bi-directional Data-
S6	RXD-, Receive Data-, Input
S7	Bi-directional Data+
S8	Bi-directional Data-

4 Software Description

4.1 UART Connection

The UART Host Interface is implemented using two pins from the Network Interface. These pins are X1-C3 and X1-C4.

The UART will be configured to operate at a fixed baud rate of 115200 baud using an 8 bit data word with even parity and 1 stop bit.

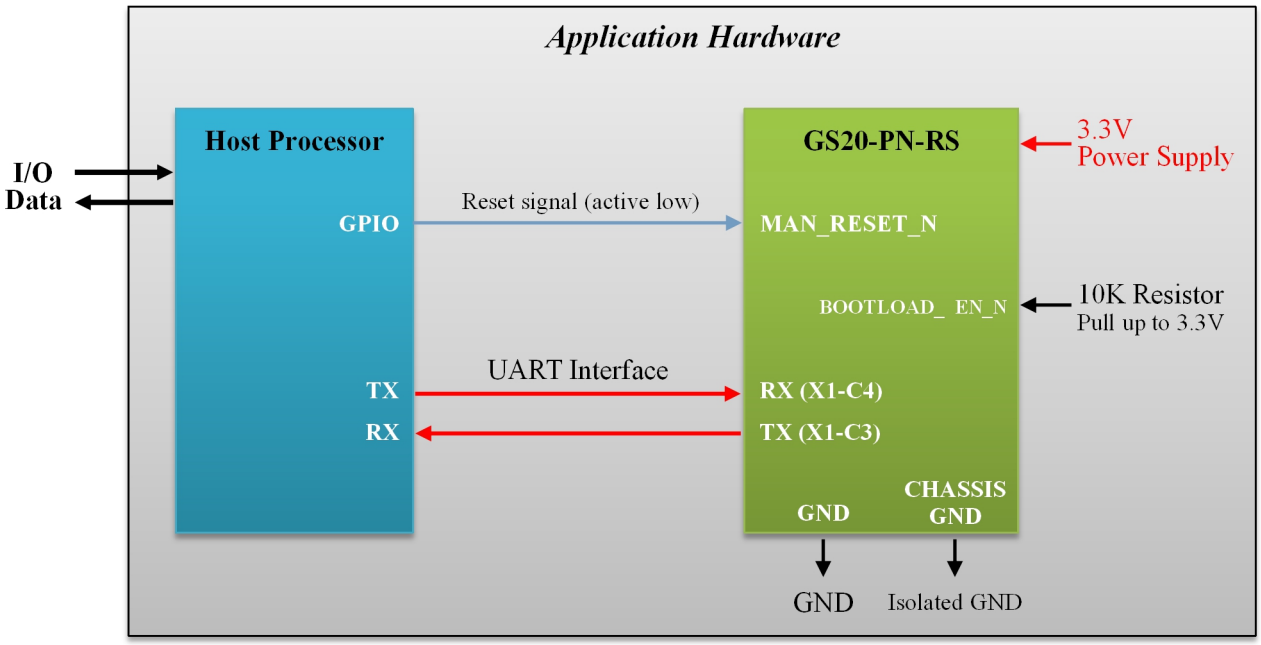


Figure 7 - UART Interface Connection

4.2 General Communication Process

The module is designed with four operational modes: Standby, Configuration, Pend and Run. On power up or reset, the module enters Standby Mode. It will remain in Standby Mode until a Host Request Status frame is received. Once this frame is received, the module will enter Configuration Mode. This process provides a way to synchronize the host and the module after start up. In Configuration Mode, the module operates in a strict command/response manner with the host issuing configuration commands and the module providing status responses. After the host issues a Host Start frame, the module will enter Pend mode while it applies the

configuration information selected by the host and waits for configuration information from the network's DHCP server (only if DHCP is enabled). Once all of the configuration information has been received and applied, the module will enter Run mode. Only when in Run Mode does the module provide unrequested data to the host. Before successfully entering Run Mode, the host must set a device (using the Add Device frame), add at least one item (using the Add Item frame)and set the network settings (using the Set Network Configuration frame). In Run Mode, the data sent to the host by the module consists of four frame types: Network Data, Module (Adapter) Events, Socket Data and Socket Events.

Notes: In lieu of sending the Add Device and Add Item frames, the host can send the Add Basket frame to satisfy the requirement of adding a device and at least one item. However, it should be noted that these two configuration methods are mutually exclusive. Once the host sends an Add Device frame or an Add Item frame, the Add Basket frame becomes locked until the module is reset and will not be processed if received by the module. Conversely, once the host sends an Add Basket frame, the Add Device and Add Item frames become locked and will not be processed.

To use network sockets with the module, the module must first be in Run mode. Once in Run mode, the host can open a socket. Once a socket has been opened, the host may set options for the socket if they are desired. The host can send and receive socket data once the socket has been successfully opened and any desired options have been set. To check the status of a particular socket, the host can poll the module for the status of the socket. Similar to the way Bus data is sent to the host, socket data is sent from the module to the host spontaneously -- the host does not need to poll the module for socket data. Once the host no longer needs a network socket, the host should close the socket. The following table indicates the conditions under which each of the different frames will be sent.

4.2.1 Initialization and Entering Configuration Mode

The initialization sequence consists of a status request from the host and a response from the module. Once the module has received a status request from the host after power up or reset, the module will enter Configuration Mode.

With the module in Configuration Mode, the host can now send an Add Basket frame (or optionally an Add Device frame plus an Add Item frame) to indicate which preloaded device and item to use. This will tell the

module the I/O data size and other information needed to configure the bus. Once the device and item(s) has/have been selected, the module will need its network settings to be configured. To do this, the host will send a Network Configuration frame containing the network configuration information. Details on the required configuration information of PROFINET protocol are contained below.

Frame Name	
Add Basket (or Add Device + Add Item)	**Set Network Configuration**
Required	Required (select "NVM" as addressing mode)

4.2.2 Enter Run Mode

Once the host has sent the required configuration, (i.e. added a device, added one or more items, and set the network configuration) the host can request that module start by sending a Host Start frame. The module will enter Pend mode while all of the applicable configuration and network settings are applied. Once this is done, the module will enter Run mode.

4.2.3 Network Communication

Once the module is in Run Mode, the host can send data to the module to be produced on the network and the module can consume data from the network and relay that data to the host.

While the module is in Run Mode, the host can also open a network socket and implement its own or any other protocol via TCP/IP or UDP/IP. To use a socket, the host must open a socket on the module by sending a Host Open Socket frame. In the Module (Adapter) Response frame, the module will supply the host with a socket handle that the host should use to reference the socket when setting socket options, or sending socket data, etc. In this manner, the host can maintain and reference multiple sockets with multiple configurations. Once the module has opened the socket and supplied the host with a socket handle, the host can immediately begin using the socket to send and receive data. The host may also set any desired socket options at this time.

Socket data reception works in the same way as network data reception. When new socket data comes in, the data is immediately relayed to the host. This process is spontaneous to the host. In addition to spontaneous socket data, the module will also relay spontaneous socket events such as a remote connection or disconnection along with the

pertinent details for the event. With this spontaneous socket event information, the host can make decisions about whether or not to keep the socket open, etc.

4.2.4 Implementation of Communication

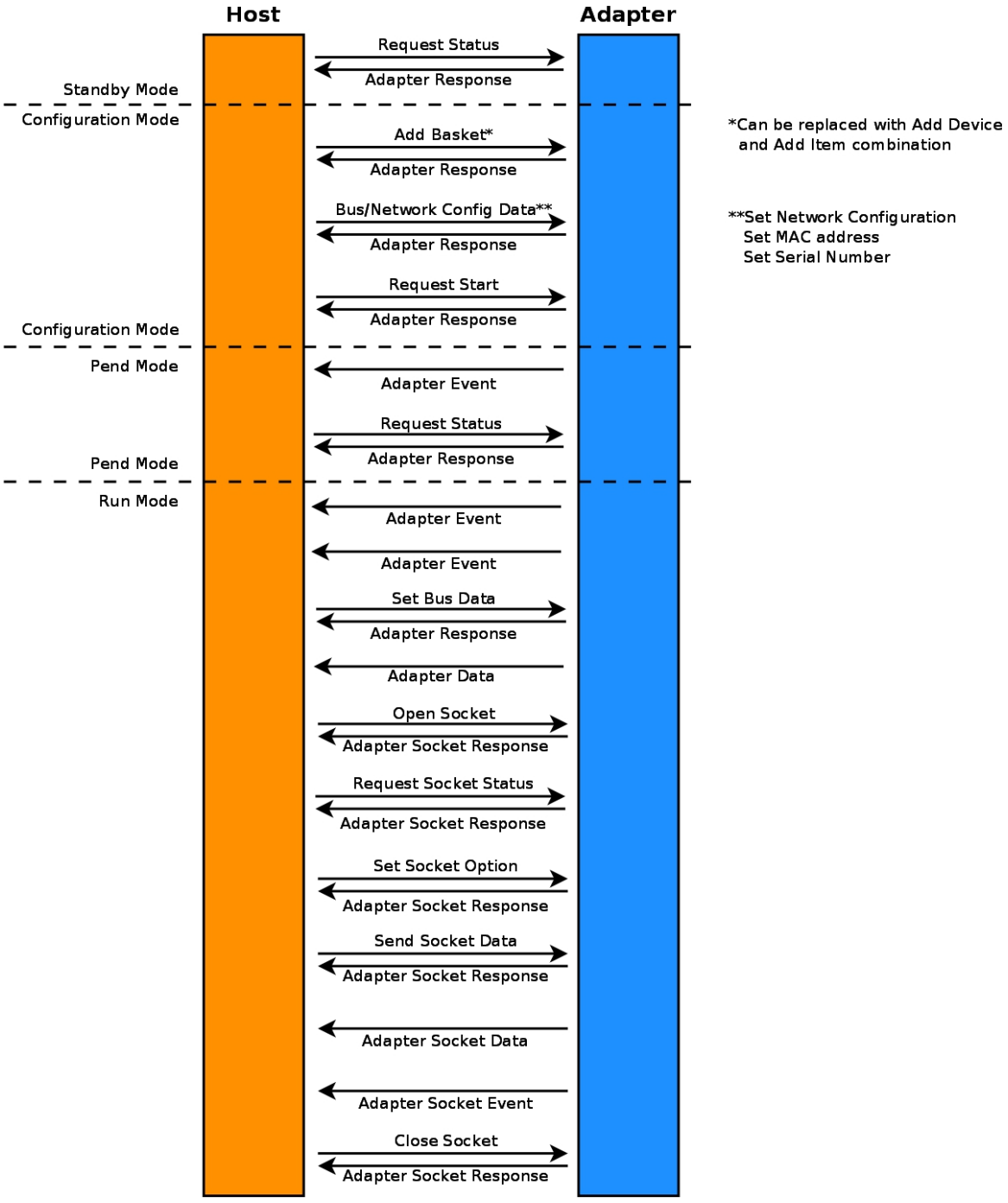


Figure 8 - Example Communication Flow

4.3 Communication between Host and the Module

The data communication between the host and the module (GS20-PN-RS) will use the same basic format. A frame consists of an ID byte, a data length byte, some data bytes, and 2-byte CRC, as shown below:

Byte 1: Frame ID byte
Byte 2: Data length
Byte 3 to Byte n: Transferred data
Byte n+1, n+2: Frame checksum (2 bytes, uses Modbus/RTU CRC algorithm)

4.3.1 Host Commands

Add Basket

Adds the specified basket content (single device and one or more items)

- Valid in Configuration Mode only - generate an error in all other modes
- Sent once, otherwise will generate an error
- Will lock Add Device frame and Add Item frame
- If either Add Device frame or Add Item frame has previously been sent, will generate an error.

Index	Data Type	Value	Number of Bytes
0	Frame ID	0x01	1 byte
1	Data length	2	1 byte
Frame Data			
2	Basket ID	--	2 bytes
4	CRC	--	2 bytes

Add Device

Adds the specified device

- Valid in Configuration Mode only - generate an error in all other modes
- Sent once, otherwise will generate an error
- Will lock Add Basket frame
- If Add Basket frame has previously been sent, will generate an error

Index	Data Type	Value	Number of Bytes
0	Frame ID	0x02	1 byte
1	Data length	4	1 byte

Frame Data			
2	Device ID	--	2 bytes
8	CRC	--	2 bytes

Add Item

Adds the specified item

- Valid in Configuration Mode only - generate an error in all other modes
- Will lock Add Basket frame
- If Add Basket frame has previously been sent, will generate an error

Index	Data Type	Value	Number of Bytes
0	Frame ID	0x03	1 byte
1	Data length	6	1 byte
Frame Data			
2	Item ID	--	2 bytes
4	Location	--	4 bytes
8	CRC	--	2 bytes

* The Item ID is a 4-digit decimal number. The first digit indicates the Item type and the other three digits indicate the Item size, as the following tables show:

Input Bytes	Item ID
1	1001
2	1002
4	1004
8	1008
16	1016
32	1032
64	1064
128	1128
512	1512

Output Bytes	Item ID
1	2001
2	2002
4	2004
8	2008
16	2016
32	2032
64	2064
128	2128
512	2512

Input&Output Bytes	Item ID
1	3001
2	3002
4	3004
8	3008
16	3016
32	3032
64	3064
128	3128
512	3512

For example, add an 8-byte Input&Output Item to No.1 slot. The corresponding Item ID is 3008, as 0x0BC0 in HEX.

Host --> Module: 03 06 0B C0 00 00 00 01 A6 B4

Module --> Host: F0 06 00 03 00 04 02 01 EC 2E (State: Configuration Mode, Ether: Broken, PLC: Broken)

Set Network Configuration

Sets Network configuration information

- Valid in Configuration Mode only - generate an error in all other modes
- If the Addressing Mode is NVM, the supplied IP address, subnet mask, gateway address, DNS1 and DNS2 addresses will be ignored and the corresponding values stored in Non-Volatile Memory will be used

Index	Data Type	Value	Number of Bytes
0	Frame ID	0x10	1 byte
1	Data length	n+21	1 byte
Frame Data			
2	Address Mode	--	1 byte (Static = 0, DHCP = 1, NVM = 2)
3	IP Address	--	4 bytes (big endian)
7	Subnet mask	--	4 bytes (big endian)
11	Gateway Address	--	4 bytes (big endian)
15	DNS 1 Address	--	4 bytes (big endian)
19	DNS 2 Address	--	4 bytes (big endian)
23	Host Name	--	N bytes (Max 63bytes), string
n+23	CRC	--	2 bytes

Set MAC Address

Sets the MAC address for the module

- Valid in Configuration Mode only
- Optional - the MAC address stored in non-volatile memory will be used if this frame is not received from the host

Index	Data Type	Value	Number of Bytes
0	Frame ID	0x11	1 byte
1	Data length	6	1 byte
Frame Data			
2	MAC Address	--	6 bytes (big endian)
8	CRC	--	2 bytes

Set Serial Number

Sets the serial number for the module

- Valid in Configuration Mode only
- Optional - the serial number stored in non-volatile memory will be used if this frame is not received from the host

Index	Data Type	Value	Number of Bytes
0	Frame ID	0x12	1 byte
1	Data length	4	1 byte
Frame Data			
2	Serial Number	--	4 bytes (big endian)
6	CRC	--	2 bytes

Request Status

Requests the current status of the module (adapter)

- Valid in all modes
- Only elicits a response frame

Index	Data Type	Value	Number of Bytes
0	Frame ID	0x20	1 byte
1	Data length	0	1 byte
Frame Data: None			
2	CRC	--	2 bytes

Request Reset

Requests a reset of the module

- Not valid in Standby Mode
- Reset request must be sent two times consecutively to reset the module

Index	Data Type	Value	Number of Bytes
0	Frame ID	0x21	1 byte
1	Data length	0	1 byte
Frame Data: None			
2	CRC	--	2 bytes

Request Start

Requests the module to enter Run Mode

- Valid in Configuration Mode only
- Request the module to enter Run Mode
- Successfully entering Run Mode requires all required configuration frames o have been sent before

Index	Data Type	Value	Number of Bytes
0	Frame ID	0x22	1 byte

1	Data length	0	1 byte
Frame Data: None			
2	CRC	--	2 bytes

Set Bus Data

Contains data to be produced on the Bus

- Valid in Run Mode only
- Request the module to process data on the bus
- The data length must match the output data size from the configuration information
- Data must be placed in the Frame Data section according to the items in use (See chapter 5 for details)

Index	Data Type	Value	Number of Bytes
0	Frame ID	0x30	1 byte
1	Data length	n	2 bytes, n = output data length (set in configuration data)
Frame Data			
3	Output Data	--	n bytes to produced on the bus
n+3	CRC	--	2 bytes

4.3.2 Module (Adapter) Commands

Module (Adapter) Data

Contains data consumed from the network

- Sent to the host by the module (adapter) every 250ms once a PLC connection has been made
- Also sent immediately when the bus data changes
- Data will be packed in the Frame Data section according to the items in use (See chapter 5 for details)

Index	Data Type	Value	Number of Bytes
0	Frame ID	0x31	1 byte
1	Data length	n	2 bytes, n = input data length (set in configuration data)
Frame Data			
3	Input Data	--	n bytes to produced on the bus
n+3	CRC	--	2 bytes

4.3.3 Module (Adapter) Responses

The module will respond to every Host Command frame or Host Data frame with an Module (Adapter) Response frame.

Module (Adapter) Responses

Contains data consumed from the network

- Sent by adapter in all modes as a response to any command from host

Index	Data Type	Value	Number of Bytes
0	Frame ID	0xF0	1 byte
1	Data length	6	1 byte
Frame Data			
2	Input Data	--	2 bytes (See Status and Mode Table)
4	Error Code	--	2 bytes (See Error Code Table)
6	Protocol Type	--	1 byte (See Protocol Type Table)
7	Interface Version Number	--	1 byte
8	CRC	--	2 bytes

State and Mode Table

Bit	Description	
0	0 = No errors	1 = Error occurred, check error code
1-2	00 = Standby Mode	
	01 = Configuration Mode	
	10 = Pend Mode	
	11 = Run Mode	
3	0 = Ethernet link state unknown	1 = Ethernet link state known
4*	0 = No Ethernet connection	1 = Ethernet connection OK
5**	0 = No IP address	1 = IP address OK
6**	0 = Network error	1 = Network OK
7**	0 = No connection to PLC	1 = PLC connection OK
8	0 = I/O data is big-endian	1 = I/O data is little endian
9-15	Reserved (0)	

* Read as 0 if link state is unknown

** Always read as 0 until the module (adapter) enters Pend or Run Mode

Protocol Type Table

Type	Protocol
1	Reserved
2	PROFINET
3	Reserved
4	Reserved
5-255	Reserved

Error Code Table

Bit	Error	Possible Reason
0	Incomplete configuration	○ No device added* ○ No items added*
1	Device/Item/Basket Add error	○ There is no device in the configuration data ○ Device added successfully ○ Error adding device ○ Error adding device in basket ○ There is no item in the configuration data ○ Error adding item ○ Error adding item in the basket ○ The maximum number of items have been added ○ Basket added successfully ○ There is no basket in the configuration data ○ The basket in the configuration data does not contain item or device ○ Items in the basket contain invalid data
2	No network configuration	○ No network configuration data is received
3	Network configuration error	○ Invalid address mode selection ○ Invalid IP address ○ Invalid DNS1/2 address ○ Invalid gateway address ○ The length of host name is 0 byte ○ The length of host name is more than 63 bytes ○ The host name contains invalid characters
4	CRC error	○ Incorrect CRC
5	Frame error	○ Unrecognized frame ID ○ Frame sent in wrong mode ○ Unsupported Host Socket Command or Socket Data frame ○ Data length does not consistent with frame type ○ Frame data is too large for buffer

6	Start Request Denied	○ The bus is not configured ○ Invalid bus configuration ○ No network configuration ○ Invalid network configuration data ○ No Ethernet connection
7	I/O data size does not match	○ The size of the host data does not match the size of the module I/O data
8	Lock violation	○ Add Basket frame sent after Add Device frame ○ Add Basket frame sent after Add Item frame ○ Add Device frame sent after Add Basket frame ○ Add Item frame sent after Add Basket frame

*Adding a basket (without errors) will add a device and one or more items, satisfying the requirement of having a device and at least one item added.

Module (Adapter) Event

Indicates the occurrence of the designed event

- Not sent in Standby Mode or Configuration Mode
- Sent spontaneously from the module (adapter) after an event occurs
- If applicable, extra event information will be sent along with the event code

Index	Data Type	Value	Number of Bytes
0	Frame ID	0xF1	1 byte
1	Data length	n+3	1 byte
Frame Data			
2	Module mode	--	1 byte (0 =Standby, 1 = Configure, 2 = Pend, 3 = Run)
3	Module event code	--	2 bytes, see Module (Adapter) Event Code Table
5	Module event information	--	n bytes, see Module (Adapter) Event Information Table
n+5	CRC	--	2 bytes

Module (Adapter) Event Code Table

Number	Event Name	Value	Description
1	Ethernet link established	0	Cable was plugged in or link was detected
2	Ethernet disconnection	0	Ethernet cable was severed or unplugged
3	Network connection established	17	Successfully connected to network
4	PLC connection established*	0	PLC made connection to adapter
5	PLC connection broken	0	PLC stopped responding

6	PLC reset	1	PLC commanded adapter to reset
>6	Reserved		

* May be sent for a connection/disconnection to each data item.

Module (Adapter) Event Information Table

Event Name	Size	Offset	Information
Network connection established	17	0	Addressing mode (Static = 0, DHCP=1)
		1	IP address in use (big endian)
		5	Subnet mask in use (big endian)
		9	Gateway address (big endian)
		13	DNS server address in use (big endian)
PLC reset	1	0	Reset mode (0 or 1)

Module (Adapter) Event Modes

Event Name	Mode	
	Pend	Run
Ethernet link established	Valid	Valid
Ethernet disconnection	Valid	Valid
Network connection established	Invalid	Valid
PLC connection established*	Invalid	Valid
PLC connection broken	Invalid	Valid
PLC reset	Invalid	Valid

4.3.4 Host Socket Commands

Open Socket

Opens a socket of the specified type and direction

- Valid in Run Mode only
- If the socket type is TCP server, the socket will begin listening and then automatically accept any incoming connections
- If the socket type is TCP client, the socket will attempt to connect to the specified server
- If the socket type is UDP server the socket will become open for use by the host
- If the socket type is UDP client, the socket will become open for use by the host
- The Module (Adapter) Socket Response frame that follows this command will contain the assigned handle number in the Socket Handle field. This value should be recorded in order to reference the newly opened socket when setting socket options, sending socket data, etc

Index	Data Type	Value	Number of Bytes
0	Frame ID	0x40	1 byte

1	Data length	8	1 byte
Frame Data			
2	Socket Direction	--	1 byte (0 = Client, 1 = Server)
3	Socket Type	--	1 byte (0 = TCP, 1 = UDP)
4	Port Number	--	2 bytes
6	IP Address	--	4 bytes (MSB is MS octet)
10	CRC	--	2 bytes

Notes:

For TCP server sockets, normal usage allows multiple TCP clients to connect to a single TCP server socket. This is still the case when using the Unified Interface. However, the host will need to send the Open Socket frame with the same socket settings (direction, type and port number) as many times as there are TCP clients the host would like to support. In other words, to support n TCP client connections, the host will need to send the Open Socket frame n times with the same TCP server settings. When opening multiple sockets in this way, the first socket opened will be considered the parent socket. Any subsequently opened sockets will be considered child sockets.

All child sockets must be closed before the parent socket can be closed. Attempting to close a parent socket before all child sockets are closed will result in an error.

Set Socket Option

Sets the specified option for the specified socket

- Valid in Run Mode only

Index	Data Type	Value	Number of Bytes
0	Frame ID	0x41	1 byte
1	Data length	8	1 byte
Frame Data			
2	Socket Handle	--	1 byte
3	Socket Option	--	1 byte (see Socket Option Table)
4	Socket Option Data	--	n bytes, dictated by socket option
n+4	CRC	--	2 bytes

Socket Option Table

Value	Socket Option	n	Action
0	Enable Broadcasts	1	Enable or disable broadcast messages
1	Reuse Address	1	Bind to a port even if it is in use

2	Type of Service	1	Set Type of Service or Traffic Class field in IP header
3	Limited Broadcast	1	Enables or disables use of the limited broadcast address. If disabled, limited broadcast address is replaced with interface broadcast address.
4	Multicast TTL	1	Join the specified multicast group
5	Add Multicast Membership	4	Join the specified multicast group
6	Drop Multicast Membership	4	Leave the specified multicast group
7	No TCP Delay	1	Enables or disables the Nagle algorithm (TCP sockets only)
8	Set TCP Keepalive	1	Enables and sets interval for keepalive queries (TCP server or client only)

Socket Option Data Table

Value	Socket Option	Size (bytes)	Option Data Description	Value
0	Enable Broadcasts (Choose only one)	1	Broadcast messages not allowed	0x00
		1	Broadcast messages allowed	0x01
1	Reuse Address (Choose only one)	1	Disable	0x00
		1	Enable	0x01
2	Type of Service	1	Type of Service value for IP header	0x00 to 0xFF
3	Limited Broadcast	1	Disable	0x00
		1	Enable	0x01
4	Multicast TTL	1	Multicast TTL Value	1-255
5	Add Multicast Membership	4	Multicast Group IP Address	0x0 to 0xFFFFFFFF (Big endian)
6	Drop Multicast Membership	4	Multicast Group IP Address	0x0 to 0xFFFFFFFF (Big endian)
7	No TCP Delay (Choose only one)	1	Disable	0x00
		1	Enable	0x01
8	Set TCP Keepalive	1	Keepalive query interval length in seconds (remote socket will be timed out after 13-15 intervals)	0x1 to 0x63

Request Socket Status

Requests the status of the specified socket

- Valid in Run Mode only
- Does nothing but elicit an Module (Adapter) Socket Response frame for the specified socket

Index	Data Type	Value	Number of Bytes
0	Frame ID	0x42	1 byte
1	Data length	1	1 byte
Frame Data			
2	Socket Handle	--	1 byte
3	CRC	--	2 bytes

Close socket

Requests the module (adapter) to close the specified socket

- Valid in Run Mode only

Index	Data Type	Value	Number of Bytes
0	Frame ID	0x43	1 byte
1	Data length	1	1 byte
Frame Data			
2	Socket Handle	--	1 byte
3	CRC	--	2 bytes

Send Socket Data

Sends data using the specified socket

- Valid in Run Mode only
- Socket data size (n) must be lees than 1500 bytes

Index	Data Type	Value	Number of Bytes
0	Frame ID	0x50	1 byte
1	Data length	n+1	2 bytes
Frame Data			
3	Socket Handle	--	1 byte
4	Socket Data	--	n bytes
n+4	CRC	--	2 bytes

4.3.5 Module (Adapter) Socket Commands and Responses

Module (Adapter) Socket Data

Data from the network on the specified socket

- Sent spontaneously by the adapter when new socket data arrives

Index	Data Type	Value	Number of Bytes
0	Frame ID	0x51	1 byte
1	Data length	n+1	2 bytes
Frame Data			
3	Socket Handle	--	1 byte
4	Socket Data	--	n bytes
n+4	CRC	--	2 bytes

Module (Adapter) Socket Responses

Indicates the status of the specified socket along with any errors

- Send from module (adapter) after every host socket frame
- When this frame is sent to the host immediately following a Host Open Socket Command, the Socket Handle field will contain the handle number that the host should use to reference the socket when setting socket options, opening the socket, etc.
- For errors that don't reference a specific socket (e.g. opening a socket), the socket handle field will be 0xFF

Index	Data Type	Value	Number of Bytes
0	Frame ID	0xE0	1 byte
1	Data length	5	1 byte
Frame Data			
2	Socket Handle	--	1 byte
3	Socket Status	--	2 bytes (see Socket Status Table)
5	Socket Error	--	2 bytes (see Socket Error Table)
7	CRC	--	2 bytes

Socket Status Table

Bit	Status
0	Listening (UDP client/server)
1	Connecting (TCP client)

2	Accepting (TCP server)
3	Connected (TCP client/server)
4	Dead (i.e. should be closed)
5	Disconnected (TCP server, status should go to Accepting shortly)

Socket Error Table

Bit	Error Name	Possible Reason
0	Socket does not exist	O Host referenced an unopened socket
1	Failed to open socket	O Module (adapter) out of sockets (too many other sockets already opened) O Module (adapter) doesn't support designated socket type O Host specified invalid IP address O Invalid Client/Server selection (was not 1 or 0) O Invalid TCP/UDP selection (was not 1 or 0) O Failed to bind socket
2	Invalid socket option	O Host specified a socket option that is not recognized
3	Invalid socket options data	O Host specified invalid socket option data O Host specified too much or too little socket option data
4	Failed to send data	O Host specified too much socket data O Host specified too little socket data O Socket was not open
5	Close failed	O Host attempted to close a parent socket

Module (Adapter) Socket Event

Indicates the occurrence of the designated event on the specified socket

- Sent from module (adapter) after a socket event occurs
- Socket status is the same as the status from the module (adapter) Socket Response frame
- Extra event information is specified in addition to the event type

Index	Data Type	Value	Number of Btyes
0	Frame head	0xE1	1 byte
1	Data length	n+4	1 byte
Frame Data			
2	Socket Handle	--	1 byte
3	Socket Status	--	2 bytes (see Socket Status Table)
5	Socket Event Code	--	1 byte (see Socket Event Code Table)
6	Socket Event Information	--	n bytes (specified by event type)

n+6	CRC	--	2 bytes

Socket Event Code Table

Number	Event Name	Description
1	Client Connected	A client has connected to the local server socket
2	Client Closed Connection	External client closed the connection
3	Connected to Server	Connection to external server was established
4	Server Closed Connection	External sever closed connection
5	Server Rejected Connection	External server rejected connection request

Socket Event Information

If an event is not listed in this table, there is no additional information associated with that event.

Number	Event Name	Size (bytes)	Description
1	Client Connected	4	IP address of client (big endian)
2	Client Closed Connection	4	IP address of closing node (big endian)
3	Server Rejected Connection	2	Error code from server

5 Items and Baskets

The information that describes the input and output data sizes for the module will be contained in items. If the host software developer would like to create a logical grouping of items, he or she can add the items (with a device) to a basket. Once the items (and basket) have been created, a configuration file can be generated using the Configuration Tool (please contact support@sstcomm.com for tools). This configuration file will be loaded into the module's non-volatile memory for the host to reference at configuration-time. If the host adds all of the desired items by adding a basket that contains them all, the network I/O data transferred to and from the host will be stacked according to the order in which the items were added to the basket using the Configuration Tool. If the host adds all of the desired items by repeatedly sending the Add Item frame, the network I/O data transferred to and from the host will be stacked according to the order in which the items were added.

Notes:

- It is completely acceptable to have only one item in a basket
- Item IDs and basket IDs are completely arbitrary; their values are up to the developer
- Multiple baskets can exist in one configuration file but only one basket can be used at a time
- Multiple items can be exported to a configuration file but not all of them have to be added by the host at run-time

5.1 PROFINET Items

For a PROFINET device, we will add 3 items; one for the digital I/O data, one for the analog I/O data and one for the hidden data:

Item 1 (Digital Input and Output Module)

- Module ID = 0x10200000
- Submodule ID = 0x10220001
- Input size: 2 bytes
- Output data size: 2 bytes

Item 2 (Analog Input and Output Module)

- Module ID = 0x10300000
- Submodule ID = 0x10330001

- Input size: 4 bytes
- Output size: 4 bytes

Item 3 (Hidden Data Module)

- Module ID = 0x10400000
- Submodule ID = 0x10440001
- Input size: 0 bytes
- Output size: 2 bytes

Notes: The Module and Submodule IDs used when creating items will need to match the GSDML file. The item location must also be the same slot number assigned in the network configuration of the PLC. As shown in the diagram below, the most significant 2 bytes in the 6 byte payload for the Host Data frame are the 2 bytes of input data for item 1; the other 4 bytes are the input data for item 2. Similarly, in the Module (Adapter) Data frame, the most significant 2 bytes in the 8 byte payload are the output data for item 1 while the next 4 bytes are the output data for item 2 and the least significant 2 bytes are for hidden data represented by item 3. Note that the order of the data in the Host Data frame and the Adapter Data frame is set by the order in which the items were added to the basket or by the order in which the items were added by the host.