

Embedded PROFINET IO RT Interface Module GS20-PN-RS

User Manual

V2.1 Rev A



SST Automation

E-mail: SUPPORT@SSTCOMM.COM

WWW.SSTCOMM.COM



Catalog

1 Product Overview.....	3
1.1 Product Function.....	3
1.2 Product Features.....	3
1.3 Technical Specifications.....	3
1.4 Product Development Process.....	5
1.5 Revision History.....	6
2 Hardware Description.....	7
2.1 Hardware Schematic Diagram and Size.....	7
2.2 Network interface module pin Definition and Description.....	8
2.3 Ethernet interface.....	12
3 Software Description.....	13
3.1 Serial port communication.....	13
3.2 Basic communication process.....	13
3.2.1 Initialize and enter configuration mode.....	14
3.2.2 Enter run mode.....	15
3.2.3 Network Communications.....	15
3.3 Host and module communication process.....	16
4 Items and Baskets.....	28
4.1 PROFINET Items.....	28



1 Product Overview

1.1 Product Function

This product is for the users who wish to develop the products with PROFINET interface through UART.

1.2 Product Features

- Short Period of Development: Users don't need to know more knowledge about PROFINET and don't need to buy PROFINET development system; They only need to write GSDML files according to their own equipment, and can complete the development of PROFINET products in a short time.
- Easy to Use: MCU of user products can read / write input / output data to GS20-PN-RS through serial interface (UART) , and GS20-PN-RS automatically converts into PROFINET Protocol and communicate with PROFINET Master.
- 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] Support PROFINET IO RT V2.2 Protocol

[2] PROFINET input-byte and output-byte number can be freely set and the maximum input bytes 384 and output bytes 384

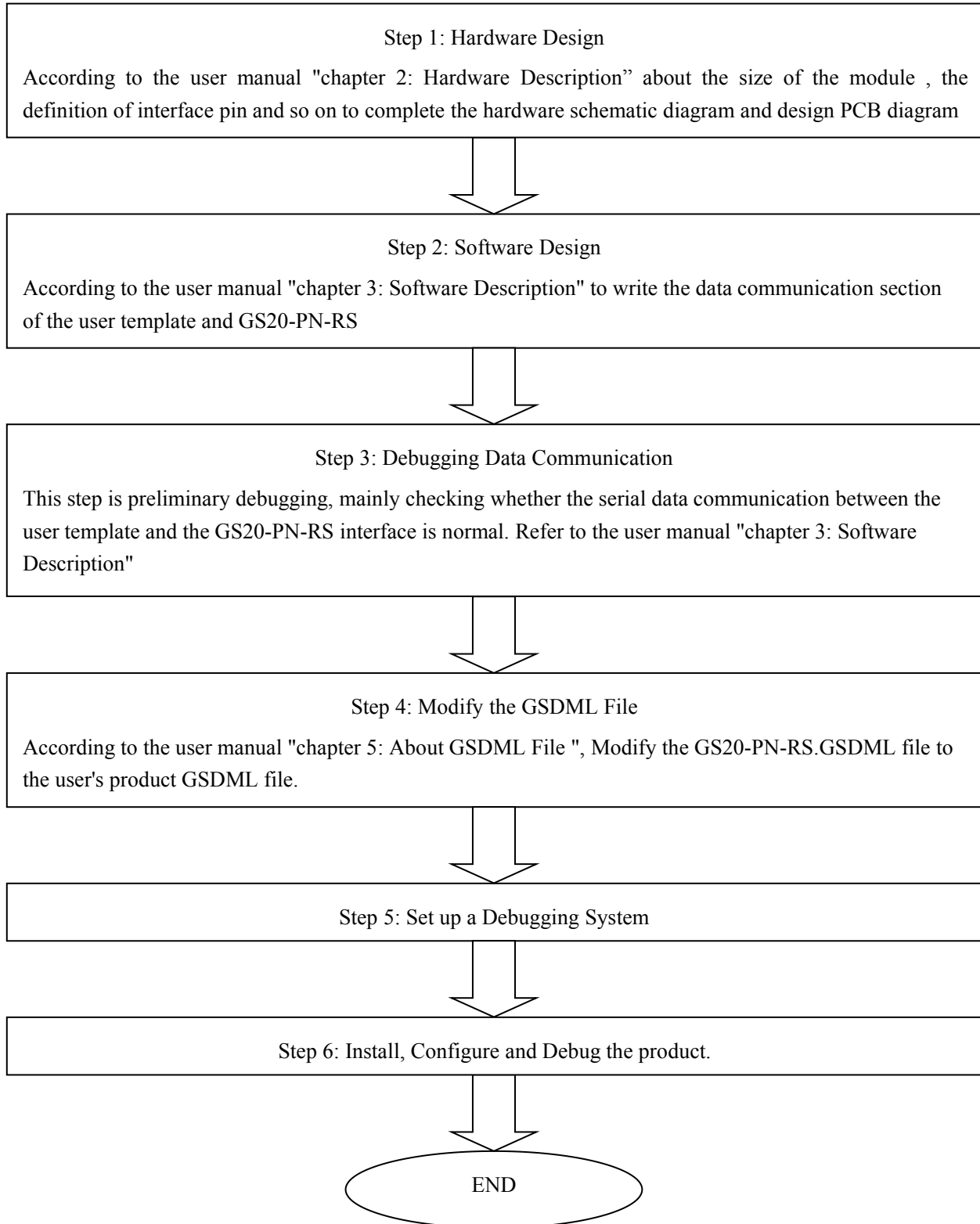
[3] Support serial interface (UART)

[4] GS20-PN-RS and user board interface are CMOS TTL asynchronous serial port, baud rate: Fixed to 115.2K, data bit: 8 bit, even check, 1 stop bit

[5] GS20-PN-RS and user template communication have character even check and longitudinal message CRC check to ensure data security

- [6] Support TCP and UDP communication
- [7] Power: 3.3VDC, 500mA, It is suggested that 2W/3.3VDC supply can be provided.
- [8] Working temperature: -4°F~122°F (-20°C~50°C), 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 Product Development Process



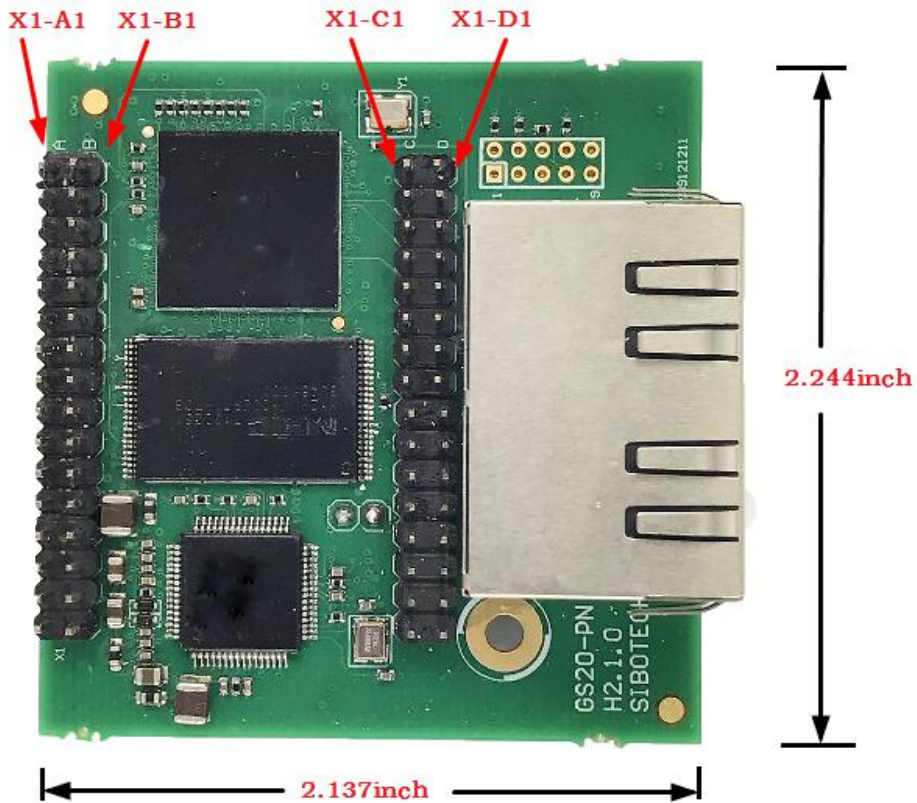


1.5 Revision History

Revision	Date	Chapter	Description
REV A	22/6/2015	All	First release V2.1 manual

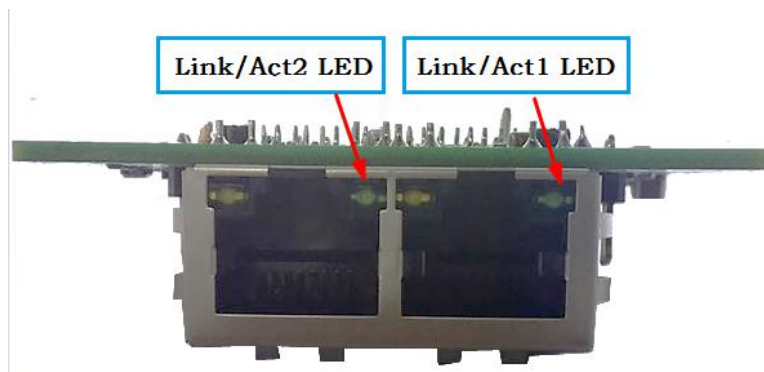
2 Hardware Description

2.1 Hardware Schematic Diagram and Size



GS20-PN-RS Vertical View and Size

Note:length:2.137 inch (54.3mm),width:2.244 inch (57mm)



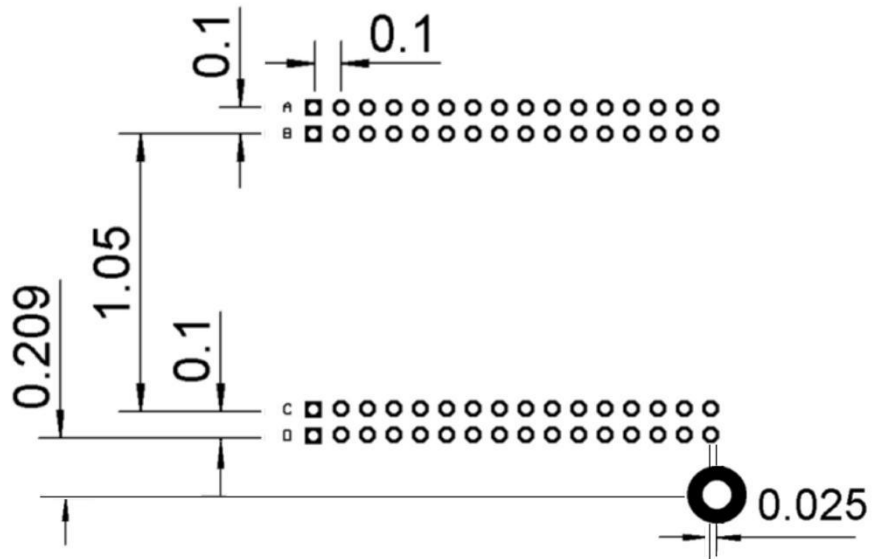
GS20-PN-RS Front View



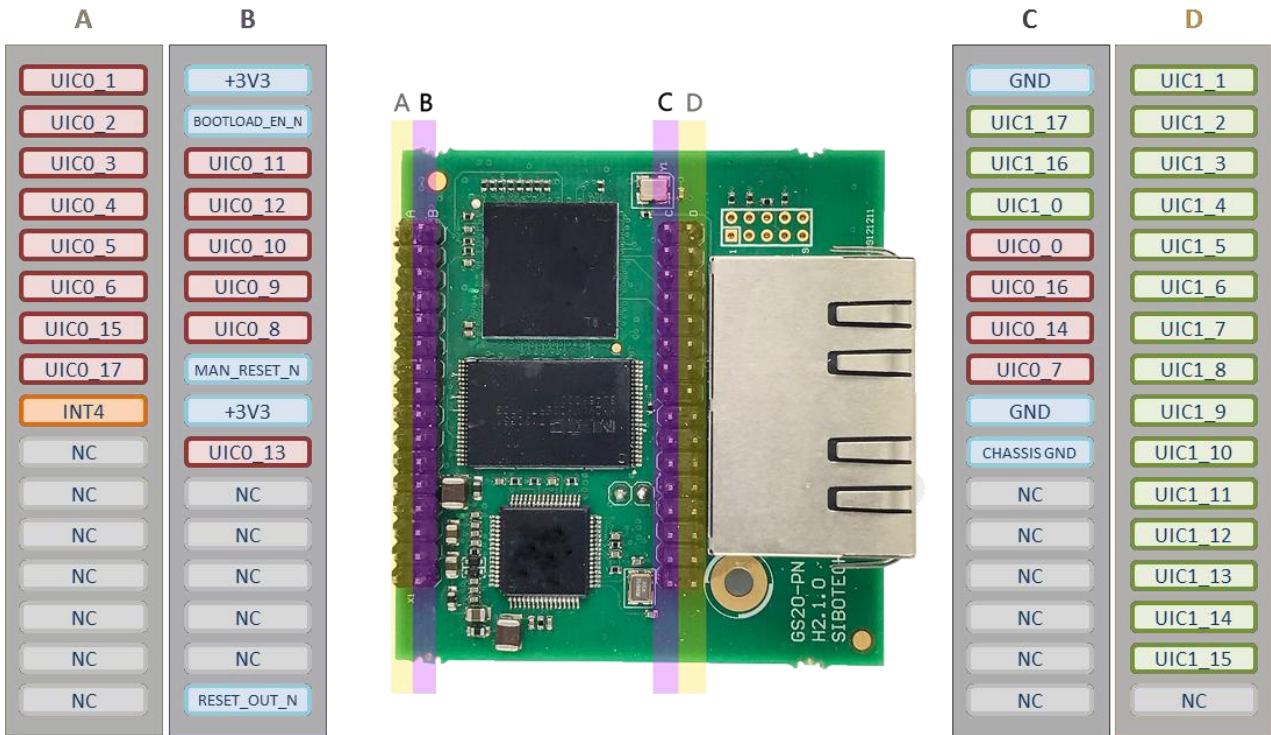
GS20-PN-RS Lateral View

2.2 Network interface module pin Definition and Description

The following sizes are all inch.



GS20-PN-RS interface module pin definition:



Interface module pin Definition

Network interface module pin Definition--X1-A

X1 Pin	Signal name	Description	Function
A1	UIC0_1	input	Reverse for Parallel
A2	UIC0_2	input	Reverse for Parallel
A3	UIC0_3	input	Reverse for Parallel
A4	UIC0_4	input	Reverse for Parallel
A5	UIC0_5	input	Reverse for Parallel
A6	UIC0_6	input	Reverse for Parallel
A7	UIC0_15	input	Reverse for Parallel
A8	UIC0_17	output	Reverse for Parallel
A9	INT4	input	Reverse for Parallel

A10	NC	disconnection	empty
A11	NC	disconnection	empty
A12	NC	disconnection	empty
A13	NC	disconnection	empty
A14	NC	disconnection	empty
A15	NC	disconnection	empty
A16	NC	disconnection	empty

Network interface module pin Definition--X1-B

X1 Pin	Signal name	Description	Function
B1	+3V3	GND	3.3V
B2	BOOTLOAD_EN _N	input	Boot load enabled (low level voltage effective) 10K pull resistance module
B3	UIC0_11	input	Reverse for Parallel
B4	UIC0_12	output	Reverse for Parallel
B5	UIC0_10	input	Reverse for Parallel
B6	UIC0_9	input	Reverse for Parallel
B7	UIC0_8	output	Reverse for Parallel
B8	MAN_RESET_N	output	Reset Pin(Low level voltage effective), connect to one of pins of host MCU to reset the module
B9	+3V3	power	3.3V
B10	UIC0_13	input	Reverse for Parallel
B11	NC	disconnection	empty
B12	NC	disconnection	empty
B13	NC	disconnection	empty
B14	NC	disconnection	empty
B15	NC	disconnection	empty
B16	RESET_OUT_N	output	Reset output, When the module reset, the pin is low

level voltage which indicates the module is in reset.

Network interface module pin Definition--X1-C

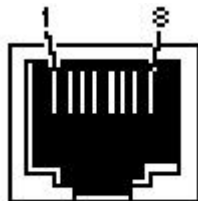
X1 Pin	Signal name	Description	Function
C1	GND	GND	Signal GND
C2	UIC1_17	output	Reverse for Parallel
C3	UIC1_16	input	UART_TX(Connect to the host RX)
C4	UIC1_0	output	UART_RX(Connect to the host TX)
C5	UIC0_0	input	Reverse for Parallel
C6	UIC0_16	input	Reverse for Parallel
C7	UIC0_14	output	Reverse for Parallel
C8	UIC0_7	output	Reverse for Parallel
C9	GND	GND	Signal GND
C10	CHASSIS GND	CHASSIS GND	CHASSIS GND
C11	NC	disconnection	empty
C12	NC	disconnection	empty
C13	NC	disconnection	empty
C14	NC	disconnection	empty
C15	NC	disconnection	empty
C16	NC	disconnection	empty

Network interface module pin Definition--X1-D

X1 Pin	Signal name	Description	Function
D1	UIC1_1	output	Reverse for Parallel
D2	UIC1_2	output	Reverse for Parallel
D3	UIC1_3	output	Reverse for Parallel

D4	UIC1_4	output	Reverse for Parallel
D5	UIC1_5	output	Reverse for Parallel
D6	UIC1_6	output	Reverse for Parallel
D7	UIC1_7	output	Reverse for Parallel
D8	UIC1_8	output	Reverse for Parallel
D9	UIC1_9	output	Reverse for Parallel
D10	UIC1_10	output	Reverse for Parallel
D11	UIC1_11	output	Reverse for Parallel
D12	UIC1_12	output	Reverse for Parallel
D13	UIC1_13	output	Reverse for Parallel
D14	UIC1_14	output	Reverse for Parallel
D15	UIC1_15	output	Reverse for Parallel
D16	NC	disconnection	empty

2.3 Ethernet interface



RJ-45 port

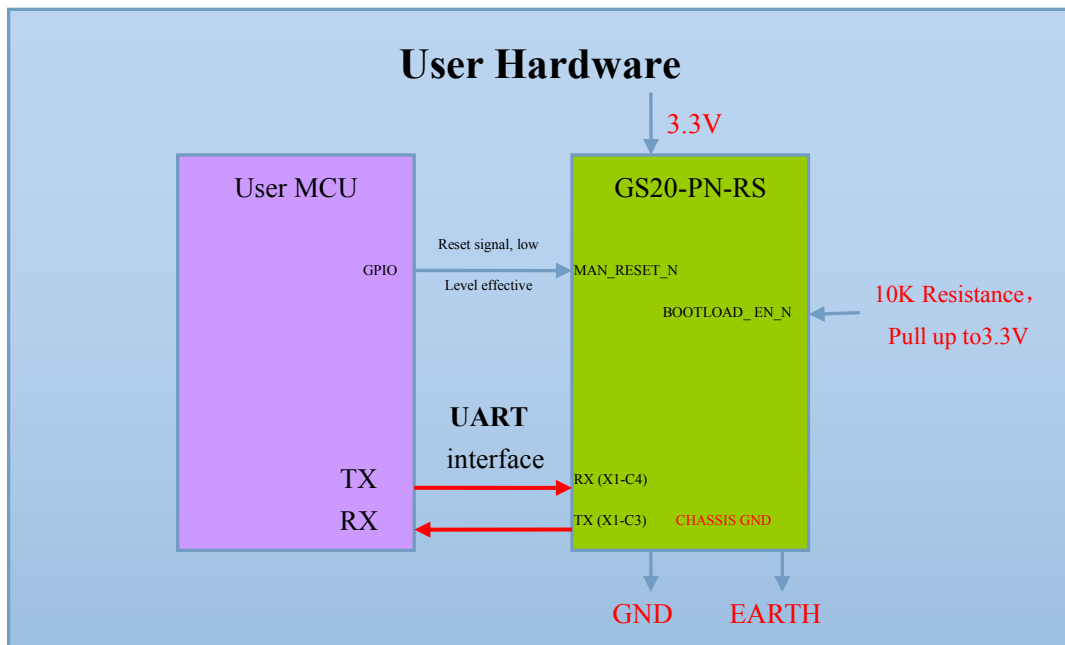
Ethernet interface uses RJ-45 connector; its pin (standard Ethernet signal) is defined as below:

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-

3 Software Description

3.1 Serial port communication

Data exchange between GS20-PN-RS module and host: serial port connection, TXD, RXD as COMS TTL level, half duplex .Asynchronous serial port data frame format: every 1 byte is transmitted by 11bits: 1 starting bit, 8 data bit, 1 even check bit, 1 stop bit, and the asynchronous serial port baud rate is 115200bps.



Serial interface wiring diagram

3.2 Basic communication process

The module has four Running modes: standby, configuration, hanging and run. Power or Reset, the module enters standby mode. The module will remain in standby mode until a host request state frame is received. Once received, it will enter the configuration mode. This process enables the host and module to synchronize. In the configuration mode, the module works in a strict command / response mode, it is that the host sends configuration commands and module provides state response. When the host sends a start frame, the module will enter the



hanging mode. At the same time, it will use the configuration information selected by the host and wait for the configuration information from the DHCP server in the network (if DHCP function starts). Once the configuration information is received and applied, the module enters run mode. Only in the run mode, the module provides unrequested data to the host. Before entering the run mode successfully, the host must set up a device (add the device frame), add at least one item (increase the item frame) and set up the network (set up the network configuration frame). In the run mode, the data that the module sends to the host consists of four data types: network data, module event, socket data, and socket event.

NOTE: The host can send an added Basket frame to satisfy the increase of device requirements and at least one item, which can replace to send add devices and item frames. However, it should be noted that the two configuration methods are mutually exclusive. Once the host sends an added device frame and added project frame, the added Basket frame is locked until the module is reset and receive no handled. In contrast, if the host sends a Basket frame, adding a device frame and adding an item frame is locked and not handled.

To use the socket of the module network, the module must first enter the run mode. In the run mode, the host can open the socket. Once a socket is opened, the host can set the options for the desired socket. Once the socket is successfully opened, the host can send and receive the socket data and any required options have been set. To detect a particular socket state, the host can poll the state of the module socket. A similar method sends data to the host, and the module can send the socket data to the host spontaneously -- the host does not need the polling module.

3.2.1 Initialize and enter configuration mode

Initialization is made up of a series of host request States and module response states. Power or reset on the host, and the module receives a request state immediately into the configuration mode. In the configuration mode, the host can send an ADD Basket frame (or random an ADD Device frame + Add Item frame), indicating the use of preloading devices and items. Tell the module about the I/O data size and information needed to configure the bus. Once devices and items are selected, the module will configure the network according to the information. For these actions, the host sends a configuration frame about the network configuration information.



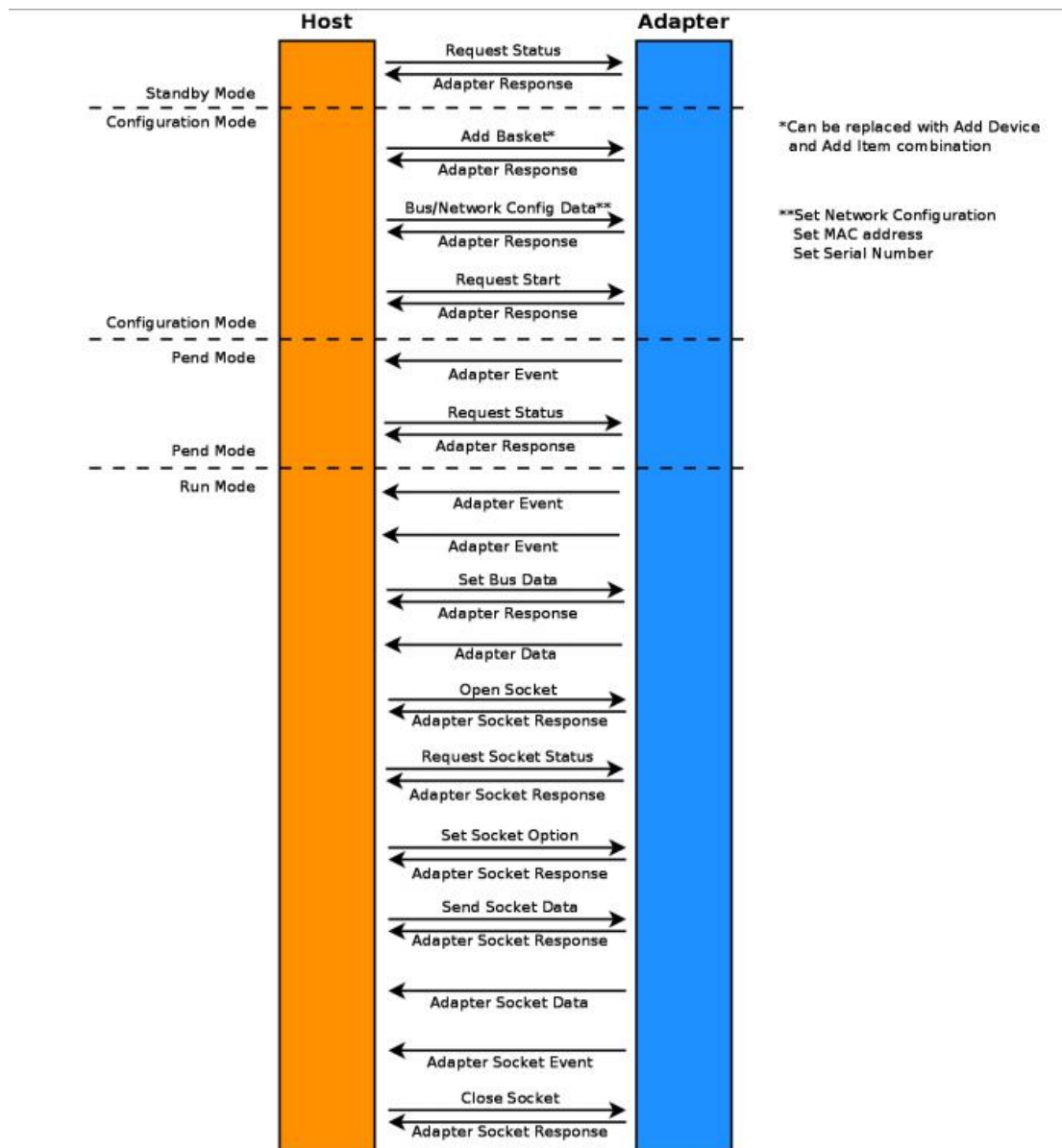
3.2.2 Enter run mode

Once the host sends the request configuration, (i.e. added a device, added one or more items, and set the network configuration). The host sends a Host Start frame that is the module start. The module will enter the hanging mode and all configuration and network settings are applied. When these are done, the module enters the run mode.

3.2.3 Network Communications

In the run mode, the host and module (Adapter) can send data to each other. The host can open the network socket and execute itself or other protocols through TCP/IP or UDP/IP to use socket, the host must send a Host Open Socket frame to open module socket. In the module response frame, the module provides a socket handle to the host that the host must refer to this socket, when the socket option is set, or the data is sent. In this way, the host can maintain and reference multiple sockets with multiple configurations. Once the module opens the socket and provides a socket handle to the host, the host can immediately receive and send data using the socket. At this time the host can set the socket option. Socket data reception work is the same as network data reception. When the new socket data comes in, the data is sent to the host immediately and the process is spontaneous. In addition to spontaneous socket data, modules will also send socket events spontaneously, such as remote connections or disconnected details related to events. With regard to the spontaneous socket event information, the host can decide whether to keep the socket open, and so on.

3.3 Host and module communication process



Communication instance flow chart

The same basic frame format is used for data communication between the host and the modules (Adapter).

Frame format: ID byte + valid data bit + data +2 byte CRC check bit

Byte 1: Frame ID bytes

Byte 2: Valid data length of a frame

Byte 3----Byte n: data

Byte n+1, n+2: Frame check bytes (2 bytes, Modbus/RTU CRC algorithm)

The host command is as follows:

1) Add Basket

Add the specified basket content (Single device and one or more items)

Only in the configuration mode is OK - other patterns produce an error

It can only be sent once or there will be an error.

Lock the Add Device frame + add Item frame

If Add Device frame or Add Item frame is sent, there will be an error:

Index value	data type	value	Number of bytes
0	Frame head	0x01	1byte
1	Data length	2	1byte
Frame data bit			
2	Basket ID	--	2bytes
4	Check	--	2bytes

2)Add Device

Index value	data type	value	Number of bytes
0	Frame head	0x02	1byte
1	Data length	2	1byte
Frame data bit			
2	Device ID	--	2bytes
4	Check	--	2bytes

3) Add Item

Index value	data type	value	Number of bytes
0	Frame head	0x03	1byte
1	Data length	6	1byte
Frame data bit			
2	Item ID	--	2bytes
4	Location	--	4bytes

8	Check	--	2bytes
---	-------	----	--------

4) Set Network Configuration

Index value	data type	value	data size
0	Frame head	0x10	1byte
1	Data length	N+21	1byte
Frame data bit			
2	Address mode	----	1byte
3	IP Address	----	4bytes
7	Subnet mask	----	4bytes
11	Gateway Address	----	4bytes
15	DNS 1 Address	----	4bytes
19	DNS 2 Address	----	4bytes
23	Host name	----	N bytes (Max 63bytes)
N+23	Check bit	----	2bytes

5) Set MAC Address

Index value	data type	value	Number of bytes
0	Frame head	0x11	1byte
1	Data length	6	1byte
Frame data bit			
2	MAC Address	--	6bytes (Big-endian format)
8	Check	--	2bytes

6) Set Serial Number

Index value	data type	value	Number of bytes
0	Frame head	0x12	1byte
1	Data length	4	1byte
Frame data bit			
2	Serial Number	--	4bytes (Big-endian format)
6	Check	--	2bytes

7) Request Status

Index value	data type	value	data size
0	Frame head	0x20	1 byte
1	Data length	0	1 byte
Frame data bit(no)			
2	Check bit	-----	2bytes

8) Request Reset

Index value	data type	value	data size
0	Frame head	0x21	1 byte
1	Data length	0	1 byte
Frame data bit(no)			
2	Check bit	-----	2bytes

9) Request Start

Index value	data type	value	data size
0	Frame head	0x22	1 byte
1	Data length	0	1 byte
Frame data bit(no)			
2	Check bit	-----	2bytes

10) Set Bus Data

Index value	data type	value	data size
0	Frame head	0x30	1 byte
1	Data length	N	2bytes,n=Output data size(Set configuration data)
Frame data bit			
3	Output data	--	Nbytes
Frame data bit			
N+3	Check bit	-----	2bytes

11) Adapter Data

Index value	data type	value	data size
0	Frame head	0x31	1byte
1	Data length	N	2bytes,n=input data size(Set configuration data)
Frame data bit			
3	Input data	--	Nbytes
N+3	Check bit	-----	2bytes

12) Adapter Responses

Index value	data type	value	data size
0	Frame head	0xF0	1byte
1	Data length	6	1byte
Frame data bit			
2	Adapter state and mode	-----	2bytes (state and mode form)
4	error code	-----	2bytes (error code form)
6	Protocol type	-----	1byte (Protocol type form)
7	version number		1bytes
8	Check bit	-----	2bytes

State and Mode form:

Bit	Description	
0	0=No error	1=Check error
1-2	00=standby mode	
	01=Configuration mode	
	10=hanging mode	
	11=run mode	
3	0=Ethernet link state is unknown	Ethernet link state
4*	0=NO connection with Ethernet	1=connection with Ethernet OK
5**	0=NO IP address	1=IP address OK
6**	0=network error	1=network OK

7**	0=NO connection with PLC	1= connection with PLC OK
8	0=I/O data is Big-endian format	1=I/O data is Small-endian format
9-15	Retain(0)	

Protocol type form:

type	Protocol
1	Reverse
2	PROFINET
3	Reverse
4	Reverse
5-255	Reverse

Error code form:

Bit	Error	Possible Reason
0	Incomplete configuration	1.Device is not added 2.Item is not added
1	Device/Item/Basket Add error	1.There is no device in the configuration data 2.Device added success 3.Device added error 4.Device added error in the Basket 5.There is no item in the configuration data 6.Item added error 7.Item added error in the Basket 8.The maximum number of Items have been added 9.Basket added success 10.There is no Basket in the configuration data 11.The basket in the configuration data does not contain items or devices. 12.Items in the basket contain invalid data
2	No network configuration	No network configuration data is received
3	Network configuration error	1.invalid address mode selection 2.invalid IP address 3.Invalid DNS address 4.invalid gateway address 5.The length of host name is 0 bytes 6.The length of host name is more than 63 bytes 7.The host name contains invalid characters
4	Frame check error	Error frame CRC check

5	Frame error	<ol style="list-style-type: none"> 1.Frame ID can not be identified 2.Error mode send frame 3.Do not support sending and receiving host socket commands or socket data 4.The frame data size is inconsistent with the frame type 5.The frame data is too big
6	Refuse to start a request	<ol style="list-style-type: none"> 1.bus is not configured 2.Invalid bus configuration 3.no network configuration 4.invalid network configuration data 5.No network 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 error	<ol style="list-style-type: none"> 1. Send the added basket frame after the device has been added 2.Send the added basket frame after the item has been added 3.Send the added device frame after the basket has been added 4.Send the added item frame after the basket has been added

13) Adapter Event

Index value	data type	value	data size
0	Frame head	0xF1	1byte
1	Data length	N+3	1byte
Frame data bit			
2	Module mode	----	1byte(0 =Standby,1 =configure 2=hang,3=run)
3	Module event code	----	2bytes (Module event code form)
5	Module event information	----	N bytes (Module event information form)
N+5	Check bit	----	2bytes

Module event code form:

value	Event name	value	Description
1	Ethernet link establishment	0	Cable insert or link detected
2	Ethernet disconnection	0	Cable disconnection
3	Network connection establishment	17	Network connection Successfully
4	PLC connection establishment	0	PLC connection establishment with adapter
5	PLC connection disconnection	0	PLC stop response
6	PLC reset	1	PLC control adapter reset
>6	Retain		

Module event information form:

Event name	size	Offset value	information
Network connection establishment	17	0	IP address mode(static = 0,DHCP=1)
		1	IP address(Big-endian)
		5	Subnet mask(Big-endian)
		9	gateway address(Big-endian)
		13	DNS server address(Big-endian)
PLC reset	1	0	Reset mode(mode0,mode1)

14) Open Socket

Index value	data type	value	data size
0	Frame head	0x40	1byte
1	Data length	8	1byte
Frame data bit			
2	Socket	-----	1byte(0 =Client,1 =server)
3	Socket mode	-----	1bytes (0=TCP,1=UDP)
4	Port number	-----	2bytes
6	IP address	-----	4bytes(MSB is MS octet)

10	Check bit	-----	2bytes
----	-----------	-------	--------

15) Adapter Socket Responses

Index value	data type	value	data size
0	Frame head	0xE0	1byte
1	Data length	5	1byte
Frame data bit			
2	Socket handle	-----	1byte
3	Socket state	-----	2bytes (socket state form)
5	Socket error	-----	2bytes (socket error form)
10	Check bit	-----	2bytes

Socket state form:

Bit	State
0	Monitor(UDP Client/server)
1	connecting (TCP Client)
2	Receiving (TCP server)
3	Already connected (TCP Client/server)
4	Crash(Should be closed)
5	Disconnect(TCPserver)

Socket error form:

Bit	Error name	Possible Reason
0	Socket inexistence	Socket unopened
1	Open socket failed	The module does not support the specified socket type The host specifies an invalid IP address. Select invalid client/server side Select invalid TCP/UDP Monitor socket failed
2	Invalid socket options	The host specifies an unrecognized socket

3	Invalid socket options data	The host specifies invalid socket options data. The host specifies too much or too little socket option data
4	Failed to send data	The host specifies too much socket data The host specifies too little socket data Socket unopened
5	close failed	The host tries to close one parent socket

16) Request socket status

Index value	data type	value	data size
0	Frame head	0x42	1byte
1	Data length	1	1byte
Frame data bit			
2	Socket Handle	--	1byte
3	Check bit	-----	2bytes

17) Set socket option

Index value	data type	value	data size
0	Frame head	0x41	1byte
1	Data length	N+2	1byte
Frame data bit			
2	Socket Handle	-----	1byte
3	Socket option	-----	2bytes(socket option form)
4	Socket option data	-----	N bytes(socket Option form domination)
N+4	Check bit	-----	2bytes

Socket Option form:

value	Socket option	value	Function
0	Enable broadcast	1	Enable or forbid broadcast message
1	Multiplexing address	1	Bind an already used port
2	Service type	1	Set Type Of Service or Traffic Class field in IP

header			
3	LAN broadcast	1	Enables or disables use of the limited broadcast address. If disabled, limited broadcast address is replaced with interface broadcast address.
4	Increase multicast permissions	4	Join the specified multicast group
5	Remove multicast permissions	4	Leave the specified multicast group
6	TCP No delay	1	Enable or forbid the use of Nagle algorithm
7	Set TCP to keep the connection	1	start and set to keep querying time intervals

18) send socket data

Index value	data type	value	data size
0	Frame head	0x50	1byte
1	Data length	N+1	2bytes
Frame data bit			
3	Socket Handle	-----	1byte
4	Socket data	-----	N bytes
N+4	Check bit	-----	2bytes

19) Adapter socket Data

Index value	data type	value	data size
0	Frame head	0x51	1byte
1	Data length	N+1	2bytes
Frame data bit			
3	Socket option	-----	1byte
4	Socket data	-----	N bytes
N+4	Check bit	-----	2bytes

20) Adapter Socket Event

Index value	data type	value	data size
0	Frame head	0xE1	1byte
1	Data length	N+4	2bytes
Frame data bit			
2	Socket Handle	-----	1byte
3	Socket state	-----	2bytes(socket state form)
5	Socket event code	-----	1byte(socket event form)
6	Socket event information	-----	N bytes(Specified event type)
N+6	Check bit	-----	2bytes

21) Close socket

Index value	data type	value	data size
0	Frame head	0x43	1byte
1	Data length	1	1byte
Frame data bit			
2	Socket Handle	-----	1byte
3	Check bit	-----	2bytes



4 Items and Baskets

This information describes the input and output data size of the module contained in the project. If the host software developer wants to create a project logic group, he or she can add items to a basket. Once the item (basket) has been created, use the configuration tool to generate a configuration file. The host reference configuration stage, this configuration file can be loaded into the module's non-volatile memory. If the host add all the required items by adding a basket, which contains the network I/O data of host communication , it will be stored in order to the project to be added to the basket using the configuration tool. If the host add all required items by repeatedly sending the Add Item frame, the network I/O data exchanged with the host will be in order to the already added Item.

Note:

- 1.A basket with only one item is perfectly acceptable
- 2.Item ID and basket ID are arbitrary values, which are determined by the developer
- 3.There are multiple baskets in a configuration file, but only one basket can be used in a single time.
- 4.Multiple items can export a configuration file, but not all items are added at runtime by the host

4.1 PROFINET Items

For PROFINET devices, we will add three items: digital I/O data, analog I/O data, and 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

Note: Module and sub module IDs use: when creating a GSDML file that the project needs to match, the project location must be in the same slot number as the PLC network configuration.

As shown below, the most important 2 bytes of the host data frame is in the 6 bytes payload, which is the 2 bytes input data in Item 1, and the other 4 bytes is the input data of item 2. Similarly, the most important 2 bytes of module data frame is in the 8 bytes payload, which is the output data in item 1, and the next 4 bytes are output data in item 2 and the most important 2 bytes data in Item 3 .Note that the Host Data frame and Adapter Data frame data sequence settings are based on the set order, the items are added to the basket or the items are added to the basket by the Host in order.

