



# Pronghorn

## **Gesnp Module**

Author: Joe Teixeira

Date: 28-Apr-04

Revision: 0.1



# Table of Contents

|                             |   |
|-----------------------------|---|
| OVERVIEW .....              | 1 |
| FUNCTIONALITY .....         | 1 |
| NETWORK CONFIGURATION ..... | 2 |
| MAPPING CONFIGURATION ..... | 4 |
| STATUS COUNTERS .....       | 5 |
| OTHER MESSAGES .....        | 6 |



## Overview

This document details the configuration interface for implementing a Pronghorn Gesnp module driver installation. This document assumes the reader is familiar with the Liaison's Pronghorn product. Refer to the *Pronghorn Users Guide* for more detailed information.

The main purpose of this module driver, named *gesnp*, is to provide connectivity to a GE Fanuc PLC (or compatible device) via the SNP protocol. This adheres to the protocol specification as detailed in the *Series 90 PLC SNP Communications User's Guide*, publication GFK-0529C.

## Functionality

The module driver's main purpose is to obtain or provide data contained in any of the Series 90 *data segments*, to or from a compatible device on the Gesnp network. The module driver can emulate a Master or Slave device on the network. This is controlled by the mapping configuration parameter *poll command*. Unsolicited command types (un, ur, uw) identifies it as a Slave device while solicited commands (r, ra, w, wu, wc) identifies it as Master device. It must be either a master or a slave and not both. Therefore mixing unsolicited with solicited commands will result in an error.

## Data Segments

There are fourteen data segment types identified in a Series 90 PLC, of which this module driver supports most.

|                    | <i>Segment</i>       | <i>Identifier</i>    | <i>Width</i> | <i>Code</i> |
|--------------------|----------------------|----------------------|--------------|-------------|
| Standard           | Discrete Inputs      | %I                   | 1 bit        | 46h         |
|                    | Discrete Outputs     | %Q                   | 1 bit        | 48h         |
|                    | Discrete Temporaries | %T                   | 1 bit        | 4Ah         |
|                    | Discrete Internals   | %M                   | 1 bit        | 4Ch         |
|                    | Discrete System A    | %SA                  | 1 bit        | 4Eh         |
|                    | Discrete System B    | %SB                  | 1 bit        | 50h         |
|                    | Discrete System C    | %SC                  | 1 bit        | 52h         |
|                    | Discrete System      | %S                   | 1 bit        | 54h         |
|                    | Genius Global Data   | %G                   | 1 bit        | 56h         |
|                    | Analog Inputs        | %AI                  | 16 bits      | 0Ah         |
|                    | Analog Outputs       | %AQ                  | 16 bits      | 0Ch         |
|                    | Registers            | %R                   | 16 bits      | 08h         |
|                    | Special <sup>i</sup> | Local Sub-block Data | %L           | 16 bits     |
| Program Block Data |                      | %P                   | 16 bits      | 04h         |

The maximum address offset for each data type depends on the model of CPU and memory configuration of the PLC. The pronghorn module driver supports up to a maximum offset of 65535.

<sup>i</sup> These segments types are only supported by a 90-70 PLC. Which use special commands not supported by this module.

## Message Types

In total the SNP protocol contains 8 different messages types that can be supported by any device. However this module driver only supports a sub-set of these:

| <i>Type</i> | <i>Description</i>          | <i>Support</i> |
|-------------|-----------------------------|----------------|
| 41h         | Attach message              | Yes            |
| 52h         | Attach response             | Yes            |
| 4Dh         | Mailbox request             | Yes            |
| 54h         | Text buffer message         | Yes            |
| 42h         | Block transfer message      | No             |
| 43h         | Connection data message     | No             |
| 55h         | Update real-time connection | No             |
| 49h         | Inquiry message             | No             |

The real workhorse of the protocol is the Mailbox request. Which is broken down into several service request codes. The following are a list of the supported service request codes:

| <i>Code</i> | <i>Description</i>     |
|-------------|------------------------|
| 00h         | Parameter Select       |
| 04h         | Read System Memory     |
| 07h         | Write System Memory    |
| 21h         | Change Privilege Level |

## Communication Interface

The SNP protocol requires that a serial interface be used for the physical communication layer. This layer is independent of the SNP application layer. Therefore configuration of the physical layer interface will greatly depend on the hardware being used. Options like RS-232, RS-485, RS-422 etc. will be configured via jumpers. Consult the hardware documentation for configuration options of the hardware being used.

## Network Configuration

The configuration of this module and its network interface is accomplished through the network comma separated values file (.csv) using software such as Excel or gnumeric. Some of the parameters are standard parameters used to configure the serial interface while others are particular to this module. Most parameters have a default value that will be used if the parameter does not exist. Additionally the parameters can be listed in any order.

| <i>Name</i> | <i>Valid Range</i>        | <i>Default</i> | <i>Description</i>   |
|-------------|---------------------------|----------------|--|
| CPU_ID      | String                    | Network Name   | Seven-byte description that identifies the module on the SNP network. The name of the network given to this module is used if no CPU_ID parameter is missing.          |
| Idle_Time   | 0 – 65535                 | 5000           | This is the connection idle timeout value (referred to as T3 time). Currently the module driver does not enforce the connection timeout. However it may in the future. |
| Password    | String                    | Null           | This the password used when changing the privilege level request. This maybe required depending on the configuration of the PLC device. (maximum 7 characters)         |
| Retries     | 0 - 9                     | 3              | This the number of times a transaction will re-transmitted upon the detection of a timeout   |
| Timeout     | 0 - 360000                | 1000           | The number of milliseconds to wait for a response  |
| Com_Port    | 1 - 16                    | 1              | The serial port interface to be used   |
| Baud        | 50 - 460800 <sup>ii</sup> | 9600           | The serial line speed being used.  |
| Stop_Bits   | 1 or 2                    | 1              | The number of stop-bits per transmitted character  |

<sup>ii</sup>Valid values are: 50, 75, 134, 200, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 56700, 115200, 230400, 460800. This depends on support of the hardware being used.

| <i>Name</i> | <i>Valid Range</i>     | <i>Default</i> | <i>Description</i>  |
|-------------|------------------------|----------------|---|
| Data_Bits   | 5 - 8                  | 8              | The number of data-bits per transmitted character         |
| Parity      | None, Even, Odd        | None           | The type of parity checking performed                     |
| Tx_Delay    | 0 – 65535              | 0              | The number milliseconds to wait before transmitting       |
| Handshake   | Yes or No              | No             | Specifies if hardware (CTS/RTS) handshaking is to be used |
| IRQ         | 0 – 16 <sup>iii</sup>  | 0              | The IRQ used to access the serial hardware.               |
| IO_Port     | 0 - 0xff <sup>iv</sup> | 0              | The port address used to access the serial hardware.      |

---

iiiA value of zero specifies to use default values for the given serial port being used. Typically COM1=IRQ3, COM2=IRQ4, COM3=IRQ3, COM4=IRQ4.

ivA value of zero specifies to use default values for the given serial port being used. Typically COM1=0x3e0, COM2=0x2e0, COM3=0x3f0, COM4=0x2f0.

## Errors

If any of the validation, for the network configuration parameters, fails a critical error message will be placed in the logs. In addition the module driver will not start. The following is a list of critical message that might be generated.

| <i>Msg ID</i> | <i>Mnemonic</i>  | <i>Description</i>  |
|---------------|------------------|---|
| 95            | CRT_BAD_COMPOR   | Invalid COM port (%d) Must be between 1 and %d<br>- The communication port specified is not in the proper range.<br>Must be numeric in the range of 1 - 16  |
| 96            | CRT_BAD_BAUD     | Invalid BAUD rate (%d)<br>- The communication speed specified is not one of the predefined values of: 50, 75, 134, 200, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 56700, 115200, 230400, 460800 |
| 97            | CRT_BAD_STOPBITS | Invalid number of stop bits (%d) must be 1 or 2<br>- The number of stop-bits specified is not one of 1 or 2.<br>Typically RTU modes use 1 stop bit  |
| 98            | CRT_BAD_PARITY   | Invalid parity (%s) must be 'None', 'Even' or 'Odd'<br>- The parity check specified is not one of the predefined mnemonics. Typically RTU modes use 'Even' parity                                     |
| 99            | CRT_DATABITS     | Invalid number of data bits (%d) must be between 5 and 8<br>- The number of data-bits specified is out of range. Typically RTU modes use 8 data bits  |
| 100           | CRT_BAD_TIMEOUT  | Invalid timeout value (%d) must be less than %d ms<br>- The timeout value specified is too large, it should be less than 360000 milliseconds  |

Additionally, some warnings may be issued. However modules execution will continue with default or corrected values.

| <i>Msg ID</i> | <i>Mnemonic</i>   | <i>Description</i>   |
|---------------|-------------------|--|
| 6001          | WRN_CPU_ID_LENGTH | The specified or defaulted CPU ID string is too long<br>- The CPU ID will be truncated to 7 characters |
| 6002          | WRN_PASSWD_LENGTH | The specified password string is too long<br>- The password will be truncated to 7 characters          |

## Mapping Configuration

The configuration of the module mappings is accomplished through the network comma separated values file (.csv) using software such as Excel or gnumeric. The parameters listed only pertain to this module. Refer to the *Pronghorn Users Guide* for a list of the other common parameters used to configure a mapping. Most parameters have a default value that will be used if the parameter does not exist. Additionally the parameters can be listed in any order.

| <i>Name</i> | <i>Valid Range</i>                            | <i>Default</i> | <i>Description</i>  |
|-------------|---|----------------|---|
| CPU_ID      | String  | Null           | The PLC (or device) name that identifies the slave to be accessed. This is only used when the module is emulating a master station. If left empty then a direct link (one slave) is assumed with the slave. |
| Segment     | Refer to <a href="#">Segments</a> description | None           | Identifies the data segment to access, within the slave device.   |
| Offset      | 1 - 65535                                     | 0              | This is the offset within the data segment. Unit values are referenced in number of widths for the specified <a href="#">segment type</a> .   |

## Errors

If any of the validation, for the mapping configuration parameters, fails an error message will be placed in the logs. In addition that particular mapping item will not be added to the internal mapping list. However the module continues with any remaining mappings.

| <i>Msg ID</i> | <i>Mnemonic</i>   | <i>Description</i>  |
|---------------|-------------------|---|
| 6003          | ERR_ADDRESS_RANGE | Offset is out of range (reg=%s%04d, len=%d)<br>- The specified <a href="#">Segment</a> offset value is out of range (in combination with the mapping length). |
| 6004          | ERR_BAD_SEGMENT   | Invalid segment type (%s)<br>- The specified segment does not point to a valid <a href="#">Segment</a> type.  |
| 6005          | ERR_BAD_ADDRESS   | No segment offset specified<br>- Enter a valid offset value   |
| 6006          | ERR_SLAVE_MODE    | Solicited not allowed in Slave mode<br>- The mode already configured for slave but a solicited poll command was found   |
| 6007          | ERR_MASTER_MODE   | Unsolicited not allowed for Master mode<br>- The mode already configured for Master but unsolicited poll command found  |

## Status Counters

This module driver maintains a set of status counter for recording the operational status of the module and the network.

| <i>Msg ID</i> | <i>Mnemonic</i> | <i>Description</i>  |
|---------------|-----------------|---|
| 6014          | CNT_MSG_SENT    | Number of Messages transmitted<br>- Keeps track of the number of transactions transmitted from this module. |
| 6015          | CNT_MSG_RECV    | Number of Messages received<br>- Keeps track of the number of transaction received for this module.         |
| 6016          | CNT_CMD_SENT    | Number of commands transmitted<br>- Keeps track of the number of command requests transmitted (Master mode) |
| 6017          | CNT_CMD_RECV    | Number of commands received<br>- Keeps track of the number of command requests received (Slave mode)        |
| 6018          | CNT_REPLY_SENT  | Number of replies transmitted<br>- Keeps track of the number of response packets transmitted (Slave mode)   |
| 6019          | CNT_REPLY_RECV  | Number of replies received<br>- Keeps track of the number of response packets received (Master mode)        |
| 6020          | CNT_ERR_SENT    | Number of errors sent<br>- Keeps track of the number of error response packets transmitted (Slave Mode)     |
| 6021          | CNT_ERR_RECV    | Number of errors received<br>- Keeps track of the number of error response packets received (Master mode)   |
| 6022          | CNT_ACK_SENT    | Number of ACKs transmitted<br>- Keeps track of the number of successful acknowledgements transmitted        |
| 6023          | CNT_ACK_RECV    | Number of ACKs received<br>- Keeps track of the number of successful acknowledgements received              |

| <i>Msg ID</i> | <i>Mnemonic</i>    | <i>Description</i>  |
|---------------|--------------------|---|
| 6024          | CNT_NAK_SENT       | Number of NAKs transmitted<br>- Keeps track of the number of negative acknowledgements transmitted                                |
| 6025          | CNT_NAK_RECV       | Number of NAKs received<br>- Keeps track of the number of negative acknowledgements received                                      |
| 6026          | CNT_RETRIES        | Number of retries<br>- Keeps track of the number of times a re-transmit had to be performed                                       |
| 6027          | CNT_REPLY_TIMEOUTS | Response timeouts<br>- Keeps track of the number of times no response was received from a slave                                   |
| 6028          | CRD_CHAR_SENT      | Number of characters transmitted<br>- The number of characters sent out the serial port (maintained by the hardware interface).   |
| 6029          | CRD_CHAR_RECV      | Number of characters received<br>- The number of characters received from the serial port (maintained by the hardware interface). |

## Other Messages

In the normal operation of this module certain types of messages may appear in the logs. The messages specific to this module are listed below. Refer to the *Pronghorn Users Guide* for a list of the common messages that can be seen in the logs.

| <i>Msg ID</i> | <i>Mnemonic</i>    | <i>Description</i>  |
|---------------|--------------------|---|
| 6008          | ERR_WRONG_RESPONSE | Message with wrong response (actual=0x%d, expected=0x%d)<br>- A response packet was received with the wrong <a href="#">message type</a> or acknowledgement (ACK=06h NAK=15h) |
| 6009          | ERR_WHILE_READING  | Error while processing network read (rc = %d)<br>- An error occurred when processing a solicited read command (refer to the error log and the return code - rc)               |
| 6010          | ERR_WHILE_WRITING  | Error while processing network write (rc = %d)<br>- An error occurred when processing a solicited write command (refer to the error log and the return code - rc)             |
| 6011          | ERR_UNREAD         | Error processing unsolicited Read (%s%d, len=%d)<br>- An unexpected error occurred while processing a read request (Slave mode)   |
| 6012          | ERR_UNWRITE        | Error processing unsolicited Write (%s%d, len=%d)<br>- An unexpected error occurred while processing a write request (Slave mode)   |
| 50            | ERR_BAD_RANGE      | The 'Privilege' value (%d) is out of range (%d - %d)<br>- A Change Privilege message was received with an invalid level number  |
| 51            | ERR_NOT_VALID      | Not valid<br>- A packet was received with an invalid source/destination ids<br>- A Change Privilege message was received with wrong password                                  |
| 52            | ERR_NOT_FOUND      | Did Not Find '%s' in 'CPU_ID'<br>- An attach message was received for a different CPU string other then was configured (slave mode)   |
| 52            | ERR_NOT_FOUND      | Did Not Find '%s' in 'CPU Device List'<br>- A attach response message was received from a slave device not found in our mapping configuration (master mode)                   |

| <i>Msg ID</i> | <i>Mnemonic</i>     | <i>Description</i>   |
|---------------|---------------------|--|
| 70            | ERR_BAD_SOH         | Start of message (header) bytes (0x%x) are not valid<br>- A packet was received with invalid start of header character (1Bh)   |
| 71            | ERR_BAD_EOH         | The End of message (trailer) bytes (0x%x) are not valid<br>- A packet was received with invalid end of header character (17h)  |
| 73            | ERR_NOT_HANDLED     | Command/Type 0x%x not handled in this context<br>- A packet with an unsupported Mailbox <a href="#">message request code</a><br>- A packet with an unsupported <a href="#">data segment code</a> |
| 74            | ERR_BAD_RESPONSE    | Command message 0x%x with response %d<br>- An error response was received with the specified Mailbox <a href="#">message request code</a>  |
| 75            | ERR_WRONG_RESPONSE  | Unexpected Response 0x%x, s/b 0x%x<br>- A packet was received with the wrong <a href="#">message type</a><br>- A packet was received with the wrong sequence number                              |
| 113           | ERR_CHECKSUM_FAILED | Checksum failed (actual=0x%x, expected=0x%x)<br>- A packet was received with actual CRC not matching the calculated CRC  |