



**Apstar**

Limited  
In Control since 1988

---

## ApStarnet Application Note 3.

Gem80 Primary using Table  
Exchange Free Running to  
ApStarnet in Multi Secondary  
Mode.

1. Revision History:

Revision	Date	Comment
001	20/09/2018	Initial Release

## Contents

1. Revision History:	2
2. Introduction	4
3. Link Specification	4
4. Gem80 - Primary	4
4.1 Port Configuration	4
4.2 Module Configuration	5
4.3 Transmission Control	5
4.4 Data Table Offsets	5
4.5 Tx/Rx Data Areas	6
5. ApStarnet – Multi-Secondary Addresses 1 to 5	6
5.1 Port Configuration	6
5.2 Transmission Control	7
5.3 Table Allocations	8
5.4 Tx/Rx Data Areas	8
6. Data Areas Transferred	9

## 2. Introduction

This document should be read in conjunction with the ApStarnet User Manual, ApStarnet Modbus Manual, Gem80 Starnet User Information (T464), and Application Note 1 “Typical Starnet Connections”.

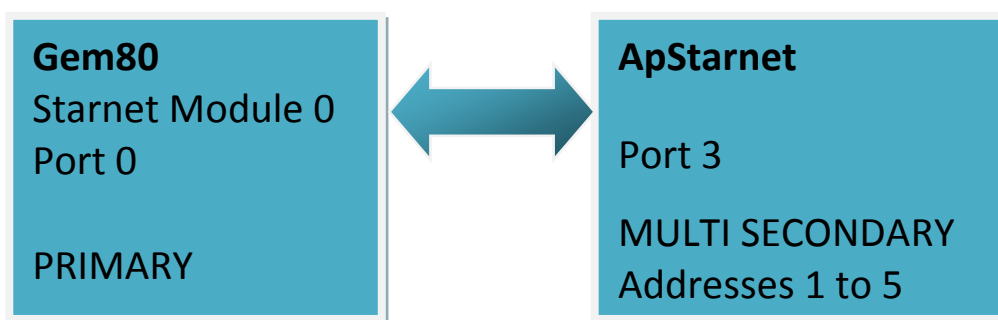
It provides details of the configuration of both sides of the link, shows what tables are used to control the link and where data is sent or received.

Connection details are not shown here; please see Application Note 1 “Typical Starnet Connections”.

## 3. Link Specification

This document covers the interconnection of Gem80 as Primary and ApStarnet as Secondary configured to look like multiple Secondary’s. The Channel type used will be HDLC at 180kbps NRZI encoding.

However all of the following is equally applicable to the other combinations of speed (48Kbps or 180kbps) and encoding (NRZI, FM0, and FM1). Note in order to run the link at other combinations changes must be made to the Channel Type in both the Gem80 configuration and ApStarnet Port configuration. Please note the restrictions in the use of 48kbps and 180kbps as covered in the Modbus Manual Section 6.1 “HDLC Limitations”.



## 4. Gem80 - Primary

The Gem80 Starnet User Information Manual (T464) should be read in conjunction with the following information. The Gem80 document gives detailed information and examples of the installation, setup, configuration and use of the Starnet Module in the Gem80 PLC.

### 4.1 Port Configuration

PLC logic in the Gem80 must be used to configure the Starnet Module by use of the T33-CONFIG Special Function. The following table shows how the configuration parameters should be set. Note each Gem80 can have up to 5 Starnet Module (0-4) and each Module can have either four ports (0-3) or two ports (0-1).

Configuration Port 0			
Address	ZM+0	0	Primary
Partners	ZM+1	5	Partner is five Secondary's
Channel Type	ZM+2	4	HDLC 180k NRZI
Tx Mode	ZM+3	-1	Overwrite
Rx Mode	ZM+4	-1	Overwrite
Mode	ZM+5	-1	HDLC Table Exchange Free Running
Partners/Scan	ZM+6	0	Partners/Scan

## 4.2 Module Configuration

The Gem80 uses N-Tables for received (Input) data, and O-Tables for send (Output) data. The amount of registers allocated to each of these tables is controlled by the setting of two P-Tables. Assuming that the Starnet module is the first Starnet in the Gem80 subrack (Module 0) then the two P-Tables used are P60 and P61. In this example they should be set as follows:

Table	Value	
P60	100	Table space per block
P61	5	Number of Partners

P60 defines the number of registers in each N-Table and O-Table. P61 defines the maximum number of partners. Note these two P-Tables are applicable to all the ports on the Starnet Module and are therefore set to the maximum numbers covering the use of all ports on the module. See Gem80 Starnet User Information (T464) for further explanation and examples.

## 4.3 Transmission Control

Although in this application the protocol is set to Free Running, the amount of data sent by the Primary to each Secondary is controlled by the setting of U-Tables. The control of information to the Secondary's is set and monitored by four U tables per Secondary. With P60/P61 set as above, the U Tables used to control and monitor data flow to the first Secondary are U0 to U3. The first table (U0) should be set to determine the quantity of O-Tables to be sent to the Secondary No.1.

Table	Value	Description
U0	100	No. of Tables to Transmit to Secondary Address 1
U4	100	No. of Tables to Transmit to Secondary Address 2
U8	100	No. of Tables to Transmit to Secondary Address 3
U12	100	No. of Tables to Transmit to Secondary Address 4
U16	100	No. of Tables to Transmit to Secondary Address 5

These values should be written into the U-Tables by the PLC Logic and should not exceed the maximum block size as declared by P60 in section 4.2 above.

Note the maximum permissible number of registers to be exchanged is 111, however using 100 registers makes the maths and register locations simpler.

## 4.4 Data Table Offsets

Each of the N-Tables, O-Tables and U-Tables used for each Port/Module configuration can be offset from N0, O0 and U0 by setting the offset value into P70, P71 and P72 (for Module 0).

In our application we have assumed that these P-Tables are all set to Zero

Table	Value	
P70	0	N - Table Offset
P71	0	O - Table Offset
P72	0	U - Table Offset

## 4.5 Tx/Rx Data Areas

Our application allows for the transmission of 100 registers from the Primary to each of the Secondary's, and 100 registers to be received from each of the Secondary's to the Primary.

Primary	Secondary
O0-99	To Secondary Address 1
O100-199	To Secondary Address 2
O200-299	To Secondary Address 3
O300-399	To Secondary Address 4
O400-499	To Secondary Address 5
N0-99	From Secondary Address 1
N100-199	From Secondary Address 2
N200-299	From Secondary Address 3
N300-399	From Secondary Address 4
N400-499	From Secondary Address 5

The received data N-Tables can be used in the PLC logic. The output O-Tables can be written to by the PLC Logic. Both tables can be monitored/written using a suitable Gem80 Programming package connected to the programming port of the Gem80.

## 5. ApStarnet – Multi-Secondary Addresses 1 to 5

The ApStarnet User Manual and Modbus Manual should be read in conjunction with the following information. These documents give detailed information of the installation, setup, configuration and use of the ApStarnet Module and Ports.

The configuration of the ApStarnet requires that registers be set inside the unit. To achieve this either the ApStarnet configuration software package running on a PC (connected via ModbusTCP or USB/RTU) or any other ModbusTCP Client (ModbusPoll, or PLC containing ModbusTCP Software) must be used.

Where the reference to TAB is made in the following text and tables, this refers to the display TAB of the ApStarnet Configuration Software package.

When writing to the ApStarnet by using an external ModbusTCP client, care must be taken to ensure the correct Modbus Unit Number is used as well as the correct Modbus Data Address. Modbus Unit Number 16 is used to read and set values which are applicable to the ApStarnet Unit. Modbus Unit Numbers 17,18,19,20 are used to read and set values which are applicable to one port, i.e. Unit 17=Port 1, Unit 18=Port 2, Unit 19=Port 3, Unit 20=Port 4.

### 5.1 Port Configuration

Our application here uses Port 3 of the ApStarnet, however for use of the other ApStarnet Ports, please note the restrictions in the use of 48kbps and 180kbps as covered in the Modbus Manual Section 6.1 "HDLC Limitations".

To avoid confusion and reduce the amount of data space allocated in the ApStarnet, all unused ports should be configured as:

TAB	Parameter	Setting
General	Port x Control	No Transmission
Port x	Port Address	1
	Link Partners	0
	Channel Type	Not in Use
	Interface Mode	HDLC or ESP Table Exchange Free Run
	PLC Timeout Action	Keep Running
	Transmit Mode	Overwrite
	Receive Mode	Overwrite

Port 3 (Modbus Unit 19) should be configured as follows:

TAB	Parameter	Setting	Note	Modbus	Value
General	Port 3 Control	Transmission enabled and auto enabled after power cycle	Activate Transmissions	Unit 16, 108	2
Port 3	Port Address	1	First Secondary Address	1001	1
	Link Partners	5	Number of Secondary's to act as	1002	5
	Channel Type	HDLC 180k NRZI		1003	4
	Transmit Mode	Overwrite		1004	-1
	Receive Mode	Overwrite		1005	-1
	Interface Mode	Multi Slave HDLC or ESP Table Exchange Free Run		1006	-3
	Not Used	Not Used		1007	0

Note, the Port Address of the ApStarnet Port does not have to start at one, For example setting the Port Address to 3 and the Quantity of Secondary's to act as to 4, the Port will send/receive data for Secondary Addresses 3, 4, 5 and 6, store the received data into memory areas 4103, 4104, 4105 and 4106 respectively, and transmit the send data from memory areas 4003, 4004, 4005 and 4006 respectively.

## 5.2 Transmission Control

The following table shows the Modbus Unit 19 Registers in the ApStarnet to control the number of registers to send back to the Primary from the Secondary's (Port 3).

Table	Value	Description
4204	100	No. of Tables to Transmit to Primary from Secondary Address 1
4208	100	No. of Tables to Transmit to Primary from Secondary Address 2
4212	100	No. of Tables to Transmit to Primary from Secondary Address 3
4216	100	No. of Tables to Transmit to Primary from Secondary Address 4
4220	100	No. of Tables to Transmit to Primary from Secondary Address 5

Note this should not exceed the maximum block size as defined in the Gem80 Primary using P60, see section 3.2 above.

### 5.3 Table Allocations

The ApStarnet uses a common internal memory area to store the data to/from all of its partners on each of its ports. If the configuration any of the ports changes then this area is reallocated dynamically.

For security, the ApStarnet will not send out any data (either on the Starnet side or the ModbusTCP side) until it has received data to be stored in that area. Therefore no data gets sent until it has been correctly received. In fact the send data memory areas are always marked as invalid until the ApStarnet receives data to put into them. Hence if the ApStarnet receives a request to read from a memory area on the ModbusTCP then it will respond with Invalid Address, until the memory area associated with that request has been written to by either the Starnet communication or from a ModbusTCP write registers command.

With the Module and Port configured as above, the number of data registers allocated to Port 3 as shown by Modbus Unit 16 Register 982 will be 1110.

Table	Value	Description
982	1110	No. of Tables allocated to Port 3 (Read Only)

Note it is possible to manually set the numbers of tables allocated to each port by writing into Modbus Unit 16 Addresses 700 to 979. However leaving these all set at zero causes the ApStarnet to allocate them at will and will declare them as necessary.

Table	Value	Description
700-979	0	Use automatic memory space allocation.

Note the Registers 840-909 are the ones used to allocate memory for Port 3.

### 5.4 Tx/Rx Data Areas

The ApStarnet uses Register Addresses 4001 to 4005 with a maximum quantity of 100 Registers (as previously configured see 4.2 above), as the Data area of the information to be sent back to the Primary from the Secondary's 1 to 5.

The ApStarnet uses Register Addresses 4101 to 4105 with a maximum quantity of 100 Registers (as previously configured see 4.2 above), as the Data area of the information which is received in the Secondary's 1 to 5 from the Primary.

When writing or reading from these areas with a ModbusTCP client, it is always important to keep the starting register the same e.g. 4001 (to 4005) for writing and 4101 (to 4105) for reading. The quantity of registers may vary but the start register must always be the same.

Always use a Write Multiple Registers (Modbus Command 16) to Address 4001 (to 4005) to transfer data to the ApStarnet to send on the Starnet Link to the Primary.

Always use a Read Holding Registers (Modbus Command 3) to Address 4101 (to 4105) to read data from the ApStarnet which has been received on the Starnet Link from the Primary.

For example:

To Send Data: 123,4,66,22,0,0 to the Primary from Secondary Address 1.

From your ModbusTCP Client issue the Modbus Command 16 with Address 4001, and data 123,4,66,22,0,0

If you then want to change the second value from 4 to let us say 44 then you must send another command but this time still starting from Address 4001, so the client should issue the Modbus Command 16 with Address 4001 and data 123,44,66,22,0,0



It is not allowed to send a single register write command to change one value in the data area, multiple register write commands must be used always starting at the address of the start of the memory area. Therefore Modbus Command 6 (write single register) to Address 4002 with data 44 to change the above value is NOT correct.

Similar rules apply to reading data from the receive memory area. Modbus Read Holding Register commands MUST always start from the first address of the memory area (in this case 4101 to 4105) and be of sufficient length to read the required amount of data. It is not allowed to read from anywhere else in the middle of the memory area.

## 6. Data Areas Transferred

The diagram below shows the transfer of data between the Primary and the Slave

