Overview

The following notes show how to map the I/O communication data when using a BM12x0 Host Interface module (Modbus TCP Interface).

- **Read Process Data (Output Data)**

  The "**Read Process Data**" are the output data that the PLC commands towards the scanner.

  They have to be written to the **HOLDING Registers (4x)**, from the address **40001** to the address **40256**.

  The Modbus TCP function to use is:
  - **"Read/Write Multiple Register"(code 23) function**, to read/write the "**Holding Registers(4x)**" area.

- **Write Process Data (Input Data)**

  The "**Write Process Data**" are the input data that come from the reader (scanned barcodes).

  They are available:
  - in the **INPUT Registers(3x)**, from the address **30001** to the address **30256**. and they are also available;
  - in the **HOLDING Registers(4x)**, from address **40257** to address **40512**.

  The Modbus TCP functions to use are:
  - **"Read Input Register"(code 4)** function, to read the "**Input Registers (3x)**" area;
  - **"Read Holding Register"(code 3)** function, to read the "**Holding Registers (4x)**" area
  - **"Read/Write Multiple Register"(code 23)** function, to read/write the "**Holding Registers (4x)**" area

The situation as in table below:

Register type	Begin Address (symbolic)	End Address (symbolic)	Content
Input Register (3x)	**30001**	**30256**	**Input data** Write Process Data
Holding Register (4x)	**40001**	**40256**	**Output data** Read Process Data
Holding Register (4x)	**40257**	**40512**	**Input data** Write Process Data

Functions & Offset:
the addresses above are "symbolic" addresses.
In order to get the register content through a dedicated reading function, we use the "offset" value.

Example:
- to read the INPUT Register area, we use the function **#4 "Read Input Registers"**.
Then, to read the first INPUT register (register 30001), we have to use that function with offset = 0;

- to read the HOLDING Register area, we use the function **#3 "Read Holding Registers"**.
Then, to read the first input data in the HOLDING Register area (register 40257), we have to use that function with offset = 256 (100 hex)

➢ Where do we get the data (Input Data)?

Where exactly can we read the barcode data?
As answer, see the following example:

The scanned bar code is **<stx>12345678<etx>** (10 bytes).

1) If NO flow control is enabled (Data Flow Control = **Disable)**

the data are available from register **30001** of the Input Registers, and from register **40257** of the Holding Registers.

Input Registers are as follows (I always assume the Big Endian format):
- 30001: 1 <stx> (offset = 0)
- 30002: 3 2
- 30003: 5 4
- 30004: 7 6
- 30005: <etx> 8 (offset = 4)

and for the **Holding Registers**
- 40257: 1 <stx> (offset = 100 hex)
- 40258: 3 2
- 40259: 5 4
- 40260: 7 6
- 40261: <etx> 8 (offset = 104 hex)

Warning: here the user needs to set the Input area size to at least the barcode length.
This means: Master Input Area Size = 10 (minimum).

2) If the Flow control is enabled (Data Flow Control = **DAD Driver**)

first 3 bytes are the header bytes for the flow control, then the data start at the register **30002** of the Input Registers area, and at the register **40257** of the Holding Register area

Input Registers are as follows :

- 30001:	<header2>	<header1>	(offset = 0)
- 30002:	<stx>	<header3>	
- 30003:	2	1	
- 30004:	4	3	
- 30005:	6	5	
- 30006:	8	7	
- 30007:	0	<etx>	(offset = 6)

and for the **Holding Registers**:

- 40257:	<header2>	<header1>	(offset = 100 hex)
- 40258:	<stx>	<header3>	
- 40259:	2	1	
- 40260:	4	3	
- 40261:	6	5	
- 40262:	8	7	
- 40263:	0	<etx>	(offset = 106 hex)

Warning: here the user needs to set the Input area size to the **barcode length + 3,** at least, in order to consider the header bytes.
This means: Master Input Area Size = 13 (minimum).

Note:
Some Schneider PLC CPUs are not able to access to the “Input Registers"(3x) area, because they do not include the “code 4” function. These PLCs need to access the “Holding Registers" (4x) area to read the Input data.

The BM12x0 module supports both access methods; any user application can access either the 3x area and the 4x area.

➢ Where do we write the commands (Output Data)?

Where exactly does the Modbus Master(ex: PLC) write the commands to the slave (ex: barcode scanner)?
As answer, see the following example:

An application needs to control the reading phase of the scanner through command strings sent from the PLC.

Actions:

1) the user sets the scanner Operating Mode as it starts reading when the character “STX” (02hex) has received;

2) the user sets the scanner Operating Mode as it stops reading when the character “ETX” (03hex) has received;

3) using the Modbus TCP function "Read/Write Multiple Register" (code 23), the user writes the “**Holding Registers (4x)**” area as following:

- to START the reading phase

- 40001: 00 **02** (offset = 0 hex)

- to STOP the reading phase

- 40001: 00 **03** (offset = 0 hex)

The scanned bar code is now available on the data area according to the previous example.