SIEMENS

Ingenuity for life

S7-200 Transmit and Receive (Freeport on RS485 / RS232)

S7-200 SMART V2.4

https://w3.siemens.co.in/automation/in/en/automation-systems/industrial-automation/s7-200-smart-

Siemens Industry Online Support

Legal information

Use of application examples

Application examples illustrate the solution of automation tasks through an interaction of several components in the form of text, graphics and/or software modules. The application examples are a free service by Siemens AG and/or a subsidiary of Siemens AG ("Siemens"). They are non-binding and make no claim to completeness or functionality regarding configuration and equipment. The application examples merely offer help with typical tasks; they do not constitute customer-specific solutions. You yourself are responsible for the proper and safe operation of the products in accordance with applicable regulations and must also check the function of the respective application example and customize it for your system.

Siemens grants you the non-exclusive, non-sub licensable and non-transferable right to have the application examples used by technically trained personnel. Any change to the application examples is your responsibility. Sharing the application examples with third parties or copying the application examples or excerpts thereof is permitted only in combination with your own products. The application examples are not required to undergo the customary tests and quality inspections of a chargeable product; they may have functional and performance defects as well as errors. It is your responsibility to use them in such a manner that any malfunctions that may occur do not result in property damage or injury to persons.

Disclaimer of liability

Siemens shall not assume any liability, for any legal reason whatsoever, including, without limitation, liability for the usability, availability, completeness and freedom from defects of the application examples as well as for related information, configuration and performance data and any damage caused thereby. This shall not apply in cases of mandatory liability, for example under the German Product Liability Act, or in cases of intent, gross negligence, or culpable loss of life, bodily injury or damage to health, non-compliance with a guarantee, fraudulent non-disclosure of a defect, or culpable breach of material contractual obligations. Claims for damages arising from a breach of material contractual obligations shall however be limited to the foreseeable damage typical of the type of agreement, unless liability arises from intent or gross negligence or is based on loss of life, bodily injury or damage to health. The foregoing provisions do not imply any change in the burden of proof to your detriment. You shall indemnify Siemens against existing or future claims of third parties in this connection except where Siemens is mandatorily liable.

By using the application examples you acknowledge that Siemens cannot be held liable for any damage beyond the liability provisions described.

Other information

Siemens reserves the right to make changes to the application examples at any time without notice. In case of discrepancies between the suggestions in the application examples and other Siemens publications such as catalogs, the content of the other documentation shall have precedence.

The Siemens terms of use (https://support.industry.siemens.com) shall also apply.

Security information

Siemens provides products and solutions with Industrial Security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens’ products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the Internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit https://www.siemens.com/industrialsecurity.

Siemens’ products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer’s exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at: https://www.siemens.com/industrialsecurity.

Table of contents

1 Introduction

1.1 Overview

User can use the Transmit (XMT) and Receive (RCV) instructions for communication between a S7-200 SMART CPU and other devices or S7-200 SMART through the CPU serial port(s). Each S7-200 SMART CPU provides an integrated RS485 port (Port 0). The standard CPUs additionally support an optional CM01 Signal Board (SB) RS232/RS485 port (Port 1). The communication protocol must be implemented in the user program..

User can select the Freeport mode to control the serial communications port of the CPU by means of your user program. When user select Freeport mode, program controls the operation of the communications port through the use of the receive interrupts, the transmit interrupts, the Transmit instruction, and the Receive instruction and entirely controls the communications protocol while in Freeport mode.

1.2 Components used

This application example has been created with the following hardware and software components:

Table 1-1

Component	Number	Article number	Note
CPU ST30	2	6ES7288-1ST30-0AA0	Firmware Ver. 2.3
STEP 7-MicroWIN SMART V2.3	1	6ES7-288S-W01-0AA0	
Ethernet Switch	1	-	

This application example consists of the following components:

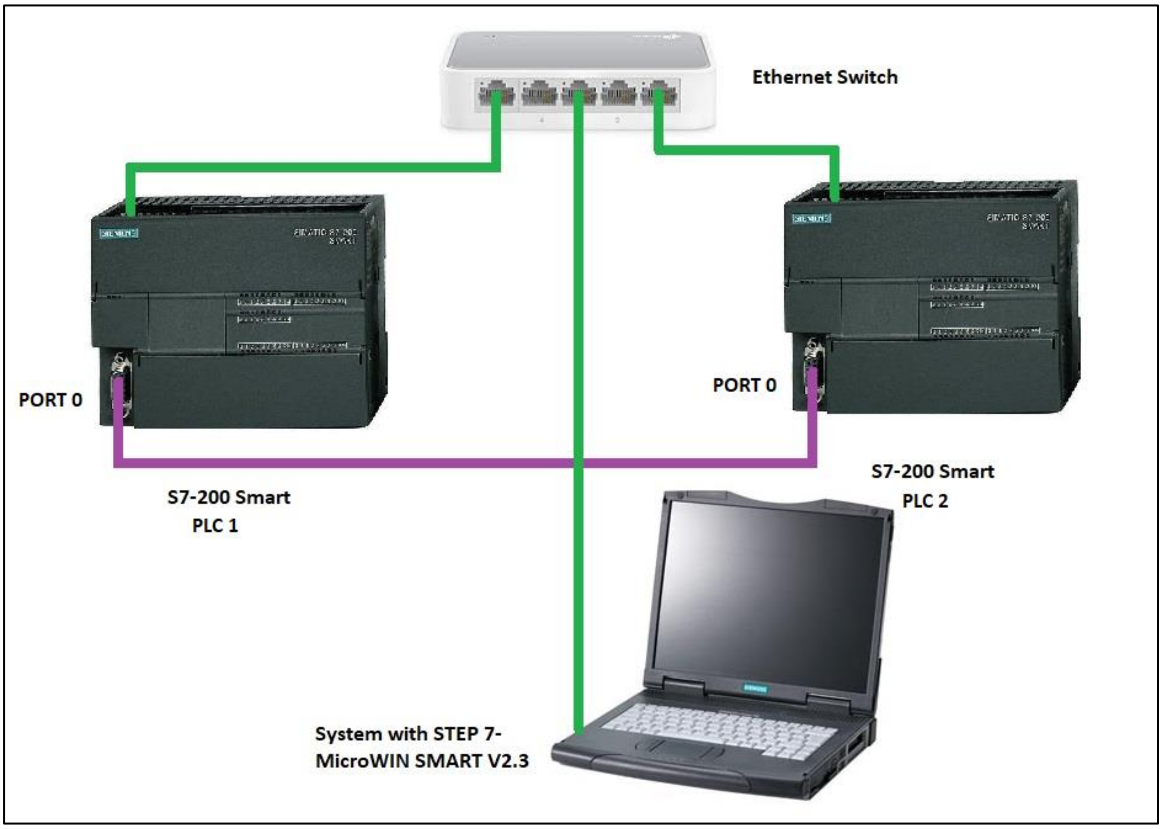
Table 1-2

Component	File name	Note
S7-200 Smart Freeport Transmit and Receive	1) PLC 1 (1st transmit then receive).smart 2) PLC 2 (1st receive then transmit).smart	

2 Engineering

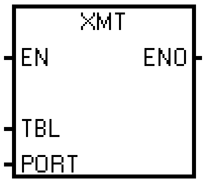
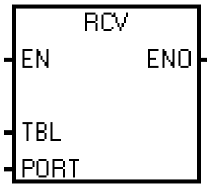
2.1 Hardware setup:

The figure below shows a schematic overview of the most important components of the solution:



3 Engineering

3.1 Description of instructions

LAD / FBD	STL	Description
	`XMT TBL, PORT`	The Transmit instruction (XMT) is used in Freeport mode to transmit data by means of the communications port(s).
	`RCV TBL, PORT`	The Receive instruction (RCV) initiates or terminates the receive message function. User must specify a start and an end condition for the receive box to operate. Messages received through the specified port (PORT) are stored in the data buffer (TBL). The first entry in the data buffer specifies the number of bytes received.

Input / output	**Data type**	**Operand**
TBL	BYTE	IB, QB, VB, MB, SMB, SB, *VD, *LD, *AC
PORT	BYTE	Constant: 0 or 1 Note: The two available ports are as follows: • Integrated RS485 port (Port 0), • CM01 Signal Board (SB) RS232/RS485 port (Port 1)

You can select the Freeport mode to control the serial communications port of the CPU by means of your user program. When you select Freeport mode, your program controls the operation of the communications port through the use of the receive interrupts, the transmit interrupts, the Transmit instruction, and the Receive instruction and entirely controls the communications protocol while in Freeport mode. You use SMB30 and SMB130 to select the baud rate and parity.

The CPU assigns two special memory bytes to the two physical ports:

1) SMB30 to the integrated RS485 port (Port 0)

2) SMB130 to the CM01 RS232/RS485 Signal Board (SB) port (Port 1)

The Freeport mode is disabled and normal communications are re-established (for example, HMI device access) when the CPU is in STOP mode.

In the simplest case, User can send a message to a printer or a display using only the Transmit (XMT) instruction. Other examples include a connection to a bar code reader, a weigh scale, and a welder. In each case, user must write program to support the protocol that is used by the device with which the CPU communicates while in Freeport mode.

User can only use Freeport communications when the CPU is in RUN mode. Enable the Freeport mode by setting a value of 01 in the protocol select field of SMB30 (Port 0) or SMB130 (Port 1). While in Freeport mode, it cannot communicate with an HMI on the same port.

3.2 Project integration

1. Changing PPI communications to Freeport mode.

SMB30 and SMB130 configure the communications ports, 0 and 1 respectively, for Freeport operation and provide selection of baud rate, parity, and number of data bits. The following figure describes the Freeport control byte. One stop bit is generated for all configurations.

MSB 7							LSB 0
p	p	d	b	b	b	m	m

pp	Parity select		d	Data bits per character	
	00 =	No parity		0 =	8 bits per character
	01 =	Even parity		1 =	7 bits per character
	10 =	No parity			
	11 =	Odd parity			
bbb	Freeport baud rate		mm	Protocol selection	
	000 =	38400		00 =	PPI slave mode
	001 =	19200		01 =	Freeport mode
	010 =	9600		10 =	Reserved (defaults to PPI slave mode)
	011 =	4800		11 =	Reserved (defaults to PPI slave mode)
	100 =	2400			
	101 =	1200			
	110 =	115200			
	111 =	57600			

2. Transmit data.

The Transmit instruction lets you send a buffer of one or more characters, up to a maximum of 255. The following figure shows the format of the Transmit buffer.

x	M	E	S	S	A	G	E
①	②						

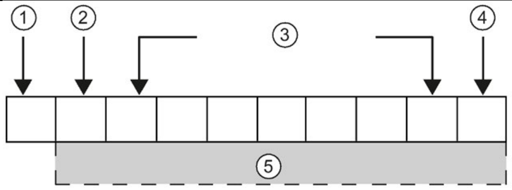
① Number of bytes to transmit
② Characters of the message

If an interrupt routine is attached to the transmit complete event, the CPU generates an interrupt (interrupt event 9 for port 0 and interrupt event 26 for port 1) after the last character of the buffer is sent.

You can transmit without using interrupts (for example, sending a message to a printer) by monitoring SM4.5 (port 0) or SM4.6 (port 1) to signal when transmission is complete.

You can use the Transmit instruction to generate a BREAK condition by setting the number of characters to zero and then executing the Transmit instruction. This generates a BREAK condition on the line for 16-bit times at the current baud rate. Transmitting a BREAK is handled in the same manner as transmitting any other message, in that a Transmit interrupt is generated when the BREAK is complete and SM4.5 or SM4.6 signals the current status of the Transmit operation.

3. Receive data.



The Receive instruction lets you receive a buffer of one or more characters, up to a maximum of 255. The following figure shows the format of the Receive buffer.

① Number of bytes received (byte field)
② Start character
③ Message
④ End character
⑤ Characters of the message

If an interrupt routine is attached to the receive message complete event, the CPU generates an interrupt (interrupt event 23 for port 0 and interrupt event 24 for port 1) after the last character of the buffer is received.

You can receive messages without using interrupts by monitoring SMB86 (port 0) or SMB186 (port 1). This byte is non-zero when the Receive instruction is inactive or has been terminated. It is zero when a receive is in progress.

As shown in the following table, the Receive instruction allows you to select the message start and message end conditions, using SMB86 through SMB94 for port 0 and SMB186 through SMB194 for port 1.

4. Receive buffer format (SMB86 to SMB94, and SMB186 to SMB194)

Port 0	Port 1	Description
SMB86	SMB186	Receive message status byte MSB 7 … LSB 0 n | r | e | 0 | 0 | f | c | p n: 1 = Receive message function terminated; user issued disable instruction. r: 1 = Receive message function terminated; error in input parameters or missing start or end condition. e: 1 = End character received. t: 1 = Receive message function terminated; timer expired. c: 1 = Receive message function terminated; maximum character count achieved. p: 1 = Receive message function terminated; a parity error.

<table>
<tr><td>SMB87</td><td>SMB187</td><td>Receive message control byte
<table>
<tr><td>MSB
7</td><td></td><td></td><td></td><td></td><td></td><td></td><td>LSB
0</td></tr>
<tr><td>en</td><td>sc</td><td>ec</td><td>il</td><td>c/m</td><td>tmr</td><td>bk</td><td>0</td></tr>
</table>
en:
0 = Receive message function is disabled.
1 = Receive message function is enabled.
The enable/disable receive message bit is checked each time the RCV instruction is executed.
sc:
0 = Ignore SMB88 or SMB188.
1 = Use the value of SMB88 or SMB188 to detect start of message.
ec:
0 = Ignore SMB89 or SMB189.
1 = Use the value of SMB89 or SMB189 to detect end of message.
il:
0 = Ignore SMB90 or SMB190.
1 = Use the value of SMB90 or SMB190 to detect start of message.
c/m:
0 = Timer is an inter-character timer.
1 = Timer is a message timer.
tmr:
0 = Ignore SMW92 or SMW192.
1 = Terminate receive if the time period in SMW92 or SMW192 is exceeded.
bk:
0 = Ignore break conditions.
1 = Use break condition as start of message detection.</td></tr>
<tr><td>SMB88</td><td>SMB188</td><td>Start of message character.</td></tr>
<tr><td>SMB89</td><td>SMB189</td><td>End of message character.</td></tr>
<tr><td>SMW90</td><td>SMW190</td><td>Idle line time period given in milliseconds. The first character received after idle line time has expired is the start of a new message.</td></tr>
<tr><td>SMW92</td><td>SMW192</td><td>Inter-character/message timer time-out value given in milliseconds. If the time period is exceeded, the receive message function is terminated.</td></tr>
<tr><td>SMB94</td><td>SMB194</td><td>Maximum number of characters to be received (1 to 255 bytes). This range must be set to the expected maximum buffer size, even if the character count message termination is not used.</td></tr>
</table>

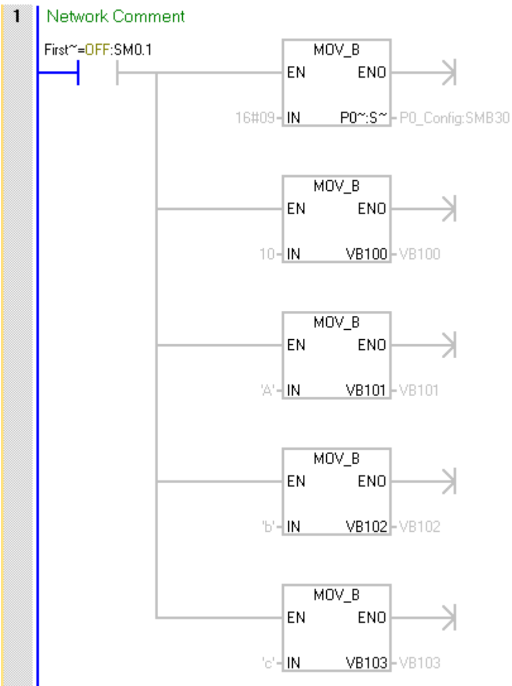
Please refer to the S7-200 SMART System Manual section 7.3.2 for details.

3.3 Operation.

In this exercise, we are going to use 2 PLCs for testing freeport of S7-200 SMART. For testing freeport we had to use XMT and RCV instructions in same program. So, we had used PLC 1 for transmitting and receiving string from PLC 2.

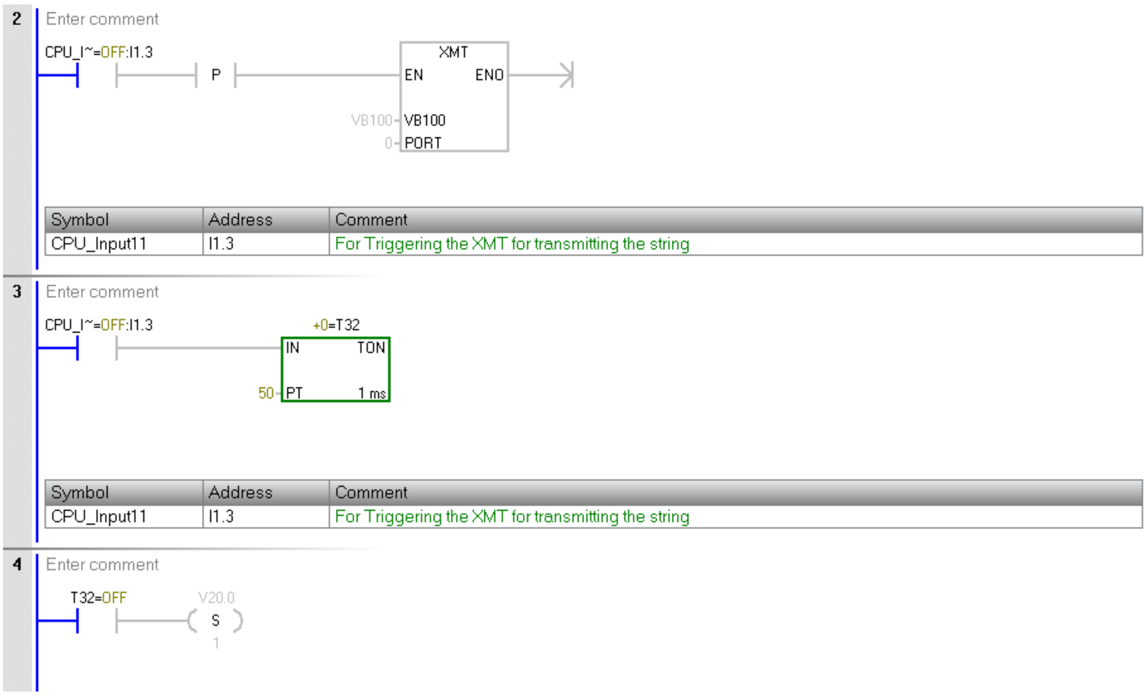
PLC 1 is used for transmitting the data string first with pre-defined start and end characters. Whereas PLC 2 receives the transmitted string from PLC 1. And then PLC 2 transmits data string and PLC 1 receives.

1. Let's Understand 'PLC 1 (1st transmit then receive).smart' . On first scan we initialize freeport (9600 baud rate, 8 data bits, no parity).



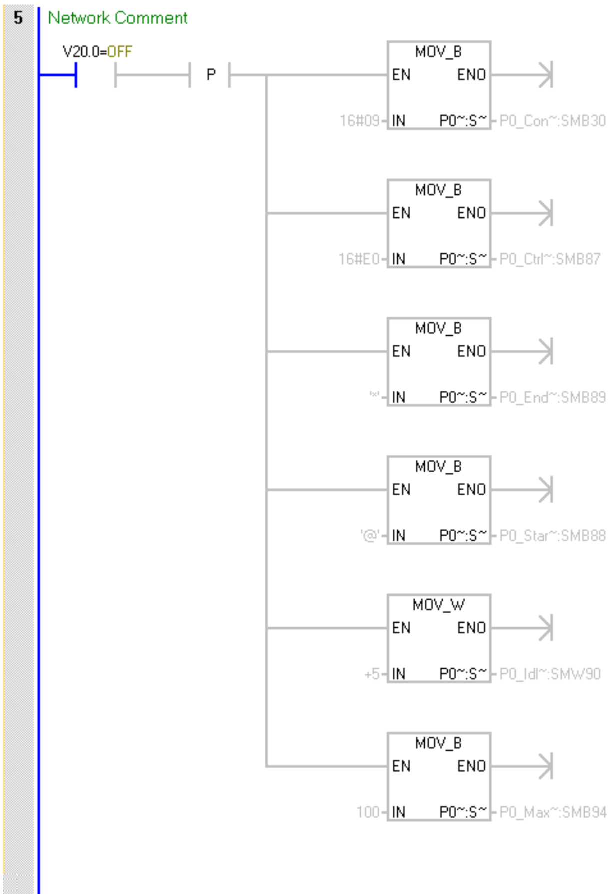
Symbol	Address	Comment
First_Scan_On	SM0.1	ON for the first scan cycle only
P0_Config	SMB30	Configure port 0 communication: parity, number of data bits per character, baud rate, and protocol

2. As soon as we set high the input I1.3, it triggers the XMT instruction which transmit string from VB100 on port 0. After 50ms timer we sets V20.0 for receive RCV instruction.



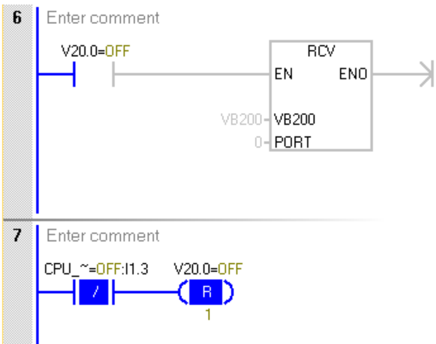
Note : Whenever you need to transmit the data (or trigger the XMT instruction) every time you have to trigger input I1.3 to high.

3. Here V20.0 enables the initialization for receiving (RCV instruction). Start character as ‘@’ and end character as ‘*’ with maximum 100-character limit. Idle line timeout set to 5 ms.



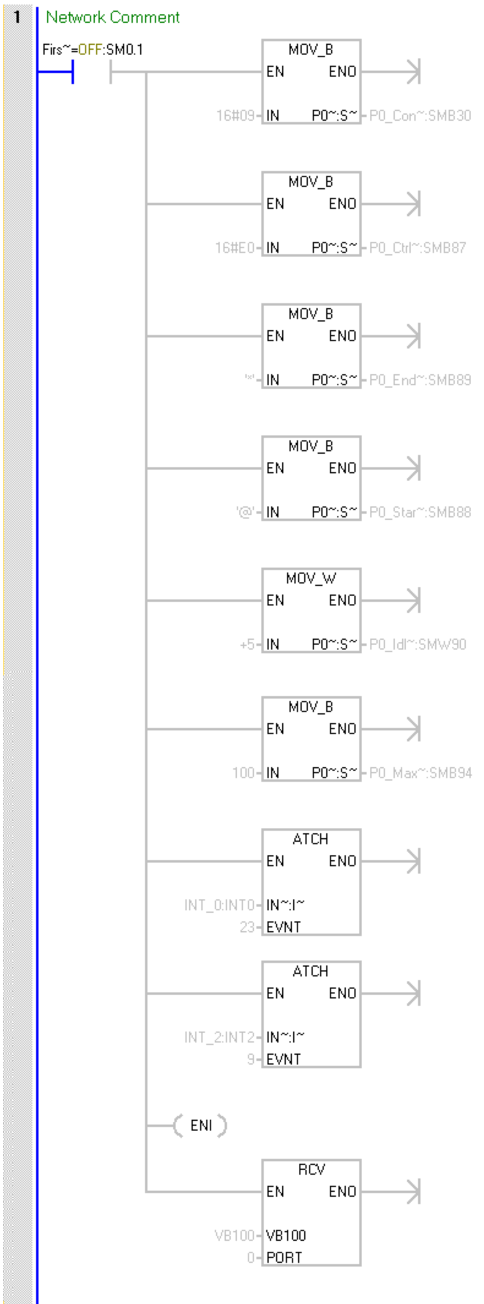
Symbol	Address	Comment
P0_Config	SMB30	Configure port 0 communication: parity, number of data bits per character, baud rate, and protocol
P0_Ctrl_Rcv	SMB87	Receive message control
P0_End_Char	SMB89	End of message character
P0_Idle_Time	SMW90	Idle line time period given in milliseconds
P0_Max_Char	SMB94	Maximum number of characters to be received (1 to 255 bytes)
P0_Start_Char	SMB88	Start of message character

4. Here it enables the receiving on port 0 with buffer at VB200. After I1.3 goes down V20.0 resets and will be ready for next trigger.



Symbol	Address	Comment
CPU_Input11	I1.3	For Triggering the XMT for transmitting the string

5. Here we start with 2nd PLC i.e. 'PLC 2 (1st receive then transmit).smart'. On first scan we initialize freeport (9600 baud rate, 8 data bits, no parity). Start character as '@' and end character as '*' with maximum 100-character limit. Idle line timeout set to 5 ms. That means this PLC will be in receiving mode after power on.



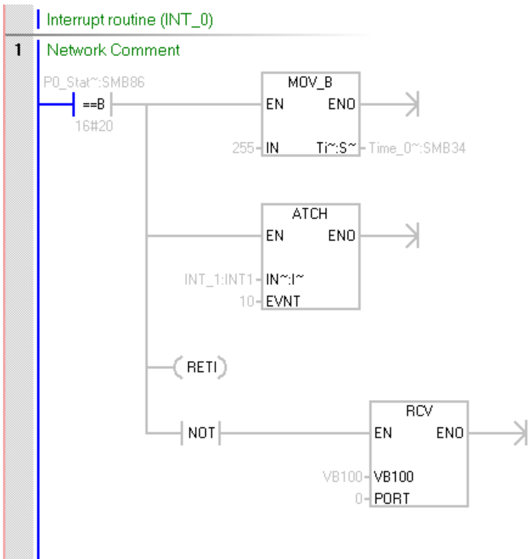
Symbol	Address	Comment
First_Scan_On	SM0.1	ON for the first scan cycle only
INT_0	INT0	Interrupt routine comments
INT_2	INT2	Interrupt routine comments
P0_Config	SMB30	Configure port 0 communication: parity, number of data bits per character, baud rate, and protocol
P0_Ctrl_Rcv	SMB87	Receive message control
P0_End_Char	SMB89	End of message character
P0_Idle_Time	SMW90	Idle line time period given in milliseconds
P0_Max_Char	SMB94	Maximum number of characters to be received (1 to 255 bytes)
P0_Start_Char	SMB88	Start of message character

Here we attach interrupt 0 to the Receive Complete event (event 23) and interrupt 2 to the Transmit Complete event (event 9) and whenever these events gets occurred PLC will enable user interrupt.

6. INT_0 (Receive complete interrupt routine):

   If receive status shows receive of end character, then attach a 255 ms timer to trigger a transmit and return.

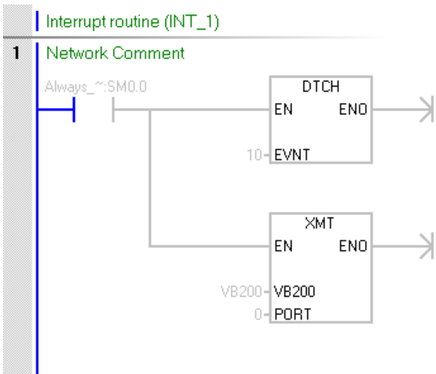
   If the receive completed for any other reason, then start a new receive.



Symbol	Address	Comment
INT_1	INT1	Interrupt routine (INT_1)
P0_Stat_Rcv	SMB86	Port 0 receive message status
Time_0_Intrvl	SMB34	Specifies time interval (in 1-ms increments from 5 to 255) of interrupt 0.

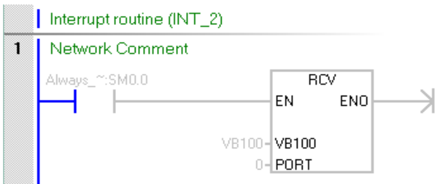
7. 255 ms Timer interrupt: it will detach timer interrupt and transmit data back to PLC 1 on port 0 from buffer VB200.

   For precaution we had taken 255 ms for transmitting that data from PLC 2 but PLC 1 had gone in receiving mode after 50 ms.



Symbol	Address	Comment
Always_On	SM0.0	Always ON

8. Transmit Complete interrupt (INT_2): After completing one cycle of receiving and transmitting data to PLC 2, PLC 1 again gets ready for receiving data from PLC 1.



Symbol	Address	Comment
Always_On	SM0.0	Always ON

9. PLCs status before input trigger:

‘PLC 1 (1st transmit then receive).smart’:

	Address	Format	Current Value	New Value
1	VB100	Unsigned	10	
2	VB101	ASCII	'@'	
3	VB102	ASCII	'1'	
4	VB103	ASCII	'2'	
5	VB104	ASCII	'3'	
6	VB105	ASCII	'*'	
7	VB106	ASCII	'$00'	
8	VB107	ASCII	'$00'	
9	VB108	ASCII	'$00'	
10	VB109	ASCII	'$00'	
11	VB110	ASCII	'$00'	
12	V20.0	Bit	2#0	

	Address	Format	Current Value	New Value
1	VB200	ASCII	'$00'	
2	VB201	ASCII	'$00'	
3	VB202	ASCII	'$00'	
4	VB203	ASCII	'$00'	
5	VB204	ASCII	'$00'	
6	VB205	ASCII	'$00'	
7	VB206	ASCII	'$00'	
8	VB207	ASCII	'$00'	
9	VB208	ASCII	'$00'	

‘PLC 2 (1st receive then transmit).smart’:

	Address	Format	Current Value	New Value
1	VB100	Signed	+0	
2	VB101	ASCII	'$00'	
3	VB102	ASCII	'$00'	
4	VB103	ASCII	'$00'	
5	VB104	ASCII	'$00'	
6	VB105	ASCII	'$00'	
7	VB106	ASCII	'$00'	
8	VB107	ASCII	'$00'	
9	SMB90	Unsigned	0	
10	P0_Stat_Rcv:SMB86	Binary	2#0000_0000	
11	P0_Ctrl_Rcv:SMB87	Hexadecimal	16#E0	
12	P0_Start_Char:SMB88	ASCII	'@'	
13	P0_End_Char:SMB89	ASCII	'*'	

	Address	Format	Current Value	New Value
1	VB200	Signed	+10	
2	VB201	ASCII	'@'	
3	VB202	ASCII	'0'	
4	VB203	ASCII	'9'	
5	VB204	ASCII	'8'	
6	VB205	ASCII	'*'	
7	VB206	ASCII	'$00'	
8	VB207	ASCII	'$00'	
9	VB208	ASCII	'$00'	
10	VB209	ASCII	'$00'	

The above 2 tables mentioned shows us the initial values data registers that has been used in transmission and reception before triggering the inputs.

In ‘Transmit 1.smart’ VB100 series is used for transmit and ‘Transmit 2.smart’ VB100 series is used to receive.

In ‘Transmit 2.smart’ VB200 series is used for transmit and ‘Transmit 1.smart’ VB200 series is used to receive.

10. PLCs status after input trigger:

'PLC 1 (1st transmit then receive).smart':

	Address	Format	Current Value	New Value
1	VB100	Unsigned	10	
2	VB101	ASCII	'@'	
3	VB102	ASCII	'1'	
4	VB103	ASCII	'2'	
5	VB104	ASCII	'3'	
6	VB105	ASCII	'*'	
7	VB106	ASCII	'$00'	
8	VB107	ASCII	'$00'	
9	VB108	ASCII	'$00'	
10	VB109	ASCII	'$00'	
11	VB110	ASCII	'$00'	
12	V20.0	Bit	2#0	

	Address	Format	Current Value	New Value
1	VB200	ASCII	'$00'	
2	VB201	ASCII	'@'	
3	VB202	ASCII	'0'	
4	VB203	ASCII	'9'	
5	VB204	ASCII	'8'	
6	VB205	ASCII	'*'	
7	VB206	ASCII	'$00'	
8	VB207	ASCII	'$00'	
9	VB208	ASCII	'$00'	

'PLC 2 (1st receive then transmit).smart':

	Address	Format	Current Value	New Value
1	VB100	Signed	+0	
2	VB101	ASCII	'@'	
3	VB102	ASCII	'1'	
4	VB103	ASCII	'2'	
5	VB104	ASCII	'3'	
6	VB105	ASCII	'*'	
7	VB106	ASCII	'$00'	
8	VB107	ASCII	'$00'	
9	SMB90	Unsigned	0	
10	P0_Stat_Rcv:SMB86	Binary	2#0000_0000	
11	P0_Ctrl_Rcv:SMB87	Hexadecimal	16#E0	
12	P0_Start_Char:SMB88	ASCII	'@'	
13	P0_End_Char:SMB89	ASCII	'*'	

	Address	Format	Current Value	New Value
1	VB200	Signed	+10	
2	VB201	ASCII	'@'	
3	VB202	ASCII	'0'	
4	VB203	ASCII	'9'	
5	VB204	ASCII	'8'	
6	VB205	ASCII	'*'	
7	VB206	ASCII	'$00'	
8	VB207	ASCII	'$00'	
9	VB208	ASCII	'$00'	
10	VB209	ASCII	'$00'	

The above 2 tables mentioned shows us the initial values data registers that has been used in transmission and reception after triggering the inputs.

4 Appendix

4.1 Service and support

Industry Online Support

Do you have any questions or need assistance?

Siemens Industry Online Support offers round the clock access to our entire service and support know-how and portfolio.

The Industry Online Support is the central address for information about our products, solutions and services.

Product information, manuals, downloads, FAQs, application examples and videos – all information is accessible with just a few mouse clicks:
support.industry.siemens.com

Technical Support

The Technical Support of Siemens Industry provides you fast and competent support regarding all technical queries with numerous tailor-made offers
– ranging from basic support to individual support contracts. Please send queries to Technical Support via Web form:
www.siemens.com/industry/supportrequest

SITRAIN – Training for Industry

We support you with our globally available training courses for industry with practical experience, innovative learning methods and a concept that's tailored to the customer's specific needs.

For more information on our offered trainings and courses, as well as their locations and dates, refer to our web page:
www.siemens.com/sitrain

Service offer

Our range of services includes the following:

- Plant data services
- Spare parts services
- Repair services
- On-site and maintenance services
- Retrofitting and modernization services
- Service programs and contracts

You can find detailed information on our range of services in the service catalog web page:
support.industry.siemens.com/cs/sc

Industry Online Support app

You will receive optimum support wherever you are with the "Siemens Industry Online Support" app. The app is available for Apple iOS, Android and Windows Phone:
support.industry.siemens.com/cs/ww/en/sc/2067

4.2 Application support

Siemens Ltd

RC-IN DF FA SUP

Thane Belapur Road
Thane 400601, India

Pre-sales Support

Email: rginslpresales-fa.in@siemens.com

4.3 Links and literature

Table 4-1

No.	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to this entry page of this application example https://w3.siemens.co.in/automation/in/en/automation-systems/industrial-automation/s7-200-smart-plc/pages/default.aspx
\3\	

4.4 Change documentation

Table 4-2

Version	Date	Modifications
V1.0	07/2019	First version