

Shared Data Access Programmed example

This is a listing of an example using a Siemens S7-315 PN/DP processor programmed in Step 7 V5.5. It demonstrates how a PLC program would access Shared Data variables over the PROFIBUS network.

Note that the IND780 Terminal is set up as a Floating Point device at node 3, with Shared Data Access Enabled.

HW Config - [SIMATIC 300 Station (Configuration) -- IND780_PBUS_FLT_SDV_V02]

Station Edit Insert PLC View Options Window Help

(0) UR

1	
2	CPU315-2 PN/DP(1)
X1	MPI/DP
X2	PN-IO
X2 P1	Port 1
X2 P2	Port 2
3	

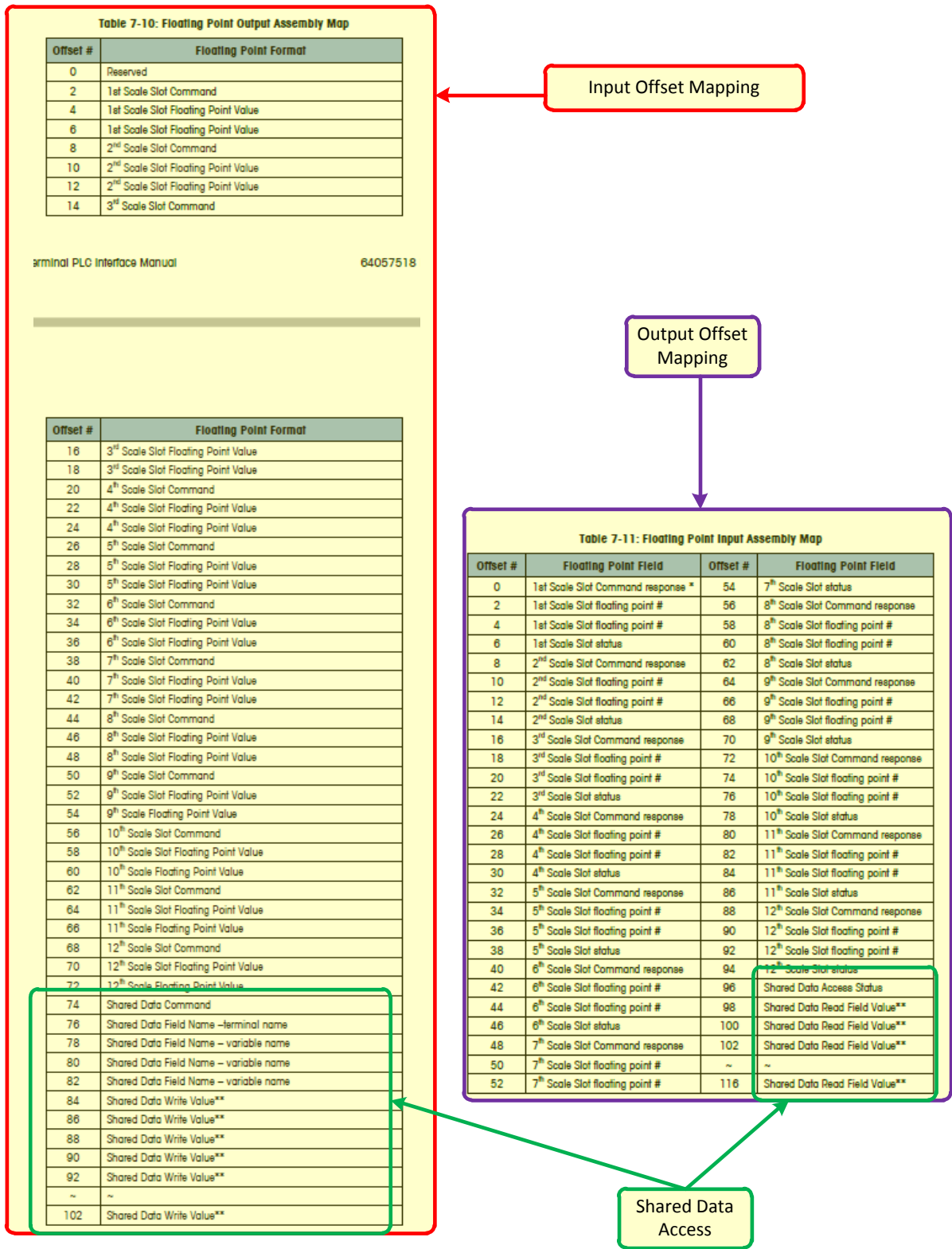
(3) IND780 DP-NORM

(3) IND780 Industrial Term

Slot	DP ID	...	Order Number / Designation	I Address	Q Address	Comment
1	16A0		I/O 104/118 Byt		256...287	
2	16A0		--> I/O 104/118 Byt		288...319	
3	16A0		--> I/O 104/118 Byt		320...351	
4	4A0		--> I/O 104/118 Byt		352...359	
5	16A1		--> I/O 104/118 Byt	256...287		
6	16A1		--> I/O 104/118 Byt	288...319		
7	16A1		--> I/O 104/118 Byt	320...351		
8	11A1		--> I/O 104/118 Byt	352...373		

I/O Address Mapping

Below are tables from the PLC manual that describe the offset mapping of the Inputs and Outputs as they are exchanged between the PLC and the IND780.

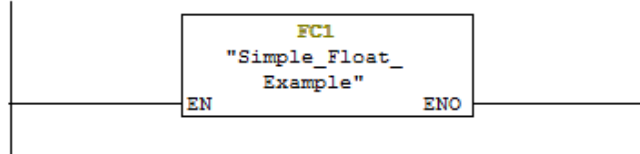


The Program Listing starts here with the OB1 routine.

OB1 : "Main Program Sweep (Cycle)"

Comment:

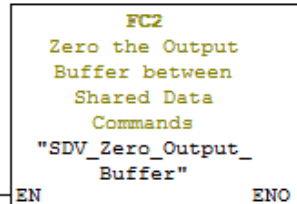
Network 1: Handle the Cyclic data



Network 2: Zero the Shared Data command buffer between commands.

DB2.DBX0.0

Trigger
the
SDV_PRG_Ac
cess
routine
to read a
ticket
"General_
Data".
Read_
Ticket

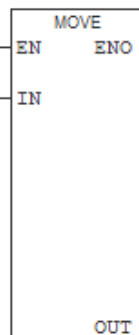


Network 3: Shared Data Access Program Example

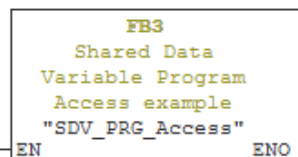
DB2.DBX0.0

Trigger
the
SDV_PRG_Ac
cess
routine
to read a
ticket
"General_
Data".
Read_
Ticket

DB2.DBX0.1
SDV_PRG_Ac
cess
routine
start One
Shot
"General_
Data".
Start_
Routine_
ONS
(p)



DB3



Shared Data via
PROGRAMMED
Access Example

This routine provides a sample for how the PLC can access Shared Data over the PROFIBUS. It does the following:

1. Monitor AS0101 for a value of 1. When seen, go to step 2.
2. Read AK0101 and place the string value into a string named Badge
3. Read AJ0101 and place the Floating Point value into Captured Weight.
4. Write the Cycle Count to AI0101.
5. Wait for AS0101 to go back to zero before resetting the sequence.

Note that this routine is for an IND780 configured in Floating Point mode. So, the mapping for Shared Data Access needs to match the Floating Point structure defined in the PLC manual (see 64057518_R08_IND780_PLC_EN.pdf).

The output offsets for Shared Data Access are as follows (see table 7-10):

Byte Offset	Description
74	Shared Data Command (0 = Idle, 1=Read, 2=Write)
76	Terminal Name (Default to 0)
78	Shared Data Variable name Characters 1&2
80	Shared Data Variable name Characters 3&4
82	Shared Data Variable name Characters 5&6
84-103	20 byte Write value, with byte zero starting at 84

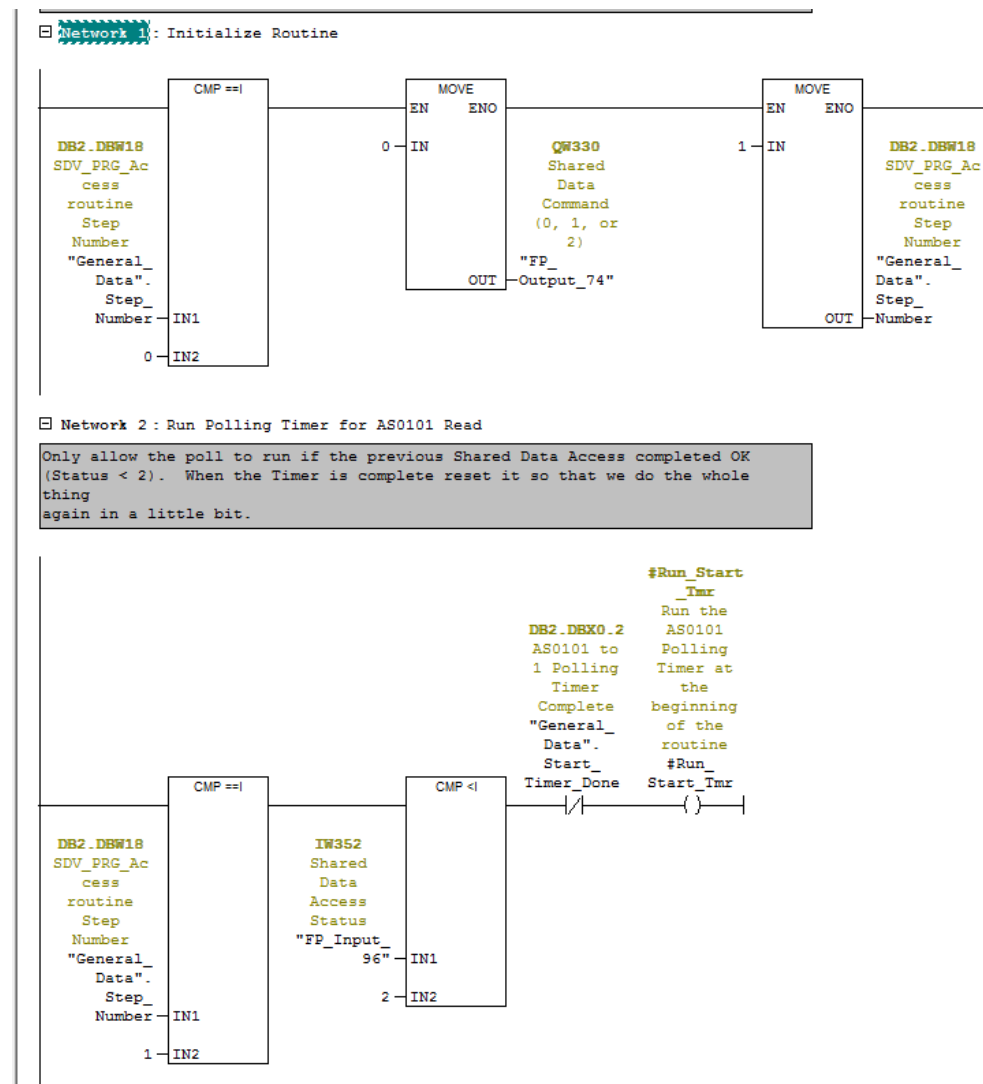
The input offsets for Shared Data Access are as follows (see table 7-11):

Byte Offset	Description
96	Shared Data Access Status (see state list below)
98-117	20 byte Read value, with byte zero starting at 98.

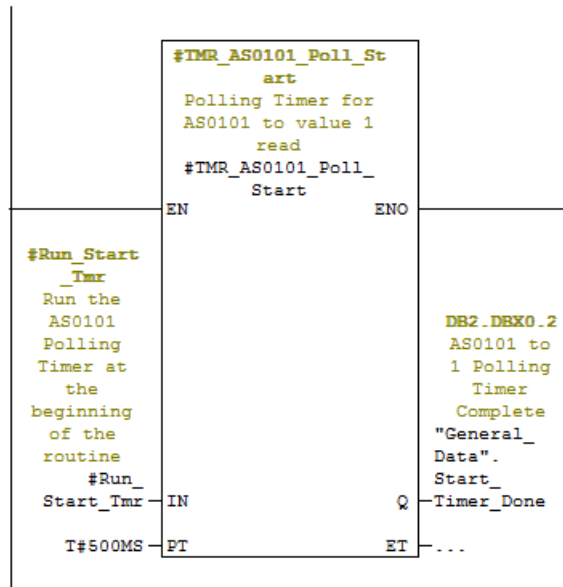
Shared Data Access Status States are as follows:

State	Description
0	Null Status
1	Command completed OK
2	Invalid Shared Data Name
3	Invalid Shared Data Command
4	Write Failed due to Write-protection (legal for trade)
5	Invalid Data Format
6	Cannot Access Remote Terminal (check terminal name value)
7	Invalid Service Request
8	Messaging not Enabled
9	Access Denied

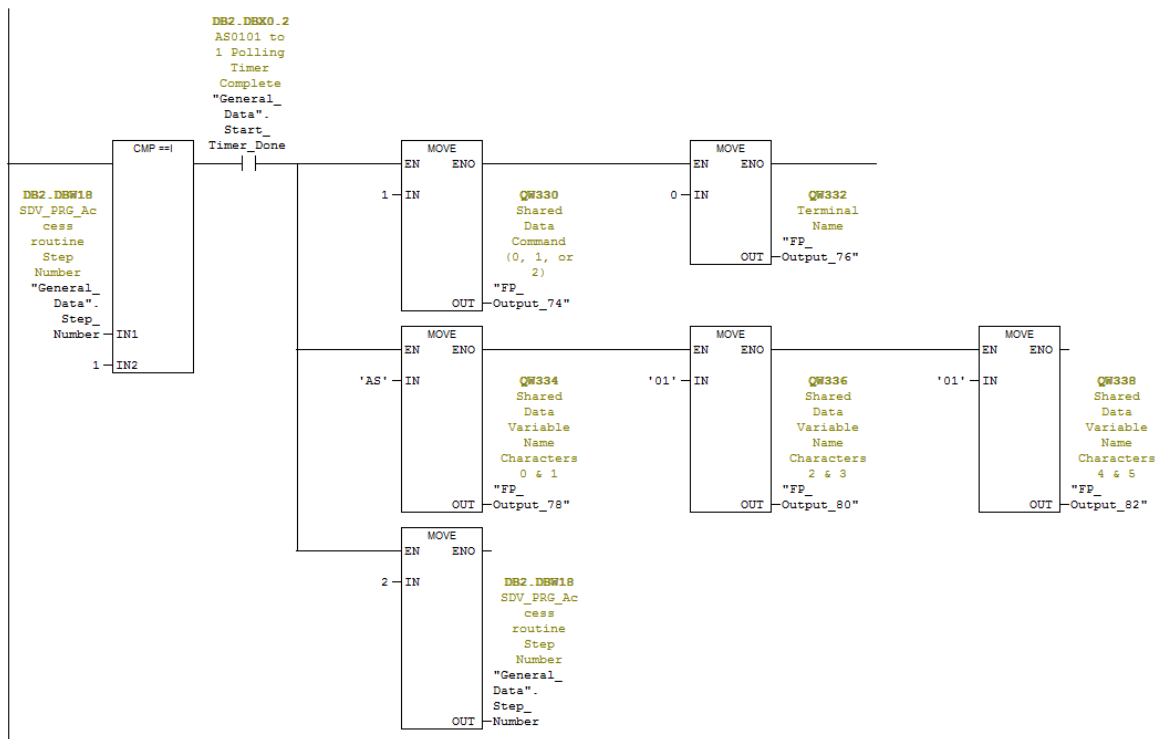
Routine FB3 is where the Programmed Shared Data Access sample is located. The listing from this point forward is from that routine.



Network 3 : Poll AS0101 for a value of 1 every 500 Milliseconds

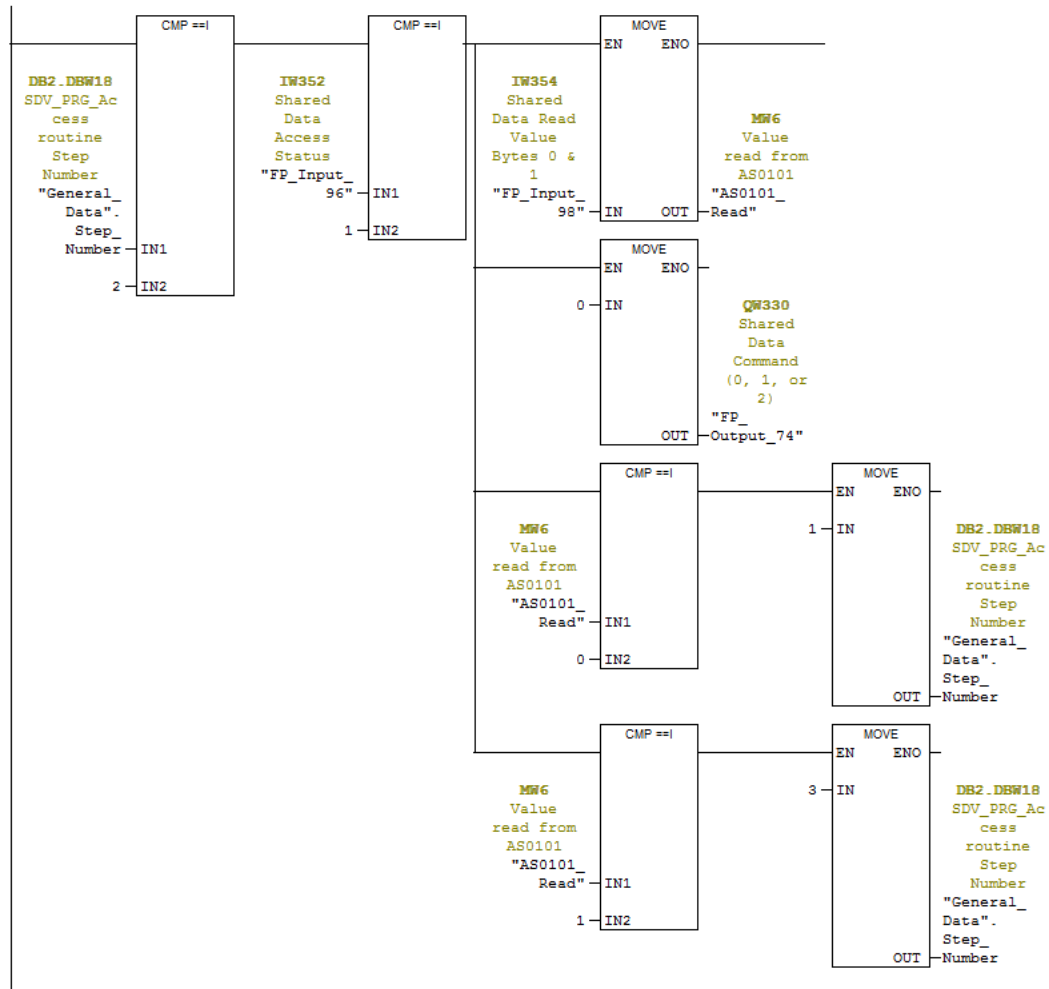


Network 4 : Set up Read of AS0101



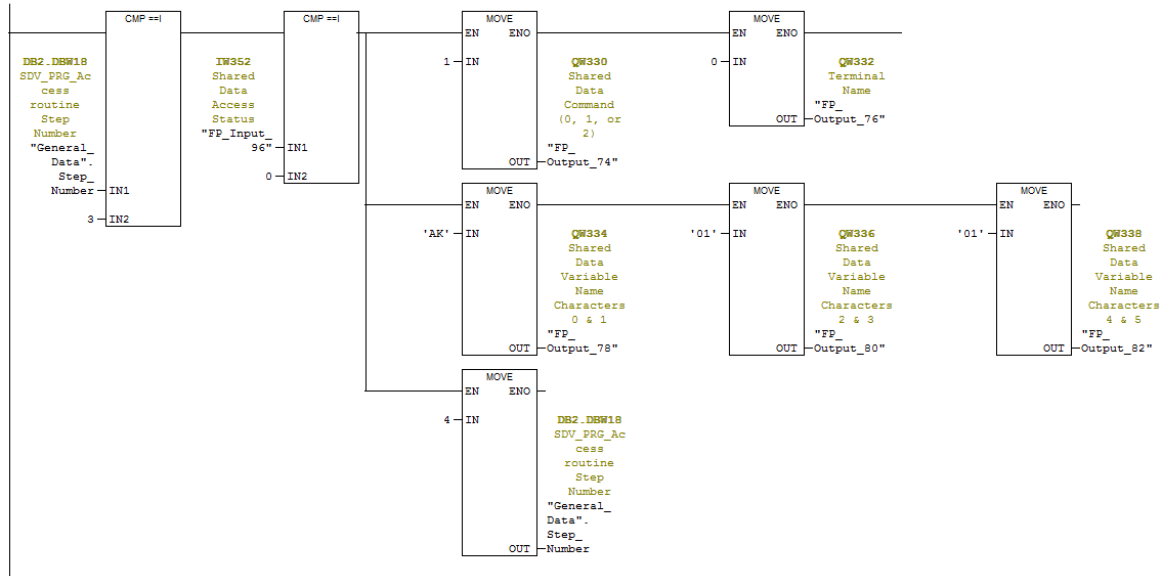
□ Network 5 : Wait for the response for AS0101

If the Command completed OK (Access Status = 1) then read the value and see if we're ready to read a ticket. If we are, then move on to step 3. Otherwise, go back to step one and do it all again. Don't forget to zero the command output first.



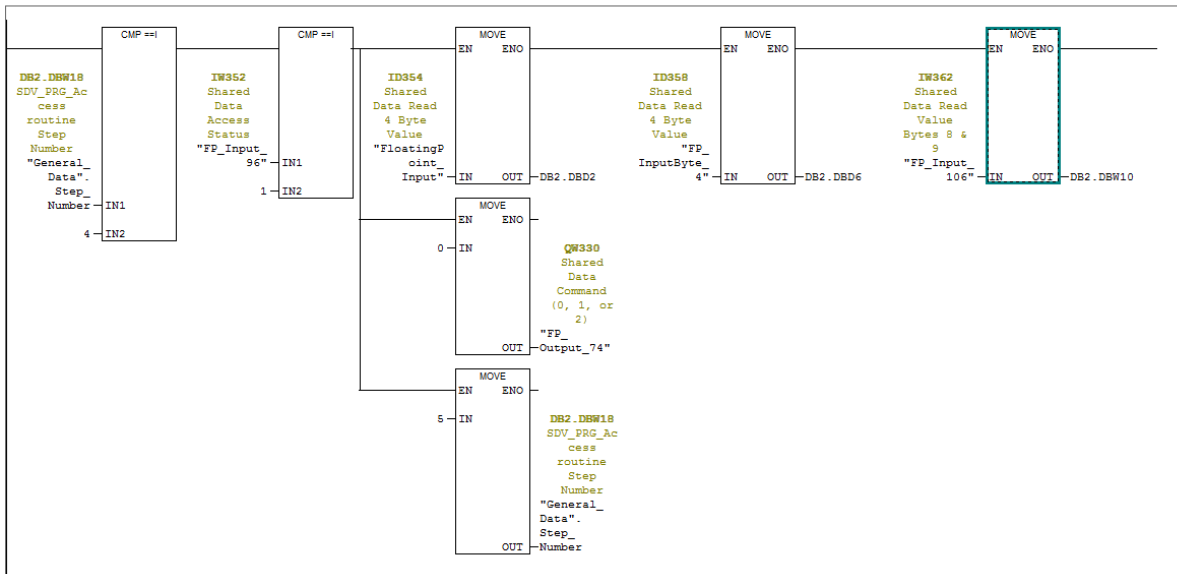
□ Network 6: Read AK0101 to get the Badge String

Wait for the Shared Data Access Status to go to zero (that means that the IND780 saw the Zeroed Command output from the previous step and is ready to receive a new command). Then set up the Read of AK0101 and move on to the next step.



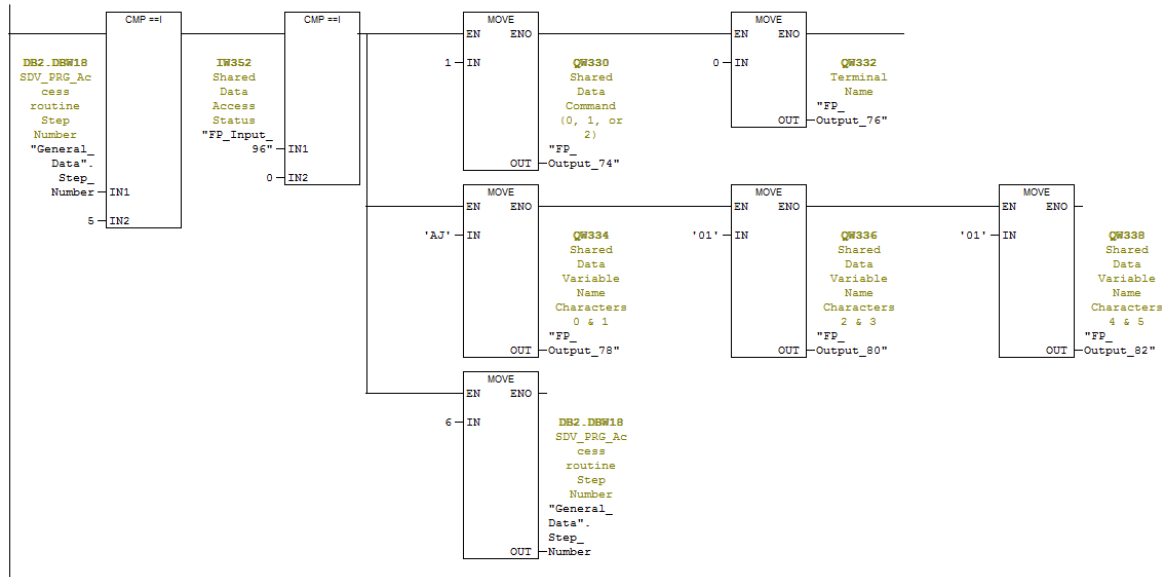
□ Network 7: Wait for the read of AK0101 to complete.

After the read is done, copy the string into the 'Badge' tag in DB2. The Zero the command and move on to the next step.



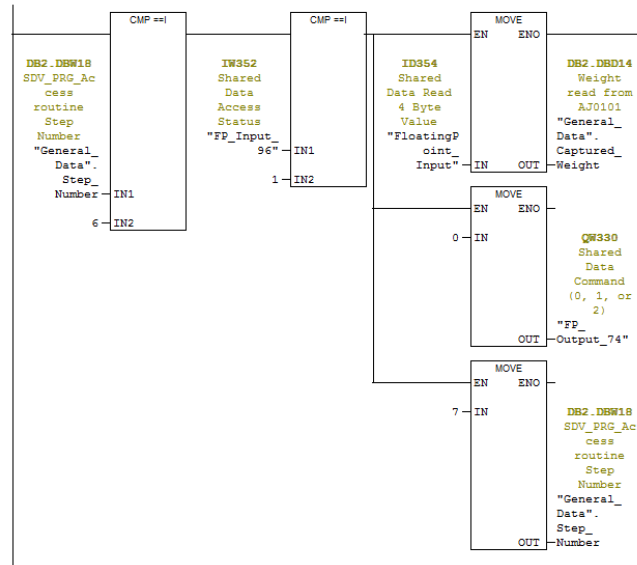
Network 8: Setup Read of AJ0101

Wait for the Shared Data Access Status to go to zero (that means that the IND780 saw the Zeroed Command output from the previous step and is ready to receive a new command). Then set up the Read of AJ0101 and move on to the next step.



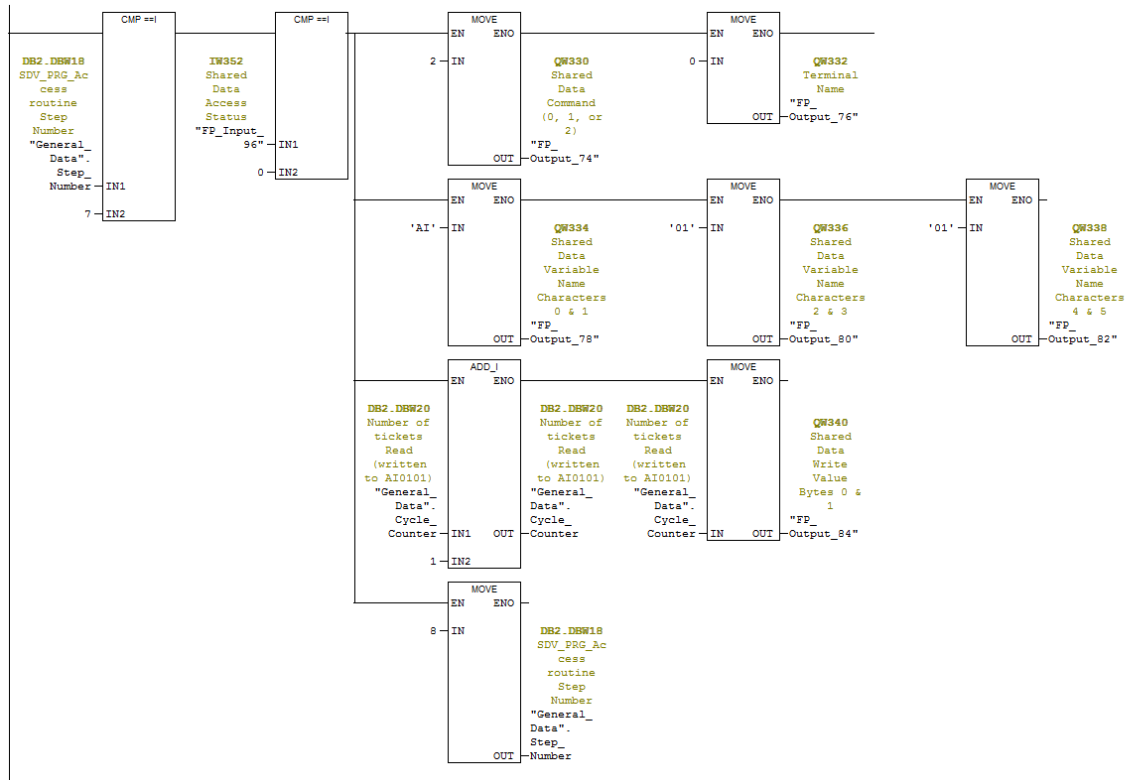
Network 9: Wait for the Read of AJ0101 to complete

When the read of AJ0101 is complete, get the Floating Point value and put it into the "Captured_Weight" value in DB2. Then, zero the Output command and move on to the next step.



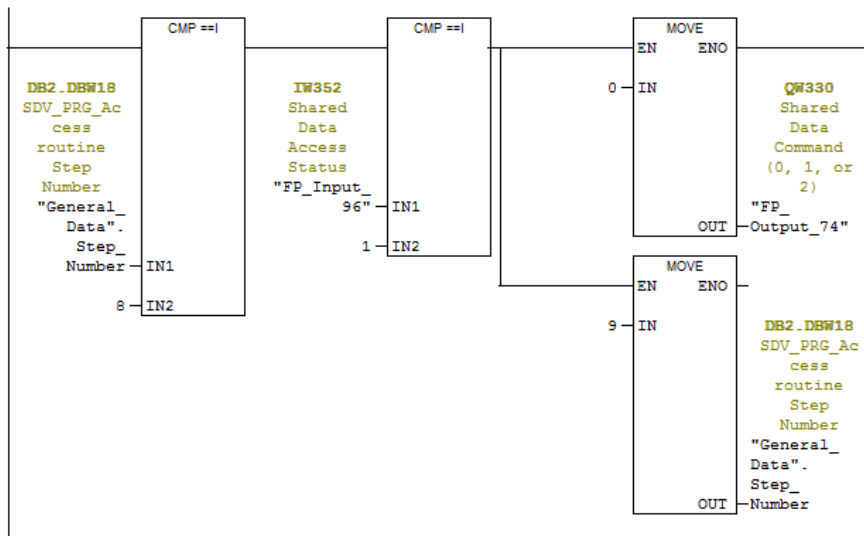
Network 10: Setup Write of AI0101

Wait for the Shared Data Access Status to go to zero (that means that the IND780 saw the Zeroed Command output from the previous step and is ready to receive a new command). Then set up the Write of AI0101 and move on to the next step. After the Shared Data name is setup, increment the cycle counter and copy it to the output (transmit) area of the I/O map.



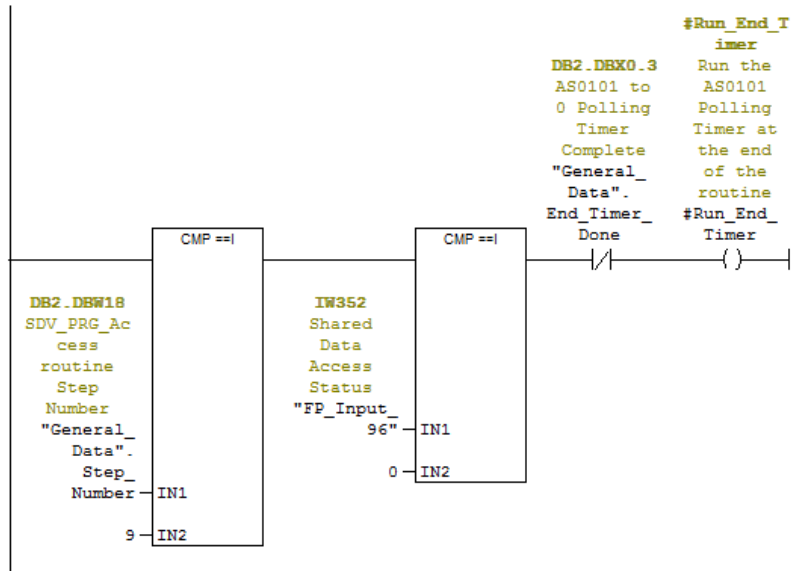
Network 11: Wait for the Write of AI0101, then wait for AS0101

When the Write of AI0101 is done, set up to Read AS0101 again and wait for it to go to zero so we can terminate the routine.

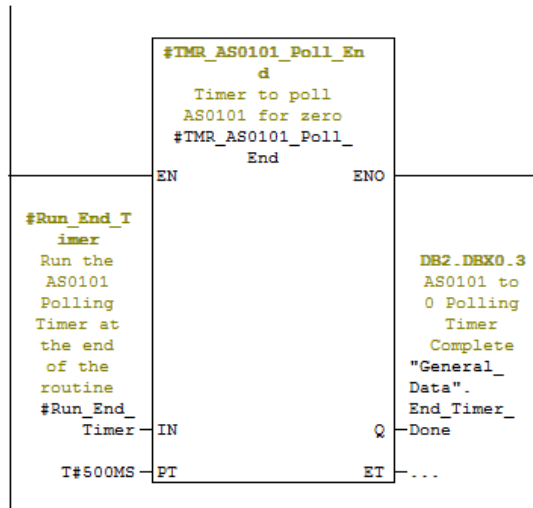


Network 12 : Run the End Timer

Wait for the IND780 to say that it saw our Zero Command, then run the Timer that allows us to poll AS0101 for going to Zero.

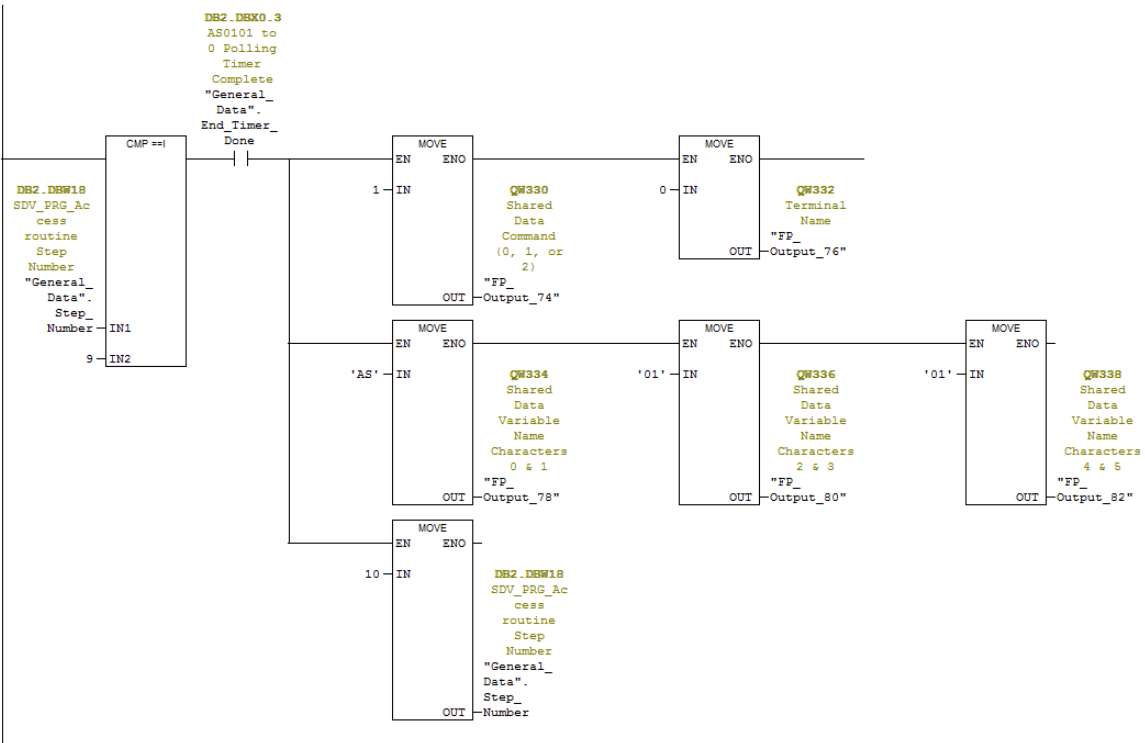


Network 13 : End Timer



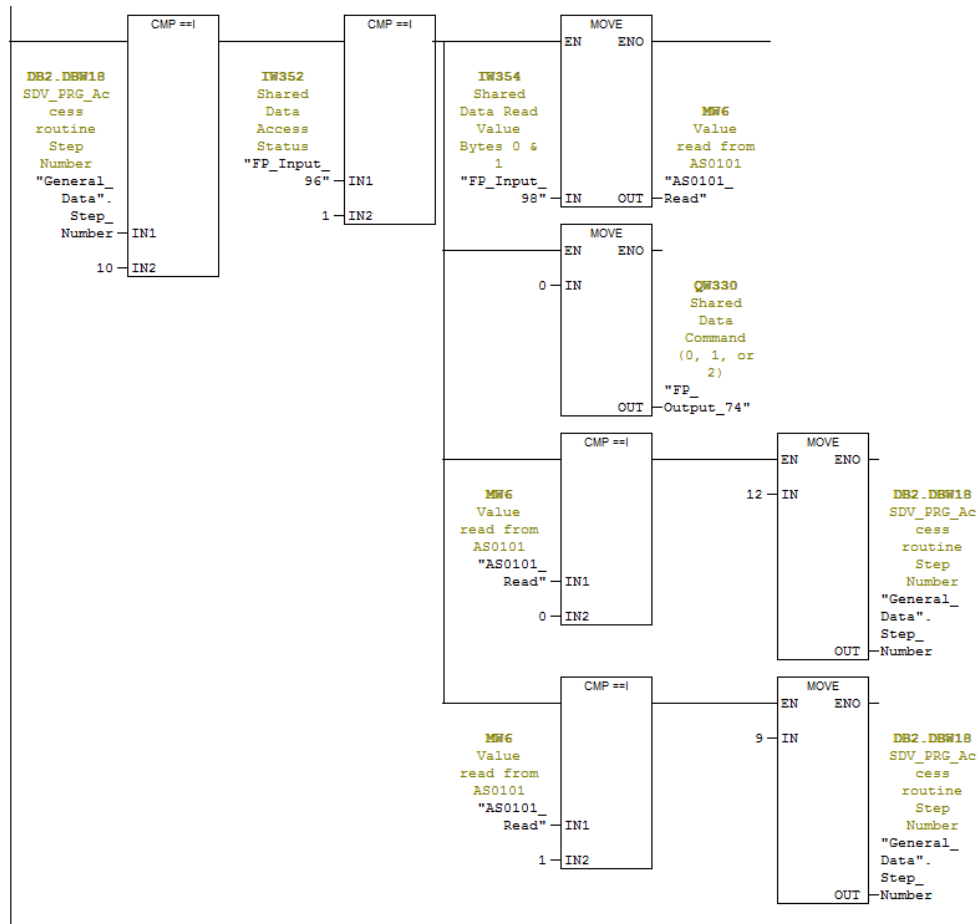
Network 14 : Set up Read of AS0101

When the End Timer has expired, set up the Read of AS0101.



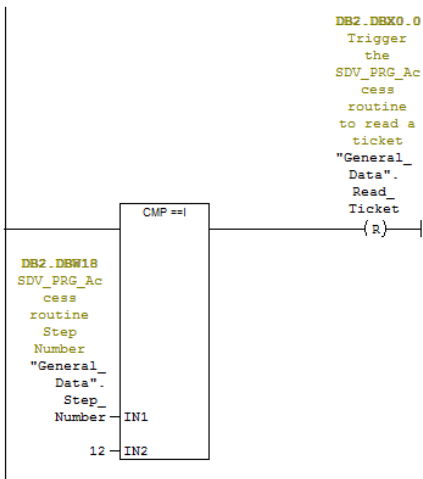
□ Network 15: Wait for the response for AS0101

If the Command completed OK (Access Status = 1) then read the value and see if it has gone back to zero. If it has then move on to step 12. Otherwise, go back to step 9 and do it all again. Don't forget to zero the command output first.



□ Network 16: We're Done!

If we made it here, then AS0101 is Zero. Terminate the routine.



The routine can be run from the SDV_Program_Access Variable Access Table (VAT) as shown below.

The top screenshot shows the SIMATIC Manager project tree for 'IND780_PBUS_FLT_SDV_V02'. The 'SDV_Program_Access' routine is highlighted in the 'Objects' list. A red box and arrow point to it, with a callout: 'Use this VAT to control the Programmed Shared Data Access Example'.

The bottom screenshot shows the 'SDV_Program_Access' routine's VAT table. A red box and arrow point to the 'false' value in the 'Status value' column for the first row, with a callout: 'Start the routine'. A green box and arrow point to the '12' value in the 'Status value' column for the second row, with a callout: 'Status of the routine'. A red box and arrow point to the '100.7' value in the 'Status value' column for the third row, with a callout: 'Values read back'.

Address	Symbol	Symbol comment	Display format	Status value	Modify value
1	// Trigger the next Ticket Read				
2	DB2.DBX 0.0	"General_Data".Read_Ticket	Trigger the SDV_PRG_Access routine to read...	BOOL	false
3					
4	// Show the SDV_Program_Access Routine Status				
5	DB2.DBW 18	"General_Data".Step_Number	SDV_PRG_Access routine Step Number	DEC	12
6	DB2.DBW 20	"General_Data".Cycle_Counter	Number of tickets Read (written to AI0101)	DEC	6
7	QW 330	"FP_Output_74"	Shared Data Command (0, 1, or 2)	DEC	0
8	IW 352	"FP_Input_96"	Shared Data Access Status	DEC	0
9					
10	// Show the Data that was Read				
11	DB2.DBW 14	"General_Data".Captured_Weight	Weight read from AI0101	FLOATING_POINT	100.7
12	DB2.DBW 2			CHARACTER	'222'
13	DB2.DBW 6			CHARACTER	'1111'
14	DB2.DBW 10			CHARACTER	W#16#0000
15					

The routine is started by setting the "General_Data".Read_Ticket bit to True (1). To see the routine execute properly, a Shared Data Server session should be opened with the IND780. Set values into the following Shared Data variables so that you can see them read back into the PLC:

AK0101 – write a string no longer than 10 bytes, "111-2222" for example.

AJ0101 – write a floating point value, 100.7 for example.

AS0101 – write a 1 to this value AFTER you have started the sample program. You will see the program steps cycle up and then stop. To terminate the current pass of the routine you will need to set AS0101 back to 0.

Set up a Callback on AI0101 to see the PLC update this value automatically.

Here is an example of the Shared Data session being used to run the process. Note that a telnet session can be started using the following command from a DOS prompt:

telnet 172.18.55.5 1701

The screenshot shows a Telnet session with the following text:

```
Telnet 172.18.55.5
53 Ready for user
>user admin
12 Access OK
>w AJ0101 100.7
00W001~OK
>w AK0101 222-1111
00W002~OK
>Callback AI0101
00W003~OK
>w AS0101 1
00W004~OK
>
00C005~a:0101=7 ~
>w AS0101 0
00W006~OK
>
```

Annotations (Callouts):

- The value 100.7 written to AJ0101 for the PLC to read.
- The string "222-1111" written to AK0101 for the PLC to read.
- Callback set up to show AI0101 when it updates
- The value 1 being written to AS0101 AFTER the Sample program has been started.
- Callback on AI0101 triggered to show the value 7.
- The value 0 being written to AS0101 to terminate the routine.