PROFINET

PROFINET is an application-level protocol used in industrial automation applications. This protocol uses standard Ethernet hardware and software to exchange I/O data, alarms, and diagnostics.

DataMan supports PROFINET I/O. This is one of the 2 “views” contained in the PROFINET communication standard. PROFINET I/O performs cyclic data transfers to exchange data with Programmable Logic Controllers (PLCs) over Ethernet. The second “view” in the standard, PROFINET CBA (Component Based Automation), is not supported.

A deliberate effort has been made to make the DataMan PROFINET communication model closely match the Cognex In-Sight family. Customers with In-Sight experience should find working with DataMan familiar and comfortable.

By default, the DataMan has the PROFINET protocol disabled. The protocol can be enabled via DMCC, scanning a parameter code or in the Setup Tool.

NOTE

If you have a wireless DataMan reader, read the section Industrial Protocols for the Wireless DataMan.

DMCC

The following commands can be used to enable/disable PROFINET. The commands can be issued via RS-232 or Telnet connection.

NOTE

Because you have to make changes to the Telnet client provided by Windows to communicate with DataMan, it is recommended you use third party clients such as PuTTY.

Enable:

```
||>SET PROFINET.ENABLED ON
||>CONFIG.SAVE
||>REBOOT
```

Disable:

```
||>SET PROFINET.ENABLED OFF
||>CONFIG.SAVE
||>REBOOT
```

Reader Configuration Code

Scanning the following reader configuration codes will enable/disable PROFINET for your corded reader.

NOTE

You must reboot the device for the change to take effect.

Enable: Disable:

Scanning the following reader configuration codes will enable/disable PROFINET for your DataMan 8000 base station.

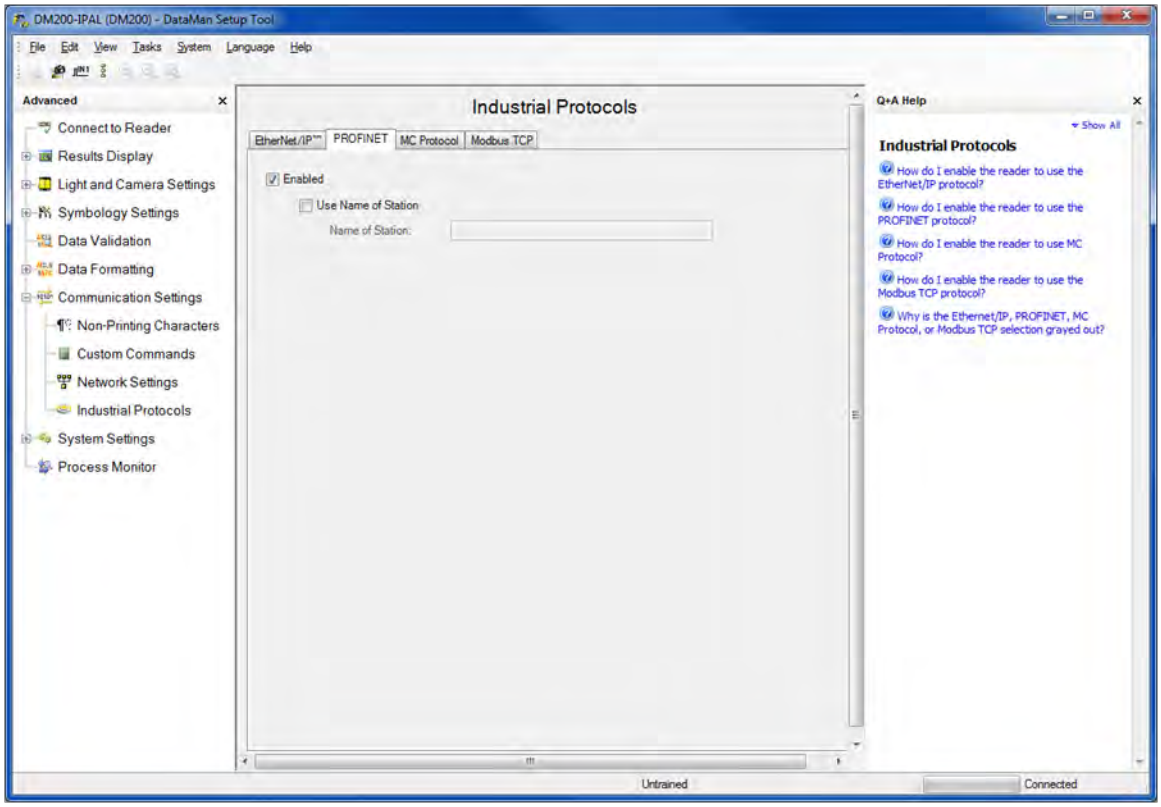
NOTE

You must reboot the device for the change to take effect.

Enable: Disable:

Setup Tool

The PROFINET protocol can be enabled by checking **Enabled** on the Industrial Protocols pane's PROFINET tab.



Make sure to save the new selection by choosing “Save Settings” before disconnecting from the reader.

NOTE

You must reboot your reader for the new settings to take effect.

Getting Started

Preparing to use PROFINET involves the following main steps:

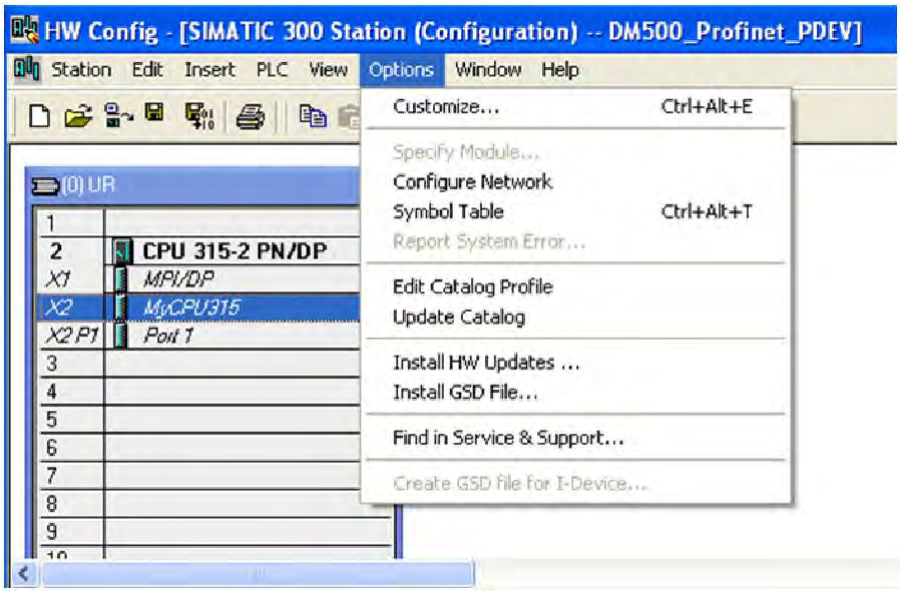
- Make sure you have the Siemens Step 7 programming software (SIMATIC) installed.
- Set up the Siemens Software tool so that it recognizes your DataMan device.

  Install the Generic Station Description (GSD) file.

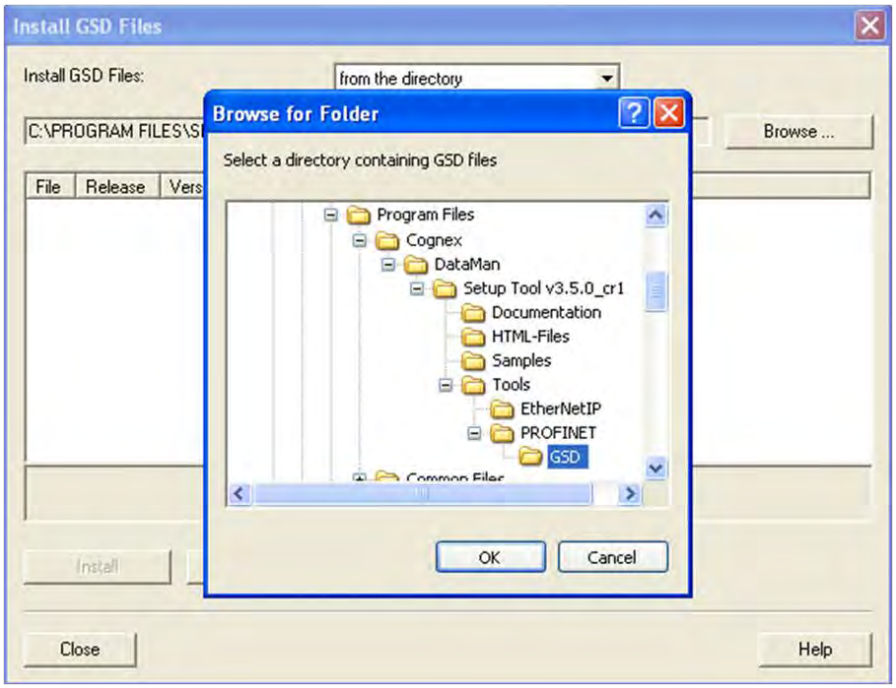
Perform the following steps to set up PROFINET:

1. Verify that SIMATIC is on your machine.
2. From the Windows Start menu, launch the SIMATIC Manager.
3. If you already have a project, select “Cancel” to skip past the New Project wizard. Otherwise, let the wizard guide you through creating a new project.

4. Once the Manager has opened the project, double-click on the “Hardware” icon to open the “HW Config” dialog screen. From the main menu, select “Options→Install GSD File…”.



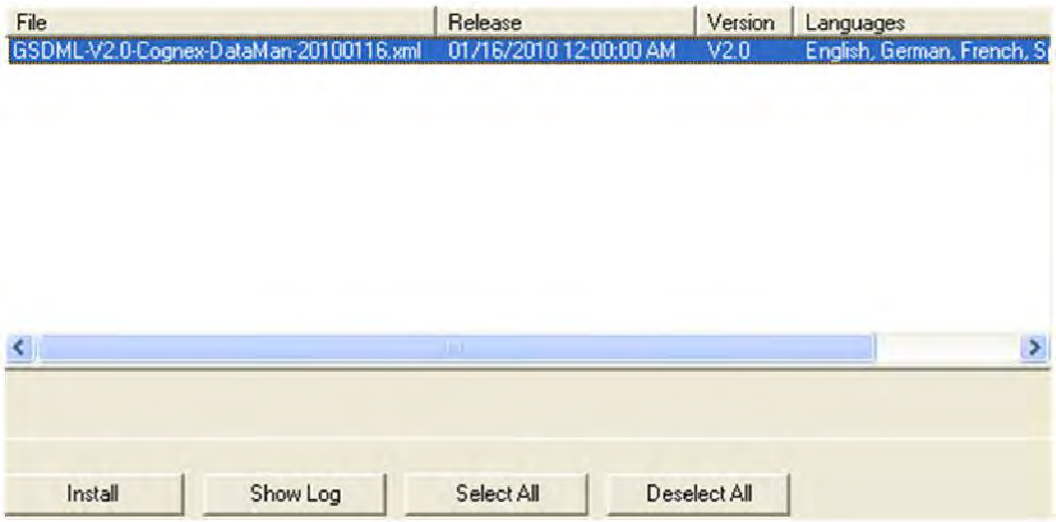
5. Browse to the location where the GSD file was installed (or the location where you saved the GSD file if it was downloaded from the web).



6. Select the GSD file you wish to install and follow the displayed instructions to complete the installation.

   NOTE

   There may be more than one GSD file in the list. If you are unsure which to install, choose the one with the most recent date.

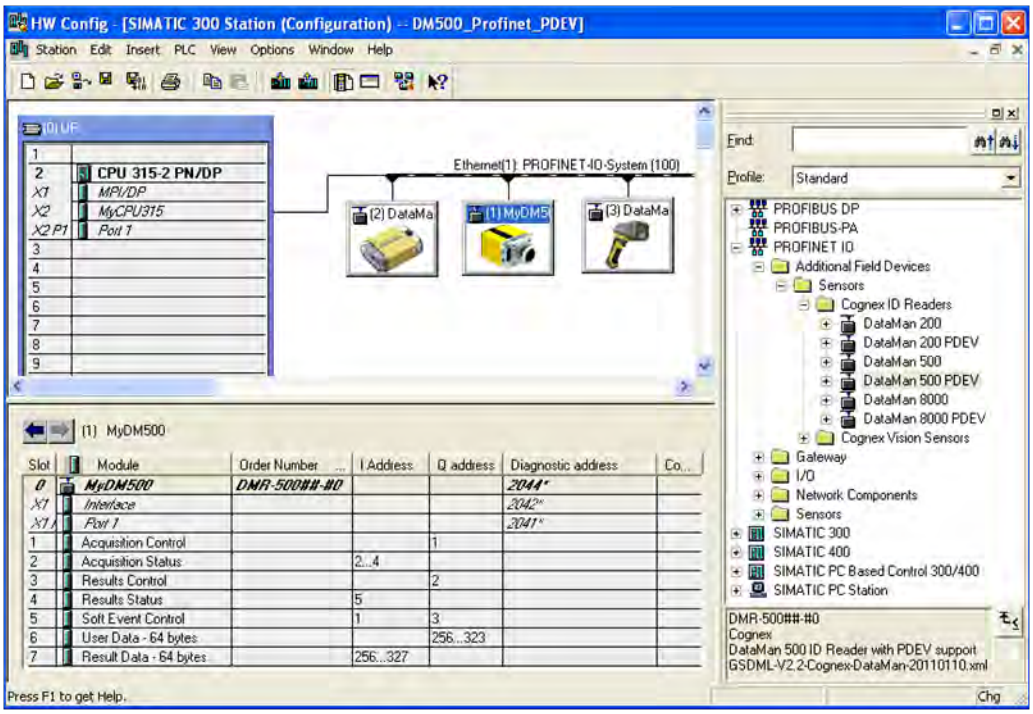


7. Add your DataMan device to your project. This makes the DataMan available in the Hardware Catalog. Launch the SIMATIC Hardware Config tool.

8. In the main menu, select View → Catalog.

9. The catalog is displayed. Expand the “PROFINET IO” tree to the “Cognex ID Readers” node.

10. With the left mouse button, drag the DataMan reader over and drop it on the PROFINET IO network symbol in the left pane.

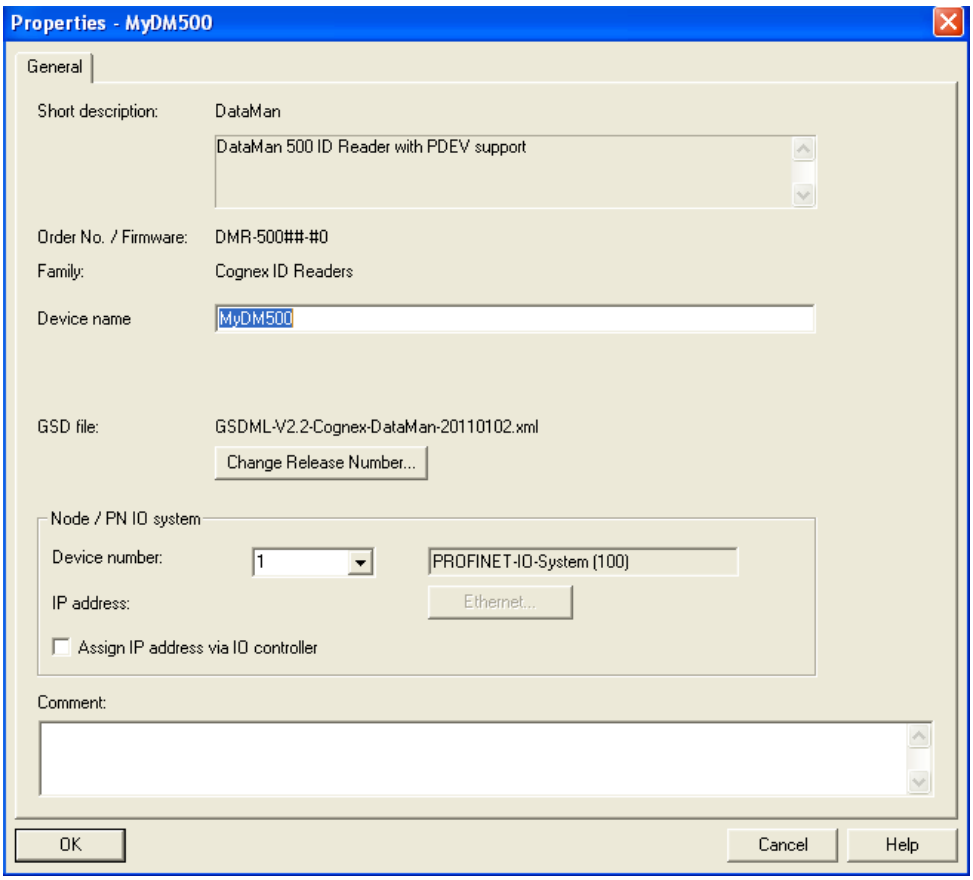


The HW Config tool automatically maps the DataMan I/O modules into the memory space.

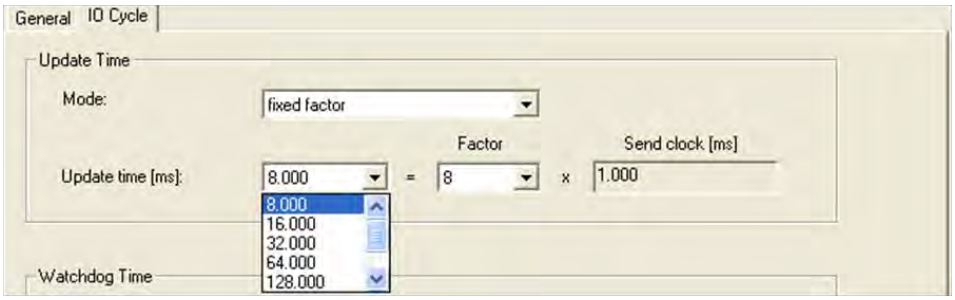
NOTE

By default, the 64 byte User Data and 64 byte Result Data Modules are inserted. There are multiple sizes available for both of these modules. To optimize performance use the module size that most closely matches the actual data requirements of your application. You can change the module simply by deleting the one in the table and inserting the appropriate sized module from the catalog.

11. Right-click on the DataMan icon and select “Object Properties…”.
12. Give the reader a name. This must match the name of your actual DataMan reader. The name must be unique and conform to DNS naming conventions. Refer to the SIMATIC Software help for details.
13. If your DataMan reader is configured to use its own static IP, uncheck the “Assign IP address via IO controller” box. Otherwise if you wish the PLC to assign an IP address, select the Ethernet button and configure the appropriate address.

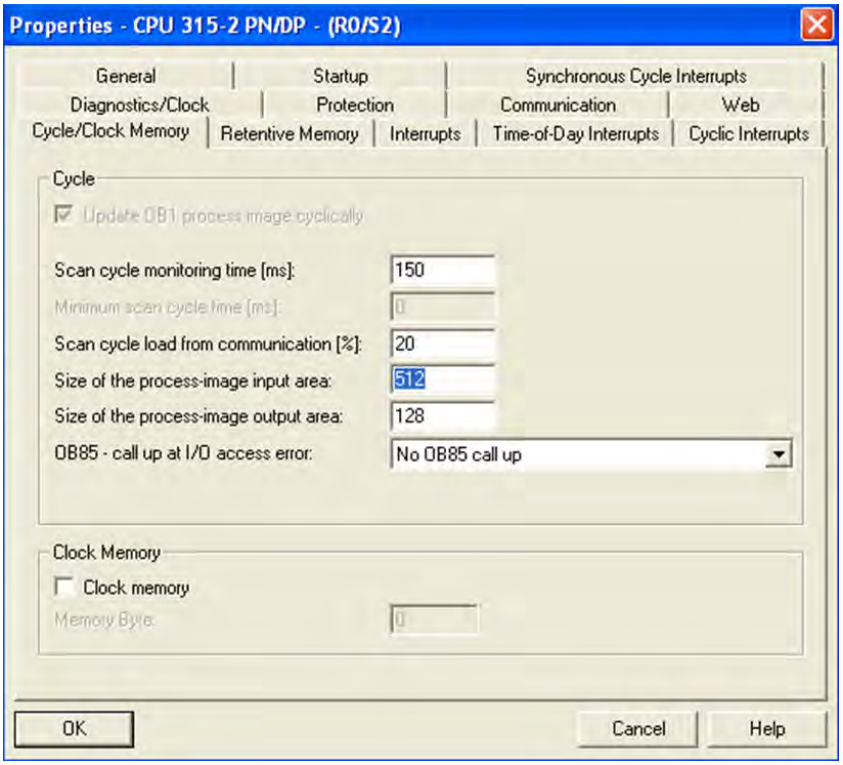


14. In the “IO Cycle” tab, select the appropriate cyclic update rate for your application.



15. By default, the SIMATIC software maps the User Data & Result Data Modules to offset 256. This is outside of the default process image area size of 128. That is, by default, data in these modules are inaccessible by some SFCs such as BLKMOV. As a solution, either remap the modules to lower offsets within the process image area or expand the process image area to include these modules.

If you choose to expand the process image area, make the size large enough for the module size plus the default 256 offset.



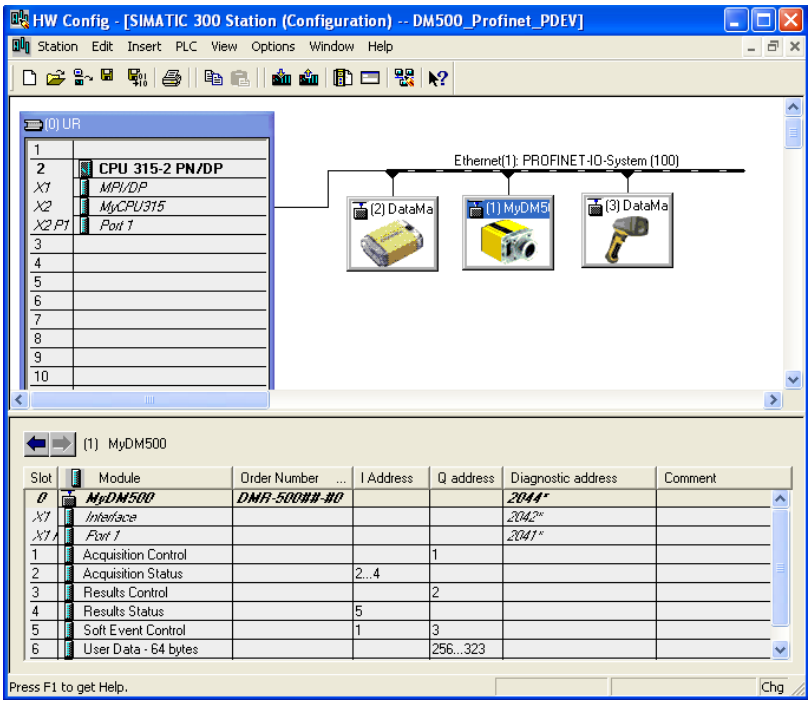
NOTE

Expanding the process image can have a performance impact on the PLC scan cycle time. If your scan time is critical, use the minimal acceptable module sizes and manually remap them down lower in the process image.

Modules

The PROFINET implementation on DataMan consists of seven I/O modules.

1. Acquisition Control Module
2. Acquisition Status Module
3. Results Control Module
4. Results Status Module
5. Soft Event Control Module
6. User Data Module
7. Result Data Module



Acquisition Control Module

Controls image acquisition. This module consists of data sent from the PLC to the DataMan device.

Slot number: 1

Total Module size: 1 byte

Bit	Name	Description
0	Trigger Enable	Setting this bit enables triggering via PROFINET. Clearing this bit disables triggering.
1	Trigger	Setting this bit triggers an acquisition when the following conditions are met: • Trigger Enable is set • No acquisition is currently in progress • The device is ready to trigger
2 – 7	Reserved	Reserved for future use

Acquisition Status Module

Indicates the current acquisition status. This module consists of data sent from the DataMan device to the PLC.

Slot number: 2

Total Module size: 3 bytes

Bit	Name	Description
0	Trigger Ready	Indicates when the device is ready to accept a new trigger. Bit is True when "Trigger Enable" has been set and the device is ready to accept a new trigger.
1	Trigger Acq	Indicates that the DataMan has received a new Trigger. This bit will remain True as long as the "Trigger" bit remains True (that is, it is interlocked with the Trigger bit).
2	Acquiring	Indicates that the DataMan is currently acquiring an image.
3	Missed Acq	Indicates that the DataMan was unable to successfully trigger an acquisition. Bit is cleared when the next successful acquisition occurs.
4 – 7	Reserved	Reserved for future use
8–23	Trigger ID	ID value of the next trigger to be issued (16-bit integer). Used to match issued triggers with corresponding result data received later. This same value will be returned in ResultID of the result data.

NOTE

The Missed Acq bit in the Acquisition Status Register will only be set if an acquisition triggered from the Acquisition Control Module could not get executed.

Results Control Module

Controls the processing of result data. This module consists of data is sent from the PLC to the DataMan device.

Slot number: 3

Total Module size: 1 byte

Bit	Name	Description
0	Results Buffer Enable	Enables queuing of "Result Data". If enabled, the current result data will remain until acknowledged (even if new results arrive). New results are queued. The next set of results are pulled from the queue (made available in the Result Data module) each time the current results are acknowledged. The DataMan will respond to the acknowledge by clearing the "Results Available" bit. Once the "Results Ack" bit is cleared the next set of read results will be posted and "Results Available" will be set True. If results buffering is not enabled newly received read results will simply overwrite the content of the Result Data module.
1	Results Ack	Bit is used to acknowledge that the PLC has successfully read the latest result data. When set True the "Result Available" bit will be cleared. If result buffering is enabled, the next set of result data will be pulled from the queue and "Result Available" will again be set True.
2 – 7	Reserved	Reserved for future use

Results Status Module

Indicates the acquisition and result status. This module consists of data sent from the DataMan device to the PLC.

Slot number: 4

Total Module size: 1 byte

Bit	Name	Description
0	Decoding	Indicates that the DataMan is decoding an acquired image.
1	Decode Complete	Bit is toggled on the completion of a decode operation when the new results are made available (0→1 or 1→0).
2	Result Buffer Overrun	Indicates that the DataMan has discarded a set of read results because the results queue is full. Cleared when the next set of results are successfully queued.
3	Results Available	Indicates that a new set of read results are available (i.e. the contents of the Result Data module are valid). Cleared when the results are acknowledged.
4 – 6	Reserved	Reserved for future use

Bit	Name	Description
7	General Fault	Indicates that a fault has occurred (i.e. Soft Event "Set Match String" or "Execute DMCC" error has occurred).

Soft Event Control Module

Used to initiate a Soft Event and receive acknowledgment of completion. Note, this is a bi-directional I/O module. Module data sent from the PLC initiates the Soft Event. Module data sent by the DataMan device acknowledges completion.

Slot number: 5

Total Module size: 1 byte (input) and 1 byte (output)

Data written from the PLC to DataMan:

Bit	Name	Description
0	Train Code	Bit transition from 0→1 will cause the train code operation to be invoked.
1	Train Match String	Bit transition from 0→1 will cause the train match string operation to be invoked.
2	Train Focus	Bit transition from 0→1 will cause the train focus operation to be invoked.
3	Train Brightness	Bit transition from 0→1 will cause the train brightness operation to be invoked.
4	Untrain	Bit transition from 0→1 will cause the untrain operation to be invoked.
5	Reserved	Reserved for future use
6	Execute DMCC	Bit transition from 0→1 will cause the DMCC operation to be invoked. Note that a valid DMCC command string must first be placed in "User Data" before invoking this event.
7	Set Match String	Bit transition from 0→1 will cause the set match string operation to be invoked. Note that match string data must first be placed in "User Data" before invoking this event.

Data written from the DataMan to PLC:

Bit	Name	Description
0	Train Code Ack	Indicates that the "Train Code" operation has completed

Bit	Name	Description
1	Train Match String Ack	Indicates that the “Train Match String” operation has completed
2	Train Focus Ack	Indicates that the “Train Focus” operation has completed
3	Train Brightness Ack	Indicates that the “Train Brightness” operation has completed
4	Untrain Ack	Indicates that the “Untrain” operation has completed
5	Reserved	Reserved for future use
6	Execute DMCC Ack	Indicates that the “Execute DMCC” operation has completed
7	Set Match String Ack	Indicates that the “Set Match String” operation has completed

User Data Module

Data sent from a PLC to a DataMan to support acquisition, decode and other special operations. Currently this module is only used to support the “Execute DMCC” and “Set Match String” soft events.

NOTE

There are actually 5 versions of the User Data module. Only one instance can be configured for use in a given application. The “User Data Option” and “User Data Length” fields are the same for each module. The “User Data” field varies in size based on the selected module. Choose the module which is large enough to exchange the amount of data required by your application.

Slot number: 6

Total Module size:
- 4 + 16 (16 bytes of User Data)
- 4 + 32 (32 bytes of User Data)
- 4 + 64 (64 bytes of User Data)
- 4 + 128 (128 bytes of User Data)
- 4 + 250 (250 bytes of User Data)

Byte	Name	Description
0 - 1	User Data Option	Currently only used by “Set Match String” soft event. Specifies which code target to assign the string (16-bit

Byte	Name	Description
		Integer). 0, assign string to all targets 1, assign string to 2D codes 2, assign string to QR codes 3, assign string to 1D / stacked / postal codes
2 - 3	User Data Length	Number of bytes of valid data actually contained in the “User Data” field (16-bit Integer).
4 …	User Data	Data sent from the PLC to the DataMan to support acquisition, decode and other special operations (array of bytes).

Result Data Module

Read result data sent from a DataMan to a PLC.

Look into the Result Data Module for non-PROFINET related data if there was an acquisition problem on the reader. The "Result Code" part contains information about trigger overruns or buffer overflows that have occurred on the reader.

NOTE

There are actually 5 versions of the Result Data module. Only a single instance can be configured for use in a given application. The “Result ID”, “Result Code”, “Result Extended” and “Result Length” fields are the same for each module. The “Result Data” field varies in size based on the selected module. Choose the module which is large enough to exchange the amount of result data required by your application.

Slot number: 7

Total Module size: 8 + 16 (16 bytes of Result Data)
8 + 32 (32 bytes of Result Data)
8 + 64 (64 bytes of Result Data)
8 + 128 (128 bytes of Result Data)
8 + 246 (246 bytes of Result Data)

Byte	Name	Description
0 - 1	Result ID	The value of the “Trigger ID” when the trigger that generated these results was issued. Used to match up

Byte	Name	Description
		triggers with corresponding result data (16-bit Integer).
2 - 3	Result Code	Indicates the success or failure of the read that produced these results (16-bit Integer). Bit 0,1=read, 0=no read Bit 1,1=validated, 0=not validated (or validation not in use) Bit 2,1=verified, 0=not verified (or verification not in use) Bit 3,1=acquisition trigger overrun Bit 4,1=acquisition buffer overflow Bits 5-15 reserved
4 - 5	Result Extended	Currently unused (16-bit Integer).
6 - 7	Result Length	Actual number of bytes of read data contained in the "Result Data" field (16-bit Integer).
8 …	Result Data	Decoded read result data (array of bytes)

Operation

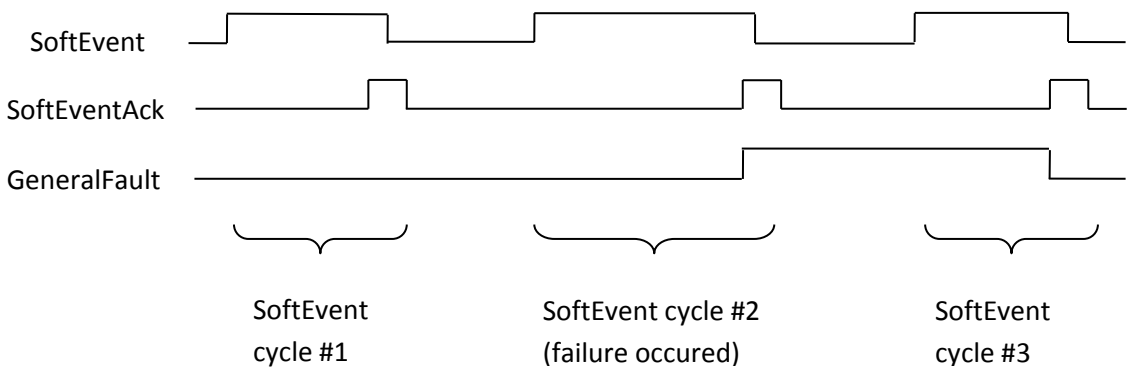
SoftEvents

SoftEvents act as "virtual" inputs. When the value of a SoftEvent changes from 0 → 1 the action associated with the event will be executed. When the action completes the corresponding SoftEventAck bit will change from 0 → 1 to signal completion. The acknowledge bit will change back to 0 when the corresponding SoftEvent bit is set back to 0.

The "ExecuteDMCC" and "SetMatchString" soft event actions require user supplied data. This data must be written to the UserData & UserDataLength area of the UserData Module prior to invoking the soft event. Since both of these soft events depend on the UserData, only one may be invoked at a time.

General Fault Indicator

When a communication related fault occurs the "GeneralFault" bit will change from 0 → 1. Currently the only fault conditions supported are soft event operations. If a soft event operation fails, the fault bit will be set. The fault bit will remain set until the next soft event operation or until triggering is disabled and again re-enabled.



Acquisition Sequence

DataMan can be triggered to acquire images by several methods. It can be done explicitly by manipulating the Trigger bit of the Acquisition Control Module, it can be triggered by external hard wired input, and finally it can be triggered via DMCC command. Manipulating the Acquisition Control Module bits will be discussed here.

On startup the "Trigger Enable" bit will be False. It must be set to True to enable triggering. When the device is ready to accept triggers, the "Trigger Ready" bit will be set to True.

While the Trigger Ready bit is True, each time the reader sees the "Trigger" bit change from 0 to 1, it will initiate an image acquisition. The client (PLC) should hold the bit in the new state until that same state value is seen back in the Trigger Ack bit (this is a necessary handshake to guarantee that the change is seen by the reader).

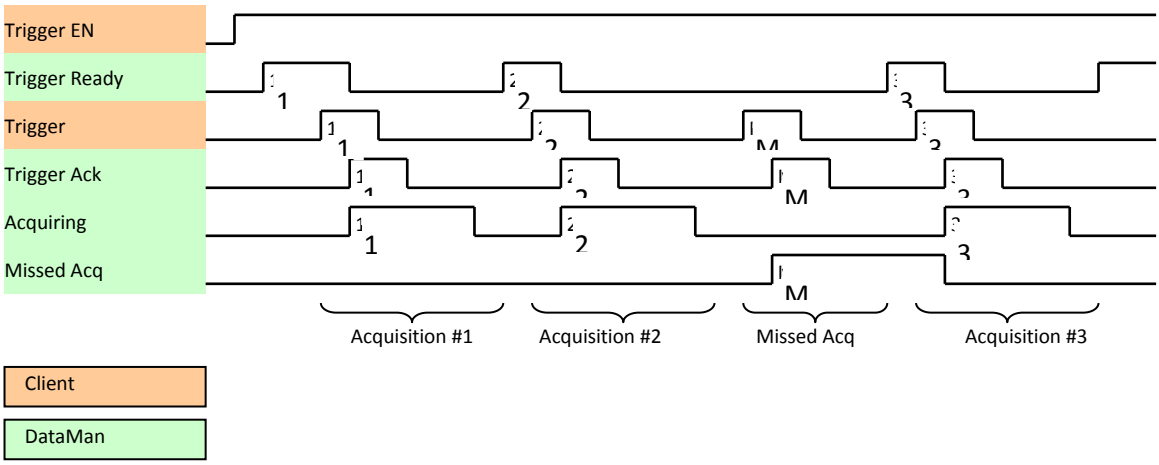
During an acquisition, the Trigger Ready bit will be cleared and the Acquiring bit will be set to True. When the acquisition is completed, the Acquiring bit will be cleared. The Trigger Ready bit will again be set True once the device is ready to begin a new image acquisition.

If results buffering is enabled, the device will allow overlapped acquisition and decoding operations. Trigger Ready will be set high after acquisition is complete but while decoding is still in process. This can be used to achieve faster overall trigger rates. If result buffering is not enabled, the Trigger Ready bit will remain low until both the acquisition and decode operations have completed.

To force a reset of the trigger mechanism set the Trigger Enable bit to False, until the Trigger Ready bit is 0. Then, Trigger Enable can be set to True to re-enable acquisition.

As a special case, an acquisition can be cancelled by clearing the Trigger signal before the read operation has completed. This allows for the cancellation of reads in Presentation and Manual mode if no code is in the field of view. To ensure that a read is not unintentionally cancelled, it is

advised that the PLC hold the Trigger signal True until both TriggerAck and ResultsAvailable are True (or DecodeComplete toggles state).



Decode / Result Sequence

After an image is acquired it is decoded. While being decoded, the "Decoding" bit of the Result Status Module is set. When decode is complete, the Decoding bit is cleared and the "Decode Complete" bit is toggled.

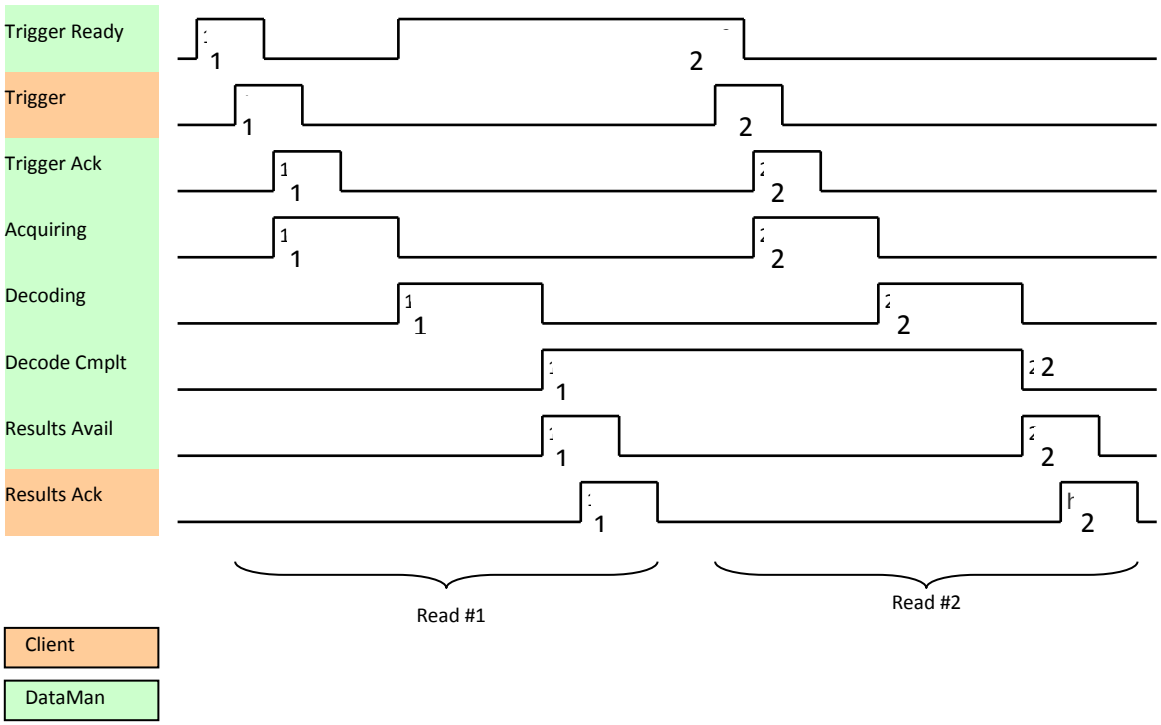
The "Results Buffer Enable" bit determines how decode results are handled by the reader. If the Results Buffer Enable bit is set to False, then the decode results are immediately placed into the Results Module and Results Available is set to True.

If the Results Buffer Enable bit is set to True the new results are queued. The earlier decode results remain in the Results Module until they are acknowledged by the client by setting the "Results Ack" bit to True. After the Results Available bit is cleared, the client should set the Results Ack bit back to False to allow the next queued results to be placed in to the Results Module. This is a necessary handshake to ensure the results are received by the DataMan client (PLC).

Behavior of DecodeStatusRegister

Bit	Bit Name	Results if Buffering Disabled	Results if Buffering Enabled
1	Decoding	Set when decoding an image.	Set when decoding an image.
2	Decode Complete	Toggled on completion of an image decode.	Toggled on completion of an image decode.

Bit	Bit Name	Results if Buffering Disabled	Results if Buffering Enabled
3	Results Buffer Overflow	Remains set to zero.	Set when decode results could not be queued because the client failed to acknowledge a previous result. Cleared when the decode result is successfully queued.
4	Results Available	Set when new results are placed in the Results Module. Stays set until the results are acknowledged by setting Results Ack to true.	Set when new results are placed in the Results Module. Stays set until the results are acknowledged by setting Results Ack to true.



Results Buffering

There is an option to enable a queue for decode results. If enabled this allows a finite number of decode result data to queue up until the client (PLC) has time to read them. This is useful to smooth out data flow if the client (PLC) slows down for short periods of time or if there are surges of read activity.

Also, if result buffering is enabled the device will allow overlapped acquisition and decode operations. Depending on the application this can be used to achieve faster over all trigger rates. See Acquisition Sequence description above for further detail.

In general, if reads are occurring faster than results can be sent out, the primary difference between buffering or not buffering is determining which results get discarded. If buffering is not enabled the most recent results are kept and the earlier result (which was not read by the PLC fast enough) is lost. Essentially the more recent result will simply over write the earlier result. If buffering is enabled (and the queue becomes full) the most recent results are discarded until room becomes available in the results queue.

NOTE

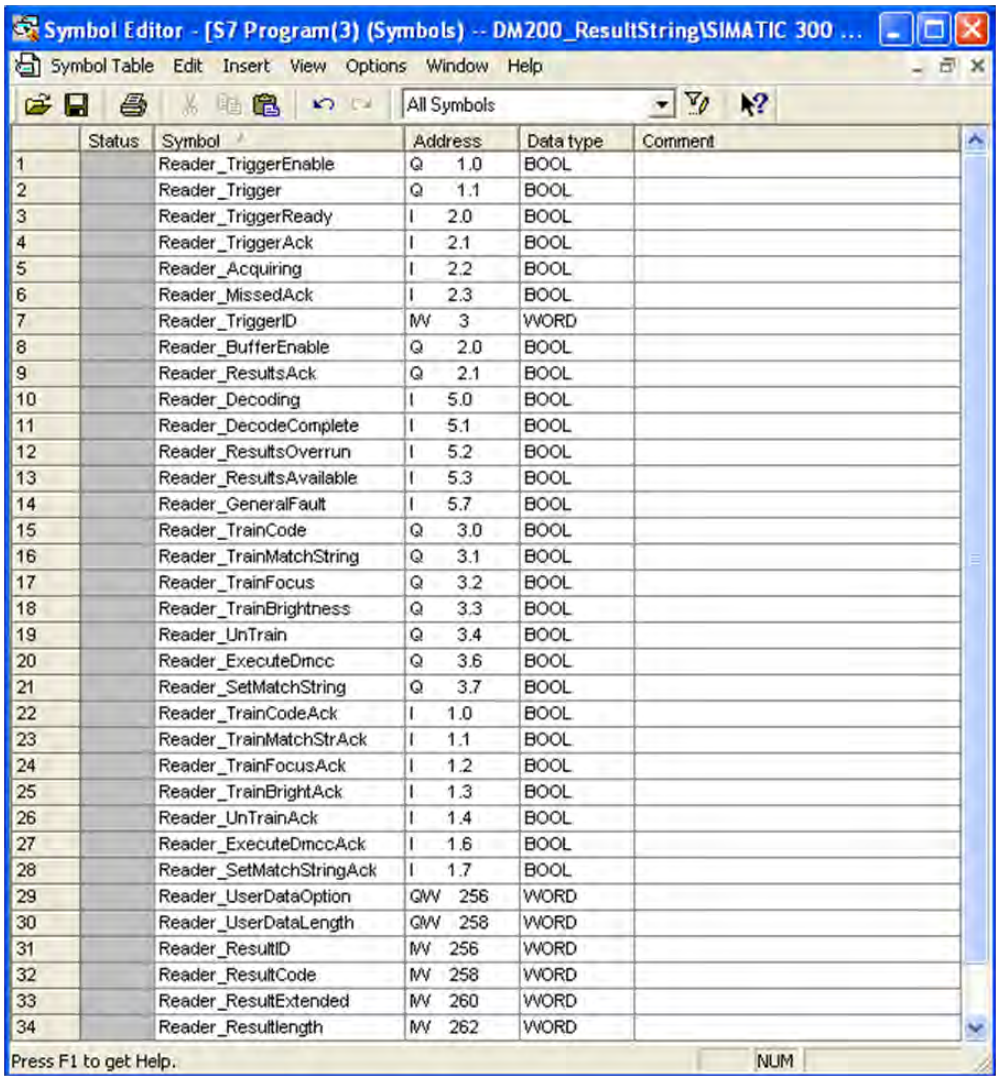
If the queue has overflowed and then buffering is disabled, there will be a greater than 1 difference between the TriggerID and ResultID values. This difference represents the number of reads that had occurred but could not be queue because the queue was full (number of lost reads equals TriggerID - ResultID - 1). After the next read, the ResultID value will return to the typical operating value of TriggerID - 1.

Siemens Examples

This section gives some examples of using the DataMan with a Siemens S7-300 PLC. It is assumed that the reader is familiar with the S7-300 and the SIMATIC programming software.

Symbol Table

Although not required, defining symbols for the DataMan I/O module elements can be extremely helpful. It makes the code much easier to read and reduces mistakes. This sample table shows symbols defined for a typical instance of a DataMan reader. Note, DataMan I/O modules may be at different addresses in your project. Make sure to adjust your symbol definitions based on the specific offsets of the I/O modules.



	Status	Symbol	Address	Data type	Comment
1		Reader_TriggerEnable	Q 1.0	BOOL	
2		Reader_Trigger	Q 1.1	BOOL	
3		Reader_TriggerReady	I 2.0	BOOL	
4		Reader_TriggerAck	I 2.1	BOOL	
5		Reader_Acquiring	I 2.2	BOOL	
6		Reader_MissedAck	I 2.3	BOOL	
7		Reader_TriggerID	MW 3	WORD	
8		Reader_BufferEnable	Q 2.0	BOOL	
9		Reader_ResultsAck	Q 2.1	BOOL	
10		Reader_Decoding	I 5.0	BOOL	
11		Reader_DecodeComplete	I 5.1	BOOL	
12		Reader_ResultsOverrun	I 5.2	BOOL	
13		Reader_ResultsAvailable	I 5.3	BOOL	
14		Reader_GeneralFault	I 5.7	BOOL	
15		Reader_TrainCode	Q 3.0	BOOL	
16		Reader_TrainMatchString	Q 3.1	BOOL	
17		Reader_TrainFocus	Q 3.2	BOOL	
18		Reader_TrainBrightness	Q 3.3	BOOL	
19		Reader_UnTrain	Q 3.4	BOOL	
20		Reader_ExecuteDmcc	Q 3.6	BOOL	
21		Reader_SetMatchString	Q 3.7	BOOL	
22		Reader_TrainCodeAck	I 1.0	BOOL	
23		Reader_TrainMatchStrAck	I 1.1	BOOL	
24		Reader_TrainFocusAck	I 1.2	BOOL	
25		Reader_TrainBrightAck	I 1.3	BOOL	
26		Reader_UnTrainAck	I 1.4	BOOL	
27		Reader_ExecuteDmccAck	I 1.6	BOOL	
28		Reader_SetMatchStringAck	I 1.7	BOOL	
29		Reader_UserDataOption	QW 256	WORD	
30		Reader_UserDataLength	QW 258	WORD	
31		Reader_ResultID	MW 256	WORD	
32		Reader_ResultCode	MW 258	WORD	
33		Reader_ResultExtended	MW 260	WORD	
34		Reader_Resultlength	MW 262	WORD	

Trigger and Get Results

Run the sample program “DM200_SampleRead” for the complete example program.

NOTE

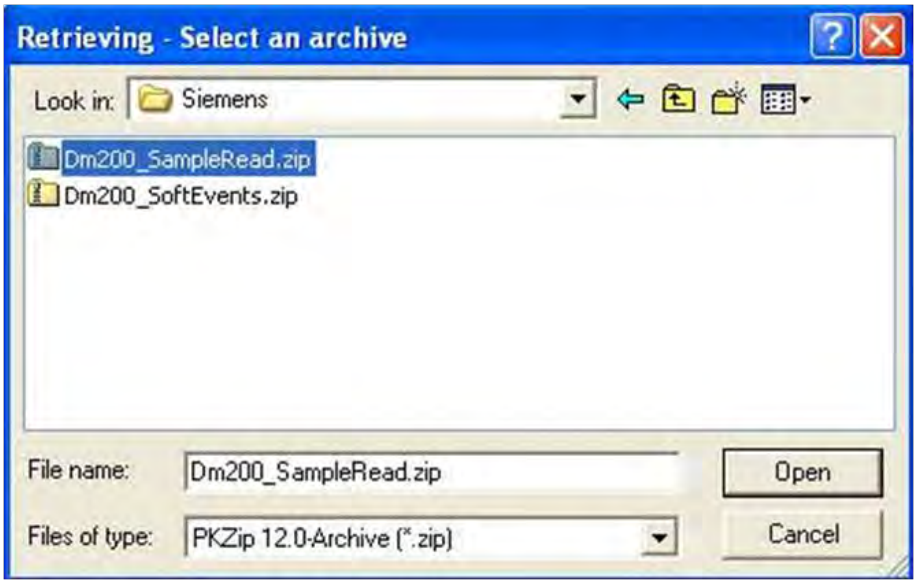
This sample can be used with any PROFINET enabled DataMan reader.

Perform the following steps to install the program:

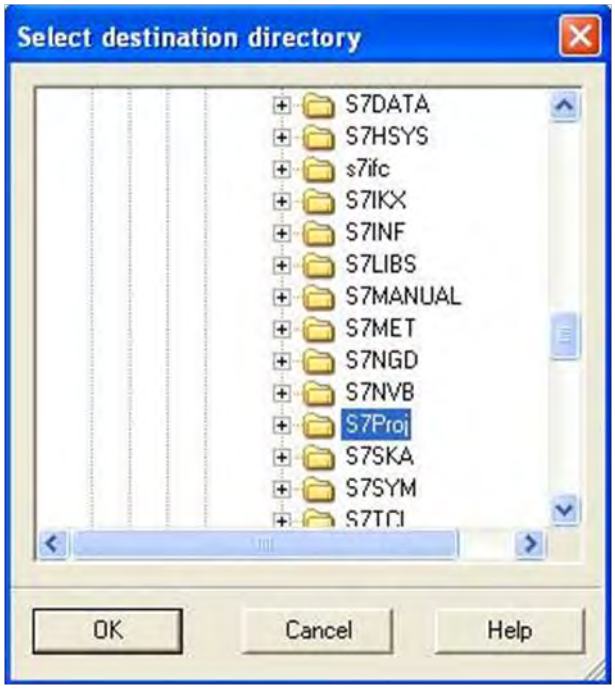
1. Start the SIMATIC Manager software.
2. Close any open applications.
3. From the main menu, select File → Retrieve…
4. Browse to find the sample file on your PC.



5. Look for the Siemens folder and select the zip file.



6. Select a destination directory to save the project on your PC.



7. The Siemens software extracts the sample archive and makes it available.

8. Reduced to the basics the process of reading and retrieving results consists of the following:

9. Define an area in your application to save read results. There are many options regarding how and where result data can be stored. For our example we define a Data

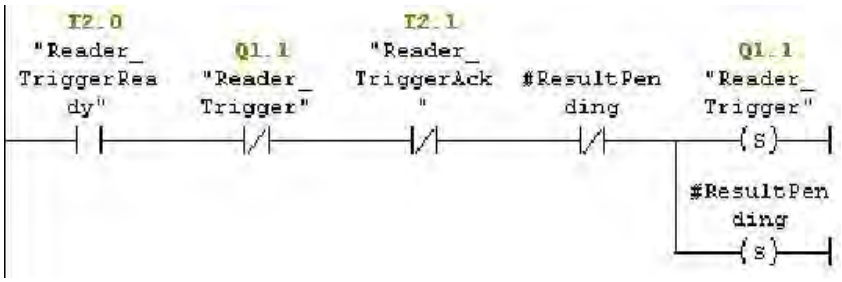
Block (DB) which contains the fields of the Result Data module that we are interested in for our application.

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	ID	INT	0	ID of this read result
+2.0	Flags	INT	0	Flags indicating success or failure
+4.0	Length	INT	0	Number of data bytes in the array Value[]
+6.0	Value	ARRAY[1..63]		Code value read
*1.0		CHAR		
=70.0		END_STRUCT		

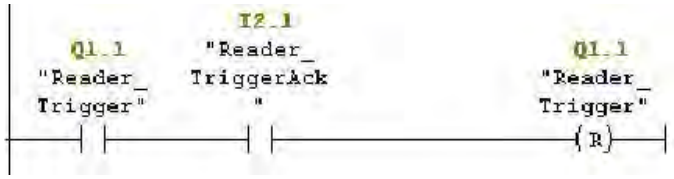
10. Enable the reader.



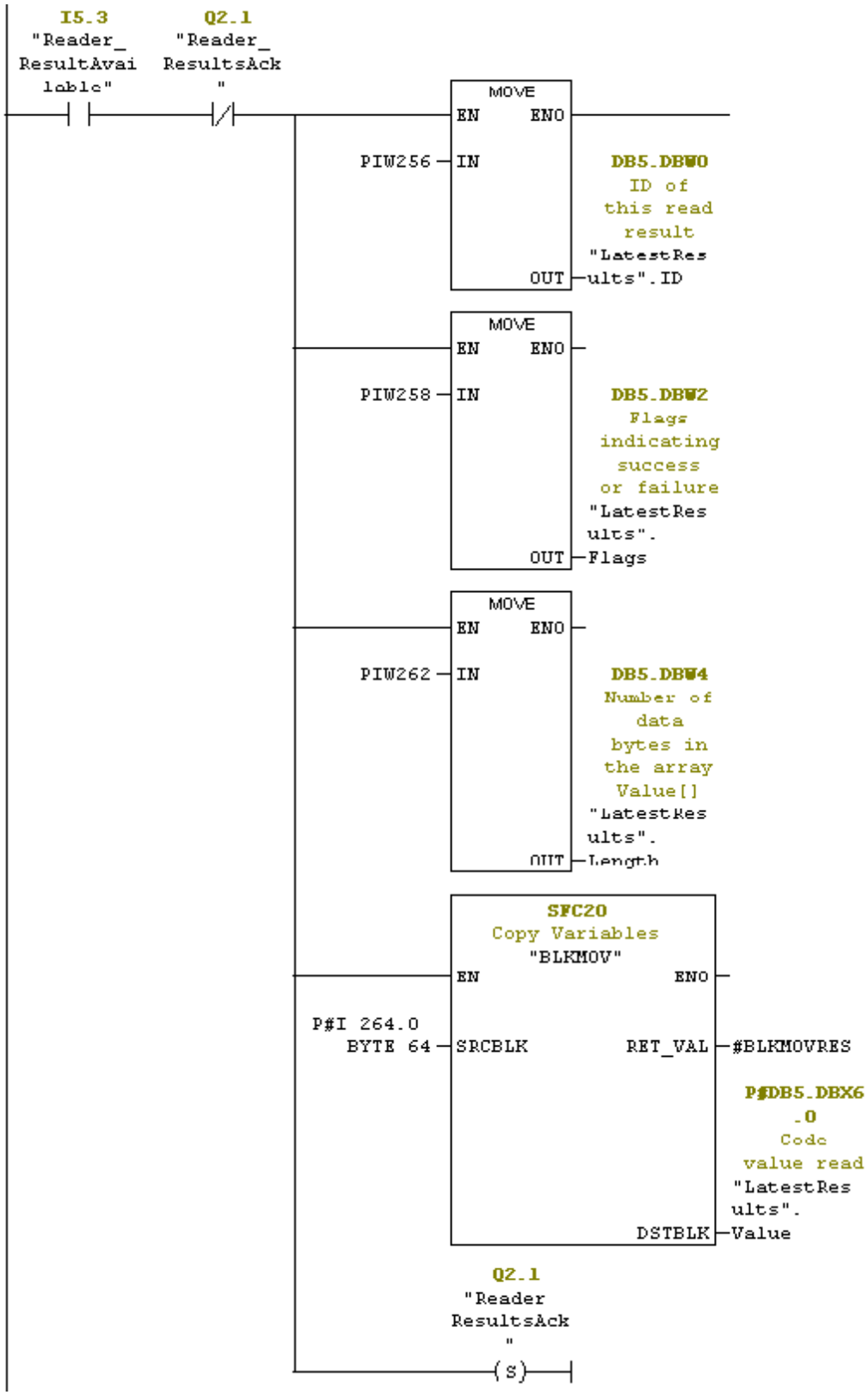
11. Set the trigger signal and set semaphore to indicate a read is pending.



12. As soon as the trigger signal is acknowledged, clear the trigger signal.



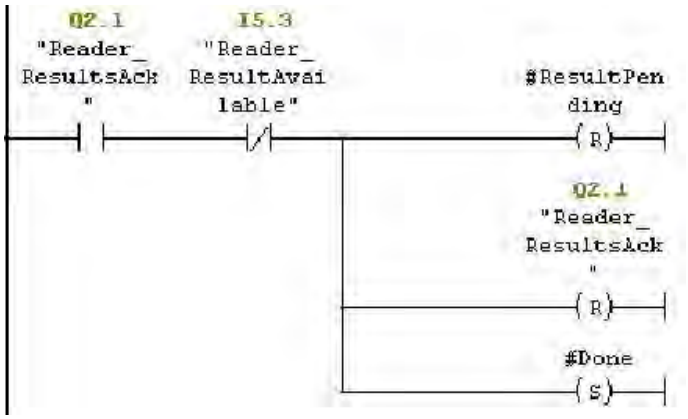
13. As soon as the results are available save a copy of the result data and set the results acknowledge signal.


I5.3
"Reader_ ResultAvai lable"
Q2.1
"Reader_ ResultsAck "
MOVE
EN
ENO
PIW256
IN
OUT
DB5.DBW0
ID of this read result
"LatestRes ults".ID
MOVE
EN
ENO
PIW258
IN
OUT
DB5.DBW2
Flags indicating success or failure
"LatestRes ults". Flags
MOVE
EN
ENO
PIW262
IN
OUT
DB5.DBW4
Number of data bytes in the array Value[]
"LatestRes ults". Length
SFC20
Copy Variables
"BLKMOV"
EN
ENO
P#I 264.0
BYTE 64
SRCBLK
RET_VAL
#BLKMOVRES
P#DB5.DBX6.0
Code value read
"LatestRes ults".
DSTBLK
Value
Q2.1
"Reader ResultsAck "
(S)

14. When the reader sees the result acknowledge signal clear result acknowledge, clear the read pending semaphore, and signal that the read process has completed.

NOTE

The reader clears “Results Available” as soon as it sees the PLC’s “Results Ack” signal.



Using Soft Events

Run the sample program “DM200_SoftEvents” for the complete example program.

NOTE

This sample can be used with any PROFINET enabled DataMan reader.

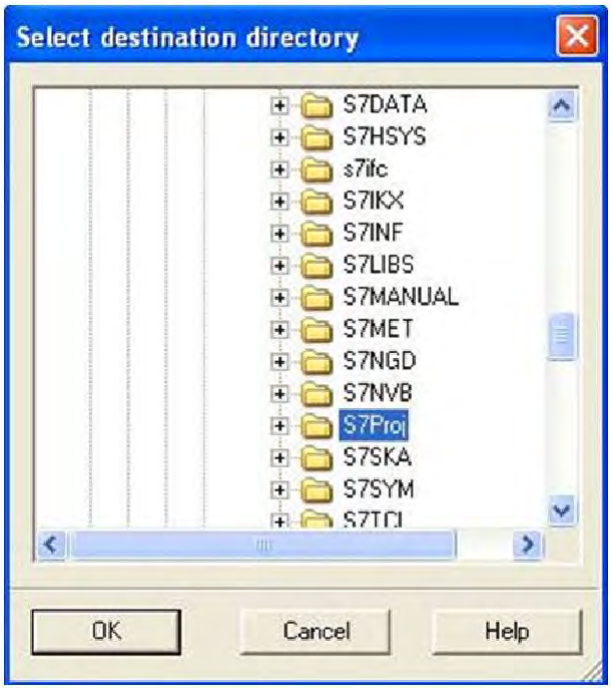
Perform the following steps to install the program:

1. Start the SIMATIC Manager software.
2. Close any open applications.
3. From the main menu, select File → Retrieve…
4. Browse to find the sample file on your PC.



5. Look for the Siemens folder and select Dm200_SoftEvents.zip.

6. Select a destination directory to save the project on your PC.

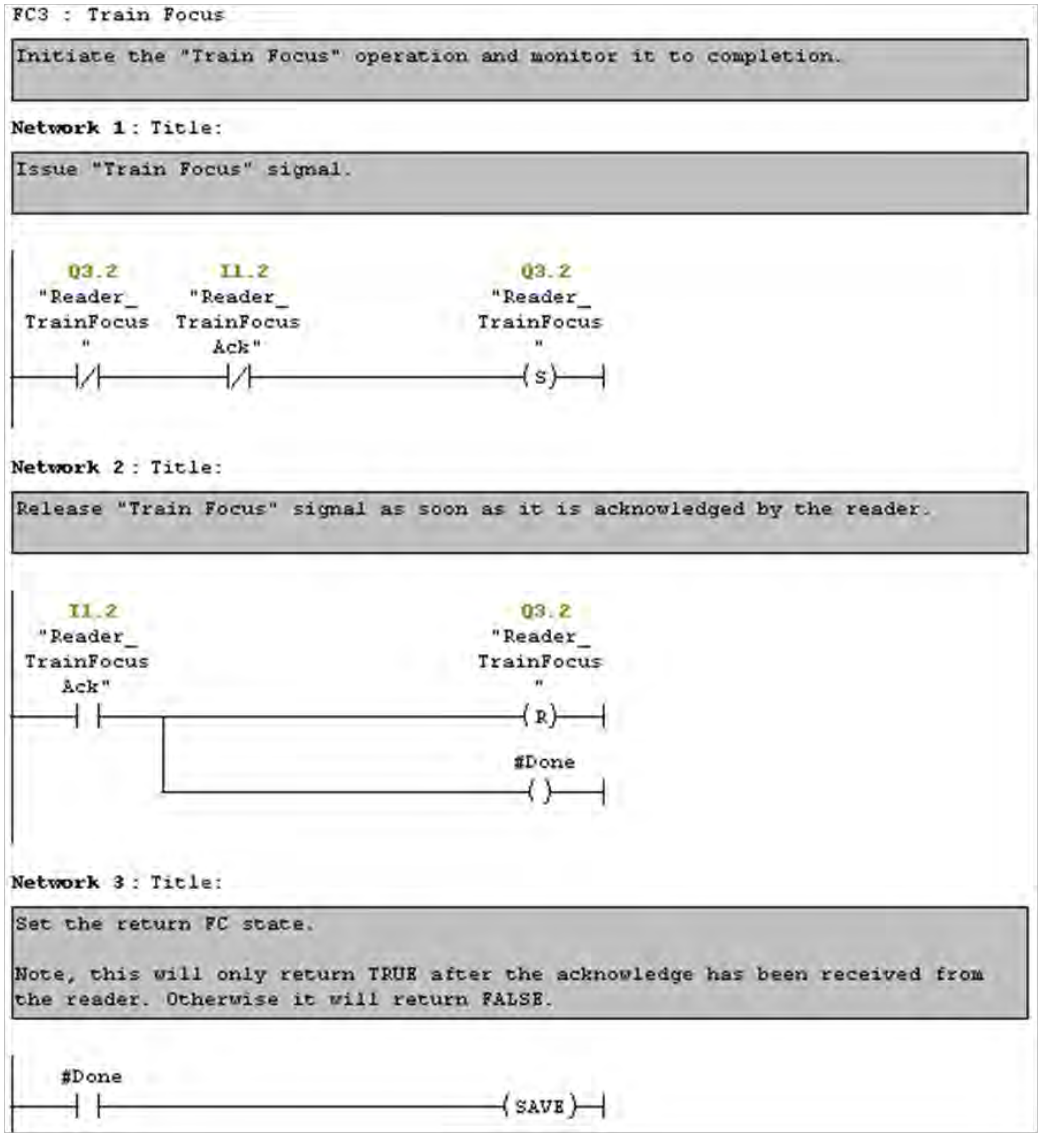


7. The Siemens software extracts the sample archive and makes it available.

Soft events are a means of invoking an activity by simply manipulating a single control bit. The activity for each bit is predefined (for more details, see Section SoftEvents). With the exception

of “Execute DMCC” and “Set Match String” all soft events may be invoked in the same way. “Execute DMCC” and “Set Match String” require the added step of loading the User Data module with application data before invoking the event.

Reduced to the basics the process of invoking a Soft Event consists of the following:



Executing DMCC Commands

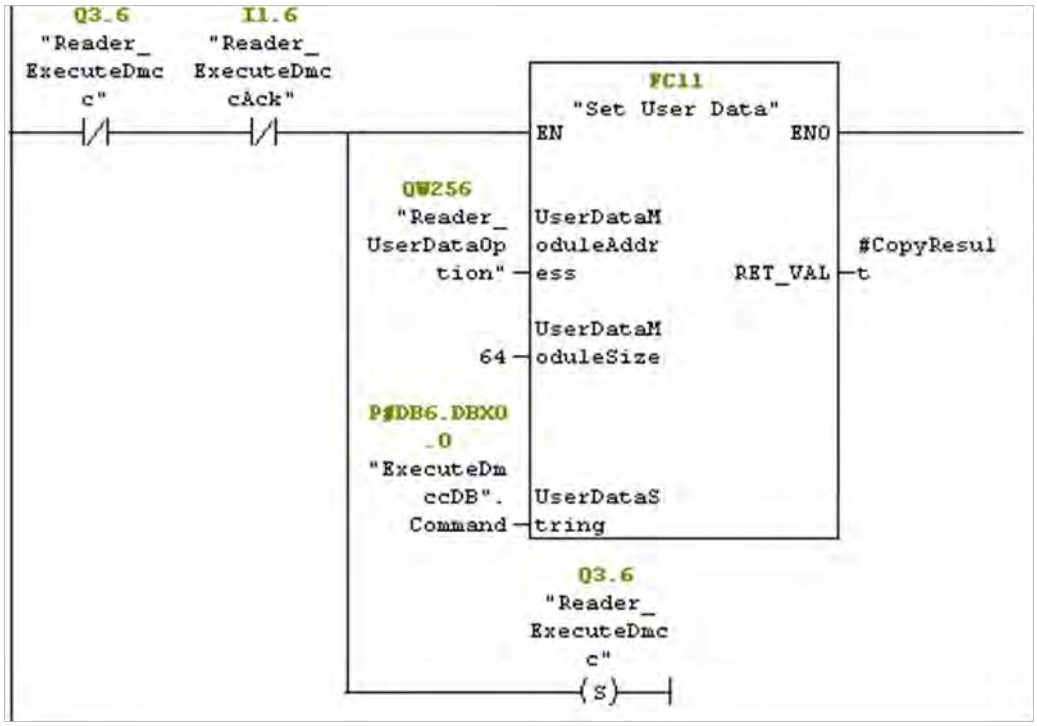
Refer to sample program “DM200_SoftEvents” for the complete example program (for information on how to install it, see Section Using Soft Events).

NOTE

This sample can be used with any PROFINET enabled DataMan reader.

“Execute DMCC” is a Soft Event which requires the added step of loading the User Data module with the desired DMCC command string before invoking the event. Note, the Soft Event mechanism does not provide a means of returning DMCC response data (other than a failure indication). So this mechanism cannot be used for DMCC “||>GET…” commands.

The process of executing a DMCC command is the same as that for all other Soft Events (see example above) except the step of invoking the Soft Event also includes copying the command string to the User Data Module. In this example the command string is exists in a Data Block. This example could be expanded to utilize a Data Block with an array of command strings that the copy function could reference by an index value. That would allow the user to pre-define all DMCC commands that are required by the application and invoke them simply by index.



The function “Set User Data” (FC11) simply copies the provided string to the User Data module. Refer to the example program for the actual STL code.