NR60 Encoder Sample Program – READ ME

NOTE: This NR60 Encoder Sample Program was created to communicate with the NR60-M28 (multi-turn 28-bit encoder) but it can be also used with either the NR60-S16 (single-turn 16-bit) or with the NR60-M30 (multi-turn 30-bit) encoders. Standard Telegram 83 (ST83) is used is this example for cyclic data exchange between the NR60 encoder and a SIEMENS controller. It is also possible to use other supported telegrams, such as ST81, ST82, ST84, and AMCI T860, and this is explained in more detail in the following text.

This NR60 encoder sample program is written assuming that you are already familiar with the NR60 encoder and are able to add it to your network. Please refer to the AMCI NR60 User Manual, which can be found on AMCI's web site, if you have any questions related to the installation.

CYCLIC DATA EXCHANGE

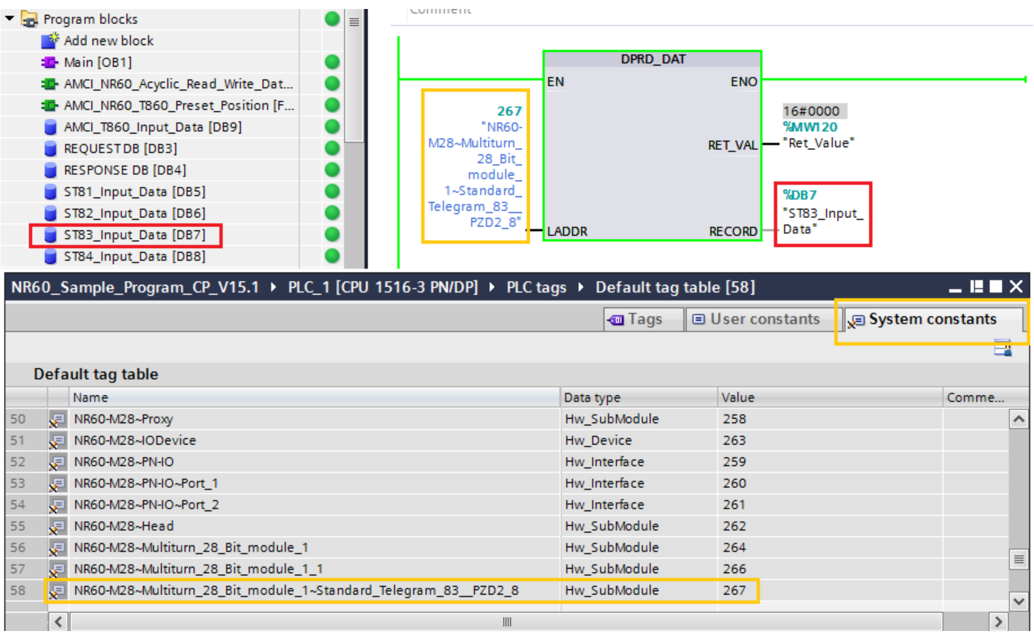
For cyclic data exchange, the NR60 encoder uses standard telegrams that cyclically update the I/O area of the SIMATIC CPU. The ST83 telegram cyclically updates the I/O input area with 16 bytes of data that includes status words, position, and velocity.

A **DPRD_DAT** instruction is used to read data from the I/O input area, store it in a dedicated data block, and ensures that consistent data from the NR60 device are all from the same bus cycle. This instruction has 3 parameters that need to be assigned:

a) The **LADDR** parameter selects the PROFINET I/O module from which data will be read. As shown in the following figure, to find an available address, open either the ***Default tag table*** or ***Show all tags*** and select the ***System constants*** tab.

b) The **RECORD** parameter defines the target ***Data Block*** (***DB***), which will contain the NR60's data that is being read by this instruction.

c) The **RET_VAL** parameter will contain an error code if an error occurs while the instruction is being executed.

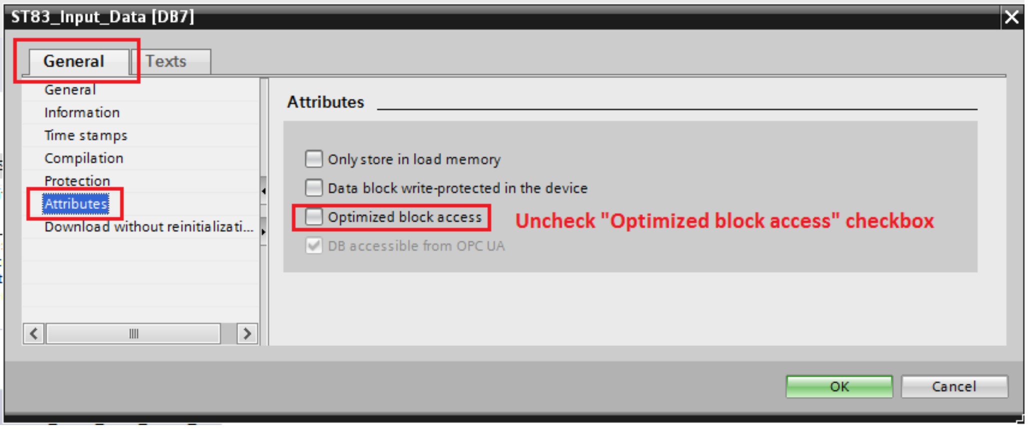


NR60_Sample_Program_CP_V15.1 ▸ PLC_1 [CPU 1516-3 PN/DP] ▸ PLC tags ▸ Default tag table [58]

Default tag table

	Name	Data type	Value	Comme...
50	NR60-M28~Proxy	Hw_SubModule	258	
51	NR60-M28~IODevice	Hw_Device	263	
52	NR60-M28~PN-IO	Hw_Interface	259	
53	NR60-M28~PN-IO~Port_1	Hw_Interface	260	
54	NR60-M28~PN-IO~Port_2	Hw_Interface	261	
55	NR60-M28~Head	Hw_SubModule	262	
56	NR60-M28~Multiturn_28_Bit_module_1	Hw_SubModule	264	
57	NR60-M28~Multiturn_28_Bit_module_1_1	Hw_SubModule	266	
58	NR60-M28~Multiturn_28_Bit_module_1~Standard_Telegram_83__PZD2_8	Hw_SubModule	267	

The ***"Optimized block access"*** attribute of the target data block (*ST83_Input_Data* [DB7] in this case) must be unchecked for the DPRD_DAT instruction to work correctly. To verify the status of this attribute, right click on the selected data block and, from the pop-up menu, choose ***Properties …*** As shown in the following image, in the ***Properties*** window under the ***General*** tab, select ***Attributes,*** and verify that the ***"Optimized block access"*** is unchecked.



ST83_Input_Data [DB7]

General | Texts

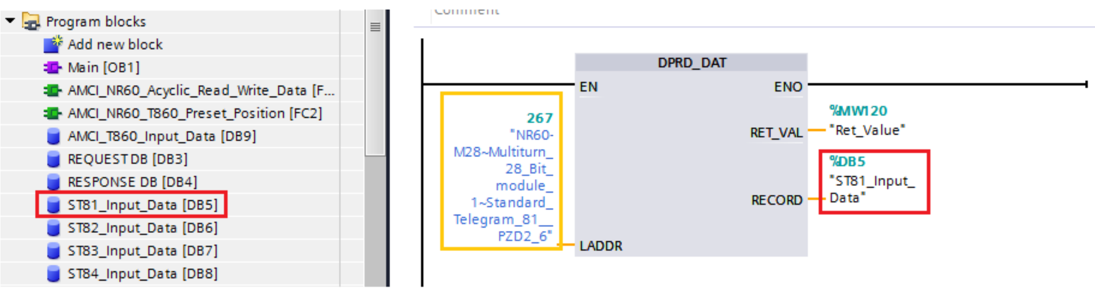
- General
- Information
- Time stamps
- Compilation
- Protection
- Attributes
- Download without reinitializati...

Attributes

☐ Only store in load memory
☐ Data block write-protected in the device
☐ Optimized block access
☑ DB accessible from OPC UA

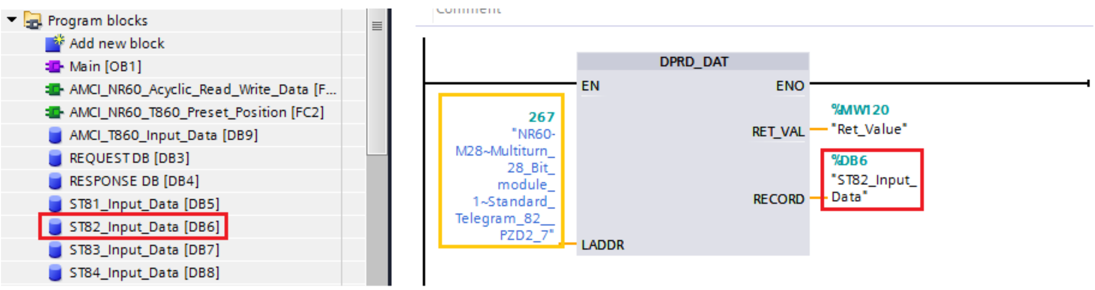
Uncheck "Optimized block access" checkbox

If you want to create a program with a different standard telegram, replace the existing ST83 with the one of your choice under the ***Device view*** tab and modify the DPR_DAT instruction to match new standard telegram. Data blocks for each standard telegram already exist:



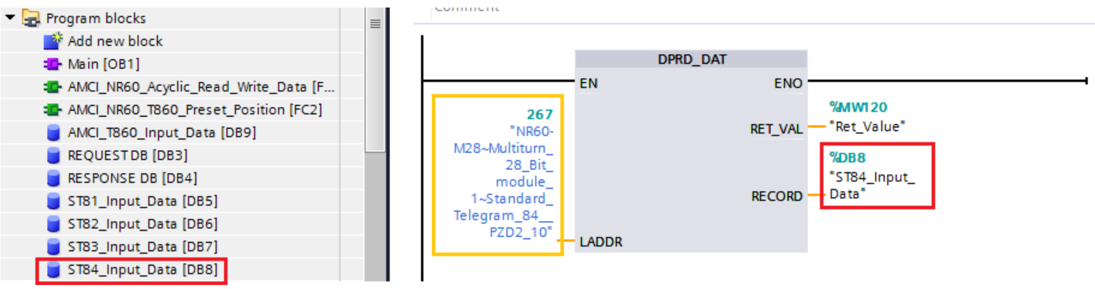
- Program blocks
  - Add new block
  - Main [OB1]
  - AMCI_NR60_Acyclic_Read_Write_Data [F...
  - AMCI_NR60_T860_Preset_Position [FC2]
  - AMCI_T860_Input_Data [DB9]
  - REQUEST DB [DB3]
  - RESPONSE DB [DB4]
  - ST81_Input_Data [DB5]
  - ST82_Input_Data [DB6]
  - ST83_Input_Data [DB7]
  - ST84_Input_Data [DB8]

DPRD_DAT: LADDR = 267 "NR60-M28~Multiturn_28_Bit_module_1~Standard_Telegram_81__PZD2_6"; RET_VAL = %MW120 "Ret_Value"; RECORD = %DB5 "ST81_Input_Data"



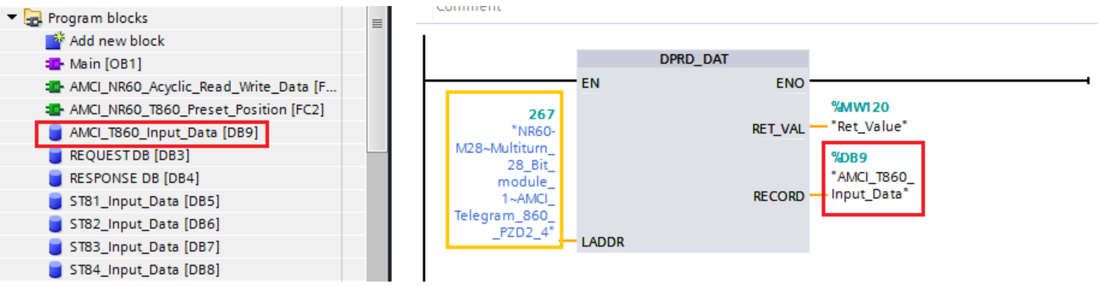
- Program blocks
  - Add new block
  - Main [OB1]
  - AMCI_NR60_Acyclic_Read_Write_Data [F...
  - AMCI_NR60_T860_Preset_Position [FC2]
  - AMCI_T860_Input_Data [DB9]
  - REQUEST DB [DB3]
  - RESPONSE DB [DB4]
  - ST81_Input_Data [DB5]
  - ST82_Input_Data [DB6]
  - ST83_Input_Data [DB7]
  - ST84_Input_Data [DB8]

DPRD_DAT: LADDR = 267 "NR60-M28~Multiturn_28_Bit_module_1~Standard_Telegram_82__PZD2_7"; RET_VAL = %MW120 "Ret_Value"; RECORD = %DB6 "ST82_Input_Data"

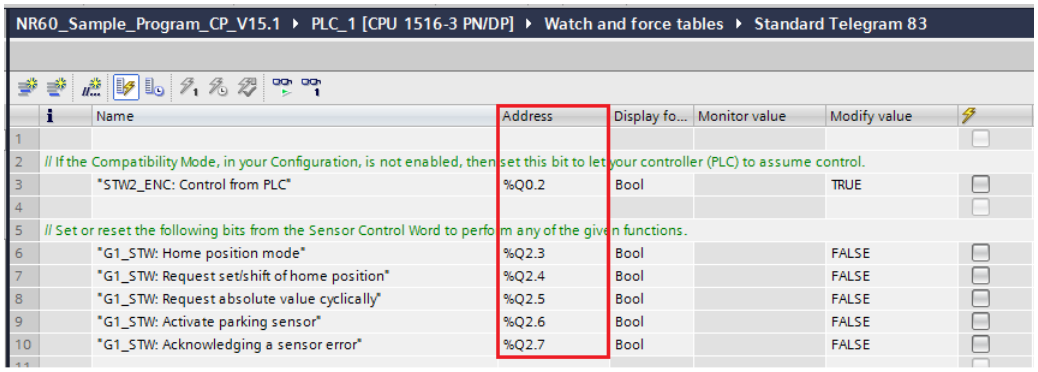


- Program blocks
  - Add new block
  - Main [OB1]
  - AMCI_NR60_Acyclic_Read_Write_Data [F...
  - AMCI_NR60_T860_Preset_Position [FC2]
  - AMCI_T860_Input_Data [DB9]
  - REQUEST DB [DB3]
  - RESPONSE DB [DB4]
  - ST81_Input_Data [DB5]
  - ST82_Input_Data [DB6]
  - ST83_Input_Data [DB7]
  - ST84_Input_Data [DB8]

DPRD_DAT: LADDR = 267 "NR60-M28~Multiturn_28_Bit_module_1~Standard_Telegram_84__PZD2_10"; RET_VAL = %MW120 "Ret_Value"; RECORD = %DB8 "ST84_Input_Data"



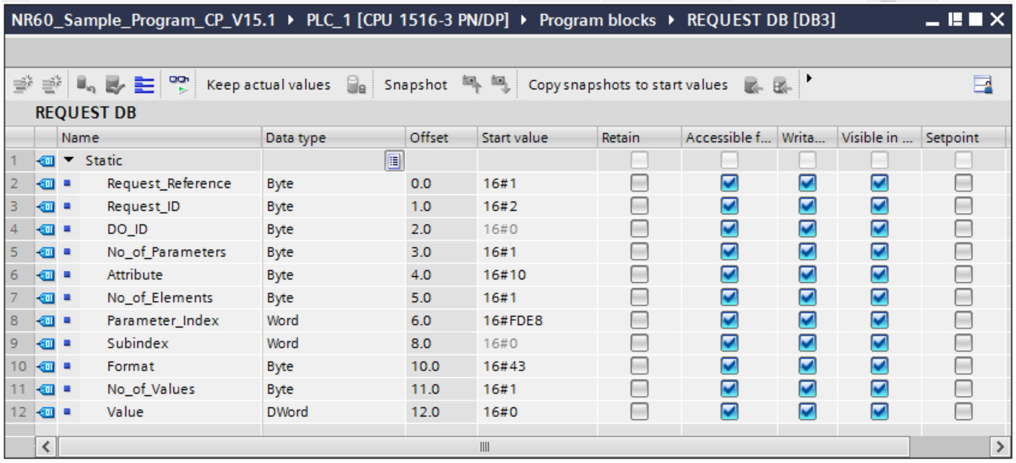
Standard Telegram ST83, like all other available telegrams, also cyclically updates the I/O output area, but with only 4 bytes of data. For this reason there is no need to use DPWR_DAT instruction and, as shown in the ***Watch and force table*** file, the data from the controller will be sent directly to the I/O output area that is used in the sample program:


NR60_Sample_Program_CP_V15.1 ▸ PLC_1 [CPU 1516-3 PN/DP] ▸ Watch and force tables ▸ Standard Telegram 83

	i	Name	Address	Display fo...	Monitor value	Modify value	⚡
1							☐
2	// If the Compatibility Mode, in your Configuration, is not enabled, then set this bit to let your controller (PLC) to assume control.						
3		"STW2_ENC: Control from PLC"	%Q0.2	Bool		TRUE	☐
4							☐
5	// Set or reset the following bits from the Sensor Control Word to perform any of the given functions.						
6		"G1_STW: Home position mode"	%Q2.3	Bool		FALSE	☐
7		"G1_STW: Request set/shift of home position"	%Q2.4	Bool		FALSE	☐
8		"G1_STW: Request absolute value cyclically"	%Q2.5	Bool		FALSE	☐
9		"G1_STW: Activate parking sensor"	%Q2.6	Bool		FALSE	☐
10		"G1_STW: Acknowledging a sensor error"	%Q2.7	Bool		FALSE	☐

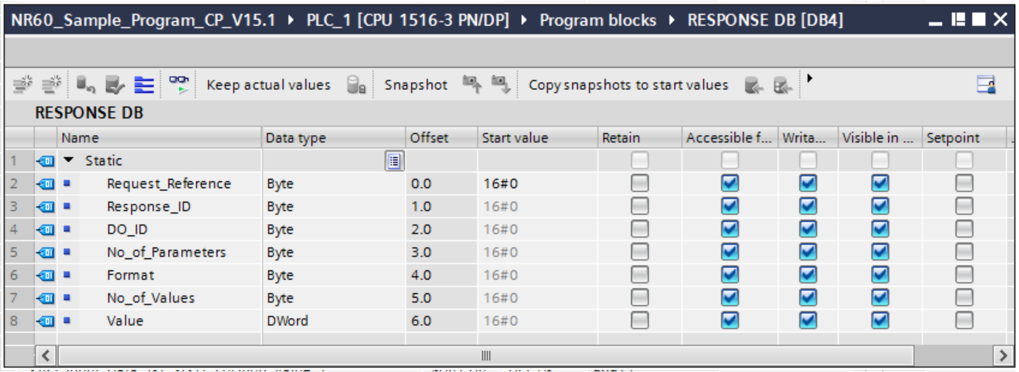
ACYCLIC DATA EXCHANGE

Acyclic data exchange writes parameters to or reads parameters from the NR60 encoder. To accomplish this task, two instructions are used: WRREC and RDREC. In addition to these two instructions, two data blocks must also be created. Note that the ***"Optimized block access"*** attribute for both data blocks must be unchecked in the same way that it was unchecked in the **Cyclic Data Exchange** section above. The first data block, named REQUEST DB, is used to hold the required request data that is sent to the NR60 encoder.


NR60_Sample_Program_CP_V15.1 ▸ PLC_1 [CPU 1516-3 PN/DP] ▸ Program blocks ▸ REQUEST DB [DB3]
Keep actual values
Snapshot
Copy snapshots to start values
REQUEST DB

	Name	Data type	Offset	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	▼ Static				☐	☐	☐	☐	☐
2	Request_Reference	Byte	0.0	16#1	☐	☑	☑	☑	☐
3	Request_ID	Byte	1.0	16#2	☐	☑	☑	☑	☐
4	DO_ID	Byte	2.0	16#0	☐	☑	☑	☑	☐
5	No_of_Parameters	Byte	3.0	16#1	☐	☑	☑	☑	☐
6	Attribute	Byte	4.0	16#10	☐	☑	☑	☑	☐
7	No_of_Elements	Byte	5.0	16#1	☐	☑	☑	☑	☐
8	Parameter_Index	Word	6.0	16#FDE8	☐	☑	☑	☑	☐
9	Subindex	Word	8.0	16#0	☐	☑	☑	☑	☐
10	Format	Byte	10.0	16#43	☐	☑	☑	☑	☐
11	No_of_Values	Byte	11.0	16#1	☐	☑	☑	☑	☐
12	Value	DWord	12.0	16#0	☐	☑	☑	☑	☐

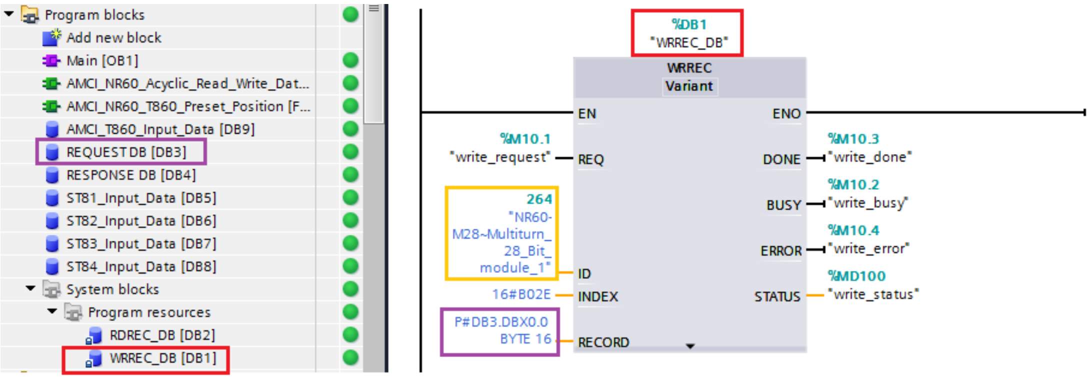
The second one, named RESPONSE DB, is used to hold the response data coming from the NR60 encoder, after the read or write request is completed.


NR60_Sample_Program_CP_V15.1 ▸ PLC_1 [CPU 1516-3 PN/DP] ▸ Program blocks ▸ RESPONSE DB [DB4]
Keep actual values
Snapshot
Copy snapshots to start values

RESPONSE DB

	Name	Data type	Offset	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint
1	▼ Static				☐	☐	☐	☐	☐
2	Request_Reference	Byte	0.0	16#0	☐	☑	☑	☑	☐
3	Response_ID	Byte	1.0	16#0	☐	☑	☑	☑	☐
4	DO_ID	Byte	2.0	16#0	☐	☑	☑	☑	☐
5	No_of_Parameters	Byte	3.0	16#0	☐	☑	☑	☑	☐
6	Format	Byte	4.0	16#0	☐	☑	☑	☑	☐
7	No_of_Values	Byte	5.0	16#0	☐	☑	☑	☑	☐
8	Value	DWord	6.0	16#0	☐	☑	☑	☑	☐

The WRREC instruction sends the read or write request to the NR60 encoder. As you can see, there is a System block (WRREC_DB) that is automatically assigned to the instruction when it is inserted in the ladder logic. The **RECORD** input is pointing to the REQUEST DB (DB3) that is 16 bytes long.



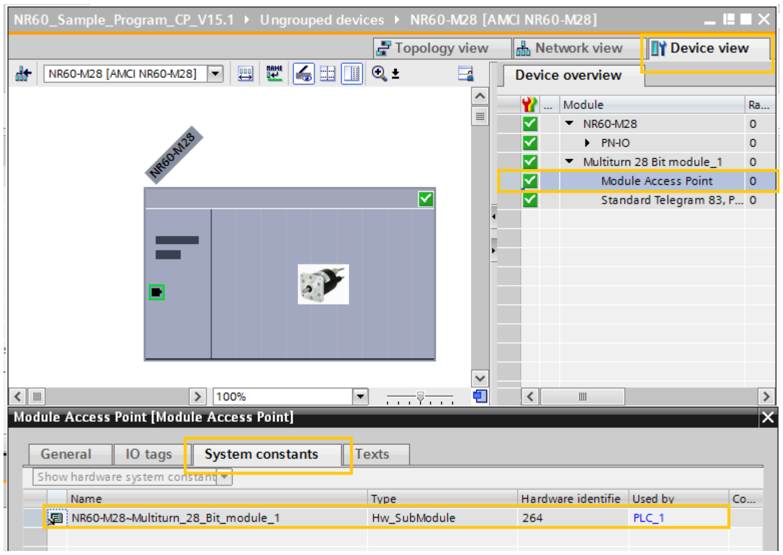
- Program blocks
  - Add new block
  - Main [OB1]
  - AMCI_NR60_Acyclic_Read_Write_Dat...
  - AMCI_NR60_T860_Preset_Position [F...
  - AMCI_T860_Input_Data [DB9]
  - REQUEST DB [DB3]
  - RESPONSE DB [DB4]
  - ST81_Input_Data [DB5]
  - ST82_Input_Data [DB6]
  - ST83_Input_Data [DB7]
  - ST84_Input_Data [DB8]
  - System blocks
    - Program resources
      - RDREC_DB [DB2]
      - WRREC_DB [DB1]

%DB1
"WRREC_DB"

WRREC
Variant

EN — ENO
%M10.1 "write_request" — REQ
DONE — %M10.3 "write_done"
BUSY — %M10.2 "write_busy"
264 "NR60-M28~Multiturn_28_Bit_module_1" — ID
ERROR — %M10.4 "write_error"
16#B02E — INDEX
STATUS — %MD100 "write_status"
P#DB3.DBX0.0 BYTE 16 — RECORD

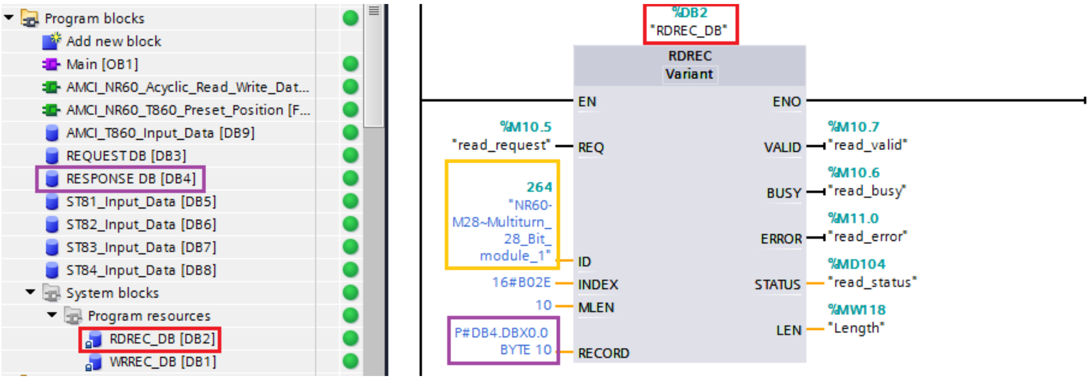
The **ID** input is a Hardware ID and, as shown in the following image, it can be found under the ***Device view*** tab, ***Module Access Point*** properties, and finally under the ***System constants*** tab.


NR60_Sample_Program_CP_V15.1 ▸ Ungrouped devices ▸ NR60-M28 [AMCI NR60-M28]
Topology view
Network view
Device view
NR60-M28 [AMCI NR60-M28]
Device overview
NR60-M28
100%
Module Access Point [Module Access Point]
General
IO tags
System constants
Texts
Show hardware system constant

Module	Ra...
▼ NR60-M28	0
▸ PN-IO	0
▼ Multiturn 28 Bit module_1	0
Module Access Point	0
Standard Telegram 83, P...	0

Name	Type	Hardware identifie	Used by	Co...
NR60-M28~Multiturn_28_Bit_module_1	Hw_SubModule	264	PLC_1	

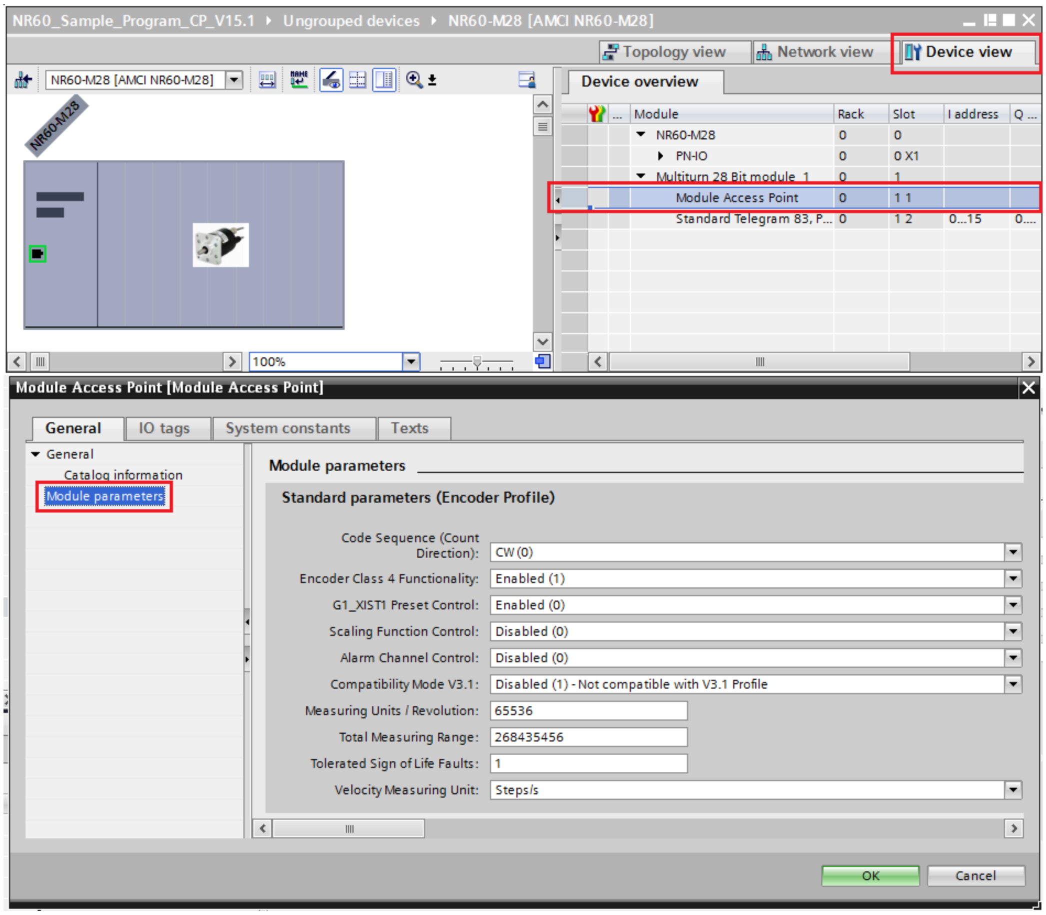
The RDREC instruction receives requested data from the NR60 encoder. There is also a System block (RDREC_DB) that is automatically assigned to the instruction. The **RECORD** input is pointing to the RESPONSE DB (DB4) that is 10 bytes long. The **ID** input is a Hardware ID that is the same as it was for the WRREC instruction.



%DB2
"RDREC_DB"
RDREC
Variant

EN — ENO
%M10.5 "read_request" — REQ
264 "NR60-M28~Multiturn_28_Bit_module_1" — ID
16#B02E — INDEX
10 — MLEN
P#DB4.DBX0.0 BYTE 10 — RECORD
VALID — %M10.7 "read_valid"
BUSY — %M10.6 "read_busy"
ERROR — %M11.0 "read_error"
STATUS — %MD104 "read_status"
LEN — %MW118 "Length"

This sample program uses the NR60's default configuration. To change any of the configuration parameters, go OFF line, make the changes to the parameters, save the changes, compile the program, and download it to the controller. Go ON line.



Module	Rack	Slot	I address	Q ...
NR60-M28	0	0		
PN-IO	0	0 X1		
Multiturn 28 Bit module_1	0	1		
Module Access Point	0	1 1		
Standard Telegram 83, P...	0	1 2	0...15	0....

Module parameters

Standard parameters (Encoder Profile)

Code Sequence (Count Direction): CW (0)
Encoder Class 4 Functionality: Enabled (1)
G1_XIST1 Preset Control: Enabled (0)
Scaling Function Control: Disabled (0)
Alarm Channel Control: Disabled (0)
Compatibility Mode V3.1: Disabled (1) - Not compatible with V3.1 Profile
Measuring Units / Revolution: 65536
Total Measuring Range: 268435456
Tolerated Sign of Life Faults: 1
Velocity Measuring Unit: Steps/s