MODBUS *RTU* COMMUNICATIONS RX/WX AND MRX/MWX

In This Chapter...

Network Slave Operation 5–2
Network Master Operation: RX/WX Instructions 5–9
Network Master Operation: DL06 MRX / MWX Instructions 5–16

Network Slave Operation

This section describes how a Modbus RTU master on a network can communicate with a D0–DCM using the Modbus RTU protocol. A network master must send a Modbus function code and Modbus address to specify a PLC memory location in the DL05/06 CPU. No CPU ladder logic is required to support Modbus slave operation.

Modbus Function Codes Supported

The D0-DCM supports the following Modbus function codes when operating as a Modbus slave.

Modbus Function Code	Function	DL05/06/205/405 Data Types Available
01	Read Discrete Output Table	Y, C, T, CT
02	Read Discrete Input Table	X, SP
03	Read Holding registers (when addressing mode is 584/984, this function is used to access analog output registers)	V
04	Read Input Registers (when addressing mode is 584/984, this function is used to access analog input registers)	V
05	Force Single Discrete Output	Y, C, T, CT
06	Preset Single Holding Registers	V
08	Loop Back / Maintenance	
15	Force Multiple Discrete Outputs	Y, C, T, CT
16	Preset Multiple Holding Registers	V

Determining the Modbus Address

There are typically two ways that most Modbus addressing conventions allow you to specify a PLC memory location. These are:

- By specifying the Modbus data type and address
- By specifying a Modbus address only.

If Your Host Software or Master Requires the Data Type and Address

Many Modbus masters allow you to specify the Modbus data type and the Modbus address that corresponds to the PLC memory location. This is the easiest method, but not all packages allow you to do it this way.

The actual equation used to calculate the address depends on the type of PLC data you are using. The PLC memory types are split into two categories for this purpose.

- Discrete – X, SP, Y, C, S, T(contacts), CT (contacts)
- Word – V-memory, Timer current value, Counter current value

In either case, you basically convert the PLC octal address to decimal and add the appropriate Modbus starting address (as required). The following tables show the exact range used for each group of data.

NOTE: *For an automated Modbus/Koyo address conversion utility, download the file Modbus conversion.xls from the* **www.automationdirect.com** *technical support > Technical and Application notes > PLC hardware> Communications> app note #* **AN–MISC–010**.

DL05 Memory Type	Qty (Dec.)	PLC Range (Octal)	Modbus Address Range	Modbus Data Type
For Discrete Data Types…		Convert PLC Addr. to Dec.	+ Start of Range	+ Data Type
Inputs (X)	256	X0 – X377	2048 – 2303	Input
Special Relays (SP)	512	SP0 – SP777	3072 – 3583	Input
Outputs (Y)	256	Y0 – Y377	2048 – 2303	Coil
Control Relays (C)	512	C0 –C777	3072 – 3583	Coil
Timer Contacts (T)	128	T0 – T177	6144 – 6271	Coil
Counter Contacts (CT)	128	CT0 – CT177	6400 – 6527	Coil
Stage Status Bits (S)	256	S0 – S377	5120 – 5375	Coil
For Word Data Types …		Convert PLC Addr. to Dec.	+	Data Type
Timer Current Values (V)	128	V0 – V177	0 – 127	Input Register
Counter Current Values (V)	128	V1000 – V1177	512 – 639	Input Register
V-Memory, user data (V)	3072	V1400 – V7377	768 – 3839	Holding Register

DL06 Memory Type	Qty (Dec.)	PLC Range (Octal)	Modbus Address Range	Modbus Data Type
For Discrete Data Types...	Convert PLC Addr. to Dec.		+ Start of Range	+ Data Type
Inputs (X)	512	X0 – X777	2048 – 2559	Input
Special Relays (SP)	512	SP0 – SP777	3072 – 3583	Input
Outputs (Y)	512	Y0 – Y777	2048 – 2559	Coil
Control Relays (C)	1024	C0 –C1777	3072 – 4095	Coil
Timer Contacts (T)	256	T0 – T377	6144 – 6399	Coil
Counter Contacts (CT)	128	CT0 – CT177	6400 – 6527	Coil
Stage Status Bits (S)	1024	S0 – S1777	5120 – 6143	Coil
Global Inputs (GX)	2048	GX0 – GX3777	0 – 2047	Input
Global Outputs (GY)	2048	GY0 – GY3777	0 – 2047	Coil
For Word Data Types ...	Convert PLC Addr. to Dec.		+	Data Type
Timer Current Values (V)	128	V0 – V177	0 – 127	Input Register
Counter Current Values (V)	128	V1000 – V1177	512 – 639	Input Register
V-Memory, user data (V)	256 3072 4096	V400 – V677 V1400 – V7377 V10000 – V17777	256 – 511 768 – 3839 4096 – 8191	Holding Register

The following examples show how to generate the Modbus address and data type for hosts which require this format.

Example 1: V2100

Find the Modbus address for User V location V2100.

1. Find V- Memory in the table.
2. Convert V2100 into decimal (1088).
3. Use the Modbus data type from the table.

PLC Addr. (Dec.) + Data Type
V2100 = 1088 decimal
1088 + Hold. Reg. = Holding Reg.

Timer Current Values (V)	128	V0 – V177	0 – 127	Input Register
Counter Current Values (V)	128	V1000 – V1177	512 – 639	Input Register
V-Memory, user data (V)	1024	V2000 – V3777	1024 – 2047	Holding Register

Example 2: Y20

Find the Modbus address for output Y20.

1. Find Y outputs in the table.
2. Convert Y20 into decimal (16).
3. Add the starting address for the range (2049).
4. Use the Modbus data type from the table.

PLC Addr.(Dec.) + Start Addr. + Data Type
Y20 = 16 decimal
16 + 2049 + Coil = Coil 2065

Outputs (Y)	320	Y0 – Y477	2049 – 2367	Coil
Control Relays (C)	256	C0 –C377	3072 – 3551	Coil

Example 3: T10 Current Value

Find the Modbus address to obtain the current value from Timer T10.

1. Find Timer Current Values in the table.
2. Convert T10 into decimal (8).
3. Use the Modbus data type from the table.

PLC Addr.(Dec.) + Data Type
T10 = 8 decimal
8 + Input Reg. = Input Reg. 8

Timer Current Values (V)	128	V0 – V177	0 – 127	Input Register
Counter Current Values (V)	128	V1000 – V1177	512 – 639	Input Register

Example 4: C54

Find the Modbus address for Control Relay C54

1. Find Control Relays in the table.
2. Convert C54 into decimal (44).
3. Add the starting address for the range (3072).
4. Use the Modbus data type from the table.

PLC Addr.(Dec.) + Start Addr. + Data

C54= 44 decimal

44 + 3072 + Coil = Coil 3117

Outputs (Y)	320	Y0 – Y477	2049 – 2367	Coil
Control Relays (CR)	256	C0 –C377	3072 – 3551	Coil

If the Host Software or Client Requires an Address ONLY

Some Modbus TCP clients do not allow you to specify the Modbus data type and address. Instead, you specify an address only. This method requires another step to determine the address, but it is not difficult. Basically, Modbus also separates the data types by address ranges as well. This means an address alone can actually describe the type of data and location. This is often referred to as "adding the offset".

The actual equation used to calculate the address depends on the type of PLC data you are using. The PLC memory types are split into two categories for this purpose.

- Discrete – X, GX, SP, Y, CR, S, T, C (contacts)
- Word – V-memory , Timer current value, Counter current value

In either case, you basically convert the PLC octal address to decimal and add the appropriate Modbus starting address (as required). The following tables show the exact range used for each group of data.

NOTE: *For an automated Modbus/Koyo address conversion utility, download the file Modbus_conversion.xls from www.automationdirect.com technical support > Technical and Application Notes > PLC hardware > Communications > app note # AN-MISC-010.*

DL05 Discrete Data Types				
PLC Memory Type	**Qty (Dec.)**	**PLC Range (Octal)**	**Modbus Address Range**	**Access**
Inputs (X)	256	X0 – X377	12049 – 123047	Read Only
Special Relays (SP)	512	SP0 – SP777	13073 – 13584	
Reserved	–	–	13585 – 20000	
Outputs (Y)	256	Y0 – Y377	2049 – 2304	Read/Write
Control Relays (CR)	512	C0 –C777	3073 – 3584	
Timer Contacts (T)	128	T0 – T377	6145 – 6272	
Counter Contacts (CT)	128	CT0 – CT177	6401 – 6528	
Stage Status Bits (S)	256	S0 – S377	5121 – 5376	
Reserved	—	—	6529 – 10000	

DL06 Discrete Data Types				
PLC Memory Type	**Qty (Dec.)**	**PLC Range (Octal)**	**Modbus Address Range**	**Access**
Global Inputs (GX)	2048	GX0 – GX1746 GX1747 – GX3777	10001 – 10999 11000 – 12048	Read Only
Inputs (X)	512	X0 – X777	12049 – 12560	
Special Relays (SP)	512	SP0 – SP777	13073 – 13584	
Reserved	–	–	13585 – 20000	
Global Outputs (GY)	2048	GY0 – GY3777	0 – 2048	Read/Write
Outputs (Y)	512	Y0 – Y777	2049 – 2560	
Control Relays (CR)	1028	C0 –C1777	3073 – 4096	
Timer Contacts (T)	256	T0 – T377	6145 – 6400	
Counter Contacts (CT)	128	CT0 – CT177	6401 – 6528	
Stage Status Bits (S)	1024	S0 – S1777	5121 – 6144	
Reserved	—	—	6529 – 10000	

DL06 Word Data					
Registers (Word)	**Qty (Dec.)**	**PLC Range (Octal)**	**Modbus 40001 Address Range**	**Modbus 30001 Address Range**	**Access**
V-Memory Timers	256	V0 – V377	40001 – 40256	30001 – 30256	Read/Write
V-Memory Counters	128	V1000 – V1177	40513 – 40460	30513 – 30640	
V-Memory Data Words	256	V400 – V777	40257 – 40512	30257 – 30512	
	3072	V1400 – V7377	40769 – 43840	30769 – 33840	
	4096	V10000 – V17777	44097 – 48192	34097 – 38192	

The following examples show how to generate the Modbus address and data type for hosts which require this format.

Example 1: V2100

Find the Modbus address for User V location V2100.

1. Find V-memory in the table.
2. Convert V2100 into decimal (1088).
3. Add the Modbus starting address for the mode (40001).

PLC Addr. (Dec.) + Mode Address
V2100 = 1088 decimal
1088 + 40001 = 41089

For Word Data Types ...	PLC Address Dec.		+	Appropriate Mode Address		
Timer Current Values (V)	128	V0 – V177	0 – 127	3001	30001	Input Reg.
Counter Current Values (V)	128	V1000 – V1177	512 – 639	3001	30001	Input Reg.
V-Memory, user data (V)	1024	V2000 – V3777	1024 – 2047	4001	40001	Holding Reg.

Example 2: Y20

Find the Modbus address for output Y20.

1. Find Y outputs in the table.
2. Convert Y20 into decimal (16).
3. Add the starting address for the range (2048).
4. Add the Modbus address for the mode (1).

PLC Addr.(Dec.) + Start Address + Mode
Y20 = 16 decimal
16 + 2048 + 1 = 2065

Outputs (Y)	320	Y0 – Y477	2048 – 2367	1	1	Coil
Control Relays (CR)	256	C0 –C377	3072 – 3551	1	1	Coil
Timer Contacts (T)	128	T0 –T177	6144 – 6271	1	1	Coil

Example 3: C54

Find the Modbus address for Control Relay C54.

1. Find Control Relays in the table.
2. Convert C54 into decimal (44).
3. Add the starting address for the range (3072).
4. Add the Modbus address for the mode (1).

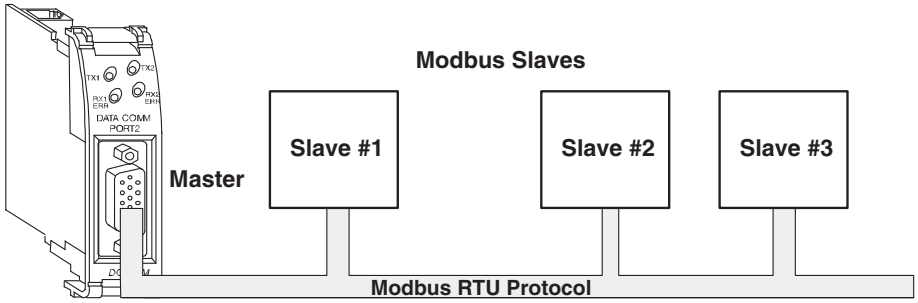
PLC Addr.(Dec.) + Start Address + Mode
C54 = 44 decimal
44 + 3072 + 1 = 3117

Outputs (Y)	320	Y0 – Y477	2048 – 2367	1	1	Coil
Control Relays (CR)	256	C0 –C377	3072 – 3551	1	1	Coil
Timer Contacts (T)	128	T0 –T177	6144 – 6271	1	1	Coil

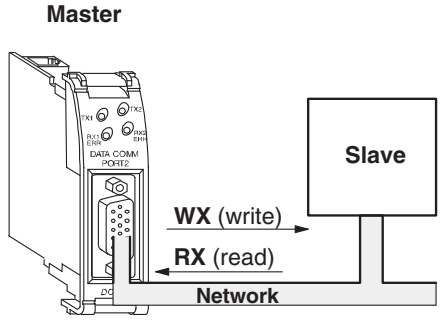
Network Master Operation: RX/WX Instructions

Overview

This section describes how the DL05/06CPU can operate as a master on a Modbus RTU network using the DCM. This section discusses how to design the required ladder logic for network master operation.



When using the DCM as a master on the network, you use simple RLL instructions to initiate the requests. The WX instruction initiates network write operations, and the RX instruction initiates network read operations. Before executing either the WX or RX commands, we need to load data related to the read or write operation onto the CPU's accumulator stack. When the WX or RX instruction executes, it uses the information on the stack combined with data in the instruction box to completely define the task.



Modbus Function Codes Supported

The DCM supports the following Modbus function codes when acting as a Modbus RTU master.

Modbus Function Code	Function	DL05/06 Data Types Available
01	Read Output Table	Y, C, T, CT
02	Read Input Table	X, SP
03	Read holding registers (when addressing mode is 584/984, this function is used to access analog output registers)	V
04	Read Input Registers (when addressing mode is 584/984, this function is used to access analog input registers)	V
15	Force Multiple Outputs	Y, C, T, CT
16	Preset Multiple Registers	V

NOTE: *The D0–DCM, as a master, does not support function code 4. Therefore, 30001 address ranges cannot be read from a slave device.*

PLC Memory Supported for Client Operation

The actual equation used to calculate the address depends on the type of PLC data you are using. The PLC memory types are split into three categories for this purpose.

- Discrete Inputs - GX, X, SP
- Discrete Outputs - Y, C, S, T, CT
- Word - Timer current value, Counter current value, Data Words

In either case, you basically take the Modbus address you are trying to target, subtract the starting Modbus of that range, convert the result to octal and add the octal number to the beginning PLC address in the appropriate PLC range. See the conversion examples on the following page. The following tables show the exact range used for each group of data.

NOTE: *For an automated* Modbus*/Koyo address conversion utility, download the file* ***modbus_conversion.xls*** *from* ***www.automationdirect.com; technical support****>Technical and Application Notes > PLC hardware > Communications > app note # AN-MISC-010.*

DL06 Discrete Data Types				
PLC Memory Type	**Qty (Dec.)**	**PLC Range (Octal)**	**Modbus Address Range**	**Access**
Global Inputs (GX)	2048	GX0 – GX1746 GX1747 – GX3777	10001 – 10999 11000 – 12048	Read Only
Inputs (X)	512	X0 – X777	12049 – 12560	
Special Relays (SP)	512	SP0 – SP777	13073 – 13584	
Reserved	–	–	13585 – 20000	
Global Outputs (GY)	2048	GY0 – GY3777	0 – 2048	Read/Write
Outputs (Y)	512	Y0 – Y777	2049 – 2560	
Control Relays (CR)	1024	C0 –C1777	3073 – 4096	
Timer Contacts (T)	256	T0 – T377	6145 – 6400	
Counter Contacts (CT)	128	CT0 – CT177	6401 – 6528	
Stage Status Bits (S)	1024	S0 – S1777	5121 – 6144	
Reserved	—	—	6529 – 10000	

DL06 Word Data					
Registers (Word)	**Qty (Dec.)**	**PLC Range (Octal)**	**Modbus 40001 Address Range**	**Modbus 30001 Address Range**	**Access**
V-Memory (Timers)	256	V0 – V377	40001 – 40256	30001 – 30256	Read/Write
V-Memory (Counters)	128	V1000 – V1177	40513 – 40460	30513 – 30640	
V-Memory (Data Words)	256	V400 – V777	40257 – 40512	30257 – 30512	
	3072	V1400 – V7377	40769 – 43840	30769 – 33840	
	4096	V10000 – V17777	44097 – 48192	34097 – 38192	

Your PC's Windows calculator can be used for number conversions (i.e. decimal to octal). The Windows calculator must be in Calculator>View>Scientific mode to enable number conversions capability.

Example 1: Calculating Word PLC Address

Find the PLC address to use to target Modbus address 41025 in a server device.

1. Subtract the beginning of the Modbus word address range (40001) from the desired Modbus address to target.
2. Convert decimal result into octal.
3. Add octal result to beginning PLC range (Input, Output or Word).

1. 41025 - 40001 = 1024 decimal
2. 1024 decimal = 2000 octal
3. V0 (octal) + 2000 (octal) = **V2000** octal

Example 2: Calculating Discrete Input PLC Address

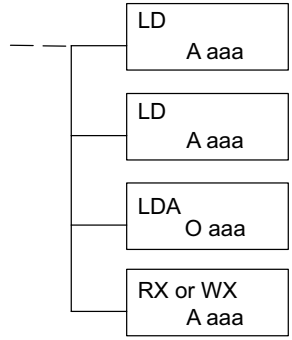
Find the PLC address to use to target Modbus address 12060 in a server device.

1. Subtract the beginning of the Modbus Input address range (12049) from the desired Modbus address to target.
2. Convert decimal result into octal.
3. Add octal result to beginning PLC range (Input, Output or Word).

1. 12060 - 12049 = 11 decimal
2. 11 decimal = 13 octal
3. X0 (octal) + 13 octal = **X13** octal

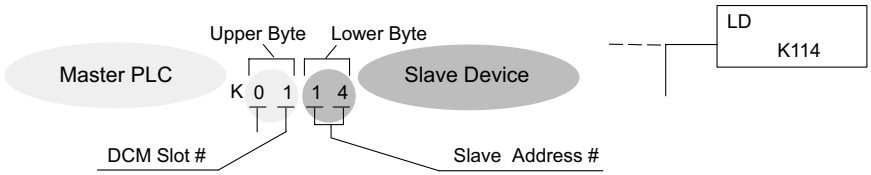
Building the Read (RX) or Write (WX) Routine

For network communications, you build the Read (RX) or Write (WX) instructions into a routine which requires the four instructions you see to the right. They must be used in the sequence shown. The following step-by-step procedure will provide you the information necessary to set up your ladder program to receive data from a network server.



Step 1: Identify ECOM Slot Location and Server Node

The first Load (LD) instruction accepts either a constant or a variable. Use a "K" to designate the number as a constant. Use a "V" if you are entering the address of a register. The contents of that register perform the same function as the constant shown below. For example, you could use V2000 in place of K0114. If the contents of V2000 is the number "114," the function would be the same. Using a variable allows changing parameters while the program is running.



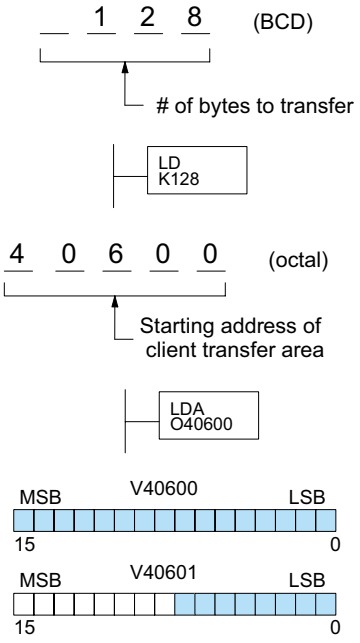
Depending on which slot the DCM is in, it has two Special Relay contacts associated with it. One indicates "Port busy", and the other indicates "Port Communication Error". The "Port Busy" bit is on while the PLC communicates with the slave. When the bit is off the program can initiate the next network request.

DL05 Special Relays	
Relay	**Option Slot**
Communication Busy	SP120
Communication Error	SP121

DL06 Special Relays				
Relay	**Slot 1**	**Slot 2**	**Slot 3**	**Slot 4**
Communication Busy	SP120	SP122	SP124	SP126
Communication Error	SP121	SP123	SP125	SP127

Step 2: Load Number of Bytes to Transfer

The second Load (LD) instruction determines the number of bytes which will be transferred between the master and slave in the subsequent WX or RX instruction. The value to be loaded is in BCD format (decimal), from 1 to 128 bytes. Requesting an even number of bytes, generates a Modbus message using Function 03, Read Holding Registers. If you need to Read Input Registers, Function Code 04, enter an odd number of bytes. For example, to read 10 Input Holding Registers, enter 2 (bytes/word) X 10 registers + 1, 21 bytes. This will request ten 30001 range addresses from the Modbus server (slave) device.



Step 3: Specify Master Memory Area

The third instruction in the RX or WX sequence is a Load Address (LDA) instruction. Its purpose is to load the starting address of the memory area to be transferred. Entered as an octal number, the LDA instruction converts it to hex and places the result in the accumulator.

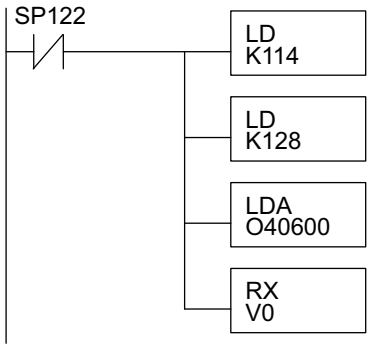
For a WX instruction, the DL05/06 CPU sends the number of bytes previously specified from its memory area beginning at the LDA address specified.

For an RX instruction, the DL05/06 CPU reads the number of bytes previously specified from the server, placing the received data into its memory area beginning at the LDA address specified.

NOTE: *Since V-memory words are always 16 bits, you may not always use the whole word. For example, if you only specify to read 3 bytes, you will only get 24 bits of data. In this case, only the 8 least significant bits of the last word location will be modified. The remaining 8 bits are not affected.*

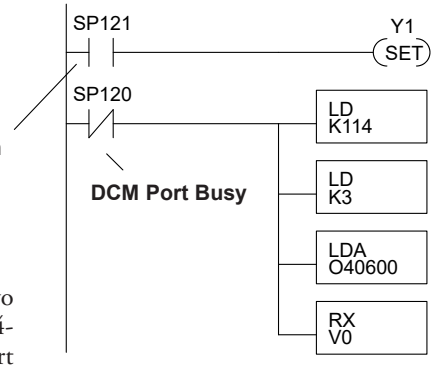
Step 4: Specify Slave Memory Area

The last instruction in our sequence is the WX or RX instruction itself. Use WX to write to the slave, and RX to read from the slave. All four of our instructions are shown to the right. In the last instruction, you must specify the starting address and a valid data type for the slave



Communications from a Ladder Program

Typically network communications will last longer than 1 scan. The program must wait for the communications to finish before starting the next transaction.



Depending on which slot the DCM is in, it has two Special Relay contacts associated with it (see page 4-11 to 4- 12 for special relays). One indicates "Port busy", and the other indicates "Port Communication Error". The example above shows the use of these contacts for an DCM that is in slot 1. The "Port Busy" bit is on while the PLC communicates with the slave. When the bit is off the program can initiate the next network request.

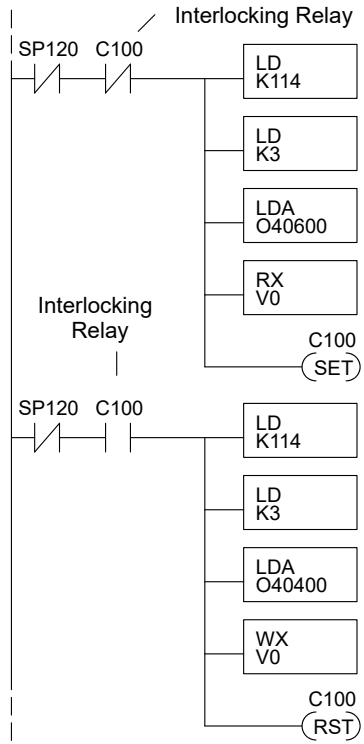
The "Port Communication Error" bit turns on when the PLC has detected an error. Use of this bit is optional. When used, it should be ahead of any network instruction boxes since the error bit is reset when an RX or WX instruction is executed.

Multiple Read and Write Interlocks

If you are using multiple reads and writes in the RLL program, you have to interlock the routines to make sure all the routines are executed. If you don't use the interlocks, then the CPU will only execute the first routine. This is because each port can only handle one transaction at a time.

In the example to the right, after the RX instruction is executed, C100 is set. When the port has finished the communication task, the second routine is executed and C100 is reset.

If your are using RLL*PLUS* Stage Programming, you can put each routine in a separate program stage to ensure proper execution and switch from stage to stage allowing only one of them to be active at a time.

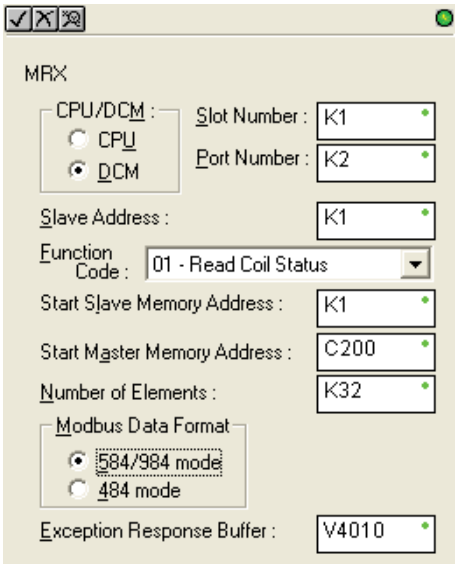


Network Master Operation: DL06 MRX / MWX Instructions

NOTE: *DirectSOFT (version 5.2 or later) and a DL06 PLC is required to use the MRX/MWX instructions.*

Modbus Read from Network (MRX)

The Modbus Read from Network (MRX) instruction is used by the DL06 DCM network master to read a block of data from a connected slave device and to write the data into V–memory addresses within the DL06 master CPU. The instruction allows the user to specify the DCM slot and port number, Modbus function code, slave station address, starting master and slave memory addresses, number of elements to transfer, Modbus data format and the exception response buffer.



- **CPU/DCM:** select DCM Port
- **Slot Number:** select the option slot the target DCM occupies (1–4)
- **Port Number:** must be D0–DCM Port 2 (K2)
- **Slave Address:** specify a slave station address (0–247)
- **Function Code:** The following Modbus function codes are supported by the MRX instruction:
  - 01 – Read Coil Status
  - 02 – Read Input Status
  - 03 – Read Holding Registers
  - 04 – Read Input Registers
  - 07 – Read Exception Status
- **Start Slave Memory Address:** specifies the starting slave memory address of the data to be read. See the table on the following page.
- **Start Master Memory Address:** specifies the starting memory address in the master where the data will be placed. See the table on the following page.
- **Number of Elements:** specifies how many coils, inputs, holding registers or input register will be read. See the table on the following page.
- **Modbus Data Format:** specifies Modbus 584/984 or 484 data format to be used
- **Exception Response Buffer:** specifies the master memory address where the Exception Response will be placed (requires 3 registers).

MRX Slave Memory Address

MRX Slave Address Ranges		
Function Code	**Modbus Data Format**	**Modbus Address Range(s)**
01 – Read Coil Status	484 Mode	1 – 999
	594/984 Mode	1 – 65535
02 – Read Input Status	484 Mode	1001 – 1999
	594/984 Mode	10001 – 19999 (5 digit) or 1000001 – 165535 (6 digit)
03 – Read Holding Registers	484 Mode	4001 – 4999
	594/984 Mode	40001 – 49999 (5 digit) or 4000001 – 465535 (6 digit)
04 – Read Input Registers	484 Mode	3001 – 3999
	594/984 Mode	30001 – 39999 (5 digit) or 3000001 – 365535 (6 digit)
07 – Read Exception Status	484 and 594/984 Mode	N/A

MRX Master Memory Addresses

MRX Master Memory Address Ranges		
Operand Data Type		**DL06 Range**
Inputs	**X**	0–1777
Outputs	**Y**	0–1777
Control Relays	**C**	0–3777
Stage Status Bits	**S**	0–1777
Timer Bits	**T**	0–377
Counter Bits	**CT**	0–377
Special Relays	**SP**	0–777
V-memory	**V**	All
Global Inputs	**GX**	0–3777
Global Outputs	**GY**	0–3777

MRX Number of Elements

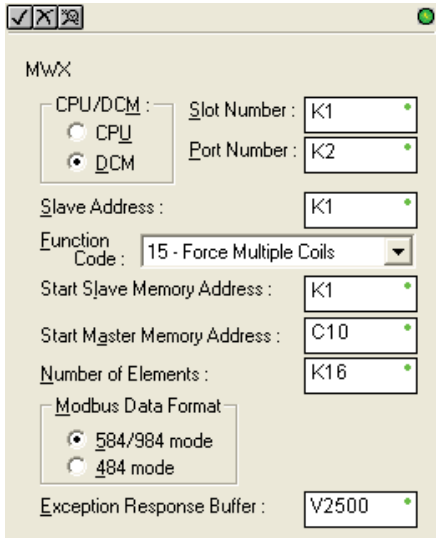
Number of Elements		
Operand Data Type		**DL06 Range**
V-memory	**V**	All
Constant	**K**	Bits: 1–2000 Registers: 1–125

MRX Exception Response Buffer

Exception Response Buffer		
Operand Data Type		**DL06 Range**
V-memory	**V**	All

Modbus Write to Network (MWX)

The Modbus Write to Network (MWX) instruction is used to write a block of data from the DL06 DCM network master memory to Modbus memory addresses within a slave device on the network. The instruction allows the user to specify the Modbus function code, slave station address, starting master and slave memory addresses, number of elements to transfer, Modbus data format and the exception response buffer.



- **CPU/DCM:** select DCM
- **Slot Number:** select the option slot the target DCM occupies (1–4)
- **Port Number:** must be D0– DCM Port 2 (K2)
- **Slave Address:** specify a slave station address (0–247)
- **Function Code:** The following Modbus function codes are supported by the MWX instruction:
  - 05 – Force Single coil
  - 06 – Preset Single Register
  - 15 – Force Multiple Coils
  - 16 – Preset Multiple Registers
- **Start Slave Memory Address:** specifies the starting slave memory address where the data will be written. See the table on the following page.
- **Start Master Memory Address:** specifies the starting address of the data in the master that is to written to the slave. See the table on the following page.
- **Number of Elements:** specifies how many consecutive coils or registers will be written to. This field is only active when either function code 15 or 16 is selected.
- **Modbus Data Format:** specifies Modbus 584/984 or 484 data format to be used
- **Exception Response Buffer:** specifies the master memory address where the exception response will be placed (requires 3 registers)

MWX Slave Memory Address

MWX Slave Address Ranges		
Function Code	**Modbus Data Format**	**Modbus Address Range(s)**
05 – Force Single Coil	484 Mode	1 – 999
	594/984 Mode	1 – 65535
06 – Preset Single Registers	484 Mode	4001 – 4999
	594/984 Mode	40001 – 49999 (5 digit) or 4000001 – 465535 (6 digit)
15 – Force Multiple Coils	484 Mode	1 – 999
	594/984 Mode	1 – 65535
16 – Preset Multiple Registers	484 Mode	4001 – 4999
	594/984 Mode	40001 – 49999 (5 digit) or 4000001 – 465535 (6 digit)

MWX Master Memory Addresses

MWX Master Memory Address Ranges		
Operand Data Type		**DL06 Range**
Inputs	**X**	0–1777
Outputs	**Y**	0–1777
Control Relays	**C**	0–3777
Stage Status Bits	**S**	0–1777
Timer Bits	**T**	0–377
Counter Bits	**CT**	0–377
Special Relays	**SP**	0–777
V-memory	**V**	All
Global Inputs	**GX**	0–3777
Global Outputs	**GY**	0–3777

MRX Number of Elements

Number of Elements		
Operand Data Type		**DL06 Range**
V-memory	**V**	All
Constant	**K**	Bits: 1–2000 Registers: 1–125

MRX Exception Response Buffer

Exception Response Buffer		
Operand Data Type		**DL06 Range**
V-memory	**V**	All

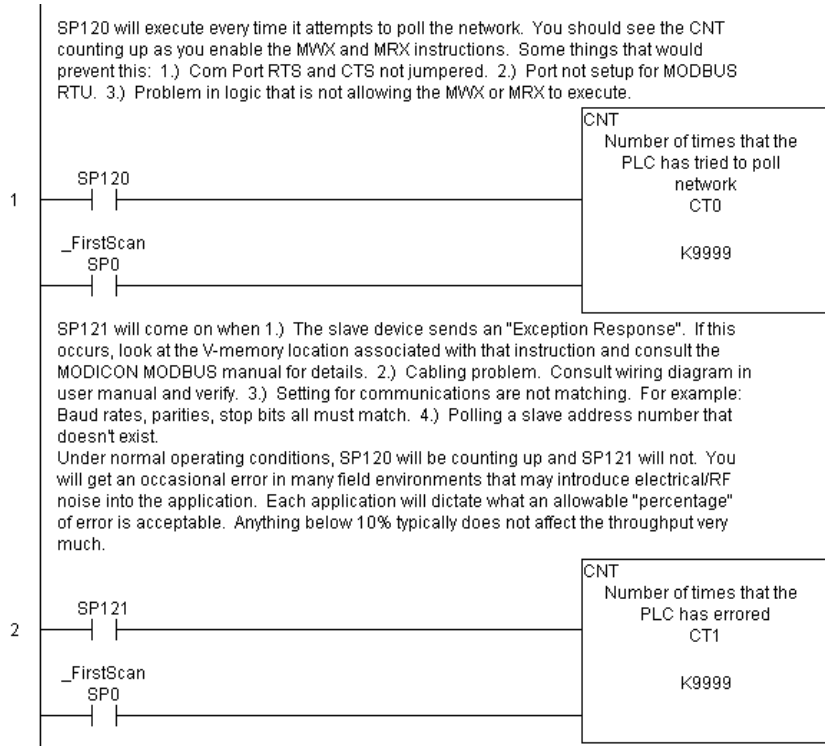
MRX/MWX Example in *Direct*SOFT

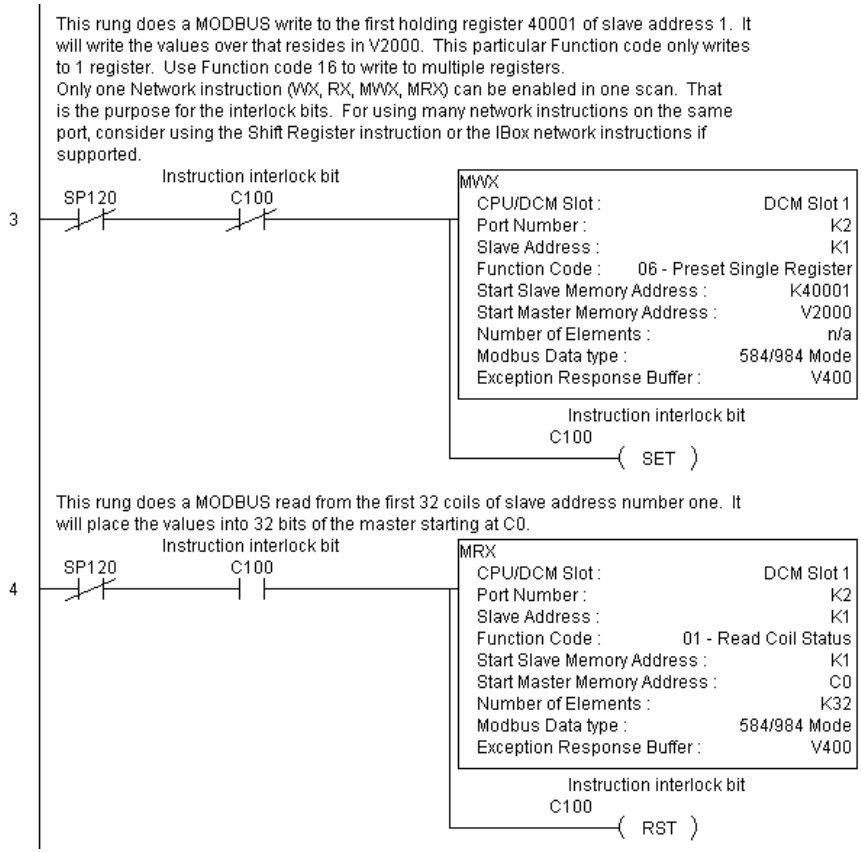
The DCM (port 2) has two Special Relay contacts associated with it (see 5–12 for comm port special relays). One indicates "Port busy" and the other indicates "Port Communication Error". The "Port Busy" bit is on while the PLC communicates with the slave. When the bit is off the program can initiate the next network request. The "Port Communication Error" bit turns on when the PLC has detected an error and use of this bit is optional. When used, it should be ahead of any network instruction boxes since the error bit is reset when an MRX or MWX instruction is executed. Typically network communications will last longer than 1 CPU scan. The program must wait for the communications to finish before starting the next transaction.

Multiple Read and Write Interlocks

If you are using multiple reads and writes in the RLL program, you have to interlock the routines to make sure all the routines are executed. If you don't use the interlocks, then the CPU will only execute the first routine. This is because each port can only handle one transaction at a time. In the example below, after the MRX instruction is executed, C100 is set. When the port has finished the communication task, the second routine is executed and C100 is reset. If you're using RLL^{PLUS} Stage Programing, you can put each routine in a separate program stage to ensure proper execution and switch from stage to stage allowing only one of them to be active at a time.

The following MRX/MWX example is for a DL06 CPU with a D0–DCM in option slot 1.




This rung does a MODBUS write to the first holding register 40001 of slave address 1. It will write the values over that resides in V2000. This particular Function code only writes to 1 register. Use Function code 16 to write to multiple registers.
Only one Network instruction (WX, RX, MWX, MRX) can be enabled in one scan. That is the purpose for the interlock bits. For using many network instructions on the same port, consider using the Shift Register instruction or the IBox network instructions if supported.
Instruction interlock bit
SP120
C100
3
MWX
CPU/DCM Slot : DCM Slot 1
Port Number : K2
Slave Address : K1
Function Code : 06 - Preset Single Register
Start Slave Memory Address : K40001
Start Master Memory Address : V2000
Number of Elements : n/a
Modbus Data type : 584/984 Mode
Exception Response Buffer : V400
Instruction interlock bit
C100
SET
This rung does a MODBUS read from the first 32 coils of slave address number one. It will place the values into 32 bits of the master starting at C0.
Instruction interlock bit
SP120
C100
4
MRX
CPU/DCM Slot : DCM Slot 1
Port Number : K2
Slave Address : K1
Function Code : 01 - Read Coil Status
Start Slave Memory Address : K1
Start Master Memory Address : C0
Number of Elements : K32
Modbus Data type : 584/984 Mode
Exception Response Buffer : V400
Instruction interlock bit
C100
RST