Networking the DL05 and DL06

All DL05 and DL06 PLCs have built-in networking capability. The DL05 family offers two 6-pin, RS-232 ports. You can use these ports for programming, networking, or connecting an operator interface device. The RS-232 ports support point-to-point communications using the optional D0-CBL cable. If you need to create a multi-drop network or require longer distances between devices, you can use the FA-ISOCON at each DL05 to convert the RS-232 signal to RS-422 or RS-485.

The DL06 family of PLCs offers even greater communications flexibility. Port 1 is a fixed baud rate port identical to port 1 on the DL05 PLCs, but port 2 is a multi-function port that can be used as RS-232, RS-422, or RS-485 (Modbus/ASCII only) without using external converters. This allows you to create multi-drop networks with minimal installation headaches.

Protocols supported

Each port is capable of communicating using K-sequence, DirectNET and Modbus RTU protocols. Port 1 can only be a Server for each of the protocols. Port 2 can serve as a K-sequence Server or a network Client or Server for either DirectNET or Modbus RTU protocols.

Serial Bus Protocols

We also offer option modules that allow you to connect a DL05 or DL06 PLC to a variety of networks as a Server device. Our D0-DEVNETS (DeviceNet) modules plug into any DL05 or DL06 PLC. The D0-DCM Data Communications module supports DirectNET and Modbus RTU protocols.

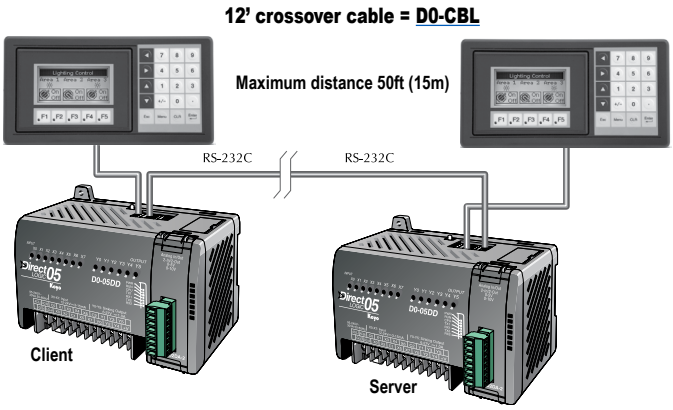
*ZIP*Link communication adatper modules

The ***ZIP***Link communications adapter modules offer fast and convenient screw terminal connection for the bottom port of the DL06 CPU. The adapter modules are RS232/422 DIP switch selectable and are offered with or without indicating LEDs and surge protection. See the Wiring Solutions section in this catalog for more information.

Optional Ethernet communication modules

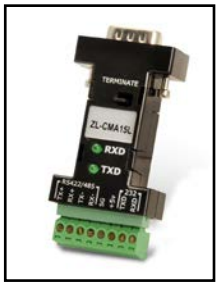
Need to connect to a high speed HMI or computer system? We offer a 100Base-T Ethernet communications module. You can use the H0-ECOM100 Ethernet communication module with our Stride Ethernet switches or with most off-the-shelf Ethernet hubs or switches. The H0-ECOM100 option module plugs into any DL05 or DL06 PLC and supports the industry standard Modbus TCP protocol.

Point-to-point

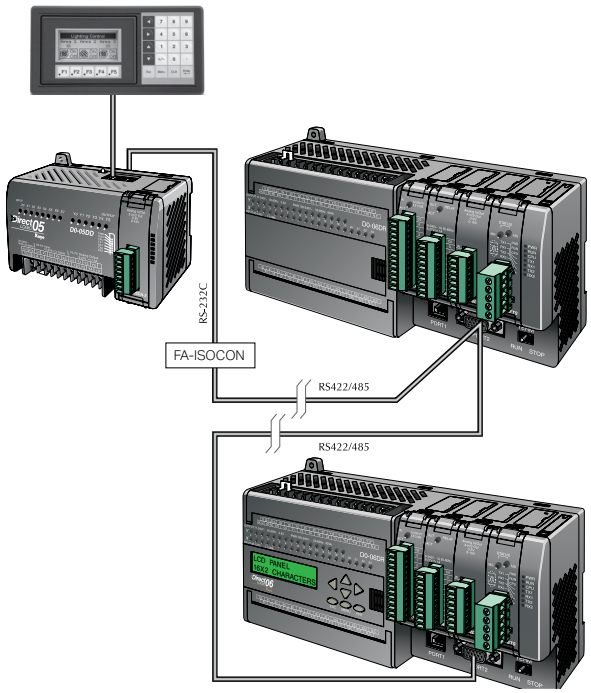


Multi-drop

ZL-CMA15



ZL-CMA15L



Maximum distance of 3,300 ft. (1000m)

Ports, Status Indicators, and Modes

Port 1

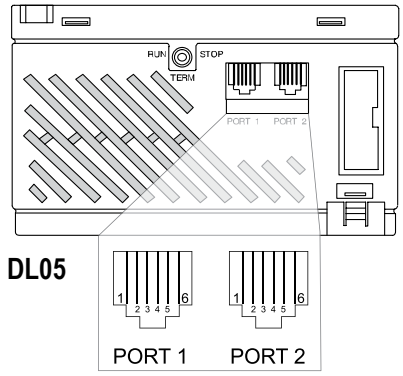
Port 1 is a 6-pin, fixed configuration port and has the same pin assignments on the DL05 and the DL06. Please refer to the table and diagrams on this page. This port can be used to connect to an HPP, DirectSOFT, an operator interface, or other external device. Features include:

- 9600 baud
- 8 data bits
- Odd parity
- 1 start bit, 1 stop bit
- Station address of 1
- Asynchronous, half-duplex, DTE

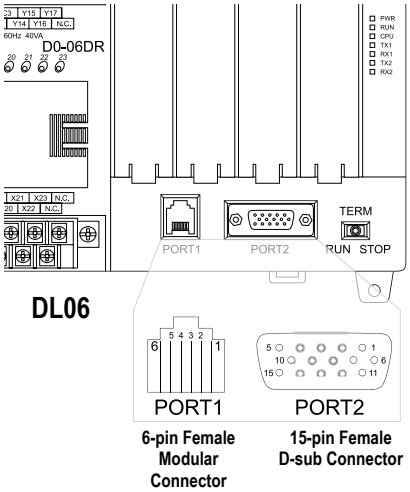
Protocols supported (as Server):

- K sequence, ***Direct***NET, Modbus RTU

DL05 & DL06 Port 1 Pin Descriptions		
1	***0V***	***Power (-) connection (GND)***
2	5V	Power (+) connection
3	RXD	Receive data (RS-232C)
4	TXD	Transmit data (RS-232C)
5	5V	Power (+) connection
6	0V	Power (-) connection (GND)



6-pin Female Modular Connector



6-pin Female Modular Connector

15-pin Female D-sub Connector

Port 2

Port 2 is a configurable port on both the DL05 and the DL06 PLCs. The DL05 PLC uses a 6-pin modular connector and offers RS-232 communications only. The DL06 PLC uses a 15-pin HD-sub connector and offers RS-232, RS-422, or RS-485 communications. Please refer to the table and diagrams on this page for more information. This port can be used to connect to an HPP, DirectSOFT, an operator interface, or other external device. Features of port 2 include:

- 300, 600, 1200, 2400, 4800, 9600 (default), 19,200, 38,400 baud
- 8 data bits
- Odd (default), even, or no parity
- 1 start bit, 1 stop bit
  - Station address:
    1 (default)
    1-90 DirectNET, K sequence
    1-247 Modbus RTU
- Asynchronous, half-duplex, DTE

Protocols supported:

- K sequence (Server), ***Direct***NET (Client/Server), Modbus (Client/Server)

DL05 Port 2 Pin Descriptions		
1	***0V***	***Power (-) connection (GND)***
2	5V	Power (+) connection
3	RXD	Receive data (RS-232C)
4	TXD	Transmit data (RS-232C)
5	RTS	Ready to send
6	0V	Power (-) connection (GND)

DL06 Port 2 Pin Descriptions		
1	***5V***	***Power (+) connection***
2	TXD	Transmit data (RS-232C)
3	RXD	Receive data (RS-232C)
4	RTS	Ready to send (RS232C)
5	CTS	Clear to send (RS232C)
6	RXD-	Receive data (-) (RS-422/485)
7	0V	Power (-) connection (GND)
8	0V	Power (-) connection (GND)
9	TXD+	Transmit data (+) (RS-422/485
10	TXD-	Transmit data (-) (RS-422/485)
11	RTS+	Ready to send (+) (RS-422/485)
12	RTS-	Ready to send (-) (RS-422/485)
13	RXD+	Receive data (+) (RS-422/485)
14	CTS+	Clear to send (+) (RS-422/485)
15	CTS-	Clear to send (-) (RS-422/485)

DL05 and DL06 status indicators

Status Indicators		
Indicator	***Status***	***Meaning***
PWR	ON	Power good
	OFF	Power failure
RUN	ON	CPU is in Run Mode
	OFF	CPU is in Stop or Program Mode
CPU	ON	CPU self diagnostics error
	OFF	CPU self diagnostics good
TX1	ON	Data is being transmitted by the CPU-Port 1
	OFF	No data is being transmitted by the CPU-Port 1
RX1	ON	Data is being received by the CPU-Port 1
	OFF	No data is being received by the CPU-Port 1
TX2	ON	Data is being transmitted by the CPU-Port 2
	OFF	No data is being transmitted by the CPU-Port 2
RX2	ON	Data is being received by the CPU-Port 2
	OFF	No data is being received by the CPU-Port 2

DL05 and DL06 mode switches

Mode Switch Position	CPU Action
RUN (Run Program)	CPU is forced into the RUN mode if no errors are encountered. No program changes are allowed by the programming/monitoring device.
TERM (Terminal)	RUN PROGRAM and the TEST modes are available. Mode and program changes are allowed by the programming/monitoring device.
STOP	CPU is forced into the STOP mode. No changes are allowed by the programming/monitoring device.

Use the optional low profile 15-pin adapter to make option module wiring easier.

ASCII and Modbus Instructions

ASCII instructions for DL06

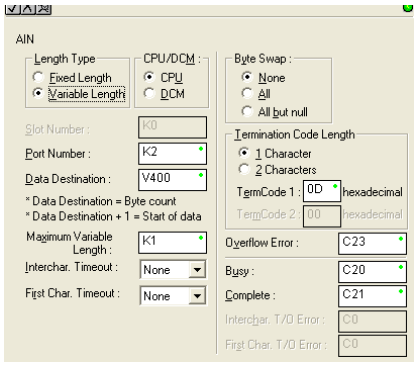
The DL06 PLC supports several easy-to-use instructions, which allow ASCII strings to be read into or written from the communication ports when using either the CPU port 2, or the D0-DCM Data Communications Module port 2.

Raw ASCII: CPU/DCM Port 2 can be used for either reading or writing raw ASCII strings, but not for both.

Embedded ASCII: With these instructions, you can use the DL06 PLC to locate ASCII strings embedded within a supported protocol via CPU/DCM Port.

Receiving ASCII strings

1. ASCII IN (AIN) - This instruction configures CPU/DCM Port 2 for raw ASCII input strings, with parameters such as fixed and variable length ASCII strings, termination characters, byte swapping options, and instruction control bits. Use barcode scanners, weigh scales, etc., to write raw ASCII input strings into CPU/DCM Port 2 based on the AIN instruction's parameters.
2. Write embedded ASCII strings directly to V-memory from an external HMI (or

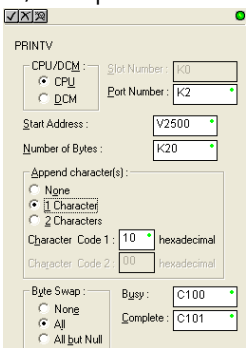


similar Client device). The ASCII string is transmitted through CPU/DCM Port 2 using any supported communications protocol. This method uses the familiar RX/WX instructions previously available.

3. If the DL06 is used as a network Client, the Network Read instruction (RX) can be used to read embedded ASCII data from a network Server device. Again, the ASCII string would be transmitted through CPU/DCM Port 2, using any supported communications protocol.

Writing ASCII strings

1. Print from V-memory (PRINTV) - Use this instruction to write raw ASCII strings out of CPU/DCM port 2 to a display panel, serial printer, etc. The instruction features the starting V-memory address, string length, byte swapping options, etc. When the instruction's permissive bit is enabled, the string is written to CPU/DCM Port 2.



2. Print to V-memory (VPRINT) - Use this instruction to create pre-coded ASCII strings in the PLC (e.g. alarm messages). When the instruction's permissive bit is enabled, the message is loaded into a pre-defined V-memory address location. Then the PRINTV instruction may be used to write the pre-coded ASCII string out of CPU/DCM Port 2. American, European, and Asian Time/Dates tamps are supported.
3. Print Message (PRINT) - This existing instruction can be used to create pre-coded ASCII strings in the PLC. When the instruction's permissive bit is enabled, the string is written to CPU/DCM Port 2. The VPRINT/PRINTV instruction combination is more powerful and flexible than the PRINT instruction.
4. If the DL06 PLC is a network Client, the Network Write (WX) can be used to write embedded ASCII data to an HMI or Server device directly from V-memory. This is done via a supported communications protocol using CPU/DCM Port 2.

More ASCII instructions

ASCII Find (AFIND) - Finds where a specific portion of the ASCII string is located in continuous V-memory addresses.

ASCII Extract (AEX) - Extracts a specific portion (usually some data value) from the ASCII find location or other known ASCII data location.

Compare V-memory (CMPV) - This instruction is used to compare two blocks of V-memory addresses and is usually used to detect a change in an ASCII string. Compared data types must be of the same format (e.g. BCD, ASCII, etc.).

Swap Bytes (SWAPB) - Swaps V-memory bytes on ASCII data that was written directly to V-memory from an external HMI or similar Client device via a communications protocol. The AIN and AEX instructions have a built-in byte swap feature.

The F0-CP128 option module is also available for more extensive ASCII communications.

Modbus RTU instructions for DL06

The DL06 CPU/DCM port 2 supports Modbus Read/Write instructions that simplify setup. The MRX and MWX instructions allow you to use native Modbus addressing, eliminating the need for octal to decimal conversions.

Function Codes 05 and 06 and the ability to read Server Exception Codes have been added. These flexible instructions allow the user to select the following parameters within one instruction window:

- 584/984 or 484 Modbus data type
- Server node (0-247)
- Function code
- Starting Client/Server memory address
- Number of bits
- Exception code starting address