

Understanding Input and Output Modules

Most motion controllers/indexers offer programmable inputs and outputs to control and interact with other external devices and machine elements.

Programmable Output Example

After indexing a table to a preset position, energize a programmable output to activate a knife that will cut material on the table.

Programmable Input Example

After indexing a table in a pick and place application, the indexer waits for an input signal from a robot arm, signaling the indexer that a part has been located on the table.

The primary reason for using I/O modules is to interface 5VDC logic signals from an indexer to switches and relays on the factory floor, which typically run on voltage levels ranging from 24VDC to 220VAC. Solid-state I/O modules are essentially a relay, utilizing light emitting diode (LED) and a transistor along with a signal conditioning circuit to activate a switch. These I/O modules isolate (no direct connection) the internal microprocessor circuitry of an indexer from oversized DC and AC voltages. The lack of a physical connection between the indexer and external devices, protects the indexer from hazardous voltage spikes and current surges.

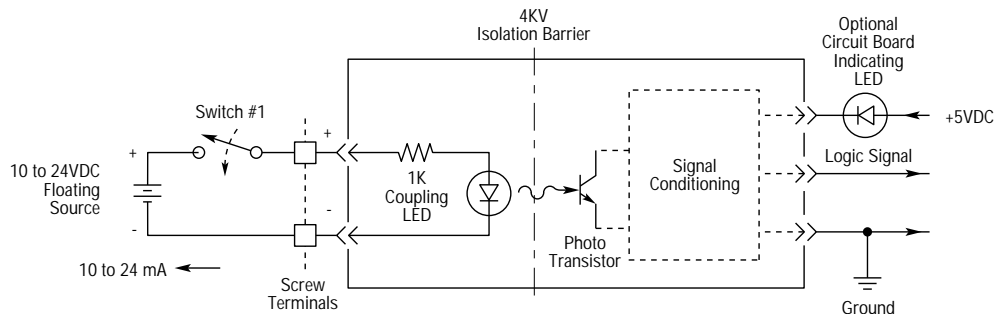
DC Input and Output Modules

As with all DC devices, this is a polarized, + and - input module. Since current will flow in only one direction, care must be taken to observe these polarities during installation.

DC input modules typically feature an input signal conditioning circuit. This circuit requires the input to remain on/off for a minimum of 5 milliseconds

before recognizing the switch. This eliminates a short voltage spike or "de-bounce" contact closure less than 5 milliseconds in duration. However, a 0.1 microfarad, ceramic disc capacitor across the actual switching contacts is still recommended to prevent switch bounce that can be as long as 10-80 milliseconds.

Fig. 5.3 Typical DC input connection diagram



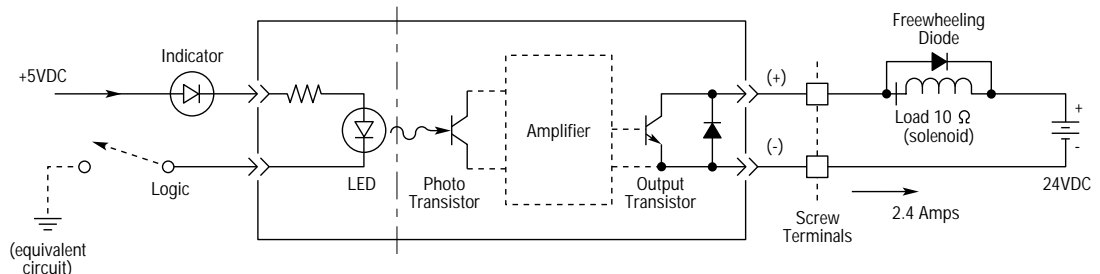
DC Input Operational Sequence

As switch #1 closes, current flows through the limiting resistor (1K ohm), and then into the LED. The light issued by the LED due to this forward current flow in turn simulates the photo transistor. Hence the term "opto" or optically isolated. The phototransistor then drives the base of the second transistor to a high level, bringing its output, or collector, to a low level.

The operation of the DC output module is similar to the DC input module. A 5VDC signal from an indexer is used to activate an LED. The output of the module is defined as open collector.

Fig. 5.4 represents a typical DC output schematic. Note the diode across the relay coil. These should always be installed to eliminate the leading inductive kick caused by the relay. A typical part number for such a diode is 1N4004. Failure to provide this protection can cause noise problems or the destruction of the output device.

Fig. 5.4 Typical DC output connection diagram



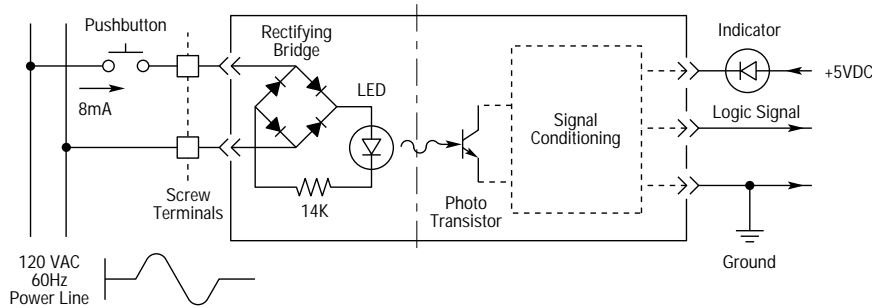
AC Input and Output Modules

AC modules are not polarized devices. This makes it virtually impossible to install a unit backwards.

AC input modules operate like DC input modules with the addition of a bridge rectifier to change AC

voltage to DC levels. AC input modules also include transient protection to filter out spikes from the AC line (caused by lightning strikes, arc welders, etc.).

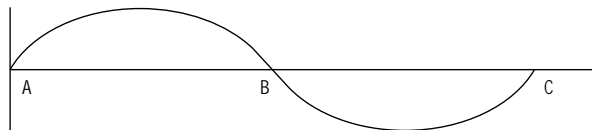
Fig. 5.5 Typical AC input connection diagram



AC output modules feature a Triac power device as its output. A Triac output offers three distinct advantages.

1. Zero voltage turn on eliminates in-rush currents to the load.
2. Zero voltage turn off eliminates inductive kick problems.
3. A snubber across the output.

Fig. 5.6 Zero voltage turn on and off

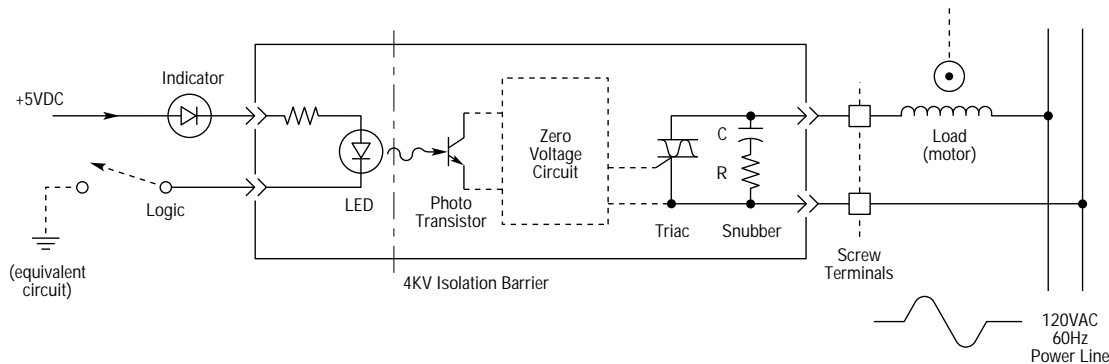


The module will only turn on or off at points A, B, or C; where the voltage is zero.

AC output modules do have leakage current, which may "turn-on" small current loads. To solve

potential problems, add a parallel resistor across the load, 5K, 5W for 120VAC and a 10K, 10W for 240VAC.

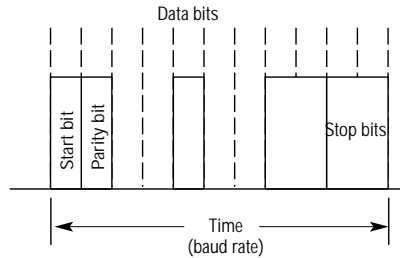
Fig. 5.7 Typical AC output connection diagram



Serial and Parallel Communications

Serial and parallel communications are methods of transferring data from a host computer to a peripheral device such as a Compumotor indexer. In the case of a Compumotor indexer, the data consist of parameters such as acceleration,

Fig. 5.8 Serial Communications



Serial

Serial communication transmits data one bit at a time on a single data line. Single data bits are grouped together into a byte and transmitted at a predetermined interval (baud rate). Serial communication links can be as simple as a 3-line connection; transmit (Tx), receive (Rx) and ground (G). This is an advantage from a cost standpoint, but usually results in slower communications than parallel communications. Common serial interfaces include RS-232C, RS-422, RS-485, RS-423.

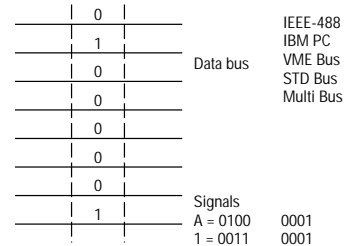
Troubleshooting

Procedure for troubleshooting 3-wire RS-232C communication.

1. Verify that the transmit of the host is wired to the receive of the peripheral, and receive of the host is wired to the transmit of the peripheral. Note: *Try switching the receive and transmit wires on either the host or peripheral if you fail to get any communication.*
2. Some serial ports require handshaking. You can establish 3-wire communication by jumpering RTS to CTS (usually pins 4 and 5) and DSR to DTR (usually pins 6 and 20).
3. Configure the host and peripheral to the same baud rate, number of data bits, number of stop bits, and parity.
4. If you receive double characters (e.g., typing "A" and receiving "AA"), your computer is set to half duplex mode. Change to full duplex mode.
5. Use DC common or signal ground as your reference, NOT earth ground.
6. Cable lengths should not exceed 50 ft. unless you are using some form of line driver, optical coupler, or shield. As with any control signal, be sure to shield the cable to earth ground at one end only.
7. To test terminal or terminal emulation software for proper 3-wire communication, unhook the peripheral device and transmit a character. An echoed character should not be received. If a character is received, you are in half duplex mode. Jumper the host's transmit and receive lines and send another character. You should receive the echoed character. If not, consult the manufacturer of the host's serial interface for proper pin outs.

velocity, move distance, and move direction configured in ASCII characters. Both communication techniques are generally bi-directional allowing the host to both transmit and receive information from a peripheral device.

Fig. 5.9 Parallel Communications



Parallel

Parallel communication requires handshaking and transmits data one byte (8 bits) at a time. When data are transferred from the host processor to a peripheral device, the following steps take place.

1. The host sets a bit on the bus signalling to the peripheral that a byte of data has been sent.
2. The peripheral receives data and sets a bit on the bus, signalling to the host that data have been received.

The advantage of communicating in parallel vs. serial is faster communications. However, since parallel communications require more communication lines, the cost can be higher than serial communications.

Parallel bus structures include:

IEEE-488, IBM PC, VME, MULTIBUS, Q and STD.

Troubleshooting

Procedure for troubleshooting parallel communication.

1. Make certain the address setting of the peripheral device is configured properly.
2. Confirm that multiple boards are not set to the same address (and each board is sealed properly into a slot).
3. Verify that peripheral subroutines to reset the board, write data, and read data work properly. Follow the handshaking procedure outlined in the device's user manual.

Note: Compumotor bus-based indexers come complete with a diskette that includes pretested programs to verify system functions and routines for simple user program development.