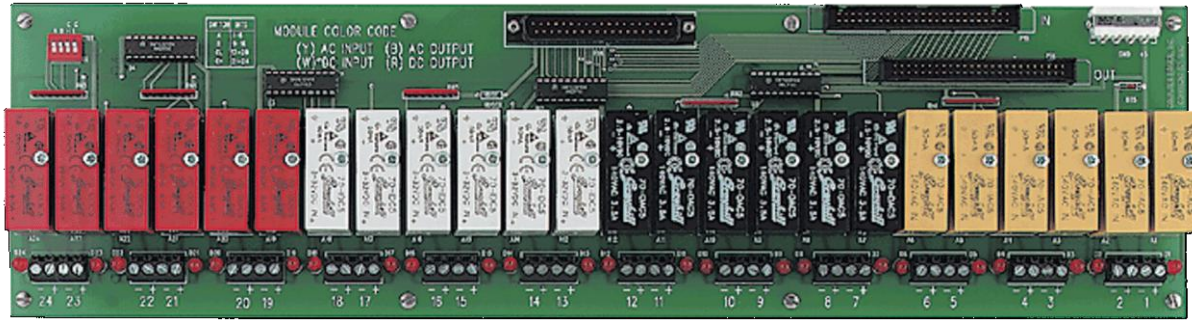


# SSR-RACK24

## 24-Channel Solid-State Relay (OPTO22, Gordos) Mounting & Interface Rack

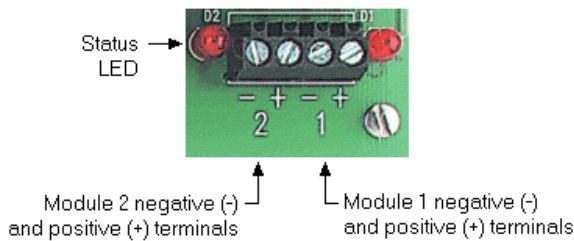


### Functional description

The SSR-RACK24 is a mounting and interface rack for use with industry-standard size solid state I/O modules. These modules provide 1500 V of isolation from harsh electrical environments for both digital inputs and outputs. Onboard TTL-level drivers provide the necessary drive current to control any compatible 5 V solid-state relay.

### Screw terminal wiring

Each SSR-RACK24 module has two independent screw terminals. Use 12-22 AWG wire to connect signals to the screw terminals. The status LED is ON when a module is active.

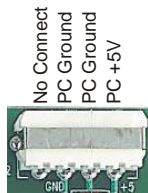


The screw terminal/module numbers correspond to 8255 ports:

- 1 to 8 correspond to FIRSTPORTA Bit 0 to Bit 7
- 9 to 16 correspond to FIRSTPORTB Bit 0 to Bit 7
- 17 to 20 correspond to FIRSTPORTC Low Bit 0 to 3
- 21 to 24 correspond to FIRSTPORTC High bits 4 to 7

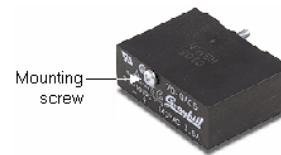
### Molex connector

The Molex connector connects to the PC power expansion plug inside the PC.



### Solid-state I/O modules

The SSR-RACK24 has 24 mounting positions for solid state I/O modules manufactured by Gordos, OPTO22, Grayhill, Western Digital, and others. These modules provide 1500 V of isolation from harsh electrical environments for both digital inputs and outputs. Mounting screw threads are provided for you to easily install the SSR modules. An example of an SSR module and mounting screw is shown below.

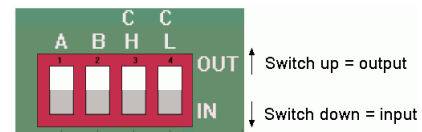


Contact MCC for compatible modules from the SSR-OAC, SSR-ODC, SSR-IAC and SSR-IDC module series.

### Module input/output control

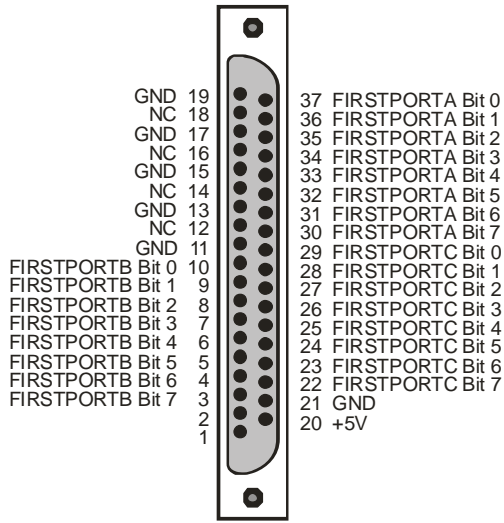
Set the module type for input or output using the DIP switch for each port.

- A** – FIRSTPORTA control
- B** – FIRSTPORTB control
- CH** – FIRSTPORTC High (controls four bits)
- CL** – FIRSTPORTC Low (controls four bits)



### 37-pin connector

The SSR-RACK24 has one 37-pin D type connector (P14). Use a C37FF-x cable to connect with 24-bit DIO boards, such as the USB-DIO24/37.

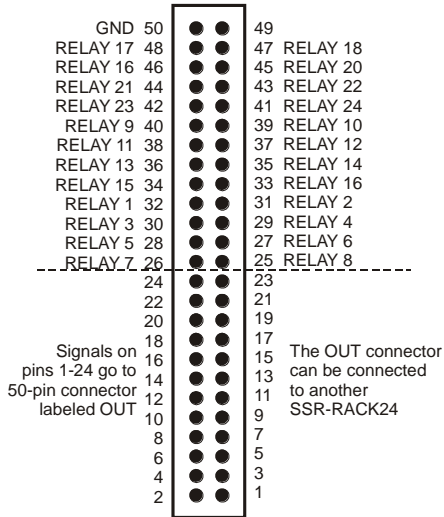


Do not make connections to pins marked "NC".

### 50-pin connectors

The SSR-RACK24 has two 50-pin connectors labeled IN (P15) and OUT (P16). Use a C50FF-x cable to connect with 48-bit DIO boards, such as the PCI-DIO48.

- Connect a cable from a DIO device to the **IN** connector.
- Connect a cable from another SSR-RACK24 to the **OUT** connector.



### Onboard buffers

Because most manufacturers of SSR racks do not supply output buffers on the SSR rack, simple digital I/O boards such as the PCI-DIO24, PCI-DIO96, and other manufacturer's 82C55-based digital I/O boards do not have the power to switch the SSRs. For example, to use an OPTO22 PB16 rack, you need to use a high-drive DIO board such as the PCI-DIO24H or PCI-DIO96H.

In order to be usable with all common TTL output boards, the SSR-RACK24 is designed with onboard buffers. With these buffers, you can plug directly into the SSR-RACK24 from your PCI-DIO24, PCI-DIO96, or any other manufacturer's 82C55-based digital I/O board.

### SSR-RACK inverting logic

On an input SSR, the presence of a voltage raises the TTL output of the SSR from TTL low to TTL high. Because the SSR-RACK uses the conventional inverting logic, the completed circuit of SSR and SSR-RACK lowers the signal to the DIO board from TTL high (+5) to TTL low (GND). An output SSR completes a circuit when the DIO board TTL signal to the SSR is low. The circuit through the SSR is open when the signal from the DIO board is high.

### Converting the SSR-RACK from inverting logic

The SSR-RACK follows the convention set by OPTO22 and followed by virtually all SSR rack manufacturers — the sense and control logic for the relays is inverted. This means that a 0 output from the digital I/O board causes an output relay to activate (complete the circuit), while a 1 (TTL high) causes the relay to deactivate. The chips which invert the logic are socketed and can be easily replaced with chips that do not invert the logic. Discuss your order with a technical sales engineer if you need non-inverting logic.