

Digital I/O Hardware

Introduction

Although many researchers and engineers often think of Keithley as a manufacturer of sensitive analog instruments, digital input/output boards make up a substantial part of the company's data acquisition products. Modern measurements often require reading switches and status signals, as well as setting indicators, relays and interlocks. Almost every experiment or test system includes a lot of digital I/O "glue." These on/off controls may not be the most glamorous part of the system, but they must work properly to obtain accurate measurements. This document reviews the use of Keithley's digital I/O line, with special attention to points that novice users may overlook or misunderstand. It is not intended to provide installation procedures for specific hardware or software products. Refer to the appropriate User's Guide for that information.

Keithley produces digital I/O hardware in four board formats:

- ISA The circuit board has one or two sections of card edge connector pins with 0.100" spacing. When examining the component side of the card with the bus connector at the bottom, the mounting bracket is on the right.
- PCI The circuit board has one or two sections of card edge connector pins with 0.050" spacing. When examining the component side of the card with the bus connector at the bottom, the mounting bracket is on the left.
- PCMCIA The circuitry is enclosed in a plug-in package approximately the size of a credit card. This format is typically used in portable computers.
- MetraBus The circuit board has cable connections to Keithley's MetraBus system. Although this product is not discussed in detail in this document, its operating principles are similar to other Keithley digital I/O products.

Hardware issues

Digital circuits are made from the same components (transistors, resistors, capacitors, etc.) as analog ones. The difference is that while digital circuits are optimized to work at the extremes—either ON or OFF—analog devices are usually biased somewhere in the linear range in between. The term "logic levels" describes the voltage ranges that the digital circuit interprets as being ON or OFF. For TTL (Transistor-Transistor Logic, the first widely used logic family), these levels are:

"0" input = 0.0–0.8V

"1" input = 2.0–5.0V

Note two important points:

The range is not symmetrical. Due to its circuit design, a TTL output is very good at pulling down to ground (basically, the output stage is a transistor switch in saturation), with a well defined "0" or low output. The "1" output could be as simple as a pull-up resistor, so its range is defined more broadly. These same output levels also apply to the inputs as to the outputs— that is essential in order for the logic family to work together.

The range is not continuous. There is a gap in the values between 0.8V and 2.0V, a "gray zone" in which the signal is ambiguous, defined as neither a '0' nor a '1' state. Digital circuits normally switch (change logic levels) so quickly that the signal is not in this undefined region long enough for the circuit to misinterpret it. However, this can lead to difficulties when attempting to read an analog or slow-moving signal.

These characteristics are properties of TTL logic, and other types differ— for example, some CMOS logic has symmetrical and very nearly gapless logic levels. However, given that TTL came first, can easily drive high load currents, and is less susceptible to damage from static electricity, it has become the de facto standard for digital I/O. Even devices implemented with other types of circuitry often use these logic levels to allow for backward compatibility.

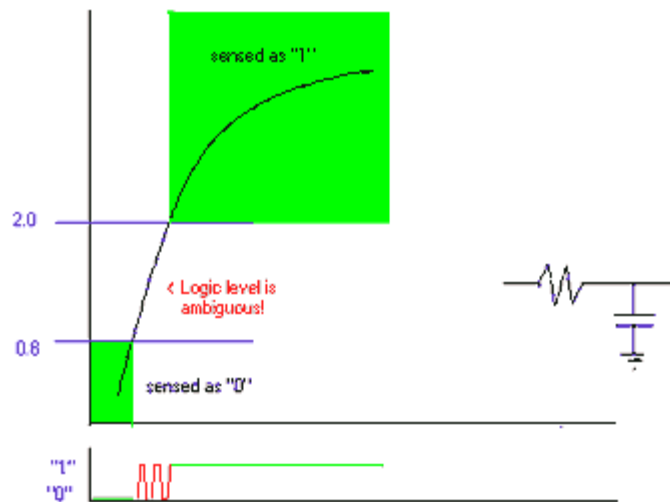
Input characteristics

The defined input logic levels for TTL are "0" input = 0.0–0.8V, "1" input = 2.0–5.0V. In the original version of TTL,

whatever was applying a '0' to the input had to be able to sink (accept) 1.6mA; when applying a '1', it had to be able to source (provide) 400mA. Later implementations of TTL-type logic substantially reduced this current requirement; today, input current on some devices is only $\pm 10\text{mA}$. Keithley's boards were designed at various times during the evolution of logic families, so the input current for a given board will fall somewhere in this range. As a practical matter, the actual input current is usually not too important because, unless many inputs are connected to a single output, exceeding the output's ability to sink or source current is unlikely.

Inputs that change slowly

As mentioned earlier, a problem can occur with digital inputs if the signal of interest is a slowly changing voltage, such as a voltage from a capacitor being charged, as shown in **Figure 1**.

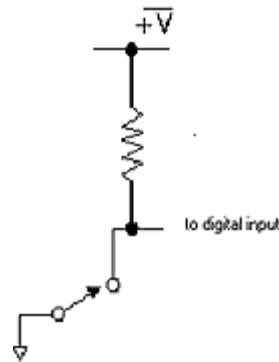


Due to the slow rate of voltage change, the signal remains in the "gray zone" too long, so it may be interpreted as shifting ON and OFF instead of changing smoothly from one logic level to the other. If one is attempting to count the input or take some action when it changes state, this can be disastrous. There are several ways to deal with this:

1. If the slow response was not intentional, correct the input signal so it changes more quickly.
2. If it's necessary to detect a specific voltage level, read the signal as an analog input or add an analog comparator (limit alarm) module.
3. If the exact switching level is not important but clean switching is required from slow or noisy signals, use a digital input with hysteresis, which will sense the change of logic state at different voltage levels, depending upon whether the signal is rising or falling. The resulting state is more stable because a greater change of the input is needed to force it to change back—it is much less likely to oscillate due to dithering near the switching point. Among Keithley boards, the KPCI-PIO24, KPCI-PIO96 and KPCI-3160 provide 0.2V of hysteresis. If this level of stability is insufficient for the application, systems with 5V TTL inputs can add a 74LS14 buffer before the input—this will typically provide 0.8V of hysteresis. (Note: Using this chip will also invert the applied signal.)
4. Much the same effect as hysteresis can be achieved in software by ignoring state changes that occur too quickly after the previous change. In some cases, this kind of pulse-width discrimination can be effective, depending on the input signal characteristics and the cost of the additional programming required vs. the cost of a hardware solution.

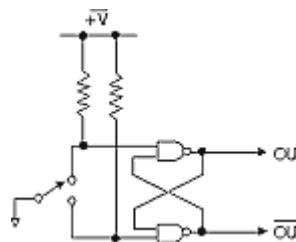
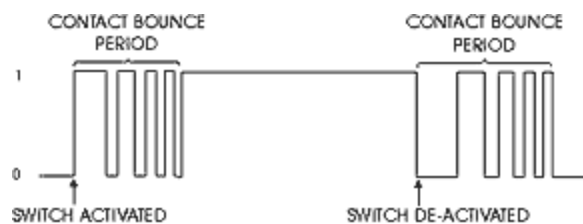
Inputs from contact closures (switches, relay outputs, etc.)

Switch contacts can be interfaced simply by adding a pull-up resistor (for example, 10Kohm). Some boards/chips already include this pull-up resistance or make it simple to install one: check the User's Guide for the board in question.



Despite the advantages they offer, there are two special problems associated with mechanical switches:

1. When the switch snaps shut, there will be a short period (typically one to five milliseconds) when the contact surfaces bounce against each other. While this is too fast for the human eye to see, the logic input may read it as a burst of ON/OFF signals. Adding hysteresis as described previously will not help because the signal is making full-scale changes. Software solutions include slowing the scan rate (to avoid multiple readings during the bounce) or ignoring short-duration signals (simple algorithm: once the system detects the switch data has changed from the previous value, it waits 10msec. to read the stabilized result). The hardware solution of adding a flip-flop circuit to generate a clean signal is seldom used because the software fix is effective and inexpensive.



2. The contact material itself presents a second difficulty. Contacts designed for relatively high currents (>100mA) have large contact areas and arc-resistant materials. These may not perform reliably over the long term when switching the small amount of current needed by the pull-up resistor. That's because the limited signal energy cannot break through the surface contamination film that tends to build up on the contact. When possible, use switches designed for "dry circuits"—these typically have softer cross-point contacts, which will work well even with small voltages and currents. This is one case in which specifying a higher rating for the contact may actually provide less reliability.

Inputs with voltages outside the TTL signal levels

The input circuits on most Keithley cards have ICs with internal diodes that will clip signals greater than 5.5V or less than -0.5V. These provide adequate protection against occasional transients, but are not intended to dissipate significant amounts of energy. If the signal remains outside TTL limits, it will damage the board and/or the signal source. Use an interface designed

for non-TTL logic levels.

- 1 A few Keithley cards have been designed with interface circuitry for special voltage ranges:

| | |
|---------------------|---|
| PDISO-8 | "0" input = 0.0–0.8V, "1" input = 5–24V, DC or AC . This is the only Keithley card that can accept AC input voltages directly. (As a consequence of rectification and filtering, it also has the slowest response.) |
| PIO-32I/O, PIO-32IN | "0" input = 0.0–0.8V, "1" input = 3.5–28V (Additional resistance can be added to allow use at even higher '1' voltages.) |
| PIO-HV | "0" input = 0.0–2.0V, "1" input = [3.5–4.0]*–32V (* Low value of the input range depends upon which port is used.) |

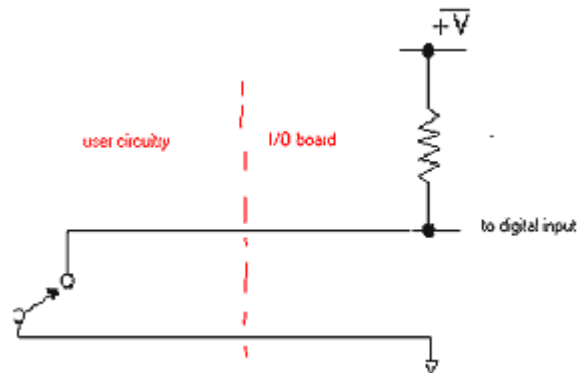
Be aware that, due to the additional circuitry, these inputs will take longer (0.33msec. to 10msec.) to respond to an applied signal—check the board's specifications for full details.

- 2 If these ranges don't suit the application's requirements or the application involves a mixture of input signals at varying levels, use a solid-state relay board, which will allow installing different signal conditioning modules for each input. This is also a prudent way to protect the digital I/O board (and the entire PC system) from damage if an input is accidentally shorted or misconnected. For more information on solid-state relay boards, refer to the discussion in the Accessories section.

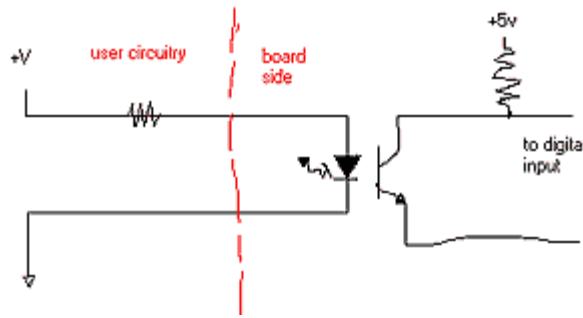
Inputs with electrical isolation

Even if the signal source supplies a TTL-level signal, it can cause damage to the hardware if it comes from a system operating at a different ground voltage. (Although this should not be a problem if all devices are plugged into the same power distribution, it can occur in some cases, such as in manufacturing environments, where the PC may be on one power system and the other equipment is fed from another.) While ground should be the same potential everywhere, don't count on it! If power will be coming from different sources, isolate the inputs with one of the following methods—

1. At the source, by using potential-free contacts: switch or relay closures. Use the +5V from the digital input board as the source for the pull-up resistor to the input—do not connect to the ground or +5V supply of the target equipment to maintain the isolation.



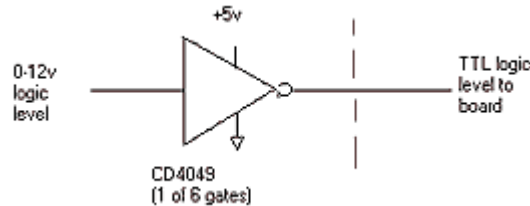
2. At the input, by using an isolated device: an optical coupler. In this case, the input side of the device must be powered from the remote hardware to maintain isolation. Keithley boards that provide optically isolated inputs include PDISO-8, PIO-32-IN, PIO-32-I/O, and PIO-HV and KPCI-PDISO8A and KPCI-PIO32IOA.



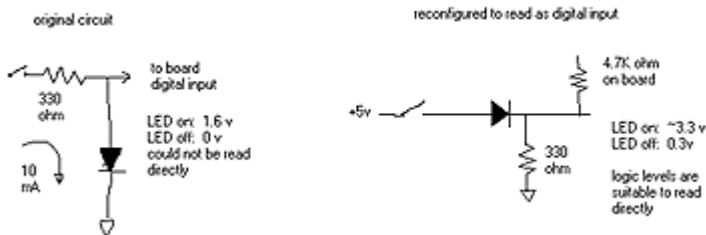
- At an isolated interface circuit: PB-24(-SM), or similar customer hardware. Refer to the Accessories section for options Keithley provides.

Input examples

- Example 1. 0-5VDC signal, TTL compatible levels. May input directly; consider if isolation necessary.
- Example 2. 0-12VDC signal, CMOS compatible levels. Cannot connect directly--use a CD4049 Hex Inverting Buffer (Non-Inverting = CD4050) or similar chip to convert the CMOS to TTL logic levels.



- Example 3. Sense whether an LED is ON in the customer's circuit (Figure 8). As originally wired, voltage levels across LED could not be read as logic inputs. Reconfiguring the circuit made it work as desired.



- Example 4. Generate an input pulse or interrupt when the AC power line voltage crosses zero. This can be done using a CA3059 chip to interface to the AC line. Rather than using the output pulse to trigger a triac, connect it to an optically isolated input line. Note: Exercise extreme caution in making input connections derived from power line sources. Keithley Instruments, Inc. cannot accept responsibility for circuit damage or personal injury caused by user connections.

Output characteristics

The defined input logic levels for TTL are "0" input = 0.0–0.8V, "1" input = 2.0–5.0V, so standard TTL outputs have about the same levels (generally with slightly tighter voltage specs—0.5V max. for '0', 2.4V min. for a '1'—to allow some voltage margin). However, in most cases, users do not use digital I/O board outputs to feed other TTL logic—the outputs are used to drive a variety of devices: indicator lamps, solenoid coils, solid-state relays, etc. Consequently, devices on Keithley digital I/O boards have a wide range of drive current capabilities. Almost all our hardware sinks more current than it sources, and most older boards have lower total drive current capabilities. But even these generalizations are not true in every case, so check the specifications carefully. In particular, note that some boards with multiple ports have different ratings depending on the port used—assign the outputs accordingly.

Types of outputs offered (Grouped by current capability. Refer to the board's data sheet for full specifications.)

TTL, sink currents less than 15mA

- CTM-05/A, -10
- DAS-8, -16, -80x, -120x, -140x, -160x, -170x, -180x
- KPCI-180xHC, -3101, -3102, -3103, -3014, -3140
- KPCMCIA -AI, -AI/AO, -AO
- PIO-12, -INT, -SSR-12, -SSR-24, -SSR-48

TTL, sink currents greater than 15mA

- KPCI-3107, -3108, -3110, -3116, -3160
- KPCI-PIO24, -PIO96
- KPCMCIA -PIO
- PDMA-16, -32
- PIO-24, -96J

Open collector

- PIO-HV: outputs rated to 50V/25mA

Relay contact closure

- PDISO-8: 3A contact rated for 125VAC/28VDC
- PIO-32I/O, -32OUT: 0.75A contact rated for 60VDC resistive
- REL-16: 0.5A contact rated for 125VRMS resistive
- REL-16/W: 1A contact rated for 125VRMS resistive

(Note: Available cables and accessories are rated for 60V maximum; use with 125VAC requires user-provided cabling.)

How do I interpret the output specification?

Let's take a reasonably typical example: The application involves using the digital outputs on a DAS-1802 to control some devices. The data sheet says the output specifications are:

OUTPUT LOW: $V_{OL} = 0.5V$ max. @ $I_{OL} = 8mA$.

OUTPUT HIGH: $V_{OH} = 2.7V$ min. @ $I_{OH} = -400\mu A$.

This means that when the output is HIGH (ON), it's possible to source (provide) up to 400mA (= 0.4mA) of load current, at which level the output voltage could be at the minimum guaranteed level of 2.7V. However, when the output is LOW (OFF), it's possible to sink (accept) up to 8mA with the output voltage still guaranteed to be no more than 0.5V above ground. Thus, if the load requires more than 0.4mA, it must be configured to be active when the output is LOW. (See Case 1 of in the section titled "Output examples.")

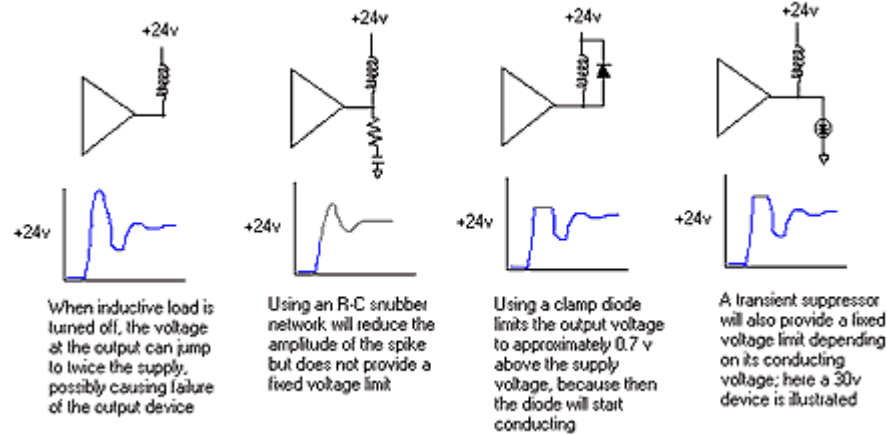
Boosting output current

If the relay or other output load requires more current or a higher voltage than the board can provide, it's possible to boost the drive current and/or voltage using the circuit shown in Figure 10. For drive-current requirements between 15mA and 100mA, select an NPN transistor rated for the required supply voltage with a collector current rating no higher than 0.5A. If higher current is needed, substitute a Darlington NPN transistor.

Protection of outputs

- *Inductive spikes.* The most common problem comes from devices with significant inductance -- solenoids, relays, motor windings, etc. -- because when the current flow stops, the collapsing magnetic field causes a large voltage spike. This can exceed the breakdown voltage of an unprotected output, causing permanent damage. There are several ways to deal with this:

- Snubber networks. A resistor-capacitor network across the output will absorb much of the spike energy, but has no inherent voltage limiting. This is adequate for relay contacts but insufficient to protect solid-state outputs. (Typical values: $R = 100 \text{ ohm}$, $C = I^2/10\mu\text{F}$, where $I = \text{load current}$)
- Clamp diodes. A clamp diode will limit the voltage at the output pin to no more than a diode-drop (typically 0.7V) greater than the output supply. The diode should be rated for at least the voltage and current of the output. (Note that the PIO-HV includes these diodes within its driver chip; in this case, connect pin 19 to the output supply.) Alternatively, a zener diode in parallel with the output (Figure 11c) provides clamping at defined positive (V_z) and negative ($V_{\text{diode drop}}$) voltage levels, but at a higher cost.
- MOV protectors. Metal-oxide-varistors have a characteristic curve similar to a bipolar zener diode, clamping at a defined voltage for both positive and negative spikes. This makes them ideal for protecting contacts switching AC voltage, so they are most commonly found in values suitable for power line voltages, e.g., 140V or 280V breakdown. However, they are equally effective for protecting solid-state DC outputs if they are available in a compatible rating. Look for transient voltage suppressors at Digi-Key and similar suppliers.



- *In-rush currents.* Some devices (such as incandescent light bulbs) may have initial currents many times higher than their steady-state operating requirement. This can cause contacts to weld or solid-state outputs to fail. A suitable small resistor in series with the load (sized to drop about 10% of the voltage at the steady-state current, i.e., $R = V/[10 \cdot I]$) can limit peak current with little effect on normal operation. In the case of light bulbs, this will also extend the life of the bulb.
- *Failure currents.* Prudent design also requires considering ways to protect system hardware from damage when devices fail, particularly for equipment that will be used over a long period of time. As with in-rush currents, small series resistors can protect against excessive current if devices become short-circuited. If nothing else, ensure that the output device supply is fuse protected so that fault currents will be interrupted quickly.

Default states

With ports that can be configured in software, the system design must take into consideration the default state that these ports will assume for the time period between power-up and the point when the program has initialized them. In general, ports will initialize as inputs, but with some chips, this can still provide enough bias current to turn on output devices unintentionally. Product advisory PA-612 (Appendix 1) discusses this issue in reference to the PIO-12, but it also applies to other hardware using the 82C55 chip.

Special output features (not applicable to all boards)

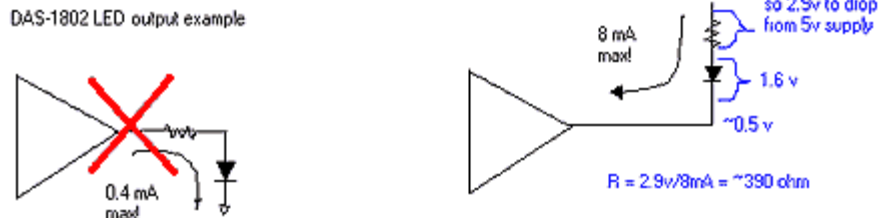
- *Read back.* Depending on the hardware implementation, most output ports can also be read as inputs, allowing the user to read the present state and modify it. This is particularly useful if different tasks are modifying bits on the same port— the other outputs can be preserved by reading the port and modifying only a portion of the bits. Note that this may mean reading the actual state of the output lines or the last value latched into the output register, depending on how the hardware is designed—normally there is no difference, unless the latched value is being overridden by an applied input. However, when using Keithley's DriverLINX software, this capability is emulated in the driver, which returns the last value output when the port is read.
- *Strobe.* Some boards (for example, the DAS-170x/180x Series) provide a strobe to allow latching of output data into an attached device every time the output port is written. The strobe signal, if any, is described in the User's Guide for the product.
- *High speed paced transfers.* The KPCI-3101/2/3/4 boards allow up to 3MHz digital I/O when used through the A/D

subsystem. A/D channels cannot be included in the scan to maintain this fast rate.

- Limited total output current. When using the KPCI-3160, be aware that because it only has two ground return pins for all 96 potential outputs, the maximum allowed total output current is 2.0A. (Theoretically, the drivers could sink a total of $96 \times 64\text{mA} = 6.1\text{A}$; this restriction limits them to an average per driver of 20.8mA.) Individual drivers can still sink up to the maximum current as long as the total load current does not exceed 2.0A. (As a practical matter, the PCI slot or computer power supply may also limit how much current is available.)

Output examples

- Case 1. A user wants to connect an LED indicator (with specs $V_f = 1.6\text{V}$ @ $I_f = 20\text{mA}$) to a DAS-1802 digital output. Perhaps the user's initial thought is to give it voltage when the output is turned on—but this output can't source nearly enough current: only 0.4mA, with the output dropping to 2.7V. The user will have to drive the LED by connecting the anode to +5V through a dropping resistor to limit current and pull the cathode down. When turned on, the output will be about 0.5V, leaving 4.5V across the LED and resistor; the resistor thus must drop $(4.5 - 1.6) = 2.9\text{V}$. For a 20mA current, the dropping resistor should be $2.9\text{V}/20\text{mA} = 145\text{ ohms}$. But look at the board's spec again: the output is only rated for a maximum of 8mA! The user will have to settle for less current and a dimmer LED—recalculate the dropping resistor as $2.9\text{V}/0.008\text{A} = 362.5\text{ ohms}$, so a 390 ohms standard value will work. Comments: (1) Note that the LED will come ON when the output is LOW=OFF, so the program logic needs to control the output accordingly. (2) An LED is usable even with a reduced current, but other types of loads aren't! For example, if the user had connected a reed relay with a 5V/20mA coil, it might work for a short time or not at all. For reliable performance, it's critical to stay within the specs. (**figure 11**)



- Case 2. The user wants to connect a relay with an 120VAC coil. Two options are available:
 - Use a board with relay contact closure outputs to switch 120VAC to the device directly, or
 - Use a solid-state relay module controlled by a digital output bit (refer to the section titled "Accessories.")
- Case 3. The user wants to connect a relay with a 24VDC coil. Several possibilities are open:
 - Use a board with relay contact closure outputs to switch 24 VDC to the device, or
 - Use a DC relay module controlled by a digital output bit (refer to the section titled "Accessories."), or
 - Drive it directly from an open-collector output such as those on the PIO-HV.

(If using the PIO-HV, note pin 19 should be tied to the 24 VDC relay supply, and the load current for the relay coil cannot exceed 24mA.)

Accessories

Two sets of input/output accessories are available, which can be grouped by interface type:

Compatible with industry standard PB-24 series:

Boards PIO-SSR-24, PIO-SSR-48, PIO-SSR-120 can be used with:

| | |
|---------|---|
| PB-24 | Holds up to 24 standard or small size solid-state relay devices |
| PB-24SM | Holds up to 24 small size solid-state relay devices |

Compatible with Keithley standard cable pinout:

Other Keithley boards, refer to Table 1.

| | |
|---------|--|
| ERA-01 | Has 8 electromechanical relays |
| ERB-24 | Has 24 electromechanical relays, includes built-in power supply |
| SRA-01 | Holds up to 8 standard or small size solid-state relay devices [4 O; 4 I or O] |
| SSIO-24 | Holds up to 24 small size solid-state relay devices [I or O by 8-8-4-4 groups] |

The modules used by both these families are:

| Standard Size | Small Size | Direction | Range |
|---------------|------------|-----------|--------------------|
| DC-05 | SM-IDC5 | Input | 3.3-32 VDC |
| AC-05 | SM-IAC5 | Input | 90-140 VDC/VAC |
| AC-5A | SM-IAC5A | Input | 180-280 VDC/VAC |
| ODC-05 | SM-ODC5 | Output | 3-60VDC @ 3A |
| ODC-05A | SM-ODC5A | Output | 3-250 VDC @ 1A |
| OAC-05 | SM-OAC5 | Output | 12-140 Vrms @ 3.5A |
| OAC-5A | SM-OAC5A | Output | 24-280 Vrms @ 3.5A |

Each of these modules buffers one bit of input or output, so a typical installation will require multiple modules. The ports are generally organized as groups of eight bits, but the SRA-01 allows splitting module assignments to have up to four inputs mixed with the outputs. Note: Given that the smaller "SM" Series modules are pin-compatible with the standard size and the pricing of both types is similar, users with mixed requirements may prefer to buy just the SM series.

How to choose which accessory to use

The PB-24(SM) accessories are usable only with Keithley's PIO-SSR-24/48/120 boards. All other Keithley board types can use the second group of accessory boards, depending on the requirement:

| | |
|-----------------------------------|---|
| Relay contact closure output: | Use ERA-01 or ERB-24, depending on number of relays required. |
| Solid-state input/output modules: | Use SRA-01 or SSIO-24, depending on number of modules required. |

Note that an SSIO-24 is the functional equivalent of a PB-24(SM), but has a different pin-out due to conformance with Keithley's cabling. Going beyond what the PB-24(SM) offers, the Keithley accessories provide both relay and solid-state options, in sizes for just one 8-bit port or for all 24 bits of a three-port grouping.

Table 1. Boards with Keithley standard digital input-output pinout

| Board type | Bus type | Applicable cable | Applicable accessory | Notes |
|------------|----------|-------------------|----------------------|------------------------------|
| CTM-05/A | ISA | C-1800 | ERA-01, SRA-01 | No access to counter signals |
| CTM-10 | ISA | CAB-4037 + C-1800 | ERA-01, SRA-01 | No access to counter signals |
| DAS-120x | ISA | CAB-4037 + C-1800 | ERB-24, SSIO-24* | From auxiliary connector |
| DAS-140x | ISA | | None recommended** | |
| DAS-16 | ISA | | None recommended** | |
| DAS-160x | ISA | CAB-4037 + C-1800 | ERB-24, SSIO-24* | From auxiliary connector |

| | | | | |
|----------------------|--------|---|---|--|
| DAS-170x | ISA | | None recommended** | |
| DAS-180x (except HC) | ISA | | None recommended** | |
| DAS-8 | ISA | | None | |
| DAS-80x | ISA | | None recommended** | |
| KPCI-180xHC | PCI | | None | |
| KPCI-3101/2/3/4 | PCI | | None | |
| KPCI-3107/8 | PCI | CAB-1284CC + STA-3108D1 + : CACC-2000 -> ADP-5037 -> CAB-SSR -> | (select one) ERB-24, SSIO-24 ERA-01, SRA-01 PB-24, PB-24SM | |
| KPCI-3110 | PCI | | None | |
| KPCI-3116 | PCI | | None | |
| KPCI-3140 | PCI | | None | |
| KPCI-3160 | PCI | CAB-1800/S + CONN-3160D1 + : CACC-2000 -> ADP-5037 -> | (up to 4 max) ERB-24, SSIO-24 ERA-01, SRA-01 | |
| KPCI-PIO24 | PCI | C-1800 | ERB-24, SSIO-24* | |
| KPCI-PIO96 | PCI | CACC-2000 -> ADP-5037 -> | ERB-24, SSIO-24 ERA-01, SRA-01 (4 total) | |
| KPCMCIA -AI | PCMCIA | | None | |
| KPCMCIA -AI/AO | PCMCIA | | None | |
| KPCMCIA -AO | PCMCIA | | None | |
| KPCMCIA -PIO24 | PCMCIA | (KCAB-PIO) | ERB-24, SSIO-24* | |
| PDISO-8 | ISA | | None | |
| PDMA-16 | ISA | | None | |
| PDMA-32 | ISA | | None | |
| PIO-12 | ISA | C-1800 | ERB-24, SSIO-24* | |
| PIO-24 | ISA | C-1800 | ERB-24, SSIO-24* | |
| PIO-32I/O | ISA | | None | |
| PIO-32IN | ISA | | None | |
| PIO-32OUT | ISA | | None | |
| PIO-96J | ISA | CACC-2000 -> ADP-5037 -> | ERB-24, SSIO-24 ERA-01, SRA-01 (up to 4 max) | |
| PIO-HV | ISA | | None | |

| | | | | |
|-------------|-----|---------|------------------|--|
| PIO-INT | ISA | C-1800 | ERB-24, SSIO-24* | |
| PIO-SSR-120 | ISA | CAB-SSR | PB-24(SM) | |
| PIO-SSR-24 | ISA | CAB-SSR | PB-24(SM) | |
| PIO-SSR-48 | ISA | CAB-SSR | PB-24(SM) | |
| REL-16 | ISA | | None | |
| REL-16/W | ISA | | None | |

* Any board that can interface with the ERB-24 and SSIO-24 can also use the ERA-01 and SRA-01, but those provide no means of connecting to the remaining unused bits.

** The SRA-01 with C-1800 cable can use the digital I/O pins available on this card, but provides no access to the remaining analog signals and controls. Not a recommended configuration.

What if it's necessary to mix relay and solid-state outputs?

Sorry, we don't have an accessory card that combines both—but it's easy to drive an external relay with one of the solid-state modules.

Why would a user need the solid-state modules?

Most of Keithley's digital I/O boards are limited to 0-5V TTL signals. (Exceptions are listed in the Input and Output sections.) The solid-state modules allow those TTL signals to be used to control outputs up to 250VDC or 280Vrms, and to read inputs up to 280 VDC/VAC as a TTL signal. These modules have two further advantages: they keep these elevated voltages physically away from the computer board, and they also provide 4000V of electrical isolation, meaning that they can be used safely with signals referenced to a different power ground.

What about grounding?

Grounding is less critical for digital signals because the large voltage difference between logic levels reduces the possibility of misinterpreting a signal. Be sure, however, that all digital grounds (except for isolated inputs) and the ground of any auxiliary power supplies for accessories are connected to the digital I/O board's ground reference.

Tips on using the accessories:

- When the computer's +5 VDC is used for accessory modules, do not draw more than 1A to avoid overloading the supply. If more current is needed, connect an external supply.
- Some boards (SSIO-24, PB24) use small 'picofuses' to protect the power supply and I/O connections. (These may be difficult to recognize as fuses because they look similar to 1/4 watt axial lead resistors.) Check the signal fuse for continuity (power off!) if a given module doesn't get its signal to/from the terminal strip. Check the power fuse if no modules are working. If the power fuse was blown and an external supply is being used, be sure it is connected with the correct polarity.
- The indicator LEDs on solid-state module boards will not light unless a module is plugged in for that bit.
- The LED is lit if the associated input or output is active. However, the logic is negative true: an energized input will read as a '0' on the port, and to energize an output, a '0' must be written to that bit. Allow for this in the program logic!
- If the I/O is not functioning as expected, check to be sure:
 1. that ports and modules are assigned consistently: an input port will not work with output modules stuffed in the SSIO-24/PB24. On SRA-01 and SSIO-24, it is necessary that both the software port direction and the hardware accessory switches be set according to whether signals are inputs or outputs.
 2. that +5V is being delivered at the modules: an overloaded supply may not deliver the proper voltage.

Interrupts

Interrupts are also digital inputs. Rather than being read on a port, they are connected to Interrupt Request (IRQ) lines, which (when suitably enabled by a driver) cause the processor to interrupt whatever it is doing and respond immediately. This can

allow the processor to handle tasks more efficiently, because it can react just when needed instead of having to wait for something to happen. A computer system has only a limited number of IRQ lines: 16, numbered 0 to 15. Many of these are preassigned to system devices (keyboard, serial ports, printer port, etc.), so finding a suitable available IRQ line can be difficult.

What does "a suitable line" mean?

A suitable line is one that the board hardware can access. Some ISA boards use just the 8-bit bus connector and can only connect to IRQ 2 to IRQ 7, so having IRQ 10 available wouldn't be useful. Board jumpers may also restrict which interrupt lines can be used, so check the User's Guide. The PCI-bus boards can usually share IRQ lines (most ISA boards/systems cannot), which allows adding hardware even if all interrupt lines are being used. However, be aware that sharing an interrupt may result in both throughput limitations and timing issues if the board is being used at close to its maximum capacity.

How can I find out which interrupt lines are available on my PC?

In Windows 95/98, select Start, Settings, Control Panel, System, Device Manager, Computer, and Properties: this page displays the IRQ assignments of which the system is aware. Selecting other buttons will display information about I/O addresses, DMA channel assignments and memory use. A similar display of system resource assignments is available on the Windows NT Diagnostics program "Resources" tab.

How can I make an IRQ line available if I need one?

This situation could occur if a board requires IRQ3, but that is already assigned to the modem on the system. If the modem is an ISA card and can use an available higher-numbered line, reassign the IRQs so the needs of both cards are satisfied. If the modem is a PCI-bus or PCMCIA card and the system is running on Windows 95/98, IRQ3 can be reserved for the ISA card so that the system will reassign the modem: do so on the same page that displays the IRQ assignments (as discussed in the previous answer).

Which boards have hardware input lines to generate an interrupt?

The following table lists only the boards with dedicated interrupt inputs. If a buffer accessory can be used with the interrupt signal, that is also indicated in the table. (Be aware that this will add the response time of the solid-state module.) Note that many boards not listed here can generate a message through their driver if a specified condition occurs; this type of software interrupt is preferred in Windows environments and can also be performed by drivers for the boards listed.

| Board type | DIO Interrupt? | Buffering accessory? | Notes |
|-----------------|-------------------|----------------------------|--|
| CTM-05/A | Input, Enable/ | SRA-01 for Interrupt Input | SRA gives no access to counter signals |
| CTM-10 | Input, Enable/ | SRA-01 for Interrupt Input | SRA gives no access to counter signals |
| DAS-8 | Input | None | |
| KPCI-3140 | Port D bit change | None | |
| KPCI-PIO24 | Input, Enable/ | SRA-01 for Interrupt Input | SRA does not buffer all I/O lines |
| KPCMCIA - PIO24 | Input | | |
| PDMA-16 | Interrupt | None | |
| PDMA-32 | Interrupt | None | |
| PIO-12 | Input, Enable/ | SRA-01 for Interrupt Input | SRA does not buffer all I/O lines |

| | | | |
|---------|--------------------------------------|----------------------------|-----------------------------------|
| PIO-24 | Input, Enable/ | SRA-01 for Interrupt Input | SRA does not buffer all I/O lines |
| PIO-INT | Port A & B bit pattern or bit change | SSIO-24 | |

How do I cause something to happen in my program when an interrupt occurs?

In a Windows environment, the driver needs to send a message so that the program knows the interrupt has taken place and can take action. The implementation varies according to the driver software.

Structure of the ports

The Beginning: the 8255 Programmable Peripheral Interface (PPI) integrated circuit

The de facto standard for digital I/O ports dates back to the mid-'70s, when Intel developed a series of interface chips for its 8080 microprocessors, the distant ancestors of today's Pentium chips. This chip, originally produced via an NMOS process, is still manufactured in a CMOS version and is second-sourced by many manufacturers. For software compatibility, many boards designed with other chips (such as Keithley's PIO-SSR-24, which uses discrete chips to obtain higher drive currents) have a hardware design that emulates the 8255's Mode 0. (In this case, emulate means that the hardware has the same arrangement of registers and responds identically to commands so that the software can utilize the ports just as though it were accessing an 8255 chip.) Its register arrangement and command set is common to so many boards, so an understanding of how this chip works is a good place to start in this review of board architecture.

How are the I/O lines organized in an 8255?

- The 8255 PPI provides 24 digital I/O lines, which make up the following three ports:
- PORT A (abbreviated PA) consists of one 8-bit data output latch/buffer and one 8-bit data input latch.
- PORT B (PB) consists of one 8-bit data output latch/buffer and one 8-bit data input buffer.
- PORT C (PC) consists of one 8-bit data output latch/buffer and one 8-bit data input buffer. This port can also be used as two 4-bit ports: PC Upper (PC4 to PC7) and PC Lower (PC0 to PC3).

How are the ports configured?

8255 PPI Address Map:

| Address | Function | Type |
|-----------------|--------------|------------|
| Base Address +0 | Port A (PA) | Read/Write |
| Base Address +1 | Port B (PB) | Read/Write |
| Base Address +2 | Port C (PC) | Read/Write |
| Base Address +3 | Control Word | Write only |

Note: This type of table is also known as a register map.

After the ports and any associated interface hardware are properly configured, the PC can output data by writing to the specified address or input data by reading from that address. (Reading an output port will return the value of the data previously written there.) Each port or half-port is configured as an input or an output according to the value of the control word in the PPI control register. The control word is a write-only register at location Base Address +3. The control word also specifies one of the following three operating modes:

Mode 0 - Basic I/O; all ports are I/O ports. [This is the mode most emulations provide.]

Mode 1 - Strobed I/O; part of Port C controls data transfer.

Mode 2 - Bi-directional I/O on Port A only; part of Port C controls data transfer.

IMPORTANT NOTES:

- A power-up or reset automatically configures all ports as inputs. This configuration remains intact until changed by a write to the control register. The initialization section of the program must set the port configuration.
- A characteristic of the 8255 PPI is that all port registers are cleared by a write to the control register. Consequently, if the program repeatedly changes the port configuration, it's essential to make provisions for restoring data to cleared ports. (Some drivers provide this capability transparently to the user.)
- These characteristics apply to boards implemented with the 8255 PPI chip. Other boards, even those that emulate the 8255, may have different behavior. (For example, the PDISO-8 will clear the output port and de-energize all relays at power up, but not for a soft boot of the computer [pressing Ctrl + Alt +Delete].) Check the manual for the particular product to verify power-up and reset actions if this is critical in the application.

Port Configuration Bits in Control Word

| Bit # | Bit Setting | Configuration |
|---------|----------------------|--|
| 7 | 0 1 | Bit set/reset mode for PC0 to PC7 Mode set active for PC0 to PC7 |
| 6 and 5 | 00 01 10 1x | Mode 0 for PA4 to PA7 and PC4 to PC7 Mode 1 for PA4 to PA7 and PC4 to PC7 Mode 2 for PA4 to PA7 and PC4 to PC7 Mode 2 for PA4 to PA7 and PC4 to PC7 |
| 4 | 0 1 | PA output PA input |
| 3 | 0 1 | PC4 to PC7 output PC4 to PC7 input |
| 2 | 0 1 | PB0 to PB3 and PC0 to PC3 output PB0 to PB3 and PC0 to PC3 input |
| 1 | 0 1 | PB output PB input |
| 0 | 0 1 | PC0 to PC3 output PC0 to PC3 input |

If only Mode 0 is being used, the following table provides values of the control word for various port settings:

PIO Control Word, Mode 0

| Control Word Value | Port A direction | Port B direction | Port C Lower (bits 0 to 3) direction | Port C Upper (bits 4 to 7) direction |
|--------------------|------------------|------------------|--------------------------------------|--------------------------------------|
| 80h | Out | Out | Out | Out |
| 81h | Out | Out | In | Out |
| 82h | Out | In | Out | Out |
| 83h | Out | In | In | Out |
| | | | | |

| | | | | |
|-----|-----|-----|-----|-----|
| 88h | Out | Out | Out | In |
| 89h | Out | Out | In | In |
| 8Ah | Out | In | Out | In |
| 8Bh | Out | In | In | In |
| 90h | In | Out | Out | Out |
| 91h | In | Out | In | Out |
| 92h | In | In | Out | Out |
| 93h | In | In | In | Out |
| 98h | In | Out | Out | In |
| 99h | In | Out | In | In |
| 9Ah | In | In | Out | In |
| 9Bh | In | In | In | In |

Register maps for Keithley boards

The ISA digital I/O boards are the oldest and generally the simplest designs; most either use or emulate the 8255 PPI chip. As such, their register maps (which can be found in the appropriate User's Guides) generally follow that of the 8255 as described previously. Some ports may be omitted depending on the number of I/O lines, and the control register itself will not exist if the direction of lines cannot be changed.

Boards with incidental digital I/O, such as the analog and counter cards, have a limited number of digital I/O lines, which are usually fixed in direction and located at a specified port relative to the base address. As such, they do not require any special configuration.

The next simplest case is the PCMCIA cards. These have similar register maps to the ISA cards, with one twist: the system automatically selects the base address when the card is inserted. To use direct register access, the user will first need to examine the Windows (95/98) Device Manager entry to determine the assigned addresses. Note that this can change if the card is removed and reinserted later, causing problems if the program is dependent on a specific address! (The DriverLINX driver software avoids this problem by getting the current addressing information from the system.)

The PCI cards present two difficulties: they are also Plug-and-Play devices (with addresses assigned by the system as for the PCMCIA cards), and most of them are memory-mapped (they have addresses at memory locations instead of using the I/O address space). The only exception is the KPCI-3160, which is register-mapped to make it easier to use. (In this case, the user needs only look up which base address the system assigned, then—under Windows 95/98 only—use register input and output statements. Windows NT does not permit direct user access to I/O addresses.) The other PCI digital I/O boards are best used with driver software like DriverLINX, which can convert the I/O requests to the equivalent memory locations. The User's Guide does contain information about programming directly using the memory map, but Keithley does not recommend or support such efforts. However, control and data registers can be manipulated in the following special situations:

- A user who wishes to program in an ActiveX hosting language other than Visual C++, Visual Basic, or Delphi. In this situation, it may be necessary to use the "Direct I/O ActiveX Automation Object" that comes with DriverLINX. The Direct I/O ActiveX Automation Object allows the user to set the control and data registers directly and bypass the DriverLINX API. This interface also permits splitting Port C into a mix of 4 bits in/4 bits out, which the DriverLINX interface does not provide.
- A user who wishes to reuse an existing program that makes port I/O calls to an ISA-bus digital I/O board, such as the PIO-96 or PIO-24. In this situation, it's possible to reuse the existing program with the KPCI board, in Windows 95/98 only, via the Hardware I/O Emulation driver included on the DriverLINX CD. The Hardware I/O Emulation driver traps the port I/O calls and applies them properly to the KPCI-PIO96 board. Refer to the DriverLINX manual for more information.
- An advanced user who needs to use the KPCI board with an operating system other than Microsoft Windows 95/98 or

Windows NT 4.0 or greater. In this situation, the user must write a new driver, bypassing DriverLINX entirely. This task requires an in-depth knowledge both of the AMCC S5933 PCI Bus Controller and the development operating system.

Direct port control for troubleshooting

Testing the PDISO-8 or other boards with fixed I/O addresses does not require any special hardware or software. An ohmmeter can be used to check operation of the relay outputs and a low voltage power supply can be used to check the isolated inputs. To exercise relay outputs or read the input ports, use the DOS DEBUG program under Win 95/98. Here's an example using DEBUG for a PDISO-8 at a base address set to &H300:

- 300 FF 'will turn all relays on
- 300 0 'will turn all relays off
- 300 08 'will turn relay 4 on etc.
- 301 'will read input port (in hex)