Published on *LabJack* (https://labjack.com)

2.8 - Digital I/O [U3 Datasheet]

Digital I/O Overview

The LabJack U3 has up to 20 digital I/O channels. 16 are available from the flexible I/O lines, and 4 dedicated digital I/O (CIO0-CIO3) are available on the DB15 connector. The first 4 lines, FIO0-FIO3, are unavailable on the U3-HV. Each digital line can be individually configured as input, output-high, or output-low. The digital I/O use 3.3 volt logic and are 5 volt tolerant.

The LabJackUD driver uses the following bit numbers to specify all the digital lines:

0-7 FIO0-FIO7 (0-3 unavailable on U3-HV)
8-15 EIO0-EIO7
16-19 CIO0-CIO3

The "F", "E", and "C" designators have little special meaning. They are just arbitrary letters used to designate different groupings of digital I/O. The generic designator DIOx is sometimes used to describe any digital I/O from 0 to 19, so for example an alternative name for EIO0 is DIO8.

The 8 FIO lines appear on the built-in screw-terminals, while the 8 EIO and 4 CIO lines appear only on the DB15 connector. See the DB15 Section of this User's Guide for more information.

Max Current & Overvoltage Protection

All the digital I/O include an internal series resistor that provides overvoltage/short-circuit protection. These series resistors also limit the ability of these lines to sink or source current. Refer to the specifications in Appendix A.

Tri-State I/O

All digital I/O on the U3 have 3 possible states: input, output-high, or output-low. Each bit of I/O can be configured individually. When configured as an input, a bit has a ~100 kΩ pull-up resistor to 3.3 volts (all digital I/O are 5 volt tolerant). When configured as output-high, a bit is connected to the internal 3.3 volt supply (through a series resistor). When configured as output-low, a bit is

connected to GND (through a series resistor).

When only lightly loaded, an input terminal will measure about 3.3 volts if measured with a DMM, and thus it can be tough to use a DMM to tell whether a line is set to input or output-high. A couple tips to tell the difference:

1. Look for a slight change where output-high measures a little higher. For example, a DMM might show 3.300 for input and 3.315 for output-high.

2. Add a load resistor. If you add a 100k from DIOx to GND, it should read about 1.6V for input and 3.3V for output-high.

5 Volt Output

The fact that the digital I/O are specified as 5-volt tolerant means that 5 volts can be connected to a digital input without problems (see the actual limits in the specifications in Appendix A). If 5 volts is needed from a digital output, consider the following solutions:

- Use the LJTick-DigitalOut5V to convert a pair of digital outputs to 5V logic.
- In some cases, an open-collector style output can be used to get a 5V signal. To get a low set the line to **output-low**, and to get a high set the line to **input** (... note that this does not work with timer outputs, e.g. PWM, as they toggle the line between output-low and output-high). When the line is set to input, the voltage on the line is determined by a pull-up resistor. The U3 has an internal ~100k resistor to 3.3V, but an external resistor can be added to a different voltage. Whether this will work depends on how much current the load is going to draw and what the required logic thresholds are. Say for example a 10k resistor is added from EIO0 to VS. EIO0 has an internal 100k pull-up to 3.3 volts and a series output resistance of about 180 ohms. Assume the load draws just a few microamps or less and thus is negligible. When EIO0 is set to input, there will be 100k to 3.3 volts in parallel with 10k to 5 volts, and thus the line will sit at about 4.85 volts. When the line is set to output-low, there will be 180 ohms in series with the 10k, so the line will be pulled down to about 0.1 volts.
- A surefire way to get 5 volts from a digital output is to add a simple logic buffer IC that is powered by 5 volts and recognizes 3.3 volts as a high input. Consider the CD74ACT541E from TI (or the inverting CD74ACT540E). All that is needed is a few wires to bring VS, GND, and the signal from the LabJack to the chip. This chip can level shift up to eight 0/3.3 volt signals to 0/5 volt signals and provides high output drive current (+/-24 mA).
- Note that the 2 DAC channels on the U3 can be set to 5 volts, providing 2 output lines with such capability.

Boot-Up Defaults

The power-up condition of the digital I/O can be configured by the user with the "Config Defaults"

option in LJControlPanel. From the factory, all digital I/O are configured to power-up as inputs. Note that even if the power-up default for a line is changed to output-high or output-low, there is a delay of about 5 ms at power-up where all digital I/O are in the factory default condition. For more information see this forum topic.

Why Are My Digital I/O "High" at Boot-Up?

The implied question here is "why do my DIO boot up as output-high from the factory". The answer is that per the "Boot-Up Defaults" section above the factory default state for all DIO is input, but since inputs have a 100k internal pull-up they will read 3.3 volts if only lightly loaded. So likely you are seeing that the state of your DIO is input, not output-high. Also see the "Tri-State I/O" section above.

Making An Input Read Low By Default

If you want a floating digital input to read low, an external pull-down resistor can be added to overpower the internal 100k pull-up. 4.7k to 22k would be a typical range for this pull-down, with 10k being a solid choice for most applications.

Software Interface

The low-level Feedback function (Section 5.2.5) writes and reads all digital I/O. For information about using digital I/O under the Windows LabJackUD driver, see Section 4.3.5. See Section 3 for timing information.

Bit-Packed Integers

Many function parameters contain specific bits within a single integer parameter to write/read specific information. In particular, most digital I/O parameters contain the information for each bit of I/O in one integer, where each bit of I/O corresponds to the same bit in the parameter (e.g. the direction of FIO0 is set in bit 0 of parameter FIODir). For instance, in the low-level function ConfigU3, the parameter FIODirection is a single byte (8 bits) that writes/reads the power-up direction of each of the 8 FIO lines:

- if FIODirection is 0, all FIO lines are input,
- if FIODirection is 1 (2^0), FIO0 is output, FIO1-FIO7 are input,
- if FIODirection is 5 ($2^0 + 2^2$), FIO0 and FIO2 are output, all other FIO lines are input,
- if FIODirection is 255 ($2^0 + \ldots + 2^7$), FIO0-FIO7 are output.

2.8.1 - Typical Digital I/O Connections [U3 Datasheet]

2.8.1.1 - Input: Driven Signals [U3 Datasheet]

The most basic connection to a U3 digital input is a driven signal, often called push-pull. With a push-pull signal the source is typically providing a high voltage for logic high and zero volts for logic low. This signal is generally connected directly to the U3 digital input, considering the voltage specifications in Appendix A. If the signal is over 5 volts, it can still be connected with a series resistor. The digital inputs have protective devices that clamp the voltage at GND and VS, so the series resistor is used to limit the current through these protective devices. For instance, if a 24 volt signal is connected through a 22 kΩ series resistor, about 19 volts will be dropped across the resistor, resulting in a current of 0.9 mA, which is no problem for the U3. The series resistor should be 22 kΩ or less, to be strong enough to overpower the internal 100k pull-up and make sure the voltage on the I/O line when low is pulled below 0.8 volts.

The other possible consideration with the basic push-pull signal is the ground connection. If the signal is known to already have a common ground with the U3, then no additional ground connection is used. If the signal is known to not have a common ground with the U3, then the signal ground can simply be connected to U3 GND. If there is uncertainty about the relationship between signal ground and U3 ground (e.g. possible common ground through AC mains), then a ground connection with a ~10 Ω series resistor is generally recommended (see Section 2.6.3.4).

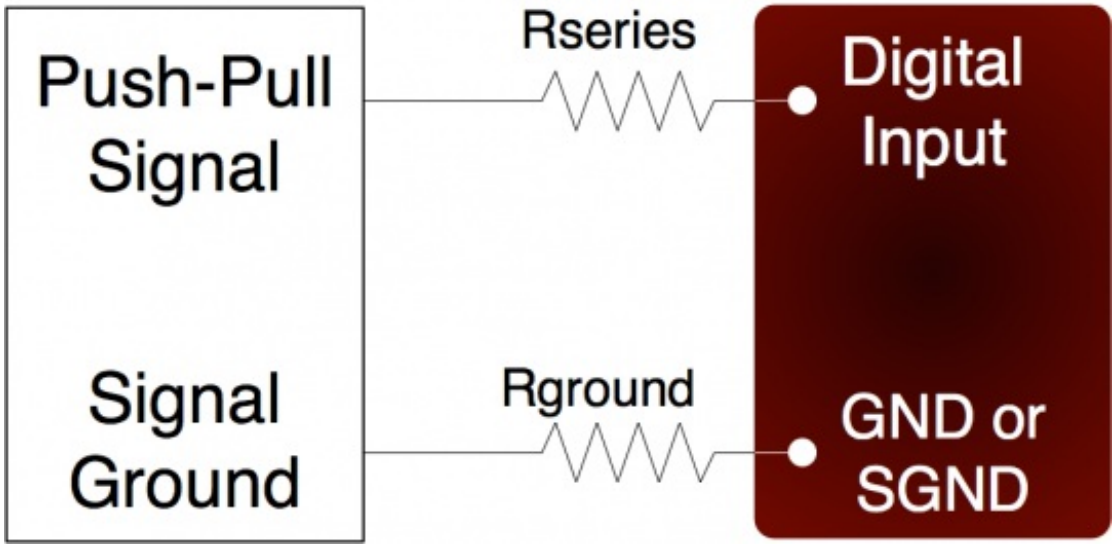

Figure 2.8.1.1-1. Driven Signal Connection To Digital Input

Figure 2.8.1.1-1 shows typical connections. Rground is typically 0-100 Ω. Rseries is typically 0 Ω (short-circuit) for 3.3/5 volt logic, or 22 kΩ (max) for high-voltage logic. Note that an individual ground connection is often not needed for every signal. Any signals powered by the same external supply, or otherwise referred to the same external ground, should share a single ground connection to the U3 if possible.

When dealing with a new sensor, a push-pull signal is often incorrectly assumed when in fact the sensor provides an open-collector signal as described next.

2.8.1.2 - Input: Open-Collector Signals [U3 Datasheet]

For details about open-collector, open-drain, NPN, or PNP connections, see theOpen-Collector Signals App Note.

2.8.1.3 - Input: Mechanical Switch Closure

[U3 Datasheet]

To detect whether a mechanical switch (dry contact) is open or closed, connect one side of the switch to U3 ground and the other side to a digital input. The behavior is very similar to the open-collector described above.

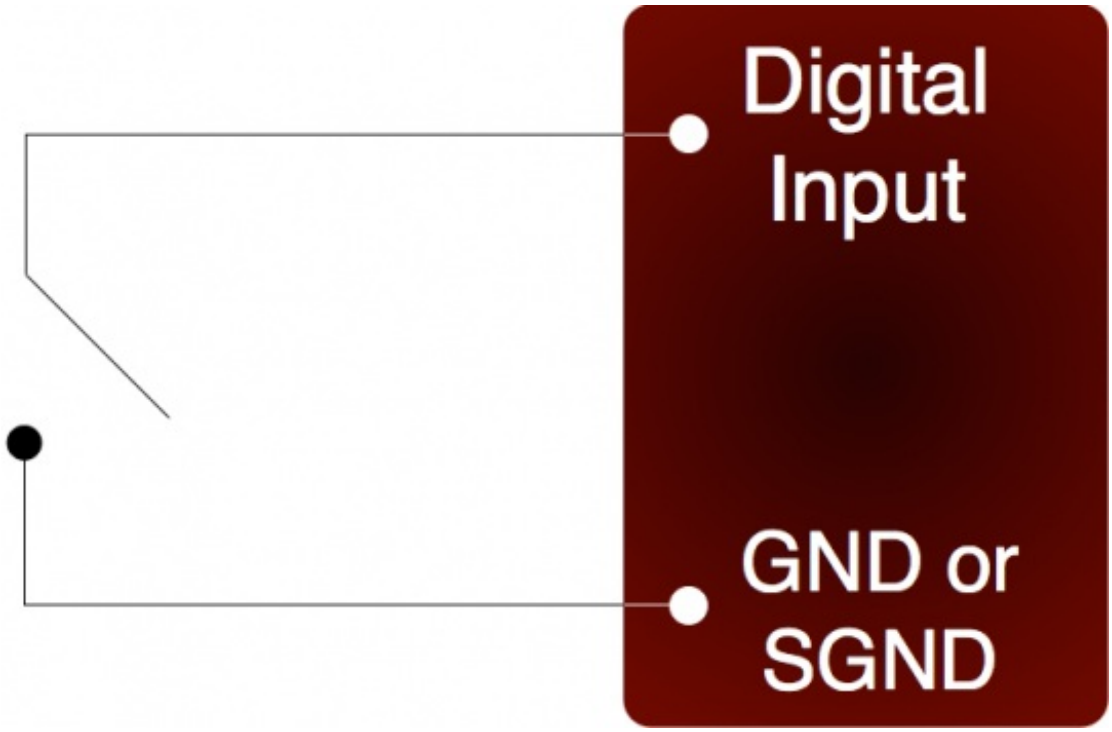

Figure 2.8.1.3-1. Basic Mechanical Switch Connection To Digital Input

When the switch is open, the internal 100 kΩ pull-up resistor will pull the digital input to about 3.3 volts (logic high). When the switch is closed, the ground connection will overpower the pull-up resistor and pull the digital input to 0 volts (logic low). Since the mechanical switch does not have any electrical connections, besides to the LabJack, it can safely be connected directly to GND, without using a series resistor or SGND.

When the mechanical switch is closed (and even perhaps when opened), it will bounce briefly and produce multiple electrical edges rather than a single high/low transition. For many basic digital input applications, this is not a problem as the software can simply poll the input a few times in succession to make sure the measured state is the steady state and not a bounce. For applications using timers or counters, however, this usually is a problem. The hardware counters, for instance, are very fast and will increment on all the bounces. Some solutions to this issue are:

- Software Debounce: If it is known that a real closure cannot occur more than once per some interval, then software can be used to limit the number of counts to that rate.
- Firmware Debounce: See Section 2.9.1 for information about timer mode 6.
- Active Hardware Debounce: Integrated circuits are available to debounce switch signals.

This is the most reliable hardware solution. See the MAX6816 (maxim-ic.com) or EDE2008 (elabinc.com).

- Passive Hardware Debounce: A combination of resistors and capacitors can be used to debounce a signal. This is not foolproof, but works fine in most applications.

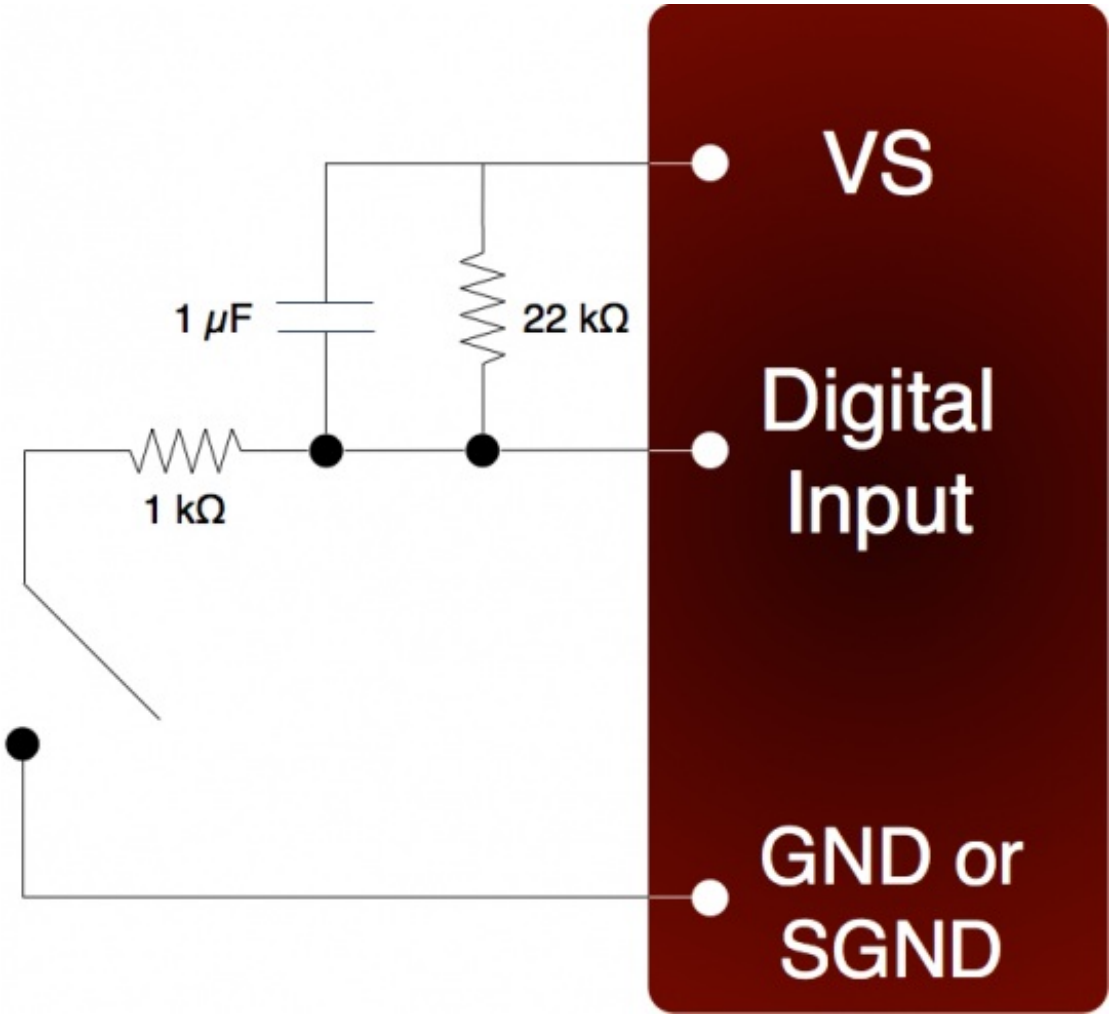

Figure 2.8-5. Passive Hardware Debounce

Figure 2.8-5 shows one possible configuration for passive hardware debounce. First, consider the case where the 1 kΩ resistor is replaced by a short circuit. When the switch closes it immediately charges the capacitor and the digital input sees logic low, but when the switch opens the capacitor slowly discharges through the 22 kΩ resistor with a time constant of 22 ms. By the time the capacitor has discharged enough for the digital input to see logic high, the mechanical bouncing is done. The main purpose of the 1 kΩ resistor is to limit the current surge when the switch is closed. 1 kΩ limits the maximum current to about 5 mA, but better results might be obtained with smaller resistor values.

2.8.1.4 - Output: Controlling Relays [U3 Datasheet]

All the digital I/O lines have series resistance that restricts the amount of current they can sink or source, but solid-state relays (SSRs) can usually be controlled directly by the digital I/O. The SSR is connected as shown in the following diagram, where VS (~5 volts) connects to the positive control input and the digital I/O line connects to the negative control input (sinking configuration).

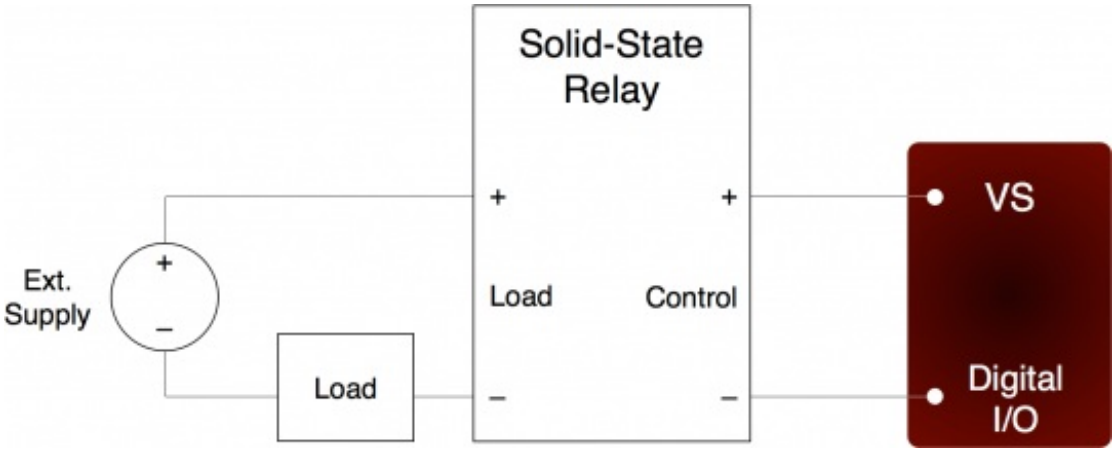

Figure 2.8.1.4-1. Relay Connections (Sinking Control, High-Side Load Switching)

When the digital line is set to output-low, control current flows and the relay turns on. When the digital line is set to input, control current does not flow and the relay turns off. When the digital line is set to output-high, some current flows, but whether the relay is on or off depends on the specifications of a particular relay. It is recommended to only use output-low and input.

For example, the Series 1 (D12/D24) or Series T (TD12/TD24) relays from Crydom specify a max turn-on of 3.0 volts, a min turn-off of 1.0 volts, and a nominal input impedance of 1500 Ω.

- When the digital line is set to output-low, it is the equivalent of a ground connection with 180 Ω (EIO/CIO) or 550 Ω (FIO) in series. When using an EIO/CIO line, the resulting voltage across the control inputs of the relay will be about 5*1500/(1500+180) = 4.5 volts (the other 0.5 volts is dropped across the internal resistance of the EIO/CIO line). With an FIO line the voltage across the inputs of the relay will be about 5*1500/(1500+550) = 3.7 volts (the other 1.3 volts are dropped across the internal resistance of the FIO line). Both of these are well above the 3.0 volt threshold for the relay, so it will turn on.
- When the digital line is set to input, it is the equivalent of a 3.3 volt connection with 100 kΩ in series. The resulting voltage across the control inputs of the relay will be close to zero, as virtually all of the 1.7 volt difference (between VS and 3.3) is dropped across the internal 100 kΩ resistance. This is well below the 1.0 volt threshold for the relay, so it will turn off.
- When the digital line is set to output-high, it is the equivalent of a 3.3 volt connection with 180 Ω (EIO/CIO) or 550 Ω (FIO) in series. When using an EIO/CIO line, the resulting voltage across the control inputs of the relay will be about 1.7*1500/(1500+180) = 1.5 volts. With an FIO line the voltage across the inputs of the relay will be about 1.7*1500/(1500+550) = 1.2 volts. Both of these in the 1.0-3.0 volt region that is not defined for these example relays, so the resulting state is unknown.

Note that sinking excessive current into digital outputs can cause noticeable shifts in analog input readings. For example, the FIO sinking configuration above sinks about 2.4 mA into the digital output to turn the SSR on, which could cause a shift of roughly 1 mV to analog input readings.

Mechanical relays require more control current than SSRs, and cannot be controlled directly by the digital I/O on the U3. To control higher currents with the digital I/O, some sort of buffer is used. Some options are a discrete transistor (e.g. 2N2222), a specific chip (e.g. ULN2003), or an op-amp.

Note that the U3 DACs can source enough current to control almost any SSR and even some mechanical relays, and thus can be a convenient way to control 1 or 2 relays. With the DACs you would typically use a sourcing configuration (DAC/GND) rather than sinking (VS/DAC).

The RB12 relay board is a useful accessory available from LabJack. This board connects to the DB15 connector on the U3 and accepts up to 12 industry standard I/O modules (designed for Opto22 G4 modules and similar).

Another accessory available from LabJack is the LJTick-RelayDriver. This is a two channel module that plugs into the U3 screw-terminals, and allows two digital lines to each hold off up to 50 volts and sink up to 200 mA. This allows control of virtually any solid-state or mechanical relay.
