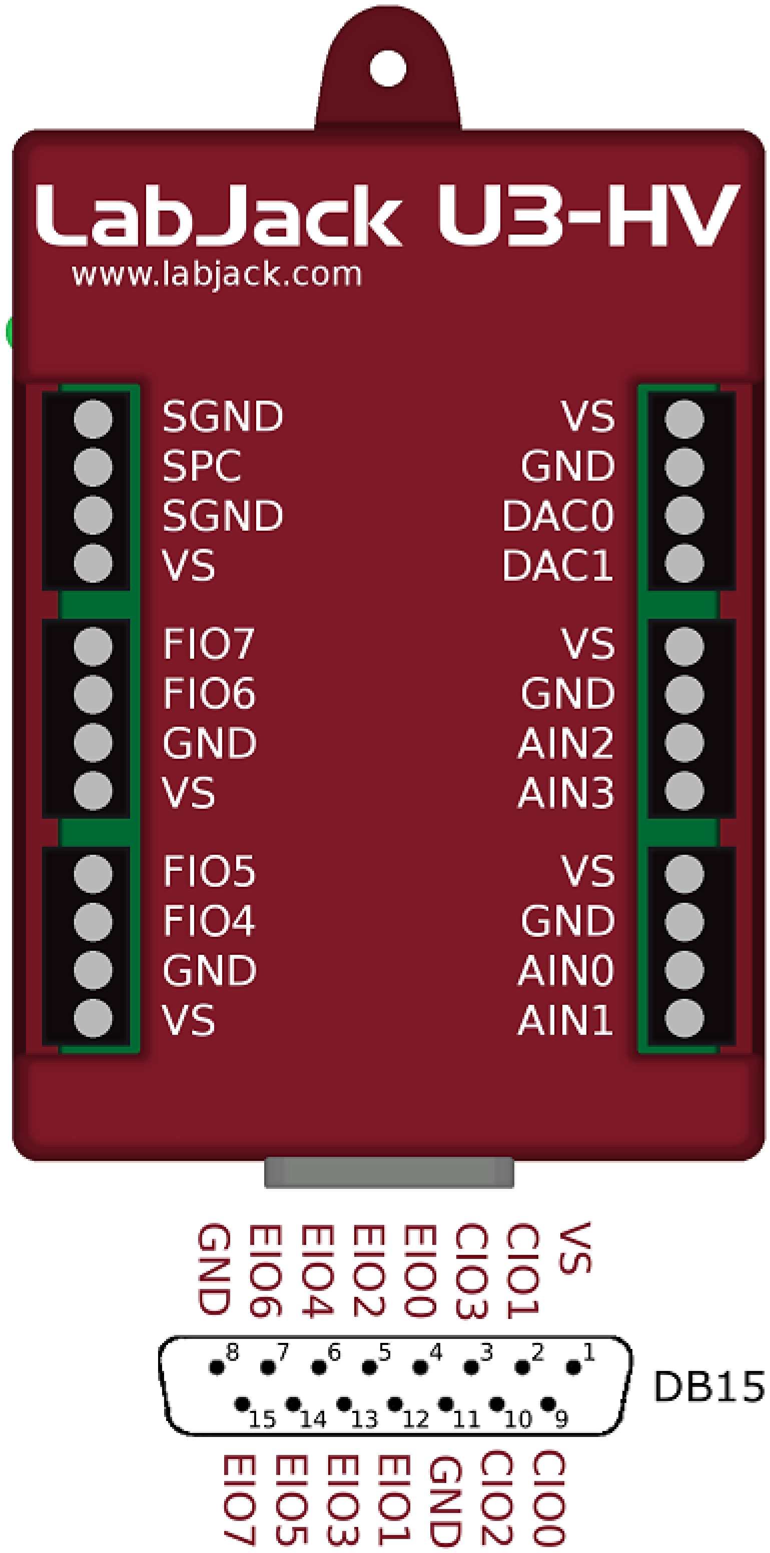
Working with the LabJack

From Kuiper-wiki

Return to Raspberry_Pi#Working_with_the_LabJack

Contents

LabJack Description



The U3 comes in two version, HV (high voltage) and LV (low voltage). However, the package and connectors are the same.

Common Terminals

VS terminals are designed as outputs for the internal supply voltage (nominally 5 volts). This will be the voltage provided from the USB cable. The VS outputs and the digital outputs conform to these specifications

Typical Voltage	Min.	Typ.	Max.	Unit

Self-Powered	4.75	5	5.25	V
Bus-Powered	4	5	5.25	V
Maximum Currrent				
Self-Powered		450		mA
Bus-Powered		50		mA

Self-powered applies to USB hubs with a power supply, all known desktop computer USB hosts, and some notebook computer USB hosts. An example of **bus-powered** would be a hub with no power supply, or many PDA ports. The current rating is the maximum current that should be sources through the U3 and out of the VS and DO terminals.

GND connections available at the screw-terminals and DB connectors provide a common ground for all LabJack functions. This ground is the same as the ground line on the USB connection, which is often the same as ground on the PC chassis and therefore AC mains ground.

SGND is located near the upper-left of the device. This terminal has a self-resetting thermal fuse in series with GND. This is often a good terminal to use when connecting the ground from another separately powered system that could unknowingly already share a common ground with the U3.

Flexible I/O

The first 8 flexible I/O lines (FIO0-FIO7) appear on built-in screw terminals. The other 8 flexible I/O lines (EIO0-EIO7) are available on the DB15 connector. A LabJack CB15 board can connect into the DB15 connector to provide additional screw terminals.

The I/O lines can be configured as digital inputs or outputs, or analog inputs (AIN0-AIN15). This table gives the mapping of I/O lines to screw terminal names.

Label	AIN0	AIN1	AIN2	AIN3	FIO4	FIO5	FIO6	FIO7
Alt.	FIO0	FIO1	FIO2	FIO3	AIN4	AIN5	AIN6	AIN7
Label	EIO0	EIO1	EIO2	EIO3	EIO4	EIO5	EIO6	EIO7
Alt.	AIN8	AIN9	AIN10	AIN11	AIN12	AIN13	AIN14	AIN15

The function of a group of I/O lines is determined according to this table:

Mode	***X*IOAnalog**	***X*IOBitDir**	**Group (*X*)**
analog input	1	0	E,F
digital input		0	C,E,F

digital output		1	C,E,F

`FIOAnalog`, `FIOBitDir`, etc., are values in the configuration dict returned by `configU3()`.

Analog Inputs: The LabJack U3 has up to 16 analog inputs available on the flexible I/O lines (FIO0-FIO7 and EIO0-EIO7). Single-ended measurements can be taken of any line compared to ground, or differential measurements can be taken of any line to any other line. Analog input resolution is 12-bits. For our LV, the range of single-ended analog inputs is normally about 0-2.44. There is a “special” 0-3.6 volt range available. The range of differential analog inputs is typically ± 2.4 volts, but is pseudobipolar, not true bipolar. The difference (positive channel minus negative channel) can be -2.4 volts, but neither input can have a voltage less than -0.3 volts to ground. For valid measurements, the voltage on every low-voltage analog input pin, with respect to ground, must be within -0.3 to +3.6 volts.

Analog Outputs

The LabJack U3 has 2 analog outputs (DAC0 and DAC1) that are available on the screw terminals. Each analog output can be set to a voltage between about 0.04 and 4.95 volts with 10 bits of resolution. The maximum output voltage is limited by the voltage supplied to the U3 over USB.The DAC outputs are derived as a percentage of Vreg, and then amplified by 1.5, so any changes in Vreg will have a proportionate affect on the DAC outputs. Vreg is more stable than Vs (5 volt supply voltage), as it is the output from a 3.3 volt regulator.

DAC1 is always enabled.

Timers and Counters

TimerCounterConfig is used to enable/disable timers and counters. The bit assignments are

Bits	Function
4-7	TimerCounterPinOffset
3	Enable Counter1
2	Enable Counter0
0-1	Number of timers enabled

A value of 0 means no timers/counters, offset=4. Timers/counters will be assigned to IO pins starting with FIO0 plus **TimerCounterPinOffset** (4-8 only). Timer0 takes the first IO pin, then Timer1, Counter0, and Counter1. Whenever this function is called and timers are enabled, the timers are initialized to mode

10, so the desired timer mode must always be specified after every call to this function. Note that Counter0 is not available when using a timer clock base that supports a timer clock divisor (TimerClockBase = 3-6).

Manual Reset

The SPC terminal is use for manually resetting default values or jumping in/out of flash programming mode.

Power up the U3 with a short from FIO6 to SPC, then remove the jumper and power cycle the device again. This resets all power-up settings to factory default values.

Factory Default Configuration

Use this method to restore bytes 9-20 to the factory default value of 0x00. That means

9	TimerCounterConfig	No timers or counter, first timer at FIO4.
10	FIOAnalog	No analog inputs in the F group
11	FIODirection	All F group lines are digital inputs
12	FIOState	Default input/output value on boot-up.
13	EIOAnalog	No analog inputs in the E group
14	EIODirection	All E group lines are digital inputs
15	EIOState	Default input/output value on boot-up.
16	CIODirection	All C group lines are digital inputs
17	CIOState	Default input/output value on boot-up.
18	DAC1Enable	Default state on boot-up.
19	DAC0	Default output value on boot-up.
20	DAC1	Default output value on boot-up.

Watchdog Timer

If the watchdog is accidentally configured to reset the processor with a very low timeout period (such as 1 second), it could be difficult to establish any communication with the device.

Testing the Software

Recognizing the LabJacks

```
pi@Umber-Pi /usr/local/lib/python2.7/DSN-Sci-packages/Electronics $ ipython

In [1]: from Electronics.Interfaces.LabJack import searchForDevices
In [2]: searchForDevices()
Out[2]:
{'320038846': {'_autoCloseSetup': False,
               'changed': {'deviceName': 'U3-LV',
                           'firmwareVersion': '1.24',
                           'hardwareVersion': '1.30',
                           'ipAddress': ,
                           'localId': 1,
                           'serialNumber': 320038846,
                           'versionInfo': 2},
               'debug': False,
               'devType': 3,
               'deviceLock': <thread.lock at 0xb6d89600>,
               'deviceName': 'U3-LV',
               'firmwareVersion': '1.24',
               'handle': None,
               'hardwareVersion': '1.30',
               'ipAddress': ,
               'localId': 1,
               'modbusPrependZeros': True,
               'serialNumber': 320038846,
               'streamConfiged': False,
               'streamPacketOffset': 0,
               'streamStarted': False,
               'versionInfo': 2}}
```

Creating LabJack class instances

```
In [6]: from Electronics.Interfaces.LabJack import connect_to_U3s
In [7]: lj = connect_to_U3s()
In [8]: lj
Out[8]: {1: <Electronics.Interfaces.LabJack.LabJack at 0x1b58250>}
```

LabJack Attributes

```
Help on LabJack in module Electronics.Interfaces.LabJack object:

class LabJack(u3.U3)
 |  U3 subclass with additional attributes and methods
 |
 |  Some inherited methods::
 |    binaryToCalibratedAnalogTemperature()
 |    binaryToCalibratedAnalogVoltage() Bits returned from AIN into voltage.
 |    close()
 |    configAnalog()
 |    configDigital()
 |    configIO()
 |    configTimerClock()
 |    configU3()
 |    getAIN()
 |    getDIOState()       Read the state of a digital I/O.  Will not change the direction.
 |    getDIState()        Set direction to input and read the state of an FIO.
 |    getFeedback()       Sends a commandlist to the U3, and reads the response.
 |    getName()
 |    getTemperature()    Reads the internal temperature sensor on the U3.
 |    loadConfig()        Takes a configuration and updates the U3 to match it.
 |    reset()             Causes a soft or hard reset.
 |    setDIOState()
 |    setDOState()        Set the state of a digital I/O.  Also set the direction to output.
 |    setName()
 |    setToFactoryDefaults()
 |    toggleLED()         Toggles the state LED on and off.
 |    voltageToDACBits()  Takes a voltage, and turns it into the bits
 |    watchdog()          read/write the configuration of the watchdog
 |
 |  Some inherited attributes::
 |    name
 |
 |  Method resolution order:
 |      LabJack
 |      u3.U3
 |      LabJackPython.Device
 |      __builtin__.object
 |
 |  Methods defined here:
 |
 |  __init__(self, serialno)
 |      Instantiate a LabJack
 |
 |  get_AINs(self, prefix)
 |      Get values for all defined analog inputs
 |
 |      @type prefix : str
 |      @param prefix : "F" or "E"
 |
 |      @return: dictionary keyed by signal names
 |
 |  get_dir_bits(self)
 |      Get direction bits
 |
 |  pulse_bit(self, bit)
 |      This drops the designated bit to state 0 for half a second
 |
 |      To ensure that it is seen as a negative pulse, the bit is
 |      set high for one second if it was not already high.
 |
 |  set_DO_bits(self, prefix, byte)
 |      Set digital output bits
 |
 |      @type prefix : str
```

and so on. The U3() class attributes are described in the LabJack Python manual.

LabJack Configuration

```
In [15]: config = lj[1].configU3()
In [16]: config
Out[16]:
{'BootloaderVersion': '0.27',
 'CIODirection': 0,
 'CIOState': 0,
 'CompatibilityOptions': 0,
 'DAC0': 0,
 'DAC1': 0,
 'DAC1Enable': 1,
 'DeviceName': 'U3-LV',
 'EIOAnalog': 0,
 'EIODirection': 0,
 'EIOState': 0,
 'FIOAnalog': 31,
 'FIODirection': 0,
 'FIOState': 0,
 'FirmwareVersion': '1.24',
 'HardwareVersion': '1.30',
 'LocalID': 1,
 'ProductID': 3,
 'SerialNumber': 320038846,
 'TimerClockConfig': 2,
 'TimerClockDivisor': 256,
 'TimerCounterMask': 64,
 'VersionInfo': 2}
```

The I/O groups are bi-directional.

```
In [17]: Cdir, Edir, Fdir = lj[1].get_dir_bits()
In [18]: Cdir,Edir,Fdir
Out[18]: (0, 0, 0)
```

So all are set for input. Note that some lines in the "F" groups are set as analog inputs:

```
In [19]: import Math
In [20]: Math.decimal_to_binary(config['FIOAnalog'],8)
Out[20]: '00011111'
```

Using the LabJack

This turns on and off an LED connected to EIO6:

```
In [13]: lj.setDOState(14)
In [14]: lj.setDOState(14,state=0)
```

We can get the raw readings of the lines configured as analog inputs:

```
In [21]: lj[1].get_AINs('F')
Out[21]:
{'FAIN0': [9168],
 'FAIN1': [9792],
 'FAIN2': [8224],
 'FAIN3': [8528],
 'FAIN4': [12608]}
```

These readings are for open inputs. With two old AAA batteries in series attached to FIO0:

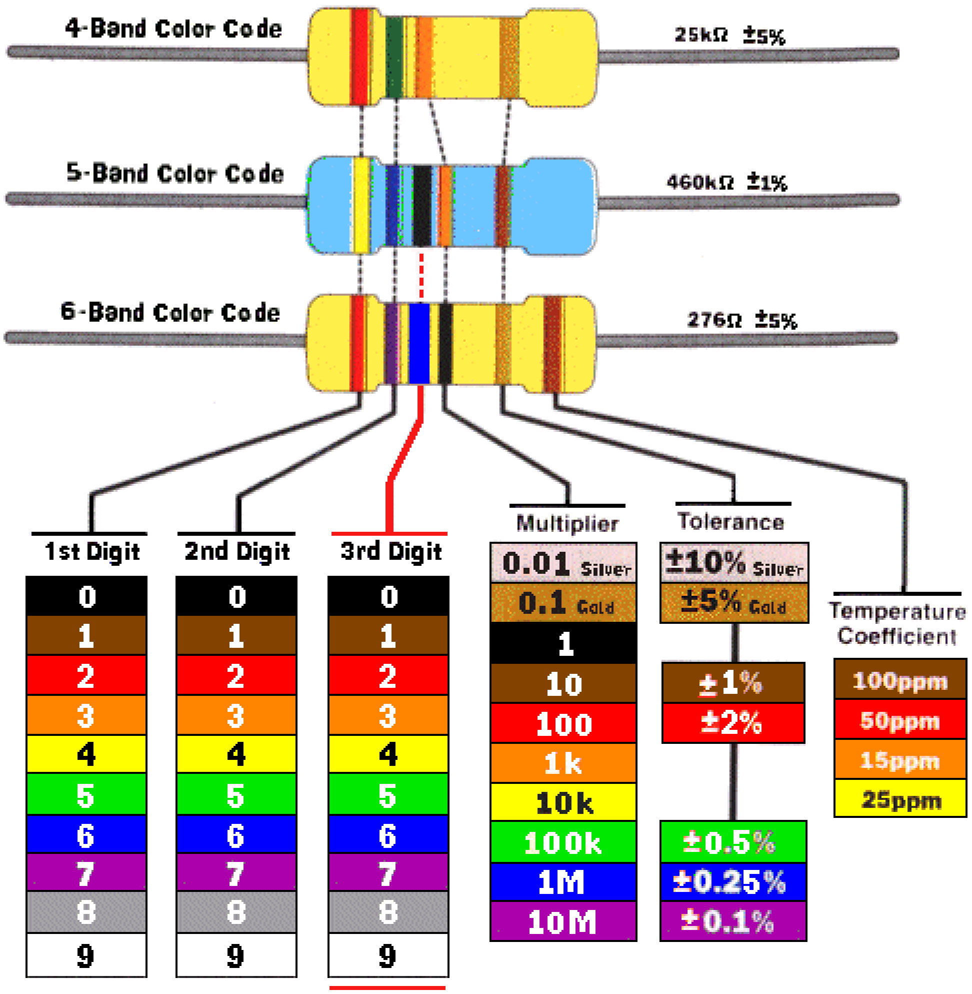
```
In [16]: lj.getAIN(0)
Out[16]: 2.4375880320000003
```

However, a nicer quicker way to get all the interesting data is like this:

```
In [38]: from Interfaces.LabJack import get_IO_states
In [39]: get_IO_states(lj)
Out[39]:
{1: {'CIOBitDir': 0,
  'CIOState': 15,
  'EIOBitDir': 0,
  'EIOState': 255,
  'FAIN0': 0.33839007024653256,
  'FAIN1': 0.3630343221593648,
  'FAIN2': 0.3137458183337003,
  'FAIN3': 0.31013934244401753,
  'FAIN4': 0.5860347480047494,
  'FIOBitDir': 0,
  'FIOState': 224,
  'Temperature': 298.9103810861707}}
```

Electronics Tips

Attaching LEDs



The maximum current draw from a FIO line should be limited to about 50 mA. (The total current drawn must be less than 450 mA for a bus-powered LabJack.) So there should be at least a 100 Ohm resistor in series with a LED. Here is a calculation for a typical LED:

```
Assuming a standard 5mm
(Vin - Vf) / R = I
Vin = 5 V
Vf = 3.2V (found from L
I = 0.02A
5V - 3.2 / R = 0.02A
R = 90 Ohm
```

Dont burn up your resistor! Calculate power in R:

```
P = I^2 * R
P = (0.02 * .02) * 90
P = .036W
```

So 100 Ohm is about right
brown-black-brown or brown-black-black-black.

The polarity of an LED can be obtained from this table

Positive	**Negative**	
terminal	anode (A)	cathode (K)
leads	long	short
exterior	round	flat
wiring	red	black

Retrieved from "http://nutmeg/mediawiki/index.php/Working_with_the_LabJack"

- This page was last modified on 9 March 2014, at 23:39.
-