T7 Datasheet

High performance, USB, Ethernet, WiFi

Multifunction DAQ

To make a PDF of the whole manual, click "Export all" towards the upper-right of this page. Doing so converts these pages to a PDF on-the-fly, using the latest content, and can take 20-30 seconds. Make sure you have a current browser (we mostly test in Firefox and Chrome) and the current version of Acrobat Reader. If it is not working for you, rather than a normal click of "Export all" do a right-click and select "Save link as" or similar. Then wait 20-30 seconds and a dialog box will pop up asking you where to save the PDF. Then you can open it in the real Acrobat Reader, not embedded in a browser. If you still have problems, try the "Print all" option instead.

If you are looking at a PDF or hardcopy, realize that it is likely out-of-date as the original is an online document.

Rather than using a PDF, though, we encourage you to use this web-based documentation. Some advantages:

- We can quickly improve and update content.
- The site search includes the datasheet, forum, and all other resources at labjack.com. When you are looking for something try using the site search.
- For support, try going to the applicable datasheet page and post a comment. When appropriate we can then immediately add/change content on that page to address the question.

One other trick worth mentioning, is to browse the table of contents to the left. Rather than clicking on all the links to browse, you can click on the small black triangles to expand without reloading the whole page.

Datasheet

Preface: Warranty, Liability, Compliance

For the latest version of this and other documents, go to www.labjack.com.

Warranty:

The LabJack T7 is covered by a 1 year limited warranty from LabJack Corporation, covering this product and parts against defects in material or workmanship. The LabJack can be damaged by misconnection (such as connecting 120 VAC to any of the screw terminals), and this warranty does not cover damage obviously caused by the customer. If you have a problem, contact support@labjack.com for return authorization. In the case of warranty repairs, the customer is responsible for shipping to LabJack Corporation, and LabJack Corporation will pay for the return shipping.

Limitation of Liability:

LabJack designs and manufactures measurement and automation peripherals that enable the connection of a PC to the real-world. Although LabJacks have various redundant protection mechanisms, it is possible, in the case of improper and/or unreasonable use, to damage the LabJack and even the PC to which it is connected. LabJack Corporation will not be liable for any such damage.

Except as specified herein, LabJack Corporation makes no warranties, express or implied, including but not limited to any implied warranty or merchantability or fitness for a particular purpose. LabJack Corporation shall not be liable for any special, indirect, incidental or consequential damages or losses, including loss of data, arising from any cause or theory.

LabJacks and associated products are not designed to be a critical component in life support or systems where malfunction can reasonably be expected to result in personal injury. Customers using these products in such applications do so at their own risk and agree to fully indemnify LabJack Corporation for any damages resulting from such applications.

LabJack assumes no liability for applications assistance or customer product design. Customers are responsible for their applications using LabJack products. To minimize the risks associated with customer applications, customers should provide adequate design and operating safeguards.

Reproduction of products or written or electronic information from LabJack Corporation is prohibited without permission. Reproduction of any of these with alteration is an unfair and deceptive business practice.

Conformity Information (FCC, CE, RoHS):

See the Conformity Page and the text below:

FCC PART 15 STATEMENTS:

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense. The end user of this product should be aware that any changes or modifications made to this equipment without the approval of the manufacturer could result in the product not meeting the Class A limits, in which case the FCC could void the user's authority to operate the equipment.

Declaration of Conformity:

Manufacturers Name: LabJack Corporation
Manufacturers Address: 3232 S Vance St STE 100, Lakewood, CO 80227, USA

Declares that the product

Product Name: LabJack T7 (-Pro)
Model Number: LJT7 (-Pro)

conforms to the following Product Specifications:

EMC Directive: 2004/104/EEC

EN 55011 Class A
EN 61326-1: General Requirements

and is marked with CE

RoHS:

The T7 (-Pro) is RoHS compliant per the requirements of Directive 2002/95/EC.

Preface

1.0 Device Overview

This document contains device-specific information for the following devices:

- T7
- T7-Pro
- T7-OEM
- T7-Pro-OEM

This family introduces a new line of high-quality analog and Ethernet data acquisition hardware, with the main traditional advantage of all LabJack data acquisition hardware, namely, high performance and rich feature set at a competitive price point. These features make the T series a logical choice for many high-performance applications, where Ethernet, WiFi, and cost are primary considerations.

1.1 Core Features

Analog I/O

- 14 Analog Inputs (16-18+ Bits Depending on Speed)
- Single-Ended Inputs (14) or Differential Inputs (7)
- Instrumentation Amplifier Inputs
- Software Programmable Gains of x1, x10, x100, and x1000
- Analog Input Ranges of ±10, ±1, ±0.1, and ±0.01 Volts
- 2 Analog Outputs (12-Bit, ~0-5 Volts)

Digital I/O

- 23 Digital I/O
- Supports SPI, I2C, and Asynchronous Serial Protocols (Master Only)
- Supports Software or Hardware Timed Acquisition
- Maximum Input Stream Rate of 100 kHz (Depending on Resolution)
- Capable of Command/Response Times Less Than 1 Millisecond

Digital I/O Extended Features

- Simple PWM Output (1-32 bit)
- PWM Output w/ phase control
- Pulse Output w/ phase control
- Positive edge capture
- Negative edge capture
- PWM measure
- Edge capture & compare
- High speed counter (TBD ~40 MHz)
- Software counter (TBD ~200 kHz)
- Software counter w/ debounce
- Quadrature Input

Other highlights

- Built-In CJC Temperature Sensor
- Watchdog system
- Field Upgradable Firmware
- Programmable Startup Defaults
- LJTick Compatible

Fixed Current Outputs

- 200 µA
- 10 µA

1.2 Family Variants Info

T7 vs T7-Pro

The T7-Pro has all features of the normal T7, with the following added:

- Wireless Ethernet 802.11b/g
- 24-bit Low-Speed ADC for 22-Bit Effective Resolution

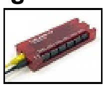
Also see the block diagram in the hardware overview section.

T7-OEM and T7-Pro-OEM
There is also an OEM version of the T7 and T7-OEM. The OEM versions are the same in terms of features, but the enclosure, and most connectors are not installed on the OEM versions, which allows customers to easily configure as needed. See Appendix A - OEM Versions for details.

3.0 Installation

1. First install LabJack software and driver bundle based on your operating system.

   T7/Digit Devices

Windows Installer	49.68 MB	2013-10-14 16:39	details
Mac OS X Package	2.5 MB	2013-11-18 18:14	details
Linux 32-bit Package	1.32 MB	2013-11-18 18:09	
Linux 64-bit Package	1.32 MB	2013-11-18 18:07	details

2. Connect the T7 to the local computer via USB.
3. Proceed through any steps to add new hardware.
4. If using Windows, open Kipling (installed with package above). Utility apps for other operating systems are still under development.
5. Use the dashboard in Kipling to view analog inputs, digital I/O, DAC outputs, etc.
6. Go to quickstart page to see more about Kipling and its use with the T7.

4.0 Communication

Talk to the T7...

Modbus TCP is the protocol used by all connections on the T7(USB, Ethernet, WiFi). All important values & data from the device can be read and/or written by using the associated Modbus register(s). Thus, the process for reading the serial number, an analog input, or a waveform is all functionally the same, you simply provide a different address. There are two main ways to communicate with a T7 using Modbus TCP.

Option 1: Use the high-level LJM library for communication with the T7.

Among other useful features, this cross-platform library allows users to access registers by name, such as "AIN4" for analog input 4. Most people will use the LJM library since they're familiar with writing code, and want to integrate a T7 into an existing software framework.

Conceptual workflow:

1. Find example code/wrappers for your desired programming language.
2. Use the LJM_Open() function to open a connection to the T7.
3. Perform reads and writes to Modbus registers using LJM_eReadName() or LJM_eWriteName().
4. Use the Close() function to close the connection.

Option 2: Directly communicate with a T7 over Ethernet or Wi-Fi using any standard COTS Modbus TCP software.

It is easy to integrate a T7 into existing Modbus platforms, since the T7 is directly compatible. People who already use Modbus software will find this option convenient. Some COTS Modbus software is very powerful, and will save users the time and money required to develop their own software.

Conceptual workflow:

1. Configure the power-up-default registers on the T7 using the Kipling software program. Change Ethernet/WiFi IP settings, any relevant analog input settings, etc. '..._DEFAULT' registers indicate that they are power-up-defaults.
2. Open COTS Modbus program.
3. Specify the Modbus registers by address, such as 8, for AIN4. Find applicable registers with the register look-up tool, or by referencing the datasheet etc.
4. See data directly from the T7 in COTS software.

4.1 Modbus Map

http://labjack.com/support/modbus/map

4.2 C-R and Streaming

Command-Response

This is the default behavior for communication with a device, and most people find the data throughput satisfactory.

Communication is initiated by a command from the host which is followed by a response from the device. In other words, data transfer is software-paced. Command-response is generally used at 1000 scans/second or slower and is generally simpler than stream mode.

Command-response mode is generally best for minimum-latency applications such as feedback control. By latency here we mean the time from when a reading is acquired to when it is available in the host software. A reading or group of readings can be acquired in times on the order of a millisecond.

Streaming

Stream mode is generally best for maximum-throughput applications. However, streaming is not recommended for feedback control operations, due to the latency in data recovery. Data is acquired very fast, but to sustain the fast rates it must be buffered and moved from the LabJack to the host in large chunks.

Stream mode is a continuous hardware-paced input mode where a list of addresses is scanned at a specified scan rate. The scan rate specifies the interval between the beginning of each scan. The samples within each scan are acquired as fast as possible. As samples are collected automatically by the LabJack, they are placed in a buffer on the LabJack, until retrieved by the host. Stream mode is generally used when command-response is not fast enough. Stream mode is not supported on the hi-res converter (resolutions 9-12 not supported in stream).

For example, a typical stream application might set up the LabJack to acquire a single analog input at 50,000 samples/second. The LabJack moves this data to the host in chunks of 25 samples each. The LJM library moves data from the USB host memory to the software memory in chunks of 2000 samples. The user application might read data from memory once a second in a chunk of 50,000 samples. The computer has no problem retrieving, processing, and storing, 50k samples once per second, but it could not do that with a single sample 50k times per second.

Command-response can be done while streaming, but streaming needs exclusive control of the analog input system so analog inputs (including the internal temperature sensor) cannot be read via command-response while a stream is running.

5.0 Hardware Overview

The T7 has 3 different I/O areas:

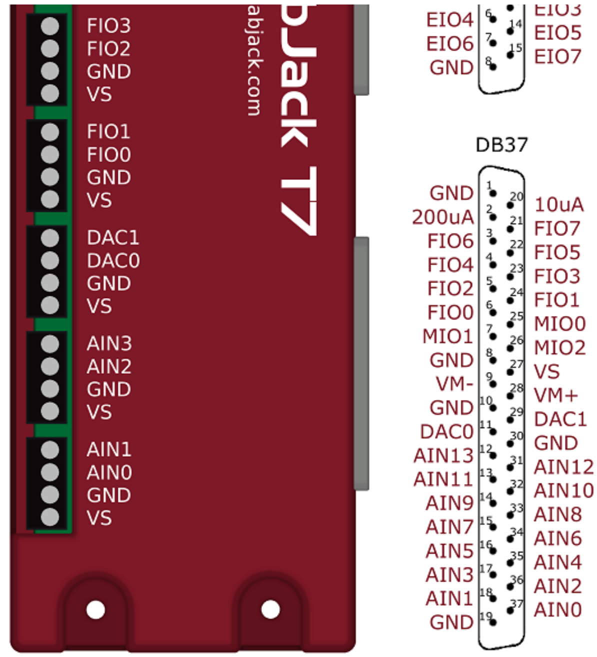
- Communication Edge
- Screw Terminal Edge
- DB Edge

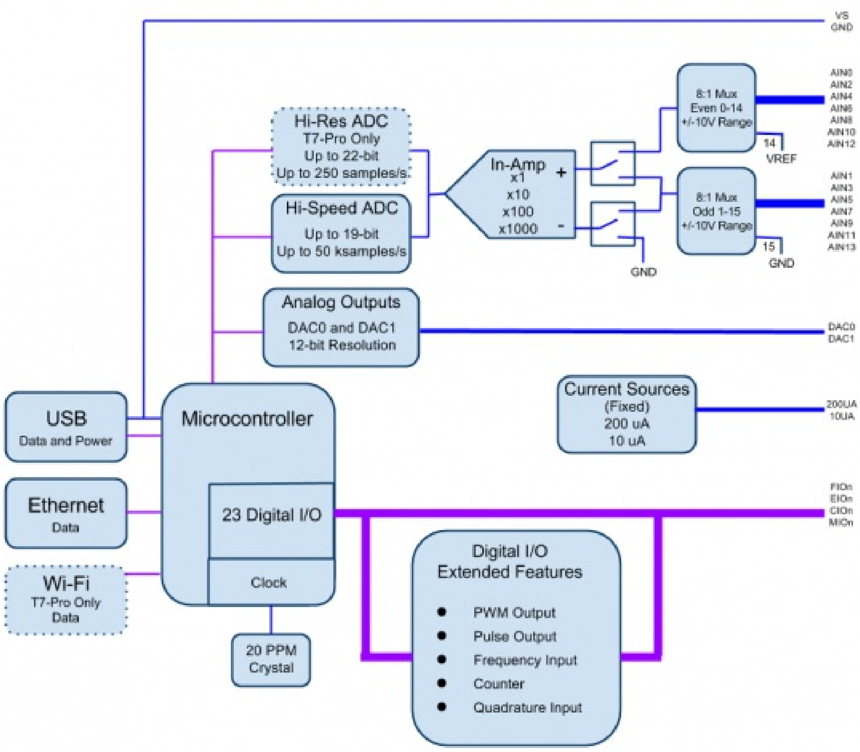
The communication edge has a USB type B connector, an RJ45 Ethernet connector, and in the case of the T7-Pro also has a SMA-RP Female Connector and a WiFi antenna. Power is always provided through the USB connector, even if USB

communication is not used.

The screw terminal edge has convenient connections for 4 analog inputs, both analog outputs, 4 digital I/O, and both current sources. The screw terminals are arranged in blocks of 4, with each block consisting of VS, GND, and two I/O. Also on this edge are two LEDs. The Comm LED generally blinks with communication traffic, while the Status LED is used for other indications.

The DB Edge has 2 D-sub type connectors: a DB15 and DB37. The DB15 has 12 additional digital I/O. The DB37 has the same I/O as the screw-terminals, plus additional analog inputs and digital I/O, for a total of 14 analog inputs, 2 analog outputs, 2 fixed current sources, and 11 digital I/O.





Digital waveforms can be output/input on various digital I/O lines, using extended features.

General Device Information

Name	Start Address	Type	Access	Default
PRODUCT_ID	60000	FLOAT32	R	
HARDWARE_VERSION	60002	FLOAT32	R	
FIRMWARE_VERSION	60004	FLOAT32	R	
BOOTLOADER_VERSION	60006	FLOAT32	R	
WIFI_VERSION	60008	FLOAT32	R	

HARDWARE_INSTALLED	60010	UINT32	R	0
ETHERNET_MAC	60020	UINT64	R	
WIFI_MAC	60024	UINT64	R	
SERIAL_NUMBER	60028	UINT32	R	
DEVICE_NAME_DEFAULT	60500	STRING	R/W	

PRODUCT_ID
The numeric identifier of the device. Such as 3 for a U3-HV.

HARDWARE_VERSION
The hardware version of the device.

FIRMWARE_VERSION
The current firmware version installed on the main processor.

BOOTLOADER_VERSION
The bootloader version installed on the main processor.

WIFI_VERSION
The current firmware version of the WiFi module, if available.

HARDWARE_INSTALLED
Bitmask indicating installed hardware options.

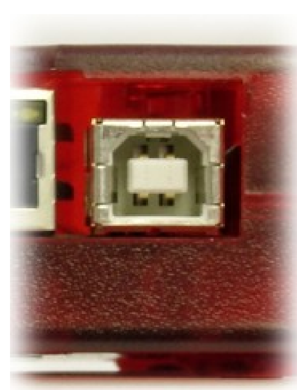
ETHERNET_MAC
The MAC address of the wired Ethernet module.

WIFI_MAC
The MAC address of the WiFi module.

SERIAL_NUMBER
The serial number of the device.

DEVICE_NAME_DEFAULT
The current device name. Up to 49 characters, cannot contain periods.

6.0 USB



Interface: **ModBus TCP**

Connector Type: **USB-B Receptacle**

Compatible: **USB 1.1+**

Power is supplied to the T7 through the 5V USB connection. If the Ethernet or Wi-Fi connection is preferred for communication, use the provided AC USB 5V adapter for power. When used for communication, it is a full-speed USB connection compatible with USB version 1.1 or higher.

Interface - Talk to the T7

Modbus TCP is the protocol used by all connections on the T7(USB, Ethernet, WiFi). Any platform that supports TCP/IP can directly communicate with a T7. However, for customers who are not interested in Modbus, we provide a cross-platform driver that extends/wraps the protocol. This driver, referred to mostly as the LJM library, provides convenient device discovery, high-level functions, and programming flexibility.

If you choose not to use the supplied LJM library, reference the Native Modbus documentation for information about packets/layout. The LJM library is cross-platform, and very flexible, so most users should not concern themselves with the actual Modbus protocol.

Power Considerations

USB ground is connected to the T7 ground (GND), and USB ground is generally the same as the ground

of the PC chassis and AC mains, since standard USB is non-isolated.

It is possible to isolate USB, and thereby protect the T7 from a power surge coming through the computer, if you use a USB isolator. USB isolators typically go for $40 to $100 USD, depending on the capabilities.

The T7-Pro will generally require a powered USB hub when in operating at full-power, since some USB ports/hubs will not supply the current necessary (500mA). Our experience with cheap USB supplies has shown them to be unreliable above 200mA. We recommend a powered USB hub rated for battery charging applications, since these are typically rated for 1-2A. See electrical specifications for details on USB current requirements.

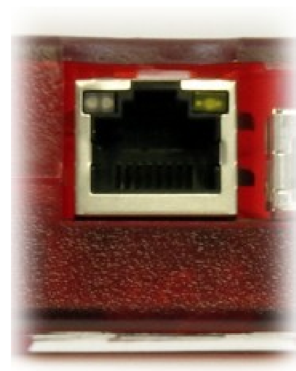
If designing your own driver...

The LabJack vendor ID is 0x0CD5. The product ID for the T7 is 0x0007.

The USB interface consists of the normal bidirectional control endpoint (0 OUT & IN), 3 used bulk endpoints (1 OUT, 2 IN, 3 IN), and 1 dummy endpoint (3 OUT). Endpoint 1 consists of a 64 byte OUT endpoint (address = 0x01). Endpoint 2 consists of a 64 byte IN endpoint (address = 0x82). Endpoint 3 consists of a dummy OUT endpoint (address = 0x03) and a 64 byte IN endpoint (address = 0x83). Endpoint 3 OUT is not supported by the firmware, and should never be used.

All commands should always be sent on Endpoint 1, and the responses to commands will always be on Endpoint 2. Endpoint 3 is only used to send stream data from the T7 to the host.

7.0 Ethernet



Connector Type: **RJ-45 Socket, Cat 5**

POE Compatible: **No**[1]

Overview

The T7 has a 10/100Base-T Ethernet connection. This connection only provides communication, so power must be provided through the USB connector. Refer to this WiFi and Ethernet tutorial to get started.

Current settings can be read from the registers below. To make changes, write the changes to the default registers, and power cycle the device. DHCP is enabled by default. To enable a static IP address, write a 0 to the **ETHERNET_DHCP_ENABLE_DEFAULT** register. Configure the Ethernet parameters in Kipling software, such as IP, gateway, etc.

Ethernet Settings

Name	Start Address	Type	Access	Default
ETHERNET_IP	49100	UINT32	R	
ETHERNET_SUBNET	49102	UINT32	R	
ETHERNET_GATEWAY	49104	UINT32	R	
ETHERNET_DNS	49106	UINT32	R	
ETHERNET_ALTDNS	49108	UINT32	R	
ETHERNET_DHCP_ENABLE	49110	UINT16	R	
ETHERNET_IP_DEFAULT	49150	UINT32	R/W	
ETHERNET_SUBNET_DEFAULT	49152	UINT32	R/W	
ETHERNET_GATEWAY_DEFAULT	49154	UINT32	R/W	
ETHERNET_DNS_DEFAULT	49156	UINT32	R/W	
ETHERNET_ALTDNS_DEFAULT	49158	UINT32	R/W	
ETHERNET_DHCP_ENABLE_DEFAULT	49160	UINT16	R/W	

ETHERNET_IP
Read the current IP address of wired Ethernet.

ETHERNET_SUBNET
Read the current subnet of wired Ethernet.

ETHERNET_GATEWAY
Read the current gateway of wired Ethernet.

ETHERNET_DNS
Read the current DNS of wired Ethernet.

ETHERNET_ALTDNS
Read the current Alt DNS of wired Ethernet.

ETHERNET_DHCP_ENABLE
Read the current Enabled/Disabled state of Ethernet DHCP.

ETHERNET_IP_DEFAULT
The IP address of wired Ethernet after a power-cycle to the device.

ETHERNET_SUBNET_DEFAULT
The subnet of wired Ethernet after a power-cycle to the device.

ETHERNET_GATEWAY_DEFAULT
The gateway of wired Ethernet after a power-cycle to the device.

ETHERNET_DNS_DEFAULT
The DNS of wired Ethernet after a power-cycle to the device.

ETHERNET_ALTDNS_DEFAULT
The Alt DNS of wired Ethernet after a power-cycle to the device.

ETHERNET_DHCP_ENABLE_DEFAULT
The Enabled/Disabled state of Ethernet DHCP after a power-cycle to the device.

Some Examples

Read IP Example: To read the wired IP Address of a device, perform a modbus read of address 49100. The value will be returned as an unsigned 32-bit number, such as 3232235691. Change this number to an IP address by converting each binary group to an octet, and adding decimal points as necessary. The result in this case would be "192.168.0.171".

Change IP Example: To change the Ethernet IP Address of a device, perform a modbus write to address 49150. The value must be passed as an unsigned 32-bit number, such as 3232235691. Change this IP address "192.168.0.171" by converting each octet to a binary group, and sticking them together.

More Details

Once default Ethernet configuration register(s) are changed, the current settings will be updated on the next power cycle. Alternatively, toggle power to the Ethernet module by writing a 0, then a 1 to the **POWER_ETHERNET** address.

Name	Start Address	Type	Access	Default
POWER_ETHERNET	48003	UINT16	R/W	
POWER_ETHERNET_DEFAULT	48053	UINT16	R/W	

POWER_ETHERNET
The current ON/OFF state of the Ethernet module. Provided to optionally reduce power consumption.

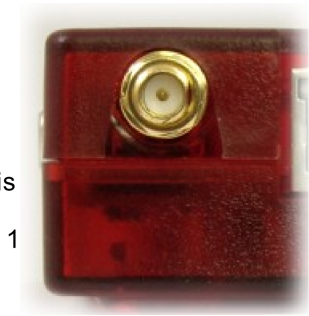
POWER_ETHERNET_DEFAULT
The ON/OFF state of the Ethernet module after a power-cycle to the device.

[1] The T7 cannot be directly powered via POE cable. However, it is relatively easy to find a POE splitter that converts 48V on POE to the 5V necessary for the T7. Such adapters run from ~$30 to ~$50 USD. Used in combination, the following parts work to split POE:

- TP-LINK TL-POE10R - To split 48V from the Ethernet cable into a 5.5mm OD, 2.1mm ID center positive barrel receptacle.
- Tensility International Corp 10-00240 - To convert 5.5mm OD, 2.1mm ID center positive barrel connector to USB-A male plug.

- Tensility International Corp 10-00648 - A Female to B Male USB cable. This will fit on the USB-A male plug (Tensility 10-00240), and insert into the T7.

8.0 WiFi



Connector Type: **Female RP-SMA**

Overview

The T7-Pro has a wireless chip. Refer to this WiFi and Ethernet tutorial to get started.

Configure the default wireless parameters in Kipling software, such as IP, gateway, DHCP etc. DHCP is enabled by default, so to get WiFi going from the factory write the desired SSID string to WIFI_SSID_DEFAULT and the proper password string to WIFI_PASSWORD_DEFAULT. Then write a 1 to WIFI_APPLY_SETTINGS and watch the status codes. If you get back code 2900 the WiFi chip is associated to your network, and you can then read the assigned IP from WIFI_IP.

Use the T7-Pro in the same way you would use a standard T7 over Ethernet, but with the Wireless IP address.

WiFi Settings

Name	Start Address	Type	Access	Default
WIFI_IP	49200	UINT32	R	
WIFI_SUBNET	49202	UINT32	R	
WIFI_GATEWAY	49204	UINT32	R	
WIFI_DHCP_ENABLE	49210	UINT16	R	
WIFI_SSID	49300	STRING	R	
WIFI_IP_DEFAULT	49250	UINT32	R/W	
WIFI_SUBNET_DEFAULT	49252	UINT32	R/W	
WIFI_GATEWAY_DEFAULT	49254	UINT32	R/W	
WIFI_DHCP_ENABLE_DEFAULT	49260	UINT16	R/W	
WIFI_SSID_DEFAULT	49325	STRING	R/W	
WIFI_PASSWORD_DEFAULT	49350	STRING	W	
WIFI_APPLY_SETTINGS	49400	UINT32	W	

WIFI_IP
Read the current IP address of WiFi.

WIFI_SUBNET
Read the current subnet of WiFi.

WIFI_GATEWAY
Read the current gateway of WiFi.

WIFI_DHCP_ENABLE
Read the current Enabled/Disabled state of WiFi DHCP.

WIFI_SSID
Read the current SSID (network name) of WiFi

WIFI_IP_DEFAULT
The new IP address of WiFi. Use WIFI_APPLY_SETTINGS.

WIFI_SUBNET_DEFAULT
The new subnet of WiFi. Use WIFI_APPLY_SETTINGS.

WIFI_GATEWAY_DEFAULT

The new gateway of WiFi. Use WIFI_APPLY_SETTINGS.

WIFI_DHCP_ENABLE_DEFAULT
The new Enabled/Disabled state of WiFi DHCP. Use WIFI_APPLY_SETTINGS

WIFI_SSID_DEFAULT
The new SSID (network name) of WiFi. Use WIFI_APPLY_SETTINGS.

WIFI_PASSWORD_DEFAULT
Write the password for the WiFi network, then use WIFI_APPLY_SETTINGS.

WIFI_APPLY_SETTINGS
Apply all new WiFi settings: IP, Subnet, Gateway, DHCP, SSID, Password. 1=Apply

Some Examples

Read IP Example: To read the wireless IP Address of a device, perform a modbus read of address 49200. The value will be returned as an unsigned 32-bit number, such as 3232235691. Change this number to an IP address by converting each binary group to an octet, and adding decimal points as necessary. The result in this case would be "192.168.0.171"

Write IP Example: To change the Wireless IP Address of a device, perform a modbus write to address 49250. The IP address must be passed as an unsigned 32-bit number, such as 3232235691. Change this IP address "192.168.0.171" by converting each octet to a binary group, and sticking them together.

More Details

Once default wireless configuration register(s) are changed, it is necessary to also write 1 to the **WIFI_APPLY_SETTINGS** register. Alternatively, the default settings will be updated on the next power cycle.

Name	Start Address	Type	Access	Default
POWER_WIFI	48004	UINT16	R/W	
POWER_WIFI_DEFAULT	48054	UINT16	R/W	
WIFI_RSSI	49452	FLOAT32	R	

POWER_WIFI
The current ON/OFF state of the WiFi module. Provided to optionally reduce power consumption.

POWER_WIFI_DEFAULT
The ON/OFF state of the WiFi module after a power-cycle to the device.

WIFI_RSSI
WiFi RSSI (signal strength). Typical values are -40 for very good, and -75 for very weak.

Update WiFi Firmware

The WiFi chip on the T7 is a separate chip from the main processor, and it can be updated using the WIFI_FIRMWARE_UPDATE_TO_VERSIONX register. If connected to the internet, the WiFi chip can download new firmware files from an ftp server. To innitiate a download and update, write a new firmware version to the WIFI_FIRMWARE_UPDATE_TO_VERSIONX register. *At the time of this writing we recommend using Kipling to update WiFi firmware, since Kipling connects to the FTP server to identify what firmware is available, and monitors the WIFI_FIRMWARE_UPDATE_STATUS register automatically.*

Name	Start Address	Type	Access	Default
WIFI_VERSION	60008	FLOAT32	R	
WIFI_FIRMWARE_UPDATE_TO_VERSIONX	49402	FLOAT32	W	
WIFI_FIRMWARE_UPDATE_STATUS	49454	UINT32	R	

WIFI_VERSION
The current firmware version of the WiFi module, if available.

WIFI_FIRMWARE_UPDATE_TO_VERSIONX
Start an update by using USB or Ethernet to write the desired version to update to.

WIFI_FIRMWARE_UPDATE_STATUS
Status Codes: 2920 Configuring, 2921 In Progress, 2923 Rebooting, 2924 Update Success, 2925 Update Failed.

WiFi Status Codes

The WIFI_STATUS register indicates the current status of the WiFi chip. During startup and upon changing settings on the WiFi module, the following codes are helpful for troubleshooting.

Name	Start Address	Type	Access	Default
WIFI_STATUS	49450	UINT32	R	

WIFI_STATUS
Status Codes: 2900 Associated, 2901 Associating, 2902 Association Failed, 2903 Unpowered, 2904 Booting, 2905 Start Failed, 2906 Applying Settings, 2907 DHCP Started, 2909 Other.

```
#define WIFI_ASSOCIATED              2900
#define WIFI_ASSOCIATING             2901
#define WIFI_ASSOCIATION_FAILED      2902
#define WIFI_UNPOWERED               2903
#define WIFI_BOOTING_UP              2904
#define WIFI_COULD_NOT_START         2905
#define WIFI_APPLYING_SETTINGS       2906
#define WIFI_DHCP_STARTED            2907
#define WIFI_OTHER                   2909
```

9.0 LEDs

STATUS - green LED

The status LED is mainly reserved to indicate when Lua scripts are running. The LED will blink when the script does something.

The status LED also activates during firmware updates to indicate various stages of the process, refer to the Combined LED Activity section.

COMM - yellow LED

The primary indicator for packet transfer. If the T7 is communicating the COMM LED will be blinking. A few blinks after connecting to the PC indicates that the T7 is enumerating. Enumeration is when the standard USB initialization takes place, and the host is gathering device information.

The COMM LED will blink when the T7 receives Modbus commands, or when streaming data. Each packet will produce a single blink. If commands are issued rapidly, the LED will blink rapidly. At high packet transfer rates the LED will blink at 10Hz, even though more than 10 packets are being processed per second.

Combined LED Activity

When the LEDs blink together, the T7 is computing checksums.

When the LEDs are alternating, the T7 is copying a firmware image.

10.0 Current Sources (200uA, 10uA)

Overview

The T7 has 2 fixed current source terminals useful for measuring resistance (thermistors, RTDs, resistors). The 10UA terminal provides about 10 µA and the 200UA terminal provides about 200 µA.

The actual value of each current source is noted during factory calibration and stored with the calibration constants on the device. These can be viewed using the Kipling software, or read programmatically. Note that these are fixed constants stored during calibration, not some sort of current readings.

Using the equation V=IR, with a known current and voltage, it is possible to calculate the resistance of the item in question. Figure 2.5-1 shows a simple setup measuring 1 resistor.

Constant Current Sources

Name	Start Address	Type	Access	Default
CURRENT_SOURCE_200UA_CAL_VALUE	1902	FLOAT32	R	
CURRENT_SOURCE_10UA_CAL_VALUE	1900	FLOAT32	R	

CURRENT_SOURCE_200UA_CAL_VALUE

CURRENT_SOURCE_10UA_CAL_VALUE

For example: To read the actual value of the 200uA current source, perform a read of Modbus address 1902, and the result would be in the form of a floating point number, e.g. 0.000197456 amps.

Some Examples

Multiple resistances can be measured by putting them in series and measuring the voltage across each. Some applications might need to use differential inputs to measure the voltage across each resistor, but for many applications it works just as well to measure the single-ended voltage at the top of each resistor and subtract in software.

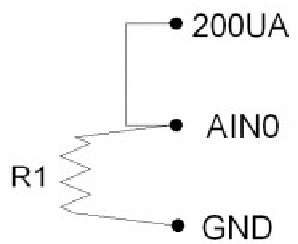


Figure 2.5-1

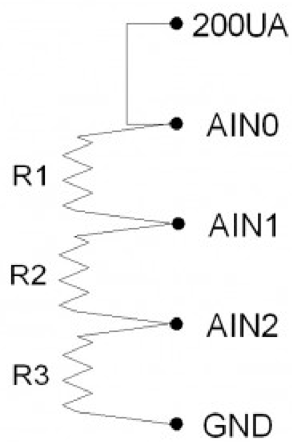


Figure 2.5-2

Figure 2.5-1 shows a simple setup measuring 1 resistor. If R1=3k, the voltage at AIN0 will be 0.6 volts.

Figure 2.5-2 shows a setup to measure 3 resistors using single-ended analog inputs. If R1=R2=R3=3k, the voltages at AIN0/AIN1/AIN2 will be 1.8/1.2/0.6 volts. That means AIN0 and AIN1 would be measured with the +/-10 volt range, while AIN2 could be measured with the +/-1 volt range. This points out a potential advantage to differential measurements, as the differential voltage across R1 and R2 could be measured with the +/-1 volt range, providing better resolution.

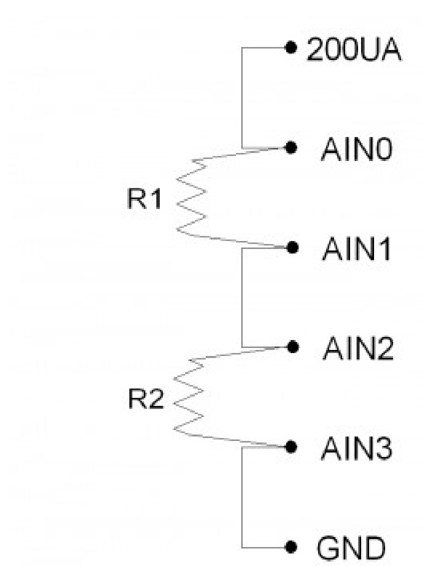


Figure 2.5-3

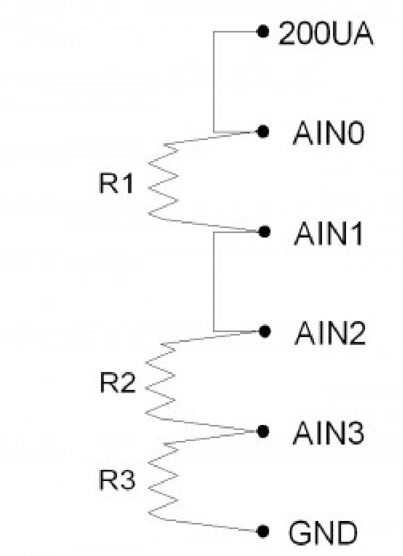


Figure 2.5-4

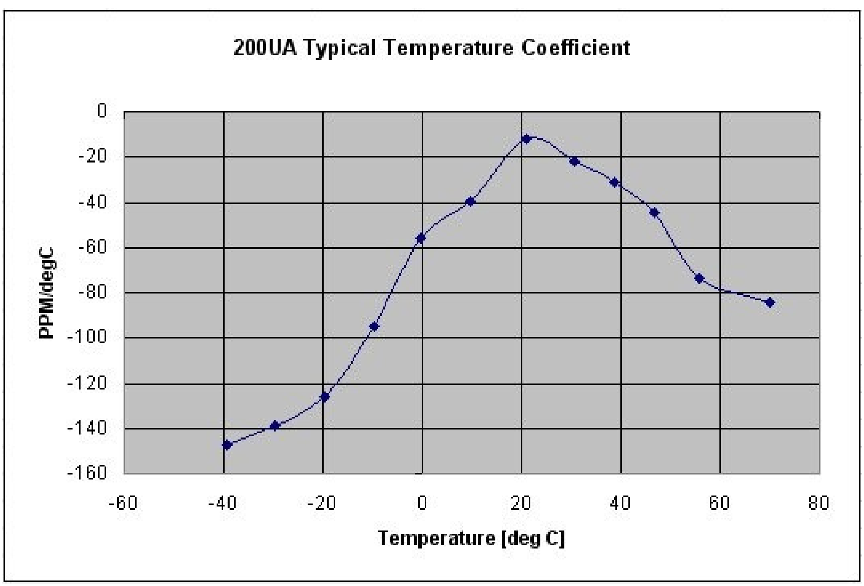
Figure 2.5-3 shows a setup to measure 2 resistors using differential analog inputs. AIN3 is wasted in this case, as it is connected to ground, so a differential measurement of AIN2-AIN3 is the same as a single-ended measurement of AIN2. That leads to Figure 2.5-4, which shows R1 and R2 measured differentially and R3 measured single-ended.

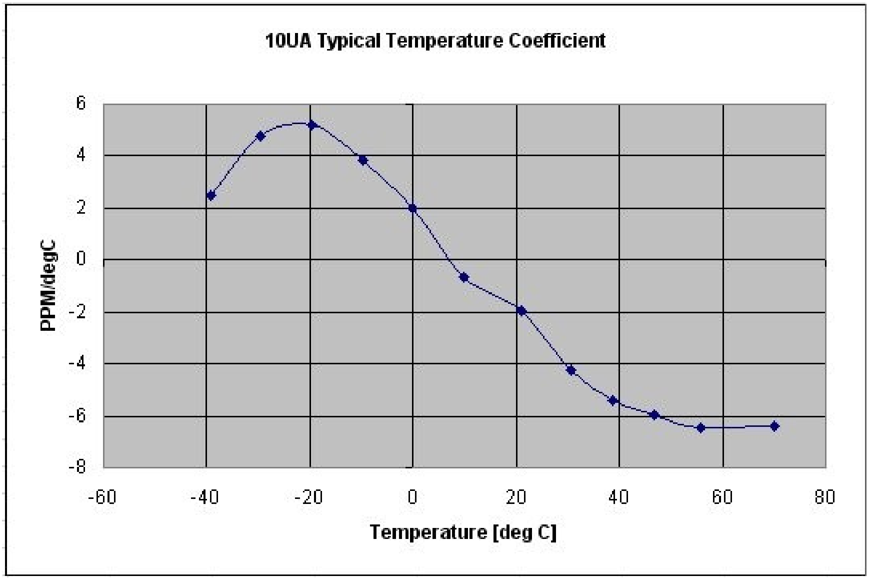
Specifications

The current sources can drive about 3 volts max, thus limiting the maximum load resistance to about 300 kO (10UA) and 15 kO (200UA). Keep in mind that high source resistance could cause settling issues for analog inputs.

The current sources have good accuracy and tempco, but for improvement a fixed resistor can be used as one of the resistors in the figures below. The Y1453-100 and Y1453-1.0K from Digikey have excellent accuracy and very low tempco. By measuring the voltage across one of these you can calculate the actual current at any time.

The following charts show the typical tempco of the current sources over temperature. The 10UA current source has a very low tempco across temperature. The 200 UA current source has a good tempco from about 0-50 degrees C, and outside of that range the effect of tempco will be more noticeable.





10.0 200uA and 10uA

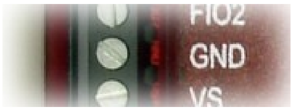
11.0 SGND and GND

SGND

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 T7.

See the AIN, DAC, and Digital I/O(FIO, EIO, CIO, MIO) application notes for more information about grounding.

GND



The GND connections available at the screw-terminals and DB connectors provide a common ground for all LabJack functions. All GND terminals are the same and connect to the same ground plane.

GND is also connected to the ground pin on the USB connector, so if there is a connection to a USB port on a hub/host (as opposed to just a power supply connection), then GND is the same as the ground line on the USB connection, which is often the same as ground on the PC chassis, which is often the same as AC mains ground.

See the AIN, DAC, and Digital I/O (FIO, EIO, CIO, MIO) Sections for more information about grounding.

12.0 SPC

Unless the T7 has problems SPC is not typically needed.

During startup the T7 will look for connections between digital I/O and the SPC terminal. The following list describes what will happen when a jumper wire is placed between SPC and a listed IO.

SPC wired to:

- **FIO0**: Force boot to main firmware (internal) image.
- **FIO1**: Force copy of backup image to overwrite internal image.

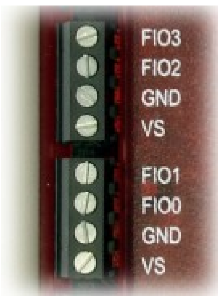
- **FIO2**: Factory reset.
- **FIO3**: Load emergency image. This option loads a firmware image with minimal functionality (kinda like Windows safe-mode). The update process is about all that can be done while in this mode.

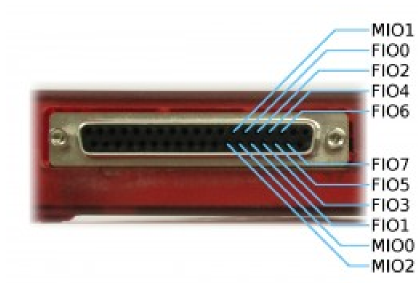
13.0 VS

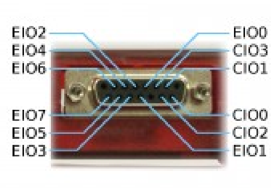
The VS terminals are designed as outputs for the internal supply voltage (nominally 5 volts). This will be the voltage provided from the USB cable. The VS connections are outputs, not inputs. Do not connect a power source to VS in normal situations. All VS terminals are the same.

Related specifications can be found in Appendix TBD.

14.0 Digital I/O







Digital I/O: **23**

Logic Level: **3.3V**

DIO is a generic name used for all digital I/O. The DIO are subdivided into different ports called FIO, EIO, CIO, and MIO.

	FIO (0-7)								EIO (0-7)								CIO (0-3)				MIO (0-2)		
DIO	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**

There are 4 types of registers used for digital I/O interaction: Simple DIO#, DIO State, DIO Direction, and DIO Inhibit

Simple Digital I/O#

Read or set the state of 1 digital I/O. Automatically configures the direction to input (when reading) or to output (when writing).

```
FIO0-FIO7 = DIO0-DIO7
EIO0-EIO7 = DIO8-DIO15
CIO0-CIO3 = DIO16-DIO19
MIO0-MIO2 = DIO20-DIO22
```

Address	Name	Type	Read/Write	Default
2000	DIO0	UINT16	Read/Write	
2001	DIO1	UINT16	Read/Write	
2002	DIO2	UINT16	Read/Write	
...	...	...	Read/Write	
2023	DIO22	UINT16	Read/Write	

For example: To read the digital state of FIO5 simply read address 2005. The value will be 1 for input-high, and 0 for input-low.

Digital I/O State Bit Masks

Each of these is a single binary-encoded value representing the state of 8 bits of I/O. Each bit represents an I/O line. Does not configure direction. A read of an output returns the current logic level on the terminal, not necessarily the output state written. The upper 8-bits of these values are inhibits. The inhibit bits prevent the corresponding state bit from being modified.

Address	Name	Type	Read/Write	Default
2500	FIO_STATE	UINT16	Read/Write	
2501	EIO_STATE	UINT16	Read/Write	
2502	CIO_STATE	UINT16	Read/Write	
2503	MIO_STATE	UINT16	Read/Write	

For example: To read the digital state of all FIO lines in a bit mask, read FIO_STATE. The value will be something like 0b11111011 representing 1 for logic high, and 0 for logic low. FIO2 is currently logic low.

Digital I/O Direction Bit Masks

Each of these is a single binary-encoded value representing the direction of 8 bits of I/O. Each bit designates an I/O line. 0=Input and 1=Output. The upper 8-bits of this value are inhibits. The inhibit bits prevent the corresponding direction bit from being modified.

Address	Name	Type	Read/Write	Default
2600	FIO_DIRECTION	UINT16	Read/Write	
2601	EIO_DIRECTION	UINT16	Read/Write	
2602	CIO_DIRECTION	UINT16	Read/Write	
2603	MIO_DIRECTION	UINT16	Read/Write	

For example: To set FIO1-7 to output, write a value of 0x01FF to FIO_DIRECTION. FIO0 is the least significant bit, so to prevent modification the corresponding inhibit bit is set with 0x01 in the most significant byte. The least significant byte is 0xFF, which is all 8 bits of FIO set to output.

Electrical Overview

All digital I/O on the T7 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 at least 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).

See electrical specifications for more details.

By default, the DIO lines are digital I/O, but they can also be configured as PWM Output, Quadrature Input, Counters, etc (see Extended Feature section of this Datasheet).

Power-up Defaults

The default condition of the digital I/O can be configured by the user. From the factory, all digital I/O are configured as inputs by default. Note that even if the default for a line is changed to output-high or output-low, there could be a small time (milliseconds) during boot-up where all digital I/O are in the factory default condition.

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.

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).

Increase logic level to 5V

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. When the line is set to input, the voltage on the line is determined by a pull-up resistor. The T7 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.

The 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: DAC0, DAC1 channels on the T7 can be set to 5 volts, providing 2 output lines with such capability.

14.1 DIO Extended Features Overview

Digital extended features measure and generate digital waveforms. Almost every digital I/O line can be assigned a featureand many can be active simultaneously. Features include things like PWM, Quadrature, and pulse generation. Features are assigned to DIOs using their type index, and configured using the options, and value registers.

The table below lists the features available on each DIO. The Digital I/O of the LabJack are on the top of the table, with the features to the left.

		FIO (0-7)								EIO (0-7)								CIO (0-3)				MIO (0-2)		
		DIO																						
Feature Type	**Type#**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**
PWM Out	**0**	✔		✔	✔	✔	✔																	
PWM Out with Phase	**1**	✔		✔	✔	✔	✔																	
Pulse Out	**2**	✔		✔	✔	✔	✔																	
Frequency In	**3,4**	✔	✔																					
Pulse Width In	**5**	✔	✔																					
Line-to-Line In	**6**	✔	✔																					
High-Speed Counter	**7**																	✔	✔	✔	✔			
Interrupt Counter	**8**	✔	✔	✔	✔			✔	✔															
Interrupt Counter with Debounce	**9**	✔	✔	✔	✔			✔	✔															
Quadrature In	**10**	✔	✔	✔	✔			✔	✔															
Interrupt Frequency In	11	✔	✔	✔	✔			✔	✔															

Table 2.9.2-1 T7 DIO Features

- **PWM Out** - Produces a rectangular output with variable frequency and variable duty cycle.
- **PWM Out with Phase** - Allows a phase difference between multiple PWM outputs.
- **Pulse Out** - You can specify the number of pulses, frequency of pulses, and pulse-width.
- **Frequency In** - Measures the period/frequency.
- **Pulse Width In** - Measures the high and low time, and thus also measures duty cycle.
- **Line-to-Line In** - Measures the time between edges on 2 different lines.
- **High-Speed Counter** - Hardware-based edge counter.
- **Interrupt Counter** - A hardware edge counter that must service an interrupt for each edge.
- **Interrupt Counter with Debounce** - Use to avoid counting bounces from mechanical switches.
- **Quadrature In** - Tracks the forward/reverse count provided by a quadrature signal.

Each digital IO has a set of registers dedicated to the configuration of and results produced by the Extended Features. These registers are used to perform four operations on the Extended Feature: Configure, Read, Update, and Reset. Below you will find general descriptions of the four operations. Details are provided in the Feature Type sections.

Configure:
Configuration is the initial setup of the Extended Feature. Configuration requires that any EF running at the pin in question first be disabled. Options can then be loaded. Then the EF can be enabled. The following seven registers are used for configuration:

DIO#_EF_ENABLE – 0 = Disable, 1 = Enable
DIO#_EF_TYPE – Index number specifying the Feature Type
DIO#_EF_OPTIONS – Bits 2-0: Specifies the clock source to use
DIO#_EF_VALUE_A – Type specific value
DIO#_EF_VALUE_B – Type specific value
DIO#_EF_VALUE_C – Type specific value
DIO#_EF_VALUE_D – Type specific value

Read:
Some Feature Types produce results or provide status information that can be read. This information is usually a binary integer. When possible, the LabJack will convert the binary integer into a real-world unit such as seconds. When available converted values can be read from the registers designated with "_F." The following registers are used to read results from a DIO Extended Feature:

DIO#_EF_READ_A – Type specific value. Reading this value takes a snapshot of READ_B.
DIO#_EF_READ_B – Type specific value. Reading this returns the snapshot acquired by READ_A.
DIO#_EF_READ_A_F – Returns READ_A converted to a real-world value and takes a snapshot of READ_B.
DIO#_EF_READ_B_F – Returns the READ_B snapshot converted to a real-world value.

Update:
Some Feature Types can be updated while running. Updating allows the Extended Feature to change it's operation parameters without restarting. Note that the ClockSource and Feature Type can not be changed in an update. When a value written takes effect depends on the Feature Type, please see the Update section in the Type description sections for more information. The following four registers can be used to update an running Extended Feature:

DIO#_EF_VALUE_A – Type specific value
DIO#_EF_VALUE_B – Type specific value
DIO#_EF_VALUE_C – Type specific value
DIO#_EF_VALUE_D – Type specific value

Reset:
Some Feature Types can be reset while they are running. Resetting can have different results depending on the Feature Type. For instance counters are reset to zero. There is only one register associated with resetting:

DIO#_EF_READ_A_AND_RESET – Type specific value. Reading this resets the Extended Feature and takes a snapshot of READ_B so that it can be read as in the Read section. Values are read before the reset.
DIO#_EF_READ_A_F_AND_RESET – Returns the same information as DIO#_EF_READ_A_F. Reading this resets the Extended Feature and takes a snapshot of READ_B. Values are read before the reset.

Digital EF Channel Registers

Name	Start Address	Type	Access	Default
DIO#(0:21)_EF_ENABLE	44000	UINT32	R/W	
DIO#(0:21)_EF_TYPE	44100	UINT32	R/W	
DIO#(0:21)_EF_OPTIONS	44200	UINT32	R/W	
DIO#(0:21)_EF_VALUE_A	44300	UINT32	R/W	
DIO#(0:21)_EF_VALUE_B	44400	UINT32	R/W	
DIO#(0:21)_EF_VALUE_C	44500	UINT32	R/W	
DIO#(0:21)_EF_VALUE_D	44600	UINT32	R/W	
DIO#(0:21)_EF_READ_A	3000	UINT32	R	
DIO#(0:21)_EF_READ_A_AND_RESET	3100	UINT32	R	
DIO#(0:21)_EF_READ_B	3200	UINT32	R	

DIO#(0:21)_EF_ENABLE

Names	Addresses
DIO0_EF_ENABLE, DIO1_EF_ENABLE,	44000, 44002, 44004, 44006, 44008, 44010,

DIO2_EF_ENABLE, DIO3_EF_ENABLE, DIO4_EF_ENABLE, DIO5_EF_ENABLE, DIO6_EF_ENABLE, DIO7_EF_ENABLE, DIO8_EF_ENABLE, DIO9_EF_ENABLE, DIO10_EF_ENABLE, DIO11_EF_ENABLE, DIO12_EF_ENABLE, DIO13_EF_ENABLE, DIO14_EF_ENABLE, DIO15_EF_ENABLE, DIO16_EF_ENABLE, DIO17_EF_ENABLE, DIO18_EF_ENABLE, DIO19_EF_ENABLE, DIO20_EF_ENABLE, DIO21_EF_ENABLE Show All	44012, 44014, 44016, 44018, 44020, 44022, 44024, 44026, 44028, 44030, 44032, 44034, 44036, 44038, 44040, 44042 Show All

DIO#(0:21)_EF_TYPE

Names	Addresses
DIO0_EF_TYPE, DIO1_EF_TYPE, DIO2_EF_TYPE, DIO3_EF_TYPE, DIO4_EF_TYPE, DIO5_EF_TYPE, DIO6_EF_TYPE, DIO7_EF_TYPE, DIO8_EF_TYPE, DIO9_EF_TYPE, DIO10_EF_TYPE, DIO11_EF_TYPE, DIO12_EF_TYPE, DIO13_EF_TYPE, DIO14_EF_TYPE, DIO15_EF_TYPE, DIO16_EF_TYPE, DIO17_EF_TYPE, DIO18_EF_TYPE, DIO19_EF_TYPE, DIO20_EF_TYPE, DIO21_EF_TYPE Show All	44100, 44102, 44104, 44106, 44108, 44110, 44112, 44114, 44116, 44118, 44120, 44122, 44124, 44126, 44128, 44130, 44132, 44134, 44136, 44138, 44140, 44142 Show All

DIO#(0:21)_EF_OPTIONS

Names	Addresses
DIO0_EF_OPTIONS, DIO1_EF_OPTIONS, DIO2_EF_OPTIONS, DIO3_EF_OPTIONS, DIO4_EF_OPTIONS, DIO5_EF_OPTIONS, DIO6_EF_OPTIONS, DIO7_EF_OPTIONS, DIO8_EF_OPTIONS, DIO9_EF_OPTIONS, DIO10_EF_OPTIONS, DIO11_EF_OPTIONS, DIO12_EF_OPTIONS, DIO13_EF_OPTIONS, DIO14_EF_OPTIONS, DIO15_EF_OPTIONS, DIO16_EF_OPTIONS, DIO17_EF_OPTIONS, DIO18_EF_OPTIONS, DIO19_EF_OPTIONS, DIO20_EF_OPTIONS, DIO21_EF_OPTIONS Show All	44200, 44202, 44204, 44206, 44208, 44210, 44212, 44214, 44216, 44218, 44220, 44222, 44224, 44226, 44228, 44230, 44232, 44234, 44236, 44238, 44240, 44242 Show All

DIO#(0:21)_EF_VALUE_A

Names	Addresses
DIO0_EF_VALUE_A, DIO1_EF_VALUE_A, DIO2_EF_VALUE_A, DIO3_EF_VALUE_A, DIO4_EF_VALUE_A, DIO5_EF_VALUE_A, DIO6_EF_VALUE_A, DIO7_EF_VALUE_A, DIO8_EF_VALUE_A, DIO9_EF_VALUE_A, DIO10_EF_VALUE_A, DIO11_EF_VALUE_A, DIO12_EF_VALUE_A, DIO13_EF_VALUE_A, DIO14_EF_VALUE_A, DIO15_EF_VALUE_A, DIO16_EF_VALUE_A, DIO17_EF_VALUE_A, DIO18_EF_VALUE_A, DIO19_EF_VALUE_A, DIO20_EF_VALUE_A, DIO21_EF_VALUE_A Show All	44300, 44302, 44304, 44306, 44308, 44310, 44312, 44314, 44316, 44318, 44320, 44322, 44324, 44326, 44328, 44330, 44332, 44334, 44336, 44338, 44340, 44342 Show All

DIO#(0:21)_EF_VALUE_B

Names	Addresses
DIO0_EF_VALUE_B, DIO1_EF_VALUE_B, DIO2_EF_VALUE_B, DIO3_EF_VALUE_B, DIO4_EF_VALUE_B, DIO5_EF_VALUE_B, DIO6_EF_VALUE_B, DIO7_EF_VALUE_B, DIO8_EF_VALUE_B, DIO9_EF_VALUE_B, DIO10_EF_VALUE_B, DIO11_EF_VALUE_B, DIO12_EF_VALUE_B, DIO13_EF_VALUE_B, DIO14_EF_VALUE_B, DIO15_EF_VALUE_B, DIO16_EF_VALUE_B, DIO17_EF_VALUE_B, DIO18_EF_VALUE_B, DIO19_EF_VALUE_B, DIO20_EF_VALUE_B, DIO21_EF_VALUE_B Show All	44400, 44402, 44404, 44406, 44408, 44410, 44412, 44414, 44416, 44418, 44420, 44422, 44424, 44426, 44428, 44430, 44432, 44434, 44436, 44438, 44440, 44442 Show All

DIO#(0:21)_EF_VALUE_C

Names	Addresses
DIO0_EF_VALUE_C, DIO1_EF_VALUE_C, DIO2_EF_VALUE_C, DIO3_EF_VALUE_C, DIO4_EF_VALUE_C, DIO5_EF_VALUE_C, DIO6_EF_VALUE_C, DIO7_EF_VALUE_C, DIO8_EF_VALUE_C, DIO9_EF_VALUE_C, DIO10_EF_VALUE_C, DIO11_EF_VALUE_C, DIO12_EF_VALUE_C, DIO13_EF_VALUE_C,	44500, 44502, 44504, 44506, 44508, 44510, 44512, 44514, 44516, 44518, 44520, 44522, 44524, 44526, 44528, 44530, 44532, 44534, 44536, 44538, 44540, 44542 Show All

DIO14_EF_VALUE_C, DIO15_EF_VALUE_C, DIO16_EF_VALUE_C, DIO17_EF_VALUE_C, DIO18_EF_VALUE_C, DIO19_EF_VALUE_C, DIO20_EF_VALUE_C, DIO21_EF_VALUE_C Show All

DIO#(0:21)_EF_VALUE_D

Names	Addresses
DIO0_EF_VALUE_D, DIO1_EF_VALUE_D, DIO2_EF_VALUE_D, DIO3_EF_VALUE_D, DIO4_EF_VALUE_D, DIO5_EF_VALUE_D, DIO6_EF_VALUE_D, DIO7_EF_VALUE_D, DIO8_EF_VALUE_D, DIO9_EF_VALUE_D, DIO10_EF_VALUE_D, DIO11_EF_VALUE_D, DIO12_EF_VALUE_D, DIO13_EF_VALUE_D, DIO14_EF_VALUE_D, DIO15_EF_VALUE_D, DIO16_EF_VALUE_D, DIO17_EF_VALUE_D, DIO18_EF_VALUE_D, DIO19_EF_VALUE_D, DIO20_EF_VALUE_D, DIO21_EF_VALUE_D Show All	44600, 44602, 44604, 44606, 44608, 44610, 44612, 44614, 44616, 44618, 44620, 44622, 44624, 44626, 44628, 44630, 44632, 44634, 44636, 44638, 44640, 44642 Show All

DIO#(0:21)_EF_READ_A

Names	Addresses
DIO0_EF_READ_A, DIO1_EF_READ_A, DIO2_EF_READ_A, DIO3_EF_READ_A, DIO4_EF_READ_A, DIO5_EF_READ_A, DIO6_EF_READ_A, DIO7_EF_READ_A, DIO8_EF_READ_A, DIO9_EF_READ_A, DIO10_EF_READ_A, DIO11_EF_READ_A, DIO12_EF_READ_A, DIO13_EF_READ_A, DIO14_EF_READ_A, DIO15_EF_READ_A, DIO16_EF_READ_A, DIO17_EF_READ_A, DIO18_EF_READ_A, DIO19_EF_READ_A, DIO20_EF_READ_A, DIO21_EF_READ_A Show All	3000, 3002, 3004, 3006, 3008, 3010, 3012, 3014, 3016, 3018, 3020, 3022, 3024, 3026, 3028, 3030, 3032, 3034, 3036, 3038, 3040, 3042 Show All

DIO#(0:21)_EF_READ_A_AND_RESET

Names	Addresses
DIO0_EF_READ_A_AND_RESET, DIO1_EF_READ_A_AND_RESET, DIO2_EF_READ_A_AND_RESET, DIO3_EF_READ_A_AND_RESET, DIO4_EF_READ_A_AND_RESET, DIO5_EF_READ_A_AND_RESET, DIO6_EF_READ_A_AND_RESET, DIO7_EF_READ_A_AND_RESET, DIO8_EF_READ_A_AND_RESET, DIO9_EF_READ_A_AND_RESET, DIO10_EF_READ_A_AND_RESET, DIO11_EF_READ_A_AND_RESET, DIO12_EF_READ_A_AND_RESET, DIO13_EF_READ_A_AND_RESET, DIO14_EF_READ_A_AND_RESET, DIO15_EF_READ_A_AND_RESET, DIO16_EF_READ_A_AND_RESET, DIO17_EF_READ_A_AND_RESET, DIO18_EF_READ_A_AND_RESET, DIO19_EF_READ_A_AND_RESET, DIO20_EF_READ_A_AND_RESET, DIO21_EF_READ_A_AND_RESET Show All	3100, 3102, 3104, 3106, 3108, 3110, 3112, 3114, 3116, 3118, 3120, 3122, 3124, 3126, 3128, 3130, 3132, 3134, 3136, 3138, 3140, 3142 Show All

DIO#(0:21)_EF_READ_B

Names	Addresses
DIO0_EF_READ_B, DIO1_EF_READ_B, DIO2_EF_READ_B, DIO3_EF_READ_B, DIO4_EF_READ_B, DIO5_EF_READ_B, DIO6_EF_READ_B, DIO7_EF_READ_B, DIO8_EF_READ_B, DIO9_EF_READ_B, DIO10_EF_READ_B, DIO11_EF_READ_B, DIO12_EF_READ_B, DIO13_EF_READ_B, DIO14_EF_READ_B, DIO15_EF_READ_B, DIO16_EF_READ_B, DIO17_EF_READ_B, DIO18_EF_READ_B, DIO19_EF_READ_B, DIO20_EF_READ_B, DIO21_EF_READ_B Show All	3200, 3202, 3204, 3206, 3208, 3210, 3212, 3214, 3216, 3218, 3220, 3222, 3224, 3226, 3228, 3230, 3232, 3234, 3236, 3238, 3240, 3242 Show All

How-To: Use a digital extended feature

1. Disable features on the DIO using ..._EF_ENABLE
2. Select a feature, and assign the corresponding type index to ..._EF_TYPE
3. Write to ..._EF_OPTIONS (if necessary)
4. Write to ..._EF_VALUE_A, ..._EF_VALUE_B, ..._EF_VALUE_C, ..._EF_VALUE_D (if necessary)
5. Enable feature on the DIO using ..._EF_ENABLE
6. Read results using ..._EF_READ_A, ..._EF_READ_B, or ..._EF_READ_A_AND_RESET

14.2 DIO EF - Clock Source

The ClockSources produce the reference frequencies used to generate output waveforms and measure input waveforms. ClockSource settings control output frequency, PWM resolution, maximum measurable period, and measurement resolution. The frequency of output modes will be the same as Clock#Frequency. PWM resolution will have a number of steps equal to RollValue. The maximum measurable period is RollValue * Clock#TickSeconds. The measurement resolution is Clock#TickSeconds. Clock# Frequency is the number of times per second that the clock source will roll from its maximum value back to zero. Clock#TickSeconds is the amount of time in seconds that it takes the clock source to increment its count by one.

Clock#Frequency = CoreFrequency / DIO_EF_CLOCK#_DIVISOR (typically 80M/Divisor)
Clock#TickSeconds = Divisor / CoreFrequency

There are 3 DIO EF clock sources available. Each clock source has an associated bit size and several mutual exclusions. Mutual exclusions exist because the clock sources share hardware with other features. A ClockSource is created form a hardware counter. CLOCK1 uses COUNTER_A (CIO0) and CLOCK2 uses COUNTER_B (CIO1). The 32-bit clock source (CLOCK0) is created by combining the 2 16-bit clock sources (CLOCK1 CLOCK2). The following list provides ClockSource bit sizes and mutual exclusions.

CLOCK0: 32-bit. Mutual Exclustions: CLOCK1, CLOCK2, COUNTER_A (CIO0), COUNTER_B(CIO1)
CLOCK1: 16-bit. Mutual Exclustions: CLOCK0, COUNTER_A (CIO0)
CLOCK2: 16-bit. Mutual Exclustions: CLOCK0, COUNTER_B (CIO1)

The clock source is not a DIO EF feature type, but the four basic operations of Configure, Read, Update, and Reset still apply:

Configure:
There are four registers associated with the configuration of clock sources:
DIO_EF_CLOCK#_ENABLE: 1 = Enable, 0 = Disable. Must be disabled to change the configuration.
DIO_EF_CLOCK#_DIVISOR: 1, 2, 4, 8, 16, 32, 64, or 256 (if this value is zero the divisor will be set to 1).
DIO_EF_CLOCK#_OPTIONS: Reserved for future use. Write 0.
DIO_EF_CLOCK#_ROLL_VALUE: The ClockSource will count to this value -1 then reset to zero and repeat.

A ClockSource can be enabled after DIO EF types have been configured. This allows several DIO EFs to be started at the same time.

Read:
DIO_EF_CLOCK#_COUNT: Returns the current value of a clock source's counter. This can useful for generating timestamps.

Update:
At this time there are no update operations available for the DIO EF clock sources. A clock source must be disabled to change any settings.

Reset:
At this time there are no reset operations available for the DIO EF clock sources.

Example:
Configure CLOCK0 as a 10 MHz clock source with a roll-value of 1000000. That means that PWM output would have a frequency of 10 Hz, and frequency input measurement will be able to count from 0-999999 with each count equal to 0.1 microseconds.

DIO_EF_CLOCK0_ENABLE = 0
DIO_EF_CLOCK0_DIVISOR = 8
DIO_EF_CLOCK0_ROLL_VALUE = 1000000
DIO_EF_CLOCK0_ENABLE = 1

Advanced:
If CLOCK0 is enabled and CLOCK1 and CLOCK2 are disabled, you can still select CLOCK1 or CLOCK2 as the source for a DIO EF channel. CLOCK1 CLOCK2 are actually the LSW & MSW of CLOCK0. The frequency of CLOCK1 is the same as CLOCK0. If DIO_EF_CLOCK0_ROLL_VALUE is >= 2^16, then the frequency of CLOCK2 is CLOCK0_freq divided by the modulus (remainder portion) of CLOCK0_freq / 2^16. If (CLOCK0_ROLL_VALUE - 1) is < 2^16, then the frequency of CLOCK2 is 0. CLOCK1_ROLL_VALUE is the modulus of (CLOCK0_ROLL_VALUE - 1) / 2^16 and CLOCK2_ROLL_VALUE is the quotient (integer portion) of (CLOCK0_ROLL_VALUE - 1) / 2^16.

Digital EF Clock Source Registers

Name	Start Address	Type	Access	Default
DIO_EF_CLOCK0_ENABLE	44900	UINT16	R/W	
DIO_EF_CLOCK0_DIVISOR	44901	UINT16	R/W	
DIO_EF_CLOCK0_OPTIONS	44902	UINT32	R/W	
DIO_EF_CLOCK0_ROLL_VALUE	44904	UINT32	R/W	
DIO_EF_CLOCK0_COUNT	44908	UINT32	R	

DIO_EF_CLOCK0_ENABLE

DIO_EF_CLOCK0_DIVISOR

DIO_EF_CLOCK0_OPTIONS

DIO_EF_CLOCK0_ROLL_VALUE

DIO_EF_CLOCK0_COUNT
Current tick count of this clock. Will read between 0 and ROLL_VALUE-1.

14.2 EF Clock Source

14.3 PWM Out

Capable DIO: **FIO0, FIO2, FIO3, FIO4, and FIO5**

Requires Clock Source: Yes

Type Index: **0**

PWM Out will generate a pulse width modulated wave form.

Operation:
PWM output will set the DIO high and low relative to the ClockSource's count. When the the count is set to zero the DIO line will be set high. When the count matches Value A the line will be set low. Therefore Value A is used to control the duty cycle, the frequency is determined by the selected ClockSource's PulseFrequency and the resolution is determined by RollValue.

PWMFrequency = PulseFrequency
*DutyCycle% = 100 * DIO#_EF_VALUE_A / DIO_EF_CLOCK#_ROLL_VALUE*

PWM Out is capable of glitch-free updates in most situations. A glitch-free update means that the PWM will finish the current period consisting of the high time then the low time before loading the new value. The next period will then have the new duty cycle. This is true for all cases except zero. When setting the duty cycle to zero the line will be set low regardless of the current position. This means that a single high pulse with duration between zero and the previous high time can be output before the line goes low.

Configure:
DIO#_EF_ENABLE: 0 = Disable, 1 = Enable
DIO#_EF_TYPE: 0
DIO#_EF_OPTIONS: Bits 0-2 specify which ClockSource to use. All other bits reserved (write 0).
DIO#_EF_VALUE_A: When the specified ClockSource's count matches this value the line will transition from high to low.
DIO#_EF_VALUE_B: Not used.
DIO#_EF_VALUE_C: Not used.
DIO#_EF_VALUE_D: Not used.

Update:
The duty cycle can be updated at any time. To update, write the new value to Value_A. The new value will not be used until the ClockSource rolls to zero. This means that at the end of the current period the new value will be loaded resulting in a glitch-free transition.

DIO#_EF_VALUE_A: Values written here will set the new duty cycle. The new value will not take effect until the selected ClockSource rolls to zero.

Read:
No information is returned by this feature type.

Reset:
Reset has no affect on this feature type.

Example:
Generate a 10 kHz PWM starting at 25% DC.
First configure the clock source. The PulseFrequency needs to be 10 kHz and the higher the roll value the greater the duty cycle resolution will be. To maximize the roll value use the smallest divisor that will not result in a RollValue greater than the ClockSource's maximum. With a divisor of 1 the roll value will be 8000: 80 MHz / (1 * 8000) = 10 kHz. Now set the registers accordingly:

DIO_EF_CLOCK0_ENABLE = 0
DIO_EF_CLOCK0_DIVISOR = 1
DIO_EF_CLOCK0_ROLL_VALUE = 8000
DIO_EF_CLOCK0_ENABLE = 1

Once the clock source is configured we can use the roll value to calculated Value_A: DC = 25% = 100 * Value_A / 8000. So Value_A = 2000. Now the PWM can be turned on by writing the proper registers:

DIO0_EF_TYPE = 0
DIO0_EF_OPTIONS = 0
DIO0_EF_VALUE_A = 2000
DIO0_EF_ENABLE = 1

14.4 PWM Out with Phase

Capable DIO: **FIO0, FIO2, FIO3, FIO4, and FIO5**

Requires Clock Source: Yes

Type Index: **1**

PWM Output with phase control generates PWM wave-forms with the pulse positioned at different points in the period. This is achieved by setting the DIO line high and low relative to the ClockSource's count.

PWMFrequency = PulseFrequency

DC = (Value_A - Value_B) / DIO_EF_CLOCK#_ROLL_VALUE
*PhaseOffset = 360º * Value_A / DIO_EF_CLOCK#_ROLL_VALUE*

When the the count matches Value_B the DIO line will be set high. When the count matches Value A the line will be set low. Therefore Value_A minus Value_B controls the duty cycle.

Configure:
DIO#_EF_ENABLE: 0 = Disable, 1 = Enable
DIO#_EF_TYPE: 1
DIO#_EF_OPTIONS: Bits 0-2 specify which ClockSource to use. All other bits reserved (write 0).
DIO#_EF_VALUE_A: When the ClockSource's count matches this value the line will transition from high to low.
DIO#_EF_VALUE_B:When the ClockSource's count matches this value the line will transition from low to high.
DIO#_EF_VALUE_B: Not used.
DIO#_EF_VALUE_B: Not used.

Update:
The duty cycle can be updated at any time. To update, write the new value to Value_A then Value_B. The value written to Value_A is stored until Value_B is written. After writing Value_B the new value will be loaded at the start of the next period. Updates are glitch-less unless switching from a very high to very low duty cycle or a very low to very high duty cycle.

DIO#_EF_VALUE_A: Values written here will set the new falling position. The new value will not take effect until Value_B is written.
DIO#_EF_VALUE_B: Values written here will set the new rising position. When Value_B is written the new Value_A is also loaded.

Read:
No information is returned by this feature type.

Reset:
Reset has no affect on this feature type.

14.5 Pulse Out

Capable DIO: **FIO0, FIO2, FIO3, FIO4, and FIO5**

Requires Clock Source: Yes

Type Index: **2**

Pulse output will generate a specified number of pulses. The high time and the low time are specified relative to the ClockSource the same way as PWM with Phase Control.

DC = (Value A - Value B) / DIO_EF_CLOCK#_ROLL_VALUE
Frequency = PulseFrequency

Configure:
First set the DIO line low. The line must start low for proper pulse generation.
DIO#_EF_ENABLE: 0 = Disable, 1 = Enable
DIO#_EF_TYPE: 2
DIO#_EF_Options: Bits 0-2 specify which ClockSource to use. All other bits reserved (write 0).
DIO#_EF_Value A: When the specified ClockSource's count matches this value the line will transition from high to low.
DIO#_EF_Value B: When the specified ClockSource's count matches this value the line will transition from low to high.
DIO#_EF_Value C: The number of pulses to generate.
DIO#_EF_Value D: Not used.

Update:
DIO#_EF_Value A: Sets a new high to low transition point. Will take effect when writing Value C.
DIO#_EF_Value B: Sets a new low to high transition point. Will take effect when writing Value C.

DIO#_EF_Value C: Writing to this value will start a new pulse sequence. If a sequence is already in progress it will be aborted. Numbers previously written to Value A or Value B will take effect when Value C is written.

Read:
DIO#_EF_READ_A: The number of pulses that have been completed.
DIO#_EF_READ_B: The target number of pulses.
DIO#_EF_READ_C: Not used.
DIO#_EF_READ_D: Not used.

Reset:
DIO#_EF_READ_A_AND_RESET: Reads number of pulses that have been completed. Then restarts the pulse sequence.

Example:
First configure a ClockSource to drive the pulse generator. Assuming the ore frequency is 80 MHz writing following registers will produce a 1 kHz frequency.
DIO_EF_CLOCK0_DIVISOR = 8
DIO_EF_CLOCK0_ROLL_VALUE = 10000
DIO_EF_CLOCK0_ENABLE = 1

80 MHz / (8 * 10000) = 1 kHz

The ClockSource is now counting from 0-9999 at 10 MHz. So it will roll over 1000 times per second; generating our 1 kHz frequency. Now that we have a clock to work with we can configure our pulse.

DIO0 = 0
DIO0_EF_TYPE = 2
DIO0_EF_OPTIONS = 0
DIO0_EF_VALUE_A = 2000
DIO0_EF_VALUE_B = 5000
DIO0_EF_VALUE_C = 3
DIO0_EF_ENABLE = 1

The LabJack will now output 3 pulses, 30% duty cycle.

14.6 Frequency In

Capable DIO: **FIO0, FIO1**

Requires Clock Source: Yes

Type Index: **3 (positive edges) or 4 (negative edges)**

Frequency In will measure a period by counting the number of ClockSource ticks between two edges. From one rising edge to the next or from one falling edge to the next. The number of ticks can be read from Value_A. The following formula will produce period in seconds.

*Period (s) = Value_A / (ClockSourcePulseRate * DIO_EF_CLOCK#_ROLL_VALUE)*

The maximum measurable period is controlled by the SourceClock PulseFrequency. The measurement resolution is controlled by the ClockSourceFrequency and RollValue.

The period is continuously measured. The most recent result can be read from READ_A.

Configure:
DIO#_EF_ENABLE: 0 = Disable, 1 = Enable
DIO#_EF_TYPE: 3
DIO#_EF_OPTIONS: Bits 0-2 specify which ClockSource to use. All other bits reserved (write 0).
DIO#_EF_VALUE_A: Not used.
DIO#_EF_VALUE_B: Not used.

DIO#_EF_VALUE_C: Not used.
DIO#_EF_VALUE_D: Not used.

Update:
No update operations can be performed on this feature type.

Read:
DIO#_EF_READ_A: Returns the measured time in ClockSource ticks from one edge to another. If a full period has not yet been observed this value will be zero.
DIO#_EF_READ_B: Not used by this feature type.
DIO#_EF_READ_A_F: Returns the length of the period in seconds. If a full period has not yet been observed this value will be zero.

Reset:
DIO#_EF_READ_A_AND_RESET: Returns the same data as DIO#_EF_READ_A and then clears the result so that zero is returned by subsequent reads until another full period is measured.

Example:
DIO_EF_CLOCK0_DIVISOR = 8
DIO_EF_CLOCK0_ROLL_VALUE = 0
DIO_EF_CLOCK0_ENABLE = 1
DIO0_EF_TYPE = 3
DIO0_EF_ENABLE = 1

14.7 Pulse Width In

Capable DIO: **FIO0, FIO1**

Requires Clock Source: Yes

Type Index: **5**

Pulse width in will measure the high time and low time of a periodic signal. The maximum measurable period is controlled by the SourceClock's PulseFrequency. The measurement resolution is controlled by the ClockSourceFrequency and RollValue.

Configure:
DIO#_EF_ENABLE: 0 = Disable, 1 = Enable
DIO#_EF_TYPE: 5
DIO#_EF_OPTIONS: Bits 0-2 specify which ClockSource to use. All other bits reserved (write 0).
DIO#_EF_VALUE_A: Not used.
DIO#_EF_VALUE_B: Not used.
DIO#_EF_VALUE_C: Not used.
DIO#_EF_VALUE_D: Not used.

Update:
No update operations can be performed on this feature type.

Read:
DIO#_EF_READ_A: Returns the measured high time in ClockSource ticks and saves the low time so that it can be read later. If a full period has not yet been observed this value will be zero.
DIO#_EF_READ_B: Returns the measured low time in ClockSource ticks. This is the value saved when READ_A was read.
DIO#_EF_READ_A_F: Returns the measured high time in seconds and saves the low time so that it can be read later. If a full period has not yet been observed this value will be zero.
DIO#_EF_READ_B_F: Returns the measured low time in seconds. This is the value saved when READ_A was read.

Reset:

DIO#_EF_READ_A_AND_RESET: Performs the same operation as DIO#_EF_READ_A and then clears the result so that zeros are returned until another full period is measured.

14.8 Line-to-Line In

Capable DIO: **FIO0, FIO1**

Requires Clock Source: Yes

Type Index: **6**

Line-to-Line In measures the time between an edge on one DIO line to an edge on another DIO line. The edges can be individually specified as rising or falling. The maximum measurable period is controlled by the selected ClockSource's PulseFrequency. The resolution is controlled by Clock#Frequency DIO_EF_CLOCK#_ROLL_VALUE.

MaximumMeasurablePeriod = 1 / PulseFrequency
*Resolution = 1 / (PulseFrequency * DIO_EF_CLOCK#_ROLL_VALUE)*

Edge compare operates in a one-shot mode. Once the specified combination of edges is observed the data is saved and measuring stops. Another measurement can be started by resetting or performing the configuration procedure again.

Configure:
Configuring an edge compare mode requires configuring two DIO extended features. The first configured should be the one expecting the first edge. Any extended features on either DIO should be disabled before beginning configuration.
DIO#_EF_ENABLE: 0 = Disable, 1 = Enable
DIO#_EF_TYPE: 6
DIO#_EF_OPTIONS: Bits 0-2 specify which ClockSource to use. All other bits reserved (write 0).
DIO#_EF_VALUE_A: 0 = falling edge. 1 = rising edge.
DIO#_EF_VALUE_B: Not used.
DIO#_EF_VALUE_C: Not used.
DIO#_EF_VALUE_D: Not used.

Update:
No update operations ca be performed on this feature type.

Read:
DIO#_EF_READ_A: Returns the measured time in ClockSource ticks. If the specified combination of edges has not yet been observed this value will be zero.
DIO#_EF_READ_B: Not used by this feature type.
DIO#_EF_READ_A_F: Returns the timer between edges in seconds.

Reset:
DIO#_EF_READ_A_AND_RESET: Performs the same operation as DIO#_EF_READ_A, then clears the result and starts another measurement.

14.9 High-Speed Counter

Capable DIO: **CIO0, CIO1, CIO2, CIO3**

Requires Clock Source: No

Type Index: **7**

The T7 supports up to 4 high-speed counters that use hardware to achieve high count rates. These counters are shared with other resources as follows:

CounterA (DIO16/CIO0): Used by EF Clock0 & Clock1.
CounterB (DIO17/CIO1): Used by EF Clock0 & Clock2.
CounterC (DIO18/CIO2): Always available.
CounterD (DIO19/CIO3): Used by stream mode.

Configuration:
DIO#_EF_ENABLE: 0 = Disable, 1 = Enable
DIO#_EF_TYPE: 7
DIO#_EF_OPTIONS: Bits 0-2 specify which ClockSource to use. All other bits reserved (write 0).
DIO#_EF_VALUE_A: When the ClockSource's count matches this value the line will transition from high to low.
DIO#_EF_VALUE_B:When the ClockSource's count matches this value the line will transition from low to high.
DIO#_EF_VALUE_B: Not used.
DIO#_EF_VALUE_B: Not used.

Update:
No update operations can be performed on this EF Type.

Read:
DIO#_EF_READ_A: Returns the current Count

Reset:
DIO#_EF_READ_A_AND_RESET: Reads the current count then clears the counter. Note that there is a brief period of time between reading and clearing during which edges can be missed. During normal operation this time period is 10-30us. If missed edges at this point can not be tollerated then reset should not be used.

14.10 Interrupt Counter

Capable DIO: **FIO0, FIO1, FIO2, FIO3, FIO6, and FIO7**

Requires Clock Source: No

Type Index: **8**

Interrupt Counter counts pulses on the associated IO line. This type of counter is not purely implemented in hardware. The firmware must service each edge. This makes it quite a bit slower than the pure hardware high-speed counter (Mode 7). Expect it to top out around TBD ~100kHz.

Configure:
DIO#_EF_ENABLE: 0 = Disable, 1 = Enable
DIO#_EF_TYPE: 8
DIO#_EF_OPTIONS: Not used.
DIO#_EF_VALUE_A: Not used.
DIO#_EF_VALUE_B: Not used.
DIO#_EF_VALUE_B: Not used.
DIO#_EF_VALUE_B: Not used.

Update:
No update operations can be performed on this EF Type.

Read:
DIO#_EF_READ_A: Returns the current Count

Reset:
DIO#_EF_READ_A_AND_RESET: Reads the current count then clears the counter. Note that there is a brief period of time between reading and clearing during which edges can be missed. During normal operation this time period is 10-30us. If missed edges at this point can not be tollerated then reset should not be used.

14.11 Interrupt Counter with Debounce

Capable DIO: **FIO0, FIO1, FIO2, FIO3, FIO6, and FIO7**

Requires Clock Source: No

Type Index: **9**

Interrupt Counter with Debounce will count when it receives the specified edge. After counting a timer is started. No received edges will be counted until the timer expires. This type of counter is not purely implemented in hardware. The firmware must service each edge. This makes it quite a bit slower than the pure hardware high-speed counter (Mode 7). Expect it to top out around TBD ~100kHz.

Configuration:
DIO#_EF_ENABLE: 0 = Disable, 1 = Enable
DIO#_EF_TYPE: 9
DIO#_EF_OPTIONS: Not used.
DIO#_EF_VALUE_A: Debounce time in microseconds (µs).
DIO#_EF_VALUE_B: bit 0: 1 = Count on Rising edges, 0 = falling edges, 2 = both edges.
DIO#_EF_VALUE_B: Not used.
DIO#_EF_VALUE_B: Not used.

Update:
No update operations can be performed on this EF Type.

Read:
DIO#_EF_READ_A: Returns the current Count

Reset:
DIO#_EF_READ_A_AND_RESET: Reads the current count then clears the counter. Note that there is a brief period of time between reading and clearing during which edges can be missed. During normal operation this time period is 10-30us. If missed edges at this point can not be tollerated then reset should not be used.

14.12 Quadrature In

Capable DIO: **FIO0, FIO1, FIO2, FIO3, FIO6, and FIO7**

Requires Clock Source: No

Type Index: **10**

Quadrature input uses two DIOs to measure a quadrature signal. Quadrature is a directional count often used in rotary encoders. This feature type uses 4x quadrature decoding. Meaning that every edge observed will increment or decrement the count. This feature type can be used if the expected frequency does not exceed the device wide edge rate limitation.

Quadrature is prone to error if the edge rate is exceeded. This is particularly likely during direction change where the time between edges can be very small. Error where two edges come in too quickly for the device to process can result in missed counts or missed change in direction. These errors will be recorded and the quantity encountered can be read. If three edges come in too quickly an undetectable error can occur.

Configure:
Configuring an edge compare mode requires configuring two DIO. The first configured will be considered the A line.

DIO#_EF_ENABLE: 0 = Disable, 1 = Enable
DIO#_EF_TYPE: 10
DIO#_EF_OPTIONS: Not used.
DIO#_EF_VALUE_A: Not used.
DIO#_EF_VALUE_B: Not used.
DIO#_EF_VALUE_C: Not used.
DIO#_EF_VALUE_D: Not used.

Update:
No update operations can be performed on this Feature Type.

Read:
DIO#_EF_READ_A - Returns the current count.
DIO#_EF_READ_B – Returns the number of detected errors.

Reset:
DIO#_EF_READ_A_AND_RESET – Performs the same operation as DIO#_EF_READ_A, then sets the count to zero.

Example:

14.13 Interrupt Frequency In

Capable DIO: **FIO0, FIO1, FIO2, FIO3, FIO6, FIO7**

Requires Clock Source: Yes

Type Index: **3 (positive edges) or 4 (negative edges)**

Interrupt Frequency In will measure the frequency of a signal on the associated DIO line. To measure the frequency the LabJack will measure the duration of one or more periods. There are several option available to control the way the LabJack does this. The number of period to be averaged, the edge direction to trigger on and whether to measure continuously or in a one-shot mode can all be specified.

TickTime = CoreFrequency / 2
*Frequency (Hz) = CoreFrequency / (2 * Value_A)*

The maximum measurable time is 107 s. The number of periods to be averaged times the maximum expected period must be less than 107 s or the result will overflow: 107 < (NumToAverage * MaxPeriod)

By default Interrupt Frequency In will measure the frequency once and return that same result until it is reconfigured or reset. At which point a second measurement will be made. The other option is continuous mode. In continuous mode the frequency is constantly being measured and read returns the most recent result. Running continuous puts a greater load on the processor.

Configure:
DIO#_EF_ENABLE: 0 = Disable, 1 = Enable
DIO#_EF_TYPE: 11
DIO#_EF_OPTIONS: Not Used.
DIO#_EF_VALUE_A: bit 1: Edge select; 1 = rising, 0 = falling. Bit 2: 1=continuous, 0=OneShot.
DIO#_EF_VALUE_B: Number of periods to be measured.
DIO#_EF_VALUE_C: Not used.
DIO#_EF_VALUE_D: Not used.

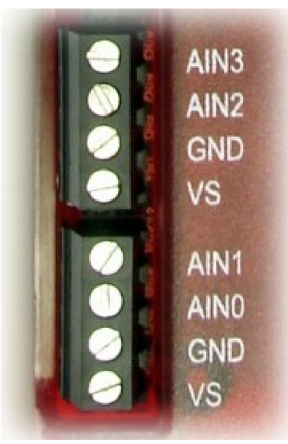
Update:
No update operations can be performed on this feature type.

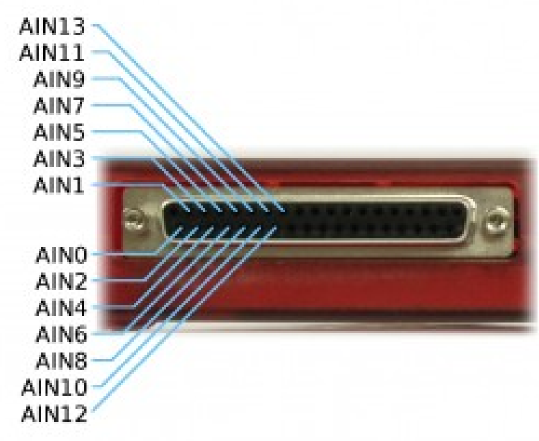
Read:
DIO#_EF_READ_A: Returns the measured time in ticks. This represents the total time elapsed during Value_A averaged periods. Until the specified number of periods has been observed this value will be zero.
DIO#_EF_READ_B: Not used by this feature type.
DIO#_EF_READ_A_F: Returns calculated frequency. Takes into account the number of periods to be averaged and the core clock speed.

Reset:
DIO#_EF_READ_A_AND_RESET: Returns the same data as DIO#_EF_READ_A and then clears the result so that zero is returned by subsequent reads until another full period is measured.
DIO#_EF_READ_A_AND_RESET_F: Returns the same data as DIO#_EF_READ_A_F and then clears the result so that zero is returned by subsequent reads until another full period is measured.

Example:

15.0 AIN





Analog Inputs: **14**

Voltage Ranges: **±10V, ±1V, ±0.1V, and ±0.01V**

Analog Input Settings

Name	Start Address	Type	Access	Default
AIN#(0..13)	0	FLOAT32	R	
AIN#(0..13)_RANGE	40000	FLOAT32	R/W	0
AIN#(0..13)_NEGATIVE_CH	41000	UINT16	R/W	199
AIN#(0..13)_RESOLUTION_INDEX	41500	UINT16	R/W	0

AIN#(0..13)
Returns the voltage of the specified analog input.

Names	Addresses
AIN0, AIN1, AIN2, AIN3, AIN4, AIN5, AIN6, AIN7, AIN8, AIN9, AIN10, AIN11, AIN12, AIN13 Show All	0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26 Show All

AIN#(0..13)_RANGE

The range/span of each analog input.

Names	Addresses
AIN0_RANGE, AIN1_RANGE, AIN2_RANGE, AIN3_RANGE, AIN4_RANGE, AIN5_RANGE, AIN6_RANGE, AIN7_RANGE, AIN8_RANGE, AIN9_RANGE, AIN10_RANGE, AIN11_RANGE, AIN12_RANGE, AIN13_RANGE Show All	40000, 40002, 40004, 40006, 40008, 40010, 40012, 40014, 40016, 40018, 40020, 40022, 40024, 40026 Show All

AIN#(0..13)_NEGATIVE_CH

Specifies the negative channel to be used for each positive channel. 199=Default=> Single-Ended.

Names	Addresses
AIN0_NEGATIVE_CH, AIN1_NEGATIVE_CH, AIN2_NEGATIVE_CH, AIN3_NEGATIVE_CH, AIN4_NEGATIVE_CH, AIN5_NEGATIVE_CH, AIN6_NEGATIVE_CH, AIN7_NEGATIVE_CH, AIN8_NEGATIVE_CH, AIN9_NEGATIVE_CH, AIN10_NEGATIVE_CH, AIN11_NEGATIVE_CH, AIN12_NEGATIVE_CH, AIN13_NEGATIVE_CH Show All	41000, 41001, 41002, 41003, 41004, 41005, 41006, 41007, 41008, 41009, 41010, 41011, 41012, 41013 Show All

AIN#(0..13)_RESOLUTION_INDEX

The resolution index for each analog input. A larger resolution index generally results in lower noise and longer sample times.

Names	Addresses
AIN0_RESOLUTION_INDEX, AIN1_RESOLUTION_INDEX, AIN2_RESOLUTION_INDEX, AIN3_RESOLUTION_INDEX, AIN4_RESOLUTION_INDEX, AIN5_RESOLUTION_INDEX, AIN6_RESOLUTION_INDEX, AIN7_RESOLUTION_INDEX, AIN8_RESOLUTION_INDEX, AIN9_RESOLUTION_INDEX, AIN10_RESOLUTION_INDEX, AIN11_RESOLUTION_INDEX, AIN12_RESOLUTION_INDEX, AIN13_RESOLUTION_INDEX Show All	41500, 41501, 41502, 41503, 41504, 41505, 41506, 41507, 41508, 41509, 41510, 41511, 41512, 41513 Show All

Some Examples

Analog Input Example: To read a voltage connected to AIN2, perform a standard read of modbus address 4, and the result would be in the form of a floating point number, like 8.82332V.

Range Example: It is known that the voltage source connected AIN1 will be 0 to 0.7V, so write 1.0 (or anything >0.1 and <=1.0) to address 40002, and the device will use the ±1V range.

Differential Analog Input Example: To perform differential readings between AIN2(pos) and AIN3(neg), change the negative channel of AIN2 to AIN3 by writing a value of 3 to address 41004.

Resolution Index Example: Change the AIN1 analog input to roughly 16-bit resolution by writing a value of 1 to address 41501. You may also use the name directly with "AIN1_RESOLUTION_INDEX".

Settling Example: Change the settling time of AIN3 to 500uS by writing a value of 500 to address 42006, although we recommend a value of 0, which corresponds to automatic settling.

Extra Details

The analog inputs are not artificially pulled to 0.0 volts, as that would reduce the input impedance, so readings obtained from floating channels will generally not be 0.0 volts. The readings from floating channels depend on adjacent channels and sample rate and have little meaning. See related floating input application note.

The **AIN#(0..13)_RANGE** parameter is actually controlling the gain of the internal instrumentation amplifier. The in-amp supports gains of x1, x10, x100, and x1000. If you set range=10, you get gain=x1, and the analog input range is ±10 volts. If you set range=1, you get gain=x10, and the analog input range is ±1 volts. Note that the device knows what the internal gain is set to and adjusts the return values to give the voltage at the input terminals, so if you connect a 0.8 volt signal to the input terminals, it will be amplified to 8.0 volts before being digitized, but the reading you get back will be 0.8 volts. Write range=10 to get a range of ±10V (default), range=1 to get a range of ±1V, range=0.1 to get a range of ±0.1V, or range=0.01 to get a range of ±0.01V. If you write a value in between the valid ranges, the larger range will be used.

The **AIN#(0:254)_NEGATIVE_CH** parameter pertains do differential readings. Differential channels are adjacent even/odd pairs only, such as AIN2-AIN3. Thus the positive channel must be even and the negative channel must be +1. For channel numbers in the extended range (above AIN15), see the Mux80 datasheet.

The **AIN#(0:254)_RESOLUTION_INDEX** parameter affects the ADC. A higher Resolution_Index results in lower noise and thus higher effective & noise-free resolution, with the tradeoff of longer sample times. The value passed for Resolution_Index is from 0-8, where 0 corresponds to default, 1 is roughly 16-bit resolution (RMS or effective), and 8 is roughly 19-bit resolution. The T7-Pro has additional Resolution_Index settings 9-12 that use the alternate high-resolution converter (24-bit sigma-delta) and correspond to roughly 19-bit to 22-bit resolution. For command-response readings, the default value of 0 corresponds to Resolution_Index=8 on a T7 and Resolution_Index=9 on a T7-Pro. For stream readings the default of 0 corresponds to Resolution_Index=1.

The **AIN#(0:254)_SETTLING_US** parameter is the time from a step change in the input signal to when the signal is sampled by the ADC, measured in microseconds. A step change in this case is caused when the internal multiplexers change from one channel to another. In general, more settling time is required as gain and resolution are increased. The default "auto" settling time ensures that the device meets specifications at any gain and resolution for source impedances up to at least 1000 ohms. In command/response mode, the effect of the settling is 0=Auto, 1=20us, 2=50us, 3=100us, 4=200us, 5=500us, 6=1ms, 7=2ms, 8=5ms, 9=10ms. Stream mode has its own settling parameter. The timings in Electrical Characteristics are measured with "auto" settling.

16.0 DAC

Output: **0V to 5V**

Resolution: **12-bit**

Source Impedance: **50 ohms**

DAC Registers

Name	Start Address	Type	Access	Default
DAC#(0:1)	1000	FLOAT32	R/W	

DAC#(0:1)
Pass a voltage for the specified analog output.

Names	Addresses
DAC0, DAC1	1000, 1002

Overview

There are two DACs (digital-to-analog converters or analog outputs) on the T7. Each DAC can be set to a voltage between about 0.02 and 5 volts with 12-bits of resolution.

For electrical specifications, See Appendix TBD.

Although the DAC values are based on an absolute reference voltage, and not the supply voltage, the DAC output buffers are powered internally by Vs and thus the maximum output is limited to slightly less than Vs.

The DACs appear both on the screw terminals and on the DB37 connector. These connections are electrically the same, and the user must exercise caution only to use one connection or the other, and not create a short circuit.

Power-up Defaults

The power-up condition of the DACs can be configured by the user. From the factory, the DACS default to enabled at minimum voltage (~0 volts). Note that even if the power-up default for a line is changed to a different voltage or disabled, there is a delay of about 100 ms at power-up where the DACs are in the factory default condition.

Protection

The analog outputs can withstand a continuous short-circuit to ground, even when set at maximum output.

Voltage should never be applied to the analog outputs, as they are voltage sources themselves. In the event that a voltage is accidentally applied to either analog output, they do have protection against transient events such as ESD (electrostatic discharge) and continuous overvoltage (or undervoltage) of a few volts.

Increase Output to ±10V

There is an accessory available from LabJack called the LJTick-DAC that provides a pair of 14-bit analog outputs with a range of ±10 volts. The LJTick-DAC plugs into any digital I/O block, and thus up to 10 of these can be used per T7 to add 20 analog outputs.

Calibration Constants

The T7 automatically returns calibrated readings, so most people should not concern themselves with this section.

If the factory applied calibration constants are of interest, they are stored on internal memory and can be accessed at any time through the use of the Modbus registers listed in the table below.

Registers: Flash Memory Access

Address	Name	Type	Read/Write	Default
61800	EXTERNAL_FLASH_KEY	UINT32	W	-
61810	pEXTERNAL_FLASH_READ	UINT32	R/W	-
61812	EXTERNAL_FLASH_READ	UINT32	R	-
61820	EXTERNAL_FLASH_ERASE	UINT32	W	-
61830	pEXTERNAL_FLASH_WRITE	UINT32	R/W	-
61832	EXTERNAL_FLASH_WRITE	UINT32	W	-

61810: Write the starting flash address. Each flash address points to 1 byte.

61812: Read 1-512 registers starting from this address to get the data. Flash is read in 32-bit chunks, so you must read an even number of registers. You can only read multiple registers starting from this Modbus address ... you can't read 61812, then read 61814, and so on. The number of registers you can read at once might be further limited by the maximum packet size of the particular interface ... if you don't want to worry about that just stick to 13 values (26 registers) or less per read.

For example: To read 8 floats out of memory, starting at external flash address 3948544, initialize the read pointer (Modbus address 61810) to a value of 3948544 using eWriteAddress(), then read Modbus addresses starting at address 61812 using eReadAddresses(). The read pointer (address 61810) does not automatically increment.

The cal constants begin at memory address 0x3C4000, or in decimal format 3948544. The structure(location) of each calibration value can be seen in the code snippet below.

Follow the above example to read out the first 8 values: PSlope, NSlope, Center, Offset (HS Gain= x1), and PSlope, NSlope, Center, Offset (HS Gain= x10).

```
typedef struct{
 float PSlope;
 float NSlope;
 float Center;
 float Offset;
}Cal_Set;

typedef struct{
 Cal_Set HS[4];
 Cal_Set HR[4];

 struct{
  float Slope;
  float Offset;
 }DAC[2];

 float Temp_Slope;
 float Temp_Offset;

 float ISource_10u;
 float ISource_200u;

 float I_Bias;
}Device_Calibration;
```

The full size of the calibration section is 164 bytes, or 41 floats.

The reason that there are 'Cal_Set's for each High Speed 'HS' and High Resolution 'HR', is that there are 2 analog converters on a T7-Pro. A standard T7 uses only the High Speed analog converter, so only the HS[4] calibration values will be populated with valid information. A T7-Pro will have calibration information for both high speed, and high resolution converters.

Additionally, there are distinct sets of positive slope(Pslope), negative slope(Nslope), Center, and Offset values for each of the 4 gain settings on the device.

High speed AIN calibration values **HS[4]**:
HS[0] = calibration for gain x1
HS[1] = calibration for gain x10
HS[2] = calibration for gain x100
HS[3] = calibration for gain x1000

High resolution (-Pro only) AIN calibration values **HR[4]**:
HR[0] = calibration for gain x1
HR[1] = calibration for gain x10
HR[2] = calibration for gain x100
HR[3] = calibration for gain x1000

Watchdog

The Watchdog system can set the outputs or reset the whole device, after a predefined time period. It is useful as a safe-guard against lost communication, or system damage. When enabled, the Watchdog timeout is reset every time a response to a command is transmitted. Thus, if communication fails, the device will automatically enter a fail-safe state as determined by the user in the Watchdog settings.

Basic Usage

The most basic way to use Watchdog is to set the **PinCfgStartup** bit, and leave the rest of the options disabled. When configured in this basic way, and communication fails, the device will respond in the same way as it does after a power failure. After a power failure the device will boot up with the default settings. The best way to modify the default settings is to configure the device as desired, then save those settings as the new defaults using IO_CONFIG_SET_DEFAULT_TO_CURRENT.

Watchdog Settings

Name	Start Address	Type	Access	Default
WATCHDOG_ENABLE_DEFAULT	61600	UINT32	R/W	
WATCHDOG_OPTIONS_DEFAULT	61602	UINT32	R/W	
WATCHDOG_TIMEOUT_S_DEFAULT	61604	UINT32	R/W	
WATCHDOG_STARTUP_DELAY_S_DEFAULT	61606	UINT32	R/W	
WATCHDOG_DIO_STATE_DEFAULT	61610	UINT32	R/W	
WATCHDOG_DIO_DIRECTION_DEFAULT	61612	UINT32	R/W	
WATCHDOG_DIO_INHIBIT_DEFAULT	61614	UINT32	R/W	
WATCHDOG_DAC0_DEFAULT	61616	FLOAT32	R/W	
WATCHDOG_DAC1_DEFAULT	61618	FLOAT32	R/W	
WATCHDOG_KEY_DEFAULT	61620	UINT32	W	
WATCHDOG_CLEAR	61640	UINT32	W	

WATCHDOG_ENABLE_DEFAULT
The current Enabled/Disabled state of the Watchdog system.

WATCHDOG_OPTIONS_DEFAULT
A single binary-encoded value where each bit is an option. See Watchdog section in datasheet.

WATCHDOG_TIMEOUT_S_DEFAULT
The number of seconds before timeout. On timeout, watchdog executes all operations defined in the options register. A clear event will reset the timeout.

WATCHDOG_STARTUP_DELAY_S_DEFAULT

Specify a startup delay after Watchdog timeout.

WATCHDOG_DIO_STATE_DEFAULT
The state high/low of the digital I/O after a Watchdog timeout. See DIO_STATE

WATCHDOG_DIO_DIRECTION_DEFAULT
The direction input/output of the digital I/O after a Watchdog timeout. See DIO_DIRECTION

WATCHDOG_DIO_INHIBIT_DEFAULT
The inhibit mask of the digital I/O after a Watchdog timeout. See DIO_INHIBIT

WATCHDOG_DAC0_DEFAULT
The voltage of DAC0 after a Watchdog timeout.

WATCHDOG_DAC1_DEFAULT
The voltage of DAC1 after a Watchdog timeout.

WATCHDOG_KEY_DEFAULT
When set to strict mode this is the value that much be written to the clear register.

WATCHDOG_CLEAR
When running in strict mode writing the key to this register is the only way to clear the watchdog.

The configuration options for the Watchdog system are stored in **WATCHDOG_OPTIONS_DEFAULT**. Each option is a bit stored in a single binary-encoded value. The table below explains the location of each option, and is followed by option descriptions.

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
-	-	-	-	-	-	PinCfgStartup	PinCfgRst	-	-	-	UseStrict	SetDAC1	SetDAC0	ApplyDIO	HardReset
bit15															bit0

Bit 15-10 **Unimplemented:** Read as '0'

bit 9 **PinCfgStartup:** Reset the pin configuration to user defaults
1 = Enable
0 = Disable

bit 8 **PinCfgRst:** Reset the pin configuration to factory defaults
1 = Enable
0 = Disable

Bit 7-5 **Unimplemented:** Read as '0'

bit 4 **UseStrict:** Watchdog 'strict mode' enable bit
1 = The watchdog timeout can only be reset by writing the value in WATCHDOG_KEY_DEFAULT to the WATCHDOG_CLEAR register. This mode insures that spurious communication, or mis-communication won't reset the Watchdog timeout.
0 = The watchdog is cleared whenever a response to a command is transmitted. This is the default.

bit 3 **SetDAC1:**
1 = On Watchdog timeout, set DAC1 to the voltage defined in WATCHDOG_DAC1_DEFAULT
0 = Do not set DAC1 on Watchdog timeout

bit 2 **SetDAC0:**
1 = On Watchdog timeout, set DAC0 to the voltage defined in WATCHDOG_DAC0_DEFAULT
0 = Do not set DAC0 on Watchdog timeout

bit 1 **ApplyDIO:**
1 = On Watchdog timeout, set the digital I/O lines according to WATCHDOG_DIO_STATE_DEFAULT, WATCHDOG_DIO_DIRECTION_DEFAULT, and WATCHDOG_DIO_INHIBIT_DEFAULT.
0 = Do not set digital I/O on Watchdog timeout.

bit 0 **HardReset:**
1 = On Watchdog timeout, perform a hard device reset.
0 = On Watchdog timeout, do not perform a hard reset.

Example: A value of 521 saved to WATCHDOG_OPTIONS_DEFAULT translates into 0b1000001001, which would act on bits 9, 3, and 0. So PinCfgStartup, SetDAC1, and HardReset would be enabled.

Details

The Watchdog options are provided in addition to power-up defaults, so that it is possible to create a unique response to failed communication, rather than performing the same response as after a power-cycle. However, in many cases it is appropriate to perform the same response, which is explained in the basic usage section above.

The Watchdog timeout is reset every time a response to a command is transmitted, unless **UseStrict** is set. Enable the **UseStrict** bit to make the device even more sensitive to failures. When enabled, strict mode will prevent a timeout reset due to spurious communication. When in strict mode, the timeout can only be reset when the value stored in WATCHDOG_KEY_DEFAULT is written to WATCHDOG_CLEAR. In strict mode, user code should reset the watchdog timeout before the timeout period elapses, otherwise the Watchdog will activate regardless of valid communication.

17.0 DB37

Number of Pins: **37**

Screw type: **#4-40**

Contacts: **Gold-coated**

Form factor: **D-Sub**

This high-density connector provides access to the T7 features that are not available on the screw terminal edge of the unit. It brings out analog inputs (AIN), analog outputs (DAC), digital I/O (FIO, MIO), and other signals. Some signals appear on both the DB37 connector and screw terminals, so care must be taken to avoid a short circuit.

Signals shared between T7 screw terminals and the DB37 are denoted in **bold**.

Pinout

DB37 Pinouts					
1	**GND**	14	AIN9	27	**Vs**
2	**200uA**	15	AIN7	28	Vm+
3	FIO6	16	AIN5	29	**DAC1**
4	FIO4	17	**AIN3**	30	**GND**
5	**FIO2**	18	**AIN1**	31	AIN12
6	**FIO0**	19	**GND**	32	AIN10
7	MIO1	20	**10uA**	33	AIN8
8	**GND**	21	FIO7	34	AIN6
9	Vm-	22	FIO5	35	AIN4
10	**GND**	23	**FIO3**	36	**AIN2**
11	**DAC0**	24	**FIO1**	37	**AIN0**
12	AIN13	25	MIO0		
13	AIN11	26	MIO2		

Table 2.11-1. DB37 Connector Pinouts

VS, GND, FIO/MIO, AIN, DAC, 200UA/10UA

Descriptions of these can be found in their related sections of this datasheet.

VM+/VM-

Vm+/Vm- are bipolar power supplies intended to power external multiplexer ICs such as the DG408 from Intersil. The multiplexers can only pass signals within their power supply range, so Vm+/Vm- can be used to pass bipolar signals. Nominal voltage is ±13 volts at no load and ±12 volts at 2.5 mA. Both lines have a 100 ohm source impedance, and are designed to provide 2.5 mA or less. This is the same voltage supply used internally by the T7 to bias the analog input amplifier and multiplexers. If this supply is loaded more than 2.5 mA, the voltage can droop to the point that the maximum analog input range is reduced. If this supply is severely overloaded (e.g. short circuited), then damage could eventually occur. If Vm+/Vm- are used to power multiplexers, series diodes are recommended as shown in Figure 9 of the Intersil DG408 datasheet. Not so much to protect the mux chips, but to prevent current from going back into Vm+/Vm-. Use Schottky diodes to minimize voltage drop.

OEM

The OEM T7 has a separate header location to bring out the same connections as the DB37 connector. This OEM header location is labeled J3. Find the pinout for J3 below:

Pinout

J3					
1	GND	2	GND	3	PIN20 (10uA)
4	PIN2 (200uA)	5	FIO7	6	FIO6
7	FIO5	8	FIO4	9	FIO3
10	FIO2	11	FIO1	12	FIO0
13	MIO0/CIO0	14	MIO1/CIO1	15	MIO2/CIO2
16	GND	17	Vs	18	Vm-
19	Vm+	20	GND	21	DAC1
22	DAC0	23	GND	24	AIN13
25	AIN12	26	AIN11	27	AIN10
28	AIN9	29	AIN8	30	AIN7
31	AIN6	32	AIN5	33	AIN4
34	AIN3	35	AIN2	36	AIN1
37	AIN0	38	GND	39	GND
40	GND				
J3 OEM Pin-Header					

The J3 OEM header hole spacing is 0.1 inches, for additional information on dimensions, see the Enclosure and PCB section.

18.0 DB15

Number of Pins: **15**

Screw type: **#4-40**

Contacts: **Gold-coated**

Form factor: **D-Sub**

The DB15 connector brings out 12 additional digital I/O. It has the potential to be used as an expansion bus, where the 8 EIO are data lines and the 4 CIO are control lines. In the Windows LabJackUD driver, the EIO are addressed as digital I/O bits 8 through 15, and the CIO are addressed as bits 16-19.

```
0-7     FIO0-FIO7
8-15    EIO0-EIO7
16-19   CIO0-CIO3
```

These 12 channels 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":/support/u6/users-guide/appendix-a. All digital I/O on the U6 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. 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).

DB15 Pinouts			
1	Vs	9	CIO0
2	CIO1	10	CIO2
3	CIO3	11	GND
4	EIO0	12	EIO1
5	EIO2	13	EIO3
6	EIO4	14	EIO5
7	EIO6	15	EIO7
8	GND		
Table 2.12-1. DB15 Connector Pinouts			

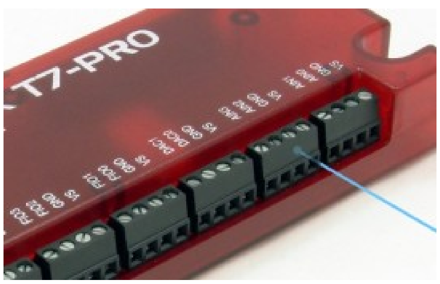
OEM

The OEM T7 has a separate header location to bring out the same connections as the DB15 connector. This OEM header location is labeled J2. Find the pinout for J2 below:

J2				
1	GND	2	VS	
3	CIO0	4	CIO1	
5	CIO2	6	CIO3	
7	GND	8	EIO0	
9	EIO1	10	EIO2	
11	EIO3	12	EIO4	
13	EIO5	14	EIO6	
15	EIO7	16	GND	
J2 OEM Pin-Header				

The J3 OEM header hole spacing is 0.1 inches, for additional information on dimensions, see the Enclosure and PCB section.

19.0 Temperature Sensor



The T7 has a dedicated internal temperature sensor. The sensor is physically located near the AIN1 screw-terminal. The internal temperature sensor is read by acquiring **analog input channel 14** and returns **degrees K**.

Offset considerations

The T7 enclosure typically makes a small (TBD °C) difference in the temperature at the internal sensor. With the enclosure on, the temperature at the sensor is typically higher (TBD °C) than ambient, while with the enclosure off the temperature at the sensor is typically higher (TBD °C) than ambient. The calibration constants have an offset of (TBD °C), so returned calibrated readings are nominally the same as ambient with the enclosure installed, and (TBD °C) below ambient with the PCB in free air.

Note on thermocouples

The screw-terminals AIN0-AIN3 are also typically (TBD °C) above ambient with the enclosure installed, so when the internal temperature sensor is used for thermocouple cold junction compensation on AIN0-AIN3, it is recommended to add (TBD °C) to its value as you want the actual temperature of the screw-terminals, not necessarily ambient temperature. If using the CB37 do not add any offset, as the screw-terminals are typically the same temperature as ambient.

20.0 Screw Terminals

The T7 screw terminals are capable of handling between 14-24 AWG gauge wire, with a maximum current rating of 2A, which far exceeds any current that should be going into/out of a LabJack device.

Physical Characteristics TBD

Electrical Characteristics TBD

OEM

JP1

JP2

JP3

JP4

JP5

JP6

21.0 Electrical Characteristics

Electrical Characteristics for describing the T7 can be broken down into four primary sections with a few sub-sections as illustrated below:

21.1 Power Requirements

21.1.0 General Characteristics:

Parameter	*Condition*	*Min*	*Typical*	*Max*	*Units*
Supply Voltage		4.75		5.25	Volts
Supply Current	No connected loads	8.1	250		mA

21.1.0 Power Consumption:

The T7 has several power domains. USB and Core speed are not yet ready user level control, but have been included in the following table to show the capabilities of the device. The values shown are typical.

Core Speed	Eth (1)	Eth Linked	AINs	WiFi	WiFi Associated	LEDs	USB (1)	Draw (mA)
80M	ON	Yes	ON	ON	Yes	ON	ON	290
80M	ON	Yes	ON	ON	No	ON	ON	285
80M	ON	Yes	ON	OFF	No	ON	ON	253
80M	ON	No	ON	OFF	No	ON	ON	210
80M	OFF	No	ON	OFF	No	ON	ON	174
80M	OFF	No	OFF	OFF	No	ON	ON	142
80M	OFF	No	OFF	OFF	No	OFF	ON	105
80M	OFF	No	OFF	OFF	No	OFF	OFF	79
20M	OFF	No	OFF	OFF	No	OFF	OFF	23
2M	OFF	No	OFF	OFF	No	OFF	OFF	8.8
250k	OFF	No	OFF	OFF	No	OFF	OFF	8.1
1) Ethernet and USB require that the core be running at least 20MHz.								

21.2 Analog Input

21.2.0 General Characteristics:

The table below highlights some general characteristics about the analog front end of the T7. More information about the T7's analog front end can be found in the correlating sub-sections.

Parameter	Conditions	Min	Typical	Max	Units
Typical Input Range (1)	Gain=1	-10.5		10.1	Volts
Max AIN Voltage to GND (2)	Valid Readings	-11.5		11.5	Volts
Max AIN Voltage to GND (3)	No Damage	-20		20	Volts
Input Bias Current (4)			20		nA
Input Impedance (4)			1		GΩ
Source Impedance (4)			1		kΩ
Integral Linearity Error	Gain=1, 10, 100		±YTBD	±0.01	%FS
	Gain=1000		±YTBD	±0.1	%FS
Absolute Accuracy	Gain=1, 10, 100		±0.01		%FS
	Gain=1000		±0.1		%FS
Temperature Drift			15		ppm/°C
Noise (Peak-To-Peak)	See Section 21.2.1			<1	µV
Effective Resolution (RMS)	See Section 21.2.1			22	bits
Noise-Free Resolution	See Section 21.2.1			20	bits
(1) Differential or single-ended					
(2) This is the maximum voltage on any AIN pin compared to ground for valid measurements on that channel. For single-ended readings on the channel itself, inputs are limited by the "Typical Input Range" above, and for differential readings consult the signal range tables in Section 21.2.1. Further, if a channel has over 13.0 volts compared to ground, readings on other channels could be affected. Because all even channels are on 1 front-end mux, and all odd channels on a 2nd front-end mux, an overvoltage (>13V) on a single channel will generally affect only even or only odd channels.					
(3) Maximum voltage, compared to ground, to avoid damage to the device. Protection level is the same whether the device is powered or not.					
(4) The key specification here is the maximum source impedance. As long as the					

Table 21.2.0.1

21.2.1 Signal Ranges

This section looks further into how a the T7 acquires an analog reading and describes in depth the range capabilities of the T7 device. This section provides details in regards to differential signal measuring.

More Information about section 21.2.1 Signal Ranges

21.2.2 Noise & Resolution Characteristics:

The table under this section provides typical noise levels of the T7 under ideal conditions. To get the most accurate and precise readings you should choose an appropriate voltage range for the signal being measured and a high resolution index. Keep in mind that the higher you set these values the longer it takes for the device to acquire each analog reading. A short preview for the table lies below:

Resolution Index	Average Latency	Rounded P2P Noise	Peak-To-Peak Resolution	Noise-Free Resolution	Rounded RMS Noise	Ef Re
	ms	24-bit counts	bits	µV	24-bit counts	bit
Gain/Range: 1/±10						
1	0.5	1280	13.7	1579.4	197	
2	0.5	768	14.4	947.7	141	
3	0.5	640	14.7	789.7	97	
4	0.5	512	15	631.8	78	
5	0.5	384	15.4	473.8	55	
6	0.7	256	16	315.9	40	
7	1	192	16.4	236.9	30	
8	1.6	128	17	157.9	22	
9	4.3	157	16.7	193.8	22	
10	14.5	79	17.7	97.5	11	
11	67.6	39	18.7	48.1	6	
12	160.5	33	19	40.7	4	
Gain/Range: 10/±1						
1	0.6	2048	13	252.7	280	
2	0.6	1408	13.5	173.7	202	
3	1	1280	13.7	157.9	156	
4	1	832	14.3	102.6	122	
5	1.7	512	15	63.2	95	
6	3	448	15.2	55.3	60	
7	3.3	256	16	31.6	40	
8	3.9	256	16	31.6	32	
9	4.3	173	16.6	21.4	23	

Table 21.2.2.1

More Information about section 21.2.2 Noise & Resolution Characteristics and Full Sized Table

21.2.3 Command Response Performance:

The tables related to this section provide typical command response speed performance results. These results may be used to aid in the creation of control systems and other time sensitive applications. It is important to note that these speeds may vary between each system in use. These speeds may also change based on the signal quality, size, topology, and congestion of your current network when communicating over wifi or ethernet. In-house testing may be required to get more exact results for your particular application.

Below are results for reading/writing to digital I/O ports and the devices DAC channels:

	USB high-high	USB other	Ethernet	Wifi[1]	Units
No I/O	TBD	TBD	0.77	TBD	ms
Read All DI	0.70	TBD	1.06	TBD	ms
Write All DI	0.70	TBD	1.06	TBD	ms
Write Both DACs	0.70	TBD	1.08	TBD	ms

Table 21.2.3.1

Below are results for reading analog channels at various gain and resolution indices:

Resolution Index	Resolution (bits)	Single Channel USB (ms)	8 Channels USB (ms)
Gain = x1 or Range ±10V			
1	16.1	0.65	1.54
2	16.4	0.70	1.55
3	16.9	0.73	1.68
4	17.5	0.77	2.02
5	17.9	0.84	3.14
6	18.4	0.96	4.51
7	18.8	1.04	6.97
8	19	2.05	11.48
9	19.7	4.57	30.79
10	20.6	15.08	110.11
11	21.3	67.93	528.65
12	22	160.33	1,261.82
Gain = x10 or Range ±1V			
1	15.5	0.68	3.59
2	15.9	0.70	3.69
3	16.5	0.96	7.02
4	17.1	1.14	7.35
5	17.5	2.16	12.01
6	18.1	3.57	20.89
7	18.3	3.95	23.16
8	18.7	4.00	27.54
9	19.6	4.56	30.73
10	20.3	15.18	110.01
11	21.3	68.00	528.48
12	21.8	160.33	1,261.61
Gain = x100 or Range ±0.1V			
1	13.9	1.85	10.94
2	14.3	3.16	19.05
3	14.8	6.36	43.11
4	15.3	6.44	43.52
5	15.8	6.45	43.95
6	16.4	11.93	84.74
7	16.8	12.00	87.12
8	17.2	12.61	90.81
9	18.6	4.57	30.56
10	19.3	15.11	110.17
11	19.7	68.06	528.66
12	19.7	160.39	1,261.72
Gain = x1000 or Range ±0.01V			
1	12.1	6.35	43.20
2	12.6	11.85	82.80
3	13	11.86	83.08
4	13.5	11.94	83.24
5	14	11.96	83.76
6	14.5	11.96	84.94
7	14.9	12.00	87.18
8	15.2	12.57	91.49
9	15.6	4.62	30.75
10	16.2	15.05	110.01
11	16.4	68.00	528.52
12	16.6	160.47	1,261.85

Table 21.2.2.2

More Information about section 21.2.3 Command Response Performance and Full Sized Tables

21.2.4 Stream Performance:

The tables related to this section provide typical stream-related performance results. These results are useful for determining what types of signals can be analyzed using a T7. A T7 is capable of streaming analog data at a steady rate so that various discrete time signal analysis tools can be utilized to interpret data. Depending on your network speed, congestion, computer performance and other factors, you may be able to get results faster than displayed below however the typical user should not rely on this extra performance before individual environment-based testing has been performed. Speeds will be different based on what interface is being used to stream data, ethernet or USB. Please note that WiFi streaming is not currently supported.

Below analog channel characteristics for streaming data at different rates with different gain & resolution options:

Res. Index	Max Stream	ENOB	ENOB	Noise	Interchannel
Gain = x1 or Range ±10V (PRELIMINARY)					
1	TBD	16.31	14.28	3.33	15
2	TBD	16.92	14.57	2.72	26
3	TBD	17.30	14.93	2.11	47
4	TBD	17.97	15.59	1.33	94
5	TBD	18.44	16.04	0.98	180
6	TBD	18.91	16.48	0.72	360
7	TBD	19.35	16.96	0.52	720
8	TBD	19.74	17.37	0.39	1,440
Gain = x10 or Range ±1V (PRELIMINARY)					
1	TBD	16.03	13.69	5.00	210
2	TBD	16.46	14.10	3.76	220
3	TBD	16.83	14.58	2.69	560
4	TBD	17.53	15.12	1.85	590
5	TBD	17.98	15.62	1.32	1,220
6	TBD	18.50	16.07	0.95	2,450
7	TBD	19.00	16.58	0.67	2,800
8	TBD	19.38	16.98	0.50	3,550
Gain = x100 or Range ±0.1V (PRELIMINARY)					
1	TBD	13.83	11.40	24.35	1,040
2	TBD	14.34	11.95	16.62	2,100
3	TBD	14.76	12.33	12.79	4,200
4	TBD	15.28	12.87	8.80	4,250

Table 21.2.3.1

More Information about section 21.2.4 Stream Performance and Full Sized Tables

21.2.1 Signal Range

The following figures show the approximate signal range of the T7 analog inputs. "Input Common-Mode Voltage" or Vcm is (Vpos + Vneg)/2.

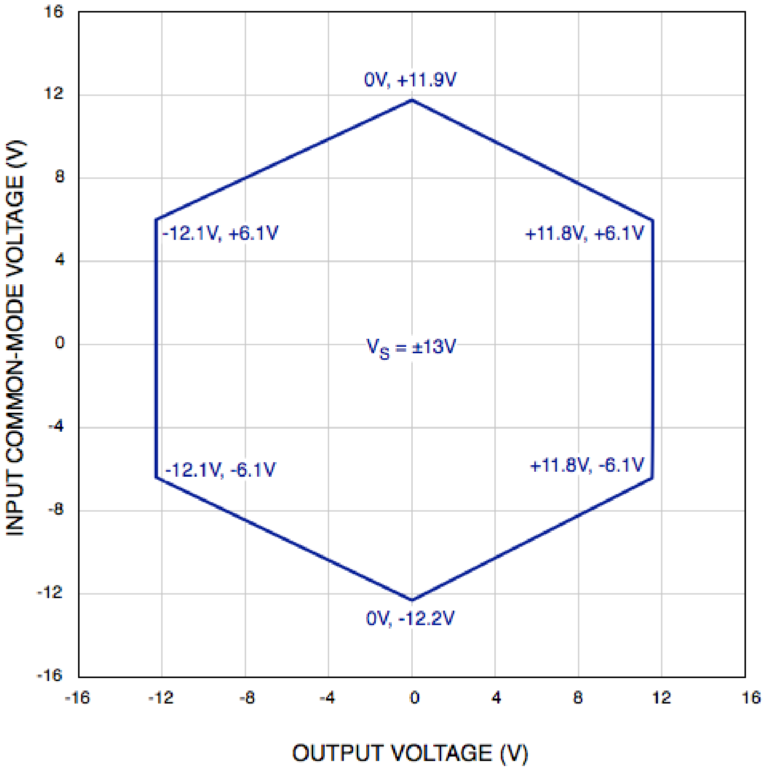
Keep in mind that the voltage of any input compared to GND should be within the Vm+ and Vm- rails by at least 1.5 volts, so if Vm is the typical ±13 volts, the signals should be within ±11.5 volts compared to GND.

Example #1: Say a differential signal is measured where Vpos is 10.05 volts compared to GND and Vneg is 9.95 volts compared to ground, and G=100. That means Vcm=10.0 volts, Vdiff=0.1 volts, and the expected Vout=10.0 volts. There is not figure for G=100 below, but Vcm=10.0 volts and Vout=10.0 volts is not valid at G=1 or G=1000, so is certainly not valid in between.

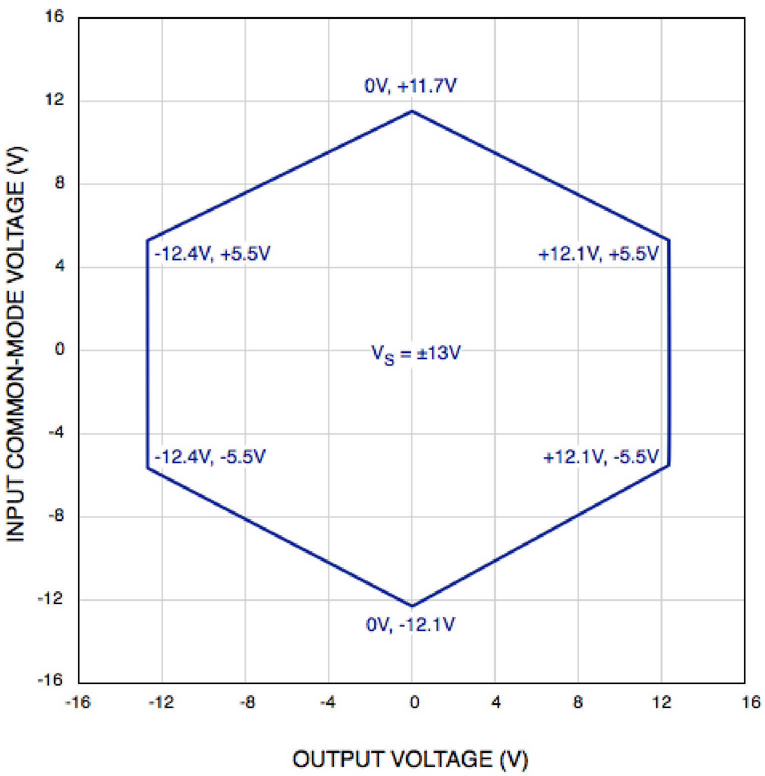
Example #2: Say a differential signal is measured where Vpos is 15.0 volts compared to GND and Vneg is 14.0 volts compared to ground, and G=1. That means Vcm=14.5 volts, Vdiff=1.0 volts, and the expected Vout=1.0 volts. The voltage of each input compared to GND is too high, so this would not work at all.

Example #3: Say a single-ended signal is measured where Vpos is 10.0 volts compared to GND and G=1.; That means Vcm=5.0 volts, Vdiff=10.0 volts, and the expected Vout=10.0 volts. This is fine according to the figure below.

Input Common-Mode Voltage Range vs. Output Voltage, G = 1



Input Common-Mode Voltage Range vs. Output Voltage, G = 1000



21.2.2 Noise And Resolution Characteristics

Overview & Testing procedure

The graphs and raw data table under this section provides typical noise levels of the T7 under ideal conditions. The resulting voltage resolution is then calculated based on the noise levels.

Measurements were taken with AIN0 connected to GND with a short jumper wire, or from internal ground channel #15.

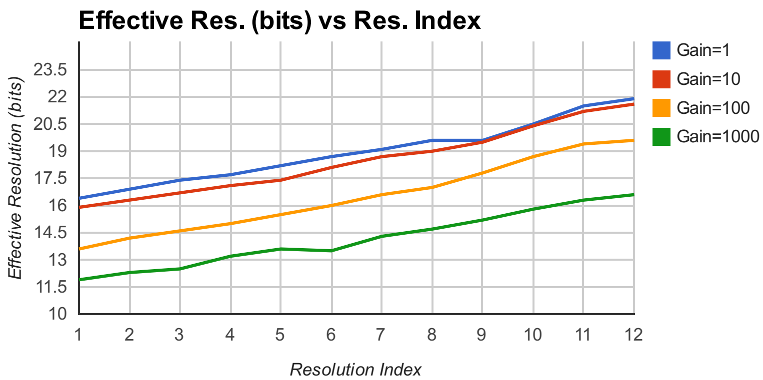
All "counts" data are aligned as 24-bit values. To equate to counts at a particular resolution (Res) use the formula counts/(2^(24-Res)). For instance, with the T7 set to resolution=1 and the ±10 volt range, there are 1024 counts of noise when looking at 24-bit values. To equate this to 16-bit data, we take 1024(2^8) which equals 4 counts of noise when looking at 16-bit values.

Noise-free data is determined by taking 2000 readings and subtracting the minimum value from the maximum value.

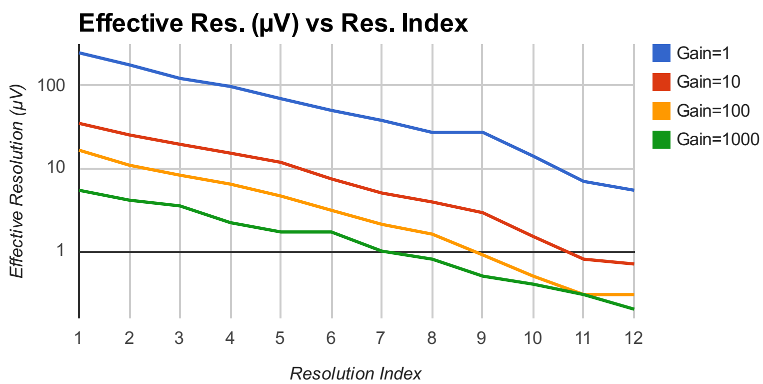
RMS and Effective data are determined from the standard deviation of 2000 readings. In other words, the RMS data represents most readings, whereas noise-free data represents all readings.

Graphical Results

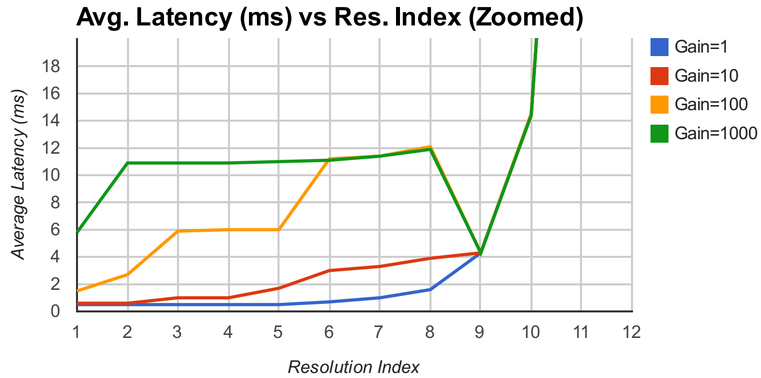
The graph below shows the Effective Resolution in bits that the LabJack is able to produce that correlate to a given input voltage at different gain and resolution configurations. It is clear to see that a higher resolution index produces a more precise result.

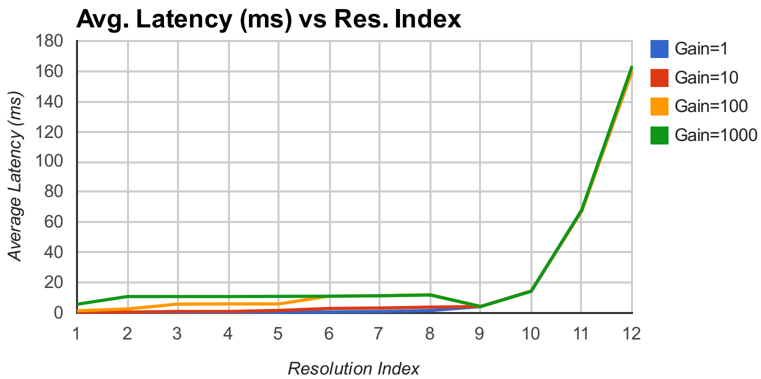


The graph below shows the Effective Resolution in µV that the LabJack is able to produce that correlate to a given input voltage at different gain and resolution configurations. It is clear to see that a higher resolution index produces a more precise result. It also becomes clear in this graph that choosing a proper gain level that corresponds to the expected voltage is important.



The graph below shows the average time it takes for LabVIEW to capture a single reading from various resolution indicies. The first chart shows a zoomed in view of the data, the second shows the full range of latencies. When using a T7-Pro the included high resolution converter starts being used at resolution 9. Relating this to the graph's, there is a noticeable drop in command response latency when jumping from resolution index 8 to resolution index 9 at all gain levels. Shown too by these graphs, the high resolution converter has a higher input impedance than the high speed converter used for resolution indicies 1-8, therefore it requires less time at all gain levels to acquire data.





Raw Data

The data table below shows all of the information collected to produce the above graphs along with some more useful data pertaining to the T7.

Resolution Index	Average Latency	Rounded P2P Noise	Peak-To-Peak Resolution	Noise-Free Resolution	Rounded RMS Noise	Effective Resolution	Effective Resolution
	ms	24-bit counts	bits	μV	24-bit counts	bits	μV
Gain/Range: 1/±10							
1	0.5	1280	13.7	1579.4	197	16.4	243.5
2	0.5	768	14.4	947.7	141	16.9	174.2
3	0.5	640	14.7	789.7	97	17.4	119.8
4	0.5	512	15	631.8	78	17.7	95.8
5	0.5	384	15.4	473.8	55	18.2	68.3
6	0.7	256	16	315.9	40	18.7	49.4
7	1	192	16.4	236.9	30	19.1	37.4
8	1.6	128	17	157.9	22	19.6	26.8
9	4.3	157	16.7	193.8	22	19.6	26.9
10	14.5	79	17.7	97.5	11	20.5	13.9
11	67.6	39	18.7	48.1	6	21.5	6.9
12	160.5	33	19	40.7	4	21.9	5.4
Gain/Range: 10/±1							
1	0.6	2048	13	252.7	280	15.9	34.5
2	0.6	1408	13.5	173.7	202	16.3	25
3	1	1280	13.7	157.9	156	16.7	19.3
4	1	832	14.3	102.6	122	17.1	15.1
5	1.7	512	15	63.2	95	17.4	11.7
6	3	448	15.2	55.3	60	18.1	7.4
7	3.3	256	16	31.6	40	18.7	5
8	3.9	256	16	31.6	32	19	3.9
9	4.3	173	16.6	21.4	23	19.5	2.9
10	14.5	91	17.5	11.2	12	20.4	1.5
11	67.6	61	18.1	7.5	7	21.2	0.8
12	160.5	43	18.6	5.3	5	21.6	0.7
Gain/Range: 100/±0.1							
1	1.5	12545	10.4	154.8	1328	13.6	16.4
2	2.7	9601	10.8	118.4	874	14.2	10.8
3	5.9	7232	11.2	89.2	665	14.6	8.2
4	6	6400	11.4	79	516	15	6.4
5	6	5312	11.6	65.5	373	15.5	4.6
6	11.2	2944	12.5	36.3	252	16	3.1
7	11.4	1856	13.1	22.9	168	16.6	2.1
8	12.1	1472	13.5	18.2	127	17	1.6
9	4.3	1604	13.4	19.8	76	17.8	0.9
10	14.5	503	15	6.2	39	18.7	0.5
11	67.5	335	15.6	4.1	25	19.4	0.3
12	160.5	285	15.8	3.5	22	19.6	0.3
Gain/Range: 1000/±0.01							
1	5.8	68818	7.9	84.9	4354	11.9	5.4
2	10.9	80075	7.7	98.8	3300	12.3	4.1
3	10.9	89828	7.5	110.8	2877	12.5	3.5
4	10.9	49375	8.4	60.9	1778	13.2	2.2
5	11	35113	8.9	43.3	1362	13.6	1.7
6	11.1	38950	8.8	48	1413	13.5	1.7
7	11.4	22129	9.6	27.3	822	14.3	1
8	11.9	10425	10.7	12.9	643	14.7	0.8
9	4.3	7622	11.1	9.4	437	15.2	0.5
10	14.4	2837	12.5	3.5	285	15.8	0.4
11	68.2	1725	13.2	2.1	210	16.3	0.3
12	163.5	2102	13	2.6	171	16.6	0.2

Table 21.2.2.1

21.2.2 Noise And Resolution

21.2.3 Command Response Speeds

Description:

Everything besides streaming is done in command/response mode, meaning that all communication is initiated by a command from the host which is followed by a response from the T7.

All communication performed with the T7 is done using single modbus read & write requests as well as a more streamlined modbus-based protocol that performs bulk reads & writes. Functions for both of these are available through the LJM library.

Testing Procedure:

The times shown in these graphs were measured using a LabVIEW program that executed a loop 1000 times and divides the total execution time by 1000 to get an overall average execution time. Thus the execution time includes windows latency, LJM driver overhead, communication time, T7 processing time, etc. for USB, ethernet, and wifi.

A "USB high-high configuration means the T7 is connected to a high-speed USB2 hub which is then connected to a high-speed USB2 host. Even though the T7 is not a high-speed USB device, such a configuration does provide improved performance. Typical examples of "USB other" would be a T7 connected to an old full-speed hub (hard to find) or more likely the T7 is connected directly to the USB host (your PC) even if the host supports high-speed.

Additional Considerations:

It is important to understand that Linux, Mac, Windows, and most other electronic devices run operating systems that are not known as "Real Time Operating Systems", they run what is known as "Best-Effort" schedulers meaning that these speeds will all vary based on each individual computer, the hardware inside of it, its currently enabled peripherals, current network traffic, strength of signal, and many more variables. These measurements are meant to be estimates or approximate speeds that you should expect to see with a typical system operating in a typical environment.

Speed Results:

Below are time results for typical read and write commands to a T7:

	USB High-High (ms)	USB Other (ms)	Ethernet (ms)	Wifi (ms)
No I/O	0.7	2.1	1.1	4.2
Read All DI	0.7	2.1	1.1	4.2
Write All DO	0.7	2.1	1.1	4.2
Write Both DACs	0.7	2.1	1.1	4.2

Table 21.2.3.1

Below are results for reading analog channels at various gain and resolution indices:

Resolution Index	Resolution (bits)	USB Average Response Time (ms)	Ethernet Average Response Time (ms)	WiFi (1) Average Response Time (ms)	8 Channels USB Average Response Time (ms)	8 Channels Ethernet Average Response Time (ms)	8 Chann Average Time (m
Gain = x1 or Range ±10V							
1	16.1	0.637	1.071	4.284	1.019	1.568	
2	16.4	0.625	0.881	4.492	1.063	1.66	
3	16.9	0.635	1.561	4.26	1.46	1.669	
4	17.5	0.619	1.626	4.482	1.541	1.795	
5	17.9	0.754	1.646	3.977	2.497	2.567	
6	18.4	1.01	1.621	3.863	3.288	3.694	
7	18.8	1.271	1.683	4.164	5.839	5.914	
8	19	1.871	2.356	5.088	10.185	10.288	
9	19.7	4.802	4.702	7.653	29.468	29.909	
10	20.6	14.942	14.803	17.008	109.323	110.338	
11	21.3	68.012	68.381	70.464	533.894	536.539	
12	22	161.008	162.277	163.667	1276.922	1278.385	
Gain = x10 or Range ±1V							
1	15.5	0.758	1.726	4.066	2.986	3.052	
2	15.9	0.815	1.653	4.336	3.115	3.176	
3	16.5	1.28	1.666	4.911	5.775	5.824	
4	17.1	1.168	1.631	3.997	5.963	6.073	
5	17.5	2.374	1.98	4.835	10.626	10.696	
6	18.1	3.555	3.563	6.159	19.619	19.649	
7	18.3	3.72	3.881	7.127	21.719	21.8	
8	18.7	4.388	4.405	7.111	26.077	26.211	
9	19.6	4.811	4.865	7.11	29.52	29.566	
10	20.3	14.844	14.786	17.137	109.315	110.923	
11	21.3	68.06	69.346	78.177	533.922	537.421	
12	21.8	160.983	161.692	169.358	1276.965	1279.533	
Gain = x100 or Range ±0.1V							
1	13.9	1.784	2.092	4.907	9.503	9.591	
2	14.3	2.938	3.256	6.581	17.575	17.602	
3	14.8	6.321	6.449	8.822	41.95	42.298	
4	15.3	6.367	6.496	9.617	42.237	42.214	
5	15.8	6.407	6.535	9.449	42.658	43.003	
6	16.4	11.737	11.648	29.648	83.815	84.196	
7	16.8	11.999	11.952	44.525	85.959	86.583	
8	17.2	12.492	12.392	15.134	90.225	91.506	
9	18.6	4.778	4.789	8.096	29.565	29.545	
10	19.3	14.861	14.772	17.846	109.289	109.984	
11	19.7	68.061	68.382	78.27	533.93	538.025	
12	19.7	160.973	162.563	172.135	1276.962	1278.298	
Gain = x1000 or Range ±0.01V							
1	12.1	6.397	6.413	9.037	41.68	42.116	
2	12.6	11.454	11.369	14.573	81.763	82.197	
3	13	11.465	11.414	14.143	81.912	82.621	
4	13.5	11.528	11.451	13.622	82.277	83.507	
5	14	11.563	11.512	14.276	82.716	82.807	
6	14.5	11.693	11.59	13.422	83.799	84.499	
7	14.9	12.009	11.853	14.318	85.945	86.311	
8	15.2	12.523	12.483	14.898	90.22	91.206	
9	15.6	4.847	4.847	7.811	29.556	30.41	
10	16.2	14.897	14.787	16.838	109.286	109.972	
11	16.4	68.079	68.357	70.408	533.925	536.769	
12	16.6	160.934	161.914	168.554	1276.969	1278.273	

Table 21.2.3.2

Note 1: Wi-Fi latency varies depending on network traffic and signal strength. 300ms to 900ms is not uncommon.
21.2.3 Command Response

21.2.4 Stream Performance

General Information:

For general information about streaming please visit section 4.2

The tables related to this section provide typical stream-related performance results. These results are useful for determining what types of signals can be analyzed using a T7. A T7 is capable of streaming analog data at a steady rate so that various discrete time signal analysis tools can be utilized to interpret data. Depending on your network speed, congestion, computer performance and other factors, you may be able to get results faster than displayed below however the typical user should not rely on this extra performance before individual environment-based testing has been performed. Maximum Speeds will be different based on what interface is being used to stream data, ethernet or USB. Please note that WiFi streaming is not currently supported.

The data below shows test results from various stream performance parameters. Quite often it may be possible to obtain results indicating faster performance than what is listed. To obtain performance results matching or exceeding the results below it may be necessary control various attributes regarding the use of your device. Stream rates can be limited by a number of different factors, USB connection speed, network traffic, program efficiency, and the running programs priority. Quite often the maximum stream rate is capped by the computer's processing capabilities as calibration of the data coming from the device is done in LJM instead of on the device to increase performance.

Procedure

Graphical Results

Raw Data

Res. Index	Max Stream	ENOB	ENOB	Noise	Interchannel
Gain = x1 or Range ±10V (PRELIMINARY)					
1	TBD	16.31	14.28	3.33	15
2	TBD	16.92	14.57	2.72	26
3	TBD	17.30	14.93	2.11	47
4	TBD	17.97	15.59	1.33	94
5	TBD	18.44	16.04	0.98	180
6	TBD	18.91	16.48	0.72	360
7	TBD	19.35	16.96	0.52	720
8	TBD	19.74	17.37	0.39	1,440
Gain = x10 or Range ±1V (PRELIMINARY)					
1	TBD	16.03	13.69	5.00	210
2	TBD	16.46	14.10	3.76	220
3	TBD	16.83	14.58	2.69	560
4	TBD	17.53	15.12	1.85	590
5	TBD	17.98	15.62	1.32	1,220
6	TBD	18.50	16.07	0.95	2,450
7	TBD	19.00	16.58	0.67	2,800
8	TBD	19.38	16.98	0.50	3,550
Gain = x100 or Range ±0.1V (PRELIMINARY)					
1	TBD	13.83	11.40	24.35	1,040
2	TBD	14.34	11.95	16.62	2,100
3	TBD	14.76	12.33	12.79	4,200
4	TBD	15.28	12.87	8.80	4,250
5	TBD	15.80	13.40	6.08	4,400
6	TBD	16.30	13.86	4.44	4,600
7	TBD	16.76	14.38	3.09	4,900
8	TBD	17.20	14.84	2.26	5,600
Gain = x1000 or Range ±0.01V (PRELIMINARY)					
1					
2					
3					
4					
5					
6					
7					
8					

Table 21.2.4.1

21.3 Analog Output

21.3.0 General Information

The T7 supports two analog output channels labeled "DAC0" and "DAC1". General characteristics of the two channels are available below.

Parameter	Conditions	Min	Typical	Max	Units
Nominal Output Range (1)	No Load	0.01		4.99	Volts
	@ ±2.5 mA	0.25		.0.25	Volts
Resolution			12		Bits
Absolute Accuracy	5% to 95% FS		TBD		% FS
Integral Linearity Error			±1.5	±2	counts
Differential Linearity Error			±0.25	±0.5	counts
Error Due To Loading	@ 100 µA		0.15		%
	@ 1mA		2.3		%
Source Impedance			TBD		Ω
Short Circuit Current (2)	Max to GND		20.5		mA
Time Constant			4		µs
(1) Maximum and minimum analog output voltage is limited by the supply voltages (Vs and GND). The specifications assume Vs is 5.0 volts. Also, the ability of the DAC output buffer to driver voltages close to the power rails, decreases with increasing output current, but in most applications the output is not sinking/source much current as the output voltage approaches GND.					
(2) Continuous short circuit will not cause damage.					

Table 21.3.0.1

21.3.1 Speed and Settling

Below you can find some characteristics involving the speed & settling times of the DAC channels.

TBD.

21.4 Digital Input/Output

21.4.0 - General Info

Below you can find information regarding the T7's Digital Inout/Output lines. More specifically, they are called FIO, EIO, and CIO lines.

Parameter	**Conditions**	**Min**	**Typical**	**Max**	**Units**
Low Level Input Voltage		-0.3		0.5	Volts
High Level Input Voltage		2.64		5.8	Volts
MaximumInput Voltage (1)	FIO	-10		10	Volts
	EIO/CIO	-10		10	Volts
Output Low Voltage (2)	No Load		0.01		Volts
---FIO	Sinking 1 mA		0.55		Volts
---EIO/CIO	Sinking 1 mA		0.15		Volts
---EIO/CIO	Sinking 5 mA		0.75		Volts
Output High Voltage (2)	No Load		3.3		Volts
---FIO	Sourcing 1 mA		2.75		Volts
---EIO/CIO	Sourcing 1 mA		3.15		Volts
---EIO/CIO	Sourcing 5 mA		2.6		Volts
Short Circuit Current (2)	FIO		6.3		mA
	EIO/CIO		22.9		mA
Output Impedance (2)	FIO		550		Ω
	EIO/CIO		180		Ω
(1) Maximum voltage to avoid damage to the device. Protection works whether the device is powered or not, but continuous voltages over 5.8 volts or less than -0.3 volts are not recommened when the T7 is unpowered, as the voltage will attempt to supply operating power to the T7 possibly causing poor start-up behavior.					
(2) These specifications provide the answer to the question. "How much current can the digital I/O sink or source?". For instance, if EIO0 is configured as output-high and shorted to ground, the current sourced by EIO0 is configured as output-high and shorted to ground, the current sourced by EIO0 into ground will be about 16 mA (3.3/180). If connected to a load that draws 5 mA, EIO0 can provide that current but the voltage will droop to about 2.4 volts instead of the nominal 3.3 volts. If connected to a 180 ohm load to ground, the resulting voltage and current will be about 1.65 volts @ 9 mA.					

Table 21.4.0.1

21.4.1 - Timers & Counters

Below you can find information regarding the T7's Timer & Counter features. The T7 doesn't feature timers and counters that are as flexible as the U3, U6, and UE9.

Timers & Counters	Conditions	Min	Max	Units
Frequency Output (1)		0.02	5 M	Hz
Counter Input Frequency (2)			5	MHz
Input Timer Total Edge Rate (3)	No Stream		TBD	edges/s
	While Streaming		TBD	edges/s
(1) Frequencies up to 40MHz are possible, but they are heavily filtered.				
(2) Hardware counters. 0 to 3.3 volt square wave.				
(3) To Avoid missing edges, keep the total number of applicable edges on all				

Table 21.4.1.1

21.4.2 - Serial Communication

Below you can find information regarding the T7's Serial Communication abilities. Please keep in mind our devices use 3.3V logic levels and provide 5V output along the VS screw terminal. Some ICs require the same logic level as provided to the chip's VCC line so extra steps may be required to integrate specific sensors.

Serial Communication	Conditions	Min	Max	Units
SPI Characteristics				
Clock Frequencies		0.08718	TBD	kHz
I2C Characteristics				

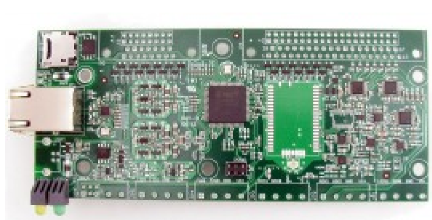
Table 21.4.2.1

21.4 Digital I/O

21.5 OEM Electrical Considerations

[This page isn't fully completed]

There is an OEM version of the T7, called the T7-OEM. Proper ESD precautions should be taken when handling the PCB directly.

USB

The USB connector is not installed on the T7 OEM, for OEM connector options, see Appendix A - Custom OEM.

J5 - Supply Power

If you connect wires to J5 and also use the USB connector, there is risk of back-powering either the USB supply or the supply attached to J5. See below pinout info for J5 connections.

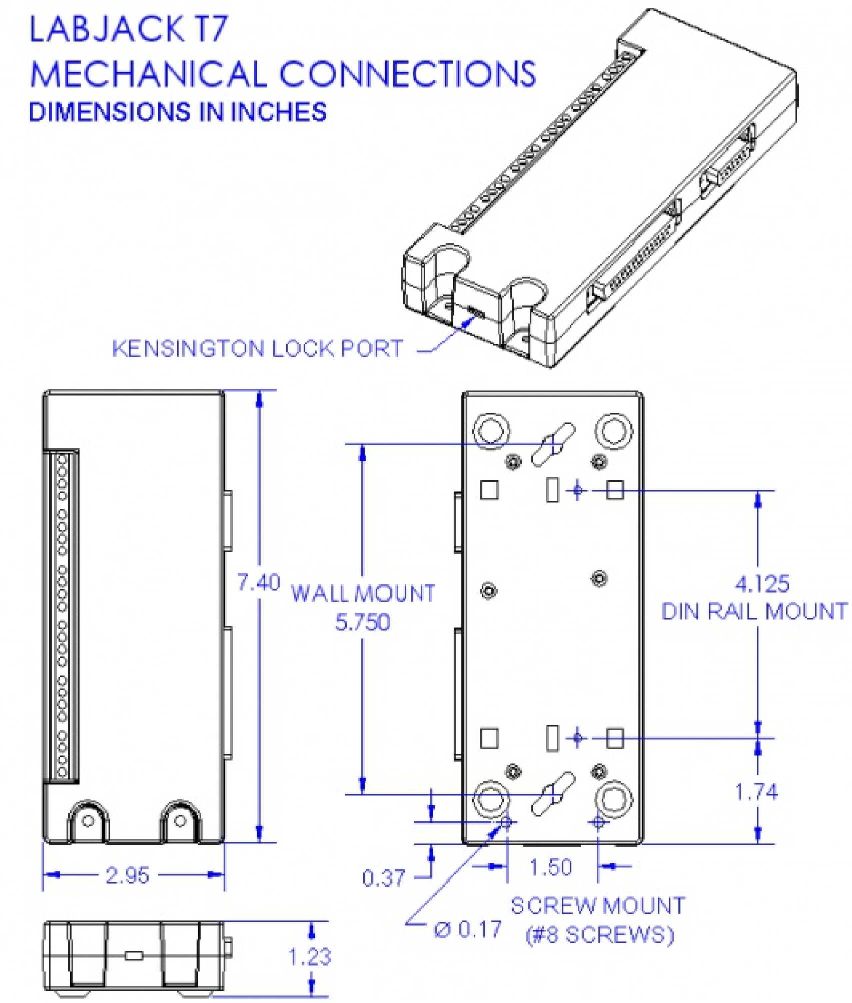
	J5					
	1	V+		2	GND	
	J5 OEM Pin-Header					

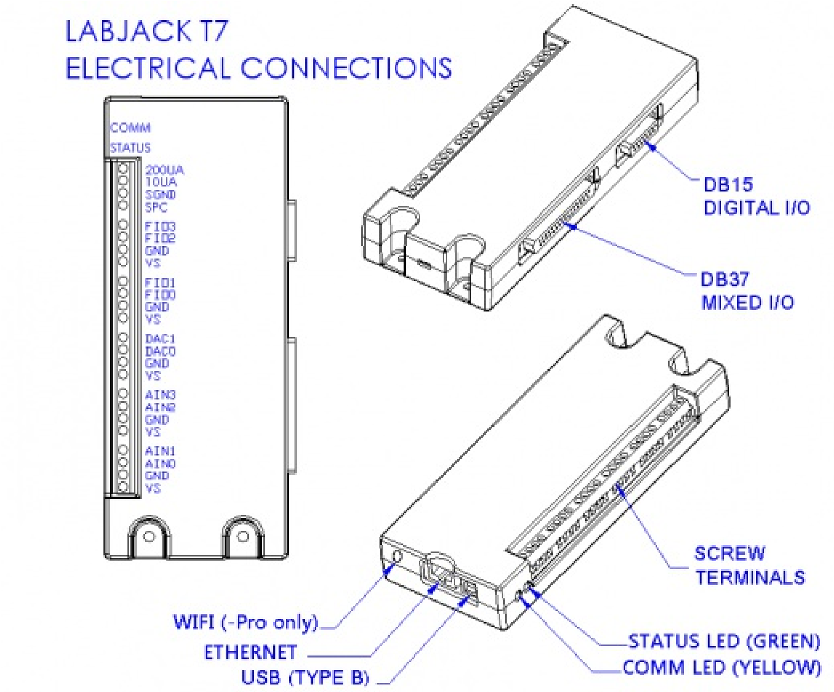
22.0 Enclosure and PCB Drawings

See below drawings of the T7.

The square holes on the back of the enclosure are for DIN rail mounting adapters (TE Connectivity(formerly Tyco) part #TKAD).

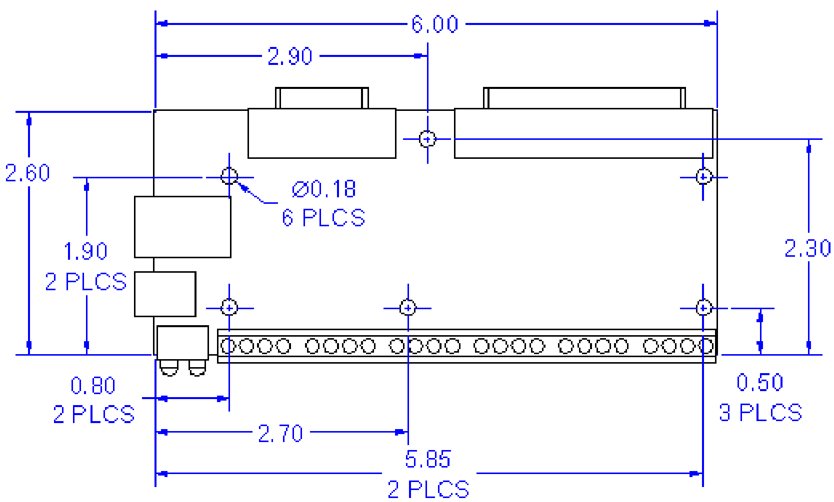
CAD drawings of the T7 enclosure are attached to the bottom of this page. (DWG, DXF, IGES, STEP)





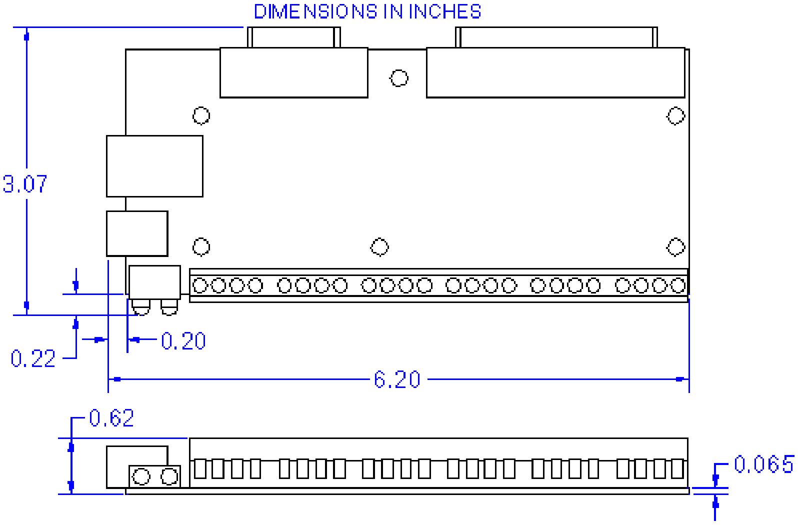
PCB MOUNTING HOLE DIMENSIONS

DIMENSIONS IN INCHES



PCB AND CONNECTOR FOOTPRINT

DIMENSIONS IN INCHES

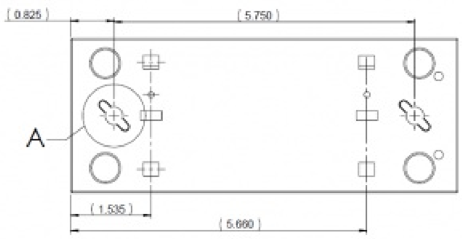


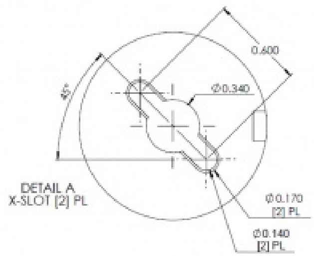
LabJack T7 Enclosure

6-2-2012

Note: Dimensions in parentheses are reference only.

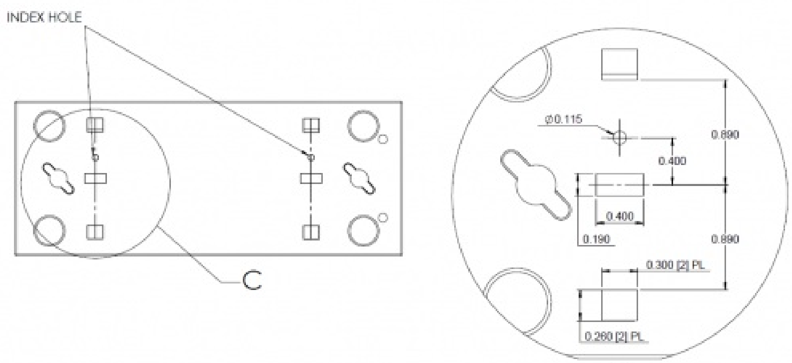
T7 ENCLOSURE BASE





DETAIL A
X-SLOT [2] PL

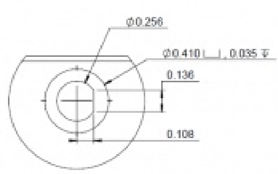
T7 ENCLOSURE BASE - DIN CLIP DETAILS



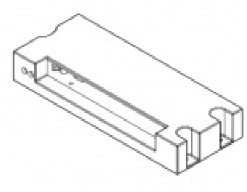
C

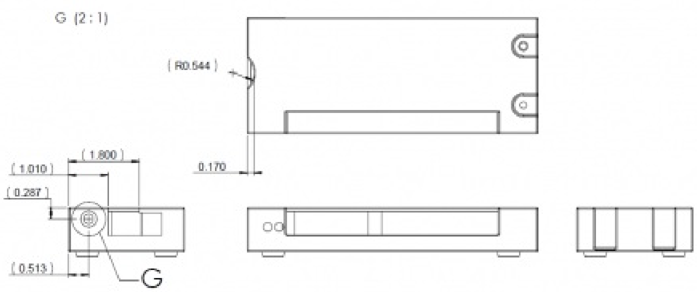
DIN RAIL CLIP MOUNTING HOLES

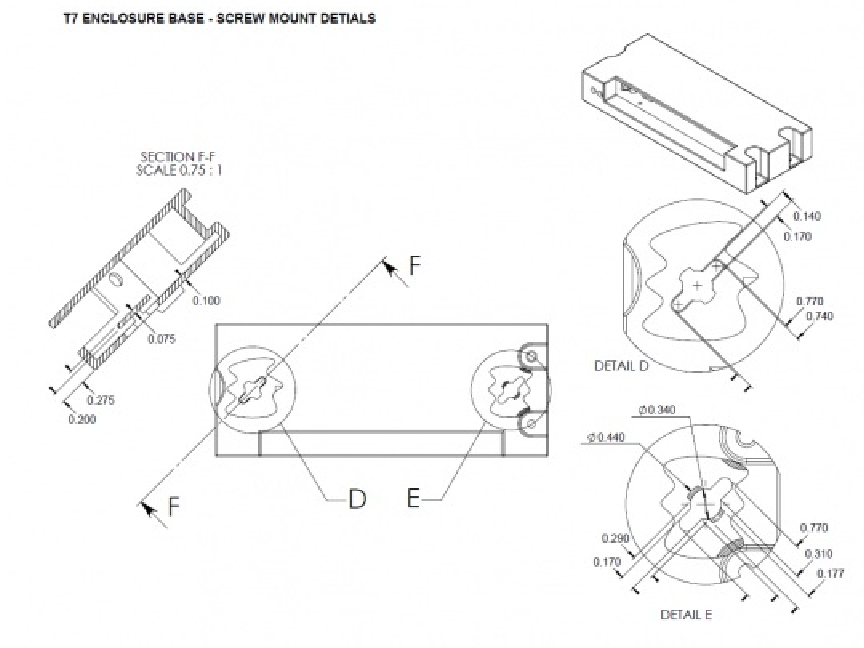
T7 ENCLOSURE TOP
CLEARANCE FOR SHROWDED ETHERNET CABLE
ANTENNA NUT BOSS DETIALS



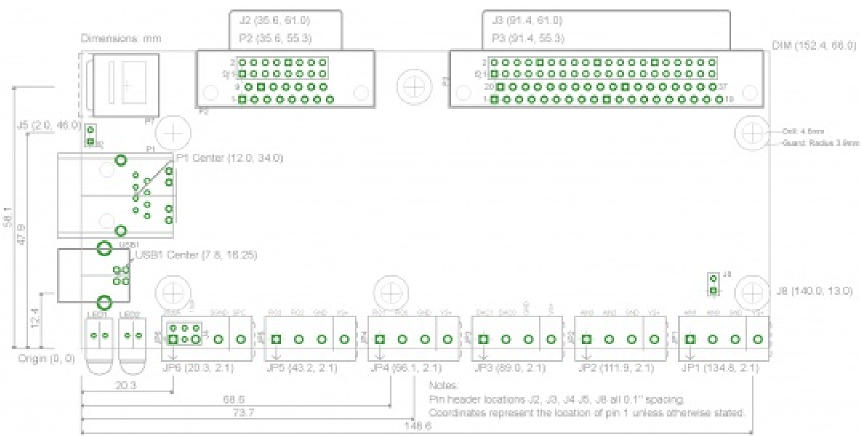
G (2 : 1)







T7 OEM PCB Dimensions



File attachment:

- T7 Enclosure DWG
- T7 Enclosure DXF
- T7 Enclosure IGS
- T7 PCB Dimensions
- T7 Enclosure STEP
- T7 Pro Enclosure DWG
- T7 Pro Enclosure DXF
- T7 Pro Enclosure IGS
- T7 Pro Enclosure STEP

23.0 Packaging Information

Package Contents:

The normal retail packaged T7 or T7-Pro consists of:

- T7 (-Pro) unit itself in red enclosure
- USB cable (6 ft / 1.8 m)
- Ethernet Cable (6ft / 1.8m)
- USB 5V power supply
- Screwdriver
- Antenna (T7-Pro only)

Other package details:

There is no software CD included, so an Internet connection is needed to download software as needed. Go to the T7 Support Homepage (labjack.com/support/t7) to get started.

Contact support@labjack.com for additional information on shipping.

Package size: 10" x 7" x 3"
Package wt: 1.2lb

24.0 Firmware Revision History

The latest T7 firmware is listed on the T7 firmware page. You will need the Kipling Program to load the firmware files onto a T7. Also use Kipling to identify the current WiFi and Firmware versions your T7 is using.

WARNING: If you are currently using WiFi firmware 2.23 and would like to upgrade, please contact us. We will swap your T7 with a newer version that can be upgraded in the field. T7 Firmware 0.9013 is that last version that supports WiFi 2.23

Change Log

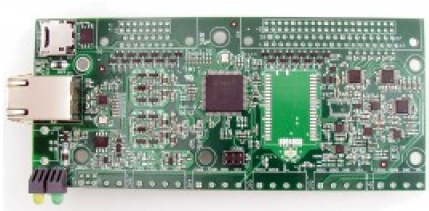
1.0000: Stream features added and timing calibrated. Stream bug fixes. I2C modbus interface updated. Added z-phase support to quadrature. Digital EF bug fixes and feature additions.

Older Versions

Appendix A - OEM Versions

For pricing/ordering, go to the main T7 Product Page.

The OEM version of the T7 and T7-Pro are shown below. The enclosure, and most connectors are not installed on the OEM version, which allows customers to choose custom connectors.

The following list describes parts that we know to be compatible with the T7 OEM hole patterns. Simply select a connector from each category, and we can order the parts and construct a custom OEM. Custom OEM boards carry additional cost, but they are often necessary for specialized enclosures, and seamless integration with other products.

Of course there are many other connector options available; we can just as easily order/install something not mentioned below. Please don't hesitate to contact us. Typically we charge $0.10 in labor per solder connection, plus the price of each part to be installed.

The PCB Dimensions can be found in the Enclosure and PCB Drawings section.

USB

The USB connector is not installed on the T7 OEM. Reference the T7 PCB dimensions for mechanical mating details. Many through-hole USB-B (USB 2.0) connectors are compatible. On Shore Technology Inc USB-B1HSW6, FCI 61729-0010BLF, and TE Connectivity 292304-2 are all good options.

A special high-retention connector such as the Samtec USBR-B-S-S-O-TH can also be used, but it does take a good deal of force to unplug a cable from these so they are only recommended when you don't want to unplug very often.

It also possible to simply solder the wires directly, using the image below as a reference.

If you have a shield wire, it can be connected to either of the large mounting holes.

J5 - Alternate Power Supply

Through the use of J5, users can supply 5V to the T7 if a USB connection is not required. The square shaped pad is V+, and the circular pad is GND. It is useful for individuals who only need Ethernet or WiFi. The J5 connector is a 2 pin 0.1" pitch rectangular header. To prevent accidentally switching V+ and GND, use a keyed connector such as TE Connectivity 3-641215-2.

J5				
1	V+		2	GND
J5 OEM Pin-Header				

The 5V supply from J5 goes through R21 (0.1 ohms) and then connects to the device-wide VS bus. The 5V supply from USB goes through R15 (0.1 ohms) and then connects to VS. On the T7-T7Pro, **R15 & R21 are both installed by default, and thus the connections for both sources are essentially shorted to each other,** and both should not be connected at the same time as one could back-feed the other. If you are going to connect to J5, and there is a possibility of power at the USB connection also, remove R15. You can also replace R15 and R21 with diodes (SMA package) to prevent back-feeding, but even Schottky diodes will have voltage drop that needs to be considered.

Ethernet

The same Ethernet connector is installed on all versions of the T7 due to the inherent magnetic complexities. However, it is possible to 'bring out' a duplicate Ethernet jack to any custom enclosure with one of the following:

- A short Ethernet cable segment and an RJ45 coupler(Plug to Plug). These couplers come in a few varieties: Free hanging (in-line), Chassis Mount, Panel Mount, Bulkhead, Wall Plate, etc. Conec 33TS3101S-88N and Emerson 30-1008KUL are both good options.
- A RJ45 Jack to Plug cable, which is just a standard Ethernet plug on one end, and a Jack (female) on the other end. Again, these come in a wide variety of mounting styles, the simplest of which is the panel mount. TE Connectivity 1546414-4 and Amphenol RJFEZ2203100BTX are both good options.

If selecting your own Ethernet interconnect, insure that it is RJ45, straight-through, and without magnetics.

WiFi

The WiFi antenna jack is a snap-on ultra miniature coaxial connector called u.fl or ipex. The T7-Pro-OEM includes an antenna, but an OEM customer may want to purchase a cable extension that snaps directly to the board. Any antenna with an ultra miniature coaxial connector should work. Most of the time a cable assembly such as TE Connectivity 2032440-1 is used to convert the u.fl connector into an RP-SMA Female Jack.

JP1-JP6 - Screw terminal Locations

The screw terminals are not installed on the OEM T7. Customers will typically use the rectangular header locations (J2, J3) instead of the screw terminals. However, if a different screw terminal style is required, it is possible to buy an OEM T7 and order a custom variety. The screw terminal holes are compatible with almost all 4 position, 0.198" (5.00mm) pitch terminal blocks. A Weidmuller 9993300000 works quite well, and accepts 14-24 AWG wire.

P2, P3 - DB(D-Sub) Locations

The DB15 and DB37 connectors are not installed on an OEM T7. Customers will typically use the rectangular header locations (J2, J3) instead of the DB connectors. However, if a different DB mating style is required, it is possible to buy an OEM T7 and order a custom variety. The DB connectors are standard D-Sub two row receptacles(female sockets), through hole, 15 pin, and 37 pin. The following represent a few valid options.

- FCI 10090099-S154VLF
- Sullins Connector Solutions SDS101-PRW2-F15-SN13-1
- FCI 10090099-S374VLF
- Sullins Connector Solutions SDS101-PRW2-F37-SN83-6

J2, J3 - Header Locations

Connectors J2 and J3 provide pin-header alternatives to the DB15 and DB37 connectors. The J2 and J3 holes are always present, but are obstructed when the DB15 and DB37 are installed.

J2 - 16 position, 2 row, 0.1" pitch, male pin rectangular header

- Unshrouded - Harwin Inc M20-9980846
- Unshrouded 3x Taller - Samtec Inc TSW-108-17-T-D
- Shrouded, Gold Finish - On Shore Technology Inc 302-S161
- Shrouded, Right Angle - TE Connectivity 1-1634689-6

J3 - 40 position, 2 row, 0.1" pitch, male pin rectangular header

- Unshrouded - Harwin Inc M20-9762046
- Unshrouded 3x Taller - Samtec Inc TSW-120-17-T-D
- Shrouded, Gold Finish - On Shore Technology Inc 302-S401
- Shrouded, Right Angle - TE Connectivity 5103310-8
- Shrouded, Gold-Palladium Finish - TE Connectivity 5104338-8

Sometimes customers order tall pin headers that mate directly to a separate custom PCB. Refer to the pinout details below for electrical connections.

J2					
1	GND	2	VS		
3	CIO0	4	CIO1		
5	CIO2	6	CIO3		
7	GND	8	EIO0		
9	EIO1	10	EIO2		
11	EIO3	12	EIO4		
13	EIO5	14	EIO6		
15	EIO7	16	GND		
J2 OEM Pin-Header					
J3					
1	GND	2	GND	3	PIN20 (10uA)
4	PIN2 (200uA)	5	FIO7	6	FIO6
7	FIO5	8	FIO4	9	FIO3
10	FIO2	11	FIO1	12	FIO0
13	MIO0/CIO0	14	MIO1/CIO1	15	MIO2/CIO2
16	GND	17	Vs	18	Vm-
19	Vm+	20	GND	21	DAC1
22	DAC0	23	GND	24	AIN13
25	AIN12	26	AIN11	27	AIN10
28	AIN9	29	AIN8	30	AIN7
31	AIN6	32	AIN5	33	AIN4
34	AIN3	35	AIN2	36	AIN1
37	AIN0	38	GND	39	GND
40	GND				
J3 OEM Pin-Header					

J4 - Constant Current Sources

Since the screw terminals are not installed on an OEM T7, the J4 header location can be used to gain access to the constant current sources. Any 6 position 0.1" pitch rectangular header will work.

J4			
1	200uA	2	GND
3	GND	4	GND
5	10uA	6	VS
J4 OEM Pin-Header			

J8 - Mechanical

The J8 pin header location is purely for mechanical support for that region of the board. There are no electrical connections to this area. It is a 2 position 0.1" pitch rectangular header.

Pricing/Ordering

For pricing & ordering, go to the main T7 Product Page.

Appendix B - Revision History

Revision C (Oct 2013) Added calibration constants information. Modified URLs. Updated many links to related support material. Updated DIO information.

Revision B (April 2013) Added many descriptions of Digital I/O extended features. Modified a bunch of URLs.

Revision A (February 2013) Original data sheet for the T7 family of devices.