LabJack Measurement & Automation Simplified

LJTick-DAC Datasheet

The LJTick-DAC works with the T7 and with any UD family device (U3/U6/UE9), except for the oldest U3 hardware revision 1.20 (U3A, pre 2007), as that U3 did not support I2C. The LJTick-DAC does not work with the U12.

(/sites/default/files/2009/08/LJTDAC-figure1.png)

Figure 1: LJTick-DAC

(/sites/default/files/2009/08/LJTDAC-figure2.png)

Figure 2: LJTick-DAC With U3

The LJTick-DAC (LJTDAC) is an analog output expansion module. It provides a pair of 14-bit analog outputs with a range of ±10 volts. The 4-pin design plugs into any of the standard DIO/DIO/GND/VS screw terminal blocks on the LabJack, and thus up to 10 of these can be used per device to add 20 analog outputs.

The update rate of the LJTDAC is limited by the communication time between the host and the device. See Section 3.1 of the U3/U6/UE9 User's Guide for detailed information, but it generally takes about 1 ms to do an update via USB "high-high" or Ethernet, while it takes about 4 ms via other USB connections. Only 1 DAC channel can be updated per low-level communication. That means, for instance, that if updates are done at the 1 ms rate to build a 100 Hz sine wave, there will only be about 5 updates per half-cycle of the waveform and it will appear to be a smooth sine. With a 10 Hz sine wave, however, there will be about 50 updates per half-cycle and the waveform will appear much smoother.

The pins shown on the right side of the LJTDAC (Figure 1) connect to the LabJack. The VS/GND pins power the LJTDAC, while the DIOA/DIOB pins are used for digital communication (I2C) between the LJTDAC and LabJack. DIOA is the serial clock (SCL) and DIOB is the serial data (SDA). Following are descriptions of the screw-terminal connections:

GND: Connected directly to LabJack ground (GND).

VS: This is the same 5 volt output as the VS terminals on the LabJack itself. This is an output terminal, not an input. It can be used to provide 5 volt (nominal) power as needed.

DACA/DACB: Output of each 14-bit digital-to-analog converter.

Low-Level I2C Communication

The LJTDAC has a non-volatile 128-byte EEPROM (Microchip 24C01C) on the I2C bus with a 7-bit address of 0x50 (d80), and thus an 8-bit address byte of 0xA0 (d160). Bytes 0-63 are available to the user, while bytes 64-127 are reserved.

EEPROM Address	Description	Nominal Value
0-63	User Area	
64-71	DACA Slope	3.1586E+03 bits/volt
72-79	DACA Offset	3.2624E+04 bits
80-87	DACB Slope	3.1586E+03 bits/volt
88-95	DACB Offset	3.2624E+04 bits
96-99	Serial Number	
100-127	Reserved	

The slopes and offsets are stored in 64-bit fixed point format (signed 32.32, little endian, 2's complement). The serial number is simply an unsigned 32-bit value where byte 96 is the LSB and byte 99 is the MSB.

The DAC (digital-to-analog converter) chip on the LJTDAC is the LTC2617 (linear.com) with a 7-bit address of 0x12 (d18), and thus an 8-bit address byte of 0x24 (d36). The data is justified to 16 bits, so a binary value of 0 (actually 0-3) results in minimum output (~-10.3 volts) and a binary value of 65535 (actually 65532-65535) results in maximum output (~10.4 volts).

For more information about low-level communication with the LJTDAC, see the I2C example in the VC6_LJUD (/support/ud/examples/vc6) archive or see the Linux example.

UD Communication

The LJTick-DAC works with any UD family device (U3/U6/UE9), except for U3 hardware revision 1.20 (U3A), as that U3 did not support I2C. It also does not work with the U12 (which is not a UD family device).

The LabJack UD driver for Windows (V2.76+) has special support for the LJTDAC. First, the following special channel is used with the put config IOType to specify where the LJTDAC is connected to the LabJack:

```
LJ_chTDAC_SCL_PIN_NUM  //Used with LJ_ioPUT_CONFIG
```

Then there is one IOType used for all further communication with the LJTDAC. The value of the Channel parameter used with this IOType is always one of the following 7 special channels:

```
LJ_ioTDAC_COMMUNICATION  //Main IOType.

LJ_chTDAC_SERIAL_NUMBER  //Read-only.
LJ_chTDAC_READ_USER_MEM  //x1 is array of 64 bytes.
LJ_chTDAC_WRITE_USER_MEM //x1 is array of 64 bytes.
LJ_chTDAC_READ_CAL_CONSTANTS //x1 is array of 4 doubles.
LJ_chTDAC_WRITE_CAL_CONSTANTS //x1 is array of 4 doubles.
LJ_chTDAC_UPDATE_DACA  //Pass DAC voltage in Value parameter.
LJ_chTDAC_UPDATE_DACB  //Pass DAC voltage in Value parameter.
```

Typical operation consists of simply setting the pin number for SCL and then updating DAC channel A and/or B:

```
//Tell the driver that SCL is on FIO0.  The driver then assumes that SDA is on FIO1.
//This is just setting a parameter in the driver, and not actually talking
//to the hardware, and thus executes very fast.
ePut(lngHandle, LJ_ioPUT_CONFIG, LJ_chTDAC_SCL_PIN_NUM,0,0);

//Set DACA to 1.2 volts.  If the driver has not previously talked to an LJTDAC
//on FIO0/FIO1, it will first retrieve and store the cal constants.  The
//low-level I2C command can only update 1 DAC channel at a time, so there
//is no advantage to doing two updates within a single add-go-get block.
ePut(lngHandle, LJ_ioTDAC_COMMUNICATION, LJ_chTDAC_UPDATE_DACA, 1.2, 0);

//Set DACB to 2.3 volts.
ePut(lngHandle, LJ_ioTDAC_COMMUNICATION, LJ_chTDAC_UPDATE_DACB, 2.3, 0);
```

For more information about UD communication with the LJTDAC, see the LJTDAC example in the VC6_LJUD (/support/ud/examples/vc6) archive.

T7 & LJM Communication

The T7 has special registers available that make controlling the LJTick-DAC very easy. Go to the Modbus Map (/support/modbus/map) and set Tags = TDAC to see the registers.

LJTick-DAC Analog Output Registers

Name	Start Address	Type	Access	Default
TDAC#(0:22)	30000	FLOAT32	W	0
TDAC_SERIAL_NUMBER	55200	UINT32	R	0

TDAC#(0:22)

Update a voltage output on a connected LJTick-DAC accessory. Even TDAC# = DACA, Odd TDAC# = DACB. For instance, if LJTick-DAC accessory is connected to FIO2/FIO3 block on main device, TDAC2 corresponds with DACA, and TDAC3 corresponds with DACB.

Names	Addresses
TDAC0, TDAC1, TDAC2, Show All (#)	30000, 30002, 30004, Show All (#)

TDAC_SERIAL_NUMBER

Returns the serial number of an LJTick-DAC, and forces a re-read of the calibration constants. Which LJTDAC is determined by the last write to TDAC# ... whether it was successful or not.

Updating an output through the LJM library is easy. For example, the following line of code:

```
err = LJM_eWriteName(handle, "TDAC11", 7.5)
```

... sets the voltage on the DAC channel associated with DIO11 (aka EIO3). That means DACB on the LJTick-DAC connected to DIO10/11 (aka EIO2/3).

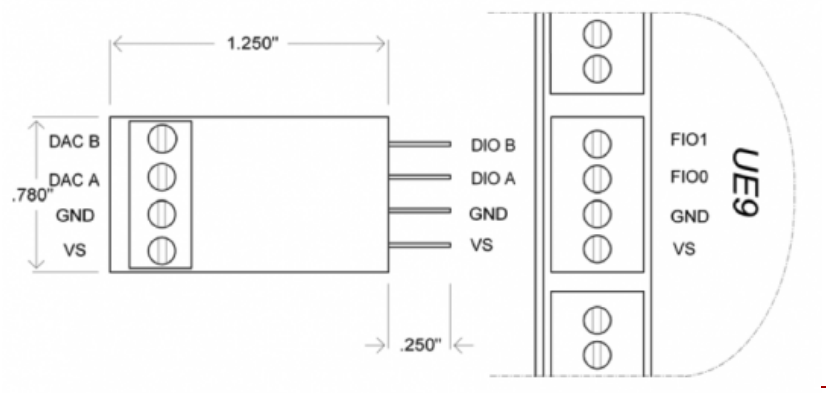
Specifications: (25 deg C, VS = 5 volts)

Parameter	Conditions	Min	Typical	Max	Units
General					
Supply Voltage		4.5	5	5.5	volts
Supply Current	Vout = 0, No Load		5		mA
	Both Outputs @ 3 mA		29		mA
Operating Temperature		0		70	°C
DIO					
Pull-Up Resistors	To VS		5100		Ω
Low Level Input Voltage				0.3*VS	volts
High Level Input Voltage		0.7*VS			volts
Low Level Output Voltage	Sink Current = 3 mA	0		0.4	volts
Clock Frequency (DIOA)				400	kHz
DAC Outputs					
Typical Output Range		-10.3		10.4	volts
Power-Up Output Voltage			0.05		volts
Resolution				14	bits
				1.22	mV
Accuracy			0.05	0.5	% FS
Differential Linearity Error				±1	counts
Integral Linearity Error			±5	±16	counts
Temperature Drift			20		ppm/°C
Update Time (1)			1		ms
Update Rate (1)			1000		Hz
Slew Rate			0.1		V/µs
Output Impedance			0.1		Ω
Output Current (2)	Total for both channels			10	mA

(1) The update time is similar to the numbers found in Section 3.1 of the U3/UE9 User's Guide. The time is typically about 1 ms over Ethernet or USB "high-high", and typically about 4 ms over USB "other".

(2) This is the current limit for both channels combined. The first thing you notice as you get close to the current limit is that the minimum output voltage increases, and this effect will be worse if your VS supply voltage is low. For example, at 10 mA, the minimum output is typically about -9.5 volts.

Dimensions:



(/sites/default/files/imagecache/support_page/2009/08/LJTDAC-dimentions.png)

LJTickDAC Testing Utility

For your convenience, the LJTickDAC (/support/accessories/ljtickdac) testing utility for UD family devices is available for download.

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 Tick DAC
Model Number: LJTick-DAC
conforms to the following Product Specifications:
EMC Directive: 89/336/EEC
EN 55011 Class A
EN 61326-1: General Requirements

File attachment:

LJTDACdatasheet.pdf (http://labjack.com/sites/default/files/2009/07/LJTDACdatasheet.pdf)

Comments

#1 (#comment-2654)
Submitted by steve.nicholson on Mon, 2014-08-18 17:15.

If I have multiple LJTDACs, do I call

ePut(lngHandle, LJ_ioPUT_CONFIG, LJ_chTDAC_SCL_PIN_NUM,pinNum,0)

before every call to

ePut(lngHandle, LJ_ioTDAC_COMMUNICATION, LJ_chTDAC_UPDATE_DACA, 1.2, 0)

#2 (#comment-2656)
Submitted by LabJack Support on Tue, 2014-08-19 10:39.

Yes, that is correct. Realize that the ePut call to set the SCL pin number is just setting a flag in the driver, not talking to hardware, so it takes relatively no time at all.

#3 (#comment-2653)
Submitted by steve.nicholson on Mon, 2014-08-18 16:18.

For more information about UD communication with the LJTDAC, see the LJTDAC example in the VC6_LJUD archive.

So far I haven't been able to find the VC6_LJUD archive. Can you turn that reference into a link?

#4 (#comment-2655)
Submitted by LabJack Support on Tue, 2014-08-19 10:35.

I added the links. To jump to it quickly from anywhere on our site do a search with the term "vc6".

#5 (#comment-2184)

Submitted by Anonymous (not verified) on Sat, 2014-02-01 08:03.

I need for the output DACA on the LJTick-DAC to go to +5V when the labjack (U6-PRO) is powered up. The LJTickDAC is connected to FI01 and FI00. Is there a way to do this?

#6 (#comment-2185)

Submitted by LabJack Support on Mon, 2014-02-03 08:45.

There is no way to do this in hardware. Your best option is to write software that is always running so that as soon as the U6 enumerates you send a command that sets the LJTDAC to 5 volts.

The chip on the LJTDAC powers up at mid-scale, which is about 0 volts.

#7 (#comment-2064)

Submitted by Nitzan (not verified) on Wed, 2013-12-04 01:34.

Hi,

I'm controlling LabJack U3HV with two LJTickDAC using
matlab. The voltage supplied from the LJTickDAC determines the position
of a mechanical stage.

When disconnecting the U3HV from matlab,
the output voltage doesn't change which is good for my needs. However, when reconnecting matlab to the
LabJack, I don't know what voltage the LJTickDAC supplies, so I don't
know the position of the mechanical stage.

I am looking of a way to READ
the set voltage of the LJTickDAC, without changing it, so my stage
won't move.

Thanks!

#8 (#comment-2065)

Submitted by LabJack Support on Wed, 2013-12-04 08:16.

The DAC chip used in the LJTick-DAC does not support any reads, only writes. Your only option then is to store the last written values somewhere else. You could store them in the user memory on the LJTDAC (/support/ljtick-dac/datasheet), the user memory on the U3 (/support/u3/users-guide/4.3.14), or the easiest place is to store the values on your PC.

The one thing this will not account for is if the LJTDAC is power-cycled, in which case it will go to mid-scale (~0 volts). You could measure the voltages with analog inputs to try to detect this situation.

#9 (#comment-761)

Submitted by dwhynman on Wed, 2011-12-07 07:05.

I just ordered one of these yesterday sine I have filled up the DAC0 and DAC1 on my U6 pro. I was using the eDAC lab view code to send out a voltage to my motor controller and a solenoid. I need to send a voltage to a proportional valve now. If I use this with the FIO inputs, which code example should I use in LabView?

Thanks

#10 (#comment-762)

Submitted by LabJack Support on Wed, 2011-12-07 07:40.

You need to make calls as shown above in the "UD Communication" section. I suggest you look at "UE9 LJTDAC Example.vi", or any of the other LJTDAC examples available for the U3 and UE9.

#11 (#comment-786)

Submitted by dwhynman on Wed, 2011-12-21 17:43.

Is there an example code for the U6? or can I use the U9 code with the U6?

Thanks

#12 (#comment-787)

Submitted by LabJack Support on Thu, 2011-12-22 10:52.

There is not a U6-specific example. Just make a copy of one of the others and change the open call. For example, use "U3 LJTDAC Example.vi" and change "LJ_dtU3" to LJ_dtU6".

#13 (#comment-400)

Submitted by Dave in Mpls. (not verified) on Sat, 2011-03-26 08:29.

My Delphi code for setting Tick Dac does not compile:

there is no LJ_ioTDAC_COMMUNICATION variable defined in LJJDelphi.pas wrapper.

On a lark I tried ..ioI2C.. compiled but no DACA or DACB voltages.

The TickDac utility works, thus TICKDAC is operational.

```
//Version number of the LJUDDelphi wrapper
LJUDDELPHI_VERSION = 0.2;

//UD Driver version this was last tested with
DRIVER_VERSION = 2.67;
```

Thanks,

Dave

#14 (#comment-405)

Submitted by LabJack Support on Mon, 2011-03-28 18:17.

Get the latest Delphi wrapper (http://labjack.com/support/ud/examples/delphi) which I just updated today. It has been updated to LJUDDELPHI_VERSION 0.3 and DRIVER_VERSION 3.25, and will have the missing LJ_ioTDAC_COMMUNICATION constant. Hopefully the new wrapper should help fix your problem.

Add new comment (/comment/reply/168#comment-form)