

LabJack

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

[Home](#) > [Support](#) > [Datasheets](#) > [T-Series Datasheet](#) > [13.0 Digital I/O](#) > 13.6 1-Wire

13.6 1-Wire [T-Series Datasheet]

[Log in](#) or [register](#) to post comments

Overview

1-Wire is a serial protocol that uses only one data line. Multiple devices can be connected to a single 1-Wire bus and are differentiated using a unique 64-bit number referred to as ROM.

1-Wire is considered an advanced topic. A good knowledge of the protocol is recommended. Troubleshooting may require a logic analyzer or oscilloscope.

The [1-Wire app-note](#) is a good place to start.

1-Wire Support

T-series devices can send and receive data over the 1-Wire bus. The below sections cover the necessary hardware setup considerations and describe how to use the various operations that can be performed over the 1-Wire bus.

[Lua scripting](#) is often convenient for serial applications. For example, a script can run the serial communication at a specified interval, and put the results in USER_RAM registers. The host software can read from the USER_RAM registers when convenient. This puts the complications of serial communication in a script running on the T-series device. The Lua examples contain several serial communication examples.

Hardware

Devices on the 1-Wire bus need to be connected to GND, to Vs, and to the data line DQ. DQ also needs a pullup resistor of 2.2-4.7 k Ω to Vs.

FIO lines cannot be used for 1-Wire. They have too much impedance, which prevents the signal from reaching the logic thresholds.

T-series devices supports a DPU (dynamic pull up). A dynamic pull up uses an external circuit such as a transistor to provide extra power to the DQ line at proper times. This can be helpful if the line is large or you are using parasitic power.

Configuration

ONEWIRE_DQ_DIONUM: This is the DIO line to use for the data line, DQ.

ONEWIRE_DPU_DIONUM: This is the DIO line to use for the dynamic pullup control.

ONEWIRE_OPTIONS: A bit-mask for controlling operation details:

- bit 0: Reserved, write 0.
- bit 1: Reserved, write 0.
- bit 2: DPU Enable. Write 1 to enable the dynamic pullup.
- bit 3: DPU Polarity. Write 1 to set the active state as high, 0 to set the active state as low.

ONEWIRE_FUNCTION: This controls how the ROM address of 1-Wire devices will be used.

ONEWIRE_NUM_BYTES_TX: The number of bytes to transmit to the device. Has no affect when the ROM function is set to Search or Read.

ONEWIRE_NUM_BYTES_RX: The number of bytes to read from the device. Has no affect when the ROM function is set to Search or Read.

ONEWIRE_ROM_MATCH_H: The upper 32-bits of the ROM of the device to attempt to connect to when using the Match ROM function.

ONEWIRE_ROM_MATCH_L: The lower 32-bits of the ROM of the device to attempt to connect to when using the Match ROM function.

ONEWIRE_PATH_H: Upper 32-bits of the search path.

ONEWIRE_PATH_L: Lower 32-bits of the search path.

ROM Functions

0xF0: Search – This function will read the ROM of one device on the bus. The ROM found is placed in ONEWIRE_SEARCH_RESULT and if other devices were detected the branch bits will be set in ONEWIRE_ROM_BRANCHS_FOUND.

0xCC: Skip – This function will skip the ROM addressing step. For this to work properly only one device may be connected to the bus.

0x55: Match – When using this function data will be sent to and read from a device whose ROM matches the ROM loaded into the ONEWIRE_ROM_MATCH registers.

0x33: Read – Reads the ROM of the connected device. For this to work properly only one device may be connected to the bus.

Sending data

When using the Match or Skip Rom functions data can be sent to the device. To do so, set the number of bytes to send by writing to ONEWIRE_NUM_BYTES_RX and write the data to ONEWIRE_DATA_RX.

Reading data

When using the Match or Skip Rom functions data can be read from the device. To do so, set the number of bytes to send by writing to ONEWIRE_NUM_BYTES_TX and write the data to ONEWIRE_DATA_TX.

Example

Configure the T4/T7's 1-Wire interface, and obtain a temperature reading from a DS18B22.

Configuration: Write the common configuration that will not change; the DQ line, DPU, and options. For this example we will use EIO6 (14) as DQ, and the DPU will be left disabled.

```
ONEWIRE_DQ_DIONUM = 14
ONEWIRE_DPU_DIONUM = 0
ONEWIRE_OPTIONS = 0
```

Read ROM: The 64-bit ROM can be read from the device using the Read ROM function if it is the only device on the bus.

```
ONEWIRE_FUNCTION = 0x33
ONEWIRE_GO = 1
```

The T4/T7 will read the ROM from the connected device and place it in `ONEWIRE_SEARCH_RESULT`. This test resulted in ROM code 0x1D000005908D4728

Search for ROM: If there is more than one device on the bus the search function can be used to find the ROM of one of the devices. Note that this method does not provide any information about which device has the ROM discovered.

```
ONEWIRE_PATH = 0
ONEWIRE_FUNCTION = 0xF0
ONEWIRE_GO = 1
```

The LabJack will perform the 1-Wire search function. If a ROM is found it will be placed in `ONEWIRE_ROM_SEARCH_RESULT` and any branches detected will be indicated in `ONEWIRE_BRANCHES`. The `ONEWIRE_PATH` field can be used to direct the LabJack to take a different path in subsequent searches.

Results:

ROM - 0x1D000005908D4728
Branches - 0x0000000000000002

Now repeat the search with path set to 2.

Results:

ROM - 0xFF00000024AD2C22
Branches - 0x0000000000000002

The search can be repeated to find the ROM codes of all devices on the bus.

Write start conversion command to the device:

Do instruct the sensor to start a reading we need to match the device's ROM and send one data byte. The data byte contains the instruction 0x44.

```
ONEWIRE_FUNCTION = 0x55
ONEWIRE_ROM = 0x1D000005908D4728
ONEWIRE_NUM_BYTES_TX = 1
ONEWIRE_DATA_TX = [0x44]
ONEWIRE_GO = 1
```

The sensor will now start a conversion. Depending on the sensor and it's settings up to 500 ms may be needed to complete the conversion.

Read conversion result from the device: After a conversion has been complete we can begin the reading process. This time we need to write the read instruction which is 0x44 and then read 9 bytes of data.






```
ONEWIRE_FUNCTION = 0x55
ONEWIRE_ROM = 0x1D000005908D4728
ONEWIRE_NUM_BYTES_TX = 1
ONEWIRE_NUM_BYTES_RX = 9
ONEWIRE_DATA_TX = [0xBE]
ONEWIRE_GO = 1
```

We can now read the 9 bytes from ONEWIRE_DATA_RX:
0x6A, 0x0A, 0x00, 0x00, 0x24, 0xAD, 0x2C, 0x22, 0x00

The 9 bytes contain the binary reading, a checksum, and some other information about the device. The devices used was set to 12-bit resolution so the conversion is 0.0625°C/bit. The binary result is $\text{data}[0] + \text{data}[1] * 256$. The binary temperature reading is $1 * 256 + 0x6A = 256 + 106 = 362$. To convert that to °C multiply by 0.0625. So the final temperature is 22.6 °C.

Register Listing

1-Wire Registers

Name		Start Address	Type	Access
 ONEWIRE_DQ_DIONUM	The data-line DIO number.	5300	UINT16	R/W
 ONEWIRE_DPU_DIONUM	The dynamic pullup control DIO number.	5301	UINT16	R/W
 ONEWIRE_OPTIONS	Controls advanced features. Value is a bitmask. bit 0: reserved, bit 1: reserved, ... 2: 1=DPU Enabled 0=DPU Disabled, bit 3: DPU Polarity 1=Active state is high, 0=Active state is low (Dynamic Pull-Up)	5302	UINT16	R/W
 ONEWIRE_FUNCTION	Set the ROM function to use. 0xF0=Search, 0xCC=Skip, 0x55=Match, 0x33=Read.	5307	UINT16	R/W
 ONEWIRE_NUM_BYTES_TX	Number of data bytes to be sent.	5308	UINT16	R/W

Name	Description	Start Address	Type	R/W Access
ONEWIRE_NUM_BYTES_RX	Number of data bytes to be received.	5309	UINT16	R/W
ONEWIRE_GO	Instructs the T7 to perform the configured 1-wire transaction.	5310	UINT16	W
ONEWIRE_ROM_MATCH_H	Upper 32-bits of the ROM to match.	5320	UINT32	R/W
ONEWIRE_ROM_MATCH_L	Lower 32-bits of the ROM to match.	5322	UINT32	R/W
ONEWIRE_ROM_BRANCHS_FOUND_H	Upper 32-bits of the branches detected during a search.	5332	UINT32	R
ONEWIRE_ROM_BRANCHS_FOUND_L	Lower 32-bits of the branches detected during a search.	5334	UINT32	R
ONEWIRE_SEARCH_RESULT_H	Upper 32-bits of the search result.	5328	UINT32	R
ONEWIRE_SEARCH_RESULT_L	Lower 32-bits of the search result.	5330	UINT32	R
ONEWIRE_PATH_H	Upper 32-bits of the path to take during a search.	5324	UINT32	R/W
ONEWIRE_PATH_L	Lower 32-bits of the path to take during a search.	5326	UINT32	R/W
ONEWIRE_DATA_TX	Data to be transmitted over the 1-wire bus. This register is a buffer.	5340	BYTE	W
ONEWIRE_DATA_RX	Data received over the 1-wire bus. This register is a buffer. Underrun behavior - buffer is static, old data will fill the extra locations, firmware 1.0225 changes this to read zeros.	5370	BYTE	R