北京迪阳世纪科技有限公司中国总代理 www.pc17.com.cn 01-62156134

LabJack T7-PRO

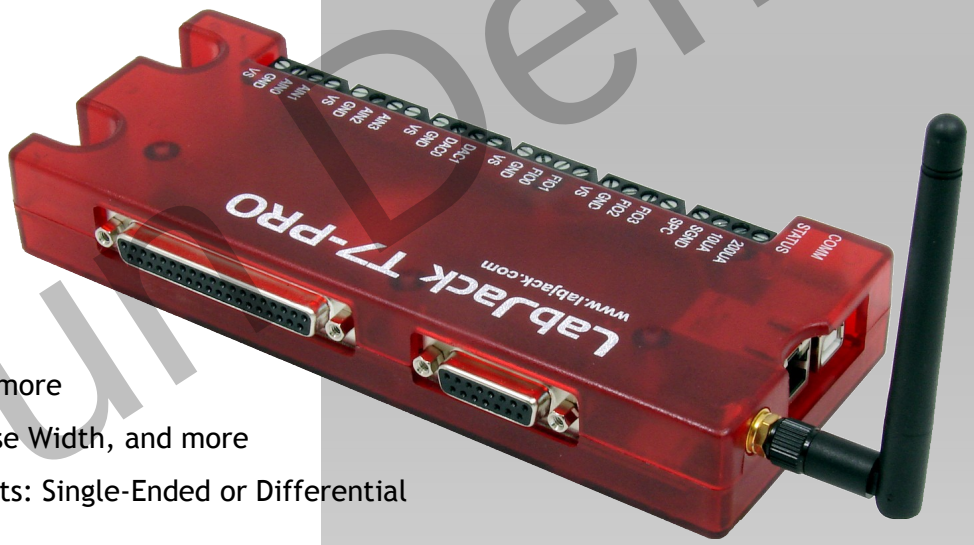
High Performance - WiFi, Ethernet, USB Multifunction DAQ

The T7-Pro combines our highest performance 24-bit analog inputs with the convenient Ethernet or WiFi communication interface.

HuDun Demo

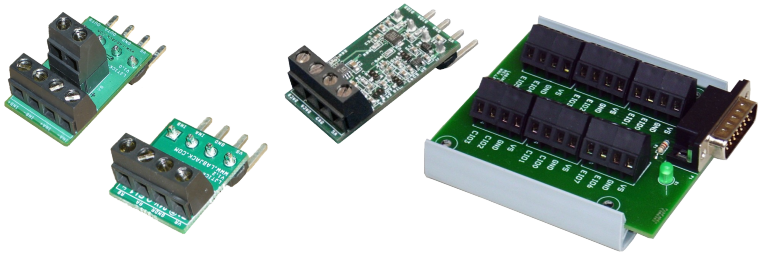
I/O Features

- **1uV noise-free** analog input resolution
- Add $150 expansion board for **84 analog inputs**
- **24-bit** low-speed ADC
- 23 Digital I/O
- 14 Analog Inputs
- Watchdog system
- Up to 10 counters
- Industrial range (-40 to +85C)
- 2 Analog Outputs (12bit 0-5V)
- Serial protocols: SPI, I2C, and more
- Up to 8 PWM, Quadrature, Pulse Width, and more
- Instrumentation amplifier inputs: Single-Ended or Differential



Other Highlights

- Each purchase includes **lifetime support**
- Several **free applications** to configure & test, and log data to file
- Example code in: **C/C++**, **C#**, **VB**, **Matlab**, **LabVIEW**, **Python**, **Java**, **Delphi**, **.NET** and more...
- Modbus TCP – Use any platform that supports TCP/IP, no driver needed!
- Free **cross-platform** driver – Extends/wraps the Modbus protocol for convenience.
- Expansion boards – Add ±10V DACs, current shunts, terminal boards, relay boards and more...



"I don't know of many other companies that provide such excellent service. I wouldn't hesitate to recommend your product to anyone."

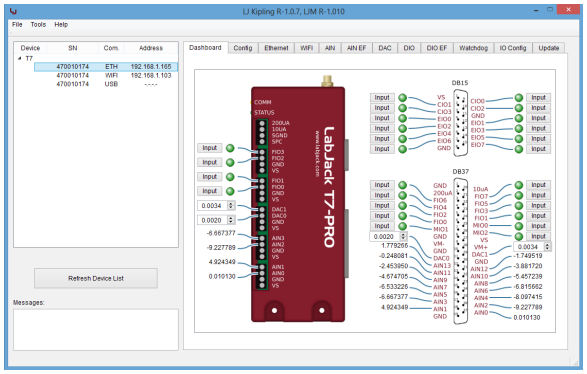
-Richard P.
Milwaukee School of Engineering

Software

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 PWM is all functionally the same, you simply provide a different address. The **LJM Library** provides names for each address, along with several other convenience functions.

- Log to file with **LJLogM** or **LJStreamM**
- Up to 1000Hz using LJLogM
- Up to 100kHz using LJStreamM

Test & Configure with Kipling



Python Example

```
from labjack import ljm
handle = ljm.openS("ANY", "ANY", "ANY")

name = "SERIAL_NUMBER"
result = ljm.eReadName(handle, name)
print("    %s = %f" % (name, result))

#Read the voltage on AIN0
name = "AIN0"
result = ljm.eReadName(handle, name)
print("    %s = %f" % (name, result))

#Set DAC0 to 3.3V
name = "DAC0"
value = 3.3
result = ljm.eWriteName(handle, name, value)
```

Why LabJack?

Legendary Support

- Email responses that actually answer your question.
- Free lifetime support includes (some) engineering design help.
- The engineers who made the product also respond to your questions.
- Free RMA diagnostics, calibration.

Flexibility

- Software integrates easily. We don't force you into a certain software or programming environment. Choose LabVIEW, C++, MATLAB, Python, Java, .NET, Delphi, Visual Basic, VB6, VBA, and more...
- Add new kinds of sensors on-the-fly. We provide inexpensive signal conditioning modules.
- Control valves, motors, lights, pumps, etc - using one of many digital I/O control options.
- Incorporate LabJack DAQ hardware using our OEM options.

Quality Hardware

- Have confidence in your measurements. Each device is individually tested and calibrated traceable to NIST standards.
- New features or fixes are readily available through field-programmable firmware.
- Each device has multiple protection mechanisms on every I/O to help prevent electrical damage.

"Your product saved me a bunch of money and time... I usually contact support organizations... about how bad their products are. I felt like I had to say how well yours worked!"

-Thomas A.
Software engineer

LabJack

HuDun Demo