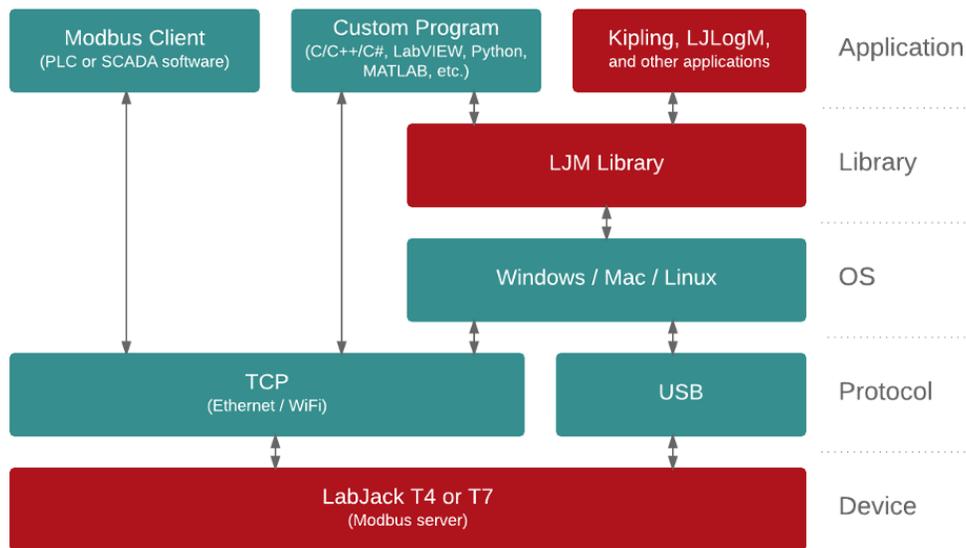


# LabJack

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

[Home](#) > [Support](#) > LJM Library

## LJM Library



LJM is LabJack's free, cross-platform driver / library for simplifying device communication. It supports the LabJack [T4](#) and [T7 series](#) devices and the LabJack [Digit-series](#) devices.

### Download

See the [LJM installer](#) page to download.

### Documentation

See the LJM [User's Guide](#).

### Features

- Cross-platform
- Supported in multiple languages
- Completely free
- [Licensed](#) with the permissive MIT X11 License
- Well-documented [User's Guide](#)
- [Device discovery](#)
- Device reconnection—automatically repairs connections

- Modbus [Name-To-Address](#) interpretation - ability to interpret LabJack register names such as "AIN0"
- [Error-To-String](#) Interpretation
- [Utility functions](#) to:
  - Convert bytes to values
  - Perform thermocouple calculations
  - Convert IP and MAC numbers
- Versatile [Library configurations](#)
- Threaded [data streaming](#) and buffering
- [Thread-safe](#)
- Basic [demo mode](#) for testing without a physical device.

## Cross-Platform

Support for Windows, macOS, and Linux allows for the same code to be run on different operating systems with the same results, whether it's on your Windows desktop, MacBook laptop, or Raspberry Pi running a distribution of Linux.

## Supported In Multiple Languages

Written in C++ with a C API, LabJack maintains and supports wrappers for many programming languages.

- [C/C++](#)
- [Python](#)
- [MATLAB](#)
- [C# .NET](#)
- [LabVIEW](#)

## C/C++

```
#include "LabJackM.h"

...

int error, handle;
double voltage;
error = LJM_OpenS("T7", "USB", "ANY", &handle);
// Read the voltage on AIN0
error = LJM_eReadName(handle, "AIN0", &voltage);

...
```

For more LJM C/C++ code, see the [C/C++ Examples](#) page.

## Python

```
from labjack import ljm

handle = ljm.openS("T7", "USB", "ANY")

# Read the voltage on AIN0
voltage = ljm.eReadName(handle, "AIN0")
```

For more LJM Python code, see the [Python Examples](#) page.

## MATLAB

```
ljmAsm = NET.addAssembly('LabJack.LJM');

...
[ljmError, handle] = LabJack.LJM.OpenS('T7', 'USB', 'ANY', handle);
% Read the voltage on AIN0
[ljmError, voltage] = LabJack.LJM.eReadName(handle, 'AIN0', 0);
...
```

For more LJM MATLAB code, see the [MATLAB Examples](#) page.

## C# .NET

```
using LabJack;

...

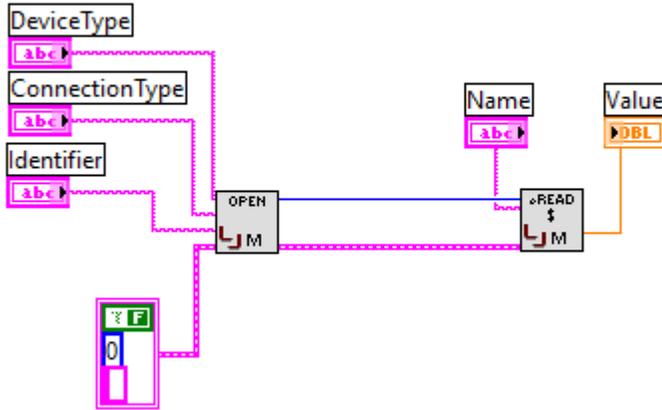
int handle;
LJM.OpenS("T7", "USB", "ANY", ref handle);

// Read the voltage on AIN0
double voltage;
LJM.eReadName(handle, "AIN0", ref voltage);

...
```

For more LJM C# .NET code, see the [C# .NET Examples](#) page.

## LabVIEW



For more LJM LabVIEW code, see the [LabVIEW Examples](#) page.

Some other supported languages:

- [LabVIEW](#)
- [Visual Basic .NET](#)
- [DAQFactory](#)
- See the [full list](#) of supported languages.

LJM's API allows for easy integration into most languages. If the language you use isn't currently supported, feel free to [contact us](#) for assistance in integrating LJM into your project.

## LJM Is For Any Level Of Expertise

You don't need to be familiar with Modbus to use LJM. Simply use LJM's [Easy Functions](#) for direct access to all the features your LabJack device provides.

If you do prefer to use Modbus, LJM's [Raw Byte Functions](#) provide byte-level control to manually send and receive Modbus packets.

## Modbus Interpretation

Using LJM's name-to-address map, you can communicate with LabJack devices using the readable names of data values, such as "AIN0", "AIN0\_RANGE", "ETHERNET\_IP", or "WIFI\_STATUS".

LJM provides functions that convert readable Modbus register names to a register addresses, such as [LJM\\_NameToAddress](#). Or, you can directly interact with your LabJack device using the Name-style [Easy functions](#). (Although you can use the Address-style Easy functions if you prefer.)

You can easily look up Modbus registers on the [Modbus Map](#) page.

## Easy Device Connection

Connecting to a LabJack device is simple. Here's some ways you can [open a connection](#):

- Just want to open any LabJack device?
  - Open a connection to the first found LabJack device.
- Just want to connect to any device on the network or on a USB hub?
  - Filter based on connection type.
- Just want to open any LabJack T4, or T7?
  - Filter based on device type.
- Got a network with static IP address?
  - Open by IP address.
- Got a T4/T7 on a DHCP network or on a large USB hub?
  - Open by serial number.
- Don't want to mess with serial numbers or IP addresses, but you still want to open a particular device?
  - Open by customizable name.

## Configurable Behavior

Customize the behavior of the LJM library to fit your needs. See the full list of [Configuration Parameters](#).

You can also put a configuration file on your system and load it into LJM, so that one configuration that works well can be loaded for multiple programs. (It's also easy to edit!)

## Questions?

- [Contact us](#)
  - See the detailed [LJM User's Guide](#)
  - See the [LJM example code](#) (available in multiple programming languages)
  - Visit the [LabJack Forums](#)
-