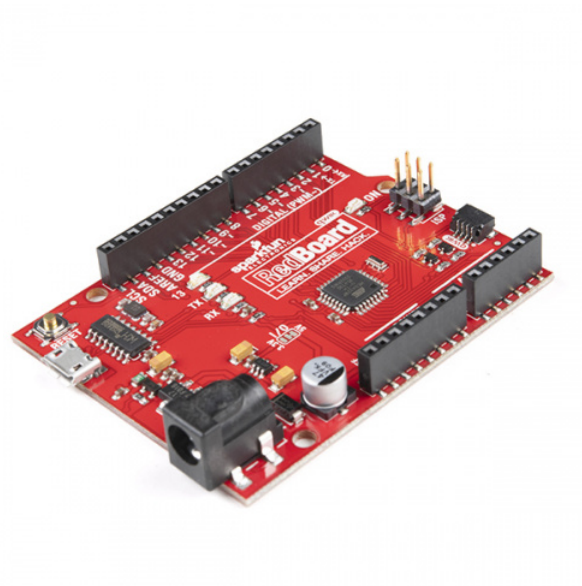
RedBoard Qwiic Hookup Guide

Introduction

The SparkFun RedBoard Qwiic is the newest edition of the Arduino-compatible development platforms in the SparkFun catalog. This updated version of the *classic* SparkFun RedBoard incorporates a few key improvements over its predecessor (see Hardware Overview). However, like the *original* SparkFun RedBoard, it is designed to be an easy-to-use learning platform for coding, physical computing, and project prototyping. These skills are becoming increasingly significant in today's education and the technological community.



SparkFun RedBoard Qwiic
DEV-15123

Product Showcase: SparkFun RedBoard Qwiic

This tutorial aims to familiarize you with the new SparkFun RedBoard Qwiic and help you get started using it. To begin, we'll guide you through the installation of the Arduino IDE (Integrated Development Environment) software, the main user interface for programming the board. Next, we will go over the hardware and features of the board. Finally, we will walk you through a few examples using the Arduino IDE.

The SparkFun RedBoard Qwiic can interact with real-world sensors, control motors, display information, and perform near-instantaneous calculations. It enables *anyone* to create unique, nifty projects from something as simple as displaying characters on an LCD display or detecting changes in light to vastly more complicated projects like an IoT cellular device (**Not** recommend for beginners... start with something simpler and work your way up.). If you're familiar with how the *original* SparkFun RedBoard worked, you may want to skim over parts of this tutorial.

Required Materials/Tools

To get started, all you need is a few things:

- SparkFun RedBoard Qwiic - You'll definitely need this; otherwise, you might be on the wrong tutorial (*wink-wink*).
- USB micro-B Cable - 6 Foot - The USB interface serves two purposes: it powers the board and allows you to upload programs to it.
- Computer with the **Arduino IDE** installed on it - That is how we will program the board and interface with it.

> **Troubleshooting Tip:** If you are not a technical or computer savy individual and you have your choice of computers, a **Windows 7 or 10** computer is highly recommended. You will usually run into the the least issues, if any, with these operating systems.

That is ALL... pretty simple right? Now you won't be able to do much since there are no additional sensors to interact with the physical world. However, you can at least blink an LED and do some math calculations.

JUMPER MODIFICATION **QWIIC EXAMPLE**

***Click the buttons** above to toggle the **additional materials** based on the tasks you wish to perform. Feel free to modify the items in your cart to fit your needs.*

Suggested Reading

The **SparkFun RedBoard Qwiic** aims to be a beginner-friendly microcontroller platform. You can get started without an innate knowledge of Ohm's Law or How Electricity Works (but a little understanding wouldn't hurt!). The following are some subjects you should be familiar with; however, to use the more advanced features of the board, it is recommended that you read up on the **Logic Levels** and **I^2C** tutorials.

What is a Circuit?

Every electrical project starts with a circuit. Don't know what a circuit is? We're here to help.

What is an Arduino?

What is this 'Arduino' thing anyway? This tutorials dives into what an Arduino is and along with Arduino projects and widgets.

Logic Levels

Learn the difference between 3.3V and 5V devices and logic levels.

I2C

An introduction to I2C, one of the main embedded communications protocols in use today.

One of the new, advanced features of the board is that it takes advantage of the Qwiic connect system. We recommend familiarizing yourself with the **Logic Levels** and **I^2C** tutorials (above) before using it, as all **Qwiic** sensors utilize an **I^2C** communication protocol. Click on the banner above to learn more about Qwiic products.

Fun Fact: Qwiic is a play on words between "quick" and I^2C or "iic".

Installing Drivers

Note: The USB-to-Serial adapter IC, used on the RedBoard Qwiic, is different from what was used on the original RedBoard. **The new CH340C chip will require a different driver than the FTDI chip used on the original RedBoard** because they are different chips made by separate manufacturers. **Please make sure to follow the driver installation guide before plugging this new board into your computer.**

The SparkFun RedBoard Qwiic uses a CH340C USB-to-Serial adapter made by WCH. The driver for the CH340C chip will need to be installed on your computer. We have tested and confirmed that the driver works on Windows 7, Windows 10, Mac OSX High Sierra, and Raspbian Stretch (11-13-2018 release) for the Raspberry Pi. On all operating systems, if you have previously installed the CH340G drivers, you will need to uninstall those drivers first before updating to the new CH340C driver. For more information, check out our How to Install CH340 Drivers Tutorial.

How to Install CH340 Drivers

AUGUST 6, 2019

How to install CH340 drivers (if you need them) on Windows, Mac OS X, and Linux.

Installing the CH340C driver allows your computer to recognize the SparkFun RedBoard Qwiic as a device and lets it communicate with the board over the USB connection. You can also find the latest version of the CH340 drivers from WCH here. (*Most of their pages are in Mandarin, but if you use a Chrome web browser, you should have the option to have the web page translated.*)

Installing the Arduino IDE

Download/Install Arduino

You can download the Arduino IDE from their website. They have installation instructions, but we will also go over the installation process as well. Make sure you download the version that matches your operating system.

CLICK FOR ARDUINO IDE DOWNLOAD PAGE

The installation procedure is fairly straightforward, but it does vary by OS. Here are some tips to help you along. We've also written a separate Installing Arduino tutorial in case you get stuck.

Troubleshooting Tips:

- We recommend using a computer with a full desktop operating system like Windows 7/10 (**avoid** Windows 8 if you can), Mac OSX, and certain flavors Linux (check the Arduino FAQ page for compatibility).
    - If you are not a technical or computer savy individual and you have your choice of computers, I highly recommend using a **Windows 7 or 10** computer. You will usually run into the the least issues, if any, with these operating systems.
- We do **NOT** recommend using a Chromebook, Netbook, tablet, phone, or the Arduino Web IDE in general. You will be responsible for troubleshooting any driver or Arduino Web IDE issues.
- As of writing this tutorial (12-14-2018), the most recent and stable release of the Arduino IDE is version 1.8.5. We recommend using that version of the Arduino IDE; you can download the previous releases here.
- On Windows 10, we do **NOT** recommend installing the Arduino IDE from the app store. You may run into issues because the OS will automatically update to the most recent release of the Arduino IDE, which may have unknown bugs (like the compiler errors in versions 1.8.6 and 1.8.7).
- Raspberry Pi users with Raspbian installed should use the **Linux ARM** download. We do not recommend using the command line installation. It will install the oldest release of Arduino, which is useless when it comes to installing new boards definitions or libraries.
- For additional troubleshooting tips, here is a troubleshooting guide from Arduino.

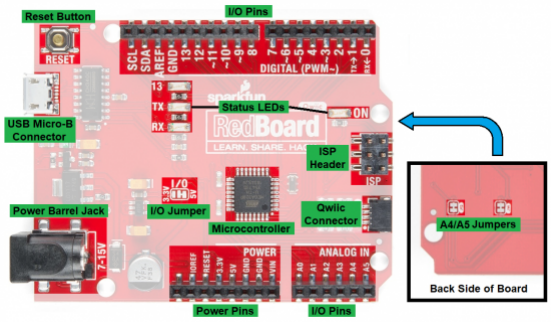
WINDOWS 7/10 **MAC OSX** **LINUX**

Click the buttons above *for OS specific instructions.*

With Arduino downloaded and installed, the next step is to plug the board in and test it out! Pretty soon you'll be blinking LEDs, reading buttons, and doing some physical computing!

Hardware Overview

Below is an annotated image, and an overview of all of the important features for the SparkFun RedBoard Qwiic:



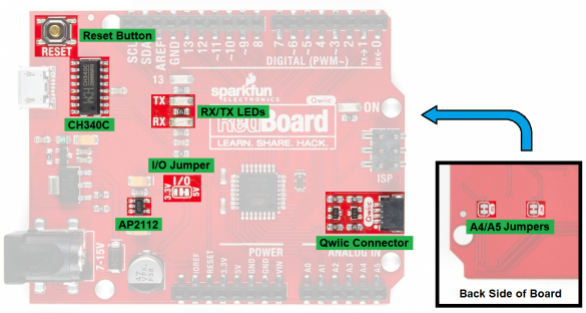
Click the buttons *below for more details.*

NEW FEATURES | DIMENSIONS | MICROCONTROLLER | POWER/RESET | STATUS LEDS | PROGRAMMING | PIN FUNCTIONS | PIN CONNECTIONS | QWIIC CONNECTION | JUMPERS

New Features

The new features of the SparkFun RedBoard Qwiic include:

- An updated Serial-USB Converter Chip
- An improved AP2112 Voltage Regulator
- A Qwiic Connector
- Available A4/A5 Jumpers
- Available Voltage Level Jumpers
- An improved Reset Button
- RX/TX LED color change



New features of RedBoard Qwiic.

Serial-USB Converter Chip

The new CH340C IC allows the SparkFun RedBoard Qwiic to utilize the USB micro-B connection and should reduce the need for users to manually install drivers. Newer operating systems should automatically recognize and install the drivers for the board.

AP2112 Voltage Regulator

The AP2112 is a more robust **3.3V** regulator to provide more power to daisy chain multiple Qwiic devices. Unlike the MIC5205, which could only source about 150mA of current; the AP2112 can source up to **600mA** of current and should be able to handle most of your Qwiic device needs.

Qwiic Connector

This connector allows the SparkFun RedBoard Qwiic to seamlessly interface with SparkFun's Qwiic Ecosystem.

A4/A5 Jumpers

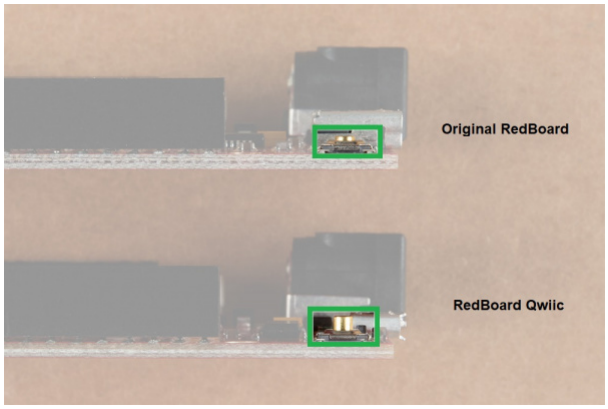
Pins **A4** and **A5** are tied directly to the I^2C bus. These jumpers can be used to disconnect the logic level converters from pins **A4** and **A5** so that they might be used independently from the Qwiic system for analog readings.

Voltage Level Jumpers

These jumpers allow users to easily switch from a **3.3V** to **5V** board. This allows the user to convert the board (and I/O pins) to either **3.3V** or **5V** based on their needs. Unlike the the *original* SparkFun RedBoard, you no longer need a logic level converter to interface directly to a **3.3V** device or sensor.

Reset Button

The board also includes a new, more pronounced reset button that is easier to push. Anyone with "*fat fingers*" can relate to this struggle.



The more ergonomic reset button that is easier to press.

RX/TX LEDs

The TX LED is now green, instead of yellow and the RX LED is now yellow, instead of red.

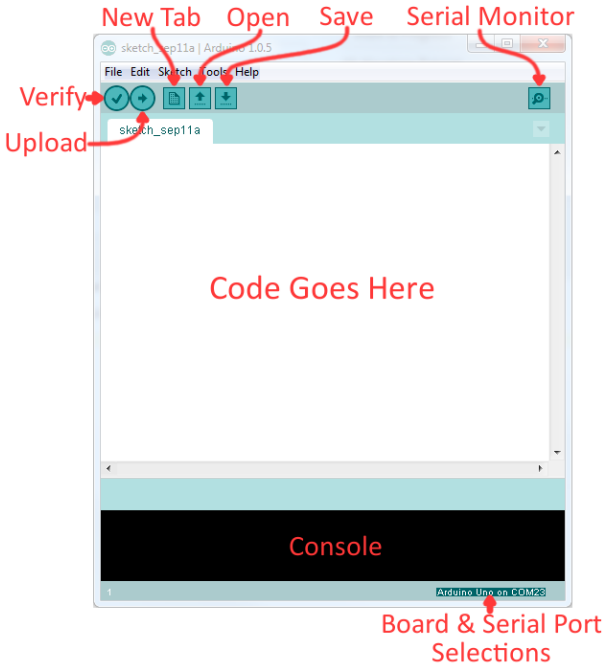
Arduino Examples

Example 1: Uploading Blink

In this example we will go over the basics of the Arduino IDE and upload a sample code. This is a great way to test the basic functionality of any board to make sure it is working.

The Arduino IDE

Now it's finally time to **open up the Arduino software**. You'll be presented with a window that looks a little something like this:

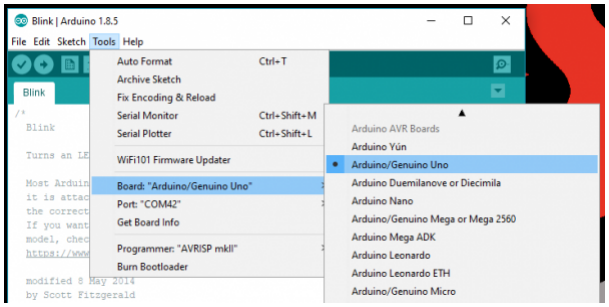


Layout of the Arduino IDE.

Before we can send the code over to the RedBoard, there are a couple of adjustments we need to make.

Select a Board

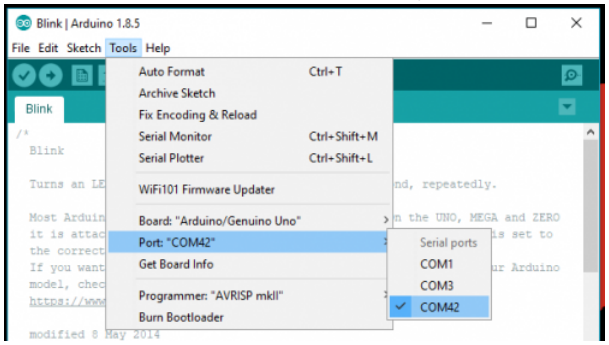
This step is required to tell the Arduino IDE *which* of the available Arduino boards, we are using. Go up to the **Tools** menu. Then hover over **Board** and make sure **Arduino/Genuino Uno** is selected.



*Screen shot of **Board** selection.*

Select a Serial Port

Next up we need to tell the Arduino IDE which of our computer's serial ports the RedBoard is connected to. For this, again go up to **Tools**, then hover over **Serial Port** and select your RedBoard's COM port.

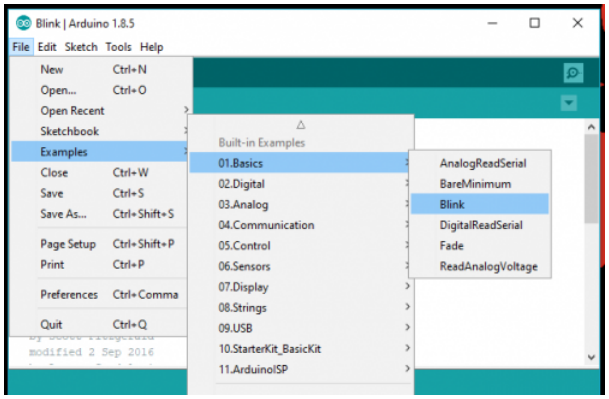


*Screen shot of **COM Port** selection.*

If you've got more than one port, and you're not sure which of the serial ports is your RedBoard, unplug it for a moment and check the menu to see which one disappears.

Blink Sketch

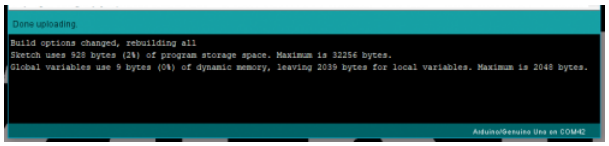
Code written for the Arduino IDE are referred to as sketches. All code in Arduino is C based. Let us upload a **Blink sketch** to make sure our new RedBoard setup is totally functional. Go up to the **File** menu in Arduino, then go to **Examples > 01.Basics > Blink** to open it up.



*Screen shot of **Blink sketch** selection.*

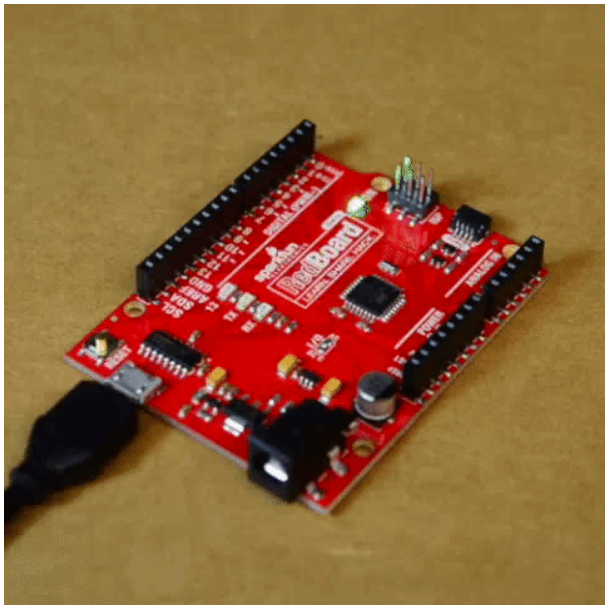
Upload!

With all of those settings adjusted, you're finally ready to upload some code! Click the **Upload** button (the right-pointing arrow) and allow the IDE some time to compile and upload your code. It should take around 10-20 seconds for the process to complete. When the code has uploaded, you should see something like this in your console window:



Screen shot of upload complete.

And if you look over to the RedBoard, you should see the blue LED turn on for a second, off for a second, on for a second, off for a second...ad infinitum (at least until it loses power).



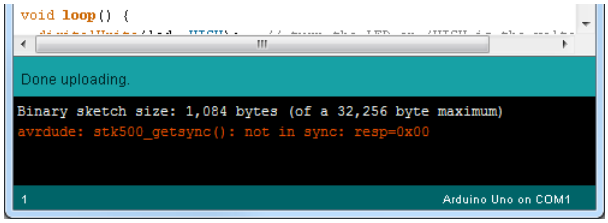
Expected response from board

If you want to adjust the blink speed, try messing with the "1000" value in the `delay(1000);` lines. You're well on your way to becoming an Arduino programmer!

Something Wrong?

Uh oh! If you didn't get a "Done Uploading" message, and instead got an error, there are a few things we can double-check.

If you got an `avrdude: stk500_getsync(): not in sync: resp=0x00` error in your console window.



*Screen shot of **Error Message** in the **Console**.*

Either your serial port or board may be incorrectly set. Again, make sure **Arduino/Genuino Uno** is the board selection (under the "**Tools** > **Board**" menu). The serial port is usually the more common culprit here. Is the Serial Port correctly set (under the "**Tools** > **Serial Port**" menu)? Did the drivers successfully install? To double check your RedBoard's serial port, look at the menu when the board is plugged in, then unplug it and look for the missing port. If none of the ports are missing, you may need to go back to driver installation.

Example 2: Qwiic Connector

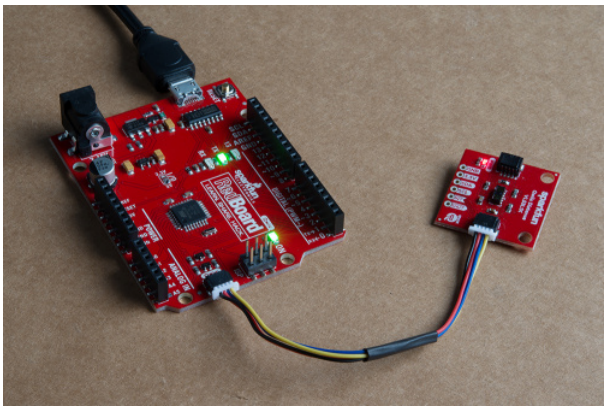
One of the great features of the RedBoard (Qwiic) is its ability to interface with I^2C devices using our Qwiic system. The Qwiic system is a solderless connection system that allows users to seamlessly daisy chain multiple I^2C devices with ease.

The Qwiic Distance Sensor

For this example, we will be running a basic sketch using the SparkFun 4m Distance Sensor (VL53L1X). For more examples with this sensor, please refer to the complete hookup guide.

Hardware Assembly

The wiring for this is simple. Use the Qwiic cable and connect the distance sensor to the board. That is it! The connections are polarized, so you don't have to worry about which side or connector you are using.



Hardware assembly for VL53L1X distance sensor example.

Let's run an example for our distance sensor to see how it behaves.

Install the Arduino Library

Note: If you have not previously installed an Arduino library, please check out our Arduino library installation guide.

First, you'll need the **Sparkfun VL53L1X** Arduino library. You can obtain these libraries through the Arduino Library Manager. Search for **Sparkfun VL53L1X Arduino Library** to install the latest version. If you prefer downloading the libraries from the GitHub repository and manually installing it, you can grab them here:

DOWNLOAD THE SPARKFUN VL53L1X ARDUINO LIBRARY (ZIP)

Example 1 - Read Distance

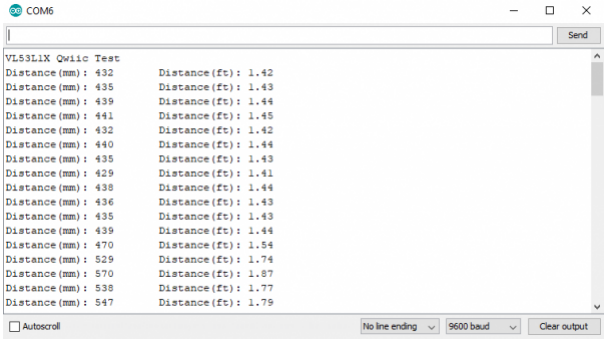
To get started with this example, open up **File** > **Examples** > **SparkFun VL53L1x 4M Laser Distance Sensor** > **Example1_ReadDistance**. In this example, we begin by creating a `SFEVL53L1X` object called `distanceSensor` with our wire port, `Wire` , and then our shutdown and interrupt pins. Then we initialize our sensor object in the `setup()` loop. The code to do this is shown below.

```
#include <Wire.h>
#include "SparkFun_VL53L1X.h"

//Optional interrupt and shutdown pins.
#define SHUTDOWN_PIN 2
#define INTERRUPT_PIN 3

SFEVL53L1X distanceSensor(Wire, SHUTDOWN_PIN, INTERRUPT_PIN);

void setup(void)
{
  Wire.begin();

  Serial.begin(9600);
  Serial.println("VL53L1X Qwiic Test");

  if (distanceSensor.init() == false)
    Serial.println("Sensor online!");

}
```

Once we've initialized our sensor, we can start grabbing measurements from it. To do this, we send some configuration bytes to our sensor using `distanceSensor.startRanging()` to initiate the measurement. We then wait for data to become available and when it does, we read it in, convert it from millimeters to feet, and print it out over serial. The `void loop()` function that does this is shown below.

```
void loop(void)
{
  distanceSensor.startRanging(); //Write configuration bytes to initiate measurement
  int distance = distanceSensor.getDistance(); //Get the result of the measurement from the sens
or
  distanceSensor.stopRanging();

  Serial.print("Distance(mm): ");
  Serial.print(distance);

  float distanceInches = distance * 0.0393701;
  float distanceFeet = distanceInches / 12.0;

  Serial.print("\tDistance(ft): ");
  Serial.print(distanceFeet, 2);

  Serial.println();
}
```

Opening your serial monitor to a baud rate of **9600** should show the distance between the sensor and the object it's pointed at in both millimeters and feet. The output should look something like the below image.



Distance readings in mm and ft

Troubleshooting

Below, we have also included some additional troubleshooting tips for issues that you may come across with the new RedBoard (Qwiic).

1. One of our employees compiled a great list of troubleshooting tips based on the most common customer issues. This is the perfect place to start.
2. For any Arduino IDE specific issues, I recommend starting with their troubleshooting guide.

If neither of the troubleshooting guides above were able to help, here are some tips you might have missed. (Most of this material is summarized from the tutorial.):

Are You Using a Recommended Computer OS?

This board is not tested using the Arduino Web IDE. We do **NOT** recommend using a Chromebook, Netbook, tablet, phone, or the Arduino Web IDE in general. If you are here, try a **RECOMMENDED** operating system (see Installing the Arduino IDE).

My Board Isn't Working:

Every board that **we** manufacture gets tested. If you didn't buy the board from us or one of our authorized distributors, it could be a knock-off. That being said, let's try a basic test to see if just the board is working. Disconnect everything that you have attached to the board; we just want to test the board.

1. **Inspect the board:**
   Check the board to make sure everything looks about right. Use the pictures on the product page to verify component placement or alignment, and bad solder joints, or damage.
2. **Power and check the status LEDs:**
   Using a known good USB micro-B cable, plug your board in to the computer. Do any of the status LEDs turn on (see Hardware Overview)?
   - New boards will come programmed with a test sketch that cycles between the RX and TX LEDs.
3. **Upload the *Blink* sketch:**
   Try to upload a blink sketch. Why blink? It is simple, known to work (from the example files), and you have an indicator LED.
   - Double check that you have the proper **Board** and **Serial Port** selected.
   - For boards that are already running the blink example, I recommend changing the timing parameters to check for a change in the board's response.

   Verify that you see the status LED blinking properly and that the Arduino IDE shows a status of "**Done uploading.**"

I Don't See My Board on a Serial/COM Port:

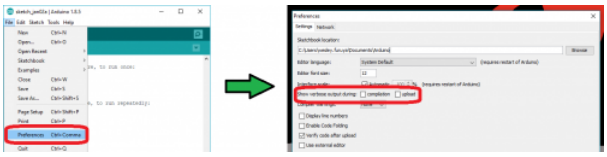
If you don't see your board as an available COM port on the Arduino IDE:

- Try to re-open the Arduino IDE.
- Check the **Device Manager** to verify that your computer recognizes the board. Click the **Driver Verification** button in the Installing Drivers section of the tutorial.
- If you have previously installed the **older CH340G drivers**, you may need to update your drivers. Particularly on Macs, you will need to delete the previous drivers and install the updated drivers.
- If that is not the case, you issue might be related to your USB cable. Check that you are using a USB cable capable of data transfers. **Some cables only have the power pins connected for charging**. A good way to test this is to plug in a device to your USB cable (like a phone). If it doesn't show up as a device or drive, then try a new USB micro-B cable.
- This rarely happens, but it is easy to check. If you are using a USB 3.0 port (you will see a blue "*tongue*" in the USB jack or bad USB port, try a different USB port. You can also try to test the board on a different computer to double check for a hardware incompatibility (usually with expansion boards).

Errors Uploading to the Board:

There are two types of issues that you will usually see in the console of the Arduino IDE, compile errors or upload errors. The easiest way to see where to start is by clicking the **Verify** button (check mark); the Arduino IDE will try to compile your code. A failure here is a compile error.

It takes a some experience, but if you enable the verbose output from the Arduino IDE **preferences**, it may give you more clues to where the issue is.



*Screen shots of how to enable verbose output. **Click** to enlarge.*

- **Compile Errors:**
  With compile errors, there are several things that could be causing issues. However, 99% of the time, it is user error. Usually something wrong with your code or the library you are using. Once in a while you will have a file structure issue if you manually added a file/folder in any of the Arduino folders (still user error).
- **Upload Errors:**
  Upload errors get a little more tricky. You will usually just see the Arduino IDE trying to upload to the board multiple times. There are usually several different causes for this, often without specific errors in the console. Here are a few common examples:
  - Wrong Board Selection:
    Double check you board selection options. If you uploaded with the wrong board selection, there is a small chance that you may have overwritten the bootloader on the board or damaged the microcontroller.
  - Missing Bootloader:
    If your board has the bootloader flashed, pin 13 will flash several times on power up.
  - Serial Port Interference:
    If a device is communicating to the microcontroller over digital pins 0 and 1, while you are trying to upload code.
  - Bad USB cable or port (see Serial Port section above).

Additional Tips:

- If an input pin is read and that is floating (with nothing connected to it), you will see random data/states. In practice, it may be useful to tie an input pin to a known state with a pullup resistor (to VCC), or a pulldown resistor (to GND).
- Pin 13 is difficult to use as a digital input because of the voltage drop from status LED and resistor soldered in series to it. If you must use pin 13 as a digital input, it is recommended that you set the pinMode() as an INPUT and use an external pulldown resistor.
- The maximum current an I/O pin can source (provide positive current) or sink (provide negative current) is 40 mA (milliamps). You can power small sections of LED strips or small motors, but will run into issue with high power devices.
- Be sure to double check that you are not trying to use an I^2C device while you are trying use analog pins A4 or A5 to read analog data. You will run into issues where the analog read will appear to be at a constant value. As a result, the analog readings will not reflect the changes seen from the sensor output. You can only do one or the other.
- Using the Serial Peripheral Interface, configures the SCK and MOSI pins to be directly managed by the SPI hardware. Therefore, while in use, pins 11 and 13 can't be used (i.e. the LED on pin 13 can no longer be used as a debug/status indicator.) Executing "SPI.end();" allows those pins 11 and 13 to be used as general I/O again.
- This issue doesn't happen too often, but it can be *arbitrarily, common*: If your mouse pointer begins to move erratically, your mouse becomes unresponsive to your inputs, and your board is sending a lot of serial data, there is a chance that your computer thinks your board as a serial mouse. The fix is to unplug your board and the plug it back in while holding the reset button down, giving your computer a chance enumerate the COM port.

Resources and Going Further

Now that you've successfully got started with your SparkFun RedBoard Qwiic, it's time to incorporate it into your own project! For more information, check out the resources below:

- Schematic (PDF)
- Eagle Files (ZIP)
- CH340C Hookup Guide (for Drivers)
- Qwiic Landing Page
- GitHub Product Repository
- SFE Product Showcase

Need some inspiration for your next project? Check out some of these related tutorials:

ARDUINO IDE | BOARDS & SHIELDS | BOARD FUNCTIONALITY | PROJECT GUIDES

SparkFun Tutorials

Installing an Arduino Library

How do I install a custom Arduino library? It's easy! This tutorial will go over how to install an Arduino library using the Arduino Library Manager. For libraries not linked with the Arduino IDE, we will also go over manually installing an Arduino library.

What is an Arduino?

What is this 'Arduino' thing anyway? This tutorials dives into what an Arduino is and along with Arduino projects and widgets.

Installing Arduino IDE

A step-by-step guide to installing and testing the Arduino software on Windows, Mac, and Linux.

Arduino Tutorials

- Getting Started > Introduction: What is Arduino and what I can use it for?
- Getting Started with Arduino and Genuino products

- Arduino Software (IDE)
- Arduino Troubleshooting
- Arduino: Contact Us