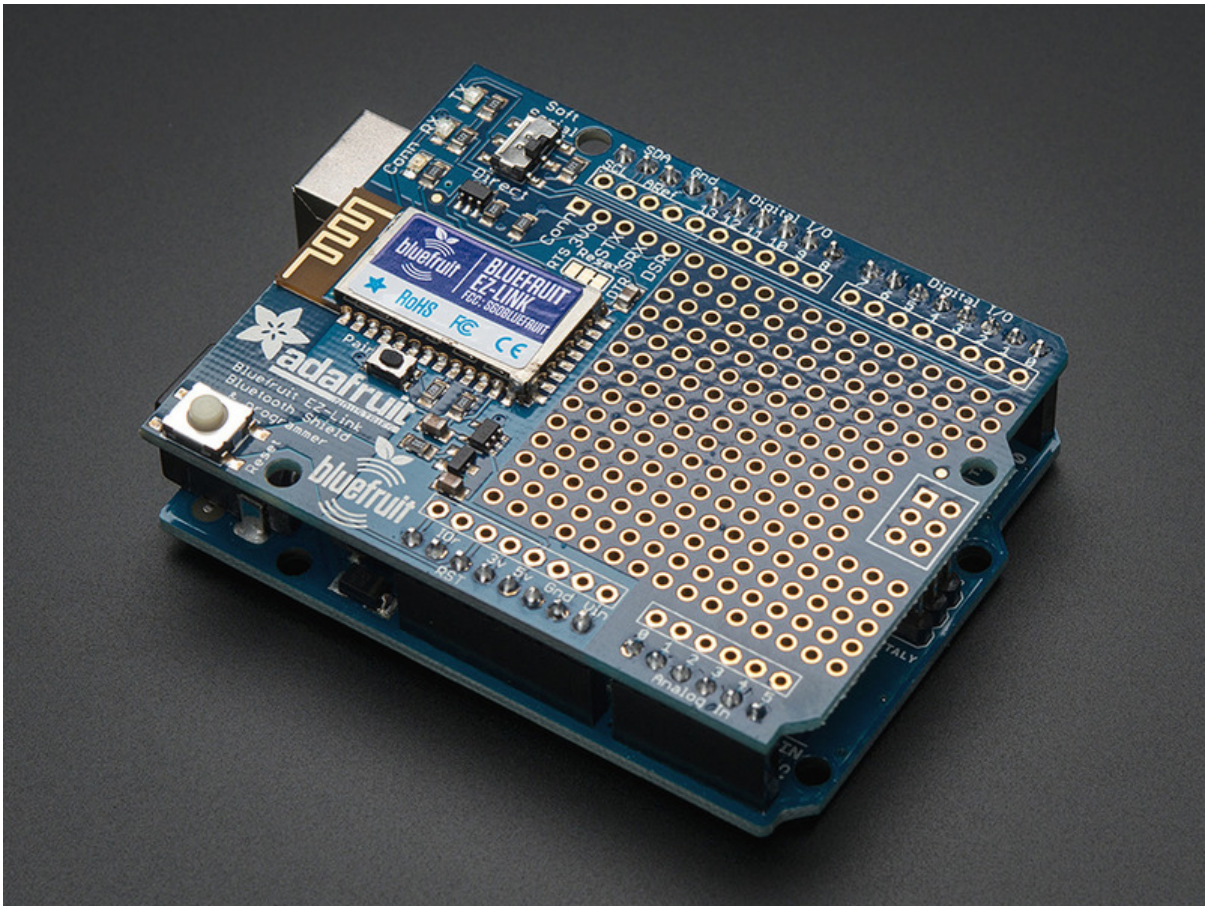
Adafruit Bluefruit EZ-Link Shield

Created by lady ada



https://learn.adafruit.com/adafruit-bluefruit-ez-link-shield

Last updated on 2024-06-03 01:34:50 PM EDT

Table of Contents

Overview 3

Pinouts 6

- Direct / Soft Serial Switch
- Extra Breakouts
- LEDs
- Pair Button
- Reset Button

Pairing 8

- Windows
- Mac OS X

Programming with Arduino 14

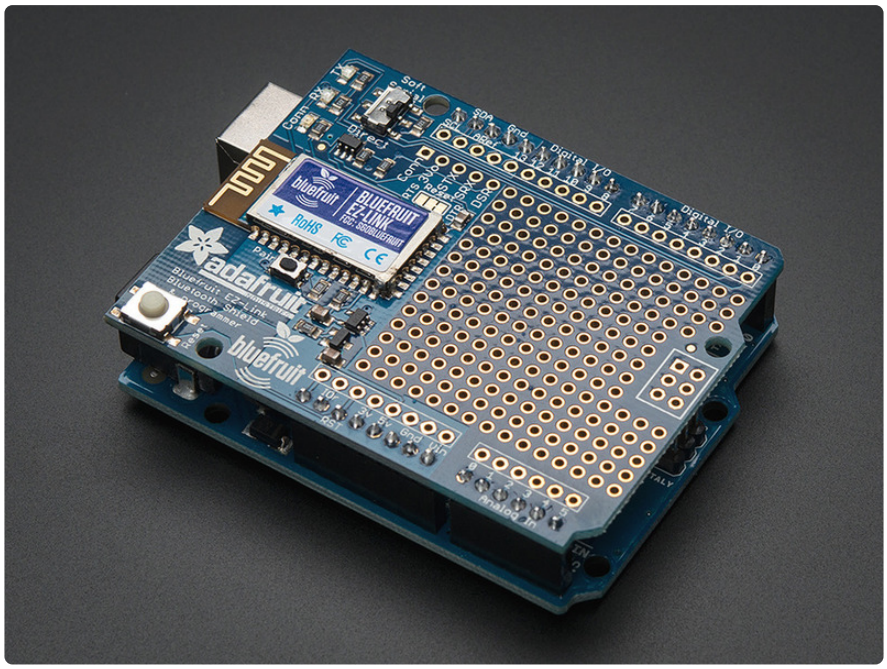
- Upload

F.A.Q. 16

Downloads 17

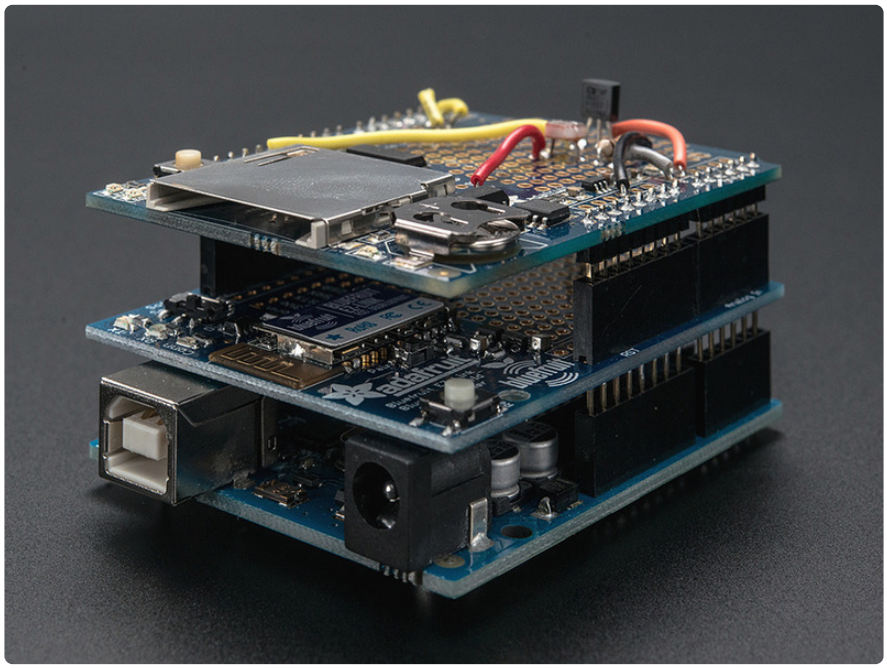
- Files
- EZ-Link shield schematics
- Fabrication Print

Overview



We are excited to add another product to our growing Adafruit Bluefruit line, this time it's the Bluefruit EZ-Link Shield: the best Bluetooth Serial shield ever made. Like you, we have purchased all sorts of Bluetooth shields with high expectations - we just wanted something that worked! But nothing ever did exactly what we wanted: there was always some configuration modes to wade through, and trying to reprogram an Arduino is impossible.

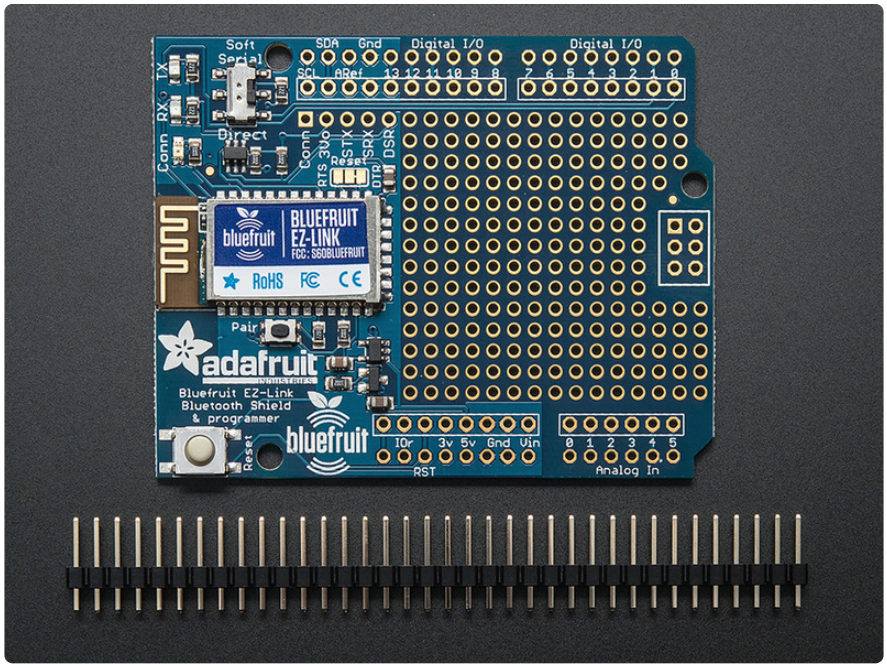
So we did what we always do, we went in and engineered something better. Something that works!



he Bluefruit EZ-Link shield is a regular 'SPP' serial link client device, that can pair with any computer or tablet and appear as a serial/COM port (except iOS as iOS does not permit SPP pairing). But here is where it gets exciting: unlike any other BT module, **the EZ-Link can automatically detect and change the serial baud rate. That means if you open up the COM port on your computer at 9600 baud, the output is 9600, 57600? 57600. Yep even 2400.** All the most common baud rates are supported: 2400, 4800, 9600, 19200, 38400, 57600, 115200 and 230400. You never have to configure or customize the module by hand - it all happens completely automatically inside the RF module.

Now if we stopped there, you'd probably think "wow that is pretty nice" but we didn't stop there! The EZ-Link has another impressive feature: the DTR/RTS/DSR flow control pins are automatically synced to the computer serial port. What this means is that if the computer sets the hardware flow control DTR or RTS pins high or low the pins on the bluetooth module will follow. If the DSR input pin is brought high or low on the EZ-Link, the computer can detect that as well. Every other Bluetooth SPP device we've ever seen, if they even have the RTS/DTR pins brought out, do not sync back to the computer, instead the flow control is for the module serial buffer itself.

Works great with Mac and Windows computers. Linux is not supported at this time.

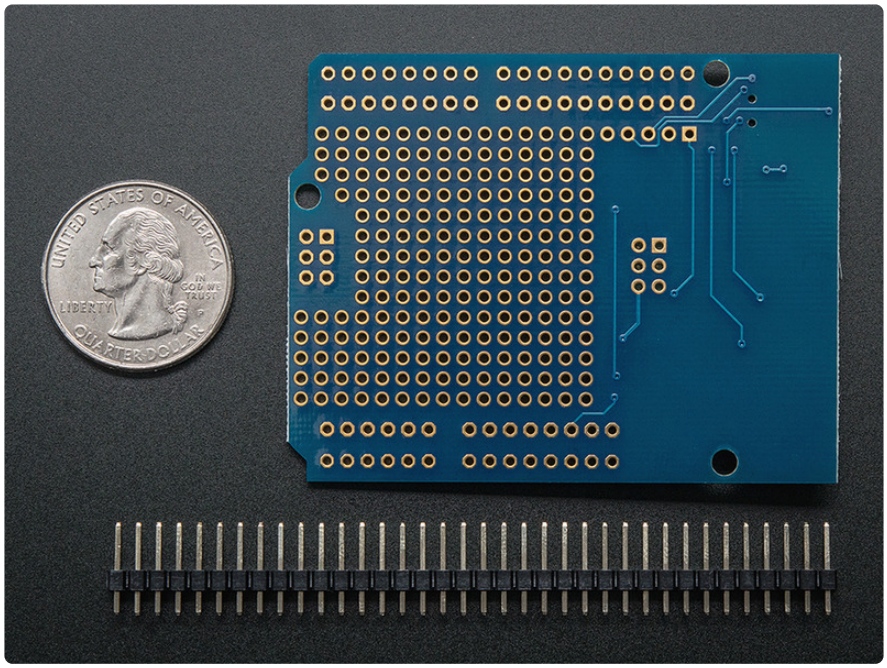


Together, this creates something pretty amazing: a Bluetooth shield that can change baud rates on the fly and toggle the DTR pin as desired. What we've got here now is a way to program an Arduino (or compatible) from 10 meters away, completely wirelessly, with no extra software, custom hardware, odd firmware hacks or modified firmware. It works great, and you can use the serial console as well. **It's the only Bluetooth shield that has completely transparent Arduino re-programming &**

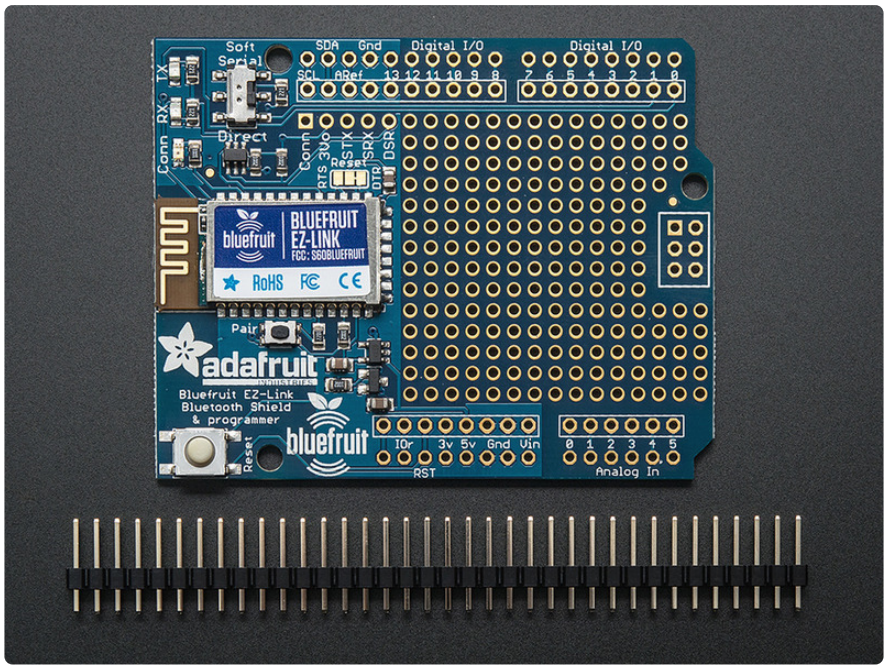
communication capability

To round out the design we also added two RX/TX indicator LEDs so you can tell when data is sent & received as well as a connection status breakout pin. When the paired computer opens up a UART connection, the status pin goes high, letting the Arduino know that it can now expect data to be sent and it's also OK to send data to the computer. You can use the shield in either Direct mode (the Bluetooth module replaces the USB/Serial chip) or in Soft Serial mode (you still want to program over USB but will use SoftwareSerial to communicate with a computer or other device).

Works great with Arduino Uno/Duemilanove/compatible and Mega. Not for use with Leonardo (the Leonardo uses direct-USB instead of USB-Serial) or Due.



Pinouts



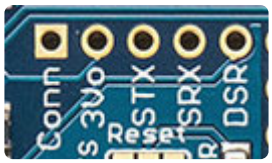
Direct / Soft Serial Switch

The most important element on the shield is the software/direct switch. The shield has two modes:

- **Direct serial**, where the bluetooth module 'replaces' the onboard USB->Serial chip. Used for reprogramming
- **Software serial**, where the Arduino can use any two pins to connect to the bluetooth module, but then it cannot be reprogrammed over bluetooth

For users who want a 'wireless link' that can reprogram the Arduino, use **Direct** only!

If you want to program over USB and then use Bluetooth to communicate with a computer over a seperate link, use **Soft Serial**



Extra Breakouts

We broke out some extra pins you may or may not need. If using the shield in **Direct** mode as just a wireless link, you can do without these pins. But advanced users may find them useful!

- **Conn** - this is the **connection** status pin, it has logic HIGH when there's a bluetooth pairing. Useful if you want to activate something when a connection is established, or stop when the connection breaks
- **3vo** - this is the 3.3V output from the onboard regulator, may be handy if you need an extra 3.3V regulator, 50mA max
- **STX** - this is the **Software Serial** TX pin, used in **Soft Serial** mode. Connect to any Arduino pin you like (that supports software serial!)
- **SRX** - this is the **Software Serial** RX pin, used in **Soft Serial** mode. Connect to any Arduino pin you like (that supports software serial!)
- **DSR** - this is a pin that you can use for sending a single bit indicator back to the computer on the DSR line. Since we have the DTR pin out for resetting the Arduino, we broke this pin out too. It's not high speed so you cant send data over it, just useful as a signaling pin

LEDs

There are three LEDs, one red and two blue.

The red Connection LED is used to indicate the state of the bluetooth connection/link. If it is slow-blinking, there is no pairing for the module. If it is fast blinking, it's paired to a computer and the wireless connection is 'live' - the computer has opened a UART

connection and is sending/receiving data.

The two blue LEDs are for monitoring the data received & sent. When data is sent from the computer to the module, the **TX** LED will blink. When data is sent from the module to the paired computer, the **RX** LED will blink. You can use this to debug your wiring and connection.

Pair Button

The pair button is used to reset the computer pairing for Bluetooth link. If you want to re-pair the module, press the button for 5 seconds, you'll see the red LED fast-blink to indicate its ready for pairing! You can now use your computer to Bluetooth-scan for the EZ-Link device.

Reset Button

The larger reset button is for resetting the Arduino, it does not affect the bluetooth module or pairing

---

Pairing

Before you start this step you should make sure your Bluefruit EZ-Link Shield is soldered with header or stacking header and is powered up.

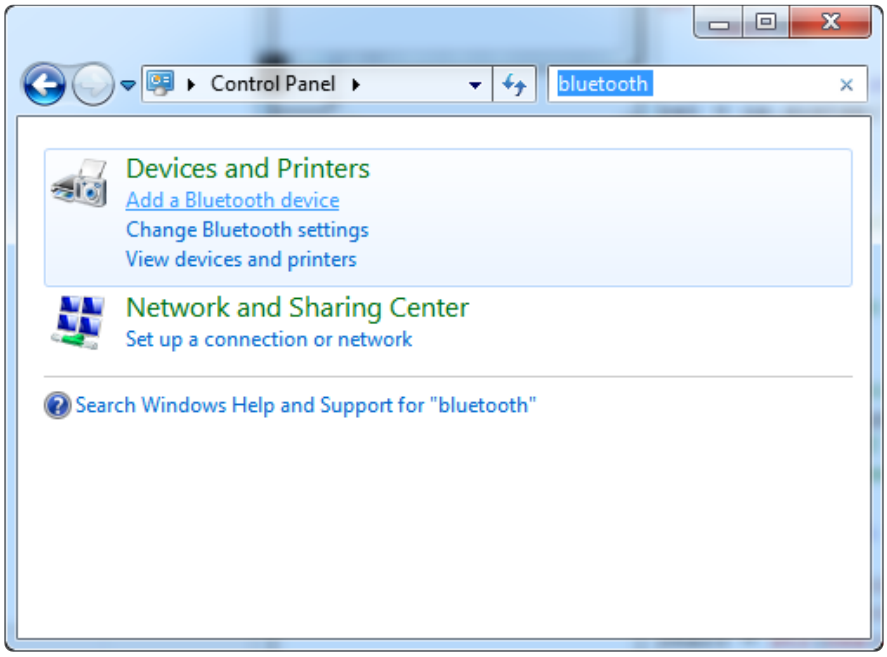
You'll also need to make sure that your computer that you'll be pairing with has Bluetooth v2.1 or great hardware installed. Nearly every Mac computer/laptop, Windows laptop and Android tablet has this already. If not, a USB Bluetooth module such as this one is ideal. (http://adafru.it/1327)Watch out for "cheap $5" bluetooth USB modules, they are often v2.0 only, and won't work with Bluefruit due to the

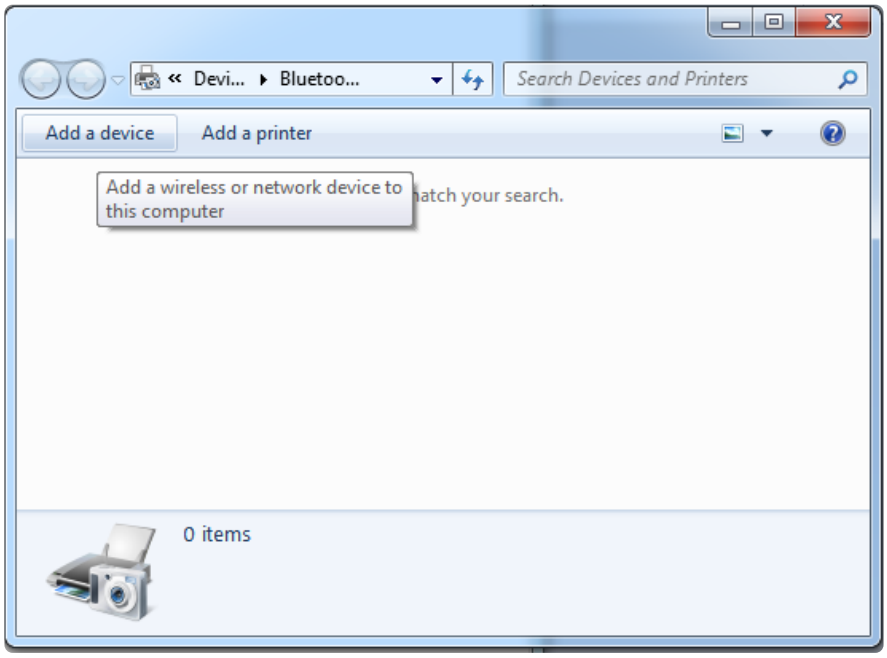
advanced firmware requirements.

It is not possible to pair the EZ-Link with iOS devices such as iPhones or iPads - Apple does not permit 'SSP' pairing! Check out the EZ-Key for a iOS compatible bluetooth keyboard module you may be able to use. (http://adafru.it/1535)

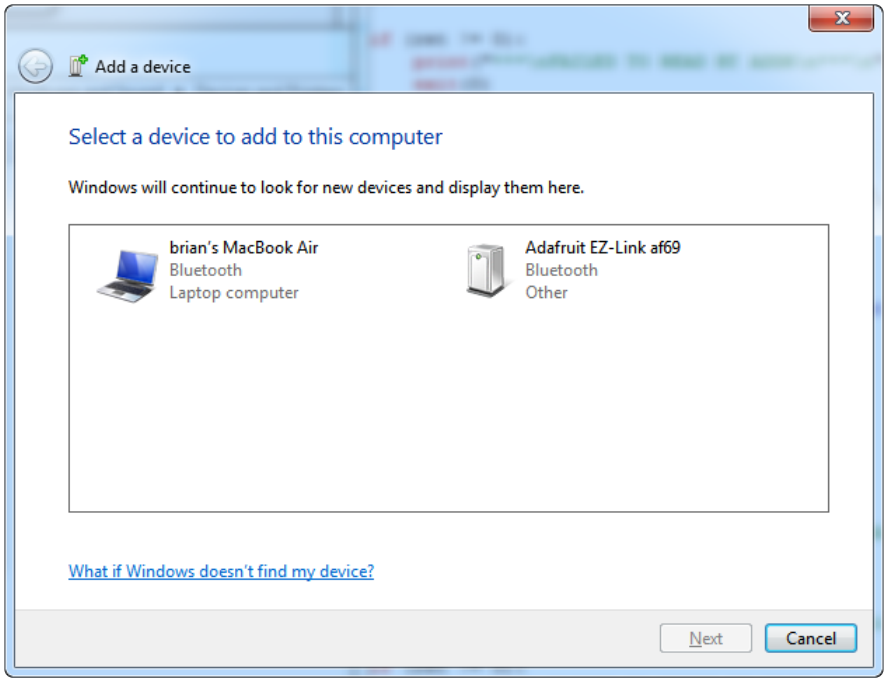
Windows

Open up the Bluetooth devices panel, on our computer we just clicked on the little Bluetooth icon in the task bar. You can also open the Control Panel and search for Bluetooth and then click Add Devices

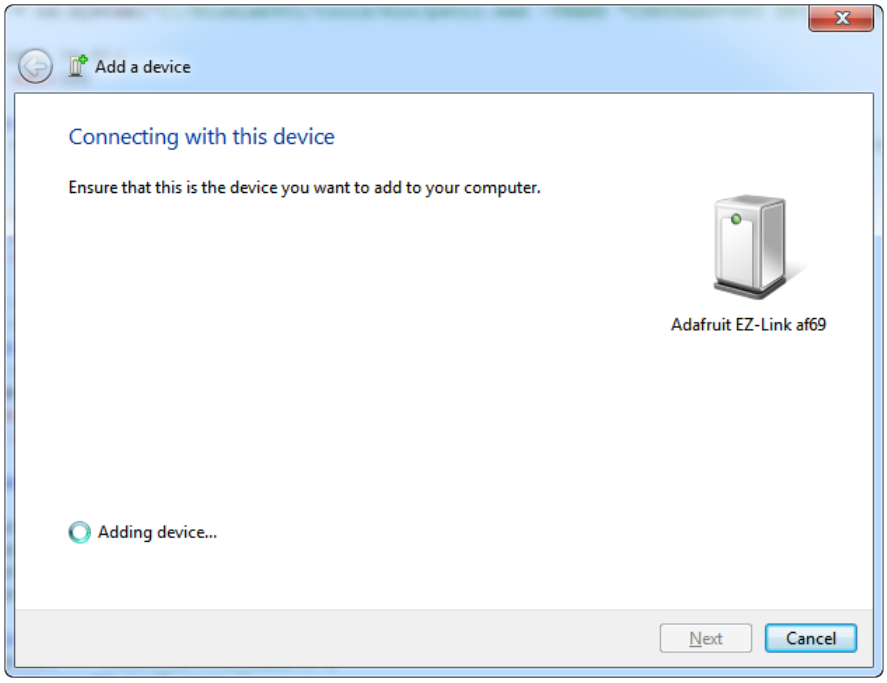


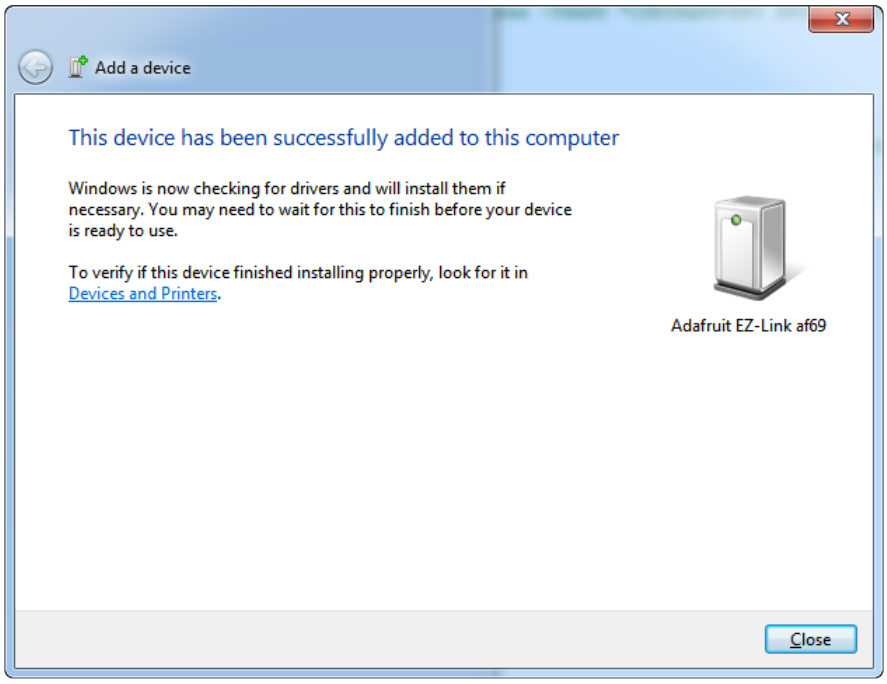


Either way, you'll get to a panel like this. Make sure the EZ-Link is plugged in and you see a once-every-two-seconds slow blink on the red LED. Within 30 seconds you should also see the **Adafruit EZ-Link xxxx** device appear. Now you can pair to this device.

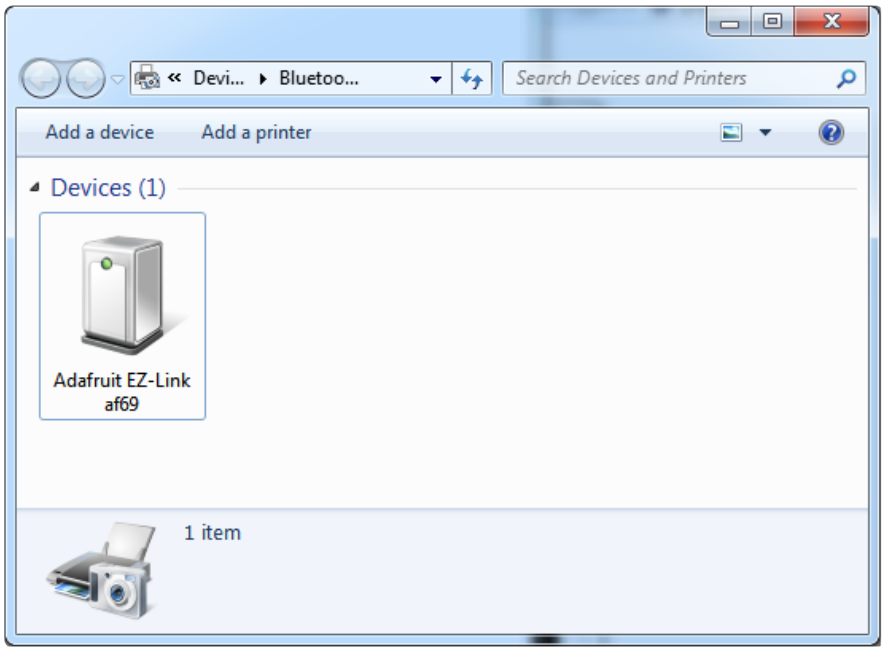


Click on it and select **Next** to add it



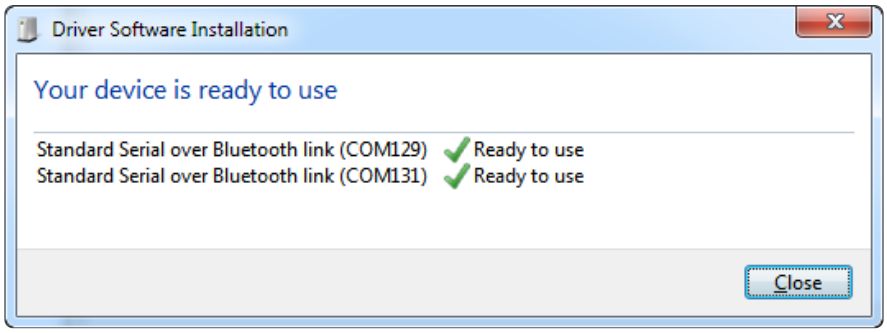


If you get asked for a pairing code, the code is **1234**
That's it, you're now paired!

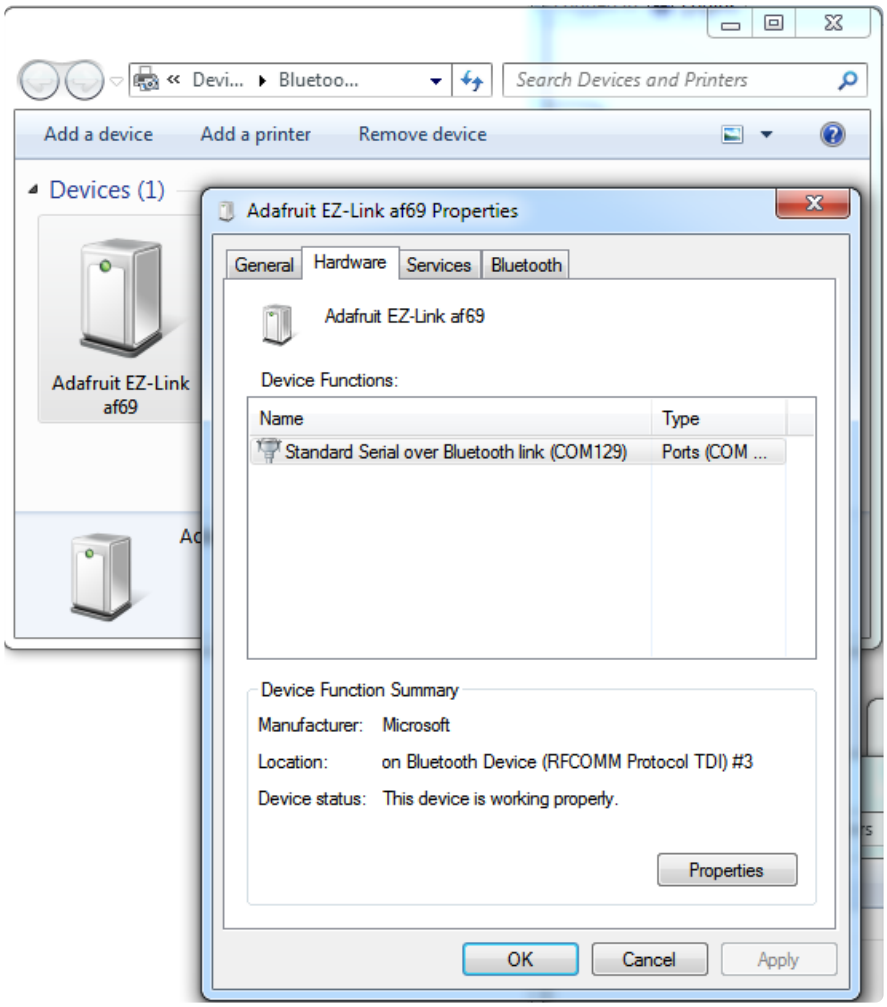


Next we will make sure we know the COM port to use. This is actually a little confusing because for some reason Windows creates two COM ports, but we only use one of them.

You may see the COM ports created by the Bluetooth driver, it'll look like this



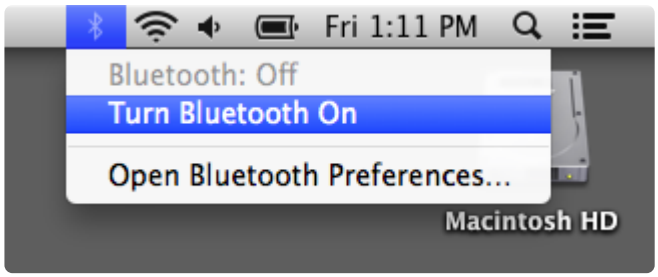
To figure out which COM port to use, go back to the Bluetooth device panel and right-click on the Adafruit EZ-Link and select **Properties** then the **Hardware** panel.



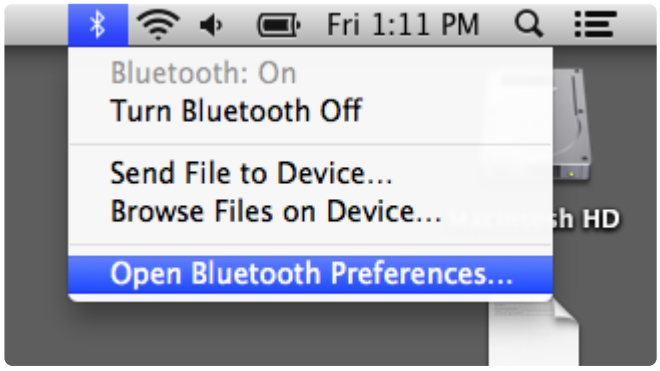
You'll see that only one COM port is mentioned here, in this case its **COM129**. That's the port you'll use when sending data to the EZ-Link

Mac OS X

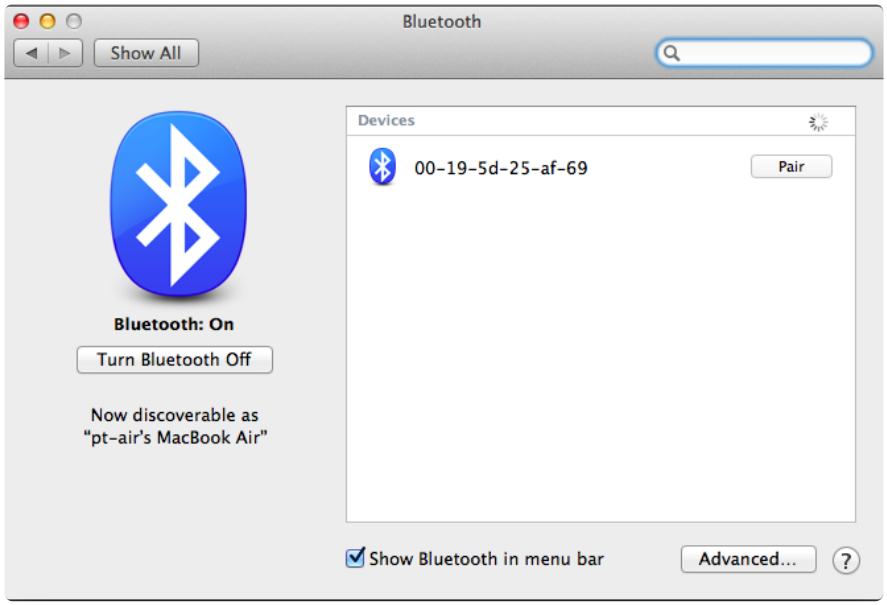
Pairing on a Mac is pretty easy. Start by turning on Bluetooth



Then go back to the BT menu and open the **Preferences...**

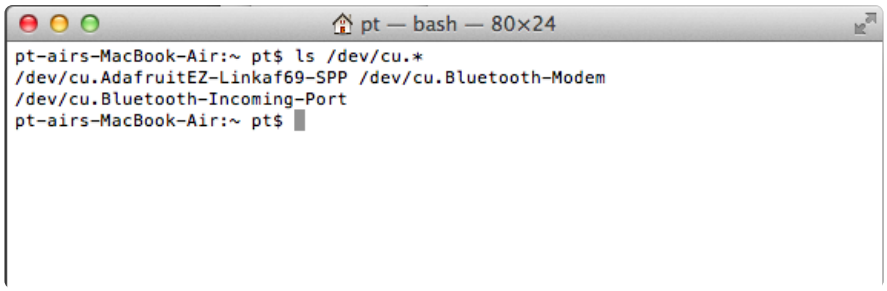


It will immediately start scanning for the device, you'll see the BT address first, then eventually it will turn into the EZ-Link name

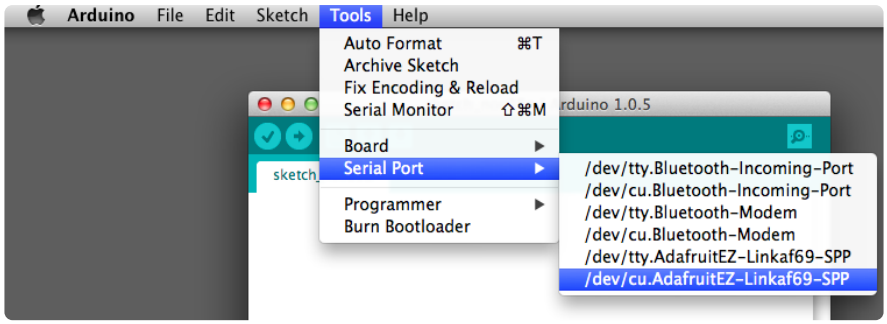


That's it! You're now paired to it. If you are asked for a code, the code is **1234**

Next you can figure out the name of the device by opening up a **Terminal** and typing in **ls /dev/cu.*** and looking for something like **/dev/cu.AdafruitEZLinkxxxx-SPP**



If you're using the Arduino IDE, it will show up under that name, use the **cu.** version not the **tty.** version



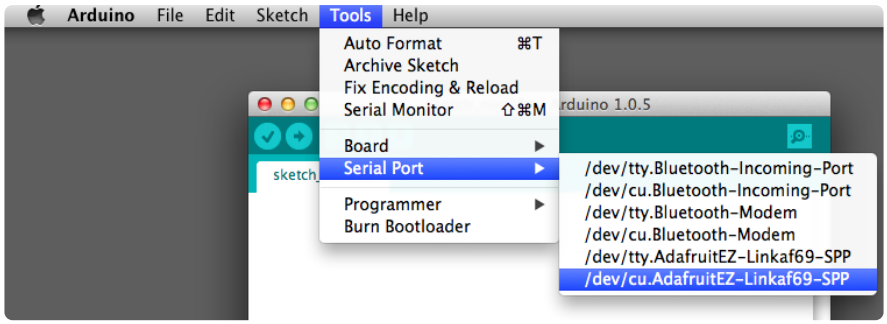
Programming with Arduino

This techique can be used for Arduino Uno, Duemilanove, Mega and any other ATmega328/168/1280/2560 based Arduinos that use a USB->Serial converter. It won't work with "USB" Arduinos such as the Leonardo/Micro/Flora since they need a direct USB connection

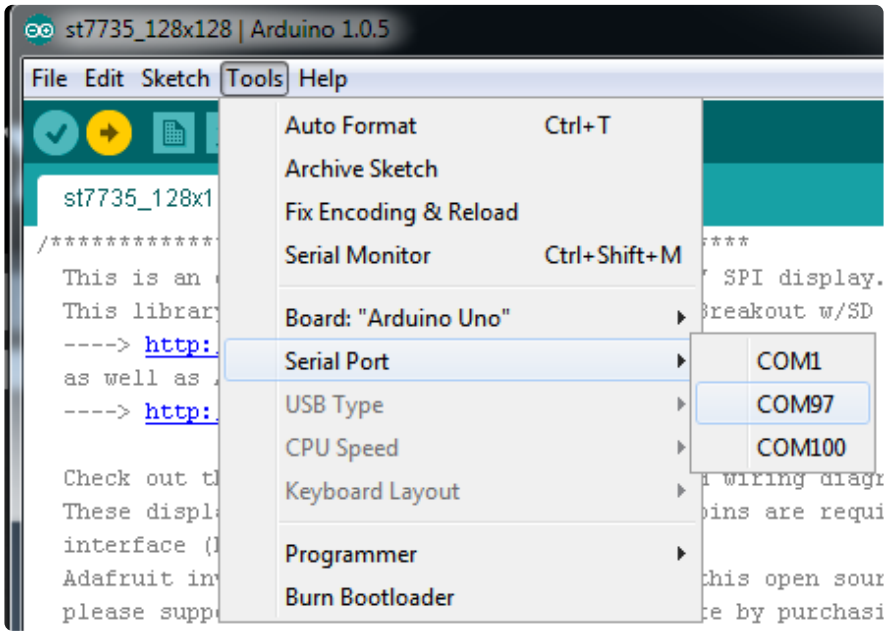
Start by making sure your Arduino + shield is powered, and that the Bluefruit shield switch is set to **Direct**

Go through the pairing procedure so that you have a COM port for the Bluefruit shield. Keep note of the COM port name (for windows, as there's two, one in and one output - confusing!)

On Macs, pick the menu labeled EZ-Link, the one starting with **cu.AdafruitEZ-Link**



Windows, pick the outgoing COM port, usually the lower of the two numbers



Upload

That's it!

If you open and close the Serial Monitor you should see the Arduino reset (the **L** pin #13 LED will blink)

Upload sketches as you normally would via the Upload button

Since opening and closing a bluetooth connection takes a few seconds, there will be an extra 10 second delay when starting the upload process. This is due to the Bluetooth wireless protocol and how the Arduino IDE opens and closes the connection a few times to reset the Arduino. Please be patient!

F.A.Q.

Can I use AT commands to configure the EZ-Link?

There is no command mode for EZ-Link, it is designed to be used out of the box. Since it automatically detects the baud rates there is no need to set that up. You can initiate pairing from any computer.

Can I change the BT name of my module?

At this time there is no way to change the BT name, it is hardcoded in and the last four digits match the lower two bytes of the MAC address

Can EZ-Link act as a BT 'master'?

No, it is a client-mode only device.

Hey! It's not working with my Android device!

Limit your Arduino sketch Serial communication speed to 9600 baud. This step is very important, if you try other baud rates the Android device will not be able to communicate with the Bluefruit EZ-link.

I'm sometimes getting odd errors on my Mac when uploading, says the port's busy?

The Mac bluetooth core sometimes doesn't fully release the bluetooth connection state as fast as we'd like. Wait a few seconds and try again!

Why is uploading sketches slower when using EZ-Link than using a USB cable?

Wireless just isn't as fast as wired. For example, with USB 'wired' data, the data is just 'sent' because the wire is trusted. With wireless/bluetooth, there has to be a lot more layers of checking and verification because there's no way to know if it was received and its suscepitble to noise. You can try to speed up by unchecking the Verify Firmware box in the Prefs of the Arduino IDE, that should cut it in half...but of course adds some risk!

I'm using Windows and unable to pair/connect to the Serial Port and/or the EZ-Link pairs but I cannot actually use it to send/receive data!

Make sure you do not have any extra Bluetooth drivers installed. In particular, make sure you don't have CSR/Harmony bluetooth drivers or any other non-Windows-native bluetooth stack. You should be using ONLY the native Windows bluetooth! Unpair the EZ-Link, uninstall any other drivers and reboot, then re-pair. For more details check out this forum thread:

https://forums.adafruit.com/viewtopic.php?f=53&t=63214&p=321810#p321622 (https://adafru.it/eca)

It this Bluetooth Low Energy?

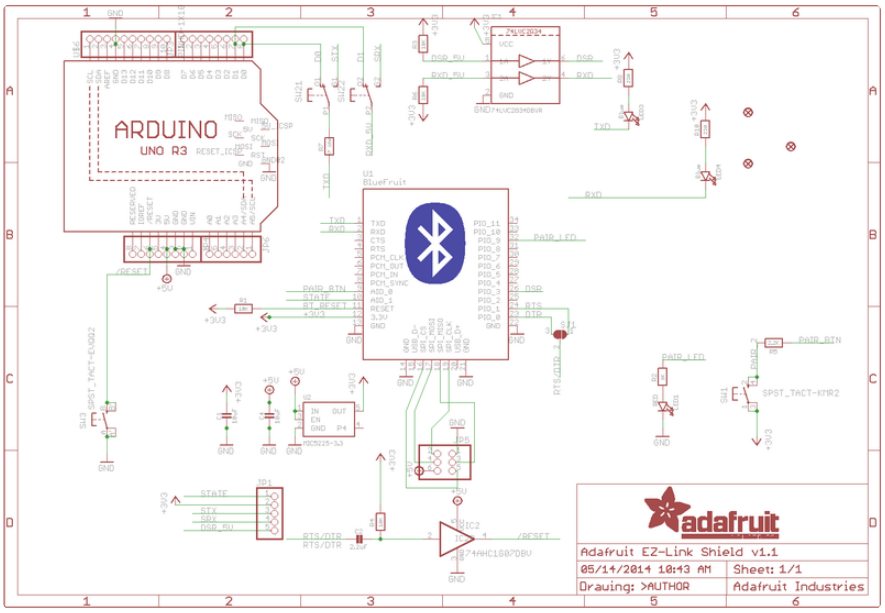
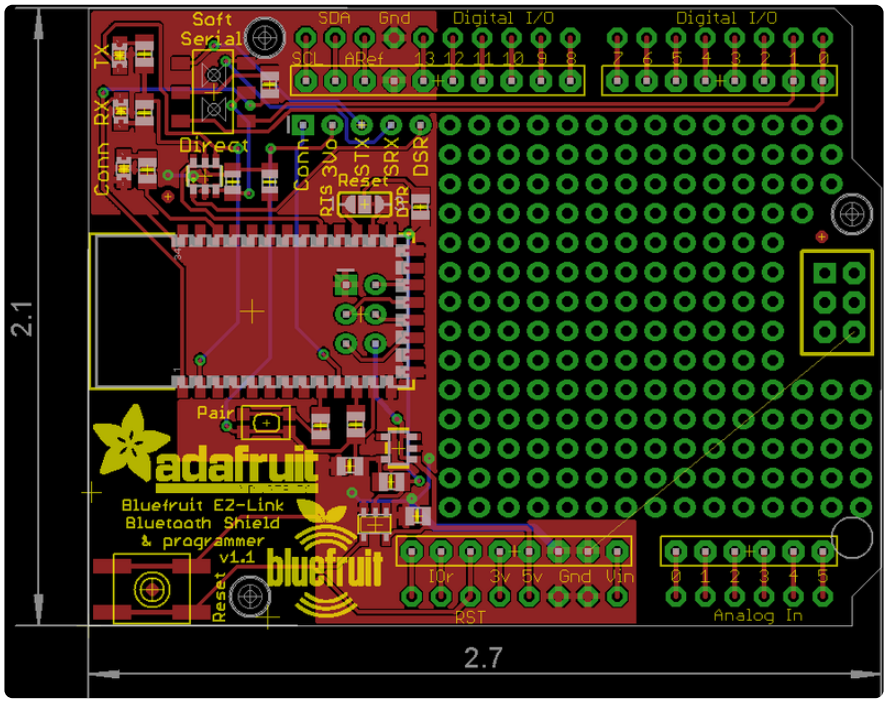
Nope! BLE is not supported on Windows 7, so we use Bluetooth Classic which is much more universal. In particular, this is a Bluetooth classic 2.0 - Class 2 (https://adafru.it/eg6)

Downloads

Files

- EagleCAD PCB files on GitHub (https://adafru.it/pC7)
- Fritzing object in Adafruit Fritzing library (https://adafru.it/c7M)

EZ-Link shield schematics



Fabrication Print