

Windows 10 BLE

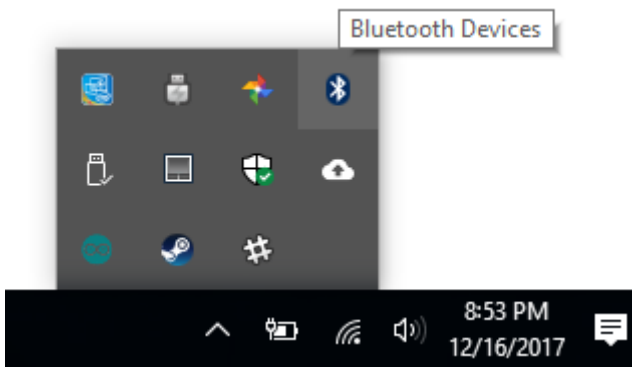
Jump to bottom

Timothy Woo edited this page on Dec 17 2017 · 11 revisions

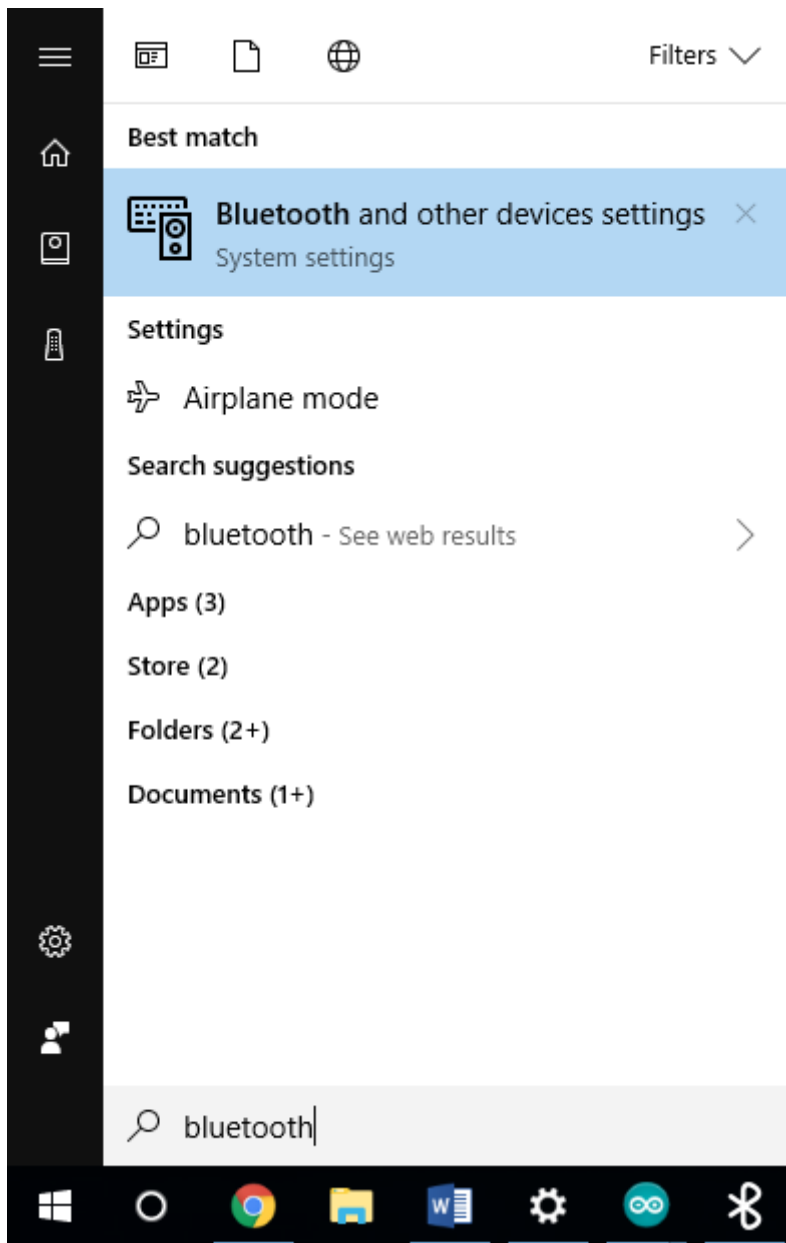
In this section we will be experimenting with how to communicate with Reflowduino using a Windows computer that has built-in Bluetooth 4.0. I've personally tested this with an Asus laptop that has Bluetooth 4.0. If you have a PC without built-in Bluetooth you can get a cheap BT 4.0 dongle on Amazon for around \$5. The following procedures also works on the HM-10 as well, with the notable exception that the default passkey is "1234" instead of "123456".

' Turn on Bluetooth

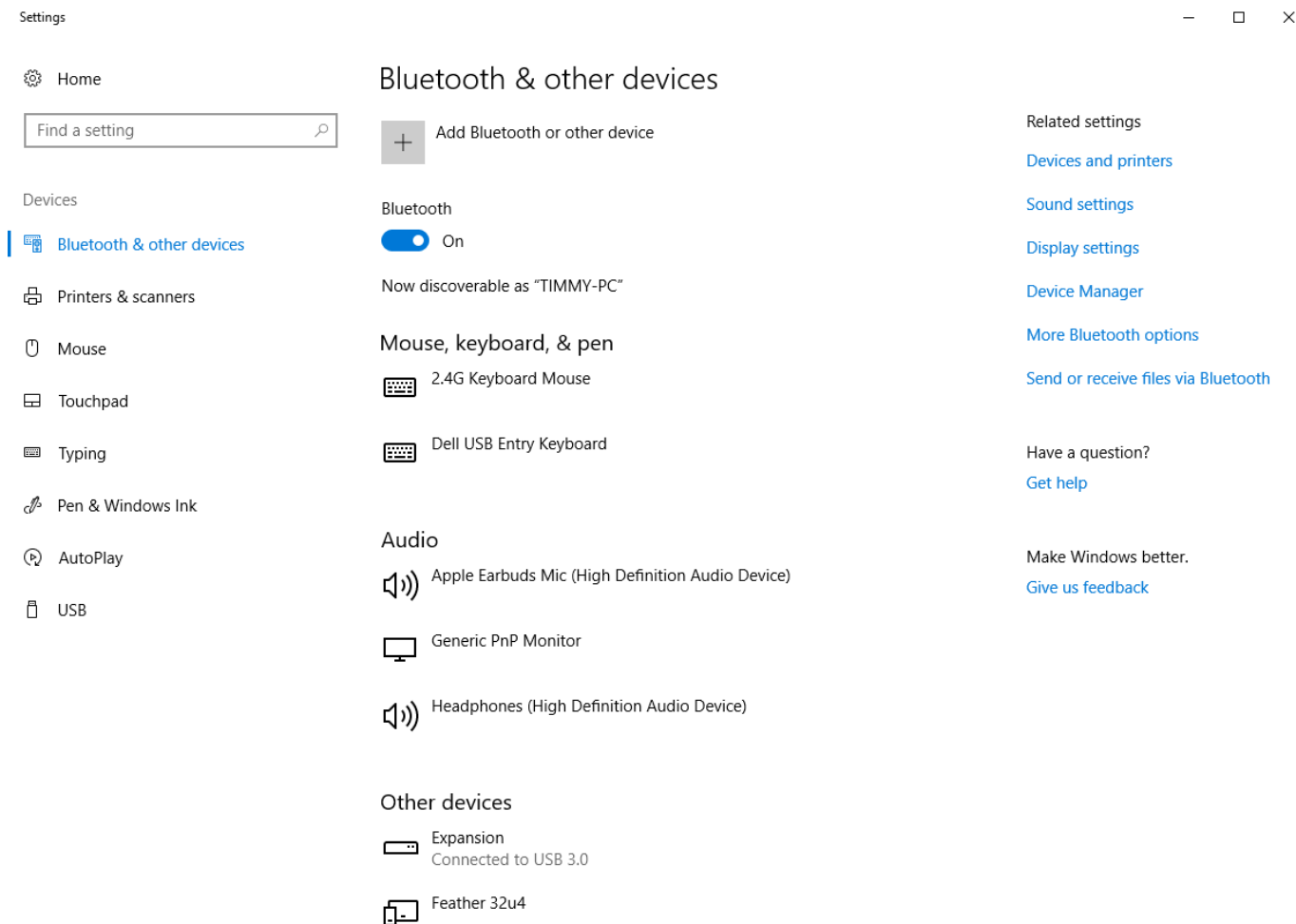
This is pretty obvious, but don't forget to do it! To turn on Bluetooth in Windows go to the task bar at the bottom right, click the little "expand" arrow and double click the Bluetooth icon.



Alternatively you can go to the Start menu and search "Bluetooth" and click on the first result that pops up.

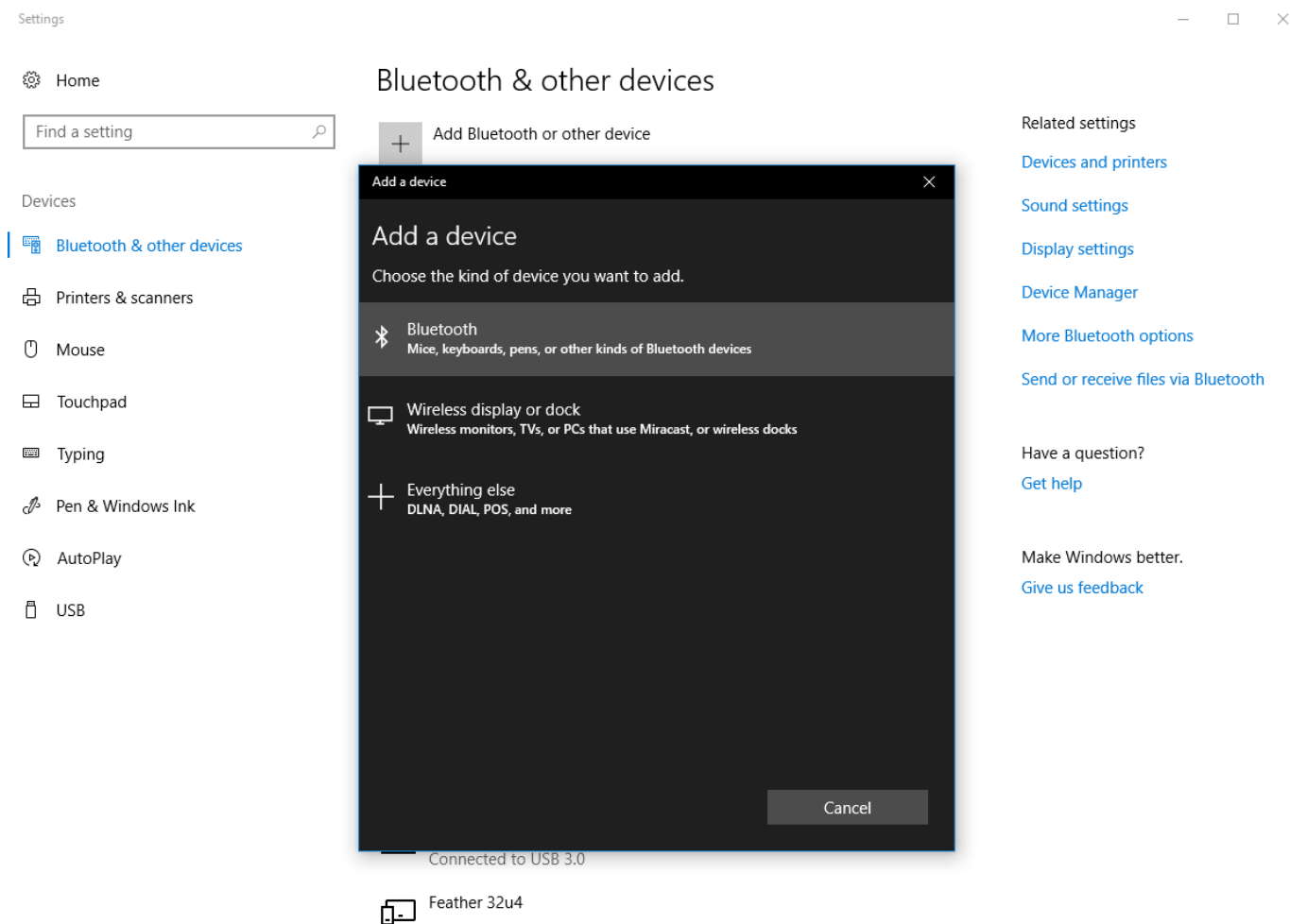


This should now bring you to the Bluetooth settings window. Turn on Bluetooth by clicking the little slide switch so that it's blue:

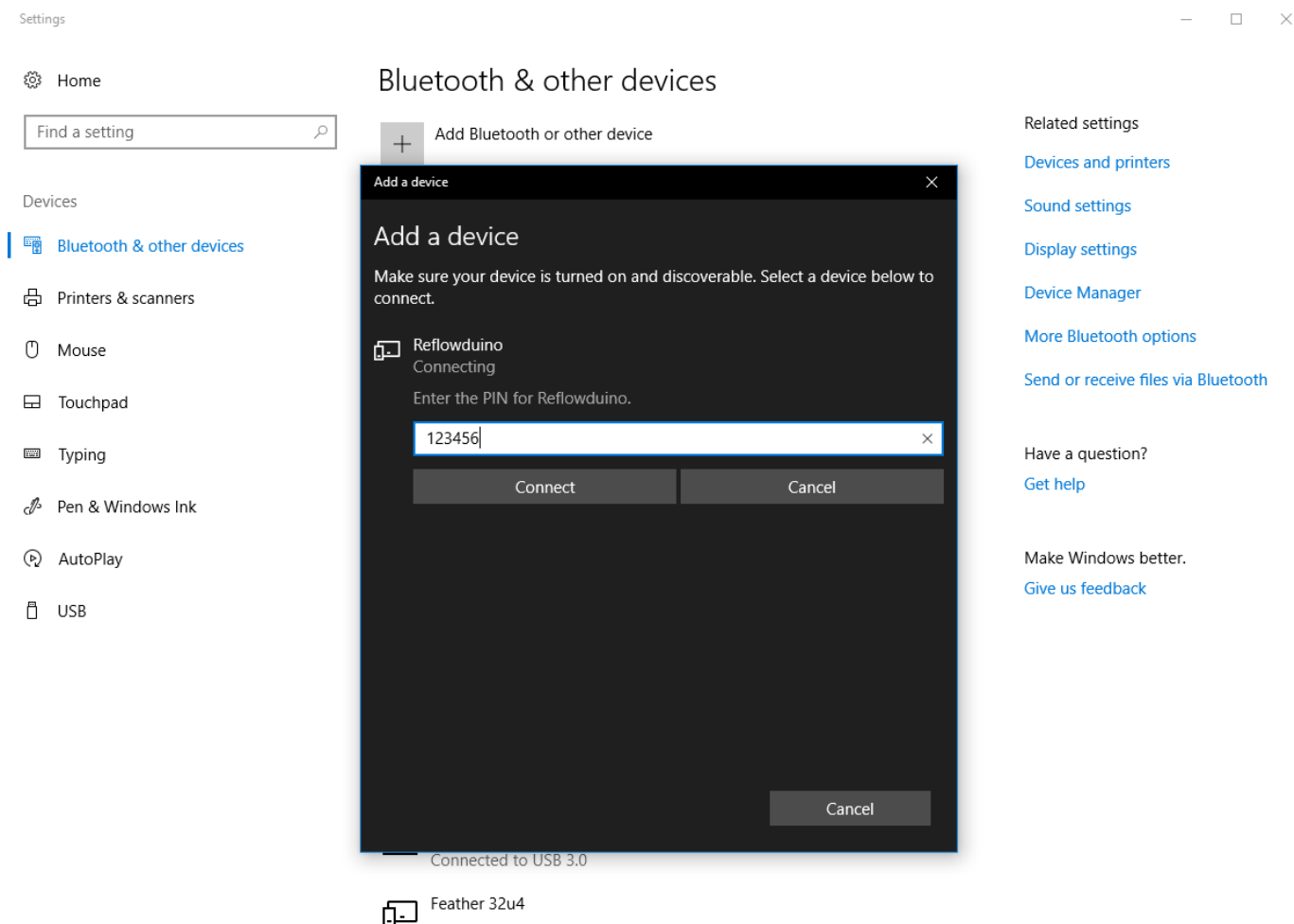


Pair with Reflowduino

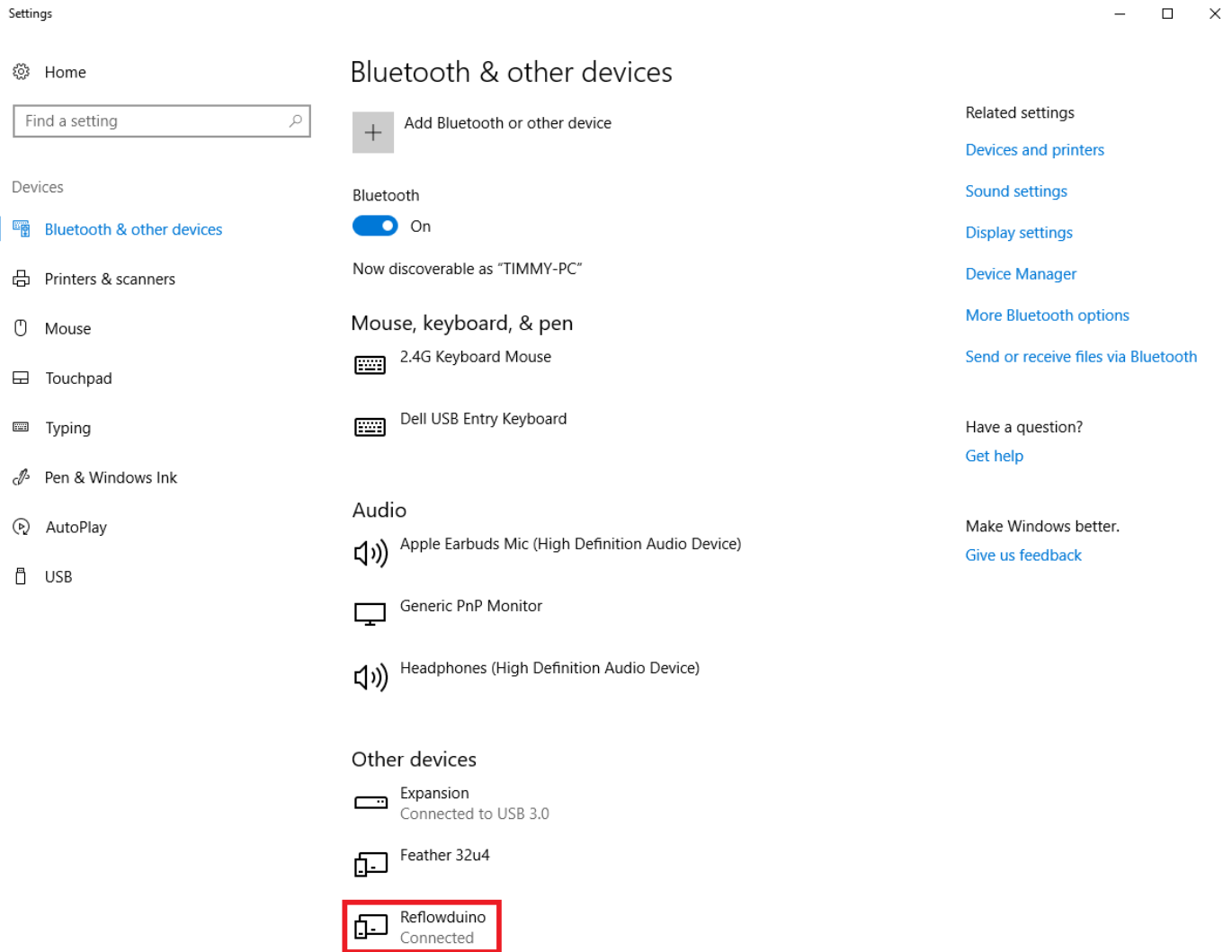
Now that you have Bluetooth turned on, click "Add Bluetooth or other device" at the top. This should bring up a window which shows available Bluetooth devices nearby, including your Reflowduino.



Click on "Reflowduino" and it will prompt you to enter a passcode to pair with it. Enter "123456" (without quotes) and click "Connect":

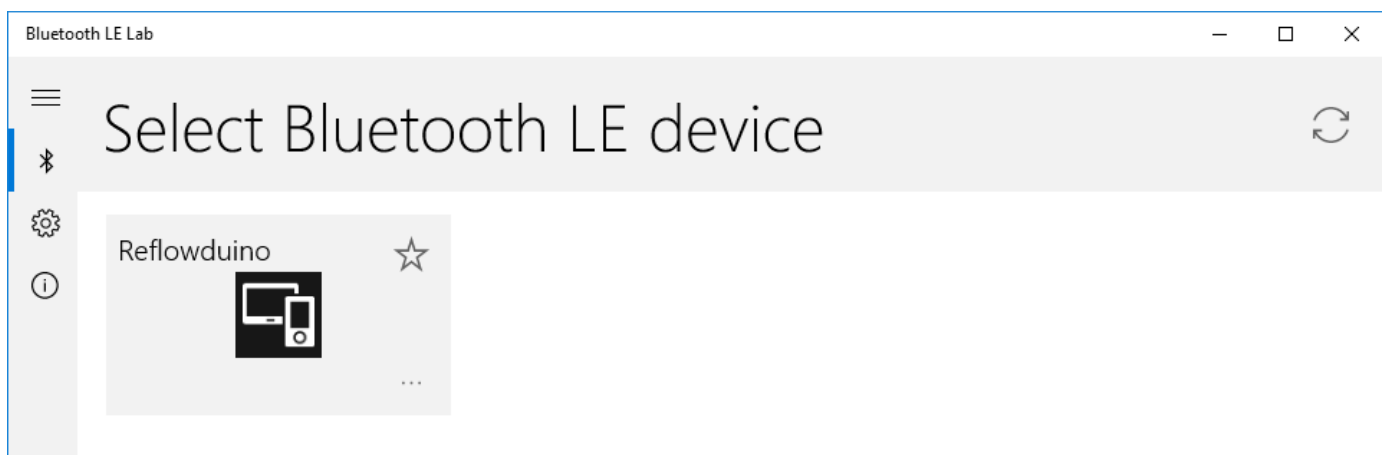


You should now see the Reflowduino appear in the list of "Other devices":



Windows BLE Application

Unfortunately by default we can't just create a Bluetooth COM port and talk with the module directly as we could do with traditional Bluetooth 2.0. Instead, we need to get a program that scans and connects to nearby BLE devices. A nice free one I found that works is Bluetooth LE Lab which has a clean interface and works great! Open the link in Microsoft Store, download and install it, and open it. You should be greeted by a screen that shows the Reflowduino.



Select the Reflowduino and you will see a list of services and characteristics. Look for the characteristic "0000FFE1-0000-1000-8000-00805F9B34FB". Note that you can deselect the "services" box to make it easier to locate this characteristic.

The screenshot shows the Bluetooth LE Lab application window. The title bar reads "Bluetooth LE Lab". The main header displays the device name "Reflowduino" and its MAC address "C4:F3:12:57:F6:E0". A "Disconnect device" button is in the top right. Below the header, there are three tabs: "Services", "Characteristics" (which is selected), and "Only favourite". A list of characteristics is shown, with the first one, "Unknown Characteristic" (UUID: 0000FFE1-0000-1000-8000-00805F9B34FB), highlighted in blue. To the right of this list, a detailed view for the selected characteristic is shown. It includes the title "Unknown Characteristic", a note that no description is available, the UUID, properties (WriteWithoutResponse, Write, Notify), attribute handle (2), and protection (Plain). Below this, there is a "Writing" section with a text input field, a "Write" button, and a checkbox for "Without response". The "Notifications" section has a checkbox for "Notify" (which is unchecked), a "Write state" button, a "Read state" button, and a "Subscribe" button. A "Notes" section at the bottom has a text input field with the placeholder "Leave your notes here...". A "Close" button is in the bottom right corner of the right-hand panel. At the bottom of the application window, a log shows the connection status: "9:02:31PM Trying to connect to device 'Reflowduino' (BluetoothLE#BluetoothLEe8:2a:ea:76:56:b4-c4:f3:12:57:f6:e0)" and "9:02:31PM+0.091s Successfully connected to 'Reflowduino'".

Optionally at the top right you can rename the characteristic to something like "TX/RX" to make it easier to identify in the future:

The screenshot shows the 'Bluetooth LE Lab' application window. The title bar says 'Bluetooth LE Lab'. The main header displays 'Reflowduino' with its MAC address 'C4:F3:12:57:F6:E0' and system device ID 'BluetoothLE#BluetoothLEe8:2a:ea:76:56:b4-c4:f3:12:57:f6:e0'. A 'Disconnect device' button is in the top right.

On the left, there are filters: 'Services' (unchecked), 'Characteristics' (checked), and 'Only favourite' (unchecked). Below these is a list of characteristics, each with a star icon:

- Unknown Characteristic 0000FFE1-0000-1000-8000-00805F9B34FB
- Unknown characteristic 0000FFE2-0000-1000-8000-00805F9B34FB
- Unknown characteristic 0000FEC7-0000-1000-8000-00805F9B34FB
- Unknown characteristic 0000FEC9-0000-1000-8000-00805F9B34FB
- Unknown characteristic 0000FEC8-0000-1000-8000-00805F9B34FB
- Device Name 00002A00-0000-1000-8000-00805F9B34FB
- Appearance 00002A01-0000-1000-8000-00805F9B34FB
- Peripheral Privacy Flag 00002A02-0000-1000-8000-00805F9B34FB
- Reconnection Address 00002A03-0000-1000-8000-00805F9B34FB
- Peripheral Preferred Connection Parameters 00002A04-0000-1000-8000-00805F9B34FB
- Service Changed 00002A05-0000-1000-8000-00805F9B34FB

On the right, the 'TX/RX' section shows 'No description available.' and UUID '0000FFE1-0000-1000-8000-00805F9B34FB'. Properties include 'WriteWithoutResponse', 'Write', and 'Notify'. Attribute handle is 2, and protection is Plain.

The 'Writing' section has a text input field, a 'UTF8' dropdown, a 'Write' button, and a 'Without response' checkbox.

The 'Notifications' section has a 'Notify' checkbox (unchecked), 'State unknown' text, 'Write state' and 'Read state' buttons, a 'Subscribe' button, and another 'UTF8' dropdown.

The 'Notes' section has a text input field with the placeholder 'Leave your notes here...' and a 'Close' button.

At the bottom, a log shows: '9:02:31PM Trying to connect to device 'Reflowduino' (BluetoothLE#BluetoothLEe8:2a:ea:76:56:b4-c4:f3:12:57:f6:e0)' and '9:02:31PM+0.091s Successfully connected to 'Reflowduino'.'

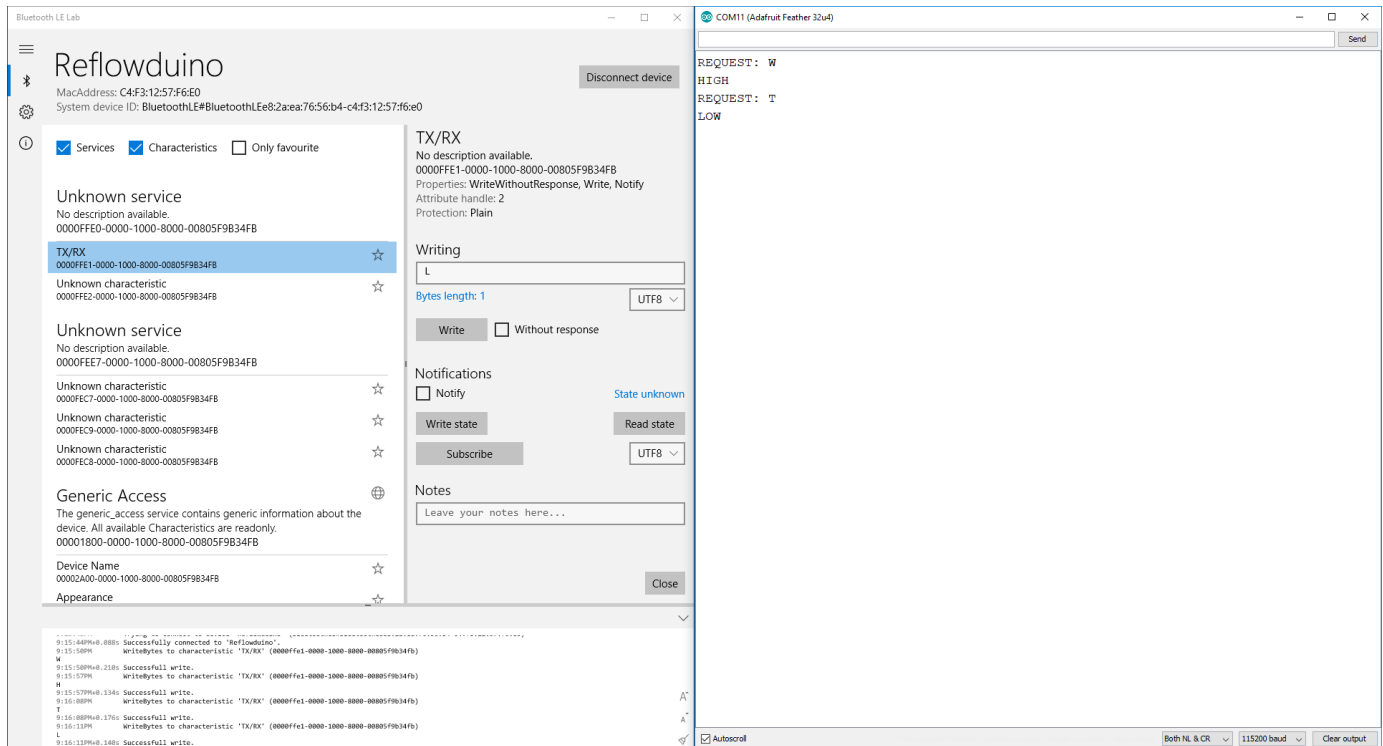
Now you're ready to start communicating with the Reflowduino via Bluetooth, but let's get some stuff set up on the Reflowduino board first so that we can verify that the two parties are actually communicating!

Reflowduino Setup

Open the code "Bluetooth_PC_Control" example sketch and upload it to the Reflowduino. After it uploads, open the serial monitor on the appropriate COM port. This code reads incoming data from the Reflowduino's BLE module and writes it in the serial monitor. You can also send "H" and "L" to turn on/off the red debug LED.

' Connect and start communicating!

Now that everything's set up, go back to the Bluetooth LE Lab window and under the characteristic mentioned earlier, make sure "UTF8" or "UTF16" is select, not "Auto". Next, type any letter in the "Writing" field and press "Write" to send the letter to the Reflowduino! In the serial monitor you should see the letter appear and the log at the very bottom of the Bluetooth LE Lab application should say "Successful write".



(Note: to make the image bigger, right click the image and click "Open image in new tab")

Write "H" to turn the red LED on, and "L" to turn it off, and experiment to your heart's content! Now the big question is, how do we make an open-source PC application where you can control the Reflowduino and graph data in real time to replace the Android app? I actually found this super cool open-source coding environment for GNU/Linux, Mac OS X, Windows, Android, and ARM called Processing that very closely resembles the Arduino IDE when you open it. It's packed with features and you can make some really nice GUI's with it with fairly simple code! Maybe more on that and a full Reflowduino application for your PC later!