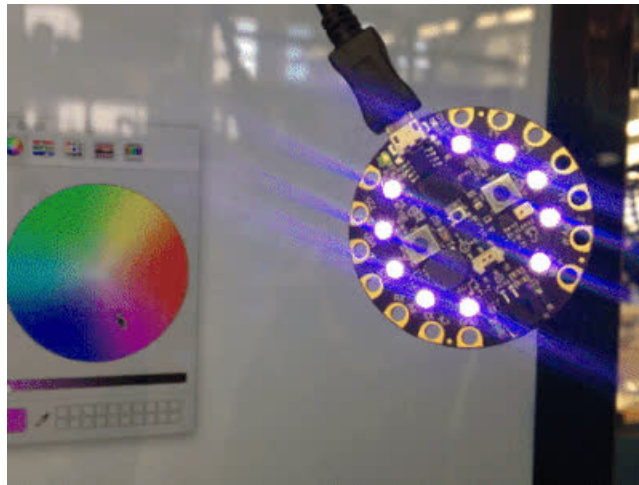


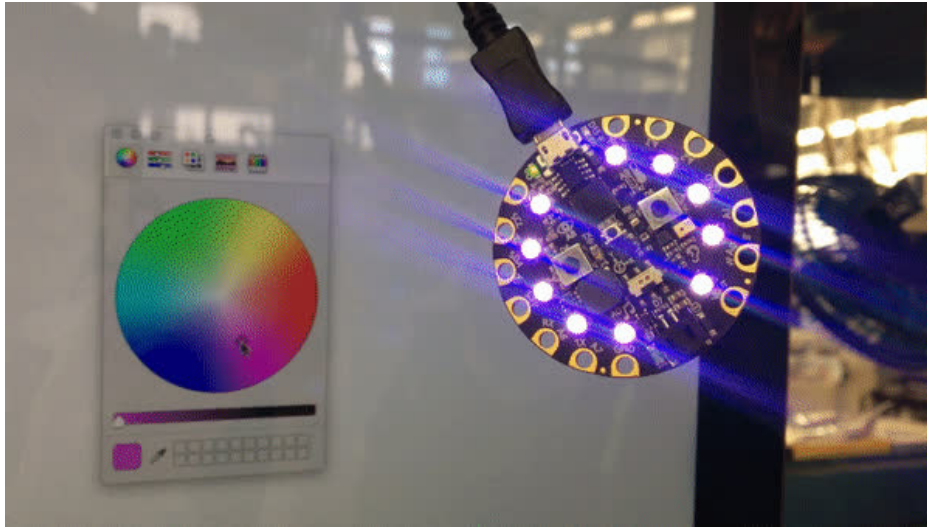
Using WebUSB with Arduino and TinyUSB

Created by Mike Barela



Last updated on 2019-08-27 08:26:51 PM UTC

Overview



This guide will show how the combination of the Open Source TinyUSB USB port software and the Chrome WebUSB browser capability provides programmers and users the ability to plug in microcontroller-based projects and have them interact with the user in a web browser. No additional code is required on the display computer other than a standard Chrome browser window (ie. no plug-ins, etc.)

This capability is ideal for schools, learning, ease of use, and assistive technology applications.



What is WebUSB?

[WebUSB \(https://adafru.it/FFD\)](https://adafru.it/FFD) is a recent standard for securely providing access to USB devices from web pages. It is available in Chrome 61 since September, 2017 and all current versions.

One of the reasons the WebUSB API has been created: to provide a way to expose USB device services to the Web. With this API, hardware manufacturers are able to build cross-platform JavaScript SDKs for their devices. But most importantly, this **makes USB safer and easier to use by bringing it to the Web**.

Let's see what you can expect with the WebUSB API:

1. Buy a USB device.
2. Plug it into your computer.
3. A notification appears right away, with the right website to go to for this device.
4. Simply click on it. Website is there and ready to use!
5. Click to connect and a USB device chooser shows up in Chrome, where you can pick your device.
6. Tada!

What would this procedure be like without the WebUSB API?

- Read a box, label, or search on line and possibly end up on the wrong website.
- Have to install a native application.
- Is it supported on my operating system? Make sure you download the "right" thing.
- Scary OS prompts popup and warn you about installing drivers/applications from the Internet.
- Malfunctioning code harms the whole computer. The Web is built to [contain malfunctioning websites \(https://adafru.it/FFE\)](https://adafru.it/FFE).
- Only use the USB device once? On the Web, the website is gone once you closed tab. On a computer the code sticks around.



How does TinyUSB Help?

[TinyUSB \(https://adafru.it/FFF\)](https://adafru.it/FFF) is an open-source cross-platform USB Host/Device stack for embedded system. It is designed to be memory-safe with no dynamic allocation and thread-safe with all interrupt events are deferred then handled in the stack's task function.

While there are some other USB stacks for microcontrollers, notably the Arduino stack, TinyUSB is compatible with a wide range of microcontrollers, has better device support, and is MIT licensed.

The TinyUSB library is on Adafruit's GitHub repo [here \(https://adafru.it/EWc\)](https://adafru.it/EWc).

WebUSB and TinyUSB Together

Adafruit has worked to ensure TinyUSB works with WebUSB. Together, they allow Adafruit and compatible microcontrollers to work with WebUSB browsers like Chrome with no drivers on the host computer/tablet/phone/chromebook. Super simple and this works well in environments like schools.

Computer and Mobile Hardware



Computers and Chrome

This project has been tested with computers running Windows 10, OSX, and Linux. The only requirement is the Chrome browser must be loaded and the version of the browser must be 61 or higher (best to be near the latest version).

A Chromebook with the latest software was tested and this project works great. The Chromebook is a Samsung model that dates from perhaps 2017. The ChromeOS software was updated to ensure the best compatibility.

Most desktops work fine with Chrome including PC, mac, and Linux. The browser is a bit of a memory hog but WebUSB compatibility doesn't add any significant memory use.

Mobile Devices

Support has been tested with iPhone and Android as they both support Chrome. There may be security prompts asking if it is ok for Chrome to use the USB port, to which you should answer yes.

Computer / Mobile Device USB Port

To use WebUSB, your computing device needs to have a free USB port (USB 2 is fine). Some devices do not have a USB port but some other connector (like Apple Lightning connectors). You will need an adaptor device or cable to convert that port to a USB port (and beware, not all adaptors are equally effective).

Some modern Android devices have a USB C port while older devices have a micro-B USB port.

The goal is getting a cable that will connect the version of USB present to a micro-B USB plug.

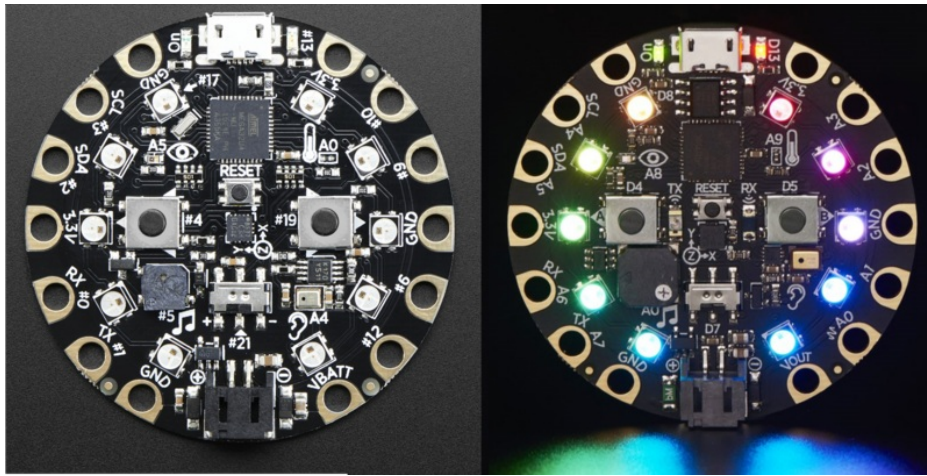
Here are some cables that may help bridge the gap between the microcontroller and the computer/mobile USB port:

- [Micro-B USB to micro-B USB OTG, 10 inches/25 cm cable \(https://adafru.it/BUg\)](https://adafru.it/BUg)
- [USB A to micro-B USB cable - 3 foot \(https://adafru.it/iia\)](https://adafru.it/iia) and [2 meter \(https://adafru.it/Cho\)](https://adafru.it/Cho)
- [USB C to micro-B USB cable \(https://adafru.it/FFG\)](https://adafru.it/FFG)
- [iOS Lightning to USB OTG Cable \(https://adafru.it/D4n\)](https://adafru.it/D4n)
- [USB 3-in-1 Sync and Charge micro-B / Type C / Lightning Cable \(https://adafru.it/FFH\)](https://adafru.it/FFH)

OTG stands for On the Go, and is required for some mobile devices to recognize external connections.

Compatible Microcontrollers

Adafruit has tested WebUSB and TinyUSB with Microchip (formerly Atmel) ATSAM21 and ATSAM51 based processor boards sold by Adafruit. The Nordic nRF52840 is also supported by TinyUSB.



Circuit Playground Express

This project will use the ATSAM21-based Adafruit Circuit Playground Express (CPX). This board has sensors, buttons, and NeoPixel LEDs to provide an excellent test platform. USB connectivity is via a micro-B USB connection.

The Circuit Playground Express supports a number of programming methods including CircuitPython, Microsoft MakeCode, and Arduino.

Adafruit has developed demonstrations of WebUSB and TinyUSB in Arduino as of the publication of this guide.

Install the Arduino IDE



The Arduino IDE (Integrated Development Environment) is a graphics interface to a language based on C/C++. It is used extensively for programming a very wide variety of microcontrollers.

If you do not have the latest version of the Arduino IDE software, it is suggested you download and install the latest version.

The latest versions of the Arduino IDE have a library manager and board manager which are needed. It also has a new feature to switch between the ARduino USB Stack and the TinyUSB stack. for those reasons, using the WebUSB code should be used only on very recent IDE versions.

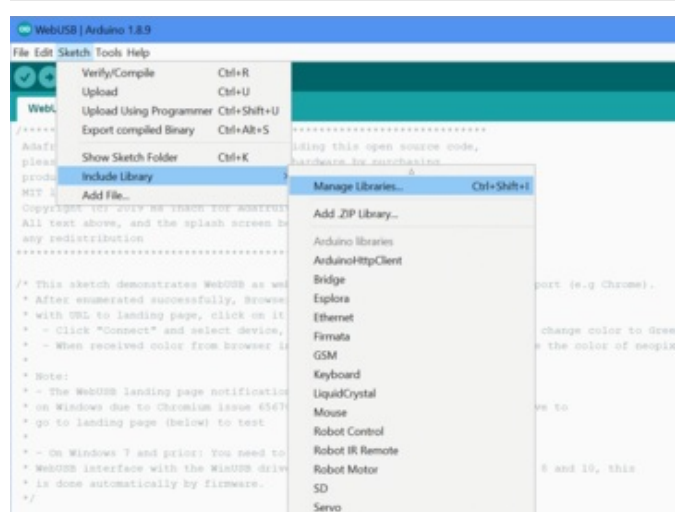
See the following pages on the steps to download the IDE and install it on your operating system.

- [Download Software \(https://adafru.it/FFJ\)](https://adafru.it/FFJ)
- [Install Software Windows \(https://adafru.it/FFK\)](https://adafru.it/FFK)
- [Install Software Mac \(https://adafru.it/FFL\)](https://adafru.it/FFL)
- [Install Software Linux \(https://adafru.it/FFM\)](https://adafru.it/FFM)

Library, Board, and TinyUSB Selection

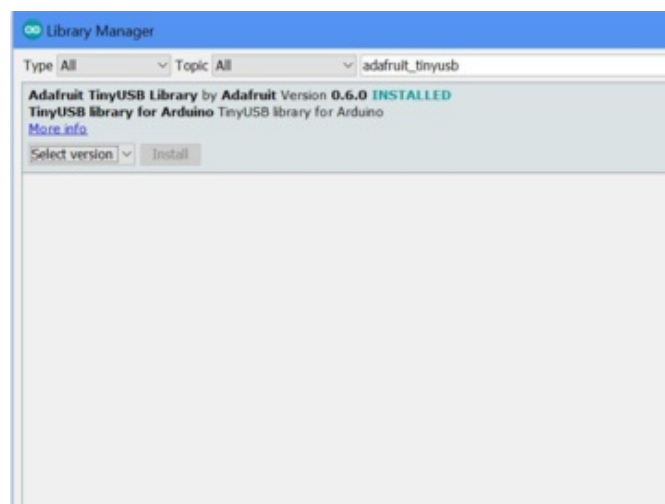
Once you have the Arduino IDE installed or updated, you will need to set some parameters to get the right libraries, board definition file, and the TinyUSB stack.

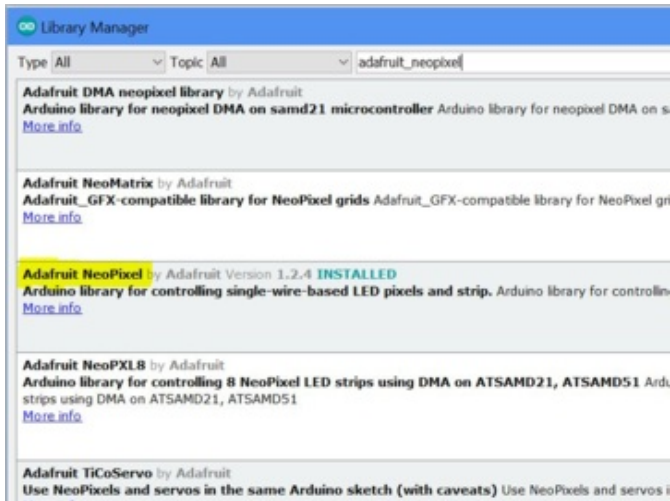
Install Libraries



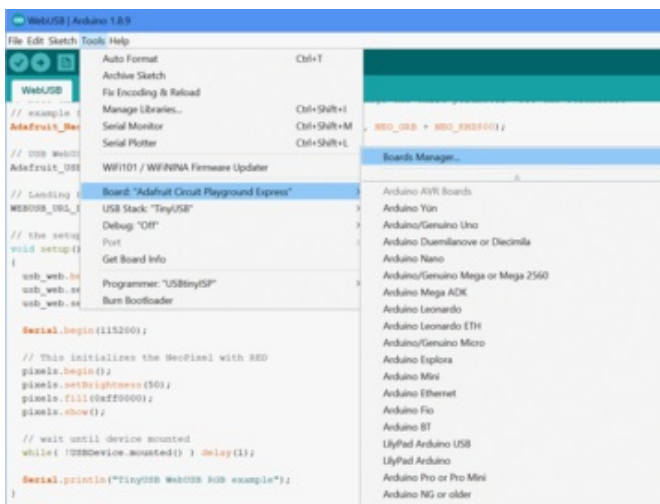
In the Arduino IDE menus, go to **Sketch -> Include Library -> Manage Libraries**. You will need to ensure you have the latest versions of the following libraries:

- **Adafruit_TinyUSB**
- **Adafruit_NeoPixel**



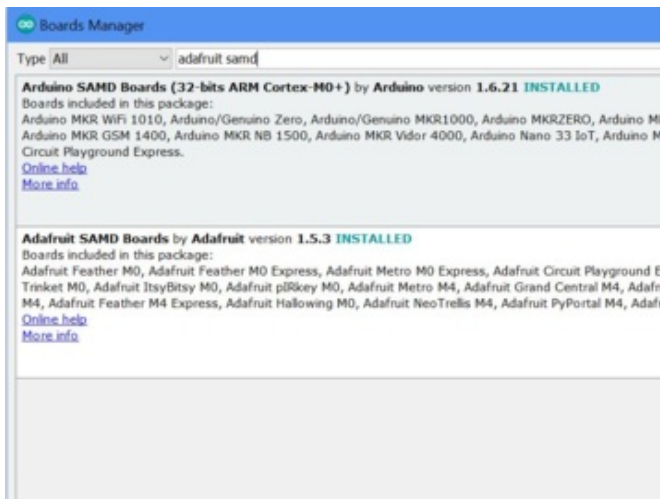


Board Definition File

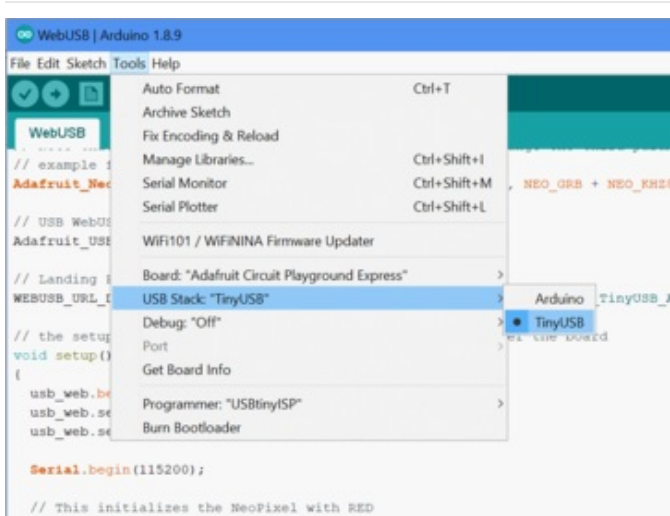


You will need to test this on an Adafruit SAMD board such as the Circuit Playground Express, Metro Express M0/M4, Feather Express M0/M4, and others.

In the menu go to **Tools -> Board -> Boards Manager**. Type in **Adafruit SAMD** to search. It will likely pull up both the Arduino SAMD board package and the Adafruit SAMD board package. Select the **Adafruit SAMD** package and ensure it is at least version 1.5.3 or higher for TinyUSB and WebUSB work.



Using the TinyUSB Stack

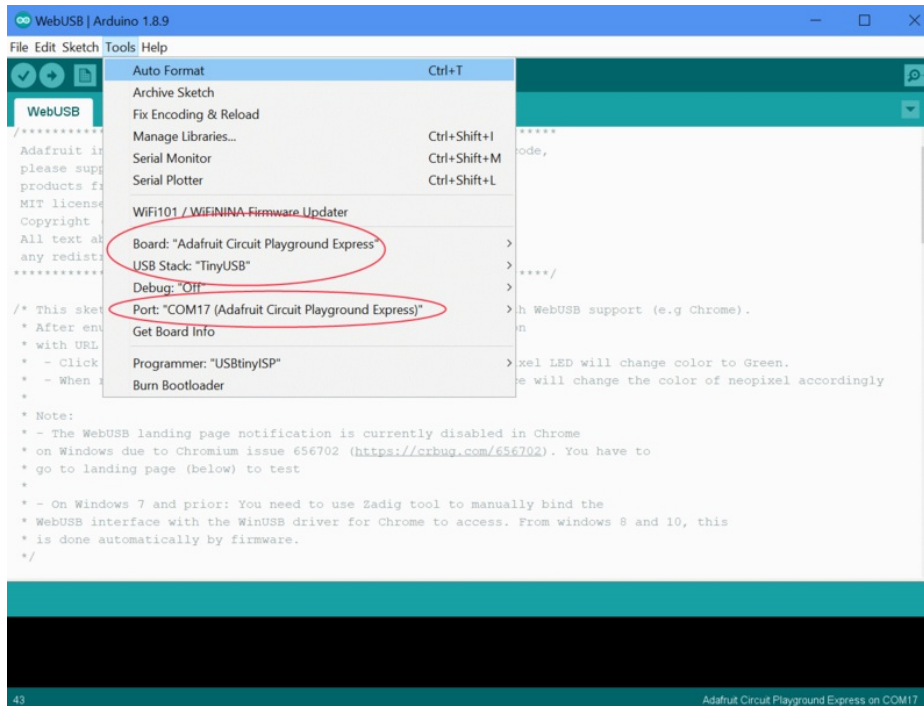


Finally, go to **Tools -> USB Stack** and select **TinyUSB**. If this is not selectable, update your version of the Arduino IDE. Switching the USB stack puts in TinyUSB instead of the default Arduino USB stack. TinyUSB has more features including WebUSB support.

All set for the Circuit Playground Express

The screen below shows the Arduino IDE with a Circuit Playground Express board. The CPX is plugged in and in bootloader mode by tapping Reset once or twice and seeing a circle of green lights.

See the settings circled in red.



Ok, that is all the configuration needed to work with the code in the two examples next.

RGB Color Picker Example

This sketch demonstrates a microcontroller receiving color information from a browser with WebUSB support (re. Chrome).

Quickstart for Circuit Playground Express

If you're using a Circuit Playground Express, you can download this UF2, and install it by double-clicking into **CPLAYBOOT** mode and dragging over the UF2 file

<https://adafru.it/FFV>

<https://adafru.it/FFV>

Load the Example Code to Your Microcontroller

Click Download in the code window below and save the Arduino sketch in your Arduino file folder. Open the code with the Arduino IDE **File** -> **Open** command.

Temporarily unable to load content:

Ensure you have set up the IDE per the previous instructions for the board type, correct library, and use of TinyUSB.

Connect the microcontroller to your computer/device via the USB cabling discussed previously.

Place your Circuit Playground Express in bootloader mode by pressing the tiny onboard reset button once or twice (depending on how it was programmed earlier). The board should flash red briefly and then display a green circle of lights. Select the correct port for the board in the IDE by going to **Tools** -> **Port** in the menus and you should see the Circuit Playground Express listed as a possible selection.

Use **Sketch** -> **Upload** (or the circle with the right arrow button) to load the Arduino code onto the Circuit Playground Express. If you get any compile errors, look on the previous page for library requirements, the correct board definition file, and selecting TinyUSB.

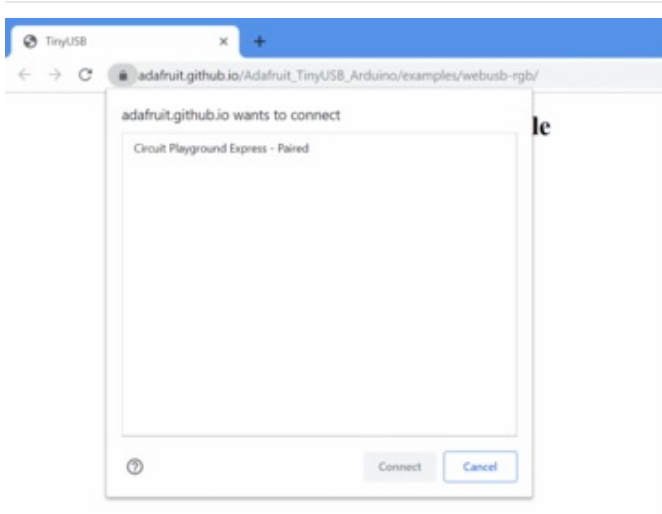
After upload, the ring of NeoPixel LEDs will turn red. The board is now ready to test.

Use

After enumerating successfully Chrome will pop-up a notification with a URL to a landing page, click on it to test. On Windows there is a [known bug \(https://adafru.it/FFP\)](https://adafru.it/FFP), you'll have to click on the button below to go to the test web page.

<https://adafru.it/FFQ>

<https://adafru.it/FFQ>



Click **Connect** and select your microcontroller. For example: Circuit Playground Express.

If you see text "Unknown Device", that can happen and it should connect if selected. Changing USB ports often clears this text up.



When connected the NeoPixel LEDs will change color to Green, the same color as on screen.



Now click the green circle. You will see an operating system dependent pop-up asking you to pick a new color for the NeoPixels. Select a color and click "Ok". I'll pick blue for my choice.

You will then see the web page again and all the LEDs will now be blue (or whatever color you chose).

TinyUSB - WebUSB RGB Example



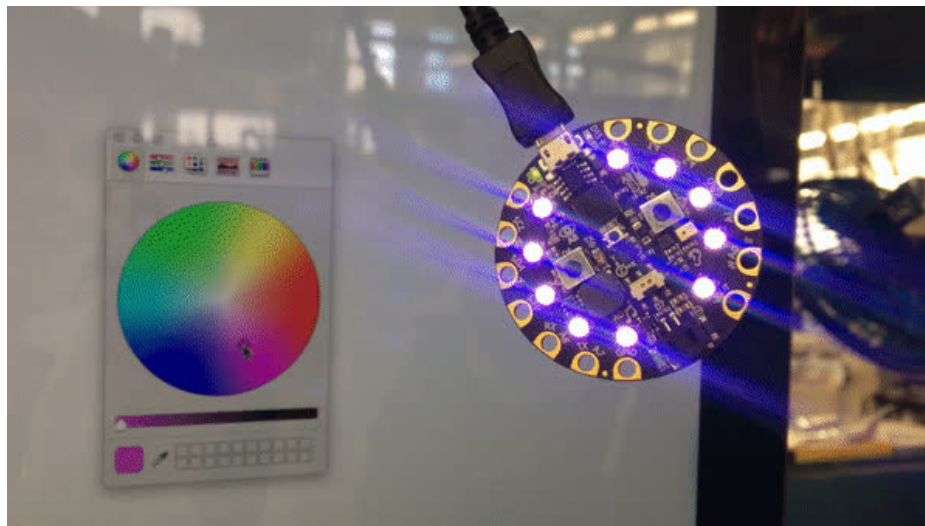
Behind the Scenes

The computer/mobile device will, behind the scenes, send a color value from the browser in the format '#RRGGBB'. The Arduino sketch will change the color of the microcontroller's onboard NeoPixels accordingly.

When connected, this process may continue indefinitely, the board will change color to whatever color you chose in the browser.

This demonstrates one way communication from the browser to the microcontroller.

When done, you can click the **Disconnect** button to stop communications.



Serial Communications Example

This sketch demonstrates WebUSB sending text from the browser to a microcontroller which in turn sends the text back to the browser.

Load the Example Code to Your Microcontroller

Click Download in the code window below and save the Arduino sketch in your Arduino sketches folder. Open the code with the Arduino IDE **File** -> **Open** command.

```
/*
*****
Adafruit invests time and resources providing this open source code,
please support Adafruit and open-source hardware by purchasing
products from Adafruit!

MIT license, check LICENSE for more information
Copyright (c) 2019 Ha Thach for Adafruit Industries
All text above, and the splash screen below must be included in
any redistribution
*****
*/

/* This sketch demonstrates WebUSB as web serial with browser with WebUSB support (e.g Chrome).
 * After enumerated successfully, Browser will pop-up notification
 * with URL to landing page, click on it to test
 * - Click "Connect" and select device, When connected the on-board LED will litted up.
 * - Any charters received from either webusb/Serial will be echo back to webusb and Serial
 *
 * Note:
 * - The WebUSB landing page notification is currently disabled in Chrome
 * on Windows due to Chromium issue 656702 (https://crbug.com/656702). You have to
 * go to landing page (below) to test
 *
 * - On Windows 7 and prior: You need to use Zadig tool to manually bind the
 * WebUSB interface with the WinUSB driver for Chrome to access. From windows 8 and 10, this
 * is done automatically by firmware.
 */

#include "Adafruit_TinyUSB.h"

// USB WebUSB object
Adafruit_USBD_WebUSB usb_web;

// Landing Page: scheme (0: http, 1: https), url
WEBUSB_URL_DEF(landingPage, 1 /*https*/, "adafruit.github.io/Adafruit_TinyUSB_Arduino/examples/webusb-ser

int led_pin = LED_BUILTIN;

// the setup function runs once when you press reset or power the board
void setup()
{
  pinMode(led_pin, OUTPUT);
  digitalWrite(led_pin, LOW);

  usb_web.begin();
  usb_web.setLandingPage(&landingPage);
  usb_web.setLineStateCallback(line_state_callback);
}
```

```

Serial.begin(115200);

// wait until device mounted
while( !USBDevice.mounted() ) delay(1);

Serial.println("TinyUSB WebUSB Serial example");
}

// function to echo to both Serial and WebUSB
void echo_all(char chr)
{
  Serial.write(chr);
  // print extra newline for Serial
  if ( chr == '\r' ) Serial.write('\n');

  usb_web.write(chr);
}

void loop()
{
  // from WebUSB to both Serial & webUSB
  if (usb_web.available()) echo_all(usb_web.read());

  // From Serial to both Serial & webUSB
  if (Serial.available()) echo_all(Serial.read());
}

void line_state_callback(bool connected)
{
  digitalWrite(led_pin, connected);

  if ( connected ) usb_web.println("TinyUSB WebUSB Serial example");
}

```

Ensure you have set up the IDE per the previous instructions for the board type, correct library, and use of TinyUSB.

Connect the microcontroller to your computer/device via the USB cabling discussed previously.

Place your Circuit Playground Express in bootloader mode by pressing the tiny onboard reset button once or twice (depending on how it was programmed earlier). The board should flash red briefly and then display a green circle of lights. Select the correct port for the board in the IDE by going to **Tools -> Port** in the menus and you should see the Circuit Playground Express listed as a possible selection.

Use **Sketch -> Upload** (or the circle with the right arrow button) to load the Arduino code onto the Circuit Playground Express. If you get any compile errors, look on the previous page for library requirements, the correct board definition file, and selecting TinyUSB.

After upload, the ring of NeoPixel LEDs on the CPX will turn off. The board is now ready to test.

Use

There is a known bug that when a WebUSB board is connected to an Android device, this example sends text but does not receive it back correctly. This is being looked into. All other devices (PC, mac, Linux,

Chromebook, etc) work as expected.

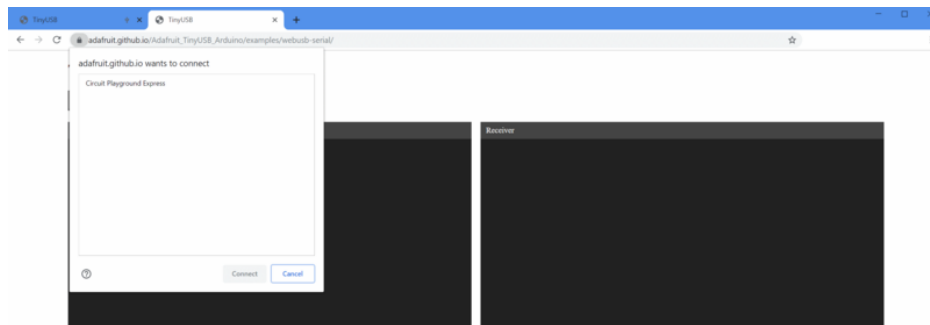
After enumerating successfully, Chrome will pop-up a notification with a URL to a landing page, click on it to test. On Windows there is a [known bug \(https://adafruit.it/FFP\)](https://adafruit.it/FFP), you'll have to click on the button below to go to the test web page.

<https://adafruit.it/FFS>

<https://adafruit.it/FFS>

Click "Connect" and select your microcontroller. For example: Circuit Playground Express.

When connected, the right Receiver window will display the text "TinyUSB WebUSB Serial example" - this is the board sending its first text message to the browser via WebUSB.

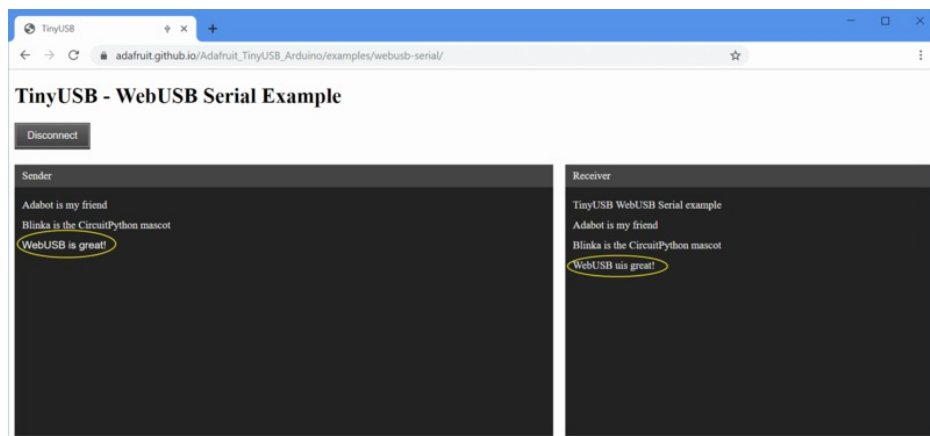


Click inside the left Sender window with your mouse or cursor.

Type something such as "Adabot is my friend"

As you type, the characters are echoed in the receiver window.

The last line shows that I used a backspace to correct a typo. The echo routine in the Arduino example code doesn't recognize backspace as going back one character so the mistyped character is echoed in the Receiver window.



Behind the Scenes

The computer/mobile device sends character data from the browser Sender window to the microcontroller via the USB serial connection. The microcontroller looks for text and immediately sends the text back via WebUSB to the browser where it appears in the Receiver window.

When connected, this process may continue indefinitely, the board will continue echoing typed text. The microcontroller does not recognize extended keyboard input at present (as shown with backspace).

This demonstrates two way communications from the browser to the microcontroller. The Arduino program can be changed to do any manner of serial communication, unidirectional or bidirectional.

When done, you can click the Disconnect button to stop communications.

