



Getting Started with Braille Output for CircuitPython REPL

Created by Isaac Wellish



<https://learn.adafruit.com/getting-started-braille-output-circuitpython-repl>

Last updated on 2024-03-08 02:53:45 PM EST

Table of Contents

Overview	3
<hr/>	
<ul style="list-style-type: none">• What's CircuitPython.... and a REPL?• Prerequisite guides• Parts• Optional	
Connecting to the Reader	4
<hr/>	
<ul style="list-style-type: none">• Orientation of Keys• Operating System• Additional Resources	
Connecting on Mac	6
<hr/>	
<ul style="list-style-type: none">• Switching to HID mode• Connect and power on the reader• Enable HID mode for reader• Turn on "VoiceOver" application• Having trouble connecting?• Still not working?• Additional Resources	
Connecting on PC	10
<hr/>	
<ul style="list-style-type: none">• Download necessary drivers• Switching to HID mode• Connect and power on the reader• Enable HID mode for reader• Connecting to the Orbit Reader• NVDA• JAWS• Dolphin Screen Reader• Have a screen reader that does not support HID?	
Outputting the REPL to the Reader	14
<hr/>	
Advanced Serial Console on Mac	15
<hr/>	
<ul style="list-style-type: none">• What's the Port?• Connect with screen	
Advanced Serial Console on Windows	17
<hr/>	
<ul style="list-style-type: none">• Windows 7 and 8.1• What's the COM?• Install Putty	

Overview

This project is a great way to learn and use [CircuitPython](https://adafru.it/CgS) (<https://adafru.it/CgS>) for those that may be visually impaired or blind. Or if you just wanna read code with your fingers instead of your eyes, that's cool too.

The [Orbit Reader 20](https://adafru.it/FDk) (<https://adafru.it/FDk>) is a hardware device that outputs content from a computer as Braille. Know what that means? We can use it to send the CircuitPython REPL from a computer screen to the Orbit via the screen reader. The Orbit Reader outputs text from the [REPL](https://adafru.it/Awz) (<https://adafru.it/Awz>) and converts it to Braille in real time!

Please note: you cannot connect the Orbit directly to the CircuitPython board without a computer, at this time.



What's CircuitPython.... and a REPL?

[CircuitPython](https://adafru.it/FGN) (<https://adafru.it/FGN>) is a programming language designed to simplify experimenting and learning to program on low-cost microcontroller boards.

The REPL, or **Read-Evaluate-Print-Loop**, allows you to enter individual lines of code and have them run immediately. It's really handy if you're running into trouble with a particular program and can't figure out why. It's interactive so it's great for testing new ideas.

Prerequisite guides

New to CircuitPython and REPLs? Check out the following guide.

- [Welcome to CircuitPython! \(https://adafru.it/BIM\)](https://adafru.it/BIM)

Parts

- [Orbit Reader 20 \(https://adafru.it/FDk\)](https://adafru.it/FDk)(a Micro-USB cable is included)
- [A CircuitPython device \(https://adafru.it/zB0\)](https://adafru.it/zB0)*
- [A micro-usb cable \(http://adafru.it/592\)](http://adafru.it/592)

*In addition to the screen reader, you will need some sort of CircuitPython enabled device to read the REPL output from. In this tutorial, we are using an Adafruit PyPortal but you may use any device that can run CircuitPython!

[Here's a list of available boards \(https://adafru.it/zB0\)](https://adafru.it/zB0) to choose from.

Optional



[Adafruit PyPortal - CircuitPython Powered Internet Display](https://www.adafruit.com/product/4116)

PyPortal, our easy-to-use IoT device that allows you to create all the things for the “Internet of Things” in minutes. Make custom touch screen interface...

<https://www.adafruit.com/product/4116>

Connecting to the Reader

Orientation of Keys

From user guide:

When properly oriented, the braille cells are closest to you. The Panning keys are at each end of the braille display.

For orientation purposes, there are three slightly raised tick marks located above the braille cells. These orientation marks are spaced by every fifth braille cell. For example, the first tick mark from the left is between the fifth and sixth braille cell.

Moving towards the top and away from you, find a row of three keys, with a wider key- in the middle. The wide key is the Space bar. The Dot 7 input key is to the left of the Space Bar, and the Dot 8 input key is to the right of the Space Bar.

Above the three keys is a navigation pad in the middle, between the Braille Input keys and Space bar. The navigation pad contains four directional arrow buttons (Up, Down, Left, Right) and the Select button.

The six traditional braille input keys are aligned horizontally along the top edge of the face of the display, Dots 3 2 1 on the left and Dots 4 5 6 on the right.



Operating System

The operating system you are using will determine how you can connect to the reader. We have instructions on how to set up with a Mac or PC.

Note there are no instructions on how to connect to Linux at the moment.

Additional Resources

On [this resources page \(https://adafru.it/FDI\)](https://adafru.it/FDI), from the Orbit Research website, download the user guide for your file format of your choice.

Feel free to use it as a reference guide.

Connecting on Mac

Switching to HID mode

In order for the Orbit Reader to be recognized and used by the computer, it needs to be put in HID or "Human Interface Device" mode. Once this mode is enabled, the reader can be used as a text input device, text reading device and more.

Connect and power on the reader

- Connect reader to Mac via a micro usb cable
- Power on the reader by holding the button on the back (next to the SD card slot)



Enable HID mode for reader

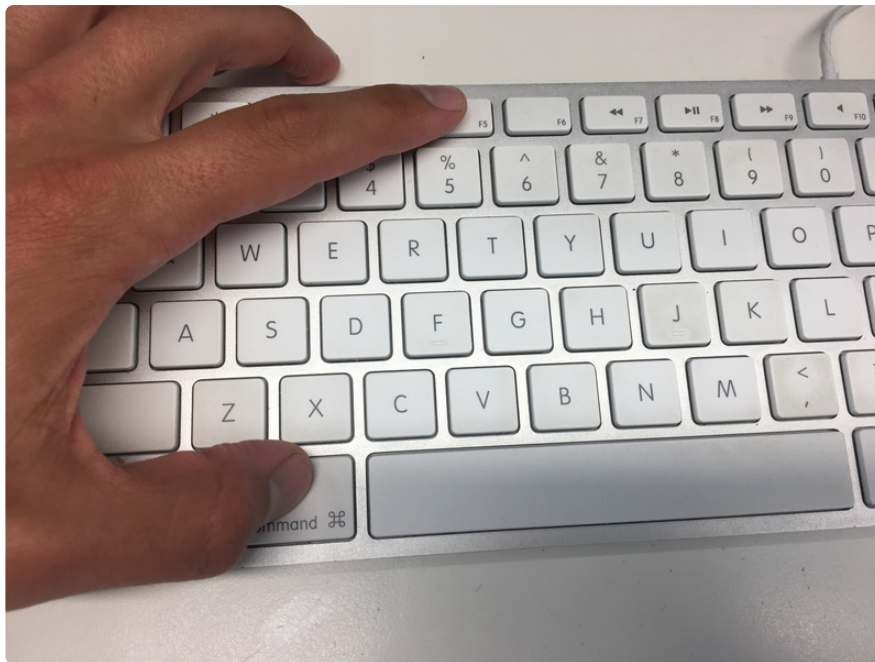
- Press buttons 2, 7, and space bar on the reader all at once to enable HID mode
- Use the diagram and GIF image below to see which buttons are which on the reader



Turn on "VoiceOver" application

Most Macs come preinstalled with an application called VoiceOver. This program is what we'll use to connect to the Orbit Reader.

- Tap command and F5 keys at the same time to enable VoiceOver
- The below pop-up will appear on the screen prompting you to enable the application



Click "Use VoiceOver" to enable the application



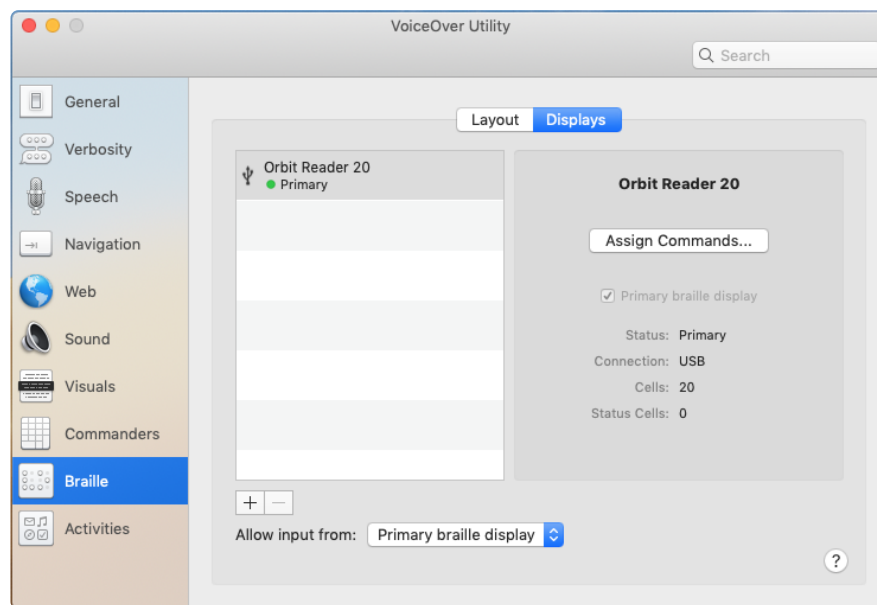
That's it! The device should be printing out braille according to what's on your screen or what you have selected. For example, I have a **CIRCUITPY** device connected (an Adafruit board running CircuitPython and presenting as a flash drive named **CIRCUITPY**), so I clicked on it and its name was printed to the screen reader through

voice over. Try clicking around and see how the reader reacts. VoiceOver will tell you what's being printed to the reader on the screen.

Having trouble connecting?

Try opening up the application "VoiceOver Utility" on your computer.

Click the "Braille" tab, then click "Displays" to see if the computer recognizes the reader.

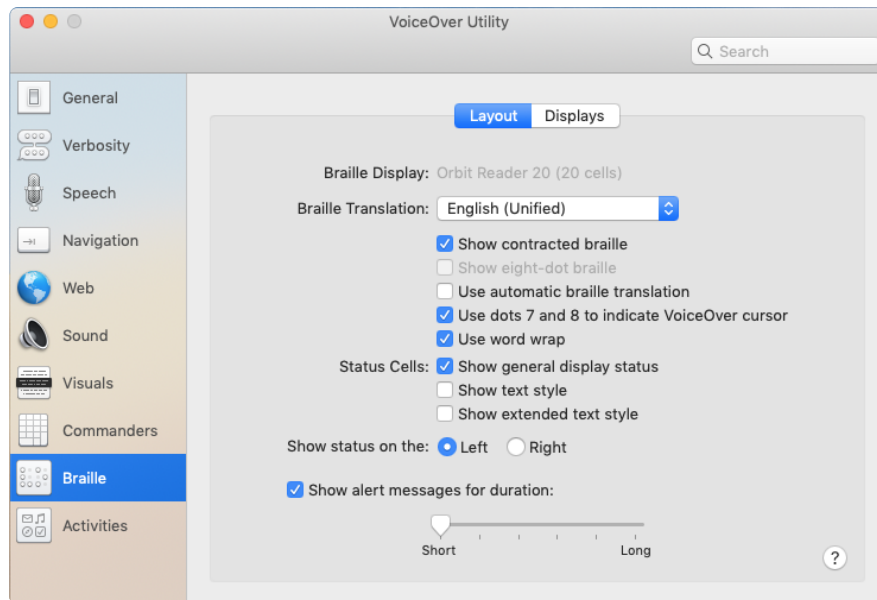


If the reader isn't showing up try:

- enabling HID mode again
- turning VoiceOver off then back on

Still not working?

- check to see if the "Layout" settings are the same as below



Additional Resources

On [this resources page \(https://adafru.it/FDI\)](https://adafru.it/FDI), from the Orbit Research website, download the user guide from your file format of your choice.

Feel free to use it as a reference guide.

Connecting on PC

The Orbit Reader must talk to "screen reader" software on your PC for it to interact properly with the computer and be able to output braille.

If you do not already have one of the below supported screen reader applications on your PC, click one of the links to download:

- [Non Visual Desktop Access \(NVDA\) \(https://adafru.it/FDm\)](https://adafru.it/FDm)
- [JAWS \(https://adafru.it/FDn\)](https://adafru.it/FDn)
- [Dolphin Screen Reader \(https://adafru.it/FDo\)](https://adafru.it/FDo)

Which one to pick? All are great options (but the first two are free).

If you use a different screen reader software than those listed above, check out the section at the bottom of this page entitled "Have a screen reader that does not support HID?".

Download necessary drivers

For the software to be able to interact with the Orbit Reader, download the necessary drivers from below.

[Click here to access driver download page \(https://adafru.it/FDp\)](https://adafru.it/FDp)

Under the "Software packages and drivers" section, pick the driver that applies to you.

Switching to HID mode

In order for the Orbit Reader to be recognized and used by the computer, it needs to be put in HID or "Human Interface Device" mode. Once this mode is enabled, the reader can be used as a text input device, text reading device and more.

Connect and power on the reader

- Connect the Orbit reader to your PC via the micro USB cable
- Power on reader by holding the button on the back (next to the SD card slot)



Enable HID mode for reader

- press buttons 2, 7, and space bar on the reader all at once to enable HID mode.
- Use diagram and gif below to see which buttons are which on the reader.



Connecting to the Orbit Reader

The following sections show to how to connect your PC to the Orbit Reader depending on the screen reader software you have installed. These sections have been taken from the user guide linked in the "Connecting to the Reader" page of this guide.

NVDA

If you have NVDA (version 2017.1 or later) installed on your PC, it automatically recognizes the Orbit Reader 20. If NVDA is not recognizing the display, go to the NVDA Preferences menu and select Braille Settings from the list. From the braille display drop-down menu, select "Baum/ Humanware/APH/Orbit Displays" and click OK. Note: NVDA works in HID only, not Serial. If Orbit Reader 20 is not one of the displays shown in the Braille Display list, upgrade NVDA to the newest version and repeat the process. NVDA turns 'On' braille output when it is configured for the display. To turn 'Off' braille support, select "No Braille" from Braille Display options in the Braille Settings menu.

JAWS

Currently, the Orbit Reader 20 is supported for JAWS versions 17, 18, and 2018 through a driver. The JAWS driver download and instructions are available on the Orbit Research Support webpage. For JAWS versions older than 17, you must use the RB18 emulation mode on the Orbit Reader. When connecting Orbit Reader 20 to JAWS by USB, it must be set to use the HID protocol by pressing Space + Dots 2 7. To connect Orbit Reader 20 by USB, follow these steps:

1. Start or restart JAWS.
2. Insert + J to bring up JAWS menu.
3. Press Enter on Options.
4. Down Arrow to Braille and press Enter.
5. Tab to Add and press Enter.
6. Arrow Up or Down to Aph Refreshabraille 18 and press Space to check the box and select it.
7. Tab to the Next button and press Enter.
8. Select USB.
9. Tab to the Next button and press Enter.
10. Select Aph Refreshabraille 18 as primary device.
11. Tab to the Finish button and press Enter.
12. Restart JAWS.

Dolphin Screen Reader

To connect Orbit Reader 20 to Dolphin Screen Reader, the device must be connected by a Standard-A to Micro-B USB cable and set in HID protocol mode.

Hit Buttons 2 + 7 + space, as shown, above, to switch to HID mode

Have a screen reader that does not support HID?

If your screen reader does not support HID, switch to the Serial protocol on the Orbit Reader 20 with the hotkey or selecting Serial from the USB option in the menu.

For Windows versions 7 and newer, the serial interface requires the installation of two drivers: one for the USB and one to make the USB port appear like a COM port. The only exception is Windows XP.

Because it is Serial only, it can just be plugged in. The second driver is required because many screens reading programs handle braille displays like a serial device. The COM port assignment driver shows the assigned port number used to communicate with the display.

Take a note of the COM port number for later use. If you need to look at it later, go to Device Manager while the display is connected and look in the section for Ports: COM and LPT.

One of the COM ports is assigned to Orbit Reader 20. You need to use that number when you set up your screen reader. Once the display is connected to the device you wish to use, configure the software to use Orbit Reader 20.

Outputting the REPL to the Reader

Now that the reader is properly connected to your computer, it's time to get the CircuitPython REPL up and running.

Please note: the Mu editor's REPL doesn't yet work with the screen reader so we will have to use the advanced serial console.

Follow the directions in the following pages to get your REPL up and running.

The following program is the example code for this project but you can use any program you'd like!

```
import time
import random
```

```

helloWorldList = ["hi earth!",
                  "hello world",
                  "howdy, partner.",
                  "hey there",
                  "ahoy maties!",
                  "hola terra!",
                  "sup.",
                  "hi ya",
                  "what's happenning???",
                  "hi hello hey there whats good?",
                  "twenty char phrasee"
                  ]

while True:
    helloNum = random.randint(0, len(helloWorldList)-1)
    print(helloWorldList[helloNum])
    time.sleep(2)

```

To use, copy the code into a text editor or Mu. Save as **code.py**. Copy the **code.py** file to the **CIRCUITPY** drive that appears when the board is connected to your computer with a known, good USB data plus power cable. The code will run once the file transfer completes.

Advanced Serial Console on Mac

Connecting to the serial console on Mac does not require installing any drivers or extra software. You'll use a terminal program to find your board, and **screen** to connect to it. Terminal and **screen** both come installed by default.

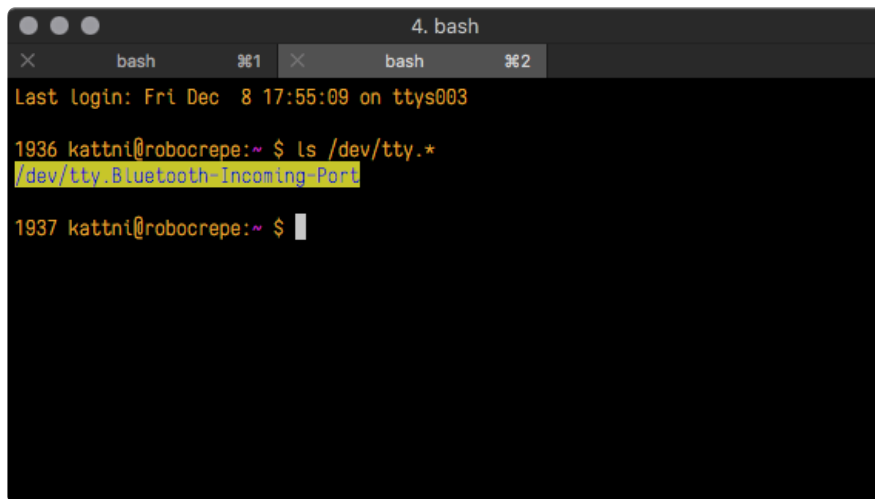
What's the Port?

First you'll want to find out which serial port your board is using. When you plug your board in to USB on your computer, it connects to a serial port. The port is like a door through which your board can communicate with your computer using USB.

The easiest way to determine which port the board is using is to first check **without** the board plugged in. Open Terminal and type the following:

```
ls /dev/tty.*
```

Each serial connection shows up in the **/dev/** directory. It has a name that starts with **tty.**. The command **ls** shows you a list of items in a directory. You can use ***** as a wildcard, to search for files that start with the same letters but end in something different. In this case, you're asking to see all of the listings in **/dev/** that start with **tty.** and end in anything. This will show us the current serial connections.

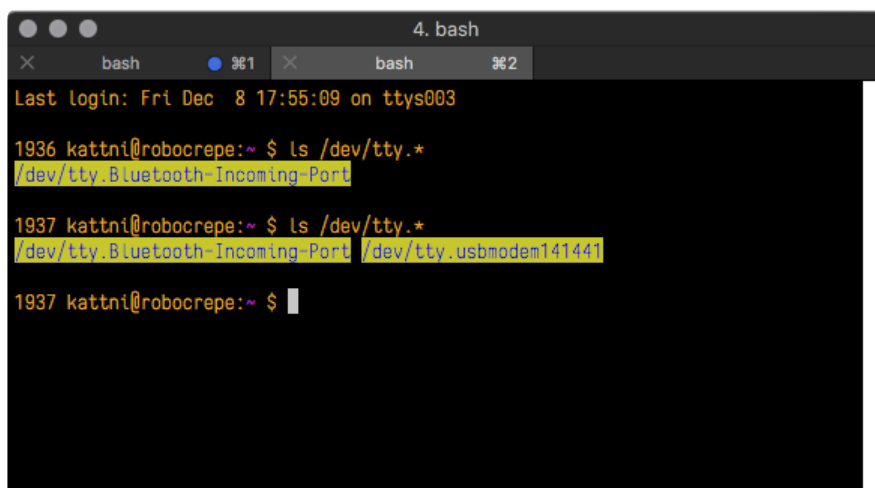
A screenshot of a macOS Terminal window titled "4. bash". It shows two tabs: "bash" with icon %1 and "bash" with icon %2. The text "Last login: Fri Dec 8 17:55:09 on ttys003" is at the top. The prompt "1936 kattni@robocrepe:~" is followed by the command "\$ ls /dev/tty.*". The output is "/dev/tty.Bluetooth-Incoming-Port". The next prompt is "1937 kattni@robocrepe:~" followed by a dollar sign "\$".

```
4. bash
bash %1 bash %2
Last login: Fri Dec 8 17:55:09 on ttys003
1936 kattni@robocrepe:~ $ ls /dev/tty.*
/dev/tty.Bluetooth-Incoming-Port
1937 kattni@robocrepe:~ $
```

Now, plug your board. In Terminal, type:

```
ls /dev/tty.*
```

This will show you the current serial connections, which will now include your board.

A screenshot of a macOS Terminal window titled "4. bash". It shows two tabs: "bash" with icon %1 and "bash" with icon %2. The text "Last login: Fri Dec 8 17:55:09 on ttys003" is at the top. The prompt "1936 kattni@robocrepe:~" is followed by the command "\$ ls /dev/tty.*". The output is "/dev/tty.Bluetooth-Incoming-Port". The next prompt is "1937 kattni@robocrepe:~" followed by the command "\$ ls /dev/tty.*". The output is "/dev/tty.Bluetooth-Incoming-Port /dev/tty.usbmodem141441". The next prompt is "1937 kattni@robocrepe:~" followed by a dollar sign "\$".

```
4. bash
bash %1 bash %2
Last login: Fri Dec 8 17:55:09 on ttys003
1936 kattni@robocrepe:~ $ ls /dev/tty.*
/dev/tty.Bluetooth-Incoming-Port
1937 kattni@robocrepe:~ $ ls /dev/tty.*
/dev/tty.Bluetooth-Incoming-Port /dev/tty.usbmodem141441
1937 kattni@robocrepe:~ $
```

A new listing has appeared called `/dev/tty.usbmodem141441`.

The `tty.usbmodem141441` part of this listing is the name the example board is using. Yours will be called something similar.

Using Linux, a new listing has appeared called `/dev/ttyACM0`. The `ttyACM0` part of this listing is the name the example board is using. Yours will be called something similar.

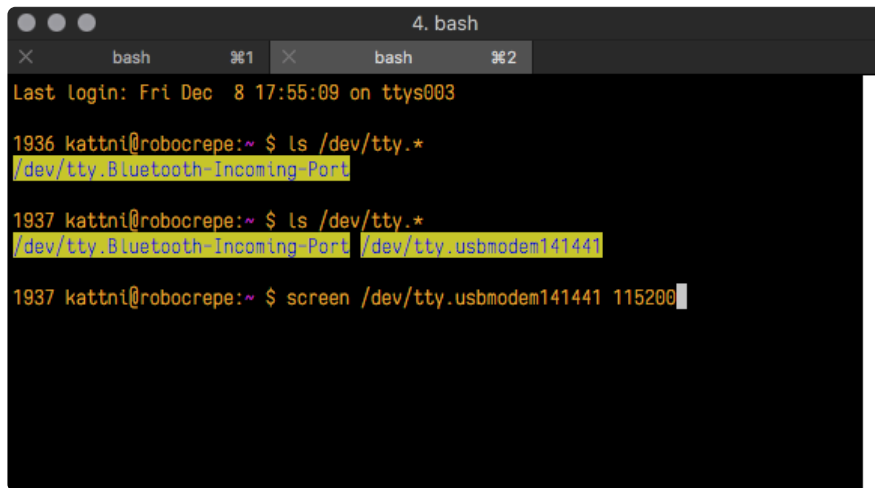
Connect with screen

Now that you know the name your board is using, you're ready connect to the serial console. You're going to use a command called `screen`. The `screen` command is included with MacOS. To connect to the serial console, use Terminal. Type the

following command, replacing `board_name` with the name you found your board is using:

```
screen /dev/tty.board_name 115200
```

The first part of this establishes using the `screen` command. The second part tells screen the name of the board you're trying to use. The third part tells screen what baud rate to use for the serial connection. The baud rate is the speed in bits per second that data is sent over the serial connection. In this case, the speed required by the board is 115200 bits per second.



```
4. bash
bash %1 bash %2
Last login: Fri Dec 8 17:55:09 on ttys003
1936 kattni@robocrepe:~ $ ls /dev/tty.*
/dev/tty.Bluetooth-Incoming-Port
1937 kattni@robocrepe:~ $ ls /dev/tty.*
/dev/tty.Bluetooth-Incoming-Port /dev/tty.usbmodem141441
1937 kattni@robocrepe:~ $ screen /dev/tty.usbmodem141441 115200
```

Press enter to run the command. It will open in the same window. If no code is running, the window will be blank. Otherwise, you'll see the output of your code.

Great job! You've connected to the serial console!

Advanced Serial Console on Windows

Windows 7 and 8.1

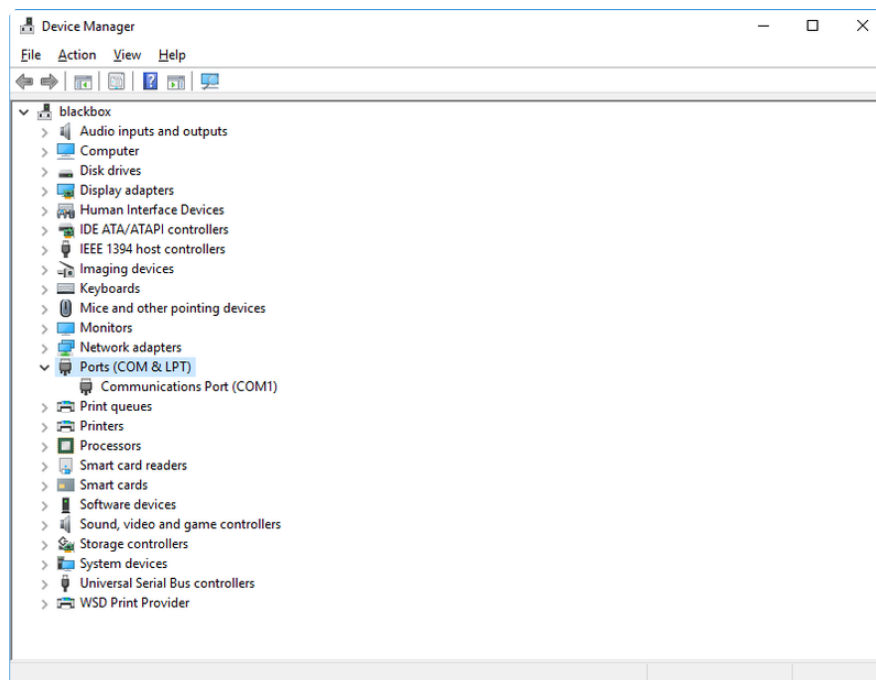
If you're using Windows 7 (or 8 or 8.1), you'll need to install drivers. See the [Windows 7 and 8.1 Drivers page \(https://adafru.it/VuB\)](https://adafru.it/VuB) for details. You will not need to install drivers on Mac, Linux or Windows 10.

You are strongly encouraged to upgrade to Windows 10 if you are still using Windows 7 or Windows 8 or 8.1. Windows 7 has reached end-of-life and no longer receives security updates. A free upgrade to Windows 10 is [still available \(https://adafru.it/RWc\)](https://adafru.it/RWc).

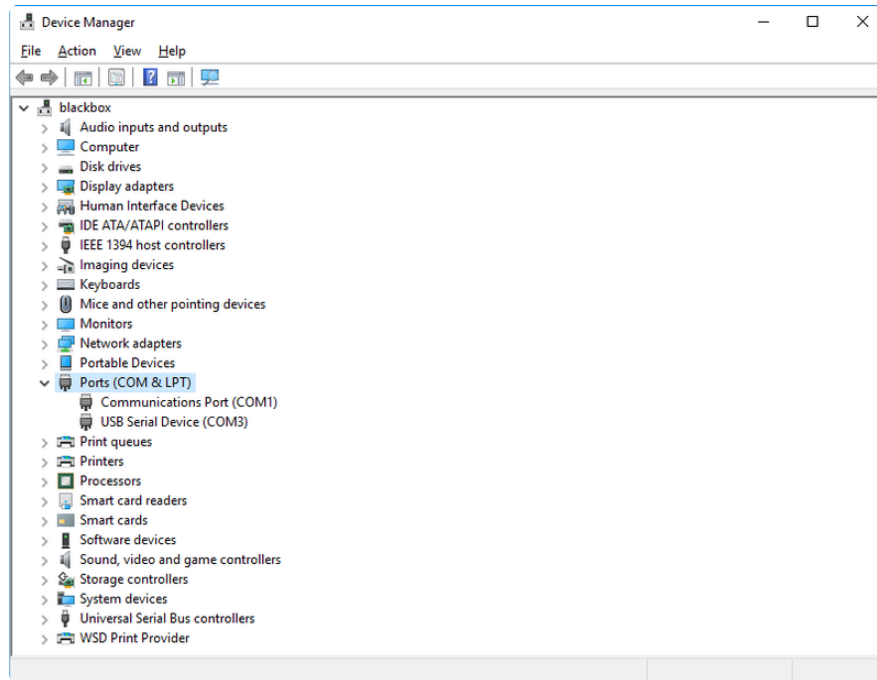
What's the COM?

First, you'll want to find out which serial port your board is using. When you plug your board in to USB on your computer, it connects to a serial port. The port is like a door through which your board can communicate with your computer using USB.

You'll use Windows Device Manager to determine which port the board is using. The easiest way to determine which port the board is using is to first check **without** the board plugged in. Open Device Manager. Click on Ports (COM & LPT). You should find something already in that list with (COM#) after it where # is a number.



Now plug in your board. The Device Manager list will refresh and a new item will appear under Ports (COM & LPT). You'll find a different (COM#) after this item in the list.



Sometimes the item will refer to the name of the board. Other times it may be called something like USB Serial Device, as seen in the image above. Either way, there is a new (COM#) following the name. This is the port your board is using.

Install Putty

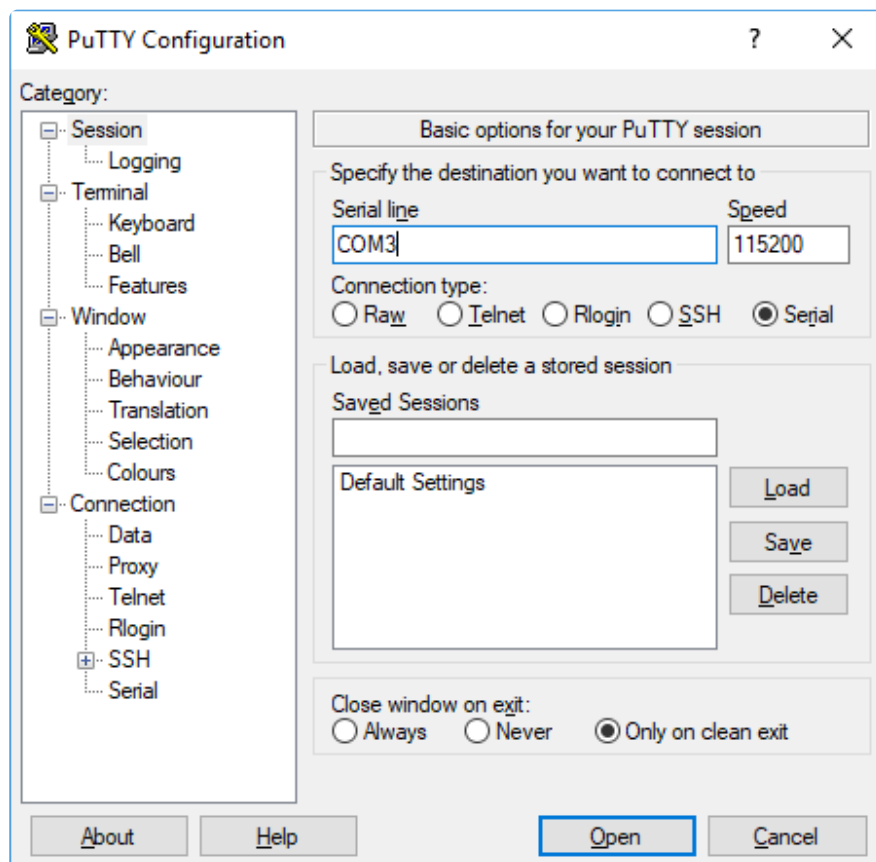
If you're using Windows, you'll need to download a terminal program. You're going to use PuTTY.

The first thing to do is download the [latest version of PuTTY \(https://adafru.it/Bf1\)](https://adafru.it/Bf1). You'll want to download the Windows installer file. It is most likely that you'll need the 64-bit version. Download the file and install the program on your machine. If you run into issues, you can try downloading the 32-bit version instead. However, the 64-bit version will work on most PCs.

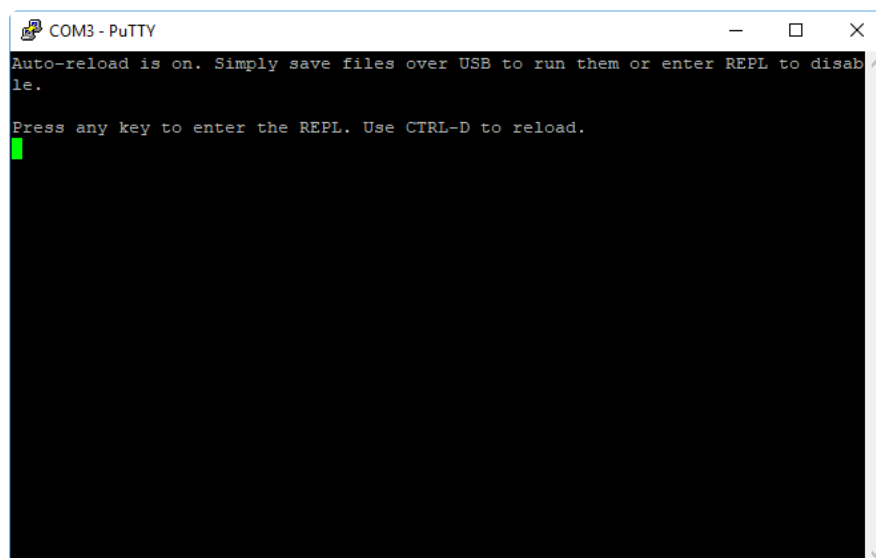
Now you need to open PuTTY.

- Under **Connection type**: choose the button next to **Serial**.
- In the box under **Serial line**, enter the serial port you found that your board is using.
- In the box under **Speed**, enter 115200. This called the baud rate, which is the speed in bits per second that data is sent over the serial connection. For boards with built in USB it doesn't matter so much but for ESP8266 and other board with a separate chip, the speed required by the board is 115200 bits per second. So you might as well just use 115200!

If you want to save those settings for later, use the options under **Load, save or delete a stored session**. Enter a name in the box under **Saved Sessions**, and click the **Save** button on the right.



Once your settings are entered, you're ready to connect to the serial console. Click "Open" at the bottom of the window. A new window will open.



If no code is running, the window will either be blank or will look like the window above. Now you're ready to see the results of your code.

Great job! You've connected to the serial console!