



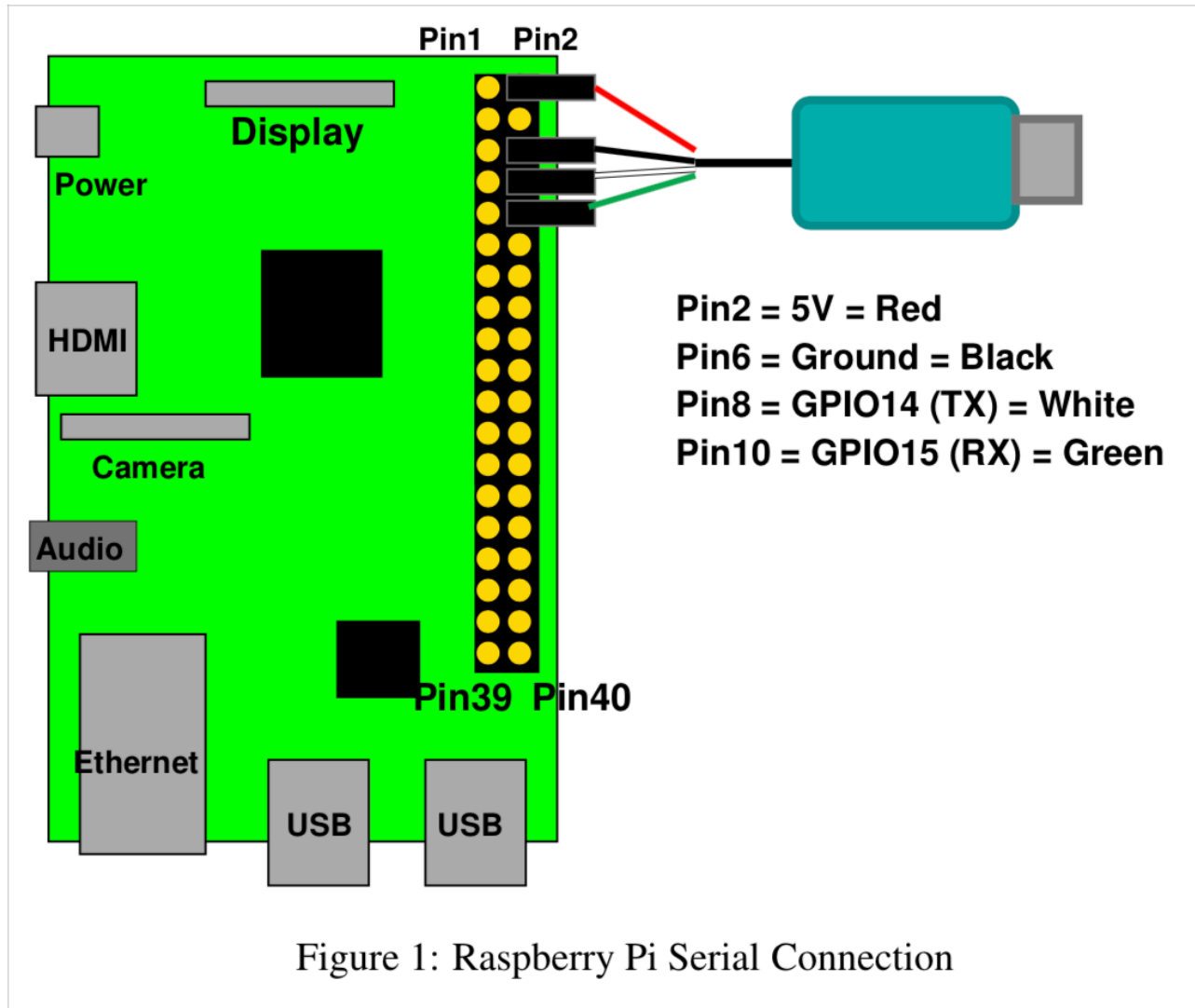
# Project I: Console Cable (4%)

ENEE 447: Operating Systems — Spring 2019

Assigned: Tuesday, Feb 5; Due: Sunday, Feb 17

## Purpose

This project shows you how to use a USB-to-serial cable to interact directly with the Raspberry Pi from your laptop. This emulates the behavior of the RPi as if it were connected to a keyboard and simple character-based terminal. This is *extremely* powerful and will be the primary way you interact with the board from now on. Like the previous project, this project is less about code development and more about giving you tools that you will use from now on.



## Get a Console Cable

The figure above shows what we’re talking about here, as well as how it is connected to the board. The “console cable” is a hybrid cable with a USB plug on one end, which goes into your laptop, and at the

other end it has 4 female pin-header connectors. The UART output (a serial port) of the Raspberry Pi uses pins 6, 8, and 10 of the GPIO header block (for *ground*, *transmit*, and *receive*, respectively), as well as pin 2 for power.

You can buy the cable at the following site:

<https://www.adafruit.com/product/954>

The site also provides a very good tutorial on how to set it up and use it:

<https://learn.adafruit.com/adafruits-raspberry-pi-lesson-5-using-a-console-cable>

We have also provided links to the website and a PDF tutorial on the class website.

Note that, even though you *can* use pin 2 to power the board, it is less than ideal for debugging, because whenever you try to toggle power by disconnecting the pin header, the serial connection gets disconnected, which can be a pain, and you might not be able to tell the difference between your code failing to boot, your code booting but not printing to the screen (i.e., failing to transmit characters over the wire), or all of it working fine, but it getting lost because it happened quickly while you were trying to re-establish a TTY connection with the RPi board.

## Hook It Up

You will need to load a driver for the cable (see the website), and once it is loaded, and the cable plugged into your laptop, you will see a special device appear, due to the cable having an ASIC in it that identifies to the laptop its special-device nature.

First, just verify that the thing works in the first place. Boot into Linux on your RPi board, but first modify the `config.txt` file to have the following statement at the bottom of the file:

```
enable_uart=1
```

Like the other statements in that file, it should be on its own line.

Follow the directions to establish a terminal connection with the USB/serial device on your laptop (e.g., on macOS, in the Terminal app, you type “`screen /dev/cu.SLAB_USBtoUART 115200`”), and when you boot the RPi board into Linux, your terminal should show an interactive Linux `login:` prompt.

When that is working, then you know you have the right drivers, your cable works, it is connected properly, and the board can both receive and transmit just fine. At that point, you can move to the next step and run the code in the Project 1 directory, which is a simple echo server. All it will do is wait for your first input and then begin echoing whatever you type, but translated to upper-case (just so that you know your text is being received, processed, and then echoed by your software). You should see this:

```
<NAME>'s system is booting.
Please hit enter to continue.
```

At this point, the software is waiting for you to type something. Once you do, it will output some useful data-size information and then print the following:

```
Running ECHO server.
```

And, from this point on, everything you type on the keyboard will be reflected back to you, except that it will be in upper-case letters.

All you need to do is change `<NAME>` to your own name.

## Customize It, Build It, Load It, Run It

Once you have it working, show us.