Raspberry Pi – FM Radio Transmitter

Jonathan Franck

Objective: In this project, I will demonstrate how to use your raspberry pi to broadcast a radio signal over an FM frequency.

# Step 1: Hardware

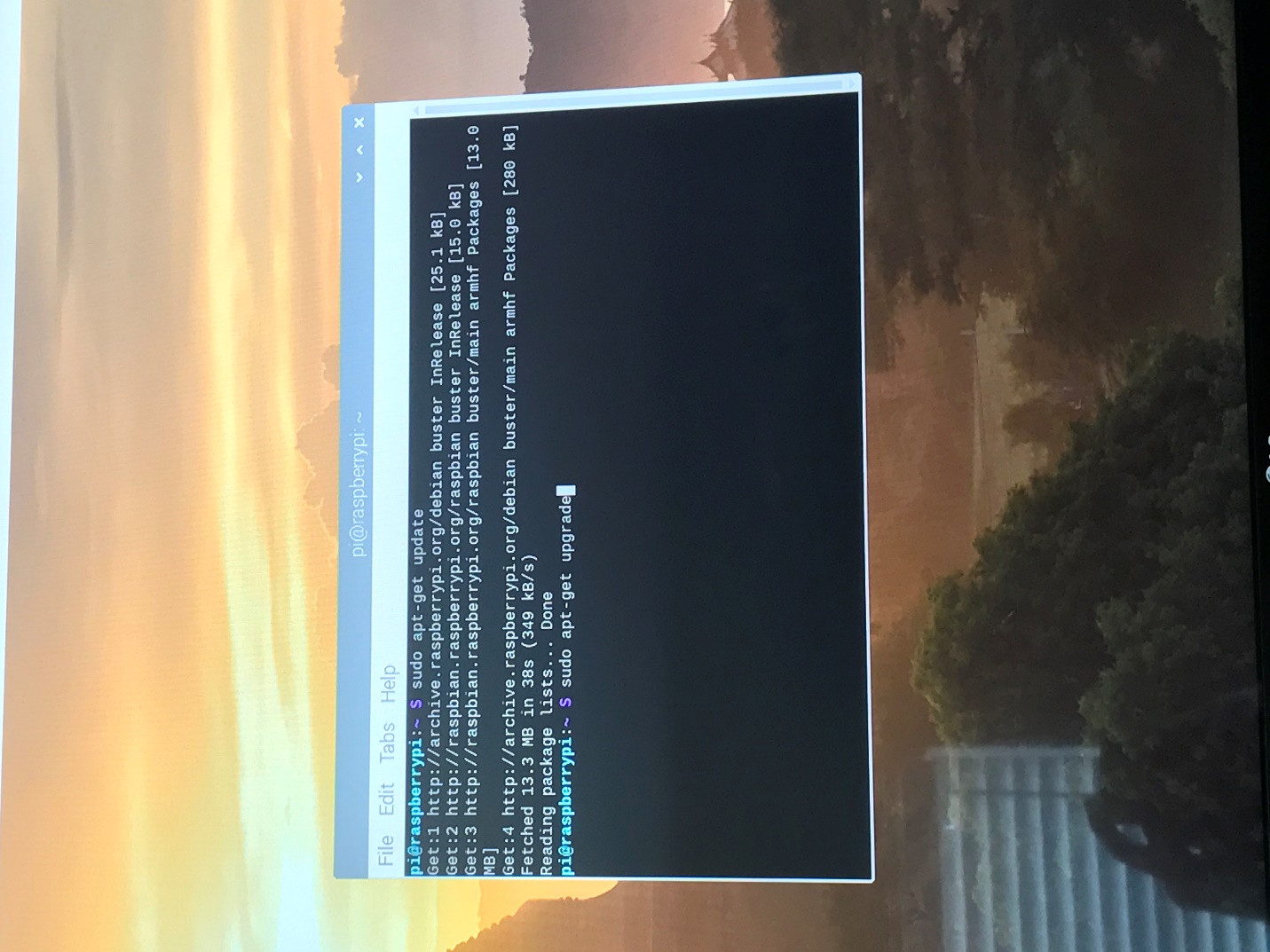
The only additional hardware that is necessary for this project is some kind of to act as an antenna, I just ordered a pack of jumper wires from amazon. Just plug the female end into pin 7, it is the fourth pin from the top of the Pi on the inside row:



\*\*\*On my Pi, I have the wires to my fan (black and red) plugged into pins 1 and 6, and the white wire plugged into pin 7 will act as the antenna\*\*\*

# Step 2: Update your Pi

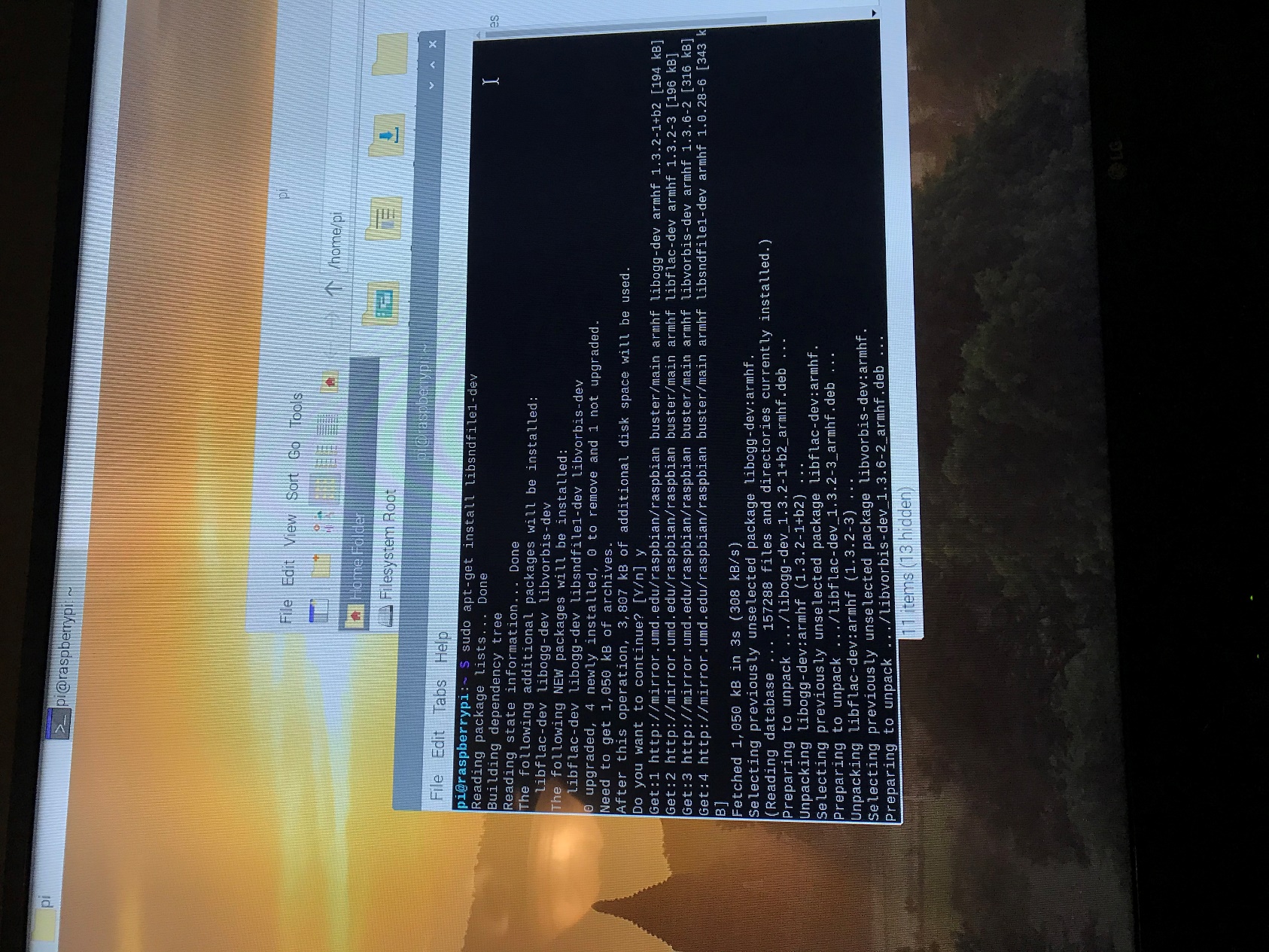
As with any other project, start by using the commands “sudo apt-get update” and “sudo apt-get upgrade” in the terminal to make sure all of your software is up to date.



# Step 3: Download and install the necessary software

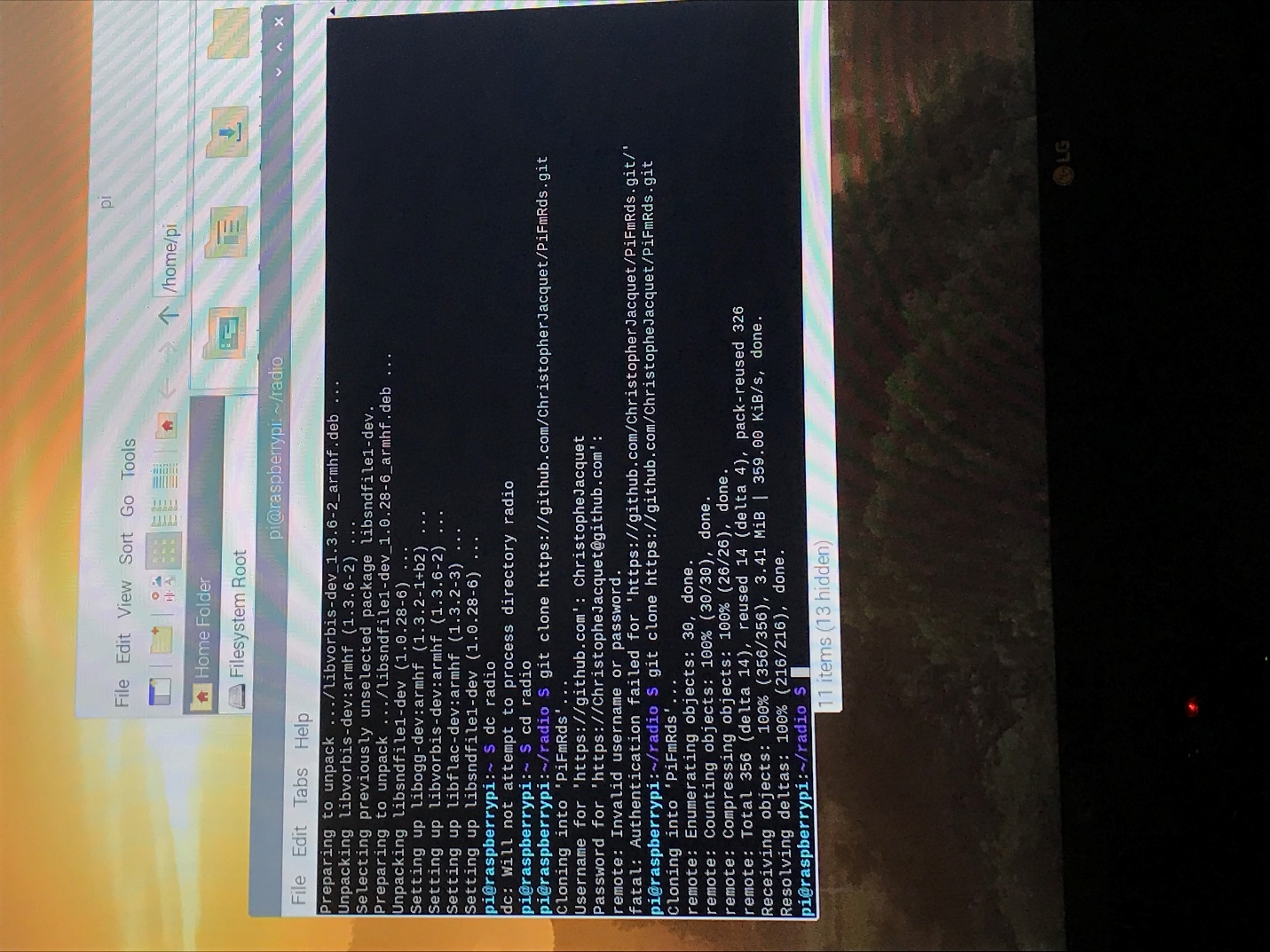
The first thing you need to install is libsndfile which is a C library used to read and write audio files.

Use the following command: “sudo apt-get install libsndfile1-dev”



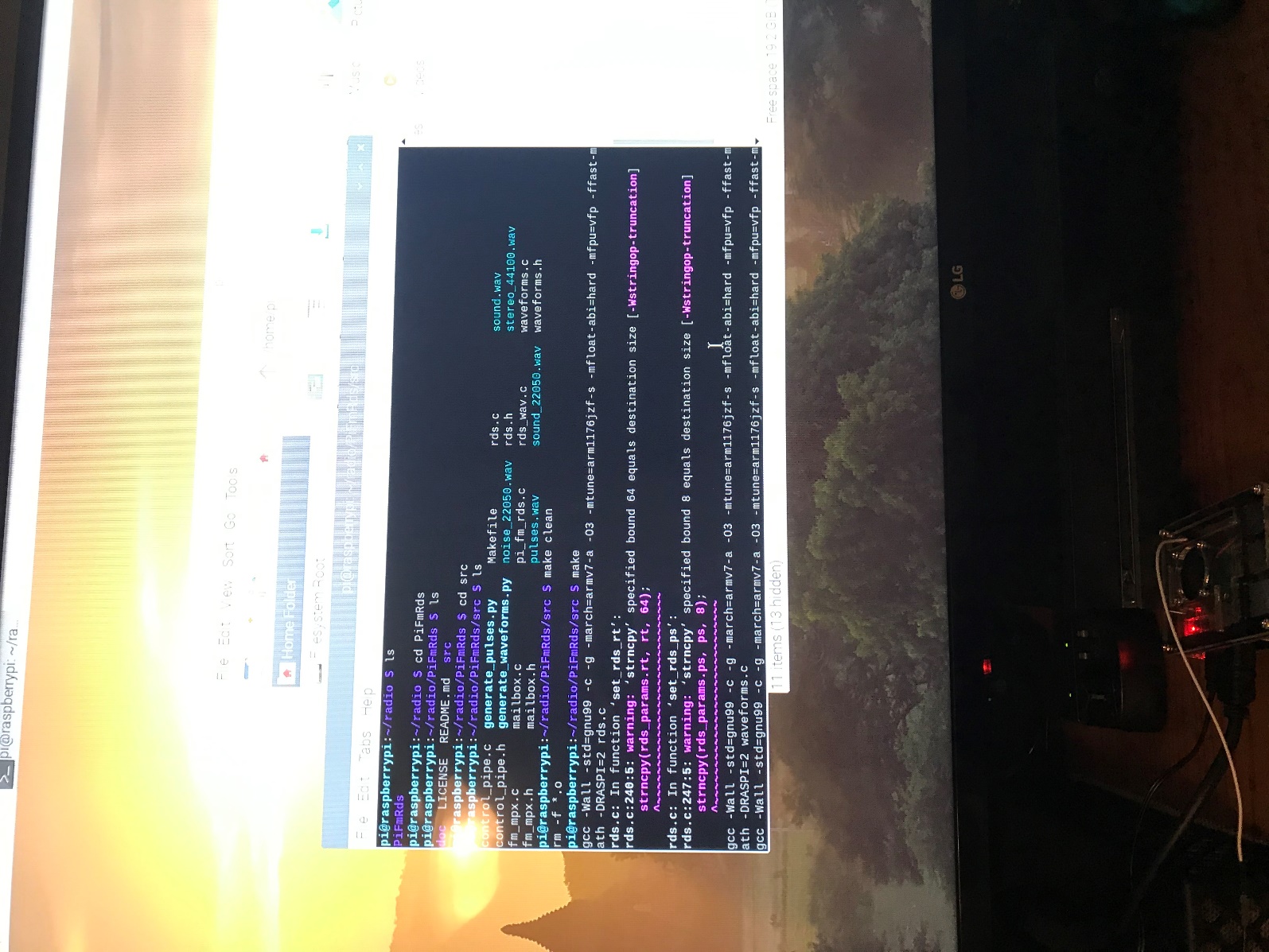
Next, you have to clone PiFmRds from GitHub to a file you will remember(I did this inside of a desktop folder named radio). This contains the executable file that actually broadcasts the signal, as well as some sample audio files. Type the following into the terminal:

“git clone <https://github.com/ChristopheJaqcuet/PiFmRds.git>”



Once it’s done cloning you must clean and make the executable with the commands:

“make clean” and “make”



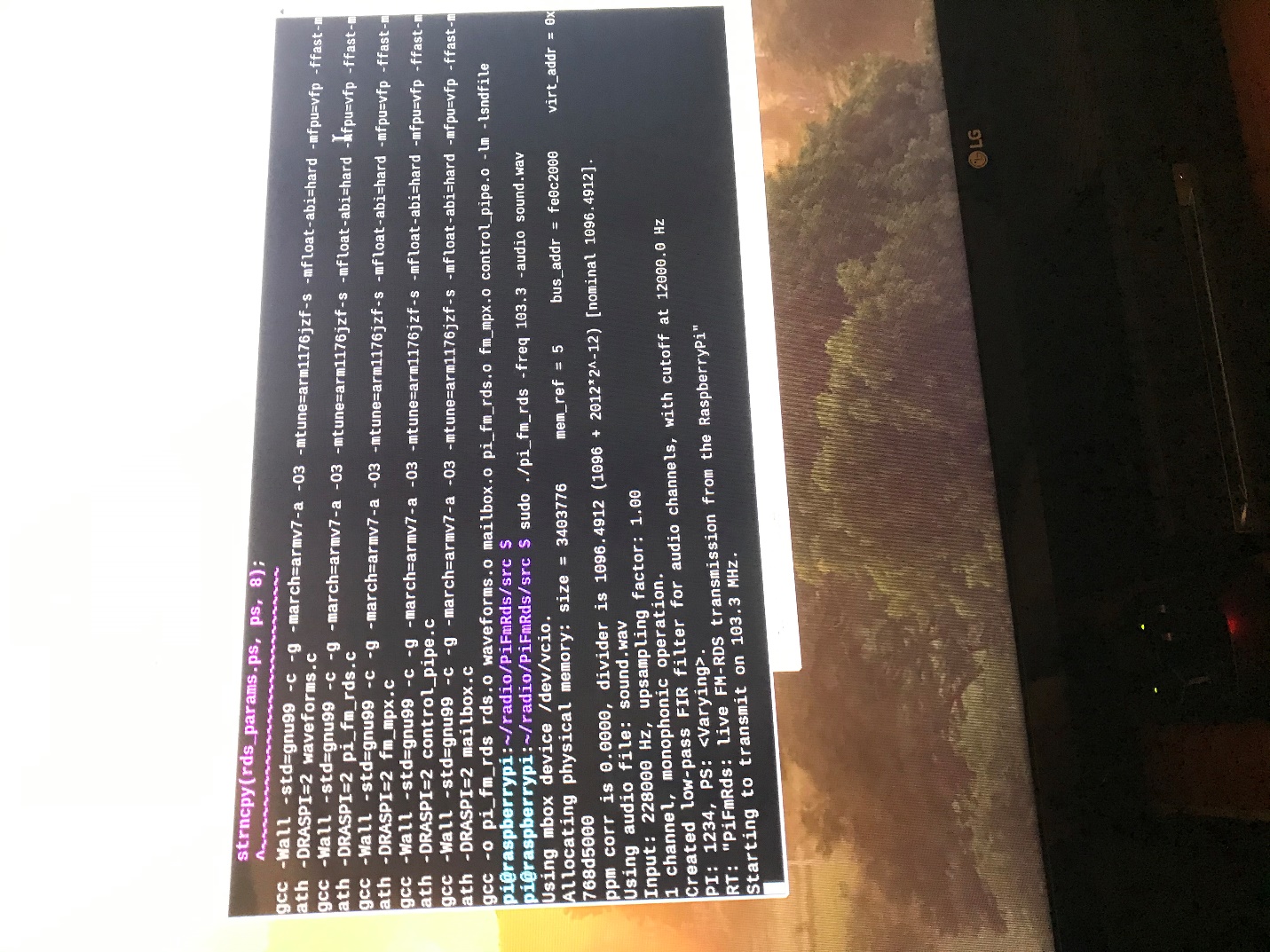
At this point you are pretty much ready to broadcast…

# Step 4: Transmit and listen to the audio

Now your Pi is all set to transmit an FM signal and a device to tune into the frequency. In order to run the executable just type: “sudo ./pi\_fm\_rds”. If you want to specify which audio file and which frequency to transmit, you must use additional parameters “-freq” and “-audio”. I played the sample “sound.wav” file on frequency 103.3 with the following command:

“sudo ./pi\_fm\_rds -freq 103.3 -audio sound.wav”

In order to stop the transmission just press “CTRL + Z”.



I used a radio alarm clock to tune into the 103.3 FM frequency

\*\*\*I have included a separate video of the alarm clock playing the audio from the radio \*\*\*

# Challenges/Difficulties:

The main issue with this project was getting the jumper wires. I couldn’t find any local retailers that sold them, so I ordered them on amazon. However, with the whole COVID-19 situation, they got stuck at the mailroom on campus for about two weeks before they were finally delivered to my house a few days ago.

While I was waiting for the wires to be delivered, I did a bit of research for this project and was confident I would be able to complete it quickly once I got them. I followed my first source exactly as it instructed, but it did not work. I then turned to another source I had already found, and it didn’t work either, saying it couldn’t find the executable. I thought it might be due to my radio and I tried to download an FM radio app on my iPhone, but to no avail. I found a new source which advised installing libsndfile, and also specified the steps to clean and make the file cloned from GitHub, and it worked first try.