

# AirController User Manual

Steven Chu, Maksym Prokopovych, Sierra Rose, Isaac Xu  
ECE 180DB, Winter 2022

# Getting Started

## *Hardware Requirements*

The AirController has the following hardware requirements:

1. Two Raspberry Pi Zeros
2. BerryIMU
  - a. Jumper cables
3. IR emitter and receiver
  - a. Wires and breadboard to connect these together with one of the Raspberry Pis
4. A computer with a webcam and microphone
5. Wifi signal

## *Hardware Set-Up*

While in an ideal scenario, the user would not be doing any of this set-up as the entire product would come preconfigured, in order to install the AirController onto pre-existing devices, the following set-up must be carried out.

For the first Raspberry Pi, we must set up the BerryIMU with 4 jumper cables, as well as enable I2C communication. For detailed instructions on how to do this, please refer to Lab 4 from fall quarter.

For the second Raspberry Pi, we must set up the IR emitter. The emitter is a standalone 3 pin device, it requires 3 pins, 3.3 V power, ground, and signal pins. The default signal pin is GPIO22, the 8th pin down on the left row of the header, with the “top” being the SD card slot.

If the user so desires, we have also provided CAD files to 3D print a case for the BerryIMU + Raspberry Pi + a suitable battery. While not necessary, the case helps to make the BerryIMU Raspberry Pi more robust, and contributes to a wireless solution for it.

## *Downloading and Installation*

Connect to wifi on both of the Raspberry Pis, as well as the webcam and microphone-enabled computer. From there, download the Team 1 Github repo onto all of the devices. If trouble is experienced with this step, please skip to the disk imaging section below for further instructions on how to connect to wifi.

## *Computer Set-up*

On the webcam and microphone-enabled computer, you must install the necessary software using the installation scripts. If you do not already have Python and pip installed, then please follow these steps:

Download an Anaconda environment (Python)

First run `conda install pip` -> this will install the pip directory

If it's already installed, update pip, or skip this step entirely

If the user is struggling to run the given scripts (such as in the case of having a Windows computer), please run each given command in the scripts individually by first installing the necessary software, including Python and pip. This will enable the user to run most commands in the script. The user can skip any commands which do not run aside from these necessary ones using pip, such as apt-get. Please follow the commands listed within `total_install_script.sh`.

If the user already has Python and pip installed, then please run the `total_install_script.sh` file to install the AirController.

### *Raspberry Pi Set-up*

On the first Raspberry Pi which is connected to the BerryIMU, run the `IMUpi_install_script.sh` present within the gesture folder. This will install the necessary dependencies. If desired, follow the instructions within `system_config_information.txt` in the gesture folder for how to make the program run on boot. Alternatively, you may flash the disk image provided. Further instructions on how to do that can be found below.

On the second Raspberry Pi which is connected to the IR emitter, you can either follow the detailed instructions within the `lirc_configs` folder's README, or you can install the disk image provided (preferable). This is because the disk image has everything configured already for IR emission on pin 22 and automatic startup on boot.

### *Disk Imaging (optional)*

Disk imaging can be accomplished with the .iso file by taking your favorite disk image utility (Rufus, dd, etc) and inputting the .iso file, and flashing an SD card with it.

Disk images for the IMU Pi & IR Pi can both be found at:

[https://drive.google.com/drive/folders/1LqcZ9hz0c\\_c1fFU0opW7Ngrds5oL3GBd?usp=sharing](https://drive.google.com/drive/folders/1LqcZ9hz0c_c1fFU0opW7Ngrds5oL3GBd?usp=sharing) .

The only change required once imaging on each Pi is changing the `wpa_supplicant.conf` file (in `/etc/wpa_supplicant/wpa_supplicant.conf`) to support your local wifi network. Additionally, on the IR Pi, you must select the type of TV used in `/home/pi/AirController/config.py`. These can be edited either on the SD card itself or by sshing into the Pi (uses the default pi username and password).

## How to Use

The AirController has 3 ways to control a TV. The first is via whole-body poses recognized by the webcam. The second is through gestures using the BerryIMU-enabled Raspberry Pi. The third is via voice commands as heard by the computer. In order to best use each of these components, please continue reading to determine how to control each section.

### *Webcam-Detected Poses*

These are the current postures that would trigger a command. To use, the person must be in camera view such that the upper body is within view, and the person must properly be detected. There must be sufficient lighting so that the camera can see one, and darker color clothing may need to be removed.

Command	Gesture(s)	Requirements
Volume Up	Right Dab	<ul style="list-style-type: none"><li>- Head into the right elbow<ul style="list-style-type: none"><li>- Distance between head and elbow doesn't have to be too extreme</li></ul></li><li>- Left arm straight and some distance from the right arm</li></ul>
Volume Down	Left Dab	<ul style="list-style-type: none"><li>- Head into the left elbow<ul style="list-style-type: none"><li>- Distance between head and elbow doesn't have to be too extreme</li></ul></li><li>- Right arm straight and some distance from the left arm</li></ul>
Channel Up	Arms Straight	<ul style="list-style-type: none"><li>- Arms make a 180 degree, from left to right</li><li>- Arms are parallel to the ground</li><li>- Arms must be in full view</li></ul>
Channel Down	Muscle Man	<ul style="list-style-type: none"><li>- Arms are at 90 degree, hands facing upward</li><li>- Arms must be in full view</li></ul>
Power On/Off	Hands Together Relaxing	<p>Hands Together</p> <ul style="list-style-type: none"><li>- Have both hands be touching each other</li><li>- Wrists must be in full view</li></ul> <p>Relaxing</p> <ul style="list-style-type: none"><li>- Have both hands be on top of the head; order doesn't matter</li><li>- Wrists must be in full view</li></ul>

## BerryIMU Gestures

There are currently 6 possible gestures which you can use to control your TV. Each gesture corresponds to a different command, with detailed tutorial videos being accessible through the GUI for extra assistance. The commands are as follows:

Command	Gesture	Detailed Explanation
Volume up	Tilt right	The BerryIMU should be parallel to the floor while you attempt this gesture. With the BerryIMU and Raspberry Pi combo facing forwards, tilt the entire device to the right.
Volume down	Tilt left	The BerryIMU should be parallel to the floor while you attempt this gesture. With the BerryIMU and Raspberry Pi combo facing forwards, tilt the entire device to the left.
Channel up	Flick right	With the BerryIMU and Raspberry Pi combo facing forwards, quickly flick the device to the right by flicking your wrist outwards.
Channel down	Flick left	With the BerryIMU and Raspberry Pi combo facing forwards, quickly flick the device to the left by flicking your wrist inwards.
Power on	Flick up	With the BerryIMU and Raspberry Pi combo facing forwards, quickly flick the device up.
Power off	Flick down	With the BerryIMU and Raspberry Pi combo facing forwards, quickly flick the device down. It is okay to lift the device first when attempting to flick down.

It is helpful to always have the BerryIMU and Raspberry Pi facing forwards towards the TV, as well as parallel to the floor. When setting down the device, do so without tilting the Pi such that it does not think you are attempting to turn off the TV. The BerryIMU also supports fall detection, so dropping the device (in most cases) will not result in accidental inputs.

## *Voice Commands*

There are currently 6 possible voice commands which you can use to control your TV, along with a few more extensions. Each voice command works by detecting certain keywords in your speech. The exact sentence structure does not matter as long as the keywords are detected. The commands are as follows:

Command	Voice Command
Volume up	"... volume ... up."
Volume up by <x>	"... volume ... increase/up ... <x>."
Volume down	"... volume ... down."
Volume down by <x>	"... volume ... decrease/down ... <x>."
Channel up	"... channel ... up."
Change channel to <y>	"... channel ... <y>."
Channel down	"... channel ... down."
Power on	"... power ... on."
Power off	"... power ... off."

Please note that it is important for the microphone to pick up your speech clearly. The best accuracy will be achieved by speaking clearly and loudly in a quiet environment. Speaking with the microphone closer will also aid in improving accuracy.

## *GUI*

The GUI of the AirController provides an interface with which to access the webcam and speech portions of the controller. By using the "Start Controller" and "Stop Controller" buttons, the user may enable webcam pose detection and voice command detection to be able to control their TV. There are also other controller options if the user only wishes to launch a certain part of the AirController.

Detailed tutorial videos are also available via the GUI, including those for the 6 BerryIMU gestures and for each of the poses. These provide a baseline for what each gesture is expected to look like, and can help the user determine which gesture to make for each command.

There are also tutorial pictures available via the GUI for the poses to better demonstrate the expected pose by the user in order to properly register the command.

### *IR Interfacing*

Assuming proper setup, the IR emitter should work generally seamlessly. The IR emitting Pi should be placed near the TV, older models tend to prefer a closer setup. Usually the IR receiver on the TV is on the right or sometimes left corner of the TV, the device should be placed within a few feet of it. If it is not picking up the signals, please place it closer and mess around with where to put it.

The IR emitter has a small LED on it indicating when it is transmitting. This can be used to make sure signals are being received through MQTT.