LESSON C-6

ACTIVITY #3:
PLAYING YOUR VOICE FILE WITH THE IR REMOTE

In this activity, you will modify the IR Remote program named **ir_rgb.py** that you built in Lesson #2, Activity #3. You will be adding the ability to play your voice file through the speaker when button 4 is pressed.

Step #1

The first step will be to create a copy of **ir_rgb.py** that you can modify to play your recording. Open Thonny and open the **ir_rgb.py** file located in /home/pi. Use **Save As** in Thonny to save a new copy as **ir_rgb_audio.py**:

Press the **OK** button at the bottom of the window to save the new copy. This way the original **ir_rgb.py** file will remain unmodified and can be used as a base for more projects in the future.

Step #2

Only a few modifications need to be made to this program to incorporate playing back your recorded file. The **os** module must be imported so CLI commands can be run from within Python. Import the **os** module by adding the highlighted command below to the end of the import block:

```
import socket
import RPi.GPIO as GPIO
import time
import os

red = 13
```

Step #3

The other modification this program needs is to assign some commands to IR Remote button #4. These commands will include:

1. Turning off the LED by using the **led_update** function
2. Setting GPIO18 to PWM mode ALT5 so it can output left channel audio
3. Playing your recorded file using aplay
4. Setting GPIO18 back to its default state as an input

These commands will only run if the **elif** block finds that the value of **button** is equal to **'BTN_4'**, which means that IR Remote button #4 was pressed. Add the highlighted block below between the **'BTN_3'** and **'BTN_OK'** blocks in the program:

```
    elif button == 'BTN_3':

        print('Blue')

        led_update(0,0,1)

    elif button == 'BTN_4':

        led_update(0,0,0)

        os.system("gpio -g mode 18 ALT5")

        os.system("aplay /home/pi/my_voice.wav")

        os.system("gpio -g mode 18 in")

    elif button == 'BTN_OK':

        print('Program Exiting...')

        led_update(0,0,0)

        GPIO.cleanup()

        raise SystemExit()
```

Make sure the indentation matches the blocks above and below so your new **elif** condition will be evaluated along with the other conditions.

Step #4

Run the program and press the buttons on the IR Remote while pointing it at the IR receiver. Buttons 1, 2, and 3 will still control the red, green, and blue elements of the LED, while button 4 will now control the playing of the recorded audio. Use the OK button on the remote to exit the program.

If the recorded audio does not play as expected, verify there are no typos in the highlighted block that you added in Step #3. If needed, return to Activity #2 and verify that you can play the recorded file in Terminal, making any audio level adjustments that might be required.

Leave this circuit built on your breadboard for use in Lesson 7.