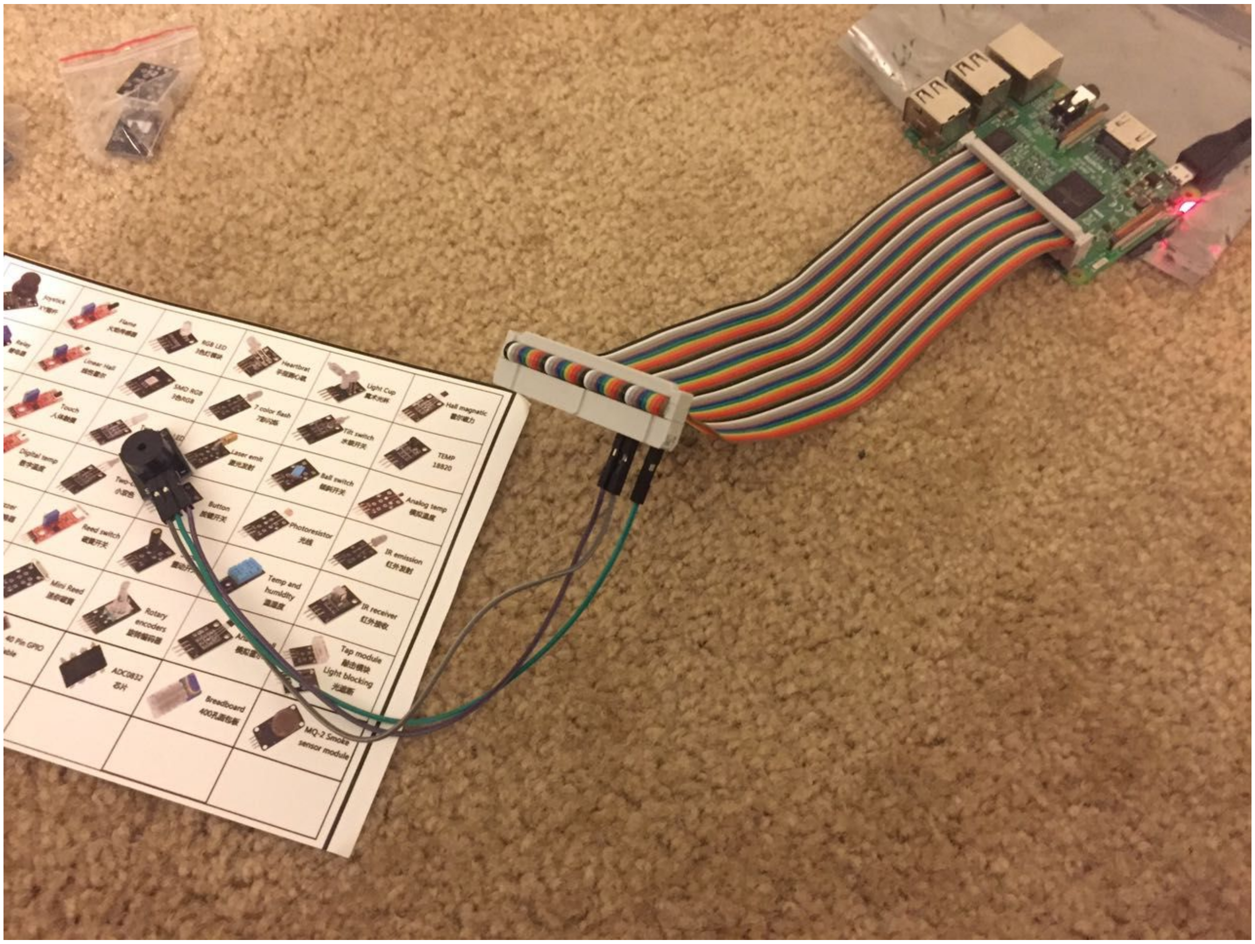
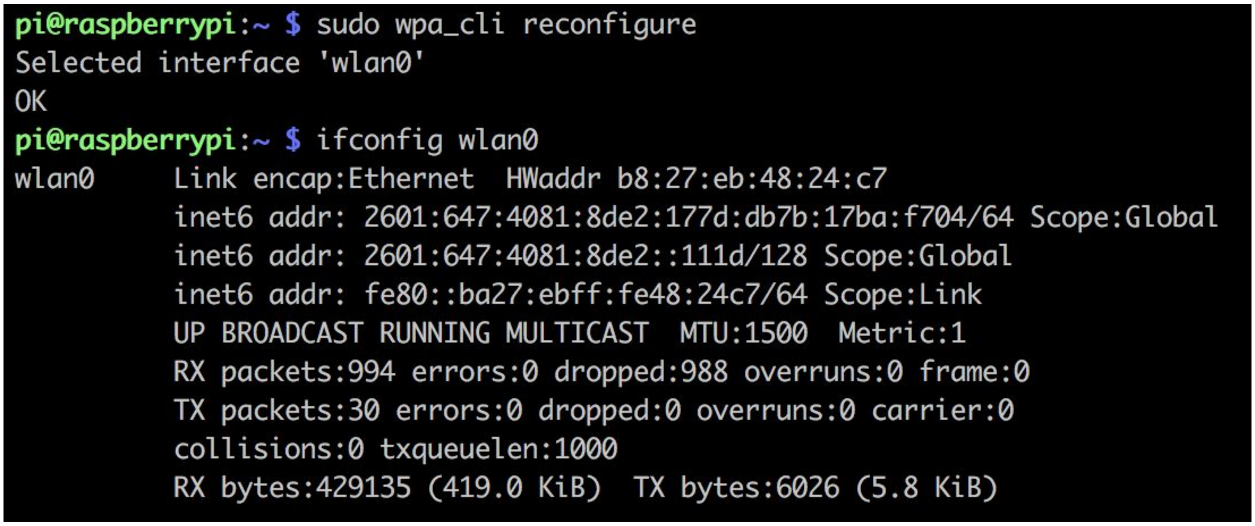
Yuhao Xu(yuhaox1)

**Start Raspberry pi and connect to a buzzer.**



**Set up wifi connection:**

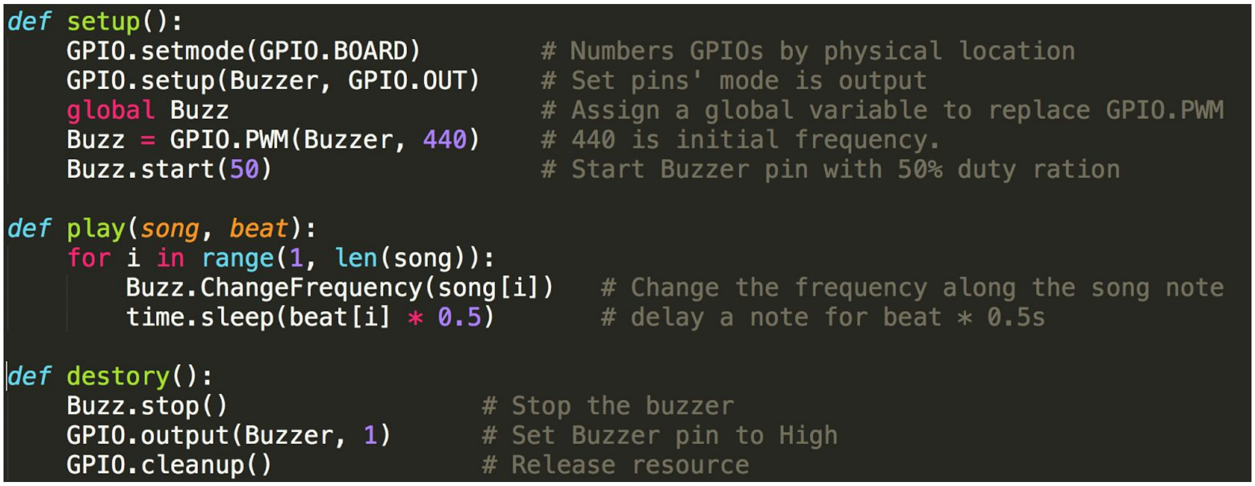


```
pi@raspberrypi:~ $ sudo wpa_cli reconfigure
Selected interface 'wlan0'
OK
pi@raspberrypi:~ $ ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr b8:27:eb:48:24:c7
          inet6 addr: 2601:647:4081:8de2:177d:db7b:17ba:f704/64 Scope:Global
          inet6 addr: 2601:647:4081:8de2::111d/128 Scope:Global
          inet6 addr: fe80::ba27:ebff:fe48:24c7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:994 errors:0 dropped:988 overruns:0 frame:0
          TX packets:30 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:429135 (419.0 KiB)  TX bytes:6026 (5.8 KiB)
```

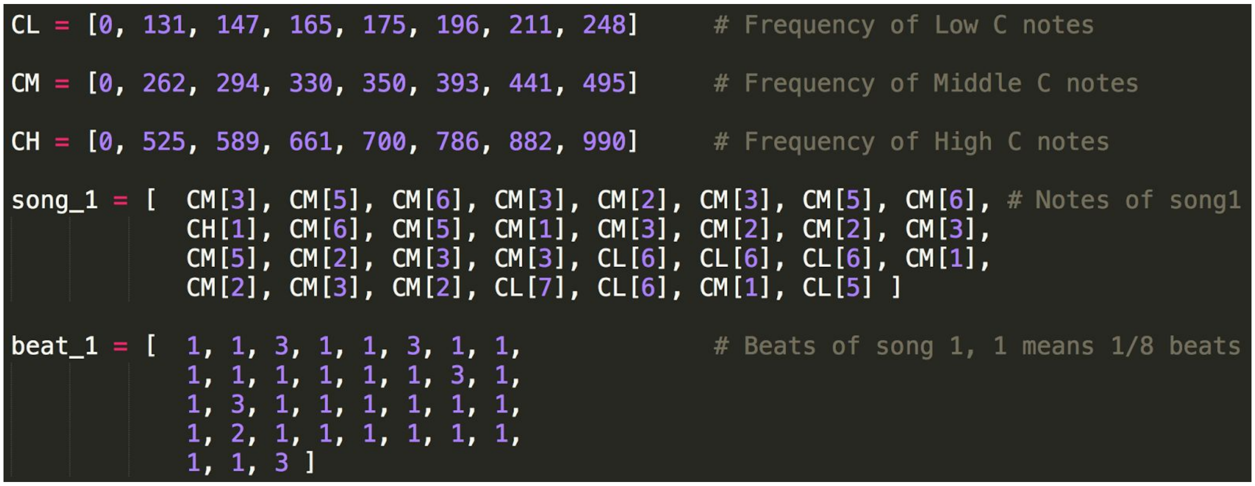
**Connection ports:**

```
#--------------------------------------------------
#                    Passive buzzer
#        VCC  ------------------  3.3V
#        GND ------------------ GND
#        SIG   ------------------  GPIO Pin 7
#
#--------------------------------------------------
```

**Main Functions:(use Python to access GPIO for singging)**

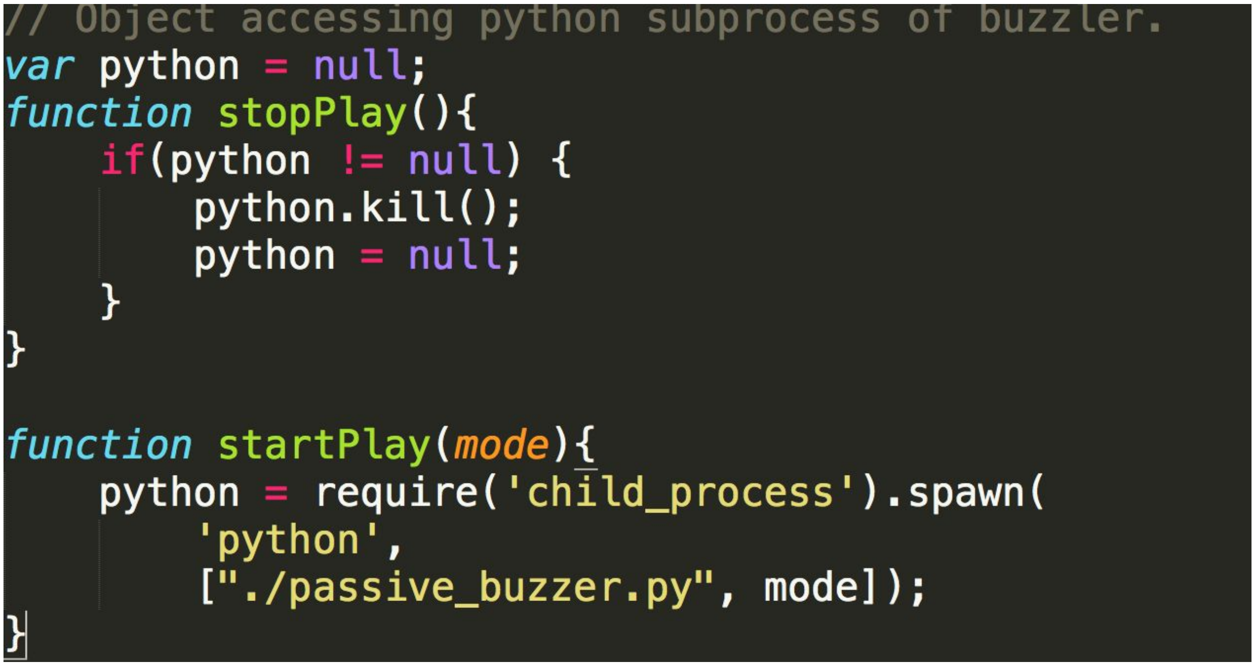


```
def setup():
    GPIO.setmode(GPIO.BOARD)        # Numbers GPIOs by physical location
    GPIO.setup(Buzzer, GPIO.OUT)    # Set pins' mode is output
    global Buzz                     # Assign a global variable to replace GPIO.PWM
    Buzz = GPIO.PWM(Buzzer, 440)    # 440 is initial frequency.
    Buzz.start(50)                  # Start Buzzer pin with 50% duty ration

def play(song, beat):
    for i in range(1, len(song)):
        Buzz.ChangeFrequency(song[i])   # Change the frequency along the song note
        time.sleep(beat[i] * 0.5)       # delay a note for beat * 0.5s

def destory():
    Buzz.stop()                 # Stop the buzzer
    GPIO.output(Buzzer, 1)      # Set Buzzer pin to High
    GPIO.cleanup()              # Release resource
```

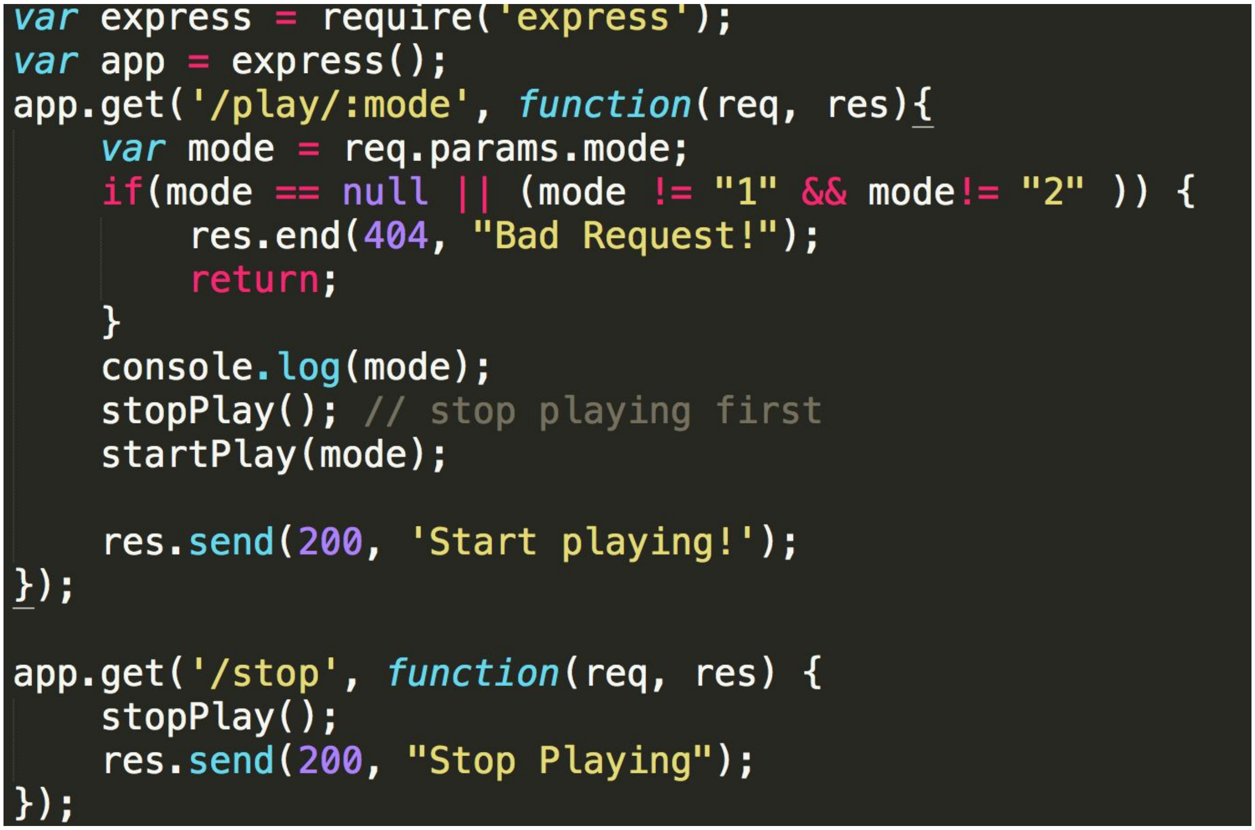


```
CL = [0, 131, 147, 165, 175, 196, 211, 248]     # Frequency of Low C notes

CM = [0, 262, 294, 330, 350, 393, 441, 495]     # Frequency of Middle C notes

CH = [0, 525, 589, 661, 700, 786, 882, 990]     # Frequency of High C notes

song_1 = [  CM[3], CM[5], CM[6], CM[3], CM[2], CM[3], CM[5], CM[6], # Notes of song1
            CH[1], CM[6], CM[5], CM[1], CM[3], CM[2], CM[2], CM[3],
            CM[5], CM[2], CM[3], CM[3], CL[6], CL[6], CL[6], CM[1],
            CM[2], CM[3], CM[2], CL[7], CL[6], CM[1], CL[5] ]

beat_1 = [  1, 1, 3, 1, 1, 3, 1, 1,                 # Beats of song 1, 1 means 1/8 beats
            1, 1, 1, 1, 1, 1, 3, 1,
            1, 3, 1, 1, 1, 1, 1, 1,
            1, 2, 1, 1, 1, 1, 1, 1,
            1, 1, 3 ]
```

Main Function:(use Node.js to receive HTTP requests and invoke singing):

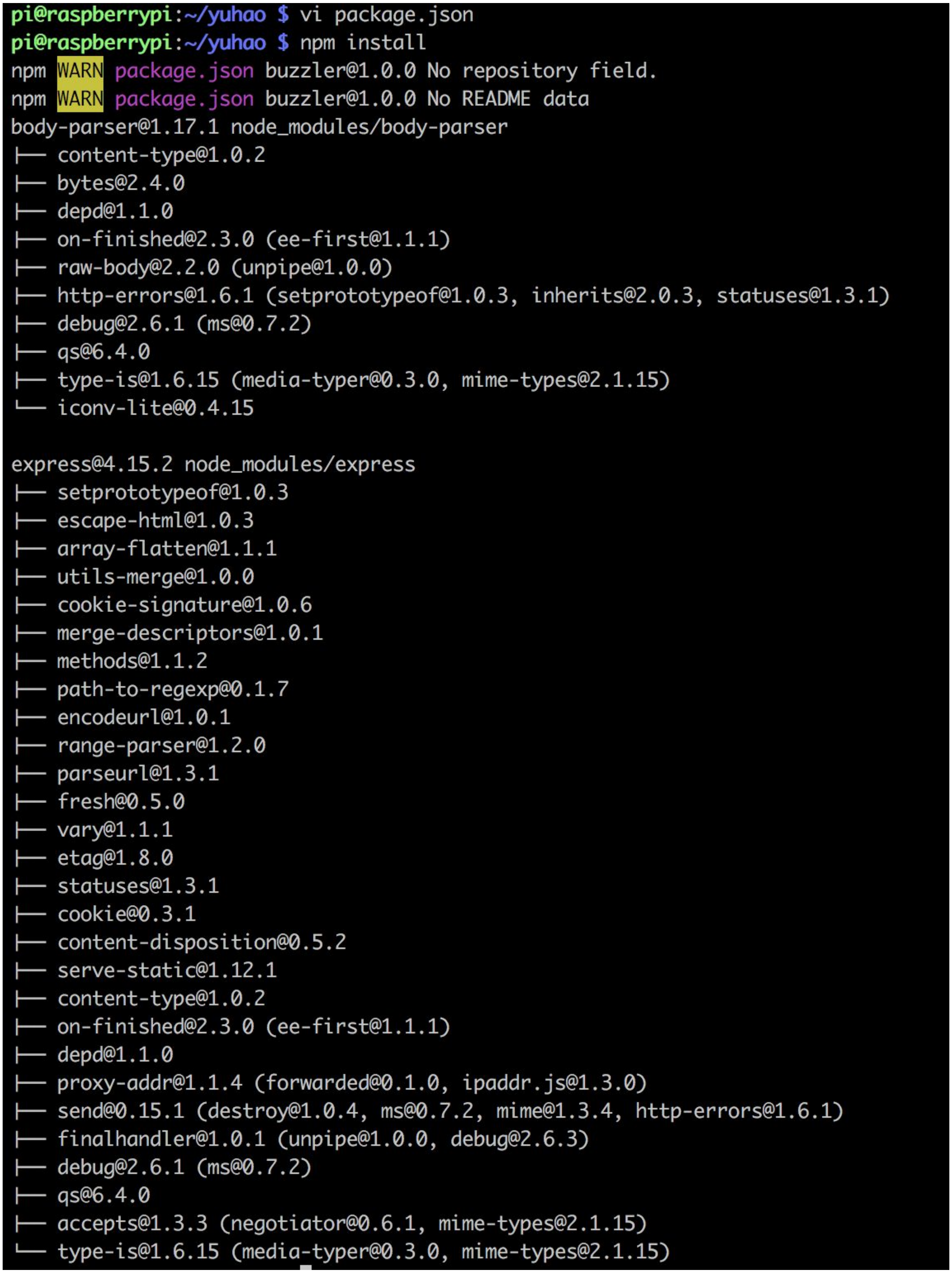


```
// Object accessing python subprocess of buzzler.
var python = null;
function stopPlay(){
    if(python != null) {
        python.kill();
        python = null;
    }
}

function startPlay(mode){
    python = require('child_process').spawn(
        'python',
        ["./passive_buzzer.py", mode]);
}
```



```
var express = require('express');
var app = express();
app.get('/play/:mode', function(req, res){
    var mode = req.params.mode;
    if(mode == null || (mode != "1" && mode!= "2" )) {
        res.end(404, "Bad Request!");
        return;
    }
    console.log(mode);
    stopPlay(); // stop playing first
    startPlay(mode);

    res.send(200, 'Start playing!');
});

app.get('/stop', function(req, res) {
    stopPlay();
    res.send(200, "Stop Playing");
});
```

**Install Dependencies:**



```
pi@raspberrypi:~/yuhao $ vi package.json
pi@raspberrypi:~/yuhao $ npm install
npm WARN package.json buzzler@1.0.0 No repository field.
npm WARN package.json buzzler@1.0.0 No README data
body-parser@1.17.1 node_modules/body-parser
├── content-type@1.0.2
├── bytes@2.4.0
├── depd@1.1.0
├── on-finished@2.3.0 (ee-first@1.1.1)
├── raw-body@2.2.0 (unpipe@1.0.0)
├── http-errors@1.6.1 (setprototypeof@1.0.3, inherits@2.0.3, statuses@1.3.1)
├── debug@2.6.1 (ms@0.7.2)
├── qs@6.4.0
├── type-is@1.6.15 (media-typer@0.3.0, mime-types@2.1.15)
└── iconv-lite@0.4.15

express@4.15.2 node_modules/express
├── setprototypeof@1.0.3
├── escape-html@1.0.3
├── array-flatten@1.1.1
├── utils-merge@1.0.0
├── cookie-signature@1.0.6
├── merge-descriptors@1.0.1
├── methods@1.1.2
├── path-to-regexp@0.1.7
├── encodeurl@1.0.1
├── range-parser@1.2.0
├── parseurl@1.3.1
├── fresh@0.5.0
├── vary@1.1.1
├── etag@1.8.0
├── statuses@1.3.1
├── cookie@0.3.1
├── content-disposition@0.5.2
├── serve-static@1.12.1
├── content-type@1.0.2
├── on-finished@2.3.0 (ee-first@1.1.1)
├── depd@1.1.0
├── proxy-addr@1.1.4 (forwarded@0.1.0, ipaddr.js@1.3.0)
├── send@0.15.1 (destroy@1.0.4, ms@0.7.2, mime@1.3.4, http-errors@1.6.1)
├── finalhandler@1.0.1 (unpipe@1.0.0, debug@2.6.3)
├── debug@2.6.1 (ms@0.7.2)
├── qs@6.4.0
├── accepts@1.3.3 (negotiator@0.6.1, mime-types@2.1.15)
└── type-is@1.6.15 (media-typer@0.3.0, mime-types@2.1.15)
```

**How to run:**

```
node app.js
```

Now you may see the video for final demo.

**References:**

https://www.sunfounder.com/learn/Sensor-Kit-v1-0-for-Raspberry-Pi/lesson-11-buzzer-sensor-kit-1-0-for-pi.html