Final Project

Introduction

This is how to make a digital mp3 player on a raspberry pi.

What you need

1. Raspberry pi
2. 3 Female to male jumper cables
3. A touch sensor or button
4. Breadboard

The wiring

Adafruit T-Cobbler Plus
for Raspberry Pi
3V3
SDA
SCL
#4
GND
#17
#27
#22
3.3V
GND
#25
CE0
CE1
EEC
GND
#12
GND
#16
#20
#21
5.0V
5.0V
GND
TXD
RXD
#18
GND
#23
CLK
EEC
#5
#6
#13
#19
#26
GND
Touch module
keyestudio
WWW.CANAKIT.C

- The wiring is very simple, all you need is 3 jumper wires, one that connects to the GPIO pin that you want to use, one that connects to voltage, and one that connects to ground.

The Code

Here is the code step by step

```
import RPi.GPIO as GPIO
import pygame
from pygame import mixer
import time
import random
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setwarnings(GPIO.BOARD)
```

This first block is setting up the rest of the code. First you need to import the GPIO library and the pygame library. Then you need to import the aspects from each library. Then you set up your GPIO pin, I used number 5 for mine. Next you need you need to set the warnings to make the code run smoothly.

```
GPIO.setup(5, GPIO.IN, pull_up_down = GPIO.PUD_DOWN) # sets up


songlist = ['song1.mp3', 'song2.mp3', 'song3.mp3', 'song4.mp3']

q = len(songlist)
a = random.randint(0, q) # this is the random song generator, i
m = songlist[a]
```

This next block of code is setting up the random generator to generate a random song from the list. You need to set up the GPIO pin to be active and inactive with the first line. Then I chose random songs and added them to the code (you can rename the songs to what I did if you want to). The next part is the actual random generator. q is the variable for the length of the song list

so it chooses between the whole list, a is the random number chosen between 0 and the length of the list, and m is showing that the song list is random.

```
if m == 'song1.mp3':
    i = 380
if m == 'song2.mp3':
    i = 240
if m == 'song3.mp3':
    i = 30
if m == 'song4.mp3':
    i = 224
```

All of these if statements are simply showing how long the song is in seconds.

```
while True:
    if GPIO.input(5) == GPIO.HIGH:
        pygame.init()
        pygame.mixer.init()
        pygame.mixer.music.load(m)
        pygame.mixer.music.play()
        time.sleep(i) # i is the nu
        print('song is over')
        exit() # exits out of pygam
```

This last block of code is what actually plays the song. It is a while loop so you can continuously play the songs randomly. The first line is when you touch the sensor then the song will play. The next 4 lines are from pygame and loads then plays the song. The time.sleep function is i because it is the length of the song. Then after the song ends it prints that the song is over and the exits out of the system.

Summary

Overall, this is a pretty simple project. You just need to be sure that all of your code is in the correct order and the GPIO pins are fully connected and in the right spot. Also be sure to download the GPIO and pygame Libraries!