Exam 2: Optical Encoders

Max Huggins – UCA Department of Physics
Astronomy

February 9, 2021

Abstract

Three experiments were conducted to explore the various operations of optical encoders. The first, a direction (clockwise or counterclockwise) was found from the reflection and absorption of light in an encoder. The second and third determined sector location of the device from absorption and reflection of light.

1 Introduction

Optical encoders can be used to produce extremely fine resolutions for directional information. They work on a simple basis; typically using digital logic to determine whether or not a sector is occupied by a combination of digital signals. An example of this combination of signals is Gray code or reflected binary code (RBC.) This type of code allows for the assignment of unique binary combinations without repetition throughout a range of bits. For example, a three bit encoder can uniquely identify eight different sectors. Each identified by a range of angles. This is shown in figure 1. These types of encoders work on the premise of light absorption and reflection. An infrared (IR) sensor transistor and IR LED can be used in each bit to determine an on/off state for each bit and therefore, can determine a unique set of up to 8 sectors:

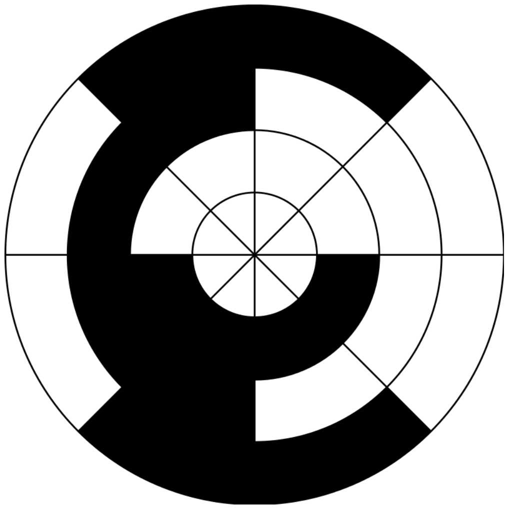
Figure 1: Here is a three bit encoder.

$$2^N = TotalCombinations \tag{1}$$

The more bits, the greater the resolution available for the sensors.

2 Analysis

In this experiment, direction and sector were detected using the method described above. Firstly, a program was written to determine the movement direction using an encoder with two bits. Each bit had a black section slightly offset with another black section in the next bit. Something like figure 2. A simple circuit was constructed to take advantage of a IR sensor transistor and LED, shown in figure 3.

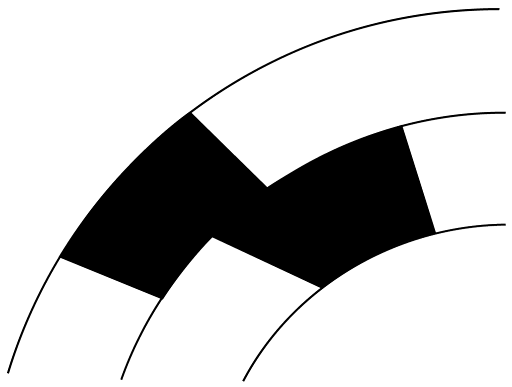
Figure 2: Here is a section of a two bit encoder.

The NPN type transistor is typically open, this means the GPIO pin reads high in the circuit. Once the base of the transistor senses IR, the switch between the emitter and collector closes and the GPIO pin goes low. In other words, when the sensor passes over a black section the GPIO pin reads True and when the sensor passes over a white section it reads True. A program was written to take advantage of this and the general flow is outlined in figure 4 and the actual code is available in the appendix.

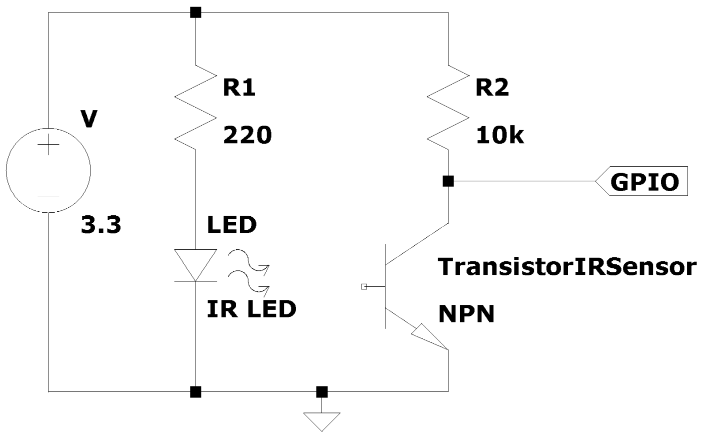

Figure 3: Circuit used to sense for a black or white section in the encoders for each sensor.

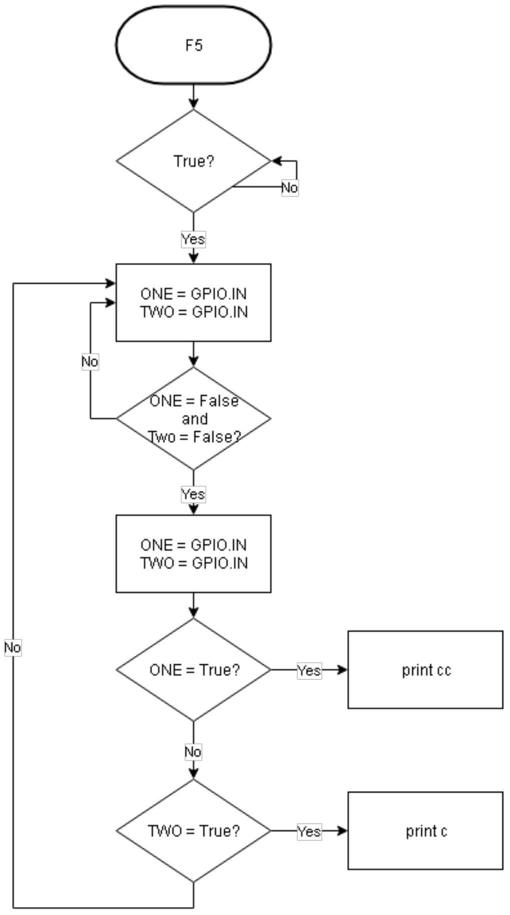

Figure 4: Flow for interpreting direction of first encoder.

The next encoder used was a 3-bit encoder like the one in figure 1. The program was similar and worked based on the same principle. Each sector (every 45 degrees) was given a unique identifier based on Gray code and these let the user determine what section they were in. The flow of the program is given in figure 5 and the actual code is available in the appendix.
The third was a 4-bit encoder. This one could allow for $2^4 = 16$ total sectors to be uniquely identified. This encoder was made in Adobe Illustrator and is shown in figure 7. The operation of this one is essentially the same as the 3-bit, but with one extra sensor. Also, the code was modified to allow for some data collection and cleaner outputs. The flow is shown in figure 6 and the actual code is available in the appendix.

The values from the text file produced are shown in the table below in order: 1,2,3,4,5,6,7,8,9 ,10,11,12,13,14,15,16,15,14,13,12,11,10,9,8,7,6,5,4,3,2,1,2,1,2. These values were produced from starting a clockwise turn at the first sector and going through to the 16th sector, then beginning a counterclockwise turn until 32 total sectors were crossed.

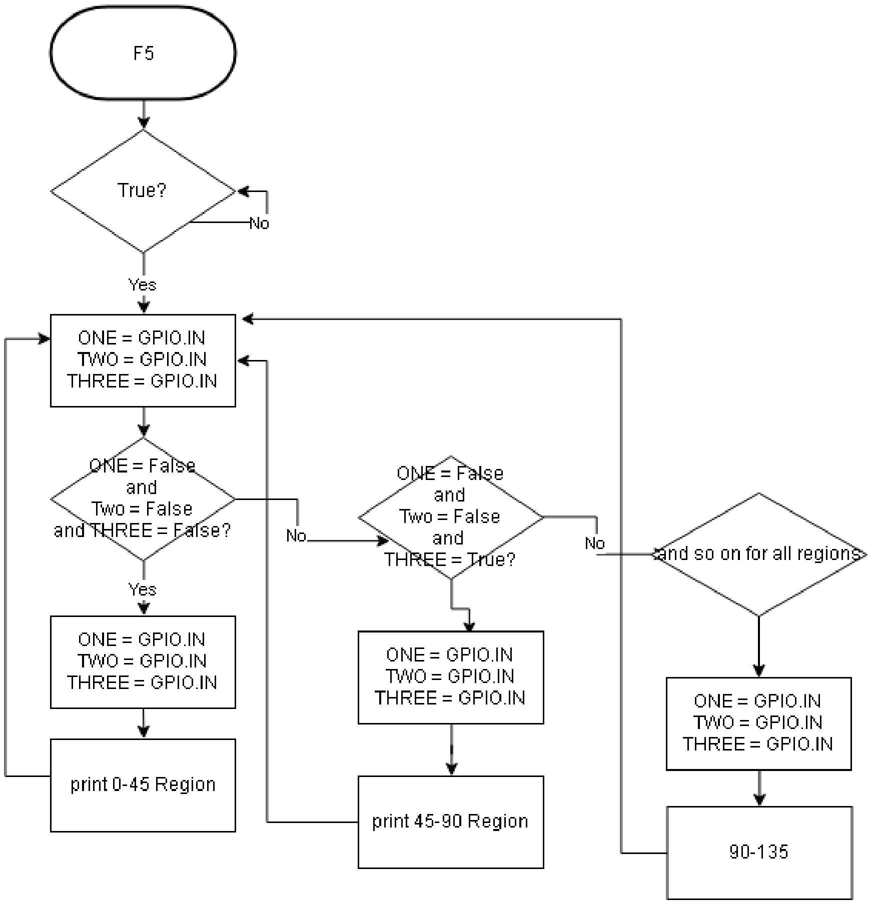

Figure 5: Flow for the three bit encoder

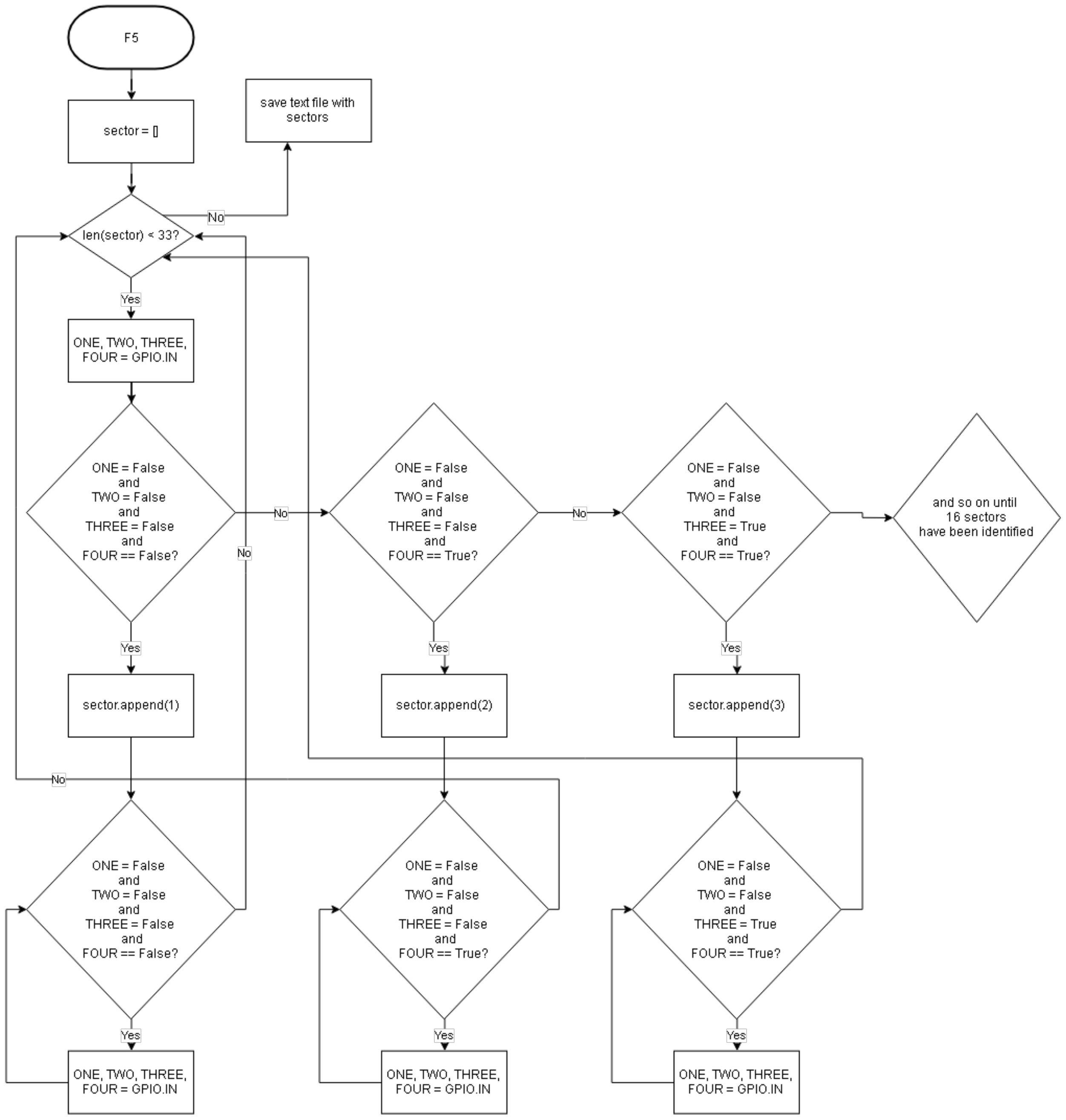
F5
sector = []
save text file with sectors
len(sector) < 33?
No
Yes
ONE, TWO, THREE, FOUR = GPIO.IN
ONE = False and TWO = False and THREE = False and FOUR == False?
No
ONE = False and TWO = False and THREE = False and FOUR == True?
No
ONE = False and TWO = False and THREE = True and FOUR == True?
and so on until 16 sectors have been identified
Yes
sector.append(1)
sector.append(2)
sector.append(3)
No
ONE = False and TWO = False and THREE = False and FOUR == False?
ONE = False and TWO = False and THREE = False and FOUR == True?
ONE = False and TWO = False and THREE = True and FOUR == True?
Yes
ONE, TWO, THREE, FOUR = GPIO.IN

Figure 6: Flow for the four bit encoder

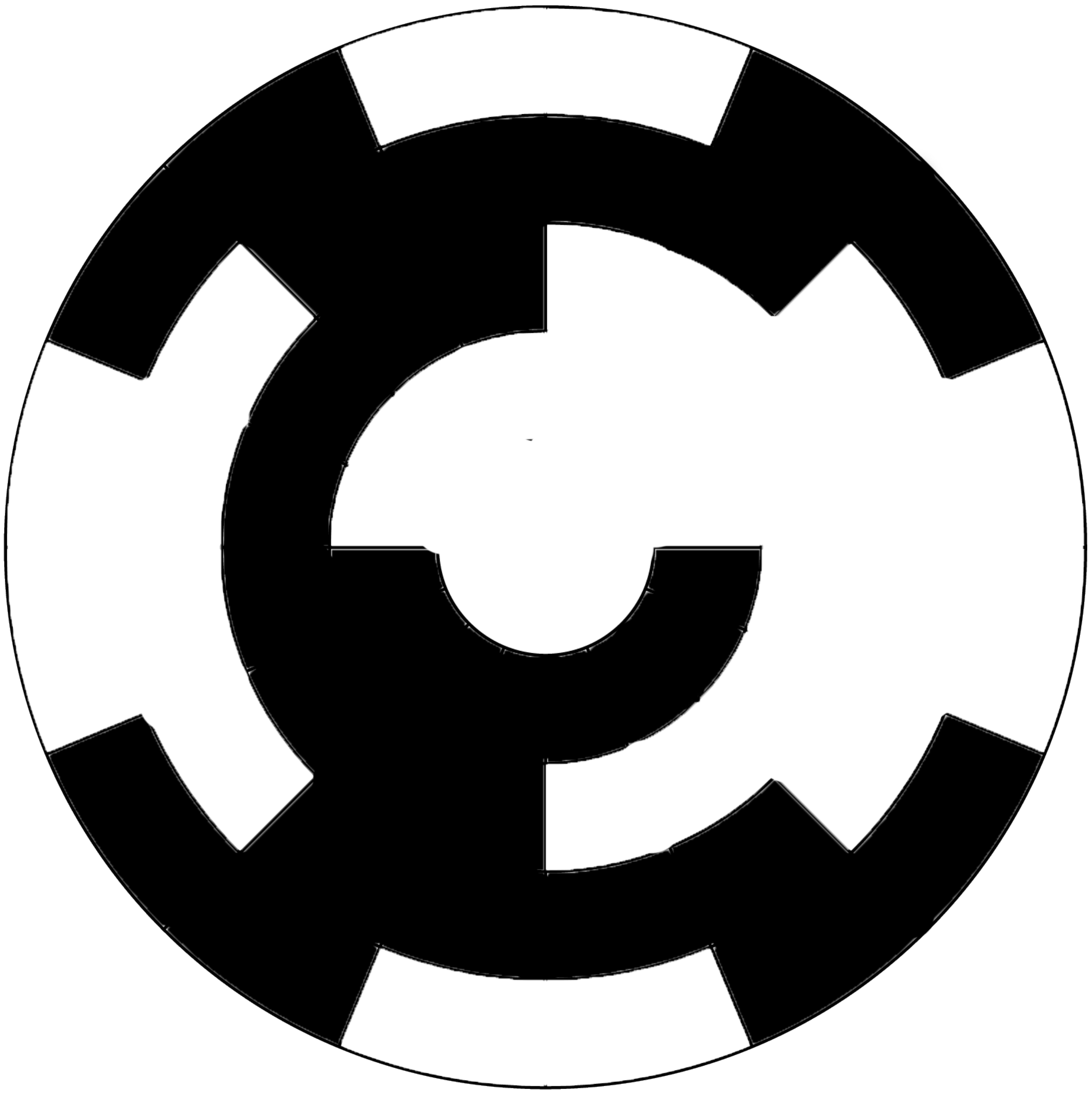

Figure 7: 4-bit encoder

3 Conclusion

Optical encoding is an interesting solution to determining orientation of objects. These three optical encoders were used for clockwise/counterclockwise movement and sector identification. The future considerations for these types of devices are certainly endless as they have found themselves in numerous environments and scenarios.

Python script for first encoder.

```python
import RPi.GPIO as GPIO
import time

IR_1 = 36
IR_2 = 32

GPIO.setmode(GPIO.BOARD)
GPIO.setup(IR_1,GPIO.IN)
GPIO.setup(IR_2,GPIO.IN)

try:
    while True:
        ONE = GPIO.input(IR_1)
        TWO = GPIO.input(IR_2)
        if ONE == False and TWO == False:
            ONE = GPIO.input(IR_1)
            TWO = GPIO.input(IR_2)
            if ONE == True:
                print('CC')
            elif TWO == True:
                print('C')

except KeyboardInterrupt:
    print('toast got burnt')

finally:
    print('clean')
    GPIO.cleanup()
```

Python script for second encoder.

```python
import RPi.GPIO as GPIO
import time

IR_1 = 36
IR_2 = 32
IR_3 = 31

GPIO.setmode(GPIO.BOARD)
GPIO.setup(IR_1,GPIO.IN)
GPIO.setup(IR_2,GPIO.IN)
GPIO.setup(IR_3,GPIO.IN)

try:
    while True:
        ONE = GPIO.input(IR_1)
        TWO = GPIO.input(IR_2)
        THREE = GPIO.input(IR_3)
        if ONE == False and TWO == False and THREE == False:
            print('0-45 Region')
        if ONE == False and TWO == False and THREE == True:
            print('45-90 Region')
        if ONE == False and TWO == True and THREE == True:
            print('90-135 Region')
        if ONE == False and TWO == True and THREE == False:
            print('135-180 Region')
        if ONE == True and TWO == True and THREE == False:
            print('180-225 Region')
        if ONE == True and TWO == True and THREE == True:
            print('225-270 Region')
        if ONE == True and TWO == False and THREE == True:
            print('270-315 Region')
        if ONE == True and TWO == False and THREE == False:
            print('315-360 Region')

except KeyboardInterrupt:
    print('toast got burnt')

finally:
    print('clean')
    GPIO.cleanup()
```

Python script for third encoder.

```python
import RPi.GPIO as GPIO
import time

IR_1 = 33
IR_2 = 36
IR_3 = 32
IR_4 = 31

GPIO.setmode(GPIO.BOARD)
GPIO.setup(IR_1,GPIO.IN)
GPIO.setup(IR_2,GPIO.IN)
GPIO.setup(IR_3,GPIO.IN)
GPIO.setup(IR_4,GPIO.IN)


try:
    sector = []
    while len(sector) < 33:
        ONE = GPIO.input(IR_1)
        TWO = GPIO.input(IR_2)
        THREE = GPIO.input(IR_3)
        FOUR = GPIO.input(IR_4)
        if ONE == False and TWO == False and THREE == False and FOUR == False:
            sector.append('1')
            while ONE == False and TWO == False and THREE == False and FOUR == False:
                ONE = GPIO.input(IR_1)
                TWO = GPIO.input(IR_2)
                THREE = GPIO.input(IR_3)
                FOUR = GPIO.input(IR_4)
        if ONE == False and TWO == False and THREE == False and FOUR == True:
            sector.append('2')
            while ONE == False and TWO == False and THREE == False and FOUR == True:
                ONE = GPIO.input(IR_1)
                TWO = GPIO.input(IR_2)
                THREE = GPIO.input(IR_3)
                FOUR = GPIO.input(IR_4)
        if ONE == False and TWO == False and THREE == True and FOUR == True:
            sector.append('3')
            while ONE == False and TWO == False and THREE == True and FOUR == True:
                ONE = GPIO.input(IR_1)
                TWO = GPIO.input(IR_2)
                THREE = GPIO.input(IR_3)
                FOUR = GPIO.input(IR_4)
        if ONE == False and TWO == False and THREE == True and FOUR == False:
            sector.append('4')
            while ONE == False and TWO == False and THREE == True and FOUR == False:
                ONE = GPIO.input(IR_1)
                TWO = GPIO.input(IR_2)
                THREE = GPIO.input(IR_3)
                FOUR = GPIO.input(IR_4)
        if ONE == False and TWO == True and THREE == True and FOUR == False:
            sector.append('5')
            while ONE == False and TWO == True and THREE == True and FOUR == False:
                ONE = GPIO.input(IR_1)
                TWO = GPIO.input(IR_2)
                THREE = GPIO.input(IR_3)
                FOUR = GPIO.input(IR_4)
        if ONE == False and TWO == True and THREE == True and FOUR == True:
            sector.append('6')
            while ONE == False and TWO == True and THREE == True and FOUR == True:
                ONE = GPIO.input(IR_1)
                TWO = GPIO.input(IR_2)
                THREE = GPIO.input(IR_3)
                FOUR = GPIO.input(IR_4)
        if ONE == False and TWO == True and THREE == False and FOUR == True:
            sector.append('7')
            while ONE == False and TWO == True and THREE == False and FOUR == True:
                ONE = GPIO.input(IR_1)
                TWO = GPIO.input(IR_2)
                THREE = GPIO.input(IR_3)
                FOUR = GPIO.input(IR_4)
        if ONE == False and TWO == True and THREE == False and FOUR == False:
            sector.append('8')
            while ONE == False and TWO == True and THREE == False and FOUR == False:
                ONE = GPIO.input(IR_1)
                TWO = GPIO.input(IR_2)
                THREE = GPIO.input(IR_3)
                FOUR = GPIO.input(IR_4)
        if ONE == True and TWO == True and THREE == False and FOUR == False:
            sector.append('9')
```

```python
            while ONE == True and TWO == True and THREE == False and FOUR == False:
                ONE = GPIO.input(IR_1)
                TWO = GPIO.input(IR_2)
                THREE = GPIO.input(IR_3)
                FOUR = GPIO.input(IR_4)
        if ONE == True and TWO == True and THREE == False and FOUR == True:
            sector.append('10')
            while ONE == True and TWO == True and THREE == False and FOUR == True:
                ONE = GPIO.input(IR_1)
                TWO = GPIO.input(IR_2)
                THREE = GPIO.input(IR_3)
                FOUR = GPIO.input(IR_4)
        if ONE == True and TWO == True and THREE == True and FOUR == True:
            sector.append('11')
            while ONE == True and TWO == True and THREE == True and FOUR == True:
                ONE = GPIO.input(IR_1)
                TWO = GPIO.input(IR_2)
                THREE = GPIO.input(IR_3)
                FOUR = GPIO.input(IR_4)
        if ONE == True and TWO == True and THREE == True and FOUR == False:
            sector.append('12')
            while ONE == True and TWO == True and THREE == True and FOUR == False:
                ONE = GPIO.input(IR_1)
                TWO = GPIO.input(IR_2)
                THREE = GPIO.input(IR_3)
                FOUR = GPIO.input(IR_4)
        if ONE == True and TWO == False and THREE == True and FOUR == False:
            sector.append('13')
            while ONE == True and TWO == False and THREE == True and FOUR == False:
                ONE = GPIO.input(IR_1)
                TWO = GPIO.input(IR_2)
                THREE = GPIO.input(IR_3)
                FOUR = GPIO.input(IR_4)
        if ONE == True and TWO == False and THREE == True and FOUR == True:
            sector.append('14')
            while ONE == True and TWO == False and THREE == True and FOUR == True:
                ONE = GPIO.input(IR_1)
                TWO = GPIO.input(IR_2)
                THREE = GPIO.input(IR_3)
                FOUR = GPIO.input(IR_4)
        if ONE == True and TWO == False and THREE == False and FOUR == True:
            sector.append('15')
            while ONE == True and TWO == False and THREE == False and FOUR == True:
                ONE = GPIO.input(IR_1)
                TWO = GPIO.input(IR_2)
                THREE = GPIO.input(IR_3)
                FOUR = GPIO.input(IR_4)
        if ONE == True and TWO == False and THREE == False and FOUR == False:
            sector.append('16')
            while ONE == True and TWO == False and THREE == False and FOUR == False:
                ONE = GPIO.input(IR_1)
                TWO = GPIO.input(IR_2)
                THREE = GPIO.input(IR_3)
                FOUR = GPIO.input(IR_4)
    file = open('Pt3.txt', 'w')
    for n in range(len(sector)):
        file.write(str(sector[n]) + ',')
    file.close()


except KeyboardInterrupt:
    print('toast got burnt')

finally:
    print('clean')
    GPIO.cleanup()
```