Final Project: Asbestos Inspection Robot

Robotics (Theory, Practice, Philosophy)

Ka Hei Pinky Chow, Bryony Bar

Final Project

Asbestos, a toxic material of high public health concern, is widely used as building material in plentiful developing countries. Every year, over 107,000 people die in the world from exposure to this chemical substance. Our robot design thereby aims for testing Asbestos level effectively in the indoor environment. It is mobile with the aid of wheels. It is equipped with an on-site Asbestos sensor, 3 LED lights (green, yellow and red). It will move in a straight line. It s and test the level of Asbestos every 5 metres. The lights will turn on according to the level of stos in the air. When the Asbestos level exceeds the warning level, the robot will produce alerting sound be g forward. When it faces the wall or obstacles for 10 seconds, it will move in the reve n to the starting point.

Spyder (Python 3.7)

File Edit Search Source Run Debug Consoles Projects Tools View Help

Editor - C:\Users\Admin\Desktop\Asbestos_Robot_Edited.py

Asbestos_Robot_Edited.py | Asbestos_Robot.py*

```
23
24 import pigpio
25 import time
26 import os
27
28 pi = pigpio.pi()
29
30 #Define GPIOs
31 LED_var1 = 2   #LED_Red
32 LED_var2 = 0   #LED_Yellow
33 LED_var3 = 7   #LED_Green
34
35 motor_output1 = 16
36 motor_output2 = 21 #Two directional movement
37
38 sensor_trig_output = 12 #Ultrasonic sensor
39 sensor_echo_input = 20
40
41 asbestos_input = 5 #Asbestos sensor
42
43 buzzer_output = 18 #Buzzer
44
45 power_ON = 22 #Switch
46
47 rotary_input = 24 #Optical rotary encoder
48
49 accumulated_rotary_counter = 0
50
```

Help

Source Console | Object

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the

Variable explorer | File explorer | Help

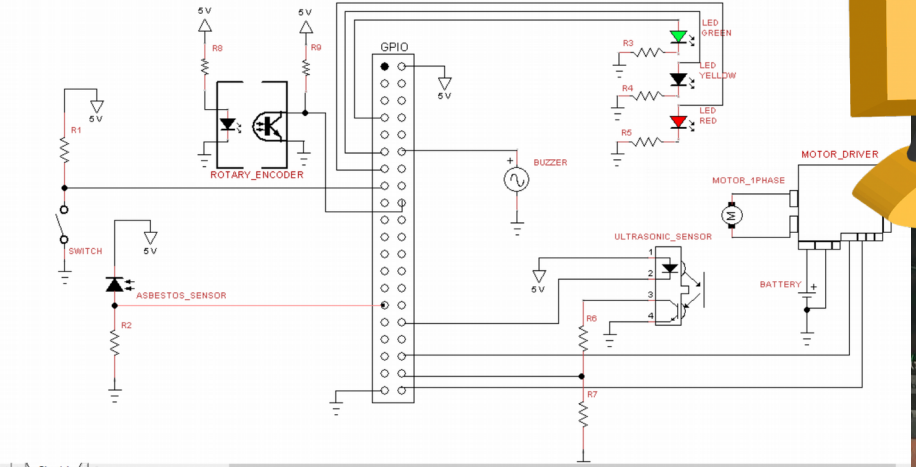
IPython console

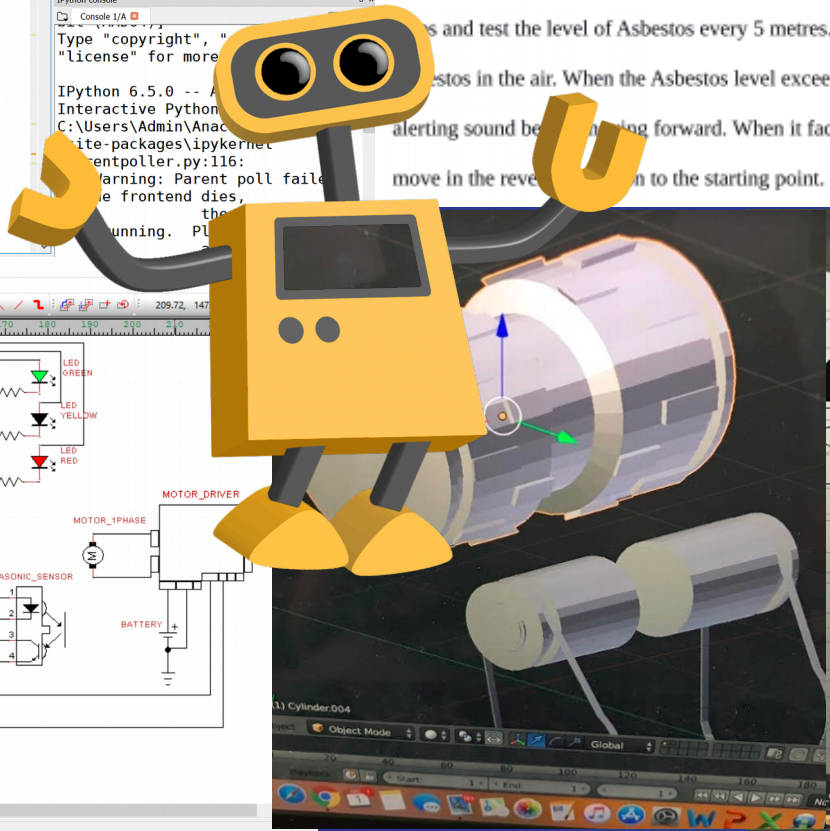
Console 1/A

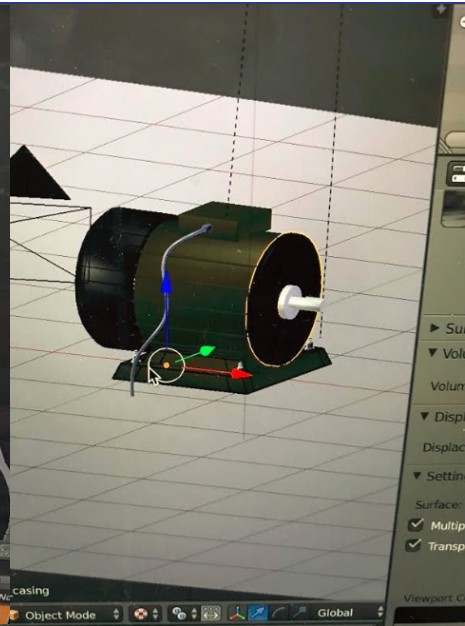
```
Type "copyright", "
"license" for more

IPython 6.5.0 -- A
Interactive Python
C:\Users\Admin\Anac
ite-packages\ipykernel
entpoller.py:116:
Warning: Parent poll faile
e frontend dies,
the
unning.  Pl
```

TinyCad - [Asbestos_Robot.sec]

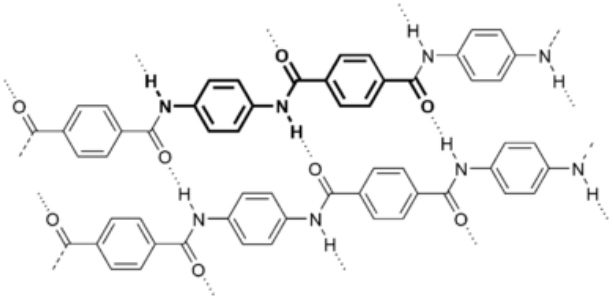
File Edit Library Special Options Window Help





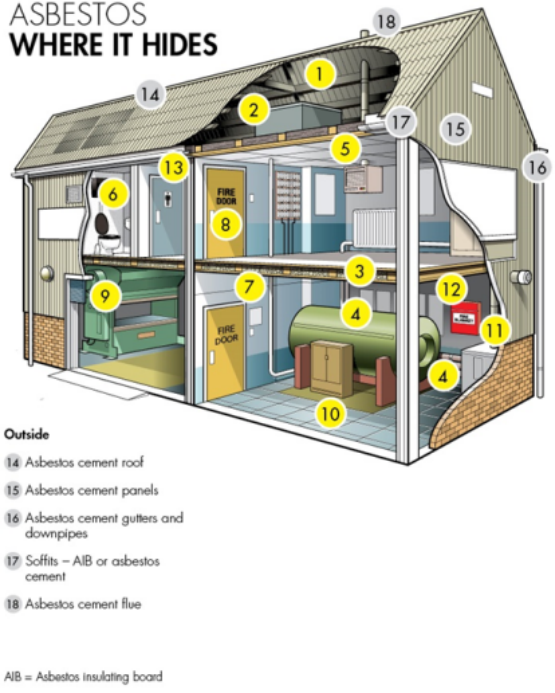


Background



Asbestos ($Mg_3Si_2O_5(OH)_4$

- Naturally occurring silicate minerals
- Known carcinogen and causes mesothelioma when inhaled
- Used as fireproofing material to construct buildings (floor, ceilings, pipes, insulation...)
- More than 107,000 people die in the world each year from exposure to asbestos.



Idea for our final project

Two-wheeled mobile robot

Working in indoor environment

Autonomous testing in buildings

For authority to assess environmental risk for public health

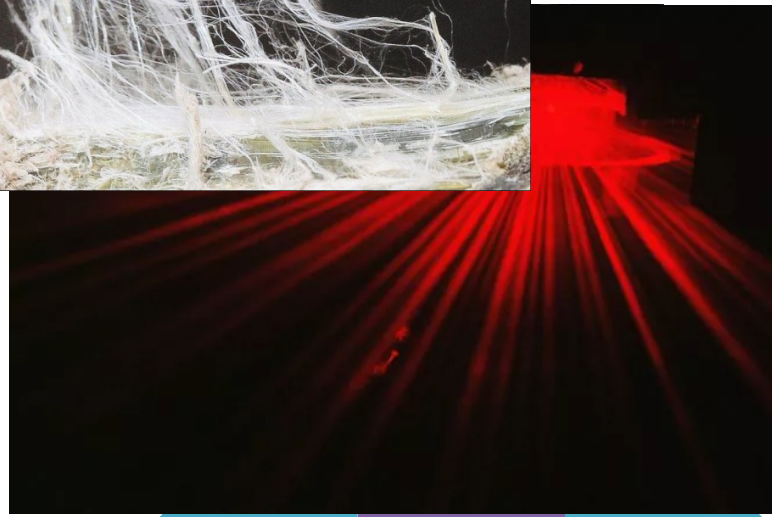
How to test Asbestos?

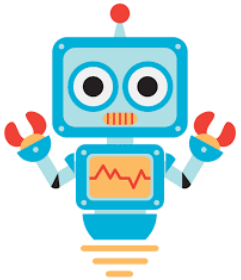
Shine a laser on a stream of airborne particles

Analyze the reflected light pattern

Pull the stream of airborne particles through a magnetic field

If the fibers align with the magnetic field





Functions of our robot design

1 | Modified Obstacle Detection

The robot will constantly detect obstacle;

- Multiple measurement is considered for better accuracy;
- Temporary obstacle will not hinder robot's performance

2 | Autonomous sampling and return

Systematic distance sampling and returning to the starting point

- Rotary encoder
- Measure travelled distance
- Adjusted behaviors

3 | Asbestos level testing

Inspection of Asbestos level

- Warning system: Alerting sound and LED Lights

Components

Raspberry Pi; Battery

Inputs:

Ultrasonic sensor

Asbestos sensor

Switch

Rotary encoder

Outputs:

Motor

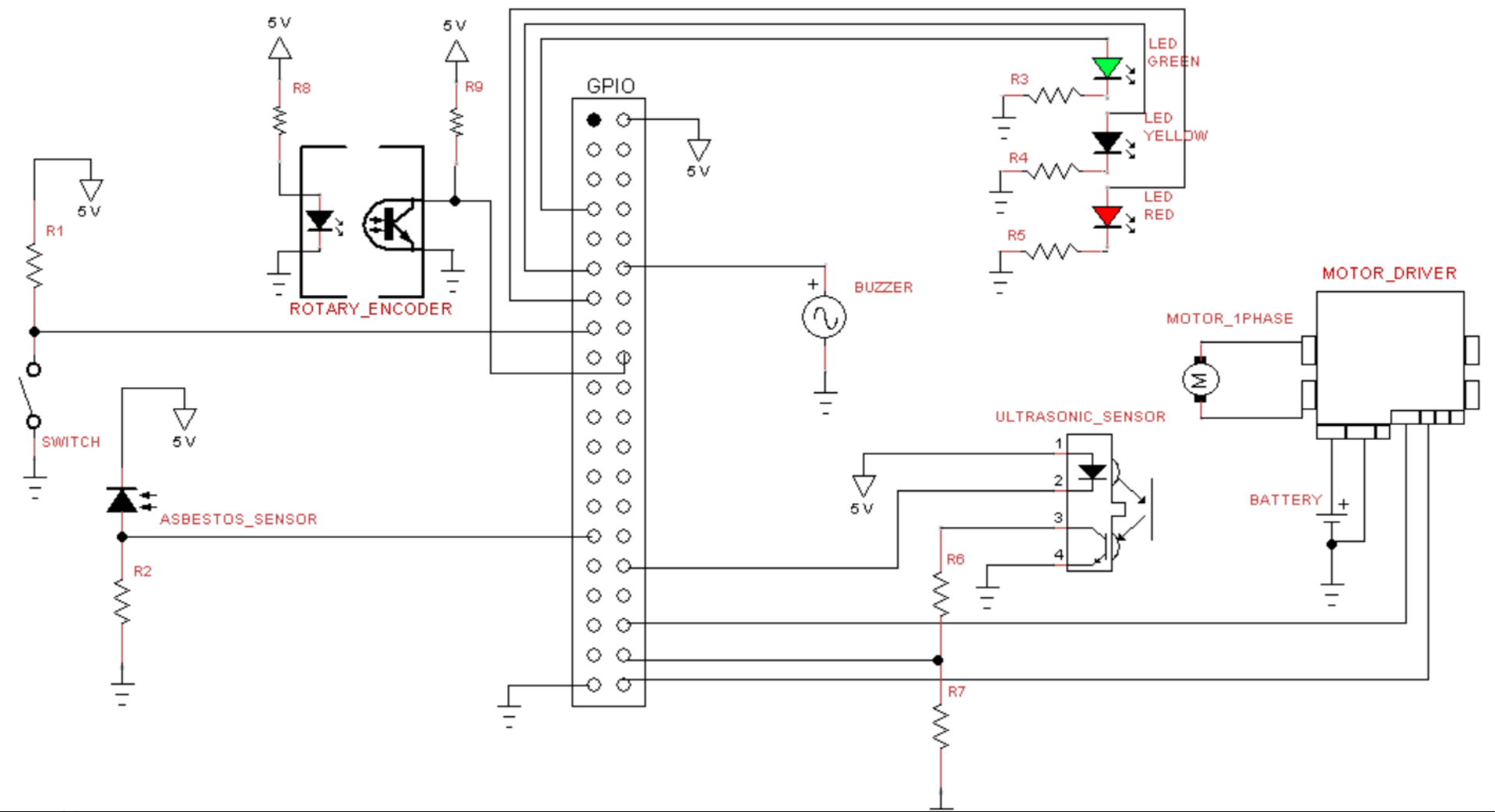
LED lights

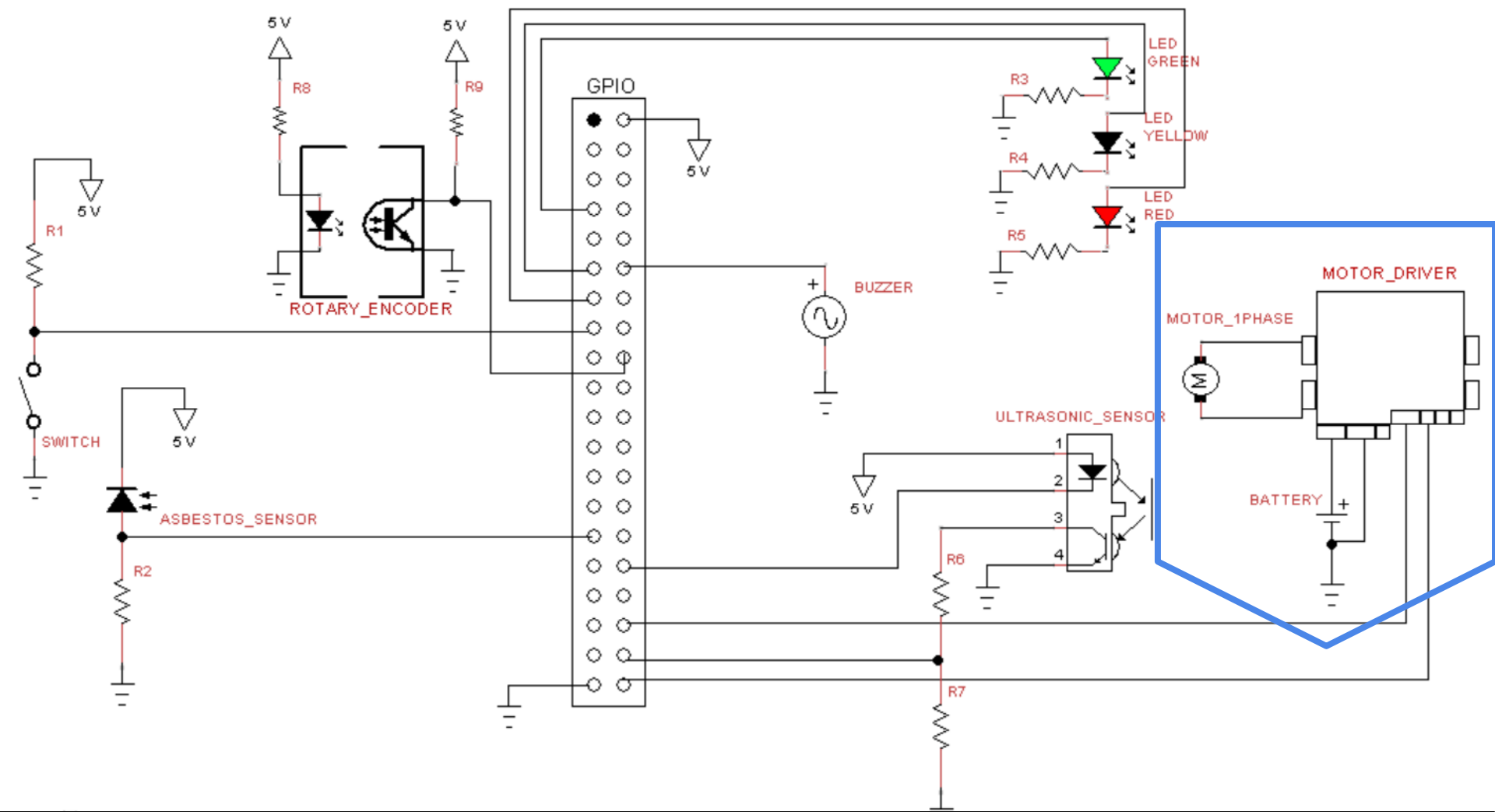
Buzzer

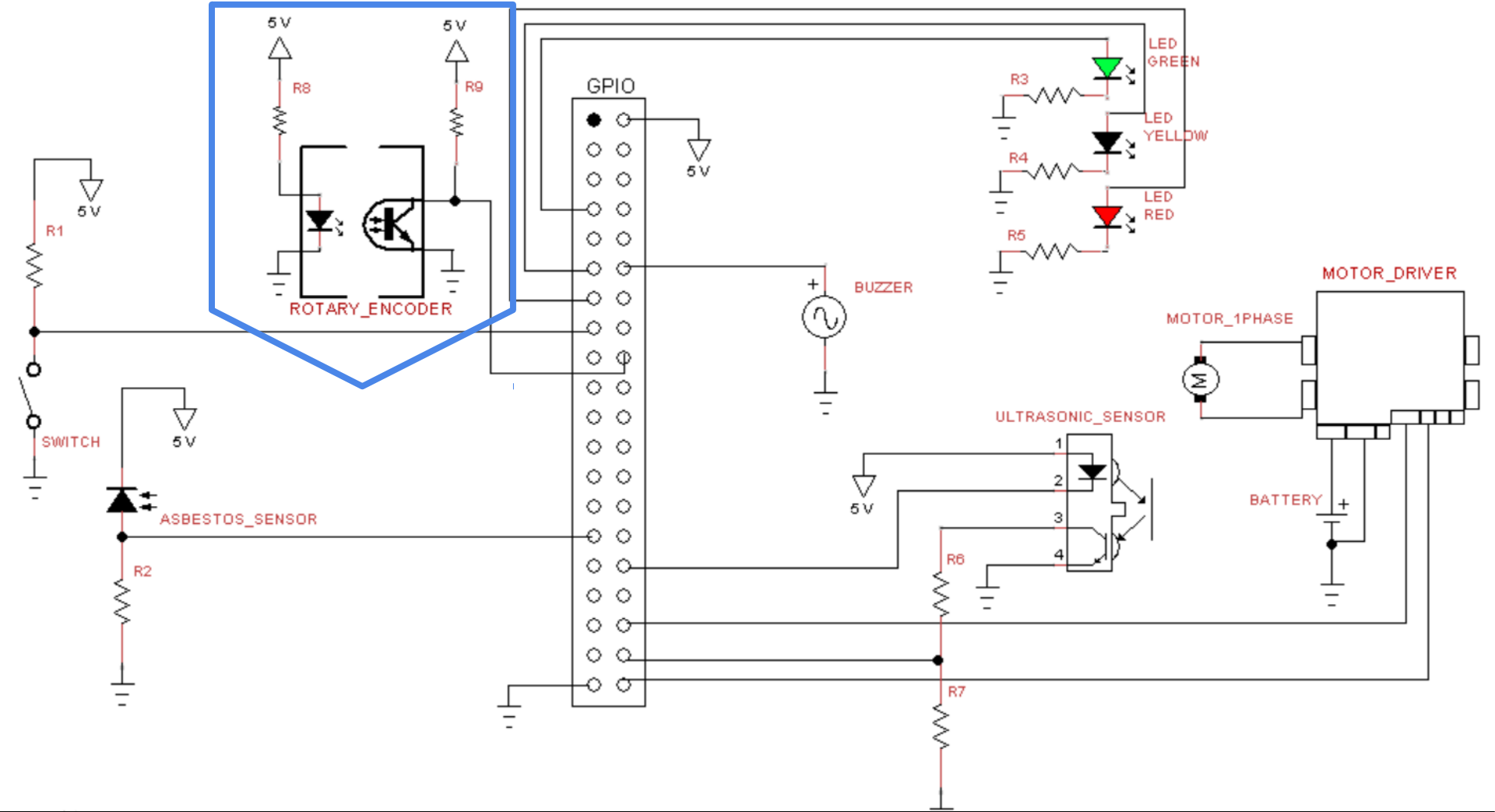
Schematic Diagram

 # TinyCAD

Components connected to the Raspberry Pi via pin GPIO.


5 V
R8
5 V
R9
GPIO
5 V
LED
GREEN
R3
LED
YELLOW
R4
LED
RED
R5
5 V
R1
ROTARY_ENCODER
BUZZER
MOTOR_DRIVER
MOTOR_1PHASE
M
SWITCH
5 V
ULTRASONIC_SENSOR
1
2
3
4
5 V
BATTERY
ASBESTOS_SENSOR
R2
R6
R7


5 V
R8
5 V
R9
GPIO
5 V
LED
GREEN
R3
LED
YELLOW
R4
LED
RED
R5
5 V
R1
ROTARY_ENCODER
BUZZER
MOTOR_DRIVER
MOTOR_1PHASE
M
SWITCH
5 V
ULTRASONIC_SENSOR
1
2
3
4
5 V
BATTERY
ASBESTOS_SENSOR
R2
R6
R7


5 V
5 V
R8
R9
ROTARY_ENCODER
GPIO
5 V
LED
GREEN
R3
LED
YELLOW
R4
LED
RED
R5
5 V
R1
BUZZER
MOTOR_DRIVER
MOTOR_1PHASE
M
SWITCH
5 V
ULTRASONIC_SENSOR
1
2
3
4
5 V
BATTERY
ASBESTOS_SENSOR
R2
R6
R7

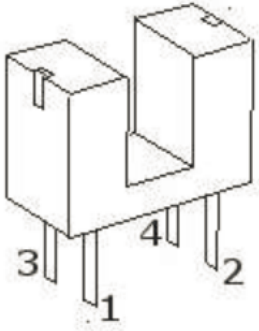
Rotary Encoder

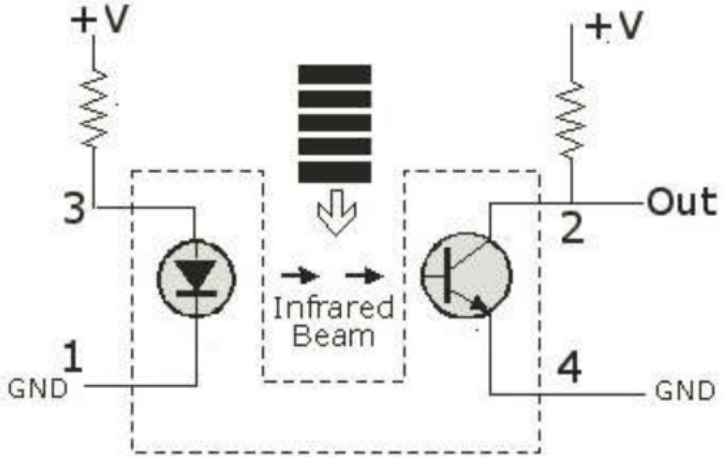
Optically measures how far something has moved

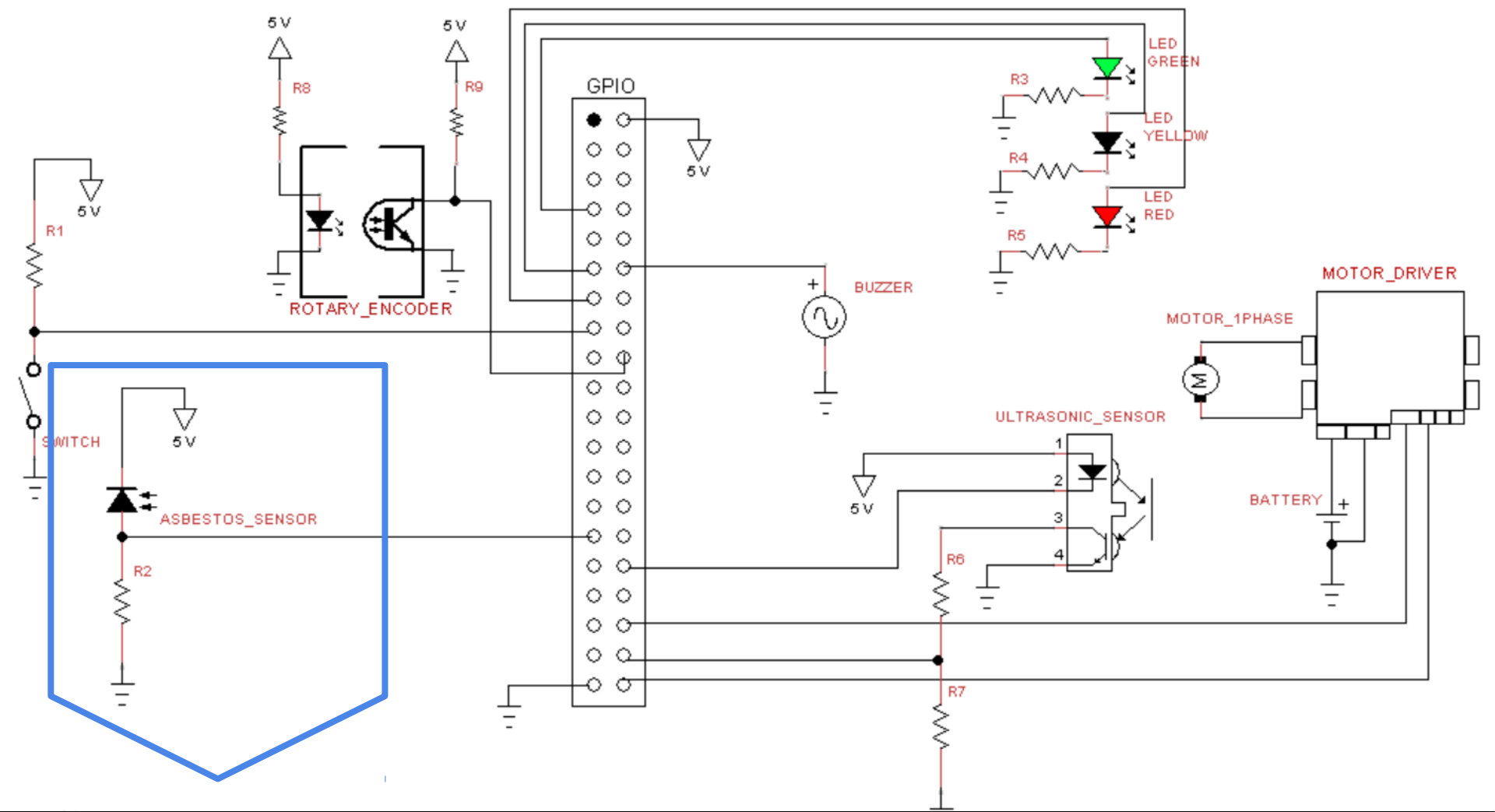
LED & a Photo Diode mounted facing each other

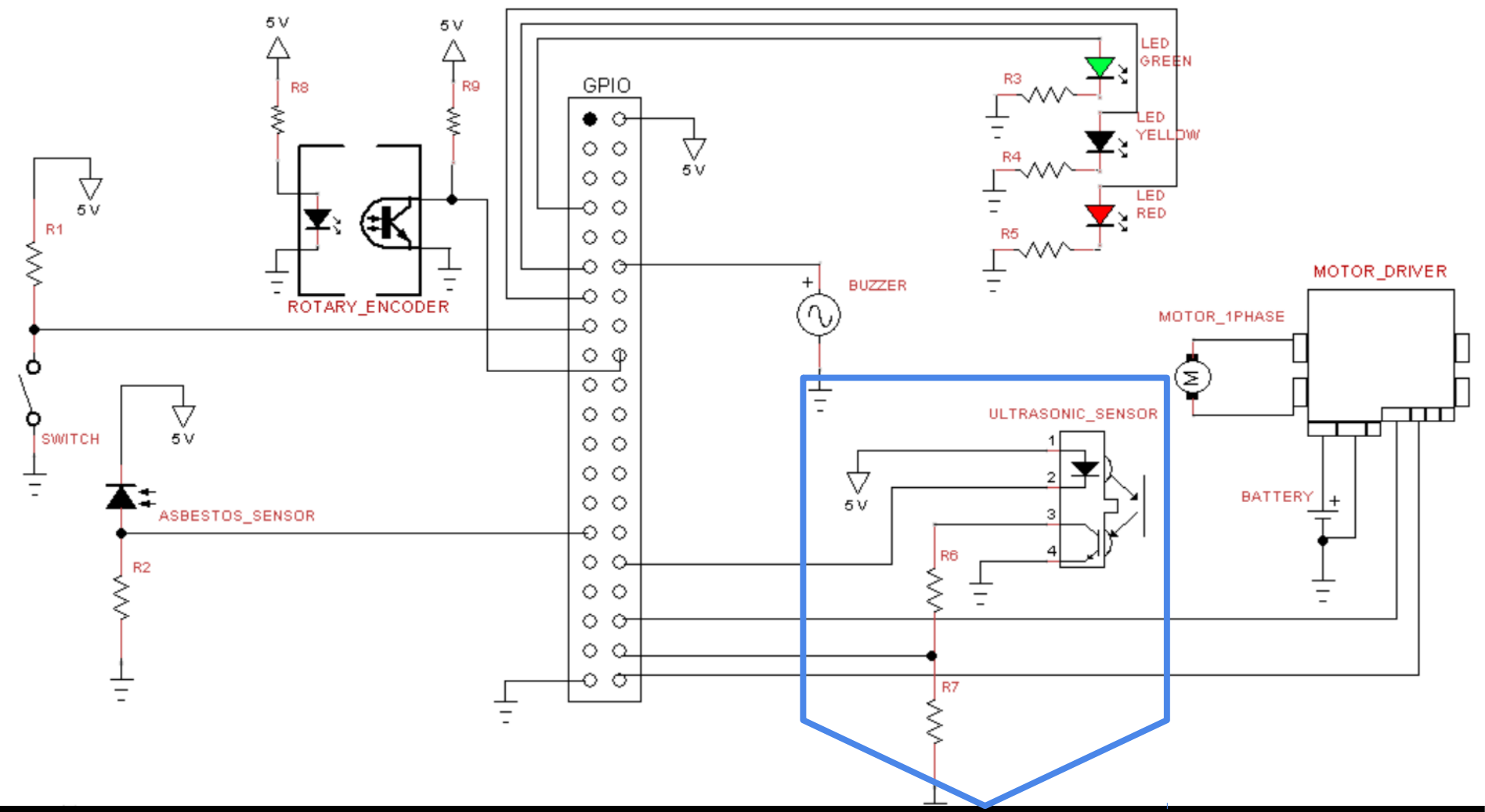
Light emitted by the IR LED is blocked because of the alternating slots of the encoder disc

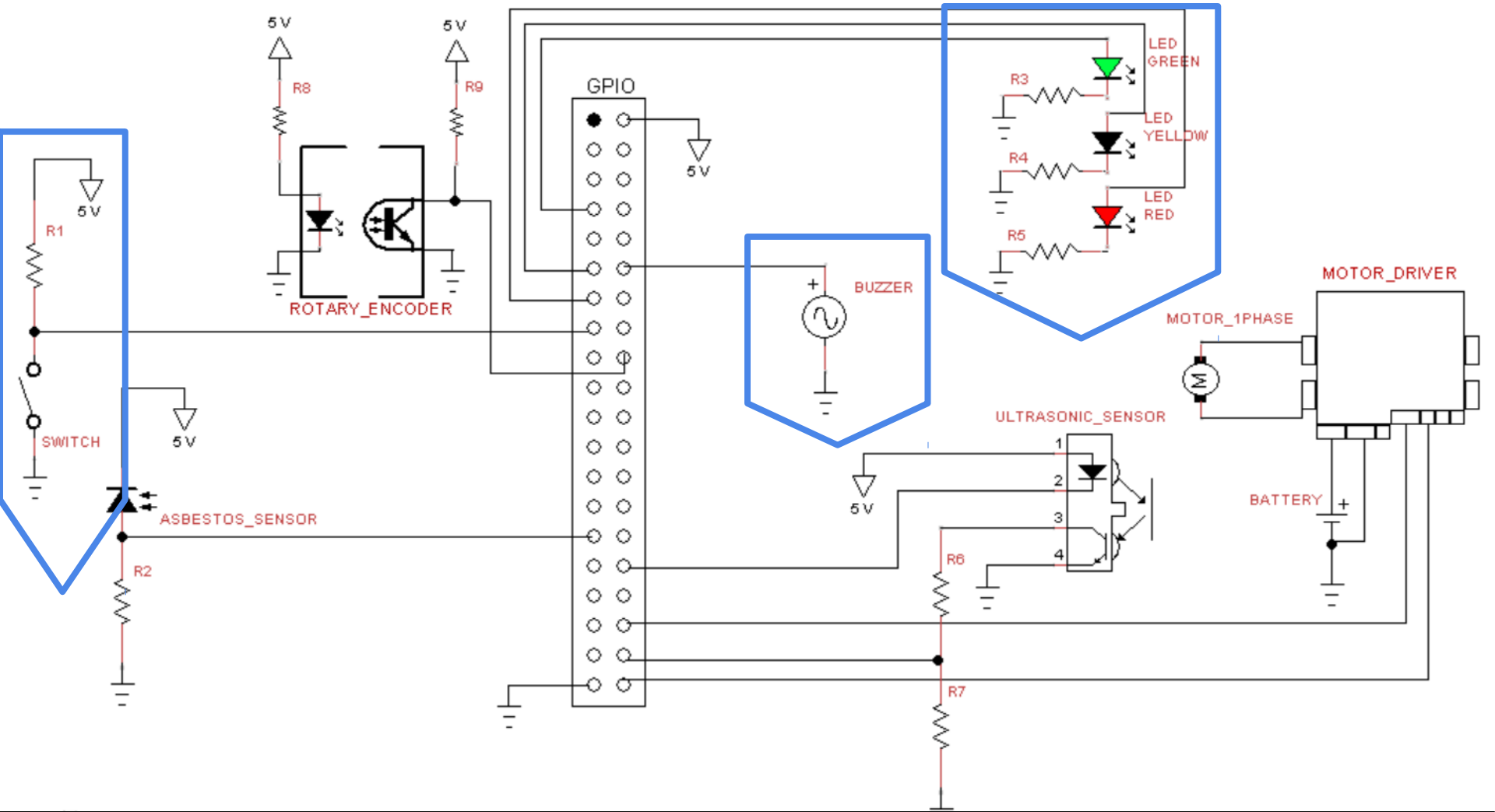
Conduction level changes

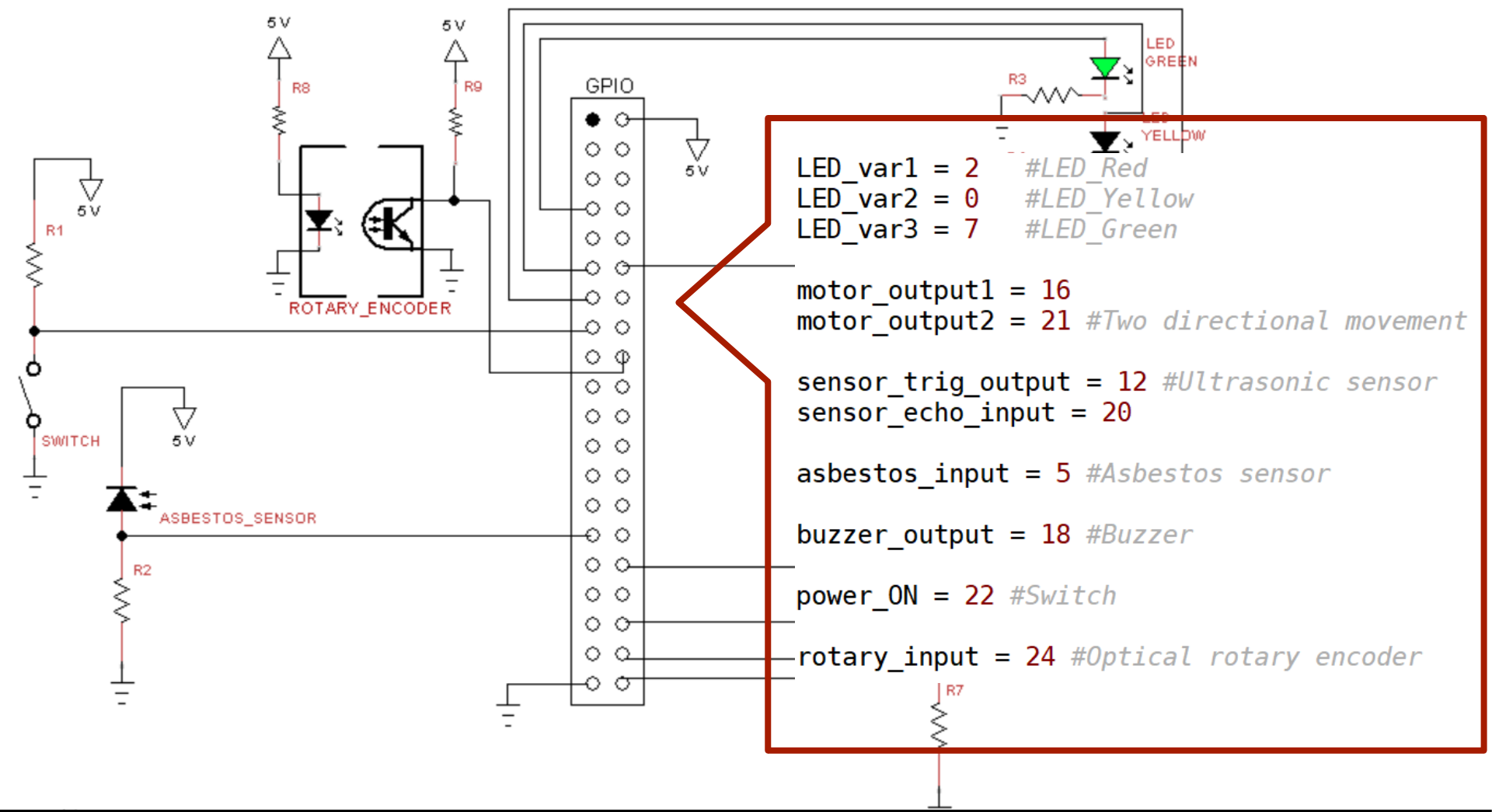





5 V
R8
5 V
R9
GPIO
LED
GREEN
R3
LED
YELLOW
R4
LED
RED
R5
5 V
R1
ROTARY_ENCODER
BUZZER
MOTOR_DRIVER
MOTOR_1PHASE
M
SWITCH
5 V
ULTRASONIC_SENSOR
1
2
3
4
5 V
BATTERY
ASBESTOS_SENSOR
R2
R6
R7


5 V
R8
5 V
R9
GPIO
LED
GREEN
R3
LED
YELLOW
R4
LED
RED
R5
5 V
R1
ROTARY_ENCODER
BUZZER
MOTOR_DRIVER
MOTOR_1PHASE
SWITCH
5 V
ULTRASONIC_SENSOR
1
2
3
4
5 V
BATTERY
ASBESTOS_SENSOR
R2
R6
R7


5 V
R8
5 V
R9
GPIO
5 V
LED
GREEN
R3
LED
YELLOW
R4
LED
RED
R5
5 V
R1
ROTARY_ENCODER
BUZZER
MOTOR_DRIVER
MOTOR_1PHASE
M
SWITCH
5 V
ULTRASONIC_SENSOR
1
2
3
4
5 V
BATTERY
ASBESTOS_SENSOR
R2
R6
R7




```
LED_var1 = 2    #LED_Red
LED_var2 = 0    #LED_Yellow
LED_var3 = 7    #LED_Green

motor_output1 = 16
motor_output2 = 21 #Two directional movement

sensor_trig_output = 12 #Ultrasonic sensor
sensor_echo_input = 20

asbestos_input = 5 #Asbestos sensor

buzzer_output = 18 #Buzzer

power_ON = 22 #Switch

rotary_input = 24 #Optical rotary encoder
```

Python Coding

Operation process:

Ultrasonic sensor - Constantly testing obstacle

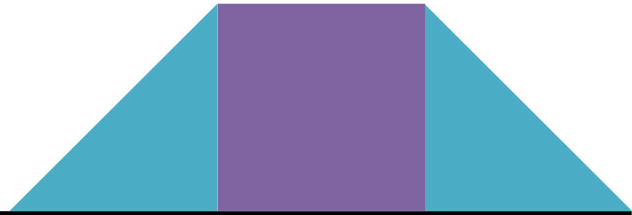
Measuring distance - Rotary sensor

Sampling - Stop every 5 metres and initiate asbestos sensor; different response according to results

Return to the starting point when finished

Functions:

1. ***process_sampling()***
2. ***process_asbestos_sensor()***
3. ***process_ultrasonic_sensor()***
4. ***detect_obstacle()***
5. ***power_off()***



Python Coding

Operation process:

Ultrasonic sensor - Constantly testing obstacle

Measuring distance - Rotary sensor

Sampling - Stop every 5 metres and initiate asbestos sensor; different response according to results

Return to the starting point when finished

```
while True:
    power_state = pi.read(power_ON)
#power on
    if power_state > 0:

#motor runs
        process_sampling()

#power off: turn off every active component
    if power_state == 0:
        power_off()
```

Python Coding

Operation process:

Ultrasonic sensor - ***Constantly testing obstacle***

Measuring distance - Rotary encoder

Sampling - Stop every 5 metres and initiate asbestos sensor; different response according to results

Return to the starting point when finished

```
def process_ultrasonic_sensor():
    print("Sending Ping..")

    distance = 0

    pi.write(sensor_trig_output, 0)
    time.sleep(0.1)

    ...

    for i in range(5): #repeated 5 times for better accuracy

        pi.write(sensor_trig_output, 0)
        time.sleep(0.1)


    ...

        distance += (end_time -  start_time) * 34300) / 2 / 2 # average

    return distance
```

Python Coding

Operation process:

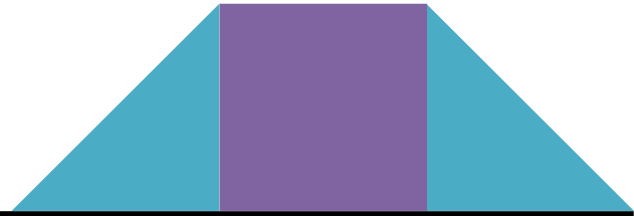
Ultrasonic sensor - ***Constantly testing obstacle***

Measuring distance - Rotary encoder

Sampling - Stop every 5 metres and initiate asbestos sensor; different response according to results

Return to the starting point when finished

```
def detect_obstacle()

    obstacle_distance = process_ultrasonic_sensor()
    print("Distance = ", obstacle_distance)
    time.sleep(0.5)

    if obstacle_distance < 50:
        obstacle = True
    else:
        obstacle = False
    return obstacle
```



Python Coding

Operation process:

Ultrasonic sensor - ***Constantly testing obstacle***

Measuring distance - Rotary encoder

Sampling - Stop every 5 metres and initiate asbestos sensor; different response according to results

Return to the starting point when finished

```
def temporary_stop():
    obstacle = detect_obstacle()
    if obstacle == False:
        return

    pi.write(motor_output1, 0)
    pi.write(motor_output2, 0)

    start = time.time()
    end = time.time()
    while end - start < 10:
        if detect_obstacle() == False
            return
        end = time.time()

    pi.write(motor_output1, 0)
    pi.write(motor_output2, 1) #robot returns

    ...
```

Python Coding

Operation process:

Ultrasonic sensor - Constantly testing obstacle

Measuring distance - Rotary encoder

Sampling - Stop every 5 metres and initiate asbestos sensor; different response according to results

Return to the starting point when finished

```
ROTARY_LIMIT = 10
#sampling distance: 10 * 0.5m (circumference of wheel) = 5m

def process_sampling():
    pi.write(motor_output1, 1)
    pi.write(motor_output2, 0)

    rotary_counter = 0

    while rotary_counter < ROTARY_LIMIT:
        while pi.read(rotary_input) == 0:
            temporary_stop()
        rotary_counter += 1
        accumulated_rotary_counter += 1

    pi.write(motor_output1, 0)
    pi.write(motor_output2, 0)
```

Python Coding

Operation process:

Ultrasonic sensor - Constantly testing obstacle

Measuring distance - Rotary encoder

Sampling - Stop every 5 metres and initiate asbestos sensor; ***different response according to results***

Return to the starting point when finished

```
def process_asbestos_sensor():

asbestos_state = pi.read_adc(asbestos_input)

#High level
if asbestos_state >= 1:
    pi.write(LED_var1, 1)
    pi.write(LED_var2, 1)
    pi.write(LED_var3, 1)
    pi.write(buzzer_output, 1)
    time.sleep(2)
    ...

#Medium level
elif asbestos_state < 1 and asbestos_state >= 0.04:
    ...

#Low level
elif asbestos_state < 0.04:
    ...
```

Python Coding

Operation process:

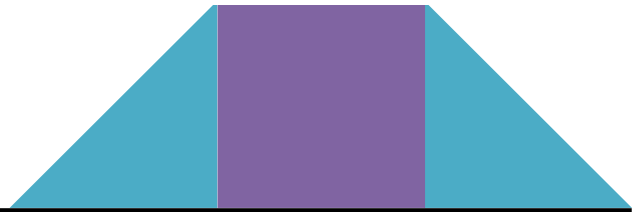
Ultrasonic sensor - Constantly testing obstacle

Measuring distance - Rotary encoder

Sampling - Stop every 5 metres and initiate asbestos sensor; different response according to results

Return to the starting point when finished

```
# Robot returns when obstacle lasts more than 10s
start = time.time()
end = time.time()
while end - start < 10:
    if detect_obstacle() == False:
        return
    end = time.time()

pi.write(motor_output1, 0)
pi.write(motor_output2, 1)
```



Python Coding

Operation process:

Ultrasonic sensor - Constantly testing obstacle

Measuring distance - Rotary encoder

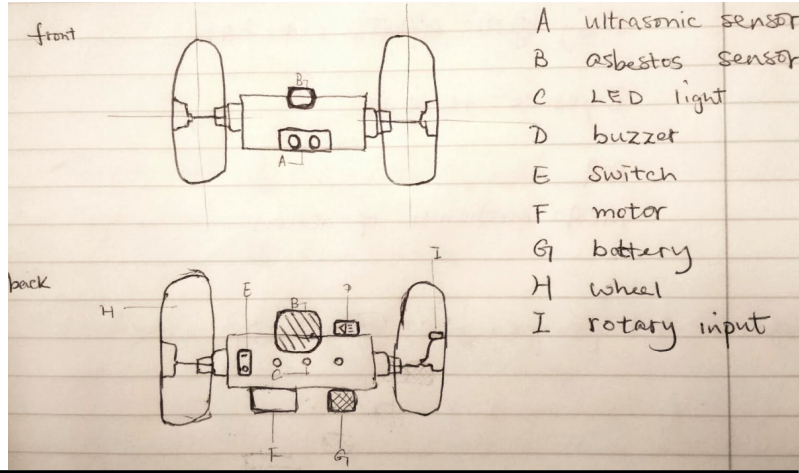
Sampling - Stop every 5 metres and initiate asbestos sensor; different response according to results

Return to the starting point when finished

```
pi.write(motor_output1, 0)
pi.write(motor_output2, 1) #robot returns

while accumulated_rotary_counter > 0:
    while pi.read(rotary_input) == 0:
        pass
    accumulated_rotary_counter -= 1

pi.write(motor_output1, 0)
pi.write(motor_output2, 0)

os.exit(0)
#The robot returns to its original position
```

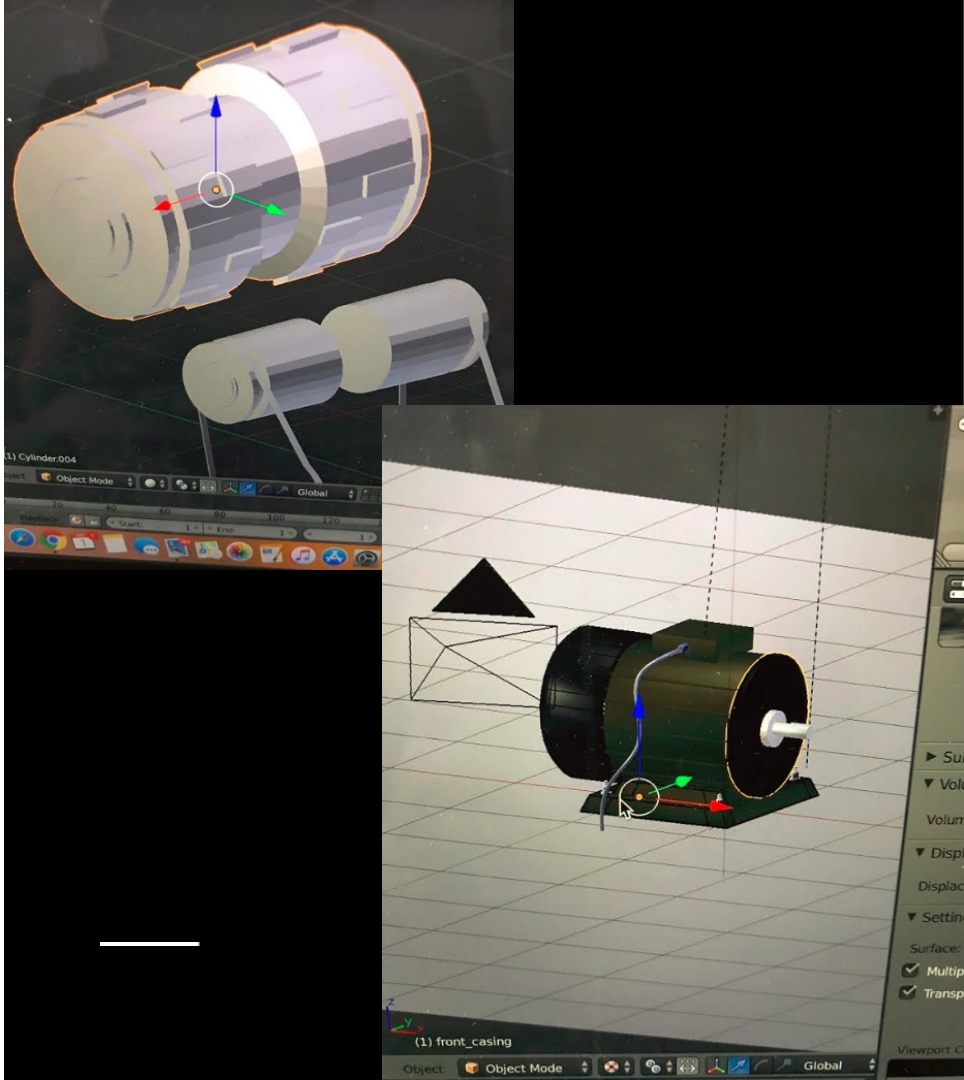
Graphic Design

1. **Two wheels, one with rotary encoder attached**
2. **Asbestos sensor and ultrasonic sensor attached on the main body in the front**
3. **Response area for LEDs and buzzer**
4. **Other components inside the body**



Blende

...for our robot graphic design



The END

Thank you for Reto’s and tutor’s help for our project and all of your attention!