|  |  |
| --- | --- |
| ***2019-2020 EECS 201 LAB PROJECT***  **Obstacle Detector Device** | |
| **STUDENTS NAME** | **RESPONSIBILITY** |
| **ENES BURAK KARAÇUKA**  **217560592** | **HARDWARE DESİGN**  **PREPARİNG TO REPORT**  **IDEA ABOUT POSSIBLE APPLICATIONS-CHALLENGES-IMPROVEMENT.**  **HELPİNG İN SOFTWARE DESİGN.** |
| **MURAT FİDAN**  **217461702** | **SOFTWARE DESİGN**  **PREPARİNG TO REPORT**  **IDEA ABOUT POSSIBLE APPLICATIONS-CHALLENGES- IMPROVEMENT.**  **HELPİNG İN HARDWARE DESİGN.** |
| **HÜSEYİN HASILCI** | **SOFTWARE DESİGN**  **PREPARİNG TO REPORT**  **IDEA ABOUT POSSIBLE APPLICATIONS-CHALLENGES- IMPROVEMENT.**  **HELPİNG İN HARDWARE DESİGN.** |

**INTRODUCTION:**

Firstly, members of groups meet together to talk and share ideas about project and decided which member is responsible about which part of project. All members of groups started to work to complete project in their own areas of project which was decided by group members. Enes Burak Karaçuka worked on hardware system of the project. Murat Fidan and Hüseyin Hasılcı worked on software system of the project. Moreover, all members of groups found ideas about improvement of project, possible applications and challenges and everyone in the group made a presentation about the area they are responsible for the project.

**GOALS:**

We want to build up an obstacle detector device. It helps to human life about finding an obstacle or warning about obstacle. We developed a project where the lights change and the sound increases as we get closer to the barrier.

**Materials Used and Description:**

|  |  |
| --- | --- |
| **ULTRASONIC SENSOR** | It consists of 2 episodes. the sound wave sent from one section hits the obstacle and comes to the other section. this process measures the distance between the device and the obstacle. |
| **LEDS** | It warns the user based on the value given by the ultrasonic sensor. |
| **BUZZER** | It alerts the user by making noise as the distance between the obstacle and the ultrasonic sensor decreases |
| **CABLES** | It provides the link between materials. |
| **RESISTANCE** | It adjusts the lighting level of the LEDs and the intensity of the buzzer's sound |
| **Adafruit METRO M0 Express** | It's the brain of the circuit. The tasks of each sensor are defined here. The code is loaded here. |

**Challenges:**

While we doing this application we struggle in the code part because we can’t try properly. Also it’s so hard to find information about the code and there is lack of data related to this devices.

**How can you make it Better?:**

We can add camera in the application. Buzzer will ring louder when we close to a plane and camera will show the area. In camera it will measure the distance and show us the output in the screen. With this we can see where we are.

**Possible Applications:**

We can use this application to measure specific distances. Also we can use this application as an park sensor with screen and camera. Or we can make music with this application. For every distance we can change the frequency and sing.

**Conclusion:**

In conclusion, we make this obstacle detector device to detect anything and find the distance between the device. Also we think how we can make it better this device and possible applications.

**Hardware Design:**

**SOFTWARE:**

import time  
import board  
import adafruit\_hcsr04  
import digitalio  
  
class ParkSensor:  
 def \_\_init\_\_(self):  
 self.green\_led = digitalio.DigitalInOut(board.D7)  
 self.yellow\_led = digitalio.DigitalInOut(board.D8)  
 self.red\_led = digitalio.DigitalInOut(board.D9)  
 self.buzzer = digitalio.DigitalInOut(board.D11)  
 self.sonar = adafruit\_hcsr04.HCSR04(trigger\_pin=board.D4, echo\_pin=board.D2)  
 self.green\_led.direction = digitalio.Direction.OUTPUT  
 self.yellow\_led.direction = digitalio.Direction.OUTPUT  
 self.red\_led.direction = digitalio.Direction.OUTPUT  
 self.buzzer.direction = digitalio.Direction.OUTPUT  
 self.red\_led.value = True  
 time.sleep(0.5)  
 self.red\_led.value = False  
 self.yellow\_led.value = True  
 time.sleep(0.5)  
 self.yellow\_led.value = False  
 self.green\_led.value = True  
 time.sleep(0.5)  
 self.green\_led.value = False  
 self.buzzer.value = True  
 time.sleep(0.5)  
 self.buzzer.value = False  
 print("Obstacle Detecter Decive is Working")  
 self.lets\_start()  
 def get\_distance(self):  
 return self.sonar.distance  
 def ligth\_red\_led(self):  
 self.red\_led.value = True  
 time.sleep(1)  
 self.red\_led.value = False  
 def ligth\_yellow\_led(self):  
 self.yellow\_led.value = True  
 time.sleep(1)  
 self.yellow\_led.value = False  
 def ligth\_green\_led(self):  
 self.green\_led = True  
 time.sleep(1)  
 self.green\_led.value = False  
 def activate\_buzzer(self):  
 self.buzzer.value = True  
 time.sleep(1)  
 self.buzzer.value = False  
 def lets\_start(self):  
 while True:  
 try:  
 if(120<int(self.get\_distance())<=200):  
 self.ligth\_green\_led()  
 print("You are safe.")  
 elif(50<int(self.get\_distance())<=120):  
 self.ligth\_yellow\_led()  
 print("Slow down your speed")  
 elif(25<int(self.get\_distance())<=50):  
 self.ligth\_red\_led()  
 print("DANGER TOO CLOSE")  
 elif(int(self.get\_distance())<=25):  
 self.ligth\_red\_led()  
 self.active\_buzzer()  
 print("The device stopped! calling the emergency services.")  
 break  
 else:  
 self.ligth\_green\_led()  
 self.ligth\_yellow\_led()  
  
 except:  
 print("Distance Error!")

park\_sensor\_object = ParkSensor()

**Describe Of Functions:**

* **\_\_init\_\_:** the circuit elements are defined in this function(LEDs, buzzer, sonar). This function is called automatically as soon as an object belonging to this class is created and provides the connection between the code and the circuit elements.
* **get\_distance:** this function return the distance between the obstacle and the sensor as measured by the ultrasonic sensor.
* **ligth\_red\_led, ligth\_green\_led, ligth\_yellow\_led:** these functions enable the LEDs to light up.
* **active\_buzzer:** this function activates the buzzer, enabling the user to be alerted by voice.
* **lets\_start:** this function is the main function. all the above functions are called by this function when they are needed.

**DIAGRAM:**

**İNİTAL STATE(\_\_init\_\_)**

**lets\_start()**

**ligth\_red\_led and active\_buzzer**

**LAST STATE()**

**Finish**

**SECOND STATE(ligth\_green\_led)**

**FİFTH STATE(ligth\_green\_led and ligth\_yellow\_led)**

**THİRD STATE(ligth\_yellow\_led)**

**FOURTH STATE(ligth\_red\_led)**

**NEEDED ACTION FOR SECOND STATE:** ultrasonic sensor reads a distance larger than 120 cm and smaller than 200 cm.

**NEEDED ACTION FOR THIRD STATE:** input reading is less than or equal to 120 cm and larger than 50 cm.

**NEEDED ACTION FOR FOURTH STATE:** input reading is lower than or equal to 50 and above 25 cm.

**NEEDED ACTION FOR FIFTH STATE:** the reading is higher than or equal to 200cm.

**NEEDED ACTION FOR LAST STATE:** the distance is equal to 25 or lower.

**References:**

<https://learn.adafruit.com/welcome-to-circuitpython/what-is-circuitpython>

<https://learn.adafruit.com/using-piezo-buzzers-with-circuitpython-arduino/circuitpython>