TED ÜNİVERSİTESİ

EE202- CIRCUIT THEORY II

Term Project Report

EARTHQUAKE ALERTER

Group Number: 24

Participants

İlknur Deniz Baloğlu **42127124856**
Mehmet Çınar **13837189268**

EARTHQUAKE ALERTER

Overview of Project

A very simple earthquake shake sensor that builds everywhere and is applicable. It is a project made by university students and it aims to show how useful and practicable this sensor is, saving many people's lives. Briefly, we live in an earthquake zone and the statistics show that there is always a risk so that especially this device will warn us against any shake.

Our Goals

1. To ensure this device is in every home and everywhere
2. To detect even very small earthquake jolts
3. To make adjustable sensitivity levels
4. To be ready for any damage of earthquake at any time
5. To show people how easy to prepare an earthquake sensor in their homes

Description

Due to the fact that our country is an earthquake zone, we encounter this natural event at certain intervals. So that, we wanted to design this project by considering our citizens and country to survive with the least damage during the earthquake. This project, on which we set off, will be mounted on the wall and we will make sure that it even informs us about the tremors we cannot feel. Moreover, at the end of this project, we will show all details about code, models, graphs, and visuals. In detail materials that we will use in this alert system as follows:

1. Arduino Pro Micro
2. MPU6050 IMU
3. Buzzer
4. LED
5. Perforated Pertinax
6. Resistor

MILESTONES

First Step

For the first step we will build a portable easily accessible circuit with ceap elements.

Future Build

We are looking forward to altering this system to be more efficient and nature friendly in terms of energy source, so a solar panel will do the job. Other than that, we are thinking that being able to check the data with a device like a phone will make the system more accessible and useful, so a bluetooth or wi-fi adapter will come in handy for the future build.

PHASE of THE PROJECT

1. As we have already stated in the proposal, we started to gather the materials together. Due to the situation, we had problems about timing for materials but eventually we got them. In the following pictures, materials are shown.

MPU6050 IMU

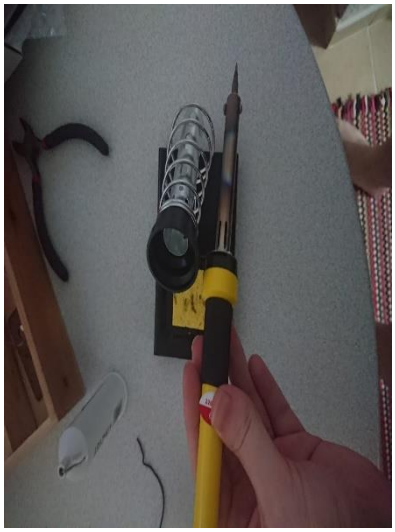
Arduino Pro Micro

LED

Resistors

Buzzer

Multimeter



Soldering Iron

Nippers

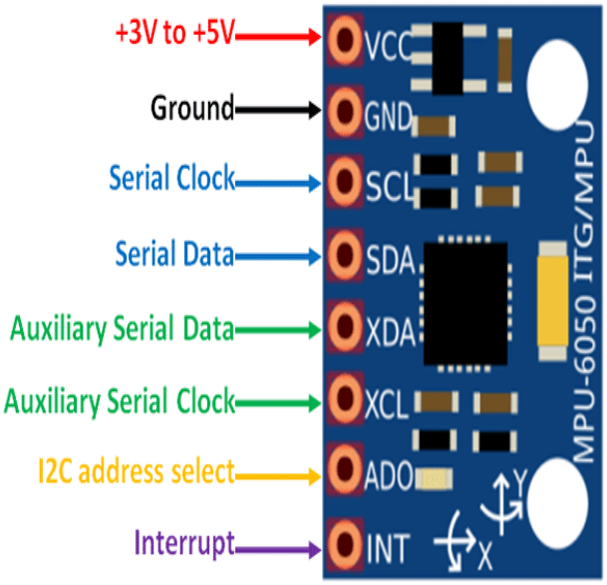
Then, we can explain the aims of some materials that we used as follows:

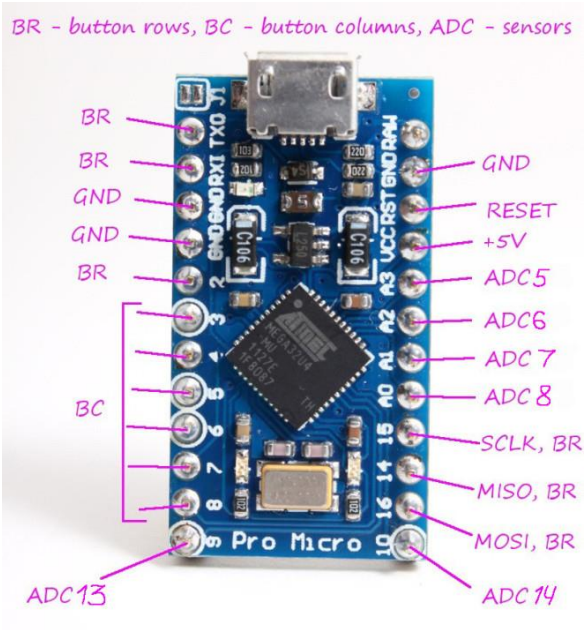
- **MPU6050 IMU:** It is the main product of our project that measures quake by using its accelerometer feature.
- **Arduino Pro Micro:** This material is the center of the system. We can connect the power and install the code with this element.
- **Buzzer:** This element will show to user that the system is working. When system worked, buzzer warn the user by making sound.
- **LED:** The job is the material is the same as the purpose of buzzer. When system worked, LED warn the user by producing light.
- **Perforated Pertinax:** We used it for soldering prototype circuit components
- **Resistor**

Auxiliary Materials

- **Multimeter**
- **Soldering Iron**
- **Nippers**

Datasheets of Components

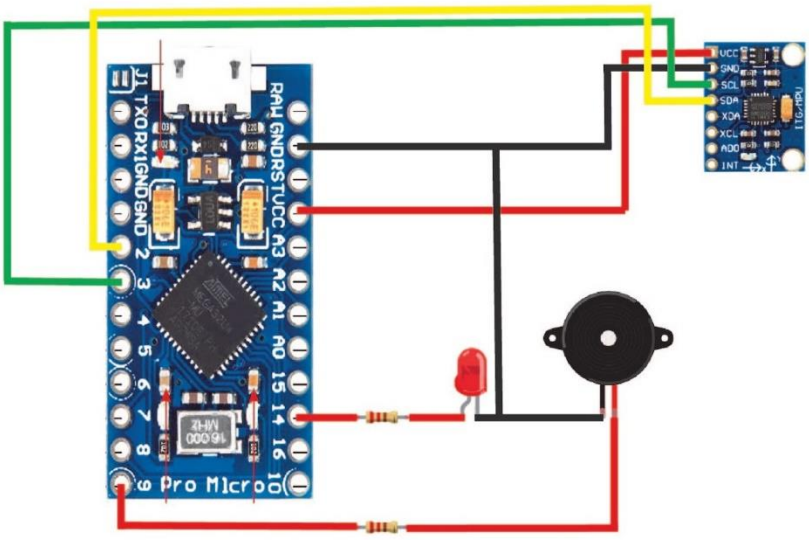




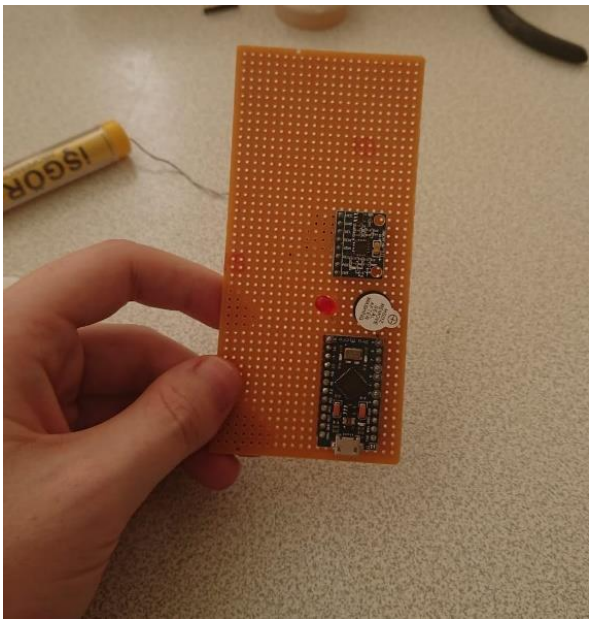
MPU6050 IMU **Arduino Pro Micro**

2. In this step, we determined the inputs of the accelerometer and the Arduino. After that we soldered the inputs in accordance with the circuit we will constitute.

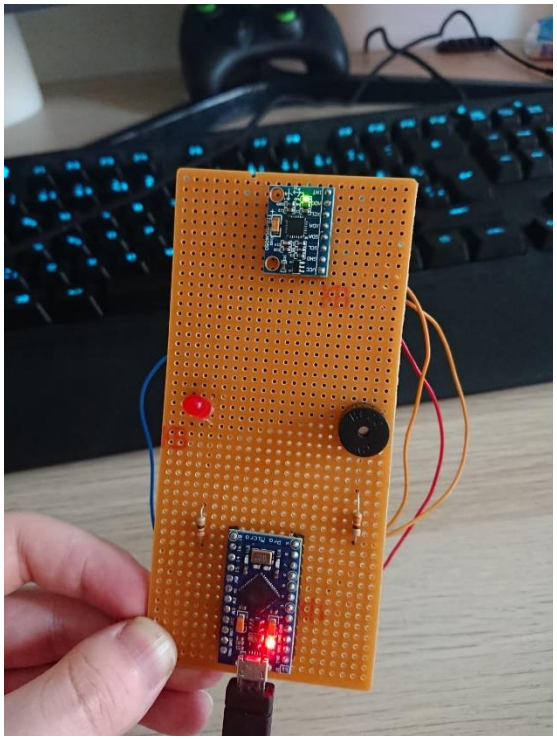
3. At the same time we started to design the schematic model and code for our circuit. The schematic model shown in below:

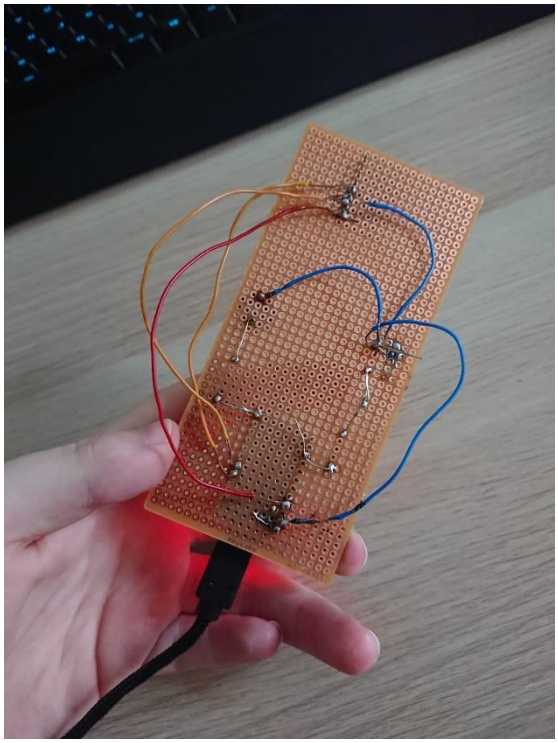


4. With this phase, we started to establish the circuit by applying buzzer, led, Arduino and accelerometer.



5. In this phase we controlled and observed that there is a current flow in the circuit which we designed. This means there is no mistake in our circuit.



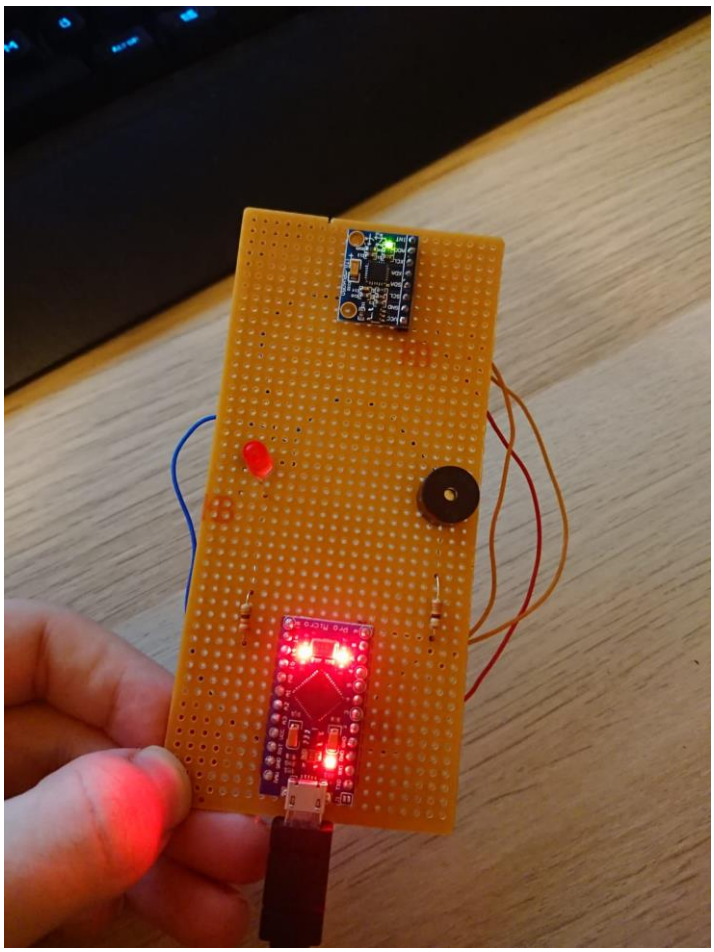


6. At this step of the our project, we prepared and we explained the code as follows:

```
#include <Arduino.h>
#include <TinyMPU6050.h> // declaring the labraries for the functions

float pos_offset = 10;
float neg_offset = -10;
long angle_x, angle_y, angle_z, offset_x, offset_y, offset_z; // Declaretion of parameters for MPU6050 IMU
MPU6050 mpu (Wire);


void setup()
{

  // Initialization
  mpu.Initialize();

  // Calibration
  Serial.begin(9600);
  Serial.println("=====================================");
  Serial.println("Starting calibration...");
  Serial.println("Calibration complete!");
  Serial.println("Offsets:");// Gathering the inputs that MPU got for all dimensions
  Serial.print("AccX Offset = ");
  Serial.println(mpu.GetAccXOffset());
  Serial.print("AccY Offset = ");
  Serial.println(mpu.GetAccYOffset());
  Serial.print("AccZ Offset = ");
  Serial.println(mpu.GetAccZOffset());
  Serial.print("GyroX Offset = ");
  Serial.println(mpu.GetGyroXOffset());
  Serial.print("GyroY Offset = ");
  Serial.println(mpu.GetGyroYOffset());
  Serial.print("GyroZ Offset = ");
  Serial.println(mpu.GetGyroZOffset());
  pinMode(14,OUTPUT); // Output pins of Arduino
  pinMode(9, OUTPUT);
  digitalWrite(14, LOW);
  digitalWrite(9, LOW);

  delay(1000);
  for(int i=0; i<200;i++)
  {
   mpu.Execute();// Execute the functions
  offset_x = mpu.GetAngX();
  offset_y = mpu.GetAngY();
  offset_z = mpu.GetAngZ();
  }
```

```
  Serial.print("offset_x = ");
  Serial.print(offset_x);
  Serial.print("  /  offsetY = ");
  Serial.print(offset_y);
  Serial.print("  /  offsetZ = ");
  Serial.println(offset_z);

}

void loop()
{

 for(int i=0; i<5;i++){
  mpu.Execute(); //Executing the angle functions
  angle_x = mpu.GetAngX();
  angle_y = mpu.GetAngY();
  angle_z = mpu.GetAngZ();
 }

  Serial.print("AngX = ");
  Serial.print(angle_x - offset_x);
  Serial.print("  /  AngY = ");
  Serial.print(angle_y - offset_y);
  Serial.print("  /  AngZ = ");
  Serial.println(angle_z - offset_z);


 if ( pos_offset < angle_x - offset_x || neg_offset > angle_x - offset_x || pos_offset < angle_y - offset_y || neg_offset > angle_y - offset_y || pos_offset < angle_z - offset_z || neg_offset > angle_z - offset_z)
 {// if any change happened at angles axecute this

  for(int i=0; i<50; i++)
  {// Give the output to the pins constantly to execute the fuctions of elements
  digitalWrite(14,HIGH);
  digitalWrite(9,LOW);
  delay(50);
  digitalWrite(14,LOW);
  digitalWrite(9,LOW);
  delay(50);
  }
  delay(5000);
  mpu.Execute();//if code executed succesfully adjust the new dimensions
  offset_x = mpu.GetAngX();
  offset_y = mpu.GetAngY();
  offset_z = mpu.GetAngZ();

 }
}
```

7.In final, we organized to the code according to type of Arduino and entry in Arduino IDE program. After that, we loaded it into Arduino.



REFERENCES

https://www.arduino.cc/

https://www.arduino.cc/en/reference/libraries