HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF MECHANICAL ENGINEERING

The Almighty Ruler

PROJECT REPORT

COURSE: INTRODUCTION TO ENGINEERING

Group:
The Deadline Kids

Lecturer:
Luu Thanh Tung

Spring 2022

Team Members

Member	ID Number	Major	Role
Luong Vinh Trung Kien	2153496	Mechanical Engineering	CAD designer
Doan Ngoc Tan	2153779	Robotics Engineering	Fund manager, report writer
Tran Thai Duc Duy	2152472	Mechatronics Engineering	Circuit engineer
Nguyen Quy Hung	2152101	Robotics Engineering	Presentation maker, presenter
Le Thi Thuy Hang	2052460	Computer Science	Team leader (participate in every task)

Contents

List of Figures

1 Acknowledgement

We would like to share our sincere gratitude to all those who help us in completion of this project. During the work we face many challenges due to our lack of knowledge and experience but these people help us get over all the difficulties.

We would also like to extend our thanks to Mr.Luu Thanh Tung for his guidance, advice and assistance in keeping our progress on schedule.

Finally, we are grateful to our families for their support and encouragement, which has inspired us to face and overcome all challenges.

2 Introduction

2.1 Arduino

Arduino is an open source electronics platform based on user-friendly hardware and software. Arduino consists of both a physically programmable circuit board (commonly known as a microcontroller) and programming software or IDE (Integrated Development Environment) that runs on a PC and is used to create and transfer PC code to the circuit board. This can be done using the Arduino programming language and the Arduino software (IDE). Unlike other programmable boards, the Arduino doesn't require any other equipment (called a software developer) to upload code to the board, one can essentially utilize a USB connection. Also, the Arduino IDE uses a rearranged version of C++, making it easier to figure out how to program. In short, Arduino make the functions of the micro-controller into a more accessible package.
There are various types of Arduino boards used for different purposes and projects. One of the most popular boards in the Arduino family, the Uno, is an exceptional option for beginners [1].

Figure 1: An Arduino UNO board

There are different types of Arduino boards for different purposes. But all boards share most of the following components in common. Starting clockwise from the top center:

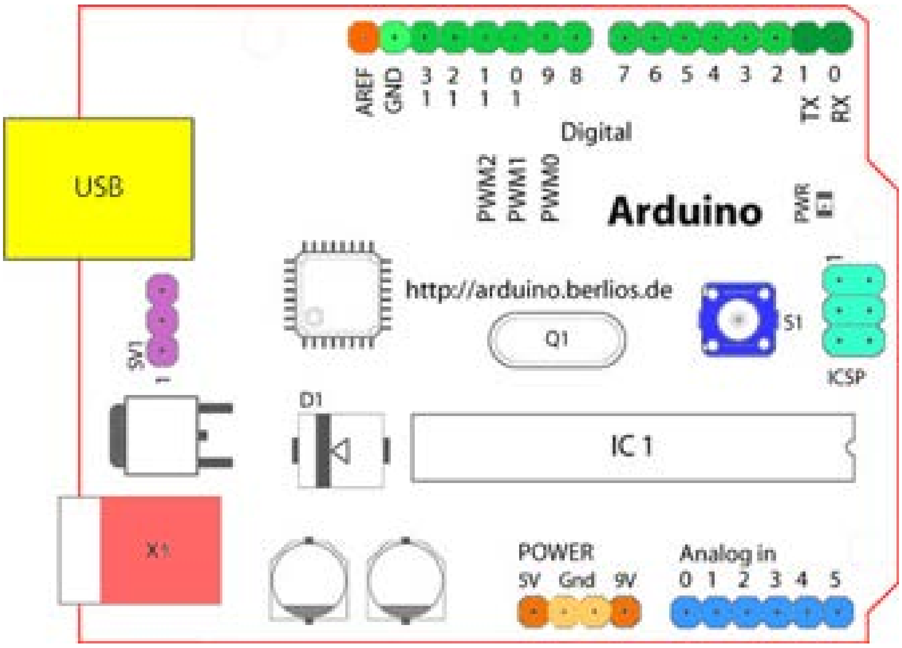


Figure 2: Common components of Arduino boards

- Analog Reference pin (orange)
- Digital Ground (light green)
- Digital Pins 2-13 (green)
- Digital Pins 0-1/Serial In/Out - TX/RX (dark green) - These pins cannot be used for digital i/o (digitalRead and digitalWrite) if using serial communication (e.g. Serial.begin).
- Reset Button - S1 (dark blue)
- In-circuit Serial Programmer (blue-green)
- Analog In Pins 0-5 (light blue)
- Power and Ground Pins (power: orange, grounds: light orange)
- External Power Supply In (9-12VDC) - X1 (pink)
- Toggles External Power and USB Power (place jumper on two pins closest to desired supply) - SV1 (purple)
- USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board) (yellow) [2]

» Some of the most common Arduino boards:

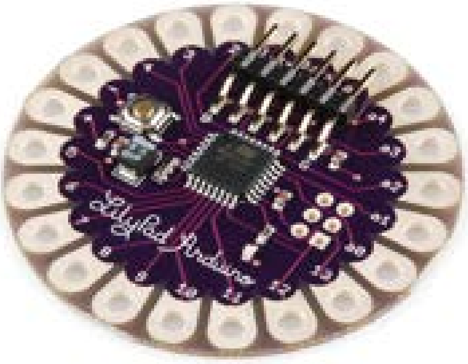


Figure 3: LilyPad Arduino

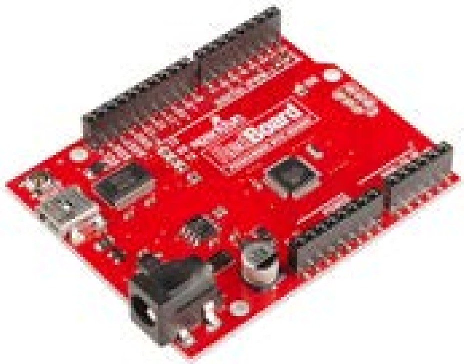


Figure 4: Arduino RedBoard

Figure 5: Arduino Leonardo

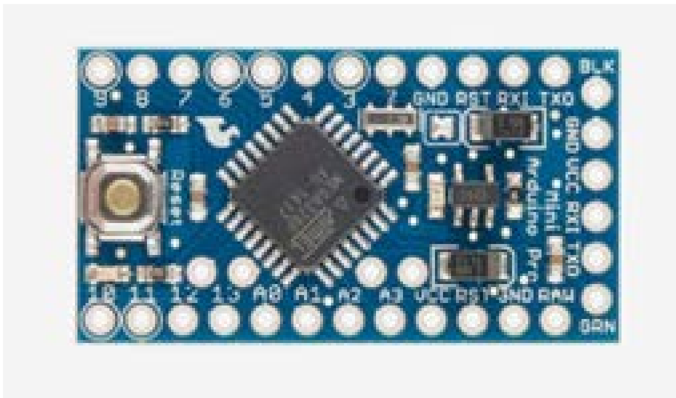


Figure 6: Arduino Pro Mini

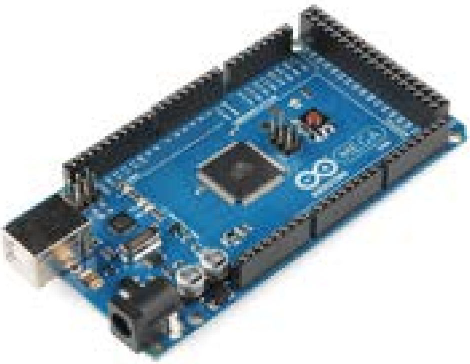


Figure 7: Arduino Mega (R3)

Figure 8: Arduino Nano

For this particular project, Arduino Pro Mini is used because of its comparatively small size.

2.2 Project: The Almighty Ruler

In engineering, precision is a consequential factor in any project as it decides the final design of the product one optate to manufacture. Thus, having the right quantifying implements is crucial, but there are many types of tools currently in utilization, making it very inconvenient at times.

Accordingly, for the Introduction to Engineering course's project, we would like to introduce "**The Almighty Ruler**".

Figure 9: "The Almighty Ruler" multitool

The Almighty Ruler is an Arduino-based measuring device that can be used for various purposes:

- Measure distance from the device to an object;
- Measure length of any curved path;
- Measure angle;
- Measure the revolution of any rotating object;
- Measure level.

Figure 10: Size comparison with commonly used portable handphones.
From left: **The Almighty Ruler**, a modern cellphone, a regular smartphone.

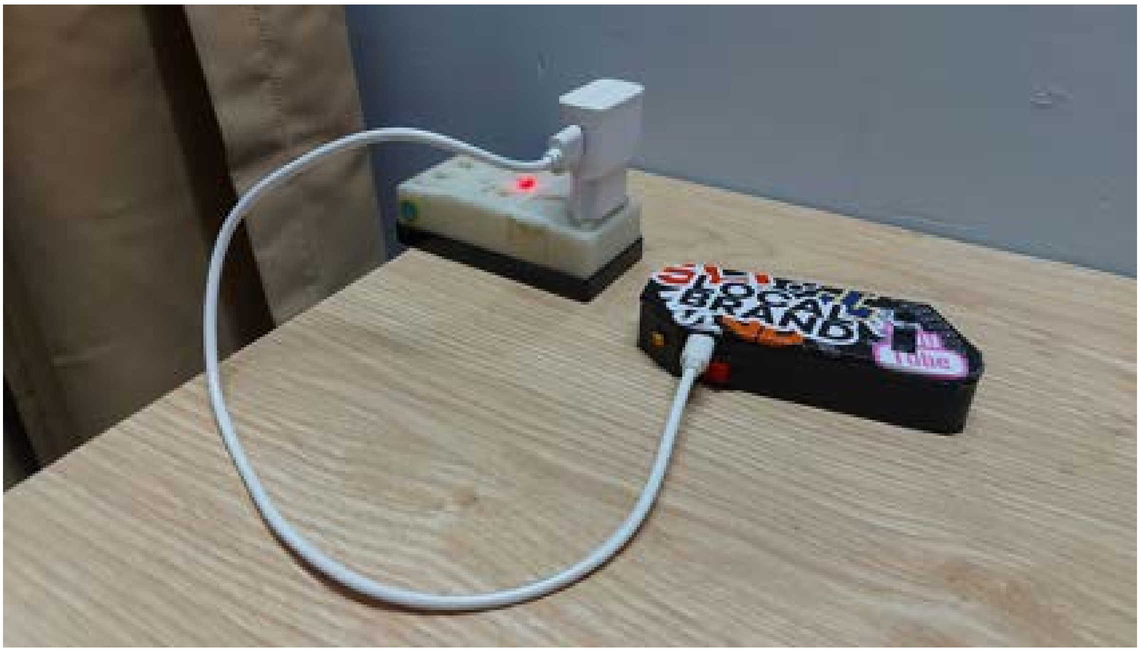


Figure 11: **The Almighty Ruler** can be recharged using a regular phone charger.

With its relatively small size and weight (similar to a portable phone), the device is tiny enough to fit inside a pocket. It is also battery-powered and can be easily recharged using a phone charger.

The Almighty Ruler uses an accelerometer and gyroscope sensor to accurately measure surface flatness and angle, a Sharp IR sensor to measure linear length without contact, and an encoder with a wheel that moves over a curved surface or curved path to get its length. Navigation through the device modes and features is done using 3 touch buttons marked as M (mode), U (unit) and R (Reset).

M - To choose between different types of measurements;
U - To choose between the units mm, cm, inches, and meter;
R - To reset the measured values to 0 after measuring a distance or angle.

Figure 12: Touch-sensitive navigating buttons M, R, U

The reason for using touch-sensitive buttons is to smoothly navigate through the modes and units without disturbing the position of the device during the measurement.
The case is designed to make the device as compact as possible and also easy to 3D print.

The Almighty Ruler is based on an available project on Instructibles.com [3]. Some features and the overall design have been modified to make them suitable for the available resources:

- Magnet part removed due to unavailable resources
- A proper toggle switch is used to replace the 3mm LED that was used as the switch
- An advanced laser module is implemented. It can be used as a laser pointer.
- The size of the case was slightly increased.

Figure 13: Newly implemented laser part

2.3 Project development plan

Phase 0: Preparation (25/01/2022 - 13/02/2022)
This is where the team members meet for the first time. The leader facilitates introductions, highlighting each person's skills and background. Team members also given the prototype of project plan and the ability to organize their responsibilities.

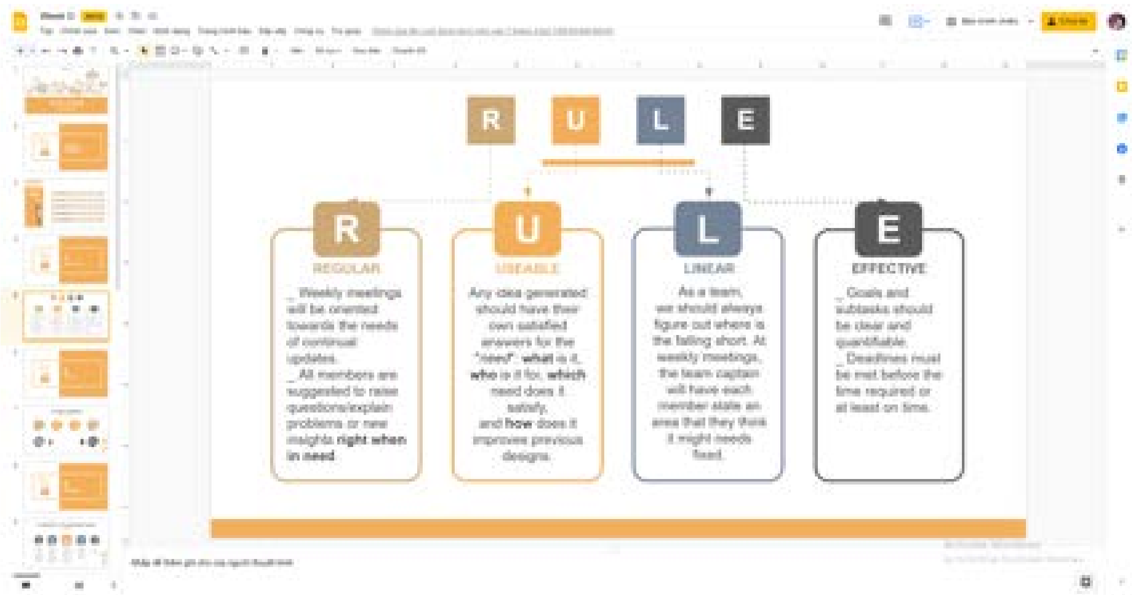


Figure 14: Part of team meeting slideshow, Phase 0: Teamworking rules

In this phase, everyone openly share their ideas for what to do in the project. Team leader helps by creating a spreadsheet in place to keep track on the "competition" among members, make communication easier.

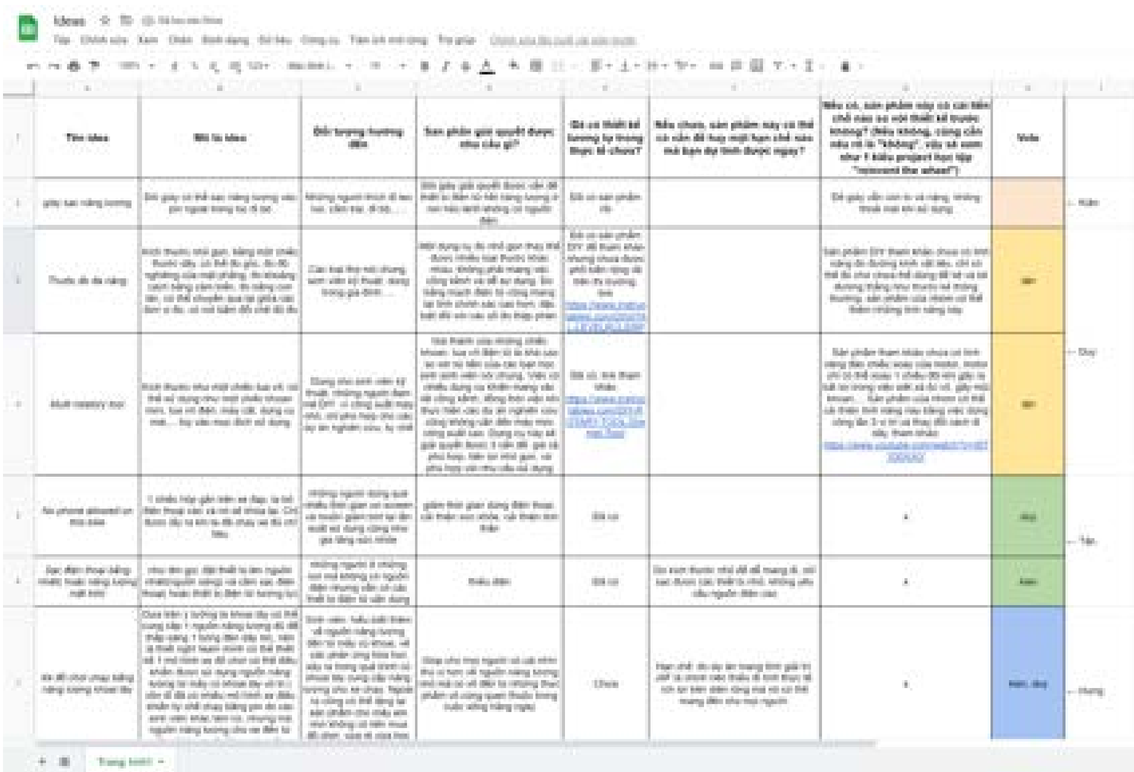


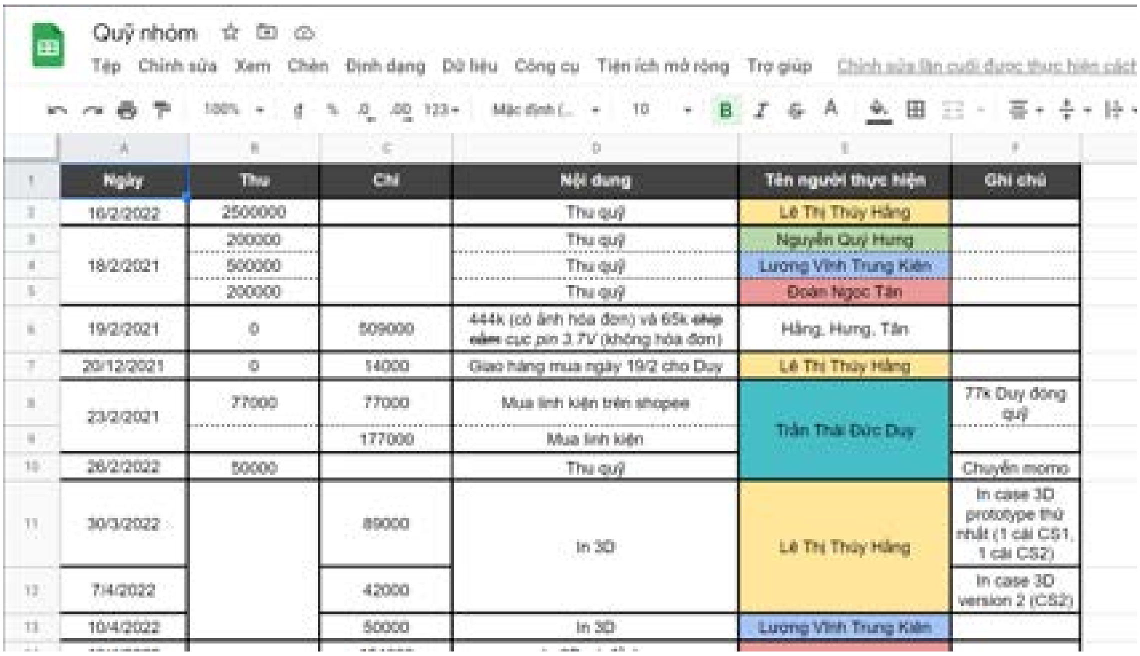
Figure 15: Ideas collection spreadsheet

Phase 1: Beginning (14/02/2022 - 06/03/2022)

At the end of the last phase, the team slowly gets used to each other. Now they have to figure out how to work together.

In this phase, responsibilities and goals are clear. Team leader explain what is expected at particular deadlines. Each person works to prepare for the required skills or tools that could be useful in the next steps.

The team leader announces the collection of the group funds and assigns a team member as fund manager.



Ngày	Thu	Chi	Nội dung	Tên người thực hiện	Ghi chú
16/2/2022	2500000		Thu quỹ	Lê Thị Thùy Hằng	
18/2/2021	200000		Thu quỹ	Nguyễn Quý Hưng	
	500000		Thu quỹ	Lương Vĩnh Trung Kiên	
	200000		Thu quỹ	Đoàn Ngọc Tân	
19/2/2021	0	509000	444k (có ảnh hóa đơn) và 65k ~~chip cảm~~ cục pin 3.7V (không hóa đơn)	Hằng, Hưng, Tân	
20/12/2021	0	14000	Giao hàng mua ngày 19/2 cho Duy	Lê Thị Thùy Hằng	
23/2/2021	77000	77000	Mua linh kiện trên shopee	Trần Thái Đức Duy	77k Duy đóng quỹ
		177000	Mua linh kiện		
26/2/2022	50000		Thu quỹ		Chuyển momo
30/3/2022		89000	In 3D	Lê Thị Thùy Hằng	In case 3D prototype thứ nhất (1 cái CS1, 1 cái CS2)
7/4/2022		42000			In case 3D version 2 (CS2)
10/4/2022		50000	In 3D	Lương Vĩnh Trung Kiên	

Figure 16: Group fund management spreadsheet

The components must be well prepared and tested. To avoid equipment failure during assembly, all components are purchased in sets of at least two pieces.

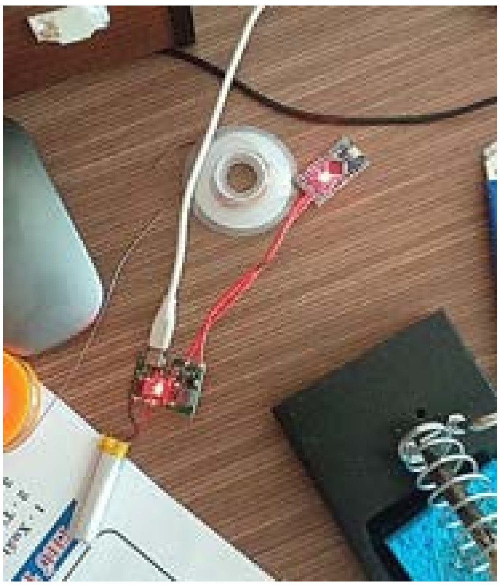


Figure 17: Testing new-bought electronic component

Phase 2: Hiatus (07/03/2022 - 31/03/2022)
Project is paused. All team members went on Military Training course pre-arranged by the university.

Phase 3: Assemble and report preparation (01/04/2022 - 24/04/2022)
In this phase, the team develops strategies to solve problems without impacting the schedule. Each member works at maximum efficiency with less oversight from the team leader.
The team is divided into two sub-teams:

- Engineering unit: Complete the exterior and assemble the circuit,
- Report unit: Prepare the presentation slides and the written report (using Overleaf, a comprehensive tool for scientific writing.)

Photos taken in the progress can be found in **Appendix 2: Photos** (Section 8.2).

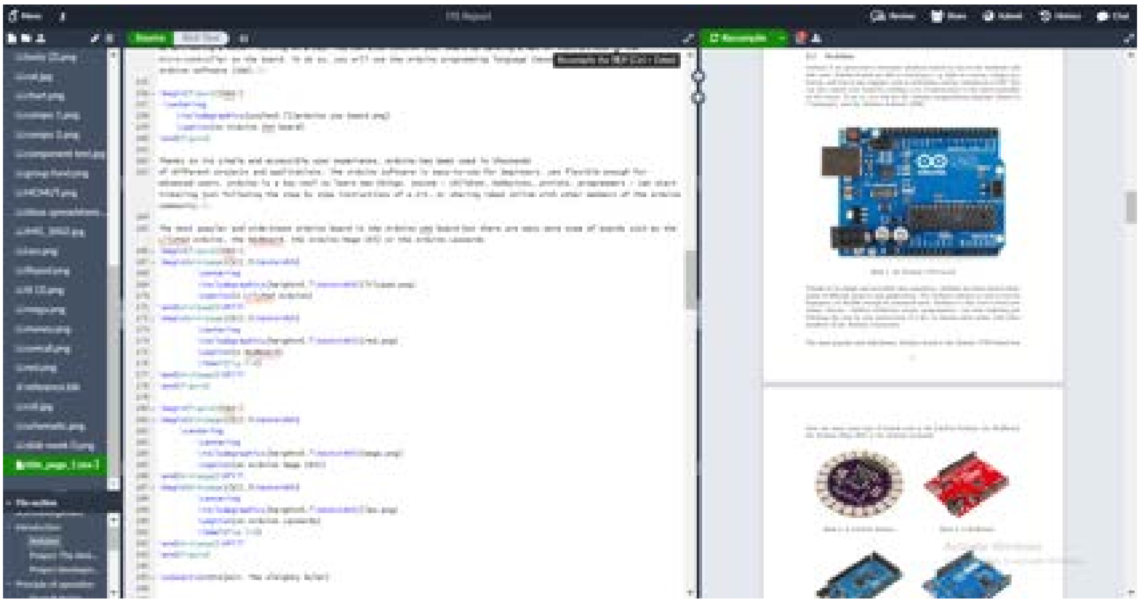


Figure 18: Paper report in the making, on Overleaf interface

Closing phase (25/04/2022 - 26/04/2022)
The team presents the project to classmates and lecturers. After that, the team leader decides what to do with the remaining budget and resources. Everyone will also report on what went well and what could be improved for future projects.

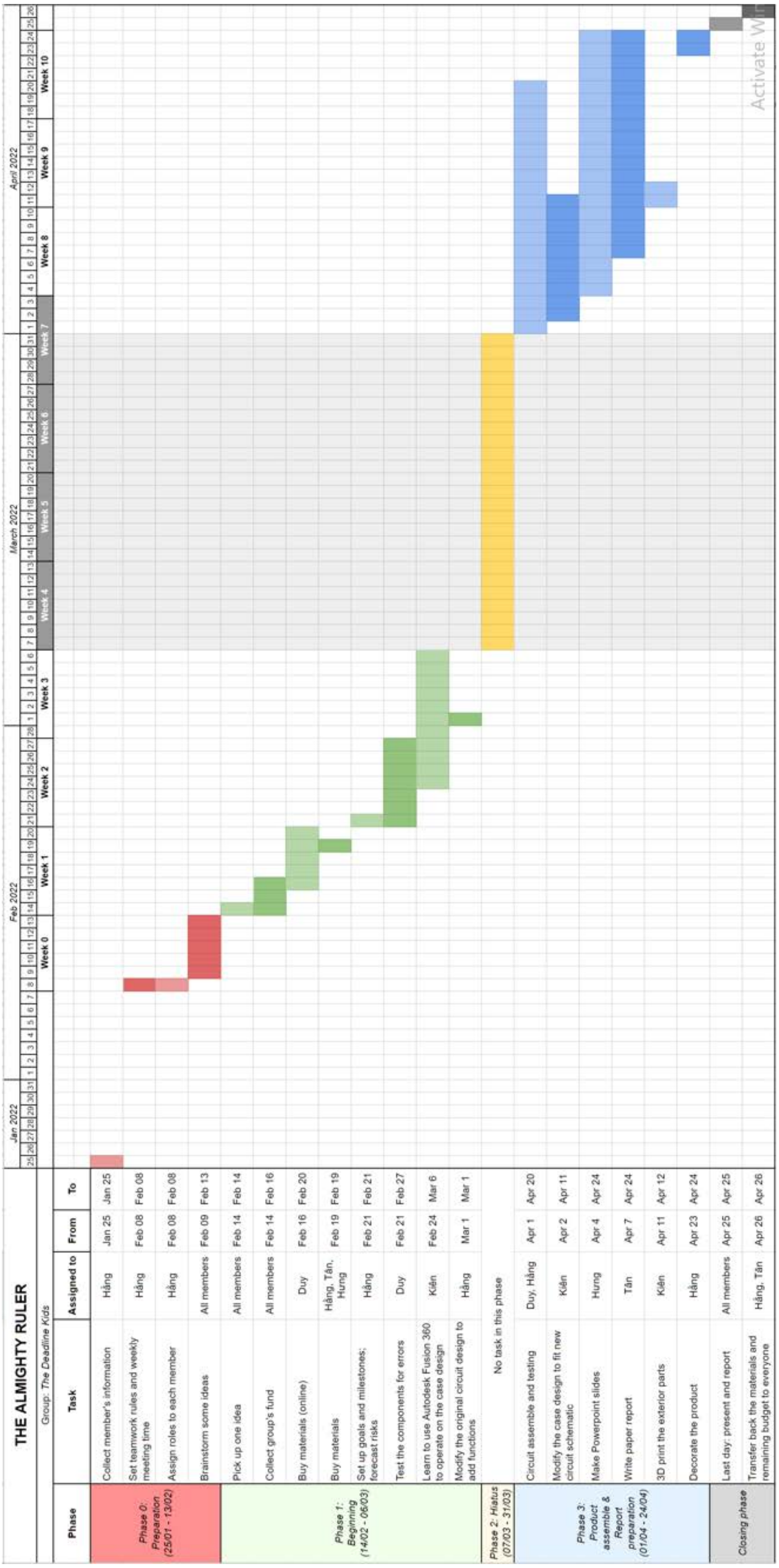


Figure 19: Gantt chart of the plan

3 How it works

3.1 Overall design

Figure 20: An overview photo capture of the product

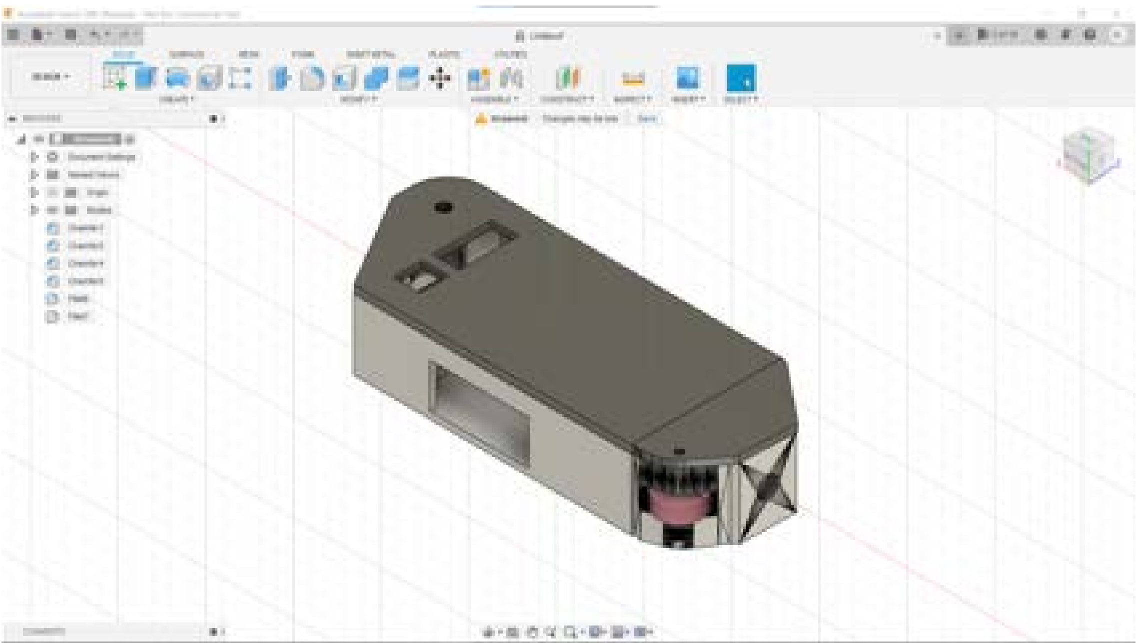


Figure 21: The completed exterior design

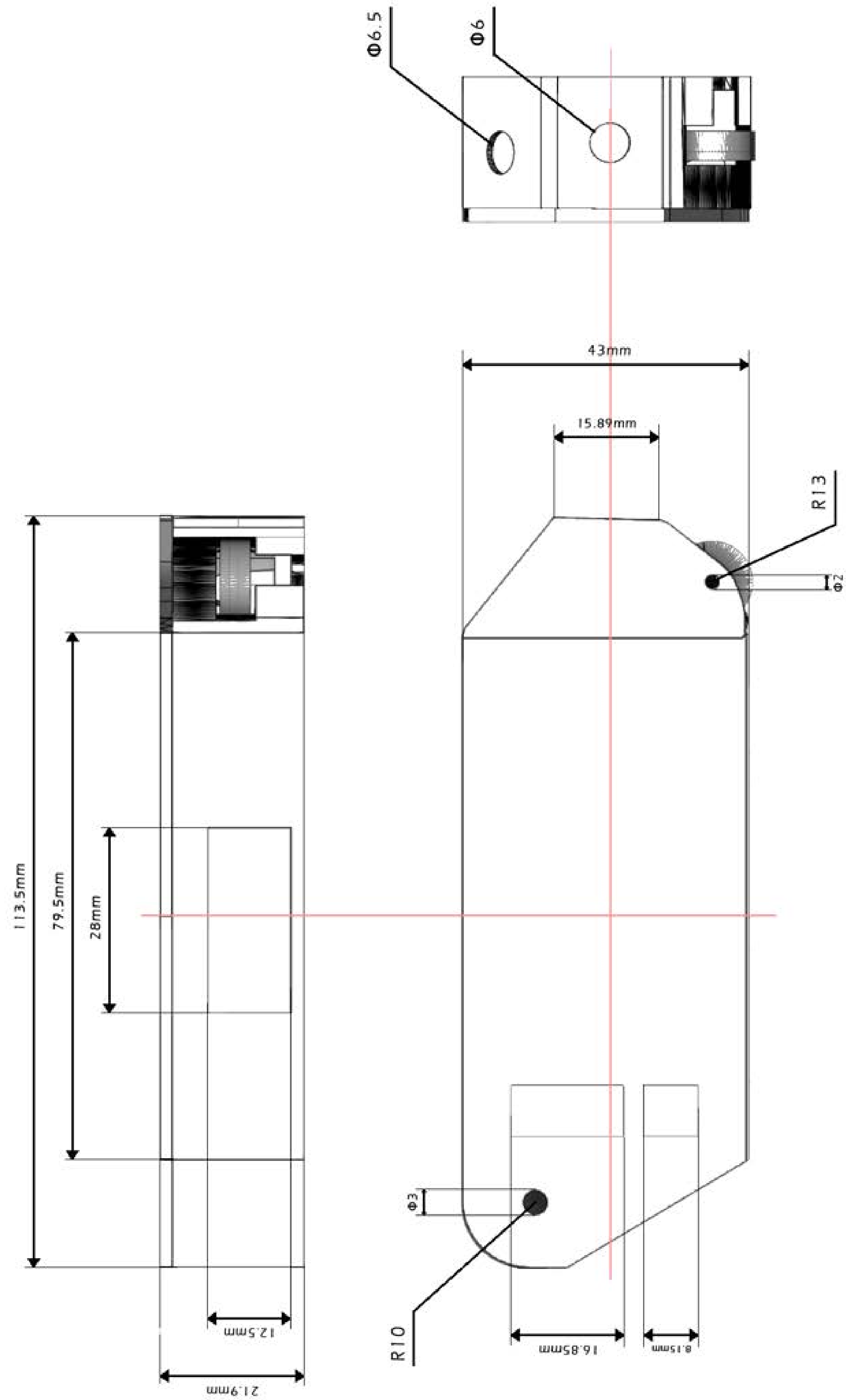


Figure 22: Technical sketches with measurements (body and lid)

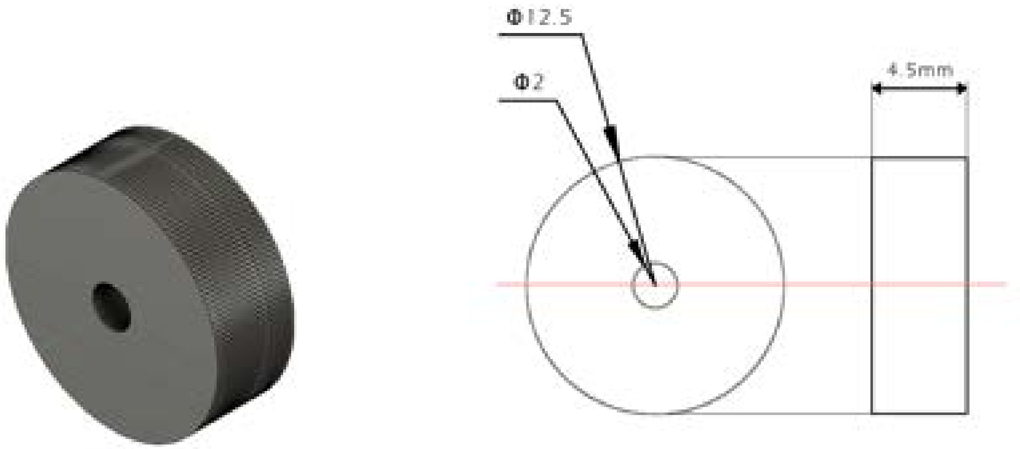


Figure 23: Technical sketches with measurements (wheel)

3.2 The case

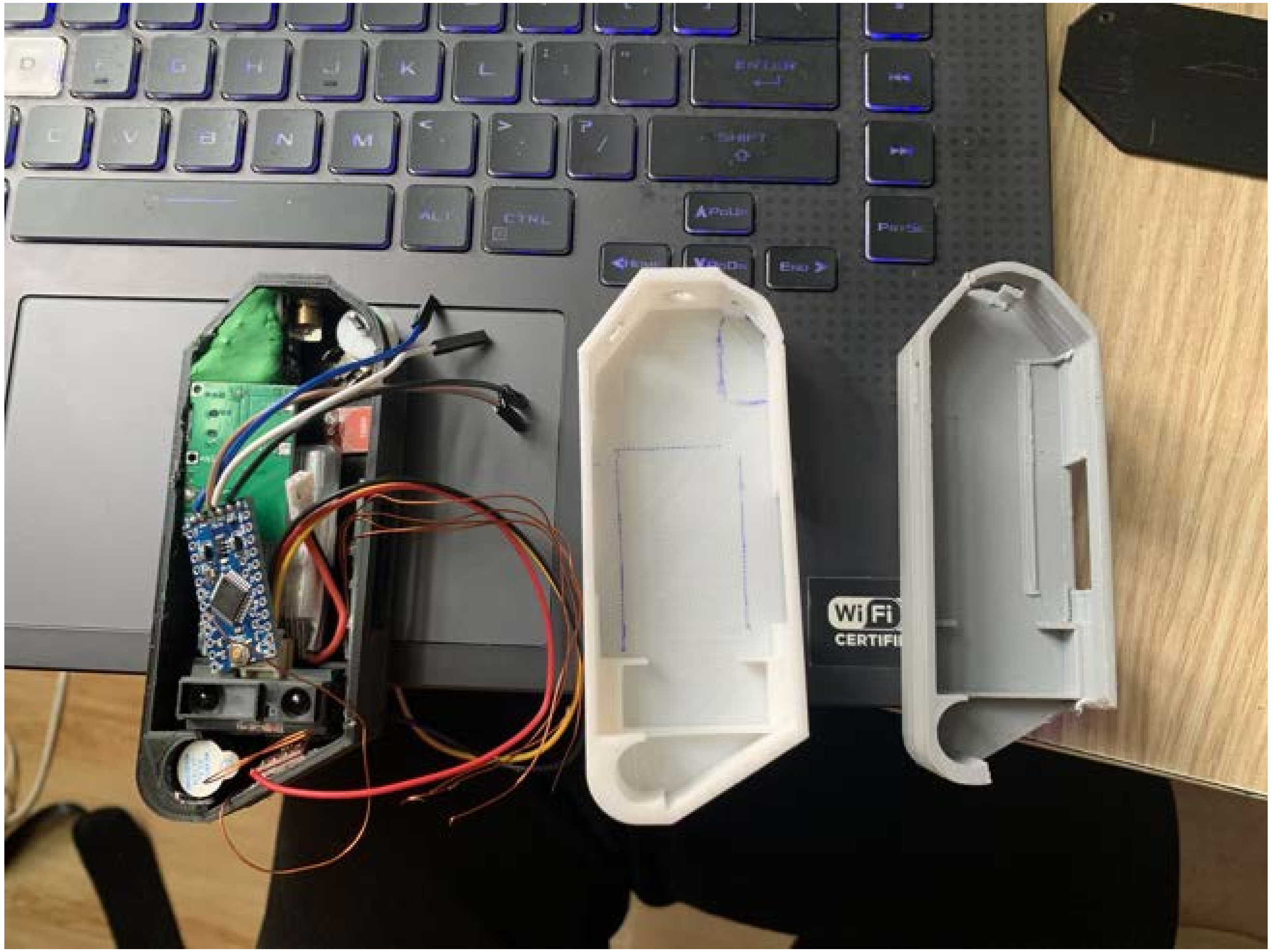
The product case consists of three separate parts: body, lid and a wheel that is 3D printed with PLA plastic filament. Autodesk Fusion 360 [4] is used to modify the STL files of the parts.

(a) Body (b) Lid (c) Wheel

Figure 24: Final STL files of the case components

The case was modified in several prototypes (total of 10 versions). The final version is slightly larger compared to original.
For decoration, we placed some vinyl stickers on the lid and put black electrical tape for the rest of the surfaces.

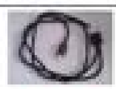
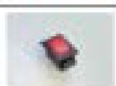
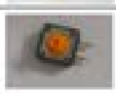
(a) A modification prototype
sketch with notes



(b) Size comparison:
lastest version - 5th version - 1st version

Figure 25: Case prototypes

3.3 Components

Order	Component	Picture	Price	Quantity
1	Sharp GP2Y0A41SK0F IR distance sensor		110.000	1
2	MPU6050 accelerometer/ gyroscope module		42.000	1
3	Battery charging module		19.000	1
4	11mm Rotary Mouse Scroll Wheel		9.000	1
5	128 X 32 0.91 OLED display		55.000	1
6	Arduino pro mini ATMEGA328 5V / 16MHz		115.000	1
7	12 mm buzzer		10.000	1
8	3.7v, 600mah lipo battery		95.000	1
9	3D printed case	 	50.000	1
10	Micro USB cable		35.000	1
11	TTP223 touch button module		10.000	3
12	CP2102 USB to UART TTL module		10.000	1
13	60(length)X2(dia) mm steel axle		1.500	1
14	5V laser module		10.000	1
15	Enamelled copper wire			1
16	Switch button			1
17	Press button			1
18	10K resistors			2

TOTAL MONEY SPENT	737.000 VND

SHARP GP2Y0A21YK0F IR DISTANCE MODULE

This is an analog sensor that provides a variable voltage output based on the object's distance from the sensor. It is used in this project to measure linear distances without contact.
Unlike other IR modules, the color of the detected object does not affect the output of the sensor.

Output voltage of the SHARP GP2Y0A21YK0F ranges from 2.3 V when an object is 10 cm away to 0.4 V when an object is 80 cm away.

MPU6050

MPU6050 is a micro-electro-mechanical systems device consisting of a 3-axis accelerometer and a 3-axis gyroscope. This helps measure acceleration, velocity, orientation, and displacement. It is an I2C based device that operates from 3.3 to 5V. In this project, MPU6050 is used to measure whether a surface is flat or not and also to measure the angle based on a line.

11MM MOUSE ENCODER

This is a mechanical incremental rotary encoder with feedback on the direction of rotation and speed. It has 24 steps per revolution. If the wheel diameter is known, it is possible to calculate the distance covered by the wheel on the encoder.
This project uses the encoder to measure curved line spacing.

ARDUINO PRO MINI

The Arduino is an important part in this project as it handles the computation and control of all the other parts.
Unlike other Arduino boards, the Pro Mini cannot be programmed directly by plugging in a USB cable as it doesn't have a built-in USB to serial TTL converter. Therefore, one should first connect an external USB to serial converter to the Pro Mini in order to program it.

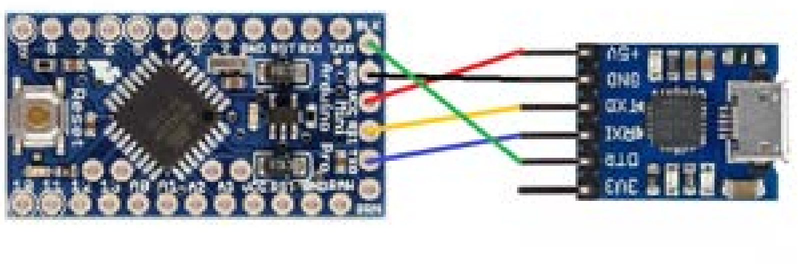


Figure 26: Connecting Arduino Pro Mini to USB-to-TTL module

3.4 Circuit schematic

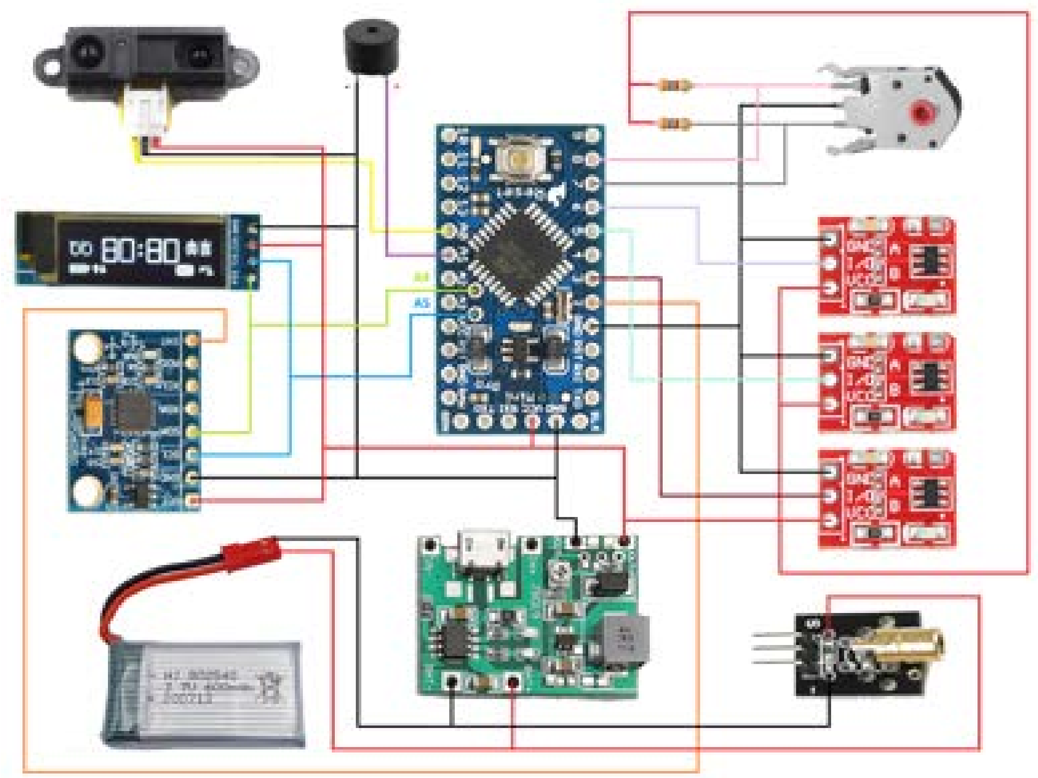


Figure 27: Device schematic

Arduino pinouts:
D2 → IN+ of MPU6050
D3 → I/O of M touch button (MODE)
D5 → I/O of U touch button (UNIT)
D6 → I/O of R touch button (RESET)
D7 → +(1) of wheel encoder
D8 → +(2) of wheel encoder
A0 → yellow wire of Sharp IR
A1 → + of Buzzer
A4 → SDA of OLED and MPU6050
A5 → SCL of OLED and MPU6050
GND → GND of all components
VCC → + of boosted charging module

BAT+ of boosted charging module → + of battery
BAT- of boosted charging module → - of battery

+ of laser module → + of battery
- of laser module → - of battery

3.5 Principle of operation

Instructions and calculations of each feature will be discussed in this section.
The "M" (Mode) button is used to change between modes.

3.5.1 First mode: Levelness measurement

This is the first feature that appears after turning on the device.
This feature is used to measure flatness of a surface. Placing the device on any surface with the OLED display facing up will measure how inclined that surface is compared to the ground. It may take a few seconds for the device to calibrate itself before using this feature.

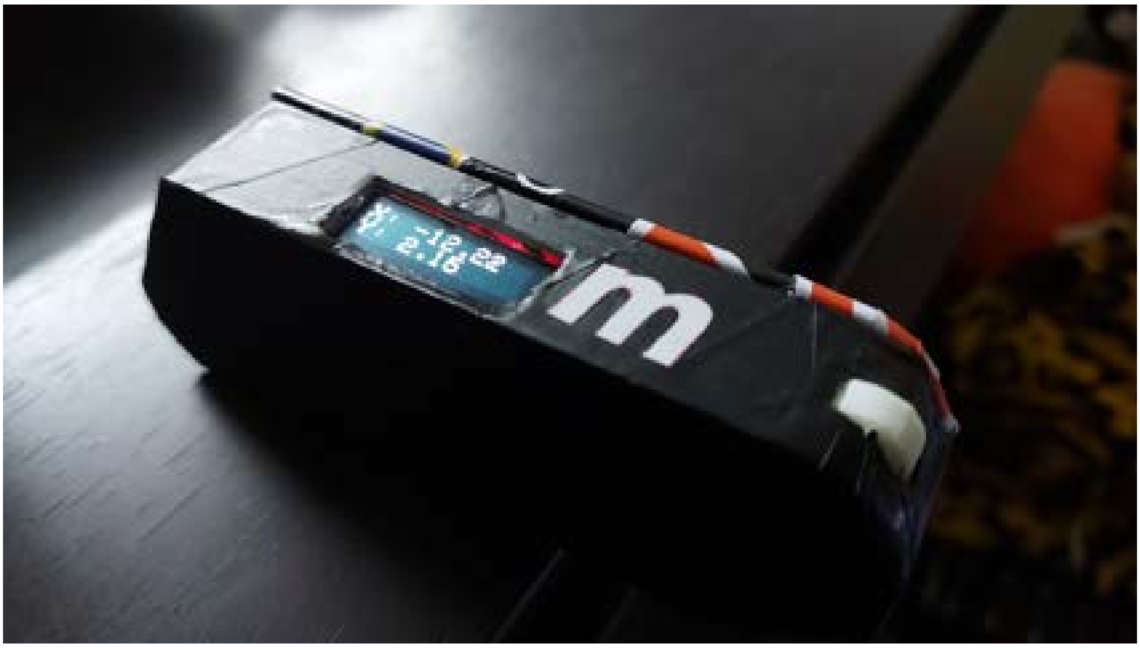


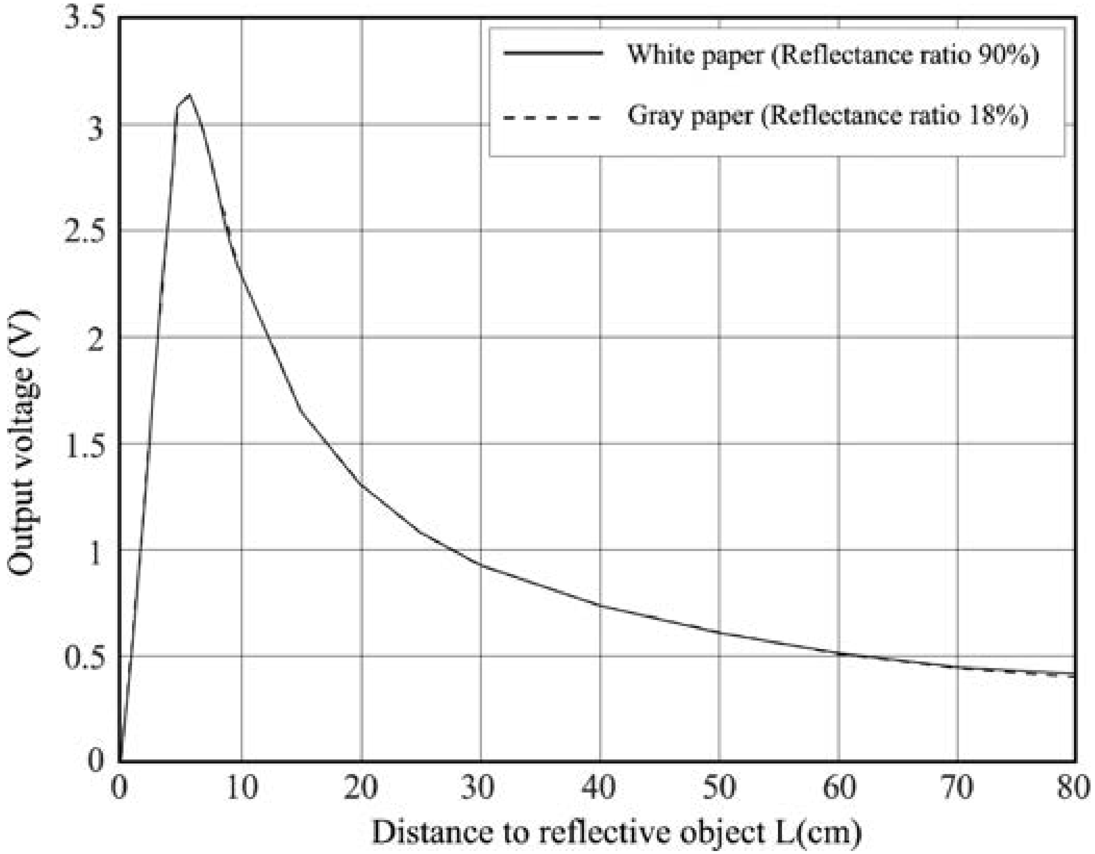
Figure 28: The surface the device is on is tilted at -12.22 (on the X axis) and 2.15 (on the Y axis).

3.5.2 Second mode: Distance measurement

Using the Sharp IR sensor, this feature measures the distance to an object in front of it. The usable detection range starts at 10 cm to around 80 cm. Note that the output voltage of an object 2 cm away is the same as the output voltage of an object 28 cm away. So the usable detection range starts after the peak at about 10 cm, or 2.3 V.
To determine the distance between the sensor and an object, this formula is provided in the Arduino source code:

$$\textit{Distance (cm)} = \textit{29.988} \times \textit{POW(Volt , -1.173)}$$

Simply hold an object in front of the built-in sensor and read the result on the display. The result can also be converted to other units such as centimeters or inches by pressing the "U" (Unit) button. [Figure 29]



(a) GP2Y0A21YK0F distance measuring characteristics (output)

(b) Distance from the object to the sensor is 100.56 millimeters.

Figure 29: Distance measurement feature

3.5.3 Third mode: Curved path measurement

This mode uses a 3D printed wheel attached to an 11mm wheel encoder to measure the distance the wheel has rolled using mathematical formulas.

Number of steps per rotation of the encoder shaft: N = 24 steps.
The diameter of the wheel, D = 12.7mm.
Circumference of the wheel:

$$C = 2 * pi * (D/2) = 2 * 3.14 * 6.35 = 39.898mm$$

Therefore, distance moved per step

$$C/N = 39.898/24 = 1.6625mm$$

To use this feature, hold the device while rolling the wheel on the surface. The result will be displayed along with other unit conversions. You can reset the value by pressing the "R" (Reset) button.

Figure 30: Roller mode in use

3.5.4 Fourth mode: Angle measurement

Angles can also be calculated using the same MPU6050 module for levelness measurement.
To use this feature, the device must be placed so that the OLED display faces up. Choose a position and press "R" (Reset) to set it as the base. Hold on one side and leisurely move the other side on a circular motion (either clockwise or counter-clockwise) to measure the angle of scanned sector.

3.5.5 Fifth mode: Number of revolution measurement

This function is useful for measuring the frequency (or period) of a rotating surface. Whenever the object rotates a full circle, you should hear a beep from the buzzer inside. Just fix the device firmly in the surface and read the result on the screen.

4 Product testing

4.0.1 Curved path measurement test result

We designed a trial test on this function as follow:

Figure 31: Pre-measure the length of chosen path (7.5 cm) with tape measure

A 7.5 cm path was chosen on the surface of a roll of tape and checked carefully with a tape measure. We put some double sided tape on the path to increase friction.

Figure 32: Rolling the wheel

After everything is set, the experimenter carefully rolled the wheel on the path.

Figure 33: Result displayed on the screen is 7.65cm.

The result obtained is 7.65 cm, which is very close to original measurement (7.5 cm), with a small error. Some reasons for such errors can be:

- Slight mistakes in 3D printing process can lead to an incorrect measurement of the wheel's figure, such as diameter or roundness of its rim.
- The Pi value appeared in formula is an irrational number; therefore results might be inaccurate.

Furthermore, the 3D-printed wheel is lack of friction, which is a very important factor if one wants to obtain accurate results. This wheel can only roll on sticky surfaces and sometimes it might still slip off the track.

4.0.2 Results of other functions

- Results of the angle and rotation measurement are satisfactory.
- Laser function is independent from the Arduino circuit, thus it can be toggled on/off without turning on the whole system.
- Levelness measurement function is working well, however the origin (where X = 0 and Y = 0) is on an unstable position.
- Distance measurement using IR sensor only works well in range of 10 to 20cm, though its range varies from 10 to 80cm. Larger distances cause inaccuracy results. This is one of the disadvantages of these types of sensors: the response is not linear. In other words, a large change in output voltage does not always correspond to a large change in range.

On the other hand, the switch button was added on the wrong side of the device (it's supposed to be on the opposite side or on the sides nearby); unfortunately this prevents the comfortable placement for measuring angle and levelness.

Figure 34: The switch was added later in the process and remains on the wrong side.

5 Optimization

Here are our recommendations to optimize this product:

- To improve the accuracy of IR sensor's readings, we may try to measure and plot many data points in Excel and fit a curve through these points. Once obtained a new function, we will change the formula used in the source code.
- Since the MPU6050 module is glued to the case, it might not be perfectly level. Therefore, we need to re-calibrate by placing it on a flat surface, record the X and Y values, then assign these values to the variables "calibx" and "caliby" in the code.
- Use different material to produce a different wheel with enhanced friction.
- Change the position of the power switch; and we also need to choose a smaller switch type.
- Make a new detachable lid design with locking mechanism to easily separate for future repairs.

Moreover, if the product is mass-produced, cost is one of the most important factors we need to reduce. The cost of one product is around VND 750000, including the cost of purchasing components, some negligible small parts and decoration; not to mention the countless parts that were damaged in the process.

Designing custom electronic boards is the most efficient way to cut costs as Arduino boards are expensive to purchase and assemble. Some other ways to lower the price are to make the case with a plastic press machine instead of a 3D printer, or to make the assembly process fully automated to reduce labor costs.

6 Conclusion

This report illustrated the development process of "**The Almighty Ruler**" - a small, yet handy multitool that fits inside a pocket. It can be used for various measurement needs.

Although there are still slight errors and malfunctions after trials and experiments, overall the device works well and meets basic measurement needs. We also presented some ideas for optimization.

It must be noted that this project was limited to a singular group of students in one semester. Employing a time-lag study will enhance the reliability and validity of the results. However, throughout this project, all group members gain knowledge and skills that will come in handy later, through interactive discussions, group work, solving unexpected problems and constant design. We develops a much broader level of knowledge and understanding, being better prepared for the world of engineering outside the classroom. This is a precious experience that will lay the foundation for our career in the future.

7 References

1. *What Is Arduino?* https://www.arduino.cc/en/Guide/Introduction. Accessed 23 Apr. 2022.
2. *Overview of the Arduino UNO Components* | Arduino Documentation. https://docs.arduino.cc/tutorials/uno-rev3/intro-to-board. Accessed 23 Apr. 2022.
3. Panikulam, Patrick. *DIGITAL MULTI-FUNCTION MEASURING TOOL*. Instructables, https://www.instructables.com/DIGITAL-LEVELRULERPROTRACTORROLL-MEASURE/. Accessed 08 Mar. 2022.
4. Fusion 360 | *3D CAD, CAM, CAE PCB Cloud-Based Software* | Autodesk. https://www.autodesk.com/products/fusion-360/overview. Accessed 23 Apr. 2022.

8 Appendices

8.1 Appendix 1: Arduino code

```
  #include <MPU6050_tockn.h>
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <SharpIR.h>
#define model 1080
SharpIR SharpIR(A0, model);



MPU6050 mpu6050(Wire);
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 32 // OLED display height, in pixels
#define sensor A0
int mod=1;
// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
#define OLED_RESET     4 // Reset pin # (or -1 if sharing Arduino reset pin)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);



void setup()
{
   display.clearDisplay();
  pinMode(3,INPUT);
  pinMode(5,INPUT);
  pinMode(6,INPUT);
  pinMode(7,INPUT);
  pinMode(8,INPUT);
  pinMode(A1,OUTPUT);
  pinMode(11,OUTPUT);
  Serial.begin(9600);
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C))
  { // Address 0x3C for 128x32
    Serial.println(F("SSD1306 allocation failed"));
    for(;;); // Don't proceed, loop forever
  }
  display.display();
  tone(A1,500,500);
  tone(A1,2000,500);
  delay(1000);
  Wire.begin();
  mpu6050.begin();
  mpu6050.calcGyroOffsets(true);

}

void loop()
{

  while(mod==1)
```

```
 {


   for(int i=0;i<=7;i++)
   {
   display.clearDisplay();
   display.setTextSize(3); // Draw 2X-scale text
   display.setTextColor(WHITE);
   display.setCursor(0,0);
   display.print(" LEVEL  ");
   display.display();
   delay(250);
   if(digitalRead(3)==HIGH){mod+=1;i=7;tone(A1,2500,30);delay(250);}
   }
 float calibx=0.5;
 float caliby=3.5;


 while(mod==1)
 {
   mpu6050.update();
   display.clearDisplay();
   display.setTextSize(2); // Draw 2X-scale text
   display.setTextColor(WHITE);
   display.setCursor(0,0);
   display.print("X: ");
   display.print(mpu6050.getAngleX()+calibx);
   display.print('\n');
   display.print("Y: ");
   display.print(mpu6050.getAngleY()+caliby);

   display.display();
   if(mpu6050.getAngleX()+calibx>-0.6 && mpu6050.getAngleX()+calibx
  <0.6 && mpu6050.getAngleY()+caliby>-0.6 && mpu6050.getAngleY()+
  caliby<0.6) {tone(A1,1000,100);}
   if(digitalRead(3)==HIGH){mod+=1;tone(A1,2500,30);delay(500);}
 }
}



while(mod==2)
{
  for(int i=0;i<=7;i++)
   {
     display.clearDisplay();
     display.setTextSize(2); // Draw 2X-scale text
     display.setTextColor(WHITE);
     display.setCursor(0,2);
     display.print(" DISTANCE");
     display.print('\n');
     display.setTextSize(1);
     display.print("     (mm,cm,inch)");
     display.display();
     delay(250);
     if(digitalRead(3)==HIGH)
         {mod+=1;i=7;tone(A1,2500,30);delay(500);}
   }
```

```
 int d=0;
 int e=0;
digitalWrite(11,HIGH);
int unit=1;
int j=1;
while(mod==2)
{
 int actualdis=0;
 float volts=analogRead(sensor)*0.0048828125;
 float distancemm=(29.998*pow(volts,-1.173))*10;
 float distancecm=(29.998*pow(volts,-1.173));


 float distanceinch=distancecm/2.54;
 display.clearDisplay();
 display.setTextSize(2); // Draw 2X-scale text
 display.setTextColor(WHITE);
 display.setCursor(0,0);

 if(unit==1)
 {
   if (j==1)
   {  display.clearDisplay();
      display.setTextSize(2); // Draw 2X-scale text
      display.setTextColor(WHITE);
      display.setCursor(0,0);
      display.print("Millimeter");
      display.display();
      delay(1000);
      display.clearDisplay();
      j=2;
   }
   display.setCursor(0,0);
   display.println("Distance");
   if(distancemm<=800)
   {
    display.print(distancemm);
   display.print(" mm");
   d=distancemm/10;
   if(d!=e){tone(A1,2500,30);e=d;}
   delay(100);
   }
   else{display.print("OffRange");}
   display.display();      // Show initial text
   if(digitalRead(5)==HIGH)
     {unit+=1;tone(A1,2500,30);delay(500);}
  }

  if(unit==2)
   {
      if (j==2)
   {  display.clearDisplay();
      display.setTextSize(2); // Draw 2X-scale text
      display.setTextColor(WHITE);
      display.setCursor(0,0);
      display.print("Centimeter");
      display.display();
      delay(1000);
```

```
          display.clearDisplay();
          j=3;
      }
         display.setCursor(0,0);
         display.println("Distance");
         if(distancecm<=80)
         {display.print(distancecm);
         display.print(" cm");
         d=actualdis;
         if(d!=e)
           {tone(A1,2500,30);e=d;}
         delay(100);
         }
         else{display.print("OffRange");}
         display.display();      // Show initial text
         if(digitalRead(5)==HIGH)
           {unit+=1;tone(A1,2500,30);delay(500);}
        }

      if(unit==3)
        {
         if (j==3)
      {   display.clearDisplay();
          display.setTextSize(2); // Draw 2X-scale text
          display.setTextColor(WHITE);
          display.setCursor(0,0);
          display.print("  Inches");
          display.display();
          delay(1000);
          display.clearDisplay();
          j=1;
      }
          display.setCursor(0,0);
         display.println("Distance");
         if(distancecm<=80)
         {
         display.print(distanceinch);
         display.print(" inch");
         d=distanceinch;
         if(d!=e)
           {tone(A1,2500,30);e=d;}
         delay(100);
         }
         else{display.print("OffRange");}
         display.display();      // Show initial text
         if(digitalRead(5)==HIGH)
         {unit=1;tone(A1,2500,30);delay(500);}
        }


      if(digitalRead(3)==HIGH){mod+=1;tone(A1,2500,30);delay(500);}
     }
 }



digitalWrite(11,LOW);
```

```
byte lastState = 0;
byte steps = 0;
byte AState = 0;
byte BState = 0;
byte State = 0;
int b=0;
int c=0;
while(mod==3)
{
  for(int i=0;i<=7;i++)
  {
  display.clearDisplay();
  display.setTextSize(3); // Draw 2X-scale text
  display.setTextColor(WHITE);
  display.setCursor(0,0);
  display.print("ROLLER");
  display.display();
  delay(250);
  if(digitalRead(3)==HIGH){mod=4;i=7;tone(A1,2500,30);delay(250);}
  }
  display.clearDisplay();
  display.setTextSize(2); // Draw 2X-scale text
  display.setTextColor(WHITE);
  display.setCursor(0,0);
    int unit=1;
    int j=1;
 while(mod==3)
 {

   // read the input pin:
AState = digitalRead(7);
BState = digitalRead(8) << 1;
State = AState | BState;

if (lastState != State){
  switch (State) {
    case 0:
      if (lastState == 2){
        steps++;

      }
      else if(lastState == 1){
        steps--;

      }
      break;
    case 1:
      if (lastState == 0){
        steps++;

      }
      else if(lastState == 3){
        steps--;

      }
      break;
    case 2:
      if (lastState == 3){
```

```
          steps++;
        }
        else if(lastState == 0){
          steps--;
        }
        break;
      case 3:
        if (lastState == 1){
          steps++;
        }
        else if(lastState == 2){
          steps--;
        }
        break;
    }
  }
  b=lastState;
  lastState = State;
  if(b!=State)
  {  c+=1;}

    float distancemm=c*1.6625;
    float distancecm=distancemm/10;
    float distanceinch=distancecm/2.54;
    float distancemeter=distancemm/1000;
    display.clearDisplay();
    display.setTextSize(2); // Draw 2X-scale text
    display.setTextColor(WHITE);
    display.setCursor(0,0);
    display.println("Distance");
    display.setTextSize(1);
    display.print(distancemm);
    display.print(" mm    ");
    //display.print('\n');
    display.print(distancecm);
    display.print(" cm");
     display.print('\n');
    display.print(distanceinch);
    display.print(" inch  ");
    display.print(distancemeter);
    display.print(" m");
    display.display();
    delay(1);
    if(digitalRead(6)==HIGH){c=0;tone(A1,2500,30);delay(250);}
    if(digitalRead(3)==HIGH){mod=4;tone(A1,2500,30);delay(500);}
    }
  }


  while(mod==4)
  {


    for(int i=0;i<=7;i++)
    {
    display.clearDisplay();
    display.setTextSize(3); // Draw 2X-scale text
    display.setTextColor(WHITE);
```

```
    display.setCursor(0,0);
    display.print(" ANGLE  ");
    display.display();
    delay(250);
    if(digitalRead(3)==HIGH){mod=5;i=7;tone(A1,2500,30);delay(250);}
    }

    float offset=0;
    float angle=0;
  while(mod==4)
  {
    mpu6050.update();
    display.clearDisplay();
    display.setTextSize(2); // Draw 2X-scale text
    display.setTextColor(WHITE);
    display.setCursor(0,0);
    display.print("ANGLE = ");
    display.print('\n');
    angle=(mpu6050.getAngleZ()-offset)/2;
    display.print(angle);
    display.print(" deg");
    display.display();
    if(digitalRead(6)==HIGH){offset=mpu6050.getAngleZ();tone(A1
   ,2500,30);delay(500);}
    if(digitalRead(3)==HIGH){mod=5;tone(A1,2500,30);delay(500);}
  }
 }


  while(mod==5)
  {


    for(int i=0;i<=7;i++)
    {
    display.clearDisplay();
    display.setTextSize(2); // Draw 2X-scale text
    display.setTextColor(WHITE);
    display.setCursor(0,0);
    display.print("REVOLUTION");
    display.display();
    delay(250);
    if(digitalRead(3)==HIGH){mod=1;i=7;tone(A1,2500,30);delay(250);}
    }

    float offset=0;
    float angle2=0;
    int d=0;
    int e=0;
  while(mod==5)
  {
    mpu6050.update();
    display.clearDisplay();
    display.setTextSize(2); // Draw 2X-scale text
    display.setTextColor(WHITE);
    display.setCursor(0,0);
    display.print("Num of rev");
    display.print('\n');
```

```
    angle2=((mpu6050.getAngleZ()-offset)/2)/360;
     display.print("    ");
    display.print(angle2);
    display.display();
    d=angle2;
      if(d!=e){tone(A1,2500,30);e=d;}
    if(digitalRead(6)==HIGH){offset=mpu6050.getAngleZ();tone(A1,2500,30);delay(500);}
    if(digitalRead(3)==HIGH){mod=1;tone(A1,2500,30);delay(500);}
  }
 }



}
```

MultiMeasure.ino

8.2 Appendix 2: Photos

Figure 35: Laser feature can be used to play with cats

Figure 36: Desoldering a component

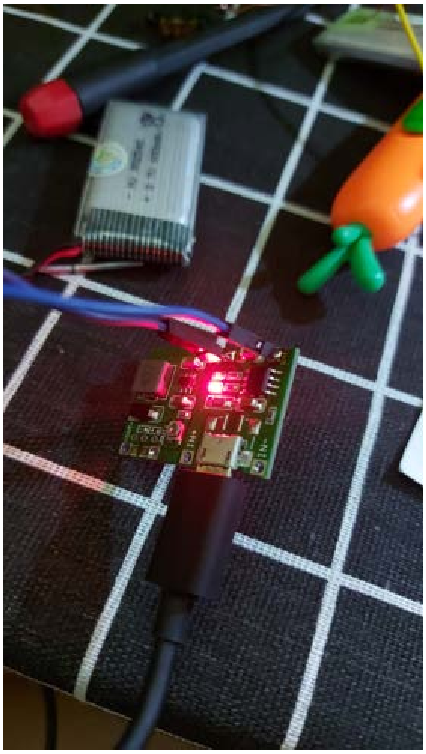


Figure 37: Testing the charger module

Figure 38: Assembling and fitting components into the case

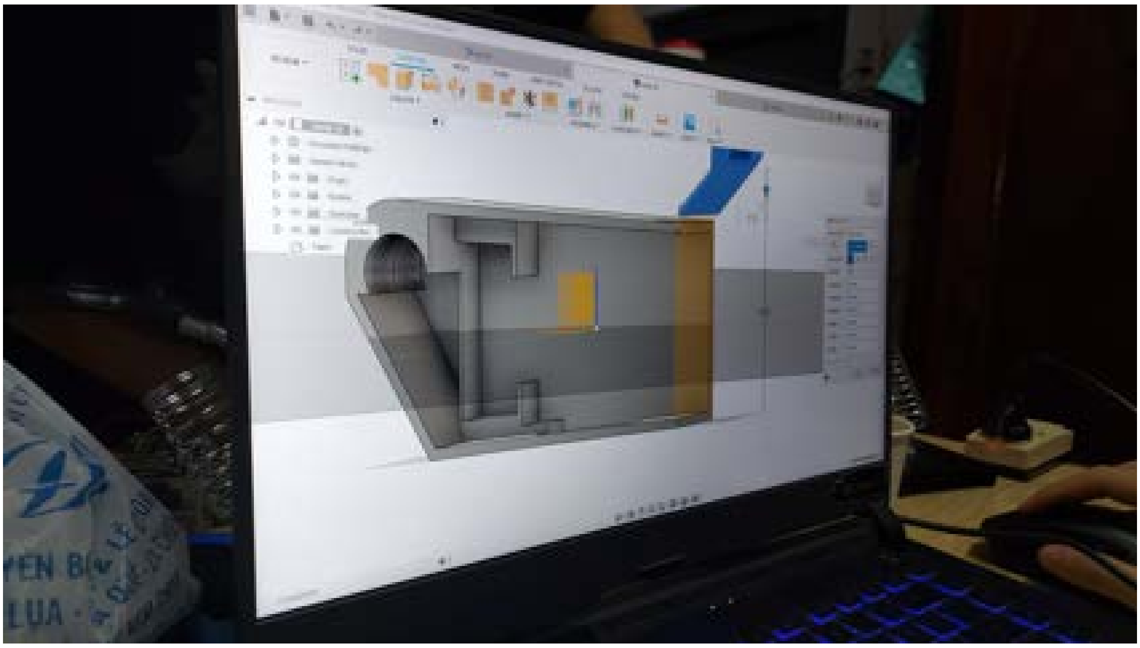


Figure 39: Modifying the case model in Fusion

Figure 40: Modifying the case in paper


Figure 41: Testing the Arduino code

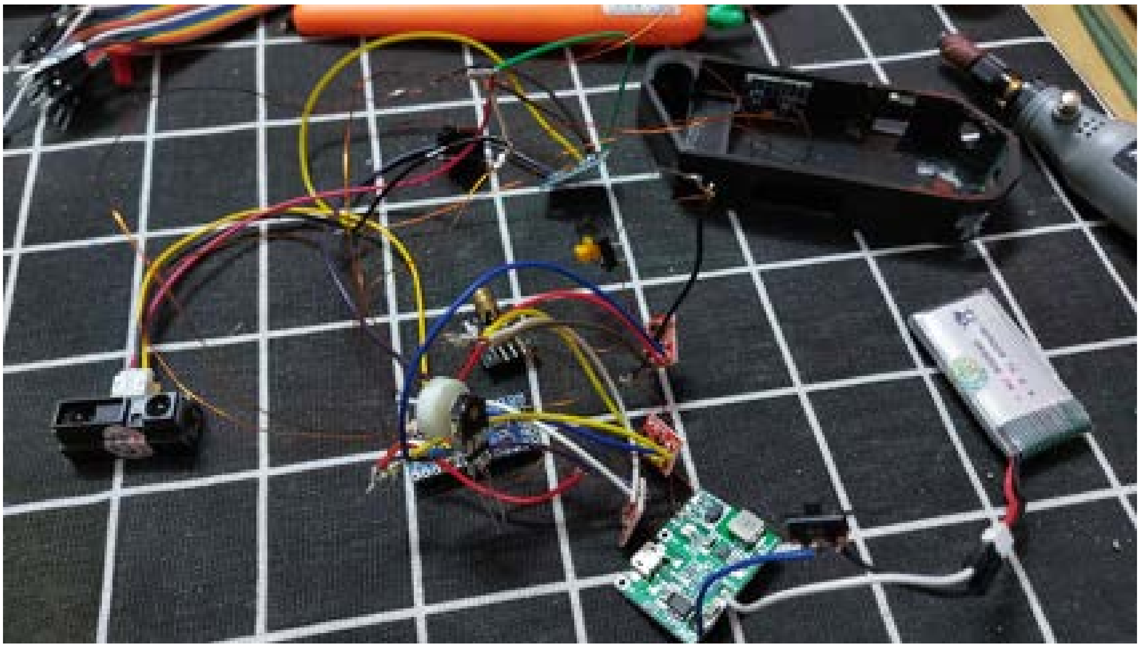


Figure 42: Testing the circuit