Enhanced features for corrosion monitoring product designed at WEL (iPEC)
P-08: 2D Mapping System

Group : TUES-03

Mudit Goyal 200070045
Keshav Patel Keval 200110055
Rishabh Ravi 200260041

Contents

Introduction

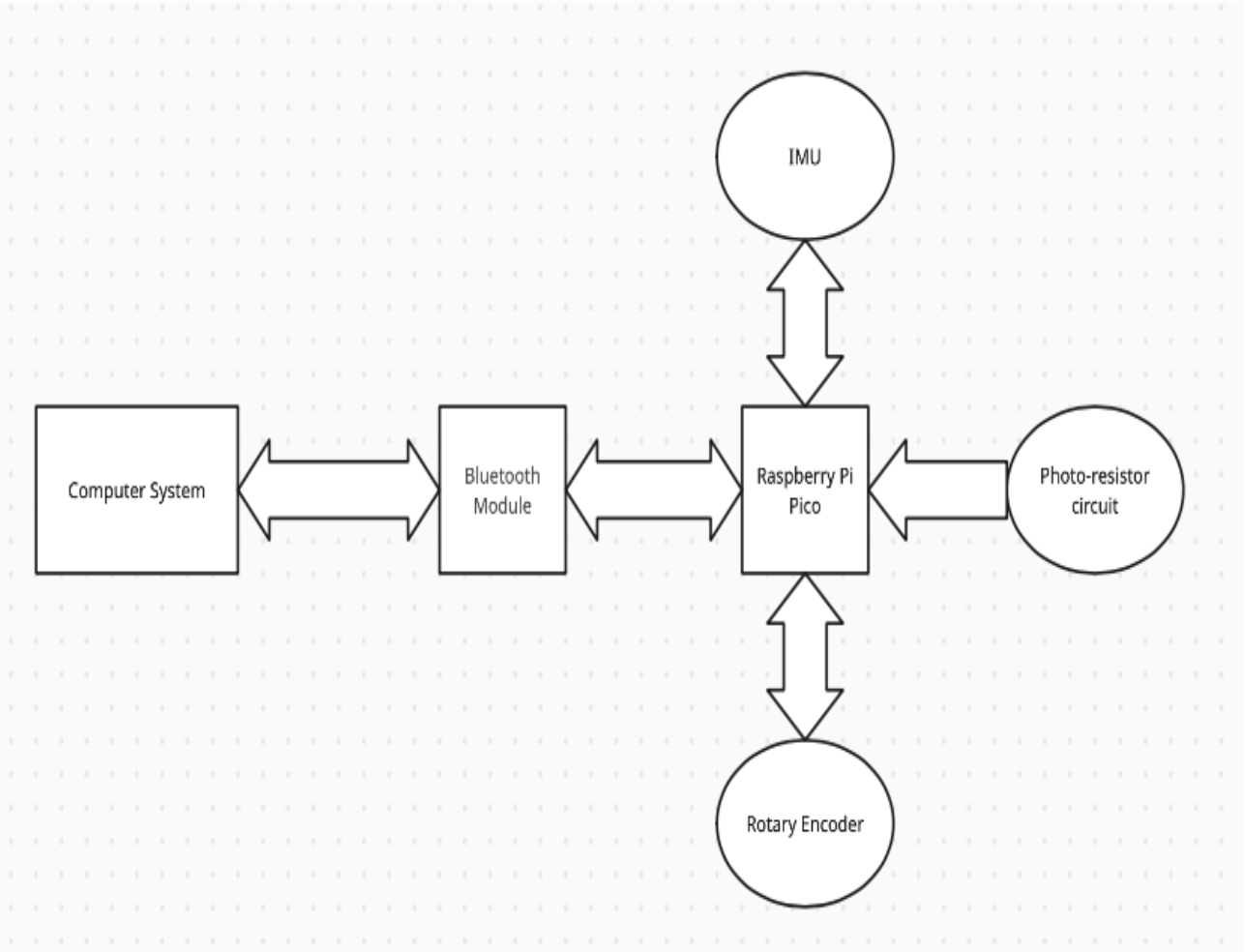
The problem at hand asks us to devise a mechanism that will remotely map the path/terrain a droid tracks. This is similar to the SLAM problem of localizing and mapping the surrounding as an autonomous vehicle moves. The droid being moved manually removes the problem of an autonomous drive. We only need to 'sense' our surroundings and transmit the readings to a base station. Alternatively, we could track the droid's movement, keeping track of its 'pose' and 'movement'. This will help us mathematically model the droid's motion in a 2D plane using Newtons's equations. Once we have an estimate of the path the droid traces, the mapping will involve plotting this on a graph.

Design Description

We have divided our design into several parts as follows:

- We will be using two different sensors for this project, a **Inertial Measurement Unit(IMU)** which includes a 6-axis gyroscope and accelerometer, and an **Rotary Encoder**. The purpose of IMU is to give us the orientation of the bot in order to track rotation. The optical encoder is used to obtain the speed of the bot which will be required to obtain X-Y coordinates.
- Additionally we will be using a photoresistor to plot the variance of ambient light as the droid traverses. This way we can plot a heat map of the ambient light intensity for different locations.
- A Bluetooth module will be used to transmit data to a computer for further computation. Python will be used for mapping.
- As per instructions, Raspberry Pi Pico MCU will be used to interface the sensors and Bluetooth module.
- The Bluetooth module and Raspberry Pi Pico MCU will be interfaced on the PCB along with the sensors, while the power will be supplied externally. All this will be packaged inside a four wheeled box.

Block Diagram



Project Plan

We have distributed the complete project in multiple parts:

1. Raspberry Pi Pico MCU programming using C SDK; **(Mudit Goyal and Keshav Patel Keval)**
2. Breadboard testing of the sensors, bluetooth module, and Raspberry pi pico MCU;**(Mudit Goyal, Rishabh Ravi and Keshav Patel Keval)**
3. Breadboard and PCB design;**(Rishabh Ravi)**
4. Python programming for processing data and generating map.(**Rishabh Ravi and Keshav Patel Keval**)
5. 3D-print a four wheeled box(**Mudit Goyal**)

A brief description of our plan of action:

1. After discussing with our respective RAs and faculty advisors, we will finalize the components for our project. (completed)
2. As soon as it's done we will submit our report containing a detail description of our design plans and start working on the above mentioned points parallelly. (completed)
3. Work will be allocated to the team members depending on their respective expertise. (completed)
4. We plan to first test it on a breadboard/development board and modify our project design accordingly , if required. Simultaneously, a PCB schematic will be made by Rishabh Ravi. If allowed then the initial testing can also be done on the Raspberry Pi kit available in the WEL lab. (Completed)
5. When we get a satisfactory result, we will start the testing on the PCB.(Completed)
6. We will then 3D-print a four-wheeled box of proper dimensions and begin testing. (to be completed)

Components Selection and Justification

The item list we seek to procure for completing the project is as follows:

1. **Raspberry Pi Pico**: The following microcontroller will be used to control

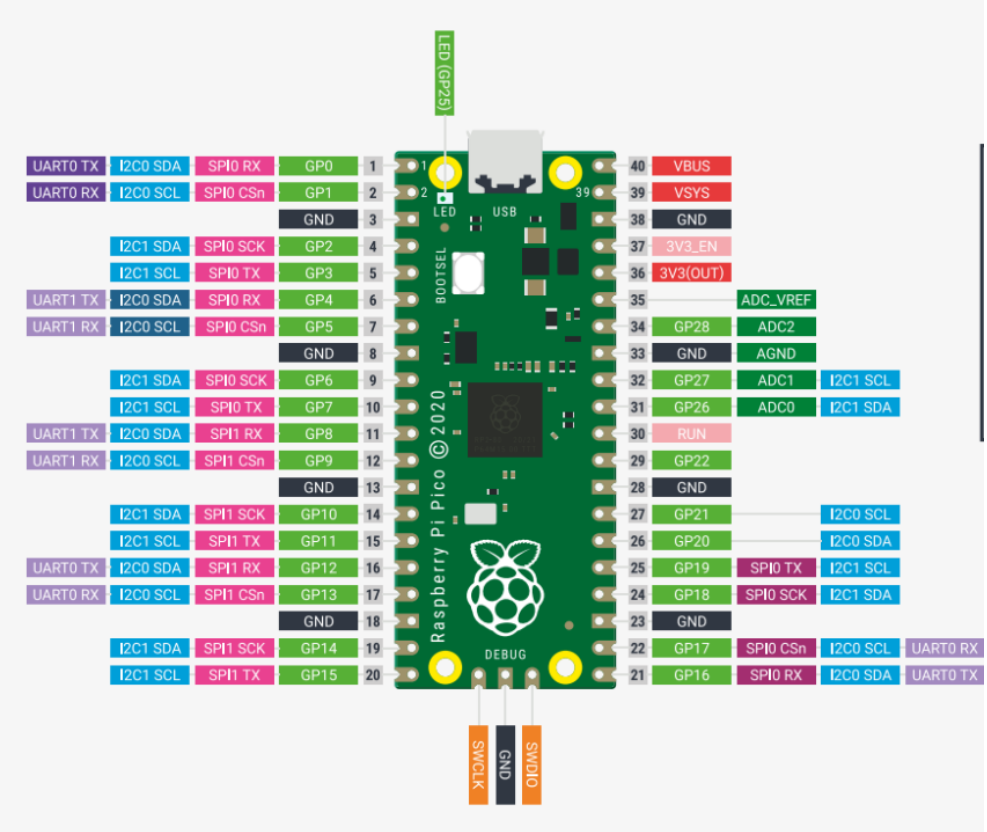


Figure 1: Raspberry Pi Pico MCU

operations on the PCB. It requires a 1.8 - 5.5V input. It comes with a 12MHz crystal oscillator, which will be its operating speed. It includes

- 2 × UART
- 2 × I2C
- 2 × SPI
- 16 × PWM channels

2. **Bluetooth module (HC - 05)**: In order to set-up a wireless communication with the base(laptop), a bluetooth module is being used. HC-05 module is chosen because it is very easy to use, and is pre-available in the WEL. It uses

Figure 2: HC-05

UART protocol, supported by Raspberry pi pico(MCU), to establish communication between the host and the base. The module can be directly powered by the MCU, thus the power requirements are also easily achieved.

3. **Rotary Encoder**: The KY-040 is a rotary input device (as in knob) that pro-

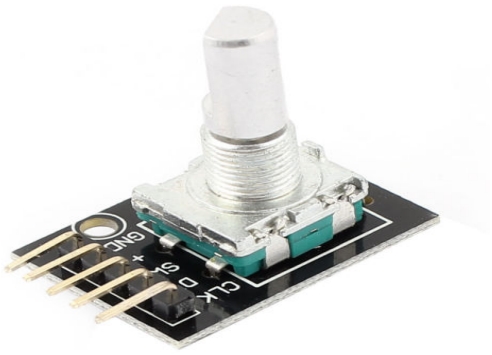


Figure 3: KY-040

vides an indication of how much the knob has been rotated and what direction it is rotating in. The output of M274 360 rotary degree encoder, provides information about the motion of the shaft, which can be easily processed by the MCU into relevant data like speed, distance, and position. Thus, making this a good choice for our project.

4. **Inertial Measurement Unit (IMU) (MPU-6500)**: Is used to obtain the remote droid's orientation. The module uses a 6-axis gyroscope to provide a higher resolution. This module directly outputs the orientation, which combined with the data of the rotary encoder allows us to find the position of the droid very easily. This module is easy to use and can be integrated with the PCB. The module takes input voltage of 3.3-5V and can be directly powered using the duracel batteries. An additional benefit of MPU6500 is that it can also output the ambient temperature readings.

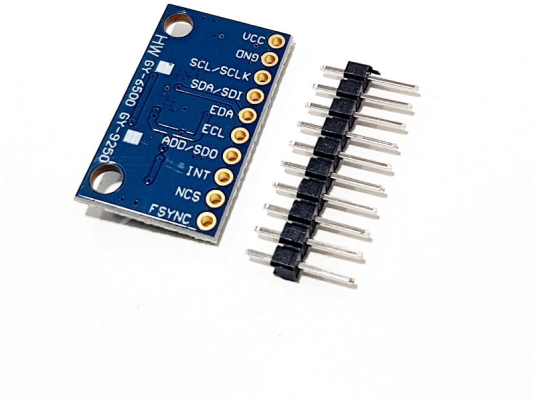


Figure 4: MPU-6500

5. **Power Source**: The maximum power requirements of the componentrs are as follows:

Component	Voltage(in V)	maxCurrent(in mA)
Raspberry Pi Pico MCU	3.3-5	450
IMU(MPU-6500)	3.3-5	3.4
Rotary Encoder(KY-040)	3.3-5	-

The bluetooth module is directly powered from the MCU and is taken into consideration by assuming that the MCU is drawing the maximum possible current. Thus, **3 AA Duracel**, i.e., 1.5V and 1800mAh each, would be enough to get the required voltage and will be able to power the device for a long duration(approx 3.5hrs). The above will require a module to stack the batteries and achieve 4.5V serially, thus a battery case is required.

6. **Packaging**: The PCB and all the sensors will be packed inside a four wheeled box. This box will be 3-D printed in the WEL. The design of the box and the working of wheels is inspired from the hot wheels. Basically, the whole

package will only be able to move in 1-D but can be moved on curved paths by making very small incremental linear steps and the radius of curvature will depend on the dimensions of the box and wheels. The dimensions of the box would most likely be 120mm x 90mm x 60mm, with wheels of diameter 30mm.

Principle of Operation for each Subsystem

1. **Wireless Communication**:
   This subsystem is about the communication between the Raspberry Pi Pico MCU(base) and the host(laptop) via bluetooth(HC-05). The block diagram of this subsystem is as follows:

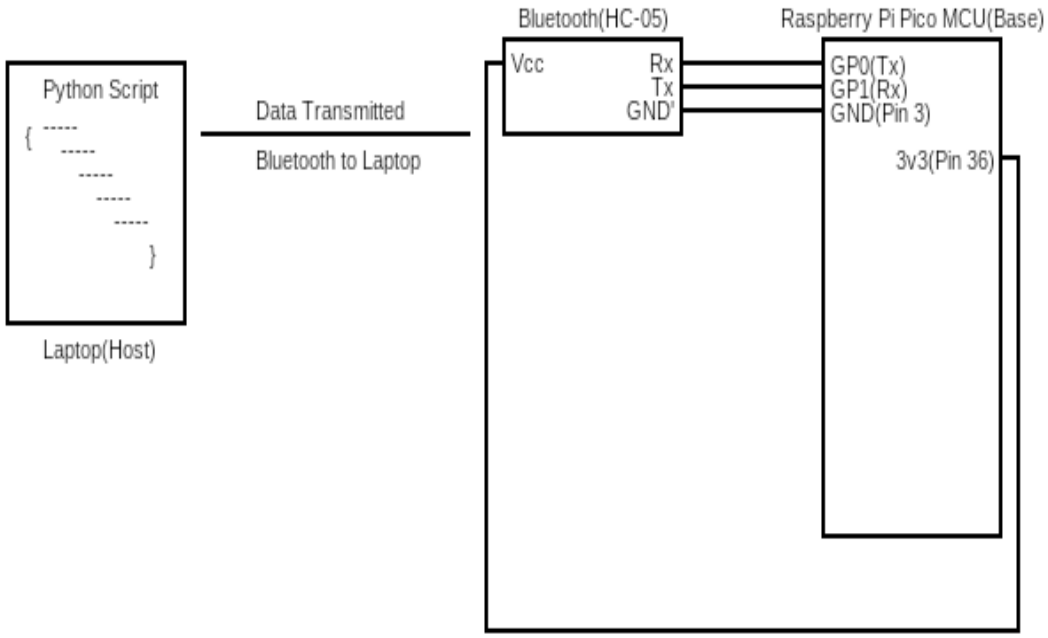


Figure 5: block Diagram of wireless connection between the base and host. This includes a Raspberry Pi Pico MCU, Bluetooth(HC-05), laptop(running a python script)

(a) **MCU and HC-05 connection:** The Rx pin of HC-05 is connected to the GP0, i.e., Tx pin of MCU, while Tx pin of HC-05 is connected the GP1, i.e., Rx pin of MCU. The pin 3 of MCU is GND which is connected to the ground of HC-05 and pin 36 of MCU is used to provide 3.3V of supply to HC-05. These connection will be mapped on to the PCB as shown in the schematic given above. UART protocol is used for communication with baud rate of 9600(usual standard).

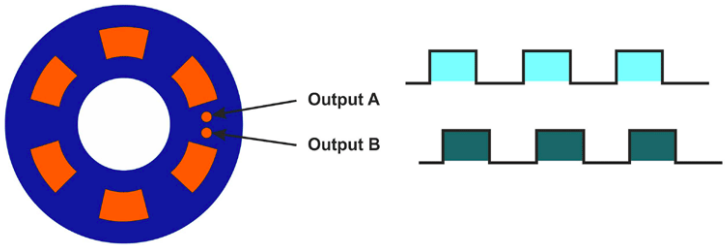
(b) **HC-05 and Host(Laptop) connection:** Serial communication using bluetooth protocol RFCOMM is used to enable communication between HC-05 and the laptop. A python script is written to do the above and data is exchanged with the MCU in real-time. The rfcomm0 port on laptop will exchange data with the base through HC-05 and will continue to do so until there is no activity on rfcomm0 port for 5 seconds continuously, after which the port will assume that no further data is to be exchanged and will close the connection.

2. **Sensor Subsystem**:
   The sensor subsystem consists of the following 3 devices :

   (a) **Inertial Measurement Unit (IMU)** : The IMU selected is MPU-6500, which is 6-axis motion sensor. The MPU-6500 has triple-axis gyroscope, which allows us to measure the angular velocity in full-scale range of ± 250, ± 500, ± 1000, and $\pm 2000^{\circ}$/sec. In addition to this, the gyroscope also has a digitally programmable low-pass filter. The operating current of the gyroscope is 3.2mA. The MPU-6500 also has a triple-axis accelerometer which allows us to measure the linear acceleration in full scale range of ± 2g, ± 4g, ± 8g and ± 16g. The normal operating current of the accelerometer is 0.45mA. Both the gyroscope and the accelerometers have 16-bit ADCs. In addition to a gyroscope and accelerometer, the MPU-6500 also has a Digital-output temperature sensor. The MPU-6500 also has a Digital Motion Processing (DMP) engine that outputs the Angular position of the device. The principle of working of the device is briefly explained in the next few lines. When the IMU is rotated, the vibration caused due to Coriolis effect is measured by the capacitive pickoff. This signal is filtered, processed and amplified to give a signal proportional to the angular velocity. The accelerometer has separate masses to measure the acceleration along each of the axes.Acceleration along a particular axis induces displacement on the corresponding proof mass, and capacitive sensors detect the displacement differentially.

   (b) **Rotary Encoder** : The rotary encoder selected is KY-040. The KY-040 rotary encoder is a device that generates an electrical signal based upon

how much the rotary input device (knob) is rotated and the direction it is rotating in. It is a position sensor with a knob and is usually used to control stepper or servo motors with precision, however, in this project it will be used to track the position of our device as it moves along an arbitrary trajectory. The KY-040 rotary encoder can be used for precise determination of the movement. The principle of working of the device is briefly explained in the next few lines(a diagram is also given below).

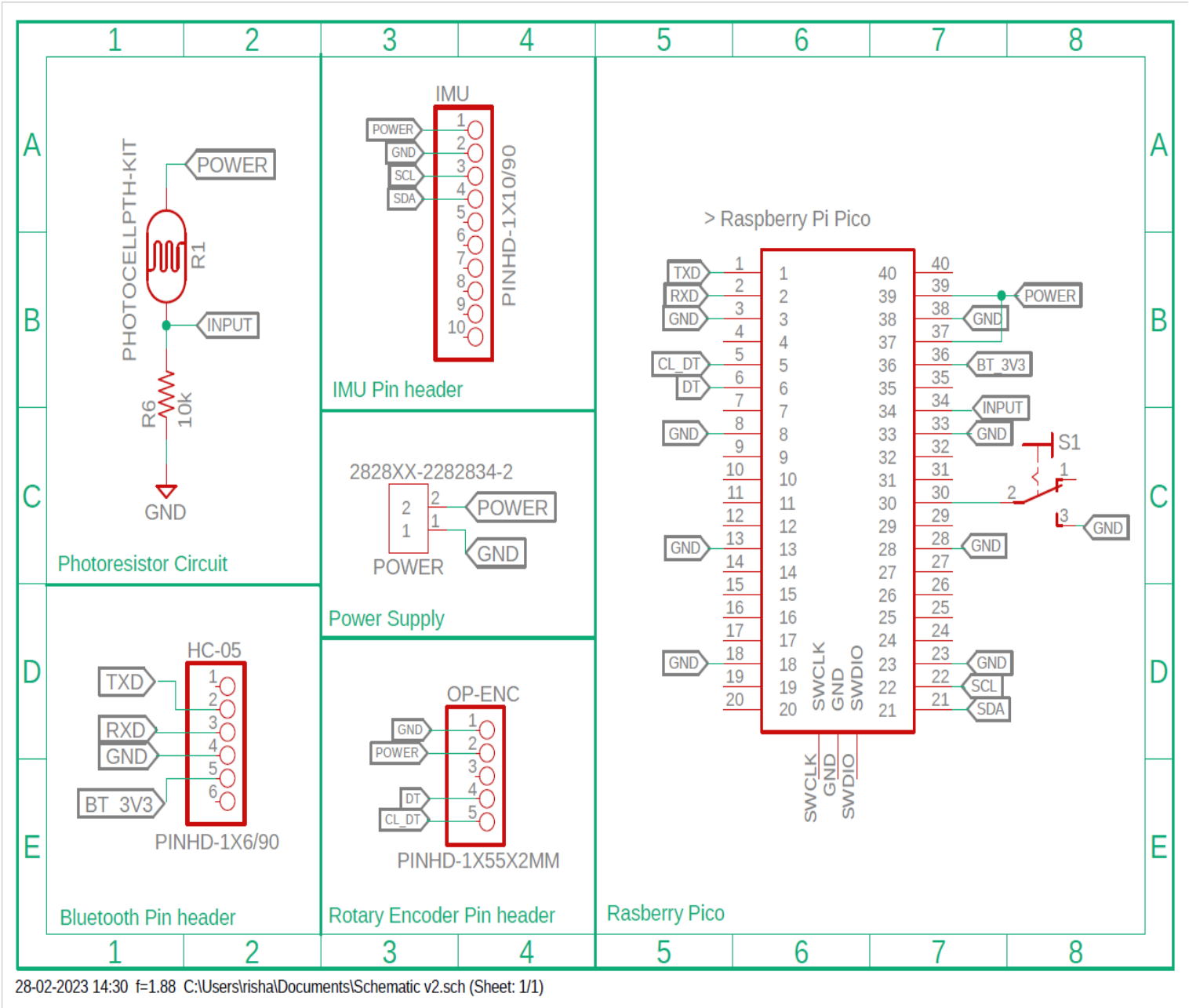


The encoder has a disk with contact pints evenly placed from each other which are connected to the common Ground. As we start rotating the knob, the disk also moves and makes contact with the output pins A and B one by one, hence generating two square waves simultaneously. Simply by counting the number of square waves generated, the rotated position can be determined. If output A changed states first, the switch is rotating in a clockwise direction, whereas if output B changes its state first, we know that the switch is rotating in a counter-clockwise direction.

(c) **Photoresistor** : The photoresistor selected for this project is the LDR (20mm in size). The resistance of the photoresistor changes based on the amount of light incident on it. We will measure the current passing through the photoresistor and use it to indirectly measure the amount of light incident on the device. This shall be mapped along the trajectory traced by the iPEC-probe. This is to test if we are able to get a 3D-plot of some quantity that varies along the trajectory of the iPEC-probe. Once completely integrated with the iPEC-probe, the device developed by us would map the probe readings along the trajectory. The working principle of the Photoresistor is mentioned in the next few lines. When light falls on a photoconductive material, it is absorbed by it which in turn creates free electrons from the valence electrons. As the amount of light incident on it increases, the number of free electrons increases, thereby increasing the conductivity of the material, and lowering the resistance of the material.

Preliminary Circuit

Preliminary Circuit Schematic



The schematic shown above was made referring to the data sheets and online tutorials of all modules used by us. It was subject to changes after breadboard testing.

Preliminary Schematic Description

- The pin headers were chosen to suit each component and the connection on each header is matched according to the connections on the breakout board of the component.
- The photoresistor circuit has a 10k resistor to drop the entire power supply across it if the photoresistor has a low resistance. Alternatively, if the photoresistor has high resistance, the potential drops across it, and the input would read 0V.
- The connections to the Pico were done based on the results of breadboard testing.
- An additional switch is added for resetting the microcontroller.

Bill Of Materials(BOM)

Sr. No.	Item name	Category	Quantity	Total Cost	Source
1	MPU-6050	Inertial Measurment Unit	1	₹227.00	Link
2	HC-05	Bluetooth module	2	₹227.00	Link
3	KY-040	Rotary Encoder	1	₹54.00	Link
4	Raspberry Pico MCU	Microcontroller	2	₹331.00	Link
6	4 x Duracell AA cell	Battery	1	₹171.00	Link
7	Battery Holder	Battery Holder	1	₹13.00	Link
9	LDR 20mm	Photoresistor	5	₹63.00	Link
Total Cost =₹1086.00					

Miscellaneous

1. The pin headers, female pin headers, and the jumper wires consumed were provided by the WEL.
2. The box required for the packaging and the appropriate wheels for the will be made using 3-D printer provided by WEL-Lab.

Preliminary analysis

The only available components with us were: Raspberry Pi Pico MCU and HC-05. Hence, we started testing the bluetooth module and its interfacing with Raspberry Pi Pico on the breadboard.

1. **Mobile as Host**: We flash a basic code, that will print a string communicated over bluetooth by using UART at 9600 baud rate.
   A terminal emulator-'**Serial Bluetooth Terminal**' app was used to receive the data-"This is the first test HI", on an **android device**, and these were the results:

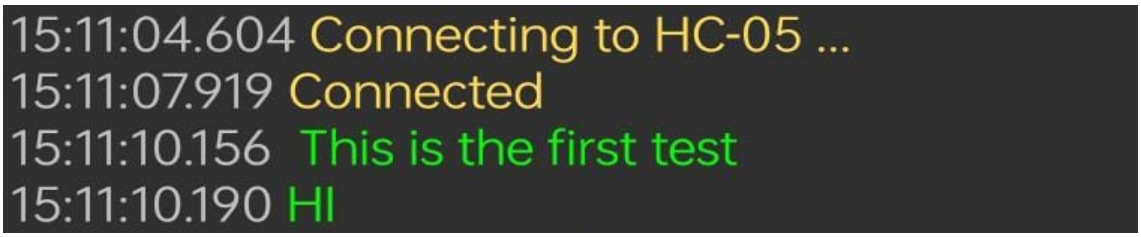
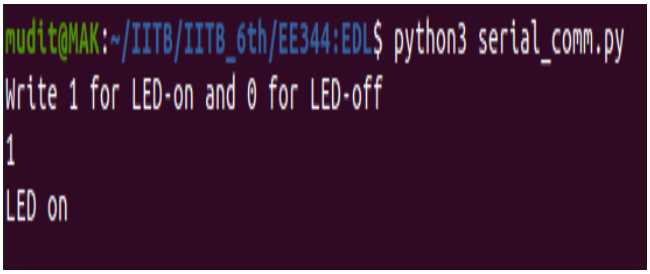
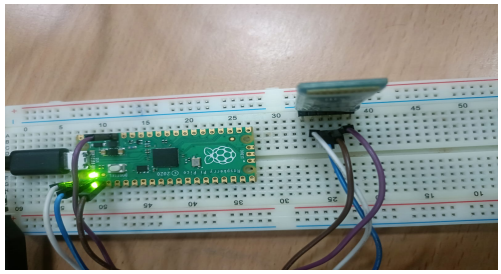


Figure 6: Bluetooth Test 1

2. **Laptop as Host**: After sucessfully receiving data on an android device, we decided to switch the host to laptop because that is our final goal. To achieve this, a **python script** was used by the host to establish connection with MCU over bluetooth, so that the host could send and receive data to the MCU. To test this connection, a **C script** was flash on the MCU which will turn on the led if the MCU receives 1 while the led will turn off if it receives 0. After multiple trials and errors, in the end the following was observed:
   In **figure 6** when the host(laptop) sends '1', the led on the Raspberry Pi turns on and sends back-"Led on" to the host, and is received in real-time by

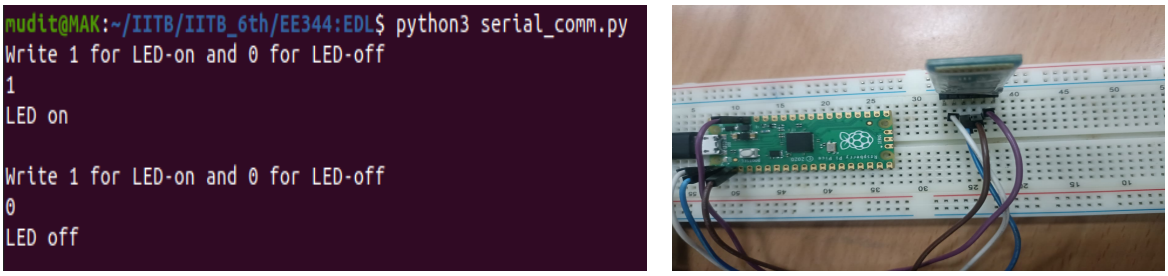


(a) Host



(b) Base

Figure 7: LED-ON state



(a) Host (b) Base

Figure 8: LED-OFF state

the host. Now, the LED will remain on until host sends '0' which will turn LED off and the base will send "LED off" back to the host and the output will be as shown in **figure 7**.

Subsystem testing

Wireless Communication

During the preliminary analysis, since only the Bluetooth Module was available to us, we tested it using the serial communication application on a mobile phone. For the current milestone, we tested the Bluetooth module by making it send data and receiving it on a computer device. The following for the testing of the HC-05 Bluetooth module. The python code written by us receives the data sent over Bluetooth and stores the data in a csv file on the computer system. The Pseudo-code for receiving the Bluetooth data on the computer device is given below:

Algorithm 1 Receiving data sent from the Bluetooth Module (HC-05)

Ensure: Baud rate is 9600
    $start\_time \leftarrow time.now()$
    while True **do**
        while serial_port_in_waiting() **do**
            $distance \leftarrow 0$
            $theta \leftarrow 0$
            $z\_val \leftarrow 0$
            $time \leftarrow 0$
            $CSV\_columns \leftarrow [`Distance', `Theta', `Time', `z\_value']$
            $Open\_file(`Data.csv', fieldname = CSV\_columns)$
            while True **do**
                $ser\_data \leftarrow serial.readline()$
                $time \leftarrow time.now() - start\_time$
                $distance, theta, z\_val \leftarrow ser\_data.decode\_unpack()$
                $Write\_CSV(filename = `Data.csv', data = [distance, theta, z\_val, time])$
            end while
        end while
    end while

Sensor Subsystem

Inertial Measurement Unit (IMU)

For the Initial testing of the IMU, an embedded C code(**main.c-provided on our github**[1]) was flashed on R-Pi to measure and receive the angle measurements: roll, yaw, pitch, and saved the data on the CSV file, as mentioned before. We need

[1]https://github.com/borlaugg/2D-mapping.git

only one angle measurement for the purpose of theta and we have chosen roll as the standard because among yaw, pitch ,and roll, roll data had the least fluctuations. We then plotted sin(roll) versus time. We got the following plot during the testing: **Video link of the complete test:imu.mp4**

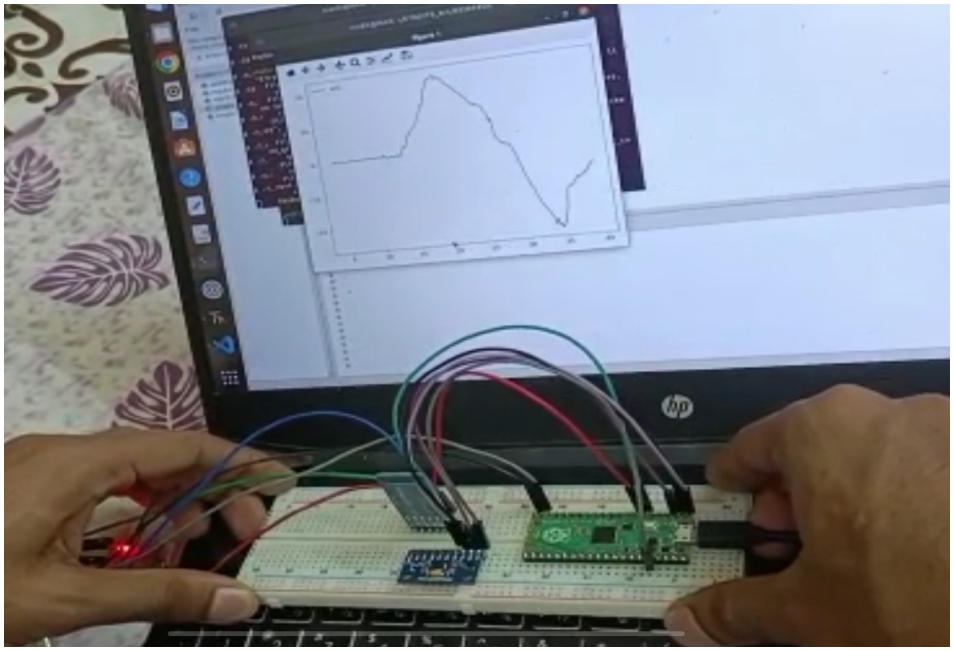


Figure 9: sin(roll) vs time plot obtained during the testing

Rotary Encoder

The rotary encoder has a total of 4 pins, V_{cc}, GND, CLK and DT. CLK and DT pins are used to read the incoming sensed data. CLK is a clock pin that produces square pulses every time the wheel rotates 12 degrees. The DT pin produces a similar signal but is delayed by a small amount based on the direction of rotation. Also, the rotation speed determines the pulse width of the CLK and DT signal.

In the figures below the blue signal is the CLK port and yellow refers to the DT port. During clockwise rotation, CLK leads DT and vice versa during anti-clockwise rotation.

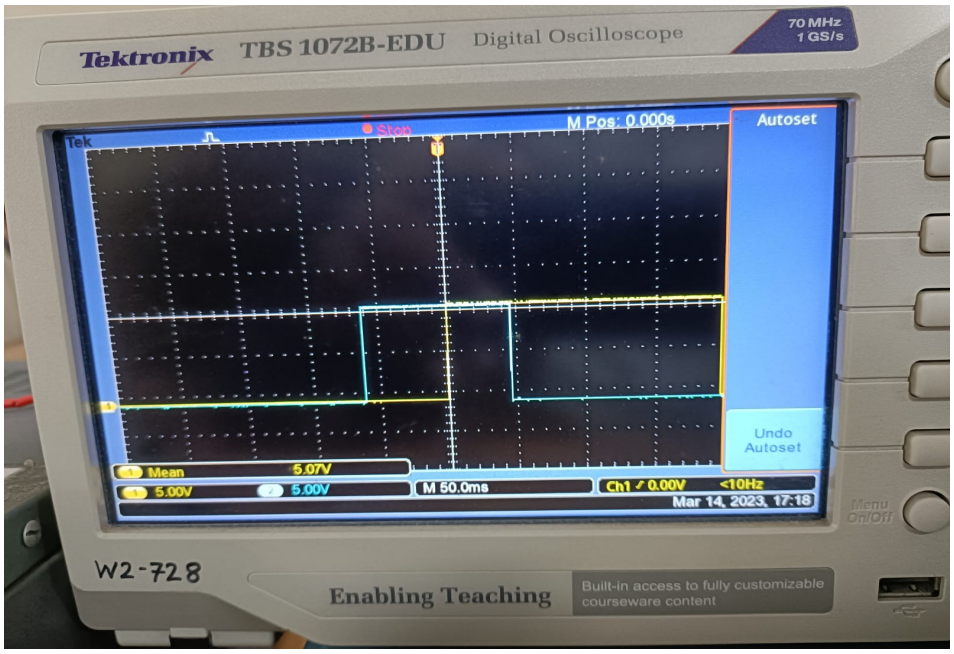


Figure 10: Low-speed clockwise rotation

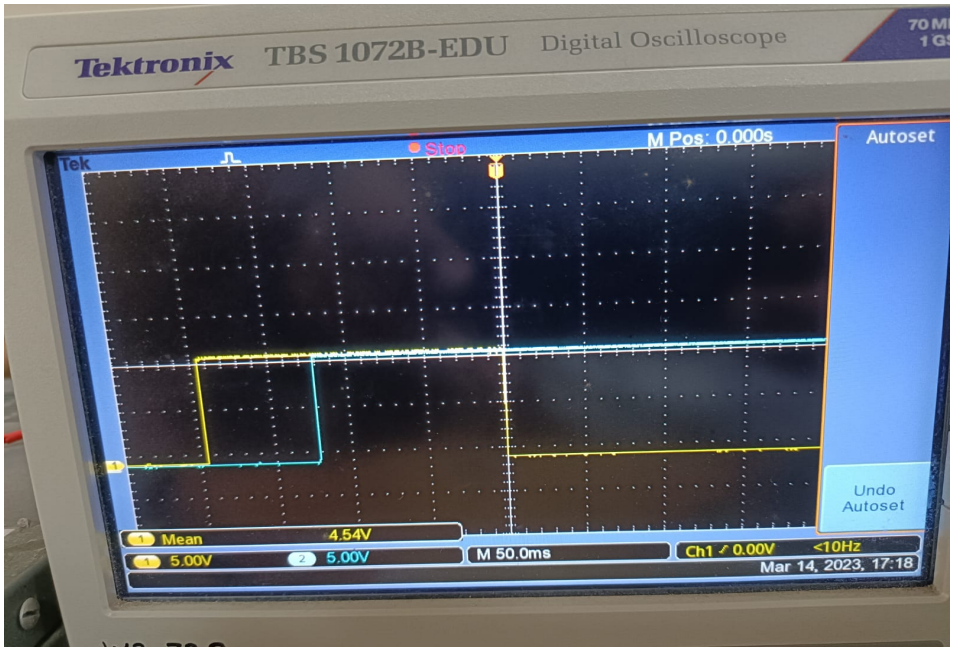


Figure 11: Low-speed anti-clockwise rotation

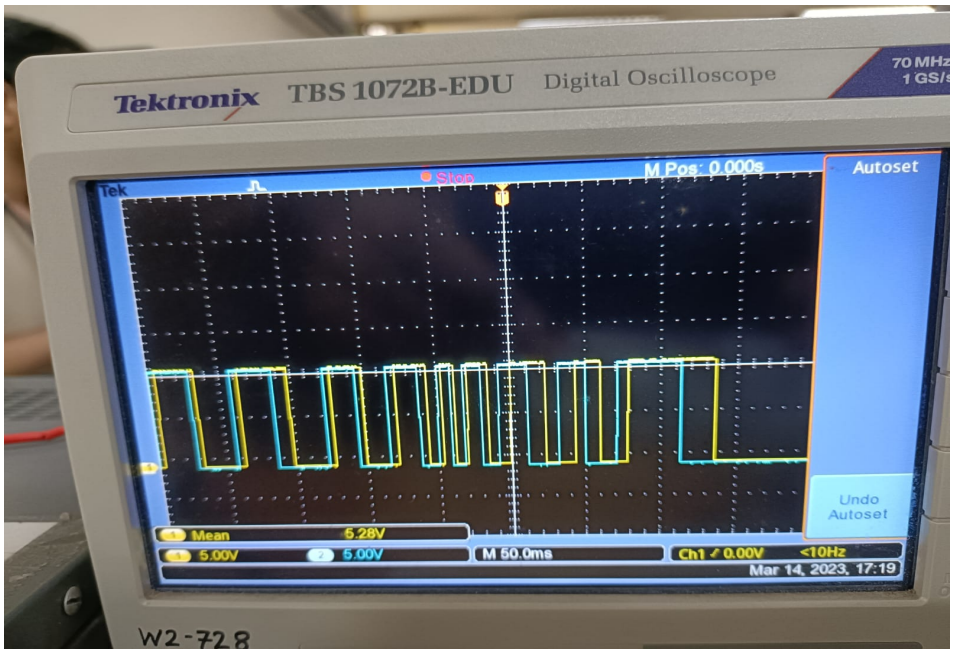


Figure 12: High-speed clockwise rotation

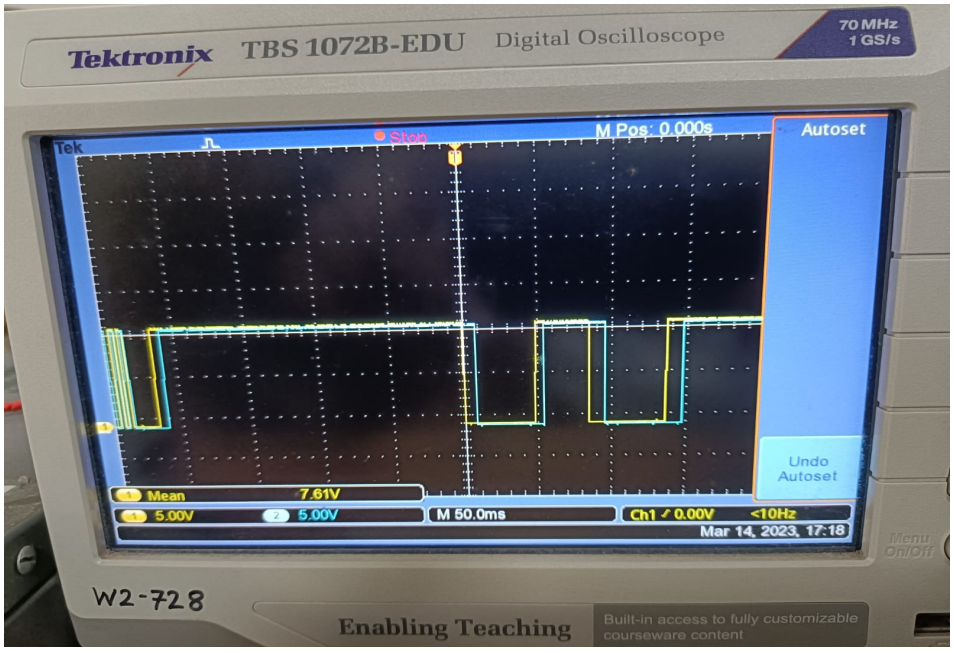


Figure 13: High-speed anti-clockwise rotation

The computation of the distance involves following steps

1. Capture the CLK and DT signal on a rising edge of the CLK signal
2. If the CLK signal's digital value matches DT's value, the rotation is clockwise, else anti-clockwise.
3. Increment a counter(ctr) on every clockwise rotation and decrement on an anti-clockwise rotation.
4. The distance traversed is 30*ctr*π*D/360.

Photoresistor

The LDR is a light-sensitive device whose resistance changes when different light intensity levels fall. When put in a circuit as shown below, the input signal reads a range of values from 5V to 0V, which is a decreasing function of the intensity of the light shown.

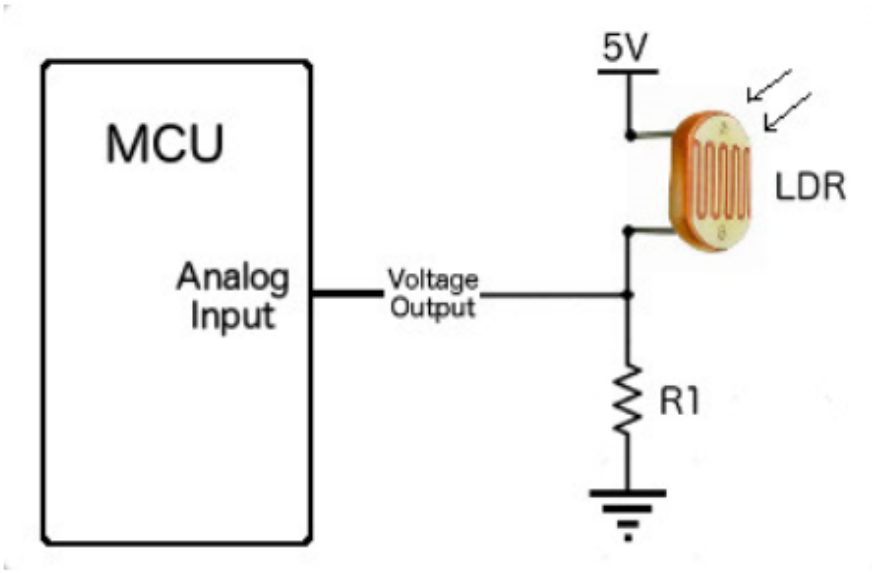


Figure 14: LDR circuit

After rigorous testing, we have chosen a resistance of 10kΩ.

Plotting the Data obtained from the Breadboard design

We tested the Breadboard design consisting of the Raspberry Pi Pico, Bluetooth Module(HC-05), the IMU(MPU-6500), Rotary Encoder(KY-040) and the Photoresistor (20mm LDR). We received data sent by the Bluetooth module on the computer system using a python script **serial_comm.py**, and plotted the real-time data using another python script **EDLplot.py**. All the source code are provided on our github[2]. We also made a **video demonstration.mp4** link.
The following points are worth noting :

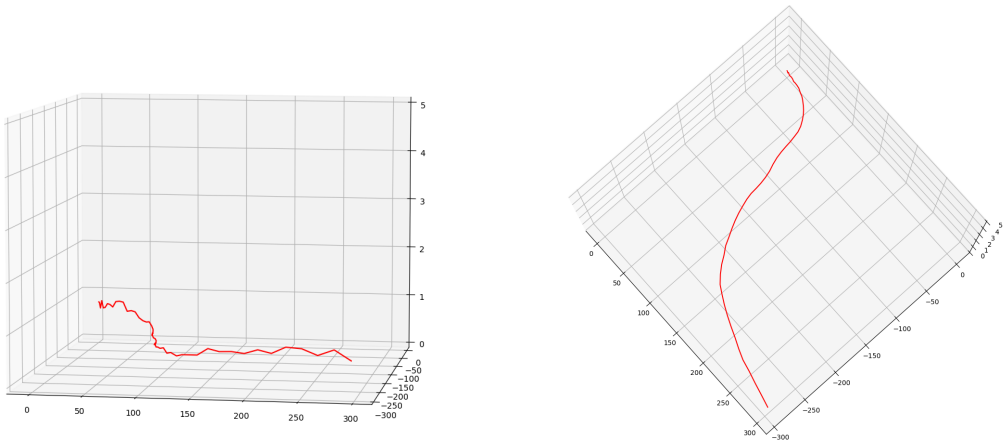
1. While we were testing, we noticed that the z-value (Voltage across the Photoresistor) fluctuated quite a lot, despite no changes in the lighting conditions. To prevent such fluctuations, at the cost of accuracy, we can possibly introduce a geometrically averaging window for the z-values as follows:(Where λ is the geometric discount factor)

$$z[t] = \frac{(1-\lambda)}{1-\lambda^W} \sum_{s=t-W+1}^{t} \lambda^{t-s} z[s], t > W$$

   (For $t \leq W$, we take the geometric average of only the values that are available during the current window)

2. We need to test whether the plots obtained are upto the mark. The procedure by which we do this is give under the section "Evaluation Metric for the plots".

We got the following plot (viewed from two different angles):



(a) Side View (b) Top View

Figure 15: Side View and Top view of the plot. Z-axis has the voltage (in V) across the Photoresistor(0 to 5V), x-axis from 0 to 300 and y-axis from 0 to -300

[2]https://github.com/borlaugg/2D-mapping.git

Evaluation Metric for the plots

Our 2D mapping system should map the trajectory of the iPEC-probe, while plotting (along the z-axis) some third quantity (ideally, the iPEC-probe readings). For the purpose of the project we have chosen to use a photo-resistor to measure the light intensity along the 2D trajectory and plot the voltage across the photoresistor. To test whether the trajectory is mapped properly, we shall do the following :

1. Check whether a set of basic trajectories can be mapped, such as a straight line, and a curved path.

2. To measure the resolution of the mapping, we plan to make the device move in a series of straight lines of increasing length (x) (4cm, 6cm, 8cm, 10cm ... 10 to 20 values) and measure the corresponding distance (measured in the same unit)plotted in the graph (y). We shall plot y versus x, and measure the slope of the line of best fit (ρ)of this graph. Ideally the slope should be 1. So the resolution of the mapping can be measured by ρ, closer it is to 1, better the resolution of the mapping.

(The first is to make sure that at the very least, our device can map the basic shape of the trajectory of the iPEC-probe)

Final Schematic and Board

Following the breadboard tests, a schematic was made and routed.Errors from the preliminary schematic were fixed. It was thoroughly reviewed by the faculty at WEL and given for printing. The schematic displayed below has been decomposed into its constituting circuits. Since most of the components are available only as a breakout board. The different subcircuits are

1. The **photoresistor circuit** and the corresponding choice of resistance 10. It has connections to power, ground, and an 'input' routed to the raspberry-pico.

2. The **IMU** circuitry. As the IMU was purchased as a breakout board, all that was required here was to add a pin header and provide the corresponding connections from the microcontroller. Here it uses the 'I2C' communication protocol to communicate, and hence two lines - SCL and SDA are present that are connected to the microcontroller. Furthermore, there are power and ground connections to power up the IMU.

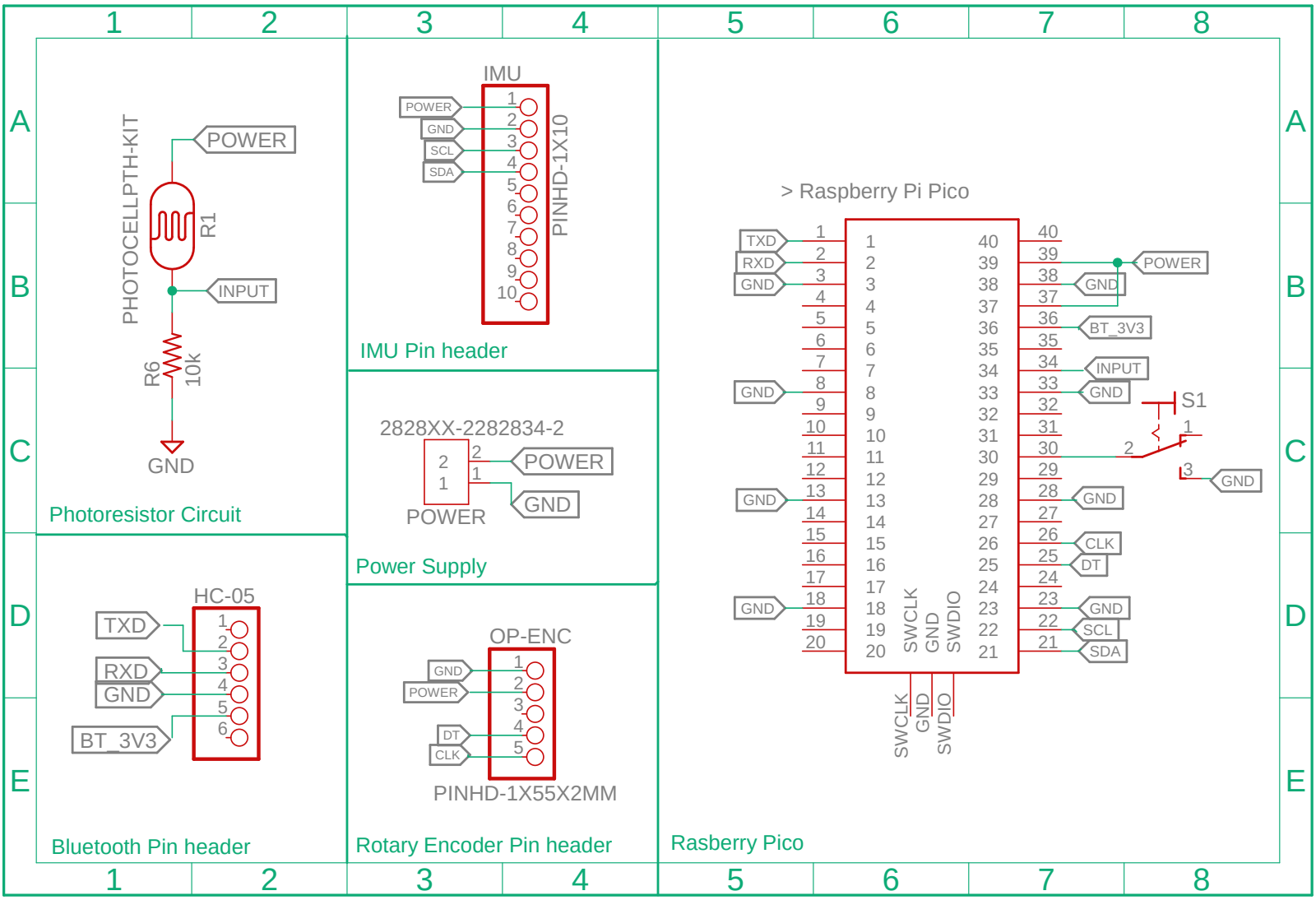
3. The **power supply** circuitry. This is where the board receives power from the external batteries. All power and ground connections are drawn from here. They correspond to the positive and negative terminals of the battery, respectively.

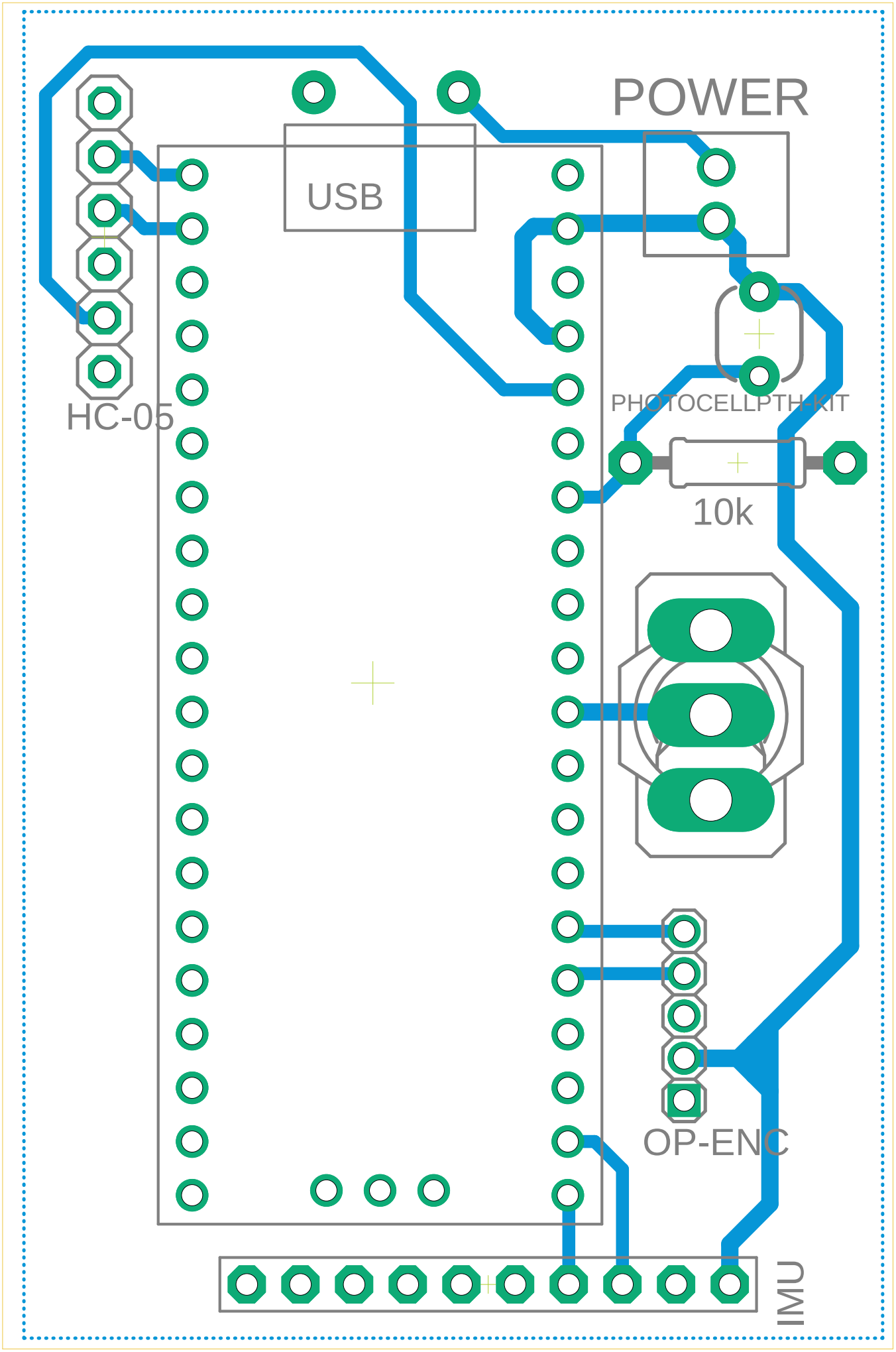
4. The **Bluetooth** circuitry. Once again, as the module was purchased as a breakout board, we were only required to provide a pin header of appropriate length and connections. The BT3V3 and GND connections are to power the breakout board. As the module requires a 3.3V power supply and the battery provides a 4.5V supply, it has been connected to the 3.3V power supply pin on the raspberry-pico. Moreover, it communicates using the 'UART' protocol, and the RX-TX pins have been mapped to the TX and RX of the microcontroller.

5. The **Rotary Encoder** circuitry. As described in the breadboard test, it requires a CLK and DT pin connected to the I/O pins of the microcontroller. Power and ground have been connected to provide power to the encoder.

6. **Raspberry Pico** circuit. This is the heart of all connections. The pin choice was made to minimize the wire lengths. A switch has been added to enable reset. Toggling the switch grounds the pin, and the microcontroller resets. This, too, has power and ground connections to power it up.

. A ground plane was added to reduce the number of connections on the PCB. Also, a ground plane seemed the better option to reduce electrical noise and interference through ground loops and prevent crosstalk between adjacent circuit traces.


1
2
3
4
5
6
7
8
A
B
C
D
E
PHOTOCELLPTH-KIT
POWER
R1
INPUT
R6
10k
GND
Photoresistor Circuit
IMU
POWER
GND
SCL
SDA
PINHD-1X10
IMU Pin header
2828XX-2282834-2
POWER
GND
Power Supply
HC-05
TXD
RXD
GND
BT_3V3
Bluetooth Pin header
OP-ENC
PINHD-1X55X2MM
Rotary Encoder Pin header
> Raspberry Pi Pico
SWCLK
GND
SWDIO
S1
Rasberry Pico


POWER
USB
HC-05
PHOTOCELLPTH-KIT
10k
OP-ENC
IMU

Evaluation of soldered PCB

The printed PCB was soldered and tested for shorts. This was done to ensure that the soldering was done precisely, without shorting neighboring pins. Also, care was taken that while soldering the pin was not shorted to the ground. All of this was done by measuring the resistance between the two points being checked. A short, if present, was indicated by 0 resistance. It was resolved by de-soldering and carefully re-soldering it back. Once it was ensured the pins were isolated, and a short existed between connected pins, the PCB was sent for testing.

Testing on PCB

The PCB connections were made as they were done in the schematic. The images are shown below.

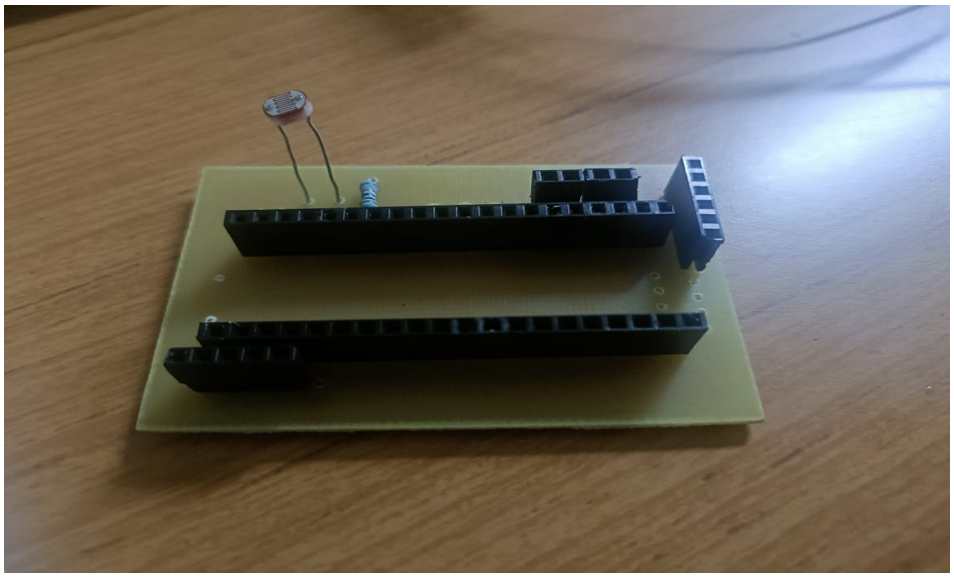


Figure 16: PCB

The rotary encoder is connected to the PCB through jumper wires because of the physical constraint of putting the shaft of the rotary encoder inside the wheel.

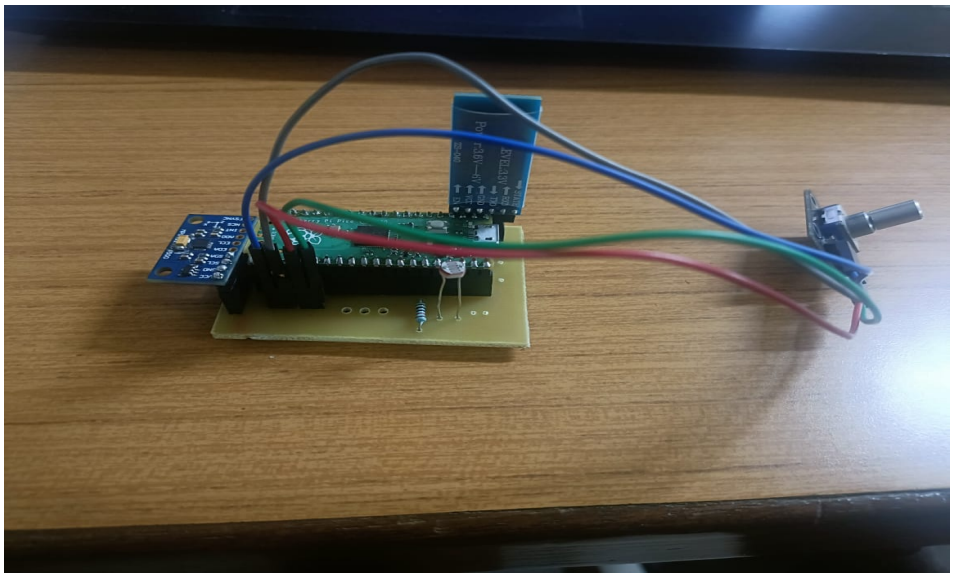


Figure 17: complete PCB

The setup was tested by moving the PCB and the encoder and observing the plots. The plots were similar to that of the breadboard testing and this helped us conclude that the PCB was perfectly functional.

Testing Status

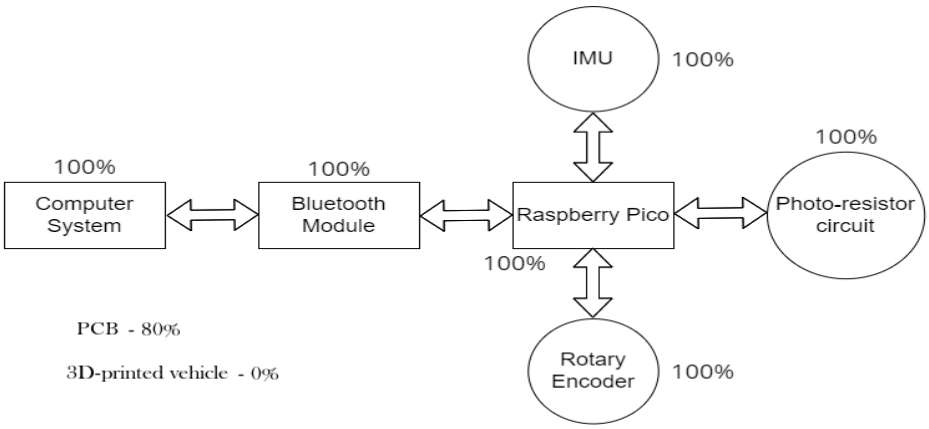


Figure 18: Testing status indicated by percentage next to each block

CAD Design

Shaft and Axle

The body of the robot will be constructed using laser cutting grey transparent acrylic sheets and the model was made on 'makercase', while axle shaft for the wheels will be 3D printed and it's model was made using 'Autodesk Fusion 360'. The holes will be drilled in the main body to insert the axle shafts. The physical constraints which were taken into account while modelling were the physical dimensions of the 2A cell battery holder and the PCB.

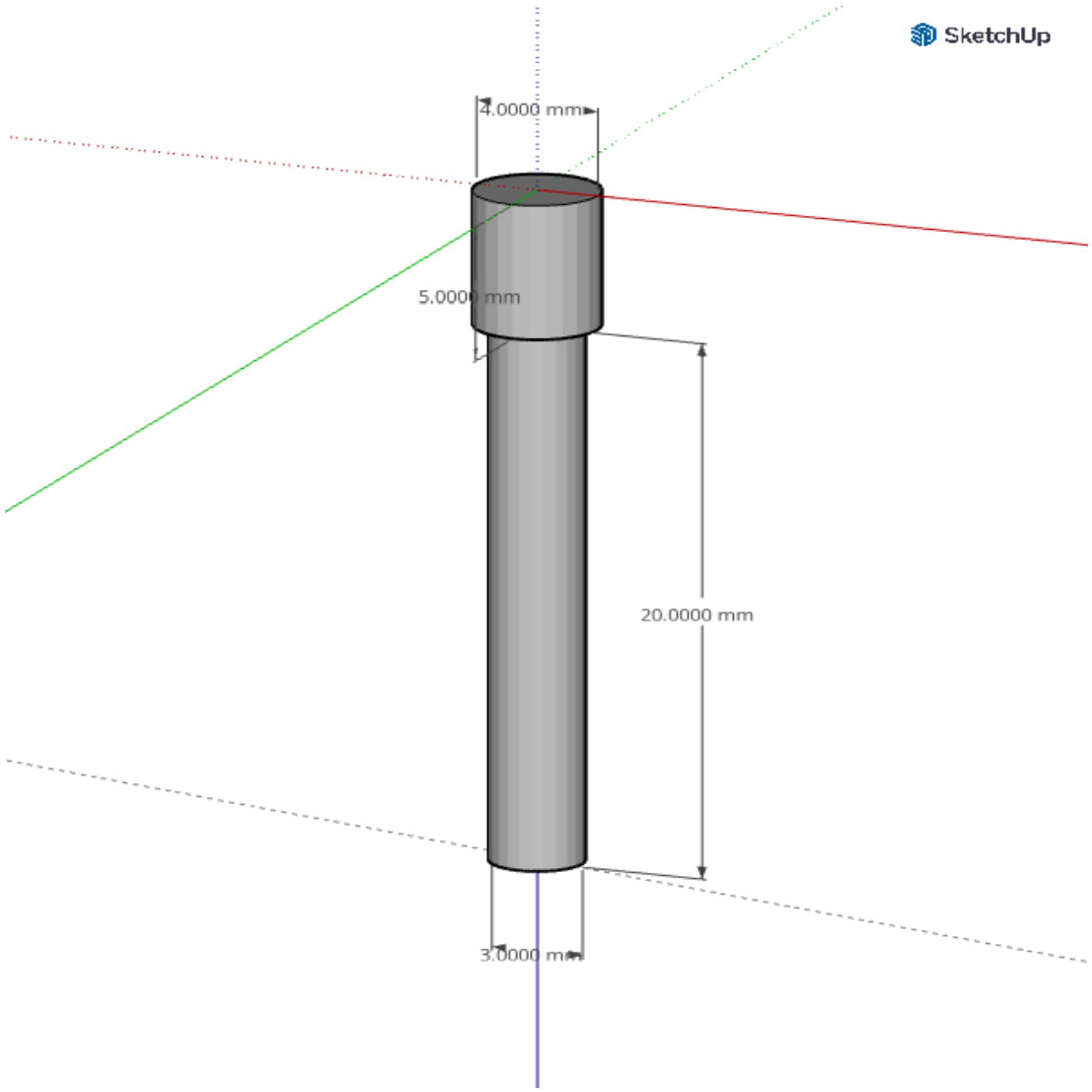


Figure 19: Axle

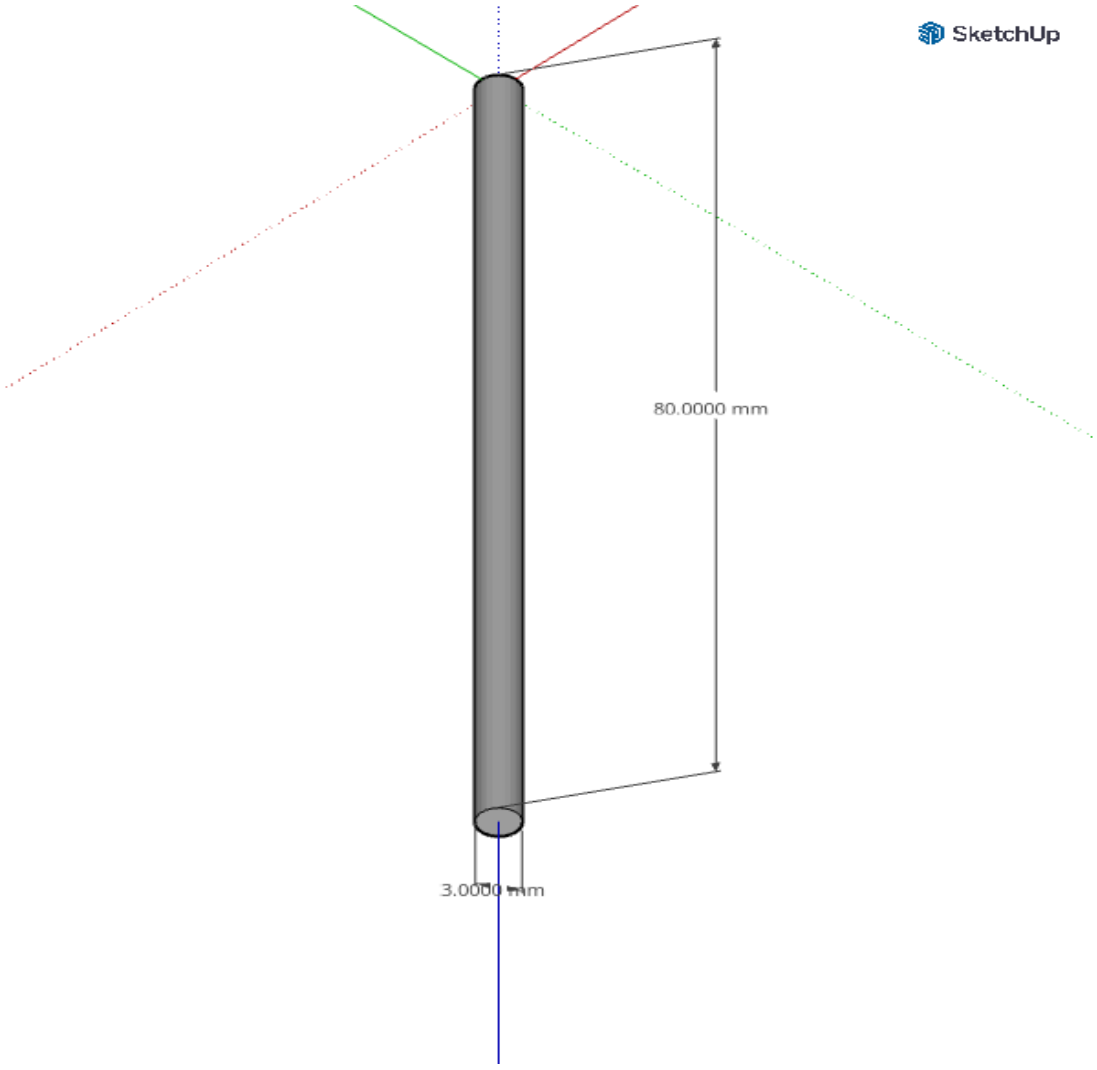
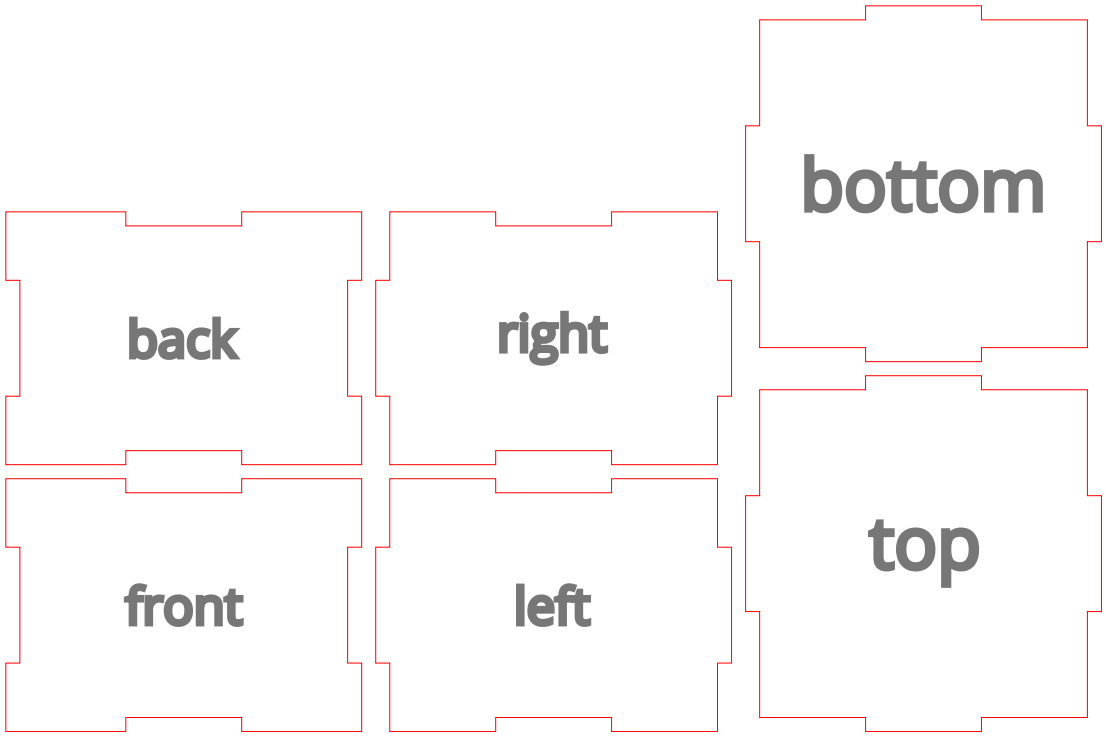


Figure 20: Shaft

Box

The material used will be **transparent acrylic sheets** of **3mm thickness** and physical dimensions of box are:

Dimension	Measurement(in mm)
Height	60
Depth	90
Width	90



Dissimilarities with other groups

The 2D mapping project is also being pursued by the TUES-04 group. The approaches taken by the teams are quite different. The following summarizes the **differences** in our approach:

1. The other team (TUES-04) has decided to use joytsticks to control the device and motors for driving the device. We do not have any such motor as our device will be moved manually (ideally it will be added to the iPEC-probe and will move along with it).

2. There are differences in the sensor modules used. We use MPU-6500 as the IMU,whereas the other team use MPU 9250. We use the KY-040 rotary encoder, while the other team use the H22A1 position encoder. In addition, they also use the L293D motor driver.

Risk Mitigation Strategies

Anticipated problem	Likelihood of problem occuring	Mitigation Strategy
The IMU (MPU-6500) does not function as expected.	Low	Since, this means that we cannot measure the angle of orientation of the device, we have to perform Raster scan of the surface and store the readings on the 2MB flash provided in the Raspberry pi pico. We can then send this data to the computer system over Bluetooth and after processing, display the 2D-map of the trajectory and light intensity(or some other quantity) along the third axis of the plot.
The photoresistor (LDR) does not work as expected.	Low	We have a temperature sensor on the IMU (MU-6500), we can use the these measurements and plot them along the third axis of our plot to demonstrate the mapping ability of our device.
The Bluetooth module (HC-05) does not work, or fails to communicate with computer system.	Low	We shall first try to replace the bluetooth module with some other compatible module. In case that does not work, we shall replace the Raspberry pi pico with the Raspberry pi Zero W, which has an onboard Bluetooth as well as WiFi module. In this case we would also have to change the PCB design slightly, hence we will try to avoid doing this.
Rotary encoder(KY-040) does not work.	Low	We shall use the linear acceleration measurements from the IMU to determine the position of the device. This will give a much less accurate position, hence we will also try to avoid this.

Mechanical Issues Encountered and their fixes

We have encountered the following issues that need to be resolved :

1. The wheels that we ordered over the internet are not compatible with the rotary encoder that we have used. This is because the hole in the wheel is smaller than the cross-section of the rotary encoder shaft. However, this was solved by expanding the hole in the wheel so that the rotary encoder shaft fits appropriately.

2. Another possible issue that may arise is the performance/range of the Bluetooth module might suffer once we place it inside the box. We avoided this by deciding the orientation of the internal components first and then taking measurements.

3. The pin headers on the PCB for the Bluetooth and rotary encoders had some loose connections. They were de-soldered, replaced, and tested again to make the connections more reliable. Once tested, a glue gun was used to seal the connections in place.

Testing Issues Encountered and their Fixes

1. The Rasberri Pi development board had a constant 18k resistance between ground and V_{cc}. This caused issues with the LDR circuitry as the resistance of the path was greater than 18k. As a result, the LDR output was almost constant as this was a very high resistance path. We fixed this by reducing the resistance by replacing it with a 500 resistor.

2. The power supply to the PCB was to come from an external battery and this required soldering wires to the PCB. Due to repeated testing and handling of these wires, they kept getting detached. This was fixed by adding a glue gun to seal the solder.

3. The IMU (MPU6500) had burned out and we had to replace it with a lower-end model (MPU6050) which as the original one was not available. This model only detected angles in one particular orientation of the IMU i.e. when it was vertical. We overcame this by allowing motion only in the vertical direction.

4. After repeated testing of the rotary encoder, our rotary encoder had been damaged and as a result, was only able to detect rotation in the forward direction. Unable to replace it within the deadline we decided to only permit forward rotation and since turns were possible, the reverse could be replaced by a 180 turn and then moving forward.

Verification of the prototype

Although we were able to develop a working model we needed a way to demonstrate that the prototype meets the deliverables. This was done by

1. Adding a path that the object would trace and matching the plotted map with the actual map,

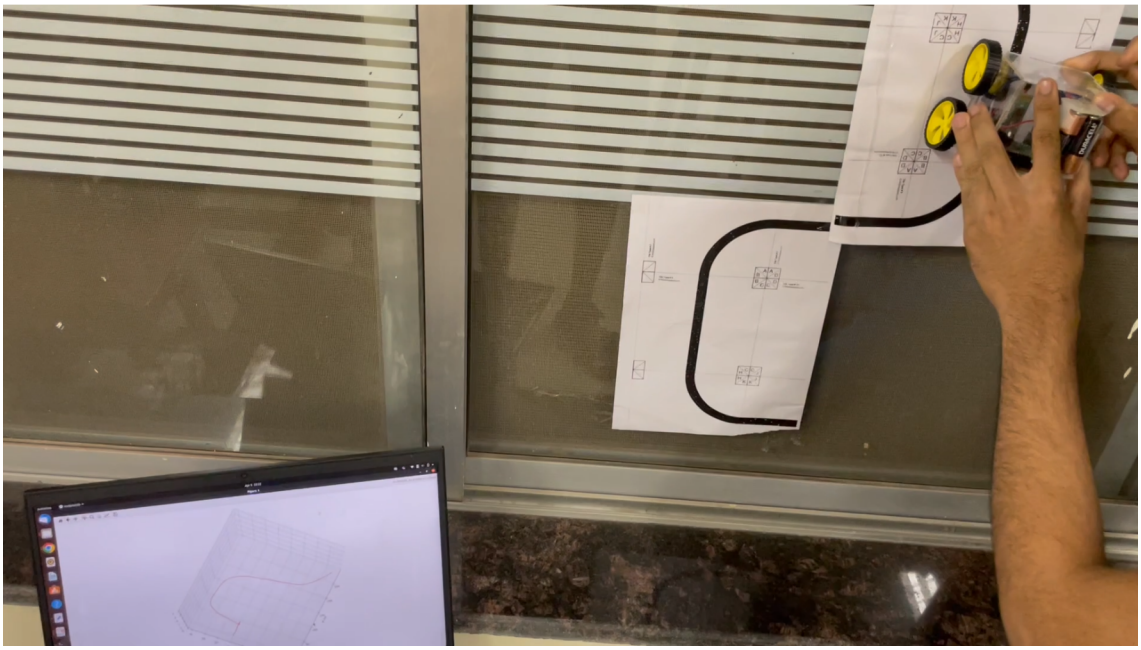


Figure 21: verifying the path traceability of the project

2. To verify its accuracy in motion we demonstrated the linear map of a 30cm ruler with a light source at the end. We got 31cm on the map. Thus , we can measure lengths with an uncertainty of 3.33%.

Figure 22: verifying the path accuracy of the project

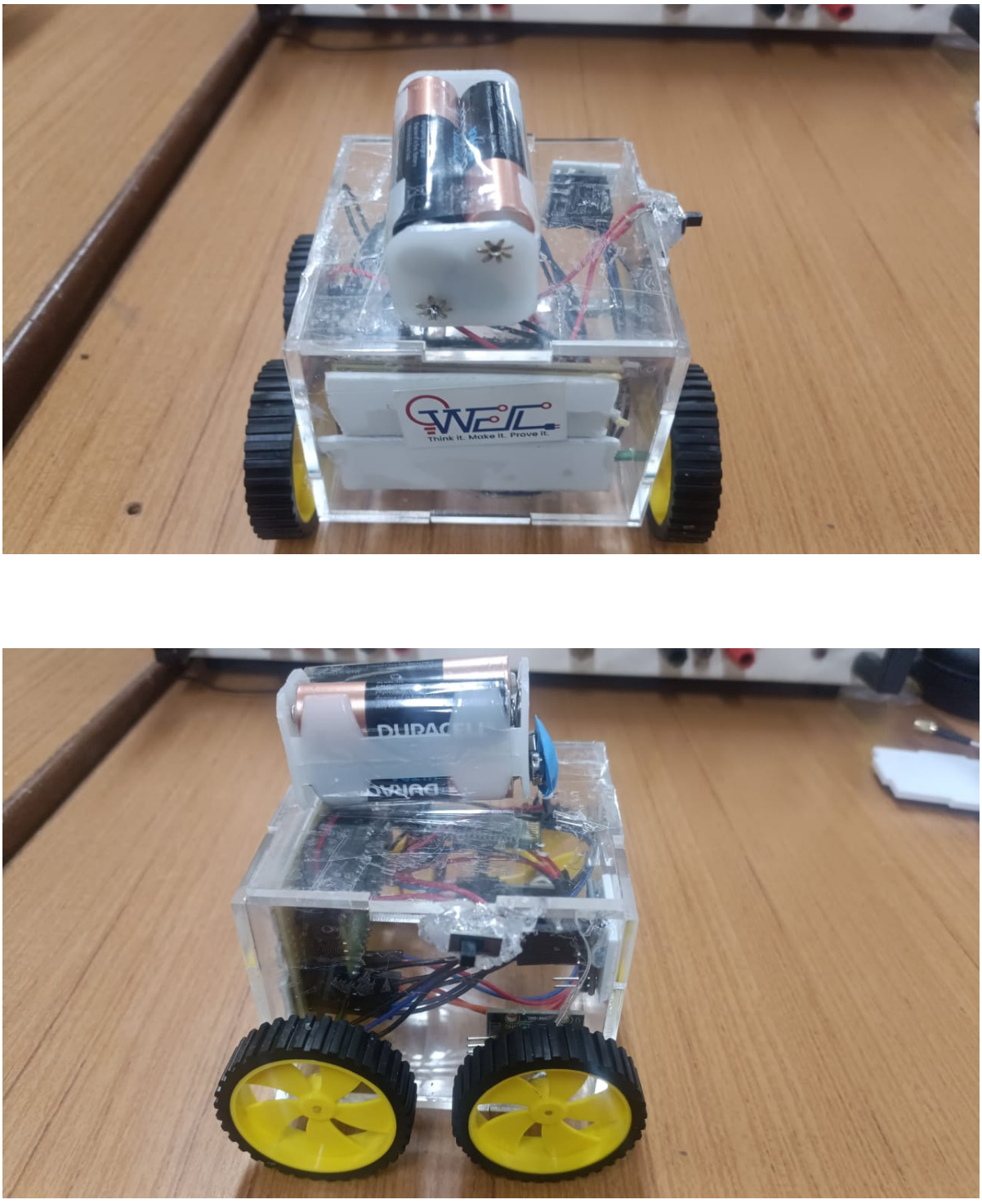


Figure 23: Final Prototype

Following video **Link** is a brief video which sums up all the work done by the group during this semester, and walks through the procedure and the challenges faced while completing this project.

Future Work

The iPEC probe sends out magnetic pulses, which could disorient the IMU and provide an inaccurate map of the path traversed. To overcome this, we could use a CMOS sensor, the ones used in an optical mouse, to detect motion. This and the optical encoder could provide accurate readings prone to magnetic disturbances. Additionally, we could make the vehicle autonomous, taking inspiration from the SLAM algorithm. Path planning algorithms could be used to decide the path to be followed and a map could be made autonomously.