KARNATAK LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG BELAGAVI -590008

KARNATAKA, INDIA.

A Course Project Report on

Omnidirectional robot

Submitted for the requirements of 4th semester B.E. in CSE

for "**PYTHON PROGRAMMING (18CSL46)**"

Submitted by

NAME	USN
MS. SHIVANI BANKE	2GI20CS140
MS. SHRADHA MALLIKARJUN PATIL	2GI20CS144
MS. SRUSHTI B MUDENNAVAR	2GI20CS158
MR.YASH HEREKAR	2GI20CS184

Dr. Manjula Ramannavar

Asst. Prof., Dept. of CSE

Academic Year 2021 - 2022 (Even semester)

KARNATAK LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG BELAGAVI -590008

KARNATAKA, INDIA.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Course Project work titled **"OMNIDIRECTIONAL ROBOT"** carried out by **Student SHIVANI BANKE, SHRADHA MALLIKARJUN PATIL SRUSHTI B MUDENNAVAR AND YASH HEREKAR** bearing **USNs: 2GI20CS140, 2GI20CS144, 2GI20CS158 AND 2GI20CS184** for **PYTHON PROGRAMMING (18CSL46) Integrated Course** is submitted in partial fulfilment of the requirements for 4th semester B.E. in **COMPUTER SCIENCE AND ENGINEERING,** Visvesvaraya Technological University, Belagavi. It is certified that all corrections/ suggestions indicated have been incorporated in the report. The course project report has been approved as it satisfies the academic requirements prescribed for the said degree.

Date:

Signature of guide

Place: Belagavi

Dr. Manjula Ramannavar

Asst., Prof., Dept of CSE

KLS Gogte Institute Of Technology, Belagavi

KARNATAK LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG BELAGAVI -590008

KARNATAKA, INDIA.

Academic Year 2021 - 2022 (Even semester)

Semester: IV

Course: Python Programming (18CSL46) Integrated Course

Rubrics for evaluation of Course Project

Student Name : Yash Herekar

Student USN : 2GI20CS184

Student Branch : Computer Science and Engineering

S.No	**Project Component**	**Max. Marks**	**Marks Earned**
1	Relevance of the project and its objectives	02	
2	Tools/Framework used	01	
3	Methodology / Design	02	
4	Implementation and Results	03	
5	Project Report	02	
	Total	**10**	

Karnatak Law Society's

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG BELAGAVI -590008

KARNATAKA, INDIA.

Academic Year 2021 - 2022 (Even semester)

Semester: IV

Course: Python Programming (18CSL46) Integrated Course

Rubrics for evaluation of Course Project

Student Name : Shivani Banke

Student USN : 2GI20CS140

Student Branch : Computer Science and Engineering

S.No	Project Component	Max. Marks	Marks Earned
1	Relevance of the project and its objectives	02	
2	Tools/Framework used	01	
3	Methodology / Design	02	
4	Implementation and Results	03	
5	Project Report	02	
	Total	**10**	

KARNATAK LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG BELAGAVI -590008

KARNATAKA, INDIA.

Academic Year 2021 - 2022 (Even semester)

Semester: IV

Course: Python Programming (18CSL46) Integrated Course

Rubrics for evaluation of Course Project

Student Name : Shradha Patil

Student USN : 2GI20CS144

Student Branch : Computer Science and Engineering

S.No	Project Component	Max. Marks	Marks Earned
1	Relevance of the project and its objectives	02	
2	Tools/Framework used	01	
3	Methodology / Design	02	
4	Implementation and Results	03	
5	Project Report	02	
	Total	**10**	

KARNATAK LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG BELAGAVI -590008

KARNATAKA, INDIA.

Academic Year 2021 - 2022 (Even semester)

Semester: IV

Course: Python Programming (18CSL46) Integrated Course

Rubrics for evaluation of Course Project

Student Name : Srushti Mudennavar

Student USN : 2GI20CS158

Student Branch : Computer Science and Engineering

S.No	Project Component	Max. Marks	Marks Earned
1	Relevance of the project and its objectives	02	
2	Tools/Framework used	01	
3	Methodology / Design	02	
4	Implementation and Results	03	
5	Project Report	02	
	Total	**10**	

ACKNOWLEDGMENTS

The success and outcome of this project were possible by the guidance and support from many people. We are incredibly privileged to have completed this project. It required a lot of effort from each individual.

We thank Principal Dr.J K Kittur and Dr.Vijay S Rajpurohit, HOD Department of Computer science for providing us this opportunity in our academic curriculum. To have collaborated as a team and hence encouraged us to inculcate qualities like leadership, time management, etc.

We appreciate and humbly thank Dr.Manjula Ramannavar, for granting us with this unique opportunity and providing us with all the support and leadership, which made us execute the project duly.

We are obliged for the consistent encouragement, support and supervision from all the teaching and non-teaching staff. We would like to continue our genuine esteems to all staff in the laboratory for their timely support.

Lastly, we would like thanking our friends and family for their constant support and guidance.

ABSTRACT

This paper deals with programming an omnidirectional robot in python and interfacing it with a GUI designed in Tkinter. An omnidirectional robot is a four wheel robot which has 2 degrees of freedom and 7 degree of movement. It has a special wheels called mecanum wheels which enable it to manoeuver like a crab.

The robot finds applications in: Industrial applications - single task robots. Pick and place, Military applications - autonomous unmanned vehicles.

With the recent advancements in artificial intelligence and machine learning, there is a tremendous scope for robotics in the future.

Some of the drawbacks are the code execution is slower as python heavily relies on a lot of dependencies, due to the overhead and dependancies the code compilation is slower.

Takes a lot of space on a single board computer such as the raspberry pi pico as it has to run a virtual machine to run the code.

We have avoided classes, objects and optimised the code, to a bare minimum input output GPIO functions.

TABLE OF CONTENTS

OMNIDIRECTIONAL ROBOT

1. PROBLEM STATEMENT

Develop a python program to control a robot and also develop a GUI to interface with it

2. OBJECTIVE AND SCOPE OF THE PROJECT

• To program raspberry pi pico using micro-python.
• To design and implement a GUI application using Tkinter.
• To Write functions such as:
  - **front:** Moves the robot forward in y axis.
  - **back:** Moves the robot backward in the y axis.
  - **left:** Turns the robot in the left direction.
  - **rightDiagonalFront:** Moves diagonally in positive X axis and positive Y axis
  - **rightDiagonalBack:** Moves diagonally in positive X and negative Y axis
  - **leftDiagonalFront:** Moves diagonally in negative X and positive Y axis
  - **leftDiagonalBack:** Moves diagonally in negative X and negative Y axis
  - **right:** Turns the robot in the right direction.
  - **leftShift:** Slides the robot to left in the x axis.
  - **rightShift:** Slides the robot to right in the x axis.
  - **printInfo:** Prints system name and embedded operating system name
  - **changeSpeed:** Updates the global speed variable ranging from 1-10 speeds
  - **stop:** Stop all the motors
• To write GUI functions similar to the the one the robot has.

3. TOOLS/FRAMEWORK USED

Table 3.1 Hardware used

Name	**Quantity**	***Description***
Raspberry pi pico RP2040 microcontroller	**1**	*control the motors and sensors*
12v DC Geared Motors	**4**	*300 rpm*
Dual tb6612fng	**2**	*H bridge for motor control*
HM-10 BLE	**1**	*Bluetooth 4.0 CC2541 wireless module*
Lipo Battery	**1**	*Power supply 2200 mah*
Mecanum wheels	**4**	*Omnidirectional wheels*

I. Raspberry pi pico microcontroller

Specifications of raspberry pi Pico

Raspberry Pi Pico is a low-cost, high-performance microcontroller board with flexible digital interfaces. Key features include:

- RP2040 microcontroller chip designed by Raspberry Pi in the United Kingdom
- Dual-core Arm Cortex M0+ processor, flexible clock running up to 133 MHz
- 264kB of SRAM, and 2MB of on-board flash memory
- USB 1.1 with device and host support
- Low-power sleep and dormant modes
- Drag-and-drop programming using mass storage over USB
- 26 × multi-function GPIO pins
- 2 × SPI, 2 × I2C, 2 × UART, 3 × 12-bit ADC, 16 × controllable PWM channels
- Accurate clock and timer on-chip
- Temperature sensor
- Accelerated floating-point libraries on-chip
- 8 × Programmable I/O (PIO) state machines for custom peripheral support

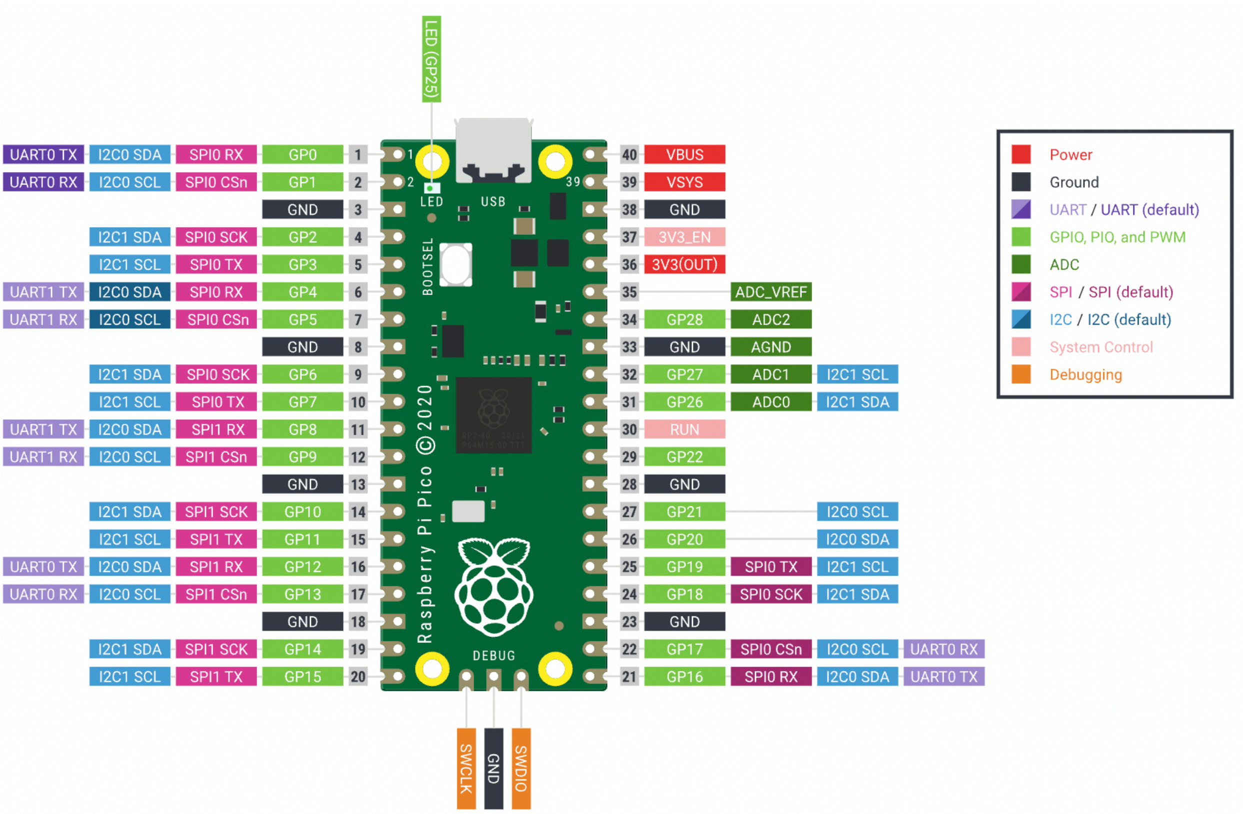


Figure A1. GPIO Pinout [4]

II. DUAL TB6612FNG MOTOR DRIVER

Specifications of Dual tb6612fng motor driver

The TB6612FNG motor driver can control up to two DC motors at a constant current of 1.2A (3.2A peak). Two input signals (IN1 and IN2) can be used to control the motor in one of four function modes - CW, CCW, short-brake, and stop. The two motor outputs (A and B) can be separately controlled, the speed of each motor is controlled via a PWM input signal with a frequency up to 100kHz. The STBY pin should be pulled high to take the motor out of standby mode. Some of the features are:

• Power supply voltage: VM = 15 V(Max)

• Output current: OUT = 1.2 A(average) and 3.2 A (peak)

• Standby (Power save) system

• CW/CCW/short brake/stop function modes

• Built-in thermal shutdown circuit and low voltage detecting circuit.

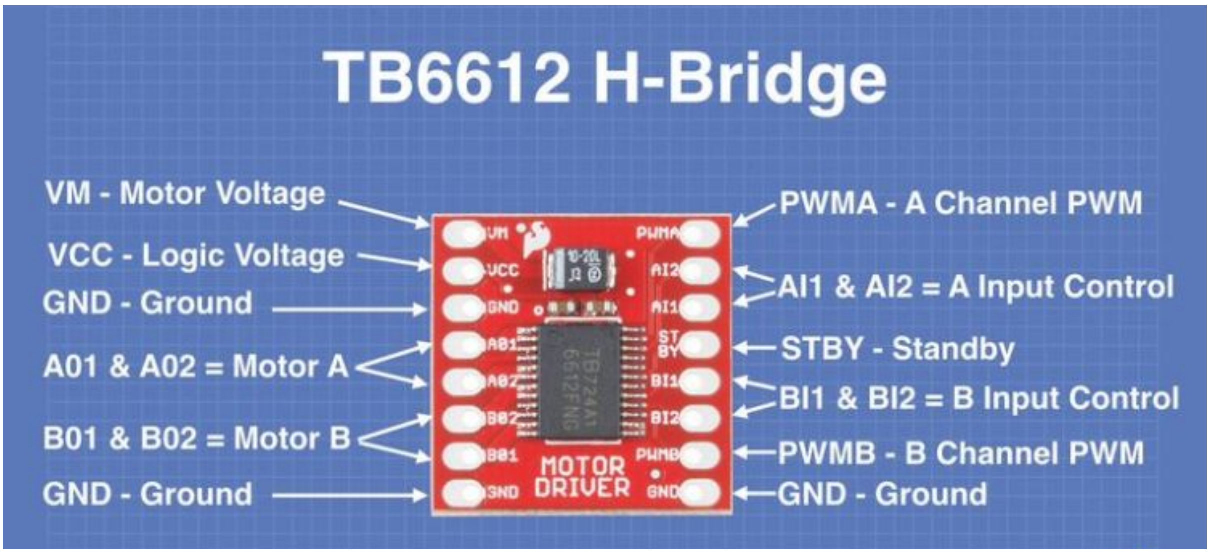


Figure A2. Dual tb6612fng motor driver

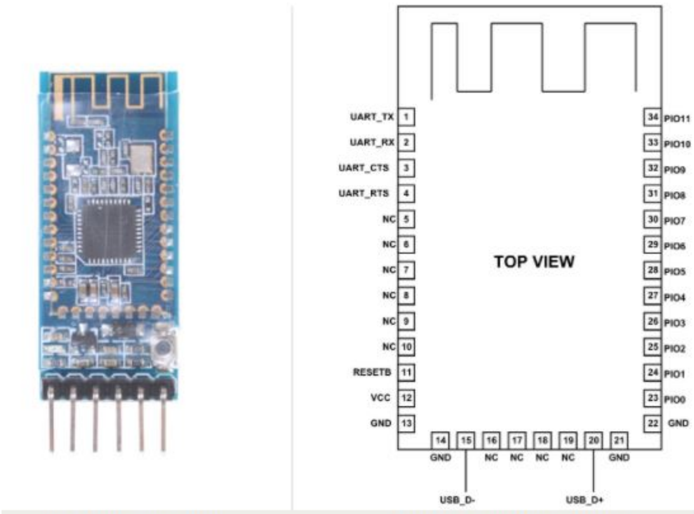
III. HM-10 BLE BLUETOOTH 4.0 CC2540 WIRELESS MODULE

Specifications of HM-10 Bluetooth module:

The HM-10 is a readily available Bluetooth 4.0 module used for establishing wireless data communication. The module is designed by using the Texas Instruments CC2540 or CC2541 Bluetooth low energy (BLE) System on Chip (SoC).

Features:

- BT Version: Bluetooth Specification V4.0 BLE
- Working frequency: 2.4GHz ISM band
- Modulation method: GFSK(Gaussian Frequency Shift Keying)
- RF Power: -23 dbm, -6 dbm, 0 dbm, 6 dbm
- Speed: Asynchronous: 2-6K Bytes Synchronous: 2-6K Bytes
- Security: Authentication and encryption
- Service: 0xFFE0 (Modifiable use AT+UUID command)
- Characteristic: 0xFFE1 (Modifiable use AT+UUID command)
- Characteristic: Notify and Write (Modifiable use AT+UU UID command)
- Power: +2.5V~3.3VDC 50mA
- Power: Active state 8.5mA; Sleep state 50~200uA
- Working temperature: –20 ~ +95 Centigrade
- Size: HM-10 27mm x 13mm x 2.2 mm



HM-10 Bluetooth Module HM-10 Bluetooth Module Pinout

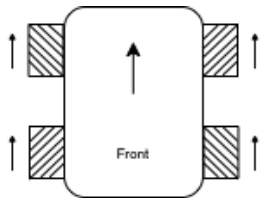
Figure A3. HM-10 Bluetooth module

4. METHODOLOGY/DESIGN

Table 4.1. Functions of the wheels

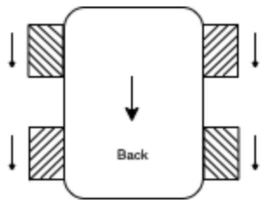
FUNCTIONS	WHEEL 1	WHEEL 2	WHEEL 3	WHEEL 4
front	↑	↑	↑	↑
back	↓	↓	↓	↓
leftTurn	↓	↓	↑	↑
rightTurn	↑	↑	↓	↓
diagonalLeftFront	↑	-	-	↑
diagonalRightFront	-	↑	↑	-
diagonalLeftBack	-	↓	↓	-
diagonalRightBack	↓	-	-	↓
leftShift	↓	↑	↑	↓
rightShift	↑	↓	↓	↑
stop	-	-	-	-

I. Front:



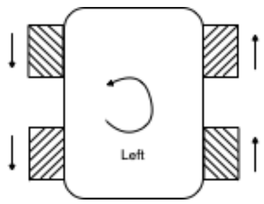
Wheel_1 <= High (Ain1<=High, Ain2<=Low)
Wheel_2 <= High (Bin1<=High, Bin2<=Low)
Wheel_3 <= High (Ain3<=High, Ain4<=Low)
Wheel_4 <= High (Bin4<=High, Bin4<=Low)

II. Back:



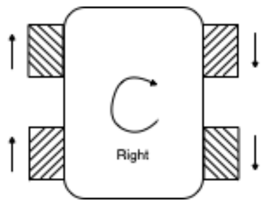
Wheel_1 <= Low (Ain1<=Low, Ain2<=High)
Wheel_2 <= Low (Bin1<=Low, Bin2<=High)
Wheel_3 <= Low (Ain3<=Low, Ain4<=High)
Wheel_4 <= Low (Bin3<=Low, Bin4<=High)

III. Left:



Wheel_1 <= Low (Ain1<=Low, Ain2<=High)
Wheel_2 <= Low (Bin1<=Low, Bin2<=High)
Wheel_3 <= High (Ain3<=High, Ain4<=Low)
Wheel_4 <= High (Bin3<=High, Bin4-<=Low)

IV. Right:



Wheel_1 <= High (Ain1<=High, Ain2-<=Low)
Wheel_2 <= High (Bin1<=High, Bin2-<=Low)
Wheel_3 <= Low (Ain3<=Low, Ain4<=High)
Wheel_4 <= Low (Bin3-<=Low, Bin4<=High)

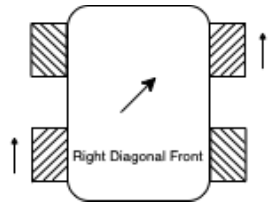
V. Right Diagonal Front

Wheel_1 <= Off

Wheel_2 <= High(Bin1<=High, Bin2<=Low)

Wheel_3 <= High (Ain3<=High, Ain4<=Low)

Wheel_4 <= Off



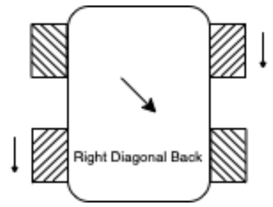
VI. Right Diagonal Back

Wheel_1 <= Off

Wheel_2 <= Low(Bin1<=Low, Bin2<=High)

Wheel_3 <= Low (Ain3<=Low, Ain4<=High)

Wheel_4 <= Off



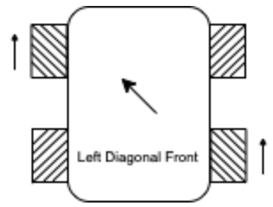
VII. Left Diagonal Front

Wheel_1 <= High(Ain1<=High, Ain2<=Low)

Wheel_2 <= Off

Wheel_3 <= Off

Wheel_4 <= High(Bin3<=High, Bin4<=Low)



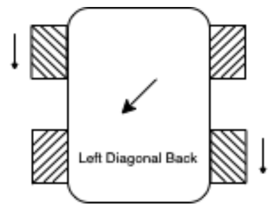
VIII. Left Diagonal Back

Wheel_1 <= Low(Ain1<=Low, Ain2<=High)

Wheel_2 <= Off

Wheel_3 <= Off

Wheel_4 <= Low(Bin3<=Low, Bin4<=High)



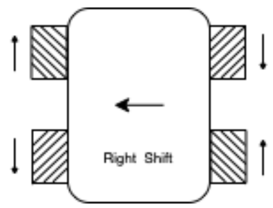
IX. Left Shift:

Wheel_1 <= Low (Ain1<=Low, Ain2<=High)

Wheel_2 <= High (Bin1<=High, Bin2<=Low)

Wheel_3 <= High (Ain3<=High, Ain4<=Low)

Wheel_4 <= Low (Bin3<=Low, Bin4<=High)



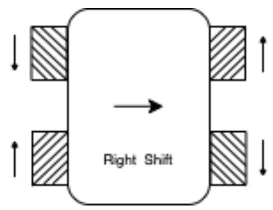
X. Right Shift:

Wheel_1 <= Low (Ain1<=Low, Ain2<=High)

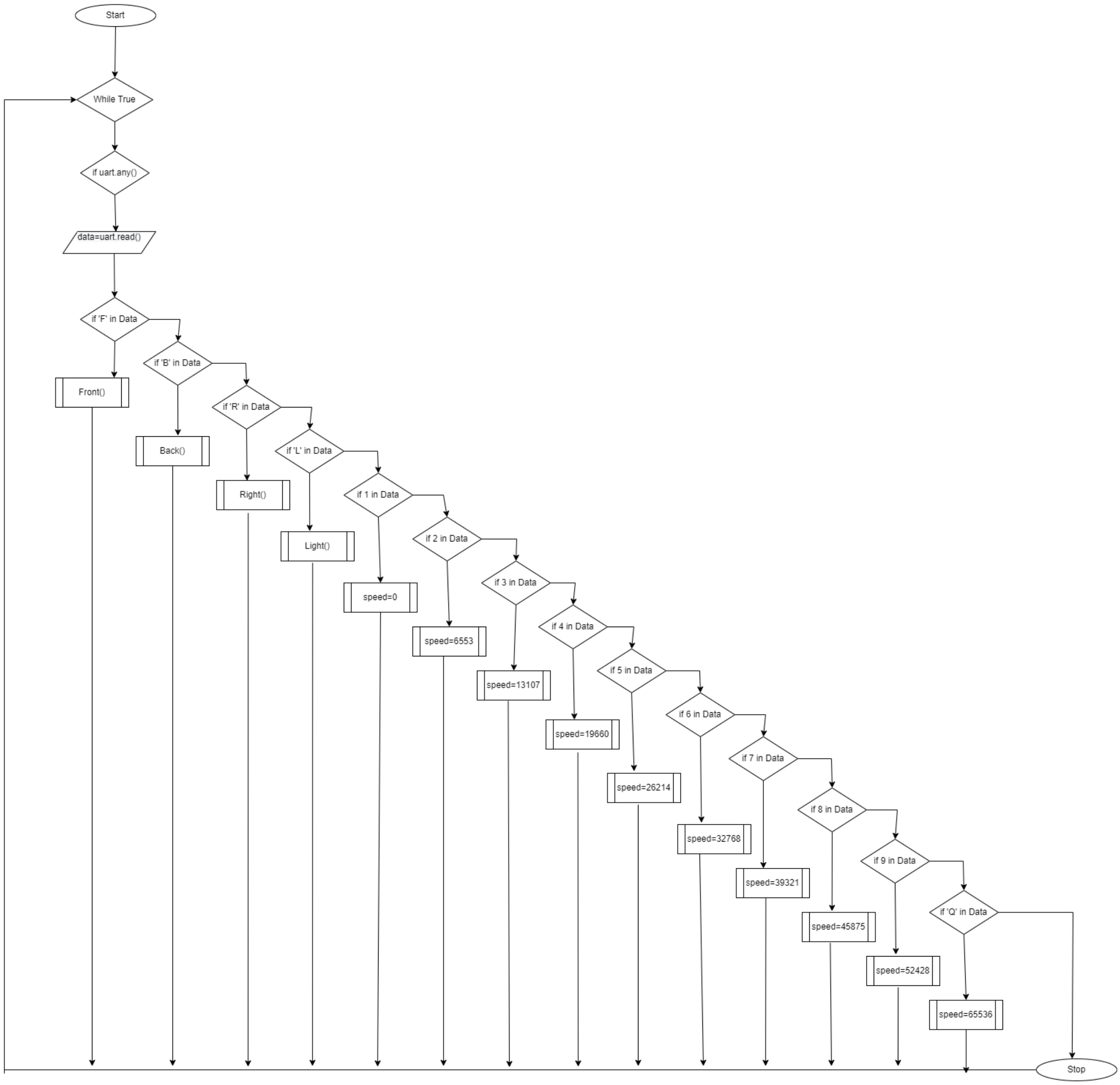
Wheel_2 <= Low (Bin1<=Low, Bin2<=High)

Wheel_3 <= Low (Ain3<=Low, Ain4<=High)

Wheel_4 <= High (Bin3<=High, Bin4<=Low)



FLOWCHART


Start
While True
if uart.any()
data=uart.read()
if 'F' in Data
Front()
if 'B' in Data
Back()
if 'R' in Data
Right()
if 'L' in Data
Light()
if 1 in Data
speed=0
if 2 in Data
speed=6553
if 3 in Data
speed=13107
if 4 in Data
speed=19660
if 5 in Data
speed=26214
if 6 in Data
speed=32768
if 7 in Data
speed=39321
if 8 in Data
speed=45875
if 9 in Data
speed=52428
if 'Q' in Data
speed=65536
Stop

5. ALGORITHM

I. ALGORITHM FOR THE GUI

Step1: Start

Step2: Scanning for the Bluetooth module

Step3: Connect to the BLE module using its MAC address

Step4: Send commands

Step4: Stop

II. ALGORITHM FOR THE ROBOT

Step1: Start

Step2: Print system info [5][3]

Step3: Initialise global variables and uart objects [1]

Step4: while True

Step5: if uart.any()[2]

Step4: data <= uart.read()[2]

Step4: If (‘F’ in data): front()

Step5: elif (‘B’ in data): back()

Step6: elif (‘L’ in data): left()

Step7: elif (‘R’ in data): right()

Step8: elif (‘I’ in data): rightDiagonalFront()

Step9: elif (‘G’ in data): leftDiagonalFront()

Step10: elif (‘H’ in data): leftDiagonalBack()

Step11: elif (‘J’ in data): rightDiagonalBack()

Step12: elif (‘W’ in data): leftShift()

Step13: elif (‘U’ in data): rightShift()

Step14: elif (‘1’ in data): speed <= 10

Step15: elif (‘2’ in data): speed <= 20

Step16: e1if (‘3’ in data): speed <= 30

Step17: elif (‘4’ in data): speed <= 40

Step18: elif (‘5’ in data): speed <= 50

Step19: elif (‘6’ in data): speed <= 60

Step20: elif (‘7’ in data): speed <= 70

Step21: elif (‘8’ in data): speed <= 80

Step22: elif (‘9’ in data): speed <= 90

Step23: elif (‘10’ in data): speed <= 100

Step24: else(): stop()

6. SCREENSHOTS

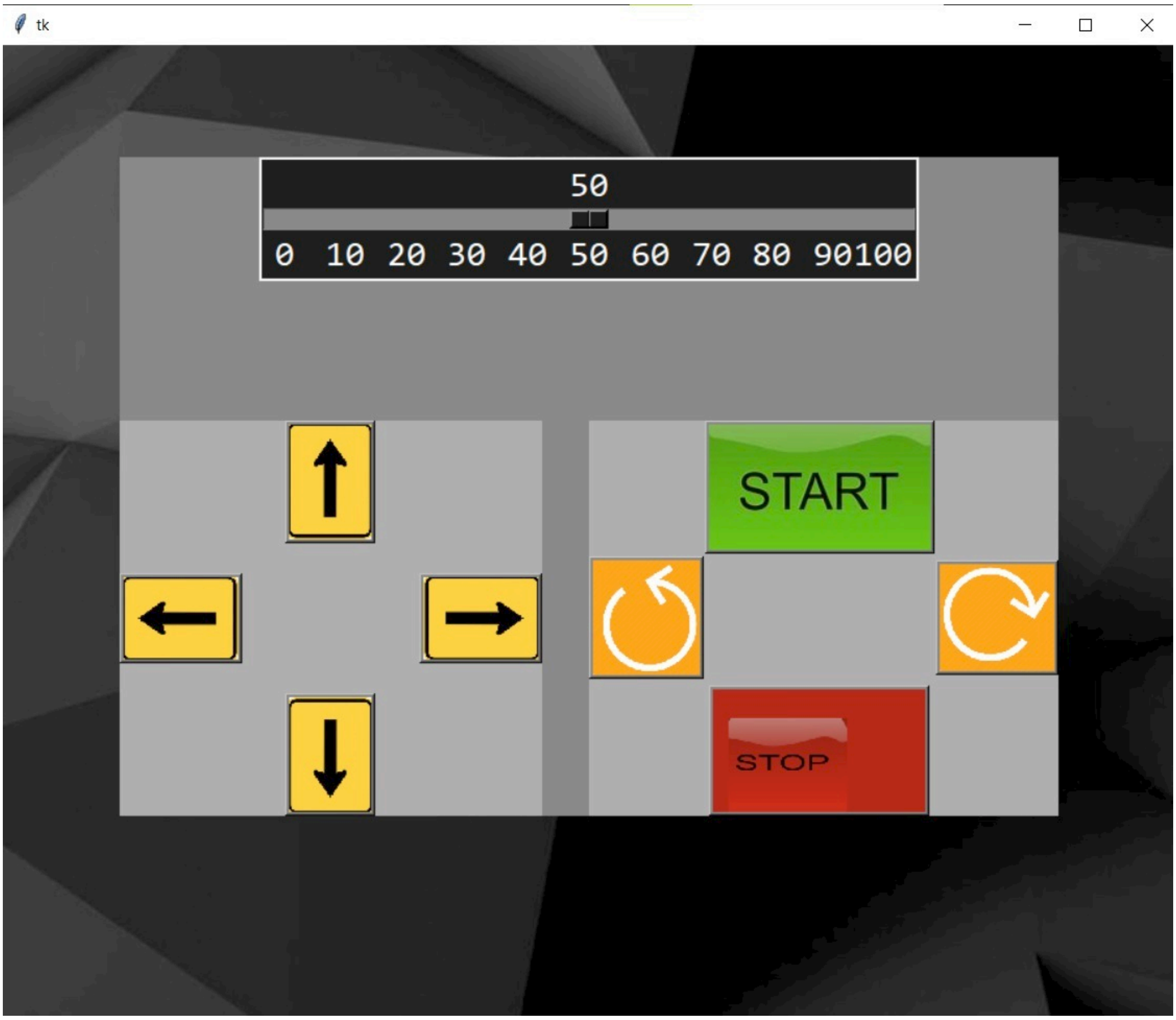
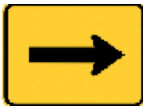


Figure 6.1. GUI designed using tkinter

 - **front:** Moves the robot forward in y axis.

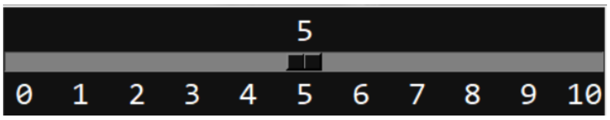
 - **back:** Moves the robot backward in the y axis.

 - **left:** Turns the robot in the left direction.

 - **right:** Turns the robot in the right direction.

 - **leftShift:** Slides the robot to left in the x axis.

 - **rightShift:** Slides the robot to right in the x axis.



- **changeSpeed:** Use to control the speed of the motors.

- **stop:** Stop all the motors.

- **start :**Use to connect to the robot and start the motors.

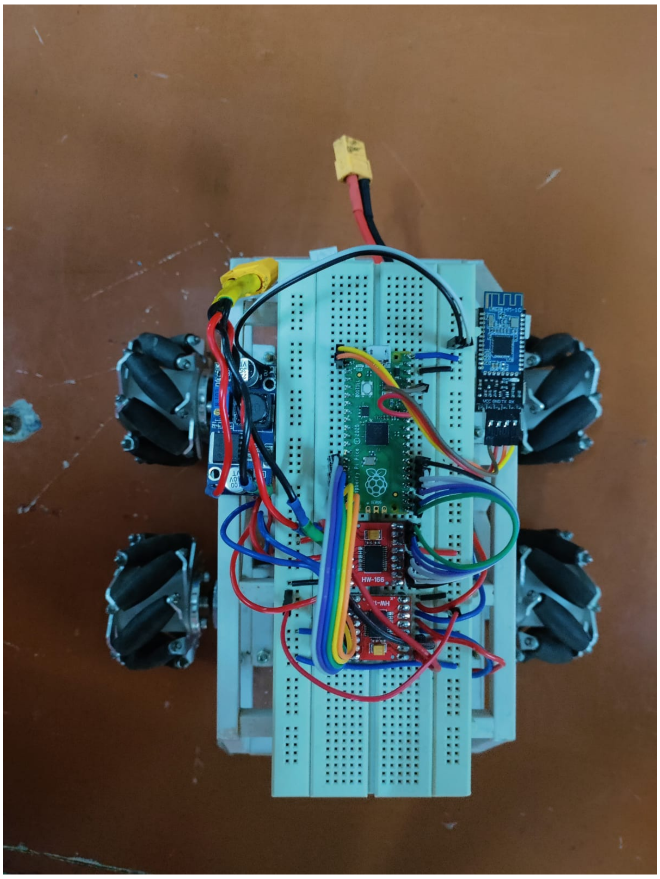


Figure 6.2 Top View of the bot

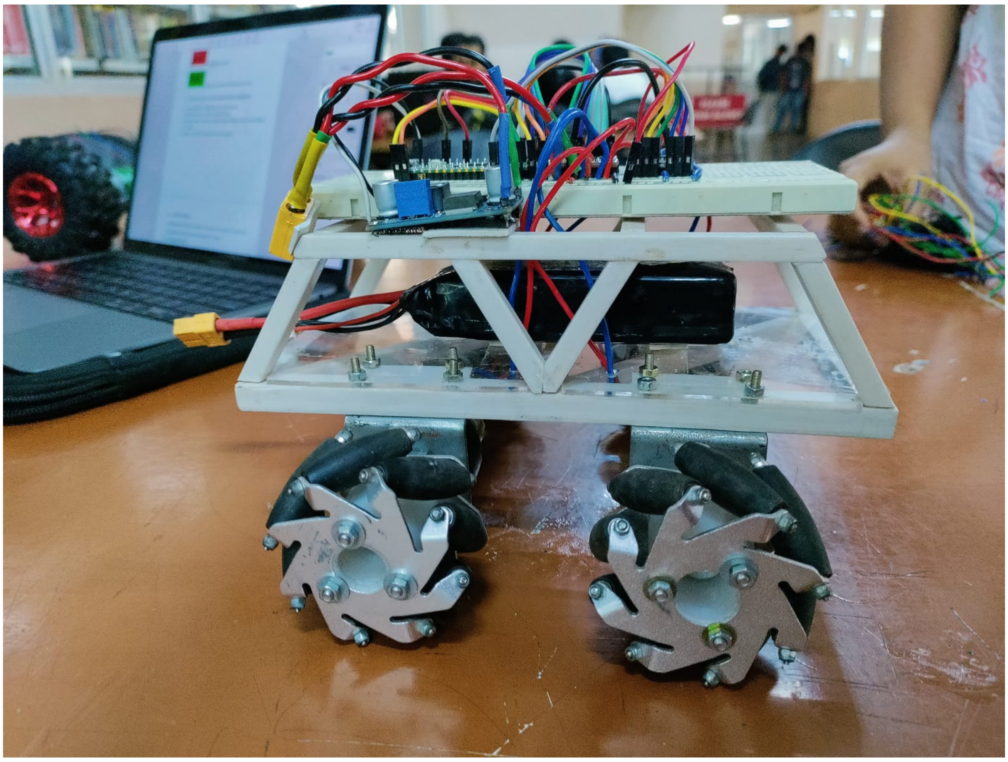


Figure 6.3 Side view of the bot

7. APPLICATION AND FUTURE ENHANCEMENT

- Industrial applications - single task robots. Pick and place, etc
- Military applications - autonomous unmanned vehicles.
- Space exploration - Curiosity rover on Mars, navigates the terrain on mars and sends feedback back to earth.
- Healthcare - The da Vinci Surgical Robot, performs surgeries.Conclusion

During the course of the project we learnt how to program a microcontroller using python and also design a GUI and connect to the robot using the GUI interface.

8. REFERENCES

[1] https://projects.raspberrypi.org/en/projects/getting-started-with-the-pico

[2] https://docs.micropython.org/en/latest/rp2/quickref.html#uart-serial-bus

[3] https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf

[4] https://datasheets.raspberrypi.com/pico/Pico-R3-A4-Pinout.pdf

[5] https://github.com/1337encrypted/Raspberry-pi-pico/blob/main/pythonbot.py

9. APPENDIX

A1. Raspberry pi pico GPIO pinout

A2. Dual tb6612fng motor driver

A3. HM<=High0 BLE bluetooth 4.0 CC2540 wireless module