Intermediate Level Development Board for Robot Navigation

U. G. U. I. Ranasinghe (E/16/309)   K. P. M. I. Jayarathne (E/16/147)
A. M. P. .S. Bandara (E/16/041)   R. M. S. K. Abeyrathne (E/16/004)

1. Abstract

A robot’s ability to perform its job safely and securely is heavily dependent upon understanding of its path and surroundings.However robot navigation is far more complex than simply programming a robot to do a task or follow a path.The intended outcome of this project is to design and develop a device for tracking and navigation of a mobile robot with respect to a given starting point. This device should be capable of measuring the orientation and position of a mobile robot. Some of the existing solutions for measuring orientation and measurements including GPS, Robot odometry, LIDAR and LORA have some practical issues such as low accuracy, limited number of applications, high dependency on external parameters, high cost and high processing power.
In Spite of these practical problems the main issue we identified is that these methods can be used only for specific applications under limited conditions. In addition to that an external module for orientation tracking needs to be used along with them to achieve the expected outcome. Considering these issues, our goal was to find a feasible solution with high accuracy to address these issues which can be used in different applications according to customer requirements.

2. Product and Concept

After identifying the problems in achieving the expected outcome several methods were suggested and discussed. While trying to recognize the optimum method, the idea of developing an intermediate level development board adaptive for a diverse range of mobile robotic applications including navigation tracking seemed to be the most customer-satisfactory solution.
When designing this development board, we basically focussed on implementing an inbuilt orientation tracking module along with I2C and SPI ports which allows the customer to use an optional module for position tracking and data transmission depending on the type of application.

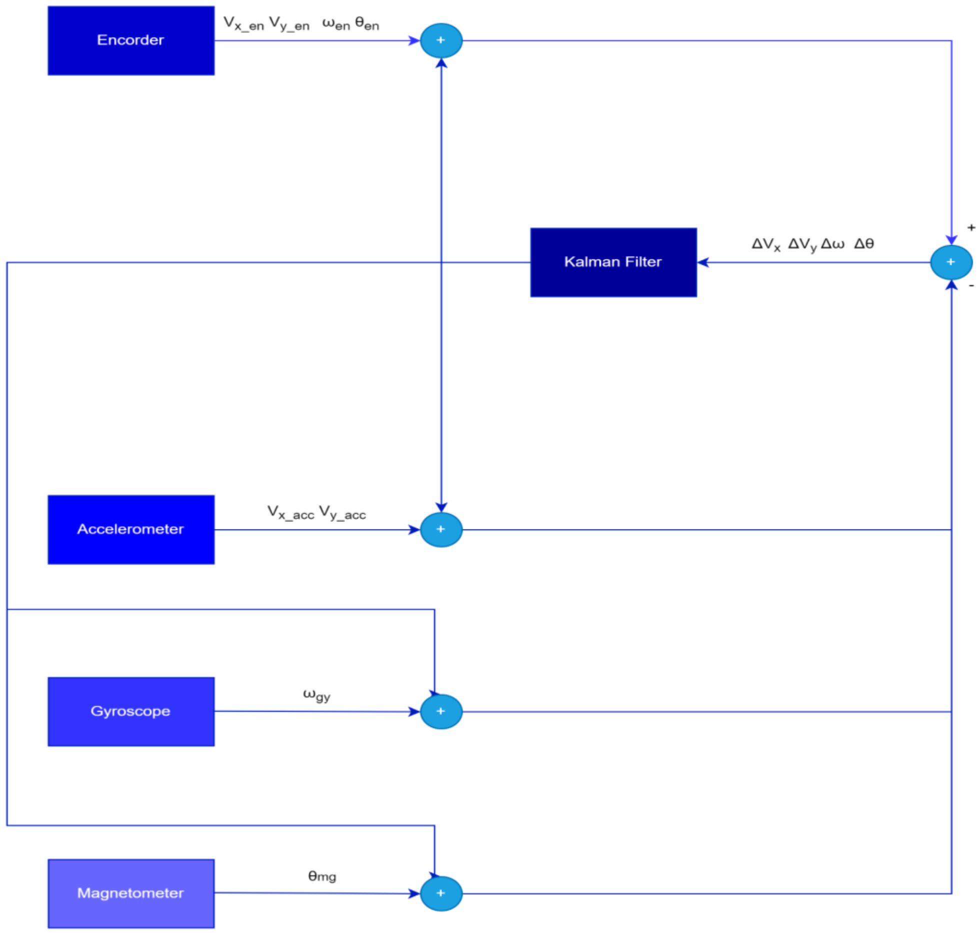


Figure 1: Block diagram

2.1 Product Specifications

- The embedded digital motion processor inside MPU9250 offloads the computation burden of the motion processing algorithm from the main processor.
- Nine degree of freedom
- Full Scale resolution of the gyroscope ± 250 ±500 ±1000 ±2000 degrees/sec
- Full Scale resolution of the accelerometer ±2g ±4g ±8g ±16g
- Full Scale resolution of the magnetometer ±4800 micro tesla
- 16 bit analog to digital converter (5-200 Hz sampling rate)
- Input voltage 3.6 - 12V
- ATSAMD21G8A-A ARM cortex -M0+ CPU (32 MHz clock speed)

- Single cycle hardware multiplier
- Memory
    - SRAM - 32kB
    - Flash Memory - 256kB
    - Read while Write Memory - 4kB
    - External micro SC - upto 1 GB
- Low Power capabilities
- Idling and standby sleep modes
- Pins
    - PWM channels -4
    - Analogue channels - 6
    - GP I/O - 11
    - Debugging ports
    - I2C Connections
    - Direct connections to other peripherals
    - Mini USB port
    - Arduino Support

2.2 Concept and components

- The board has a high board rate which makes real time tracking possible and is capable of giving out the real time orientation, speed and acceleration of a mobile robot with its inbuilt components and sensors (MPU 9250 processor with gyroscope , magnetometer and accelerometer)
- Depending on the application, the user can connect external modules such as GPS, odometry and LIDAR to give the position input to the board.
- The board merges data from above sensors and performs error reduction through kalman filtering and outputs the navigation(orientation, position, velocity, acceleration and angular velocity) of the robot with high accuracy.
- The board also contains an SD card and a USB port which allows the user to store the real time data and analyse later in applications where real time data transmission is difficult.
- There is an inbuilt serial port for remote data transmission where the user can connect a bluetooth, wifi or a radio frequency module depending on the external conditions and application

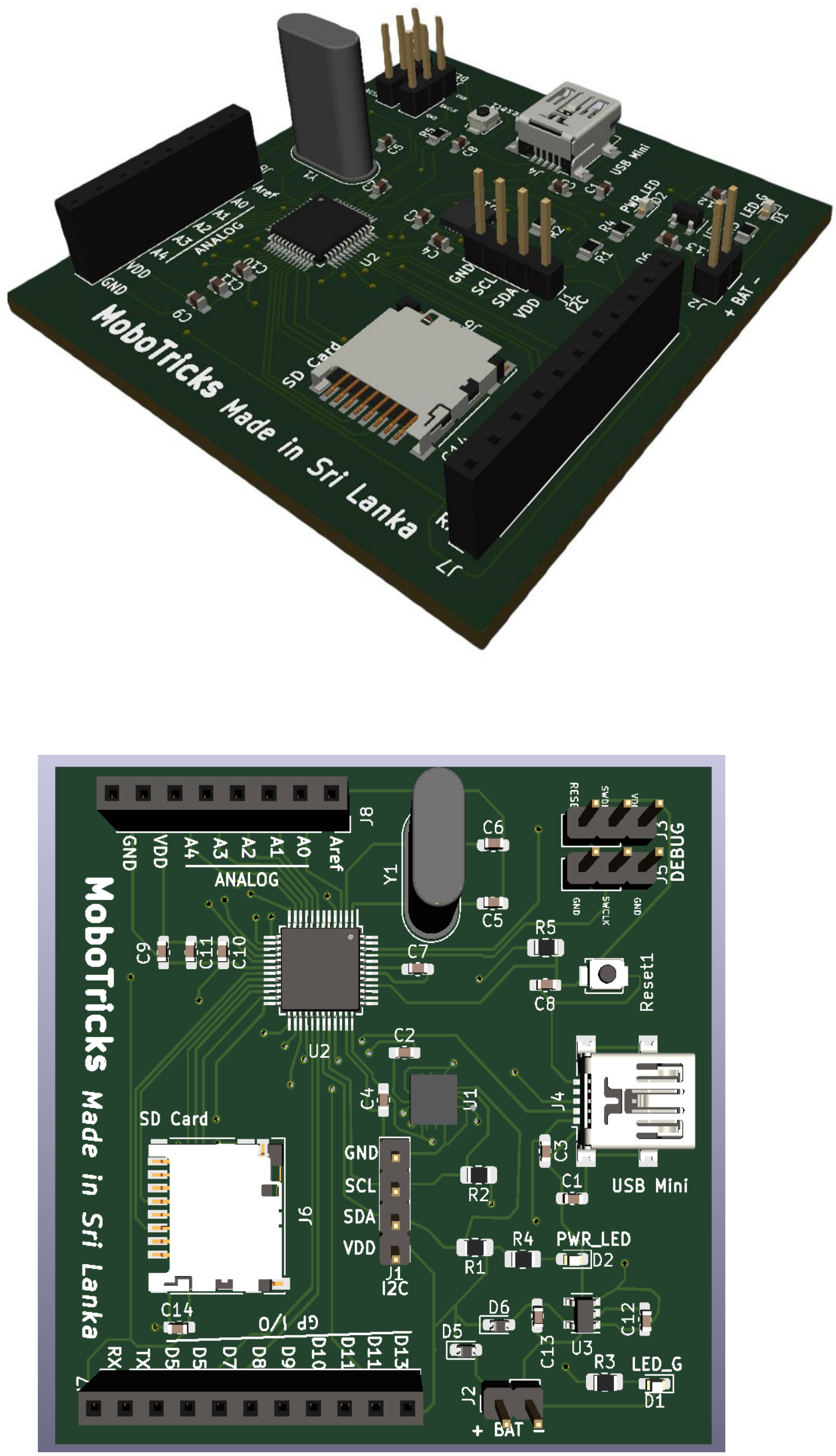


Figure 2(a),(b) Proposed design for the final product

3. Procedure

After initial brainstorming sessions and researching, a block diagram and a work plan was designed including the expected functionalities of the product. The most suitable circuit components for the product were chosen by referring to the data sheets of a variety of components and processors available for each task.

3.1 Software Designing

- The PCB design of the development board was designed using KICAD software and the components were arranged neatly to make the outer appearance of the product more attractive as the board itself is the final product.

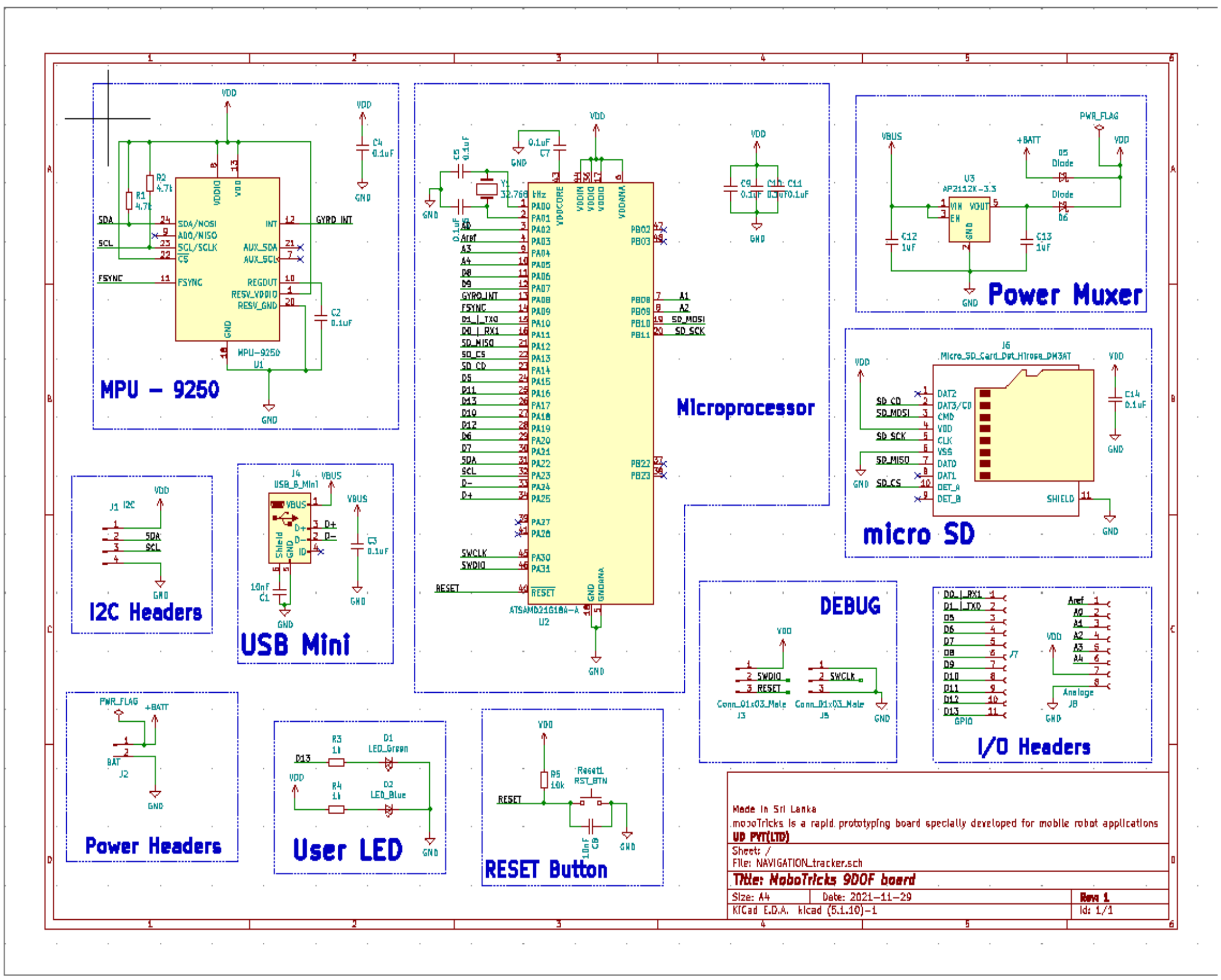


Figure 3: Circuit Diagram

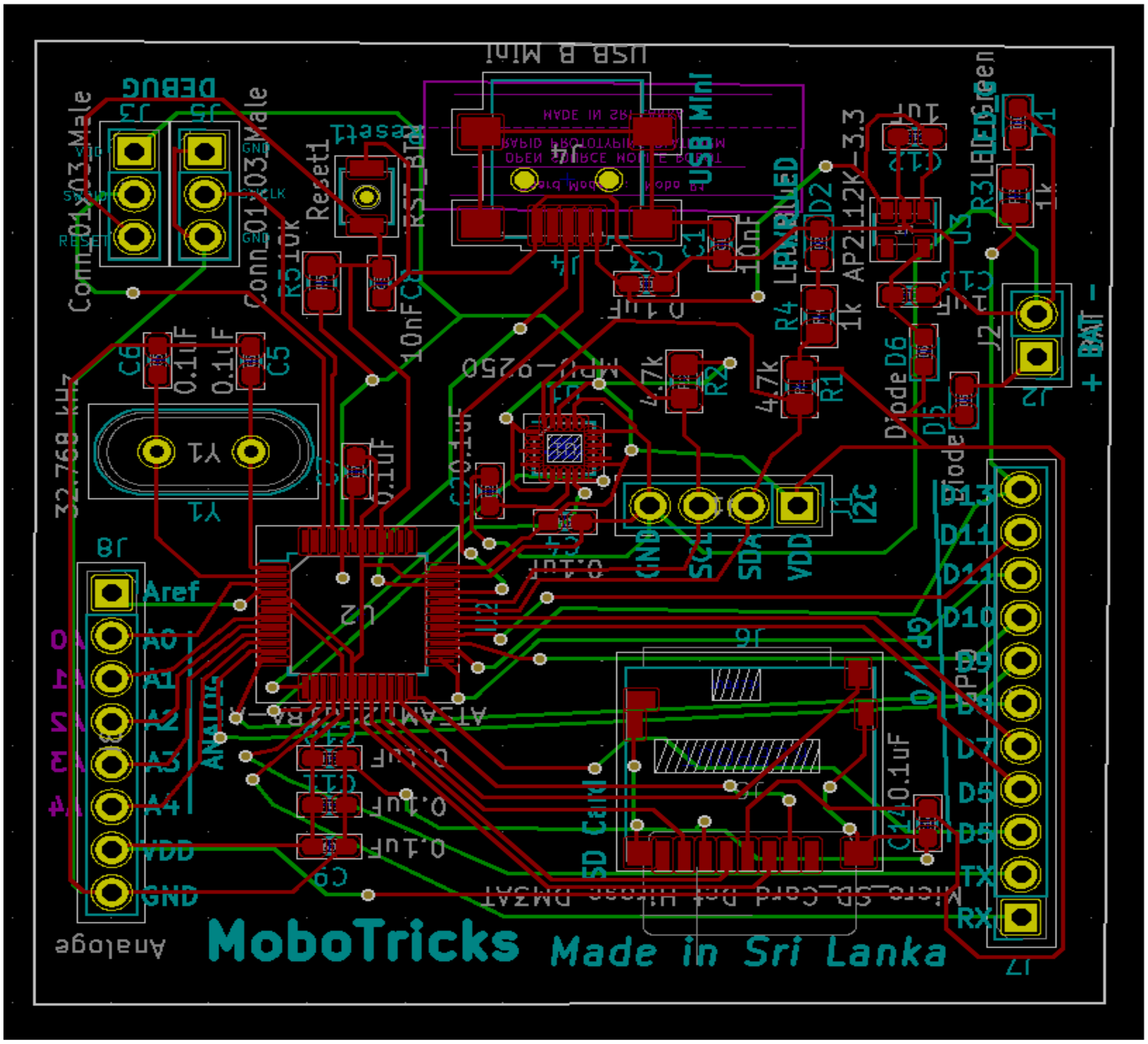


Figure 4: PCB Design using KICAD

3.2 Coding and software simulations

- Matlab -
  - Used to connect the bluetooth module to the development board to transmit and display the output through a mobile app developed along with the board
  - Used to design and program a kalman filter for error reduction. The position measurements from user input (encoding, GPS, LIDA, LORA) are used as the model along with position measurements obtained through axial components as the plant of the kalman filter.

- Pseudo code for kalman filtering

From wheel odometry

$$V_{en}(k+1) = \left(\frac{V_L(k)+V_R(k)}{2}\right)$$

$$\omega_{en}(k+1) = \left(\frac{V_L(k)-V_R(k)}{L}\right)$$

$$\theta_{en}(k+1) = \left(\frac{V_L(k)-V_R(k)}{L}\right)\Delta T$$

$$V_{x_{en}}(k+1) = V_{en}(k).cos(\omega_{en}(k)\Delta T)$$

$$V_{y_{en}}(k+1) = V_{en}(k).sin(\omega_{en}(k)\Delta T)$$

From wheel odometry

$$V_{en}(k+1) = \left(\frac{V_L(k)+V_R(k)}{2}\right)$$

$$\omega_{en}(k+1) = \left(\frac{V_L(k)-V_R(k)}{L}\right)$$

$$\theta_{en}(k+1) = \left(\frac{V_L(k)-V_R(k)}{L}\right)\Delta T$$

$$V_{x_{en}}(k+1) = V_{en}(k).cos(\omega_{en}(k)\Delta T)$$

$$V_{y_{en}}(k+1) = V_{en}(k).sin(\omega_{en}(k)\Delta T)$$

From Accelerometer

$$V_{x_{acc}}(k+1) = V_{x_{acc}}(k) + C_{x_{acc}}x(k)\Delta T + B_{x_{acc}}$$

$$V_{y_{acc}}(k+1) = V_{y_{acc}}(k) + C_{y_{acc}}y(k)\Delta T + B_{y_{acc}}$$

From Fusion of Wheel odometry data and Accelerometer data

$$\Delta V_x(k+1) = V_{x_{en}}(k)cos(\omega_{en}(k)\Delta T) - V_{x_{acc}}(k) - C_{x_{acc}}x(k)\Delta T - B_{x_{acc}}$$

$$\Delta V_y(k+1) = V_{y_{en}}(k)cos(\omega_{en}(k)\Delta T) - V_{y_{acc}}(k) - C_{y_{acc}}y(k)\Delta T - B_{y_{acc}}$$

From Gyroscope

$$\omega_{gy}(k+1) = C_{gy}\Omega_z(k) + B_{z_{gy}}$$

From Fusion of Wheel odometry data and Gyroscope data

$$\Delta\omega(k+1) = \omega_{en}(k) - C_{gy}\Omega_z(k) - B_{z_{gy}}$$

From Magnetometer

$$\theta_{mg}(k+1) = \theta(k) + B_{AZ_{mg}}$$

From Fusion of Wheel odometry data and Magnetometer data

$$\Delta\theta_{mg}(k+1) = \omega\Delta T - \theta(k) - B_{AZ_{mg}}$$

$$X(k+1) = \Phi X(k) + \Gamma U(k)$$

$$Y(k+1) = CX(k)$$

$$X(k+1)=\begin{bmatrix} V_{x_{en}}(k+1) \\ V_{y_{en}}(k+1) \\ \theta_{en}(k+1) \\ \omega_{en}(k+1) \\ V_{x_{acc}}(k+1) \\ V_{y_{acc}}(k+1) \\ \omega_{gy}(k+1) \\ \theta_{mg}(k+1) \end{bmatrix} \quad X(k)=\begin{bmatrix} V_{x_{en}}(k) \\ V_{y_{en}}(k) \\ \theta_{en}(k) \\ \omega_{en}(k) \\ V_{x_{acc}}(k) \\ V_{y_{acc}}(k) \\ \omega_{gy}(k) \\ \theta_{mg}(k) \end{bmatrix} \quad \Phi=\begin{bmatrix} \cos(\omega_{en}(k)\Delta T) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sin(\omega_{en}(k)\Delta T) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Delta T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Gamma=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{L} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{x_{acc}}\Delta T & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{y_{acc}}\Delta T & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & C_{gy} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad U(k)=\begin{bmatrix} 0 \\ 0 \\ 0 \\ V_L(k)-V_R(k) \\ x(k)+\frac{B_{X_{acc}}}{C_{x_{acc}}\Delta T} \\ y(k)+\frac{B_{y_{acc}}}{C_{x_{acc}}\Delta T} \\ \Omega(k)+\frac{B_{z_{gy}}}{C_{gy}} \\ B_{AZ_{mg}} \end{bmatrix}$$

$$Y(k)=\begin{bmatrix} V_{x_{en}}(k)-V_{x_{acc}}(k) \\ V_{y_{en}}(k)-V_{y_{acc}}(k) \\ \theta_{en}(k)-\theta_{mg}(k) \\ \omega_{en}(k)-\omega_{gy}(k) \end{bmatrix} \quad C=\begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix}$$

- Applying Kalman Filter to reduce error

<u>Time – Update</u>

$X\hat{}_k^{-} = \Phi X\hat{}_{k-1}^{-} + \Gamma U_k$

$P_k^{-} = \Phi P_{k-1}\Phi^T + Q$

<u>Measurement – Update</u>

$K_k = P_k^{-} C^T (C P_k^{-} C^T + R)^{-1}$

$X\hat{}_k = X\hat{}_k^{-} + K_k(y_k - C X\hat{}_k^{-})$

$P_k = (I - K_k C) P_k^{-}$

ΔT	*- Sampling time*
$V_L(k)$	*- Left wheel velocity*
$V_R(k)$	*- Right wheel velocity*
$V_{x_{en}}(k)$	*- x axis velocity from encoder*
$V_{y_{en}}(k)$	*- y axis velocity from encoder*
$\theta_{en}(k)$	*- Angular displacement form encoder*
$\omega_{en}(k)$	*- Angular velocity from encoder*
$V_{x_{acc}}(k)$	*- x axis velocity from accelerometer*
$V_{y_{acc}}(k)$	*- y axis velocity from accelerometer*
$\omega_{gy}(k)$	*- Angular velocity from gyroscope*
$\theta_{mg}(k)$	*- Angular velocity from magnetometer*
$X\hat{}_k$	*- Estimated state vector of* $X(k)$
K_k	*- Kalman Gain at time k*
P_k	*- Estimation error covariance at time k*

- Webot simulation software -
    - Used to realize a full implementation of the kalman filtering process and to obtain a full simulation of error reduction in velocity measurements.

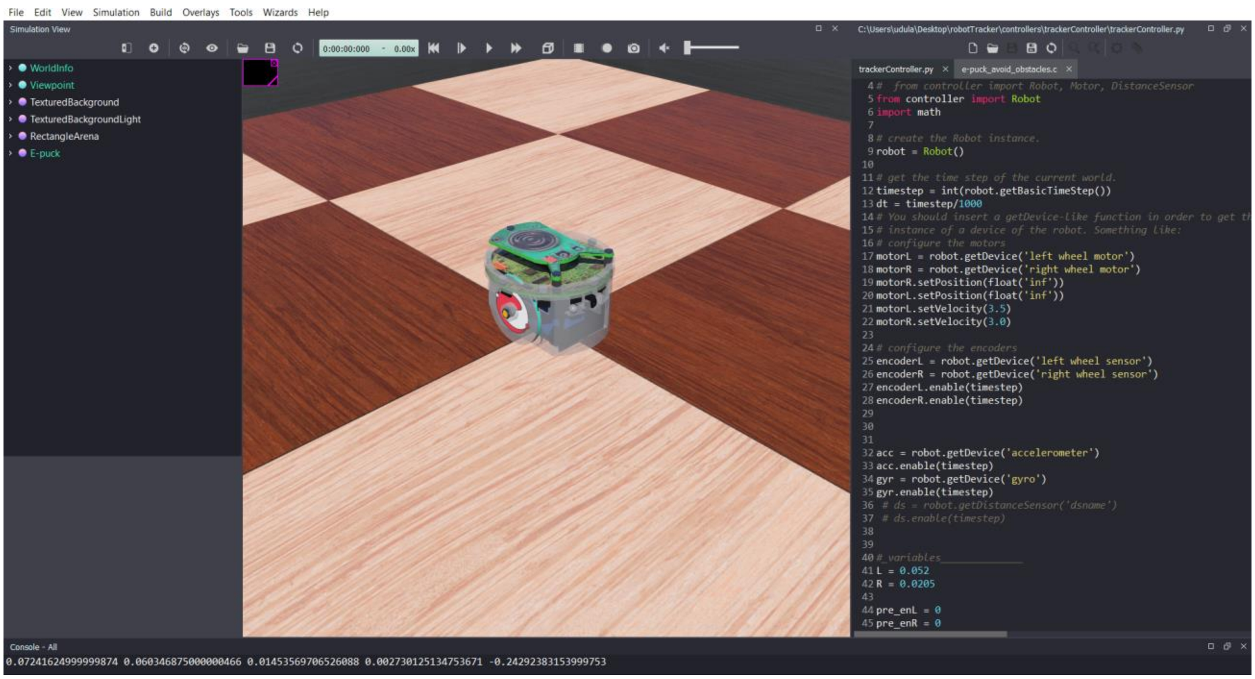


Figure 5: Simulation using webot

- C Programming -
    - Used to configure the microprocessor
    - Used to implement the communication nod between hardware and software

3.3 Hardware Implementation

- After designing the PCB, it was handed over to a specialized PCB soldering company to get the final product done as surface mounted PCB soldering technologies are not available in Sri Lanka.
- Designed a differential robot for hardware testing to measure the orientation using gyroscope module and to measure position using encoder odometry
- Implemented separate test circuits for orientation and position measurements using arduino and tested them using differential robot
  - For orientation measuring at the beginning the test circuit was implemented using MPU 6050 sensor which consist of inbuilt 3 axis gyroscope and an 3 axis accelerometer. To achieve a higher accuracy level the sensor was replaced with MPU 9250 which consist of a magnetometer along with above 2 modules
  - For position measuring a test circuit was implemented using rpm technology(encoder)
- Implemented a bluetooth communication module to get the output through a mobile application

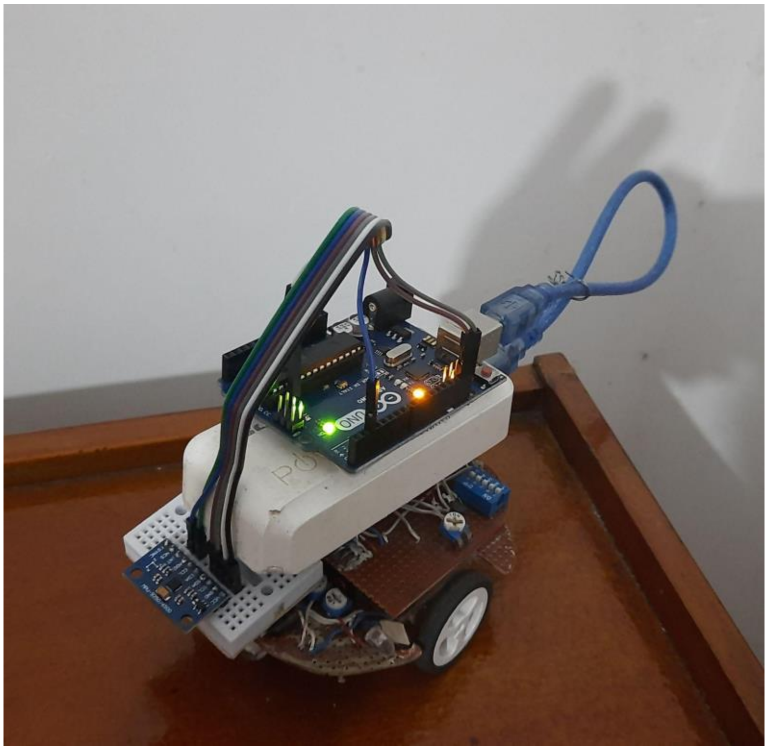
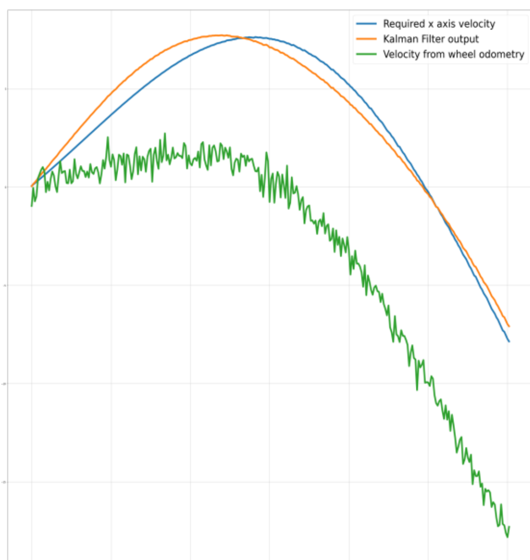


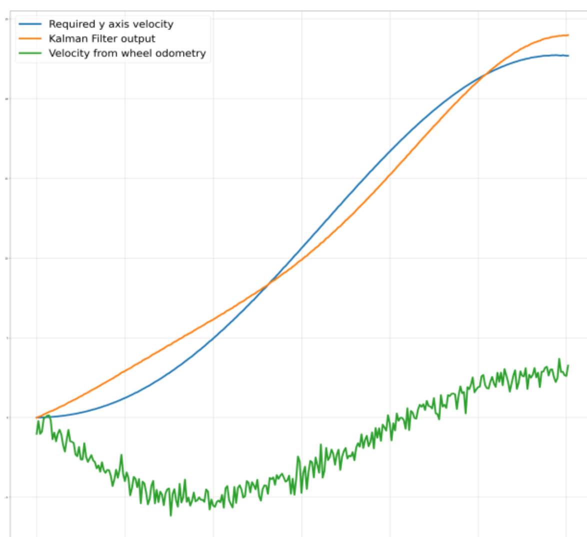
Figure 6: Differential robot for testing

3.4 Results

The development board is capable of giving the following measurements as outputs.

- X direction velocity and acceleration
- Y direction velocity and acceleration
- XY coordinates of position
- Orientation
- Angular velocity
- Z direction velocity, acceleration and position(at current stage z direction measurements are low in accuracy)
- Error rate of each measurement





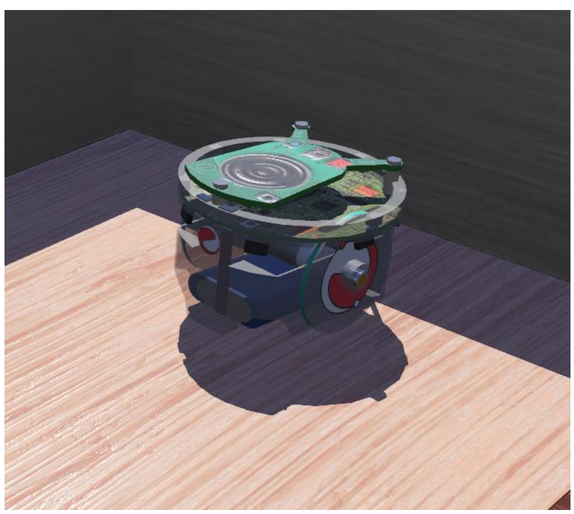


Figure 7(a),(b),(c): Simulation of velocity output using WEBOT software

3.5 Problems faced and solutions

- At the beginning using arduino for real time tracking was not efficient
    - Discussed to designed a intermediate level development board with a high board rate(32MHz) for tracking and navigation
- After designing the PCB we Needed a robot for demonstration purposes.
    - At first we discussed using a remote control toy car for this and finally designed a differential robot for testing and demonstration
- Due to the Gyroscopic noise higher attenuation was observed at the output
    - Used RC components and implemented a Low Pass Filter to minimize the attenuation at the orientation measurement output
- Implementing kalman filter for 2 or more measurement matrices
    - Used extended kalman filter
- Finding suitable sampling rate that it will not cause distortions in the data but also being able to transfer via the available communication link
    - Selected a suitable sampling rate through trial and error
- Arduino could not handle the higher computations
    - Sent data to the computer for further processing

3.6 Progress

	Weeks 1-3	Weeks 4-6	Weeks 7-10	Weeks 11-13	Weeks 16-18	Weeks 19-21	Weeks 22-23
Understanding the problem and discussing possible solutions							
Designing the block diagram and circuit diagram							
Designing of PCB and Robot for demonstration purposes							
Coding (kalman,bluetooth module)							
Separate hardware testing using arduino							
Implementation on printed PCB and developing the mobile app							
Testing and Debugging							

Table 1: Gunn diagram of the progress

4. Applications and key features

4.1 Applications

- Rapid prototyping of a mobile robot
- Real time 2D orientation and position measurement
- Analysing the navigation of a mobile robot with stored data
- Acceleration and velocity measurement

4.2 Key features

- High accuracy of angle measurements with 9DOF
- Can be used in any type of robot both outdoors and indoors
- Ability to connect external devices through I2C, SPI ports and USB ports.
- Inbuilt SD card
- Competitive price point with the available market solutions.
- Low space consumption
- User friendly
- Durability

4.3 Application Guidelines

- After purchasing the development board, the user needs to install the mobile application issued along with the board to their device.
- Then two wheels of the robot have to be connected to an encoder.
- The user can obtain the output measurements by connecting the board to the application via bluetooth.

5. Future Expansions

- Developing an operating system specially devoted for robotic applications
- Real time 3D measurements (At the moment the board is capable of obtaining real time 2D measurements with a high accuracy. The error range of the z direction is comparatively high when compared with that. We hope to increase this accuracy to make the product to be used for real time 3D measurements.)
- Increasing the capabilities of the processor for applications that require higher computational power

6. References

https://github.com/kd8bxp/RFID-indoor-robot-navigation
https://www.directionsmag.com/article/1598
https://blog.abdurrosyid.com/2021/07/22/fusing-wheel-odometry-imu-data-and-gps-data-using-robot_localization-in-ros/
https://www.mathworks.com/help/fusion/ug/Estimating-Orientation-Using-Inertial-Sensor-Fusion-and-MPU-9250.html
https://crackeconcept.blogspot.com/2014/03/arduino-and-matlab-interfacing-via.html