DE-42 (DC & SE)

NUST COLLEGE OF
ELECTRICAL AND MECHANICAL ENGINEERING

DESIGN AND DEVELOPMENT OF DIGITAL DRIVER PANEL WITH DATA LOGGING CAPABILITIES (GYROCOMPASS, ODOMETER SPEEDOMETER)

A PROJECT REPORT

DE-42 (DC & SE)

Submitted by

GC MUHAMMAD SHARJEEL

GC HAMZA ILYAS

GC HAMZA YAHYA

BACHELORS

IN

COMPUTER ENGINEERING

YEAR

2024

PROJECT SUPERVISOR

DR. AQIB PERVAIZ

DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING
COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING
NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY,
ISLAMABAD, PAKISTAN

YEAR 2024

Certification

This is to certify that GC Muhammad Sharjeel, 358975 and GC Hamza Ilyas, 358833 and GC Hamza Yahya, 322404 have successfully completed the final project Design and development of DDP with data logging capabilities (gyrocompass, odometer speedometer), at the College of Electrical and Mechanical Engineering,(CEME) to fulfill the partial requirement of the degree 42.

Signature of Project Supervisor
Dr. AQIB PERWAIZ
Director of RDC

Sustainable Development Goals (SDGs)

SDG No	Description of SDG	SDG No	Description of SDG
SDG 1	No Poverty	**SDG 9**	Industry, Innovation, and Infrastructure
SDG 2	Zero Hunger	SDG 10	Reduced Inequalities
SDG 3	Good Health and Well Being	SDG 11	Sustainable Cities and Communities
SDG 4	Quality Education	SDG 12	Responsible Consumption and Production
SDG 5	Gender Equality	SDG 13	Climate Change
SDG 6	Clean Water and Sanitation	SDG 14	Life Below Water
SDG 7	Affordable and Clean Energy	SDG 15	Life on Land
SDG 8	Decent Work and Economic Growth	SDG 16	Peace, Justice and Strong Institutions
		SDG 17	Partnerships for the Goals

Sustainable Development Goals

Complex Engineering Problem

Range of Complex Problem Solving

	Attribute	**Complex Problem**	
1	Range of conflicting requirements	Involve wide-ranging or conflicting technical, engineering and other issues.	X
2	Depth of analysis required	Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models.	
3	Depth of knowledge required	Requires research-based knowledge much of which is at, or informed by, the forefront of the professional discipline and which allows a fundamentals-based, first principles analytical approach.	X
4	Familiarity of issues	Involve infrequently encountered issues	X
5	Extent of applicable codes	Are outside problems encompassed by standards and codes of practice for professional engineering.	X
6	Extent of stakeholder involvement and level of conflicting requirements	Involve diverse groups of stakeholders with widely varying needs.	
7	Consequences	Have significant consequences in a range of contexts.	
8	Interdependence	Are high level problems including many component parts or sub-problems	X

Range of Complex Problem Activities

	Attribute	**Complex Activities**	
1	Range of resources	Involve the use of diverse resources (and for this purpose, resources include people, money, equipment, materials, information and technologies).	X
2	Level of interaction	Require resolution of significant problems arising from interactions between wide ranging and conflicting technical, engineering or other issues.	X
3	Innovation	Involve creative use of engineering principles and research-based knowledge in novel ways.	X
4	Consequences to society and the environment	Have significant consequences in a range of contexts, characterized by difficulty of prediction and mitigation.	
5	Familiarity	Can extend beyond previous experiences by applying principles-based approaches.	X

Dedicated to our advisor, Dr. Aqib Pervaiz and RDC

Acknowledgment

We, first and foremost, would need to say thanks to Allah Ta'ala for His favors on us in finishing this undertaking. We are tremendously thankful to Allah for giving us information and cognizance. Our greatest wellspring of motivation and support has come from His superb heading, which has permitted us to defeat provokes and offer lucidity to cloudy circumstances.

We are thankful to our supervisor, Dr. Aqib Pervaiz and Maj Abdul Rafe whose direction, support, and sharp advice have demonstrated basic skills during this modern task. His skill and help fundamentally affect this venture's direction and have given us essential direction at each stage.

We would like to express our sincere gratitude and appreciation to everyone who assisted us in finishing this thesis.We might want to thank every individual who assisted us with finishing this project with the most profound appreciation.

Abstract

The goal of this project is to create a digital driver panel for a tank by incorporating three essential sensors: a speedometer, an odometer, and a gyrocompass. The main goal is to use the ESP32 micro controller to convert the analog signals from these sensors into digital format so that the readings may be shown on a Nextion touchscreen interface. The objective of this transition is to improve operating efficiency and give the tank operator real-time data visibility.

A direction sensor gives data with the goal that directional mindfulness can kept up while work. While the speedometer estimates the vehicle's speed utilizing beat recurrence flags that match the tank's speed, the odometer estimates the distance the tank has covered. The ESP32 sensor yields are altered by signal molding circuits to empower right taking care of and show of this information. Correction, smoothing the signal are important to give exact information readings.

A printed circuit board (PCB) houses all the parts safely and ensures dependability and longevity in the operational environment of the tank. Before the PCB is constructed, the PCB layout is designed using PCB design software. The ESP32 microcontroller is customized utilizing the Arduino IDE, after which it processes the sensor information and communicates it to the Nextion HMI. The Nextion GUI is an instinctive interface that shows continuous sensor information.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

In cutting-edge military and commercial automobiles, unique instrumentation is critical for effective operation and choice-making. Traditional analog gauges, at the same time as dependable, frequently lack the precision, clarity, and integration abilities required in nowadays technologically superior environments. The analog instruments in tanks, which includes speedometers, odometers, and gyrocompasses, offer vital facts for navigation and operation but may be hard to study accurately and speedily beneath diverse situations.

1.2 Background

Historically, tanks have relied on mechanical and analog systems to display vital operational records. These structures, though robust, include inherent boundaries which includes susceptibility to wear and tear, reduced accuracy over the years, and issue in integrating with cutting-edge digital structures. Analog gauges require frequent calibration and may be hard to interpret rapidly, particularly in excessive-strain situations.

With the advent of virtual generation, there's a large push toward integrating digital presentations and sensors in military automobiles. Digital systems provide several benefits such as higher precision, ease of integration with other virtual systems, higher readability, and the capacity to log and examine information for post-project tests. The use of microcontrollers like the ESP32 allows for the processing and show of facts fromnumerous sensors, presenting a unified interface for the operator.

1.3 Motivation

The motivation for this project stems from the need to modernize the instrumentation in tanks to improve operational performance and situational attention. By replacing conventional analog gauges with a digital motive force panel, we purpose to decorate the accuracy and clarity of important data. This challenge will offer a platform for integrating more than one sensors, processing their outputs, and showing the records on an intuitive and person-friendly interface.

1.4 Key Motivations

The objectives we have to complete in order to make to the project fully operational are:

- Digital displays provide clear and more precise readings compared to analog gauges, which is critical and difficult in high-pressure situations.
- Digital structures can log records through the years, bearing in mind certain analysis of automobile universal performance and operational patterns, which may be beneficial for preservation and strategic planning.
- Digital interfaces can without difficulty combine with other structures, which include navigation and communique gear, supplying a complete operational evaluation.
- Modern virtual sensors and shows are less prone to mechanical screw ups and may

be greater long lasting in harsh operational environments.

- A unified digital display reduces the cognitive load on operators, allowing them to awareness on task-crucial obligations without being distracted via hard to-examine gauges.

By addressing these motivations, the assignment pursuits to seriously beautify the operational skills of tanks, providing a strong and modern-day answer that aligns with the wishes of current army era. This virtual transformation will now not simplest improve the on-the-spot usability of the automobile's instrumentation but additionally pave the manner for future improvements in military car layout and operation.

1.5 Problem Statement

Speedometers, odometers, and gyrocompasses are examples of outdated analog instrumentation in tanks that present several challenging scenarios that evade situational focus and operational effectiveness. These devices need regular calibration, are prone to errors over time, and can be challenging to read quickly, especially in high-strain situations. Furthermore, their ease of integration with modern digital frameworks has diminished, so impeding the capacity to perform thorough data analysis and make operational decisions in real time.

Modernizing these systems is essential in an era where records must be accessed quickly and accurately. Analog gauges may not be up to date with the clarity, integration potential, and information logging capabilities that current army operations require. The difficulty is exacerbated by the difficulty in seamlessly integrating those devices with other virtual systems inside the tank, leading to fragmented and inefficient operational protocols.

The following issues need to be addressed in this project:

- Analog gauges are not accurate and good for adjustment and decision making at critical times and situations.

- Analogue gauges and system requires a lot care and maintenance which is time consuming and more staff is required as well.
- This project is manally develop or choosed to create a digital driver panel for tanks by combining an ESP32 microcontroller, a Nextion touchscreen display, and three sensors (gyrocompass, odometer, and speedometer) so that we can overcome these issues.

1.6 Objectives

Improve Operational Accuracy: Our objective is to develop a digital driver panel which will improve the accuracy of the readings obtaining from the sensors.

Enhance Data Integration: We must make sure that all the data from the sensors which we are getting are integrated accurately and without any error.

Boost Reliability and Cut Down on Maintenance: As analogue gauges required more staff and maintenance requirement this issue is going to resolve by this approach of making digital driver panel.

Facilitate Real-Time Data Analysis: We will be using ESP32 for integration with the sensors and getting data over real time.

Improve Situational Awareness: This approach will improve the situational awareness for the driver sitting in the tank.

1.7 Scope

The goal of this project is to develop a digital driver panel for a tank and integrate three sensors with the nextion display to get data or values on a real time.

Sensor Integration: The proper digital sensors (gyrocompass, odometer, and speedometer) must be selected and integrated to detect vehicle direction, distance traveled, and speed precisely.

Confirming that the sensor output signals can be handled by the ESP32 microcontroller.

Circuit Design and Simulation: Designing and simulating of the circuit is done on a proteus software to make it easy for PCB making.

Microcontroller Programming: Creating algorithms and libraries according to the sensors respectively on Arduino IDE software and creating logic in the code.

GUI Development: Creating a graphical user interface (GUI) to show real-time sensor data on the Nextion touchscreen display.

Prototyping and Testing: Constructing and developing a digital driver panel prototype to test it thoroughly.

Deployment and Calibration: generating comprehensive system documentation, including code, circuit schematics, and user manuals.

Instructing users and maintenance personnel on how to operate and take care of the digital driver panel.

The project focuses to give tank operators a cutting-edge, dependable, and efficient digital driver panel that enhances operational accuracy, integration, and situational awareness by defining this scope.

1.8 Report Orginization

The orginization of the thesis is as follows:

- Chapter 2, it mainly explains the background and literature review and discuss the technologies of microcontroler and also the work related to our project.
- Chapter 3, it deals with the materials and the components that we have used the sensors and the software for integration and making of interface.

- Chapter 4, it explains the methodology of our working that how we have integrated the sensors and designed the GUI,PCB and its casing.
- Chapter 5, it shows the testing and the validation process and discuss the safety parameters.
- Chapter 6, it comprises of concluding the report and examining future prospects and directions of the project.

Chapter 2

Background and Literature Review

Tank driver panel design has come a long way in terms of technological innovation. In the past, analog instruments were often used in tanks to show heading, speed, and distance traveled. They have a variety of other drawbacks, such as less precision and more maintenance work, and they cannot interface with modern digital systems due to their analog design. More precision, ease of interaction with other digital systems, better dependability, and increased design flexibility for the user interface are just a few of the ways that digital displays outperform their analog counterparts.

With the advent of sensor technology advancements, and microcontroller capabilities; we can easily design a digital driver panel that gives us high-degree-accuracy real-time data.

ESP32 Microcontroller — one of the best microcontrollers in terms of integration with sensors and data communication due its versatile nature and powerful performance In addition, Nextion touchscreen displays permit user-friendly and modifiable GUI for more efficient operation of monitoring responses to real-time data.

2.1 Literature Review

2.1.1 Digital Transformation in Military Vehicles:

Increased situational awareness and improved operational efficiency are are the two main driving factors that we see in military vehicles transitioning from analog to digital systems. There are countless studies that show how digital systems can relieve operators from the responsibility of managing lots of information by presenting only what they need to know differently to minimize pressure load. [1]

2.1.2 Technology:

It is possible that sensor technology of today will be marginally more precise and reliable than that of the past. Since then, the accuracy of gyrocompasses, odometers, and speedometers has greatly increased; these instruments are crucial for operation and navigation. Studies show that digital sensors are far less likely to drift and require a great deal less maintenance than their analog counterparts.

- **Microcontroller Integration:**

  It was exactly this device, the ESP32 microcontroller, that is widely used in IoT since it possesses noteworthy performance, Bluetooth connectivity, as well as Wi-Fi interfaces. It has the ability to compute specific details that are essential in processing and presenting data It has several sensor interfaces that enable it to solve complex problems. As this brief analysis underlines the identified the ESP32 achieves the effective sensor data management and establishes simultaneous interconnection with other devices and display units.

- **User Interface Design:**

  They include the concerns of general contrasts, the simplicity of the user interface surface, and functional stability under different conditions of use and environmental factors relevant to military use.

The technicalities relating to the interfaces that developers design can hence be useful while at the same time being rather easy to use.

The versatility of Nextion screens has seen a considerate level of flexibility that makes the screens suitable for usage. One cannot help but learn of the benefits of introducing or maintaining an easy to understand and effortless to use interface which similarly reduces the implicit exhaustion of the operators.

- **Data Communication and Logging:**

  Modern digital systems depend on the effective transfer and logging of data. The ESP32 is a great choice for real-time data transmission due to its wireless communication capabilities. Studies have underscored the need of robust and secure data recording systems, guaranteeing that critical data is precisely recorded and readily available for examination and judgment.

- **Designing of PCB :**

  We have designed the PCB of this overall circuit.

- **Designing and printing of casing 3D model:**

  We have designed its casing model in solid works and gets its 3D printing done where we will place our display and PCB.

- **Case Studies:**

  Multiple case studies have showcased the effective deployment of digital driver panels across different military vehicles. These studies offer valuable perspectives on the difficulties and remedies involved in the shift from analog to digital systems. They emphasize the advantages of heightened data precision, enhanced user interfaces, and seamless integration with other digital systems.

2.1.3 Related Work:

The development of a digital driver panel for a tank that includes an odometer, speedometer, and gyrocompass makes use of a number of important technological and research advances in the domains of embedded systems, digital display interfaces, and sensor integration. This section examines the foundational research and associated works that guide and support this undertaking. Tank VT4 has Digital Driver Panel installed in it.

- **Odometer Systems:**

  Finally, digital types of odometers, which offer better accuracy and additional interfacing options, have displaced mechanical odometers mainly.

- **Speedometer Systems:**

  Using accelerometers and hall effect sensors, systems of digital speedometers can provide near exact current speed values in real time. These systems have undergone developments over and above these by means of including standards such as adaptive speed dials and built-in vehicle telematics.

  Key to increasing the accuracy and sensitivity of speed measurements has been the goal of

  This field of research, particularly with regard to the growth of traffic patterns on the roads.

- **Nextion Display Technology:**

  Nextion displays have, therefore, become popular in embedded systems as they offer a strong platform for interfacing with HMI (Human Machine Interface).

  The basic framework that is being discussed here includes Ergonomics/HSSE (Health, Safety, Social, Environment), Machine Interface) features and user-friendliness. They help in developing multifaceted touch-enabled graphical user interfaces. Research and practical experience reveals that

nextions can be of great impact in increasing the engagement level and data presentation in

embedded devices.

- **HMI in Military Vehicles:**

  HIMS (Human Machine Interface) are installed in modern tanks nowadays.The features that Nextion displays are endowed with include a highly effective HMI (Human-Machine Interface) and the users.

- **Integrating Sensors into Vehicle Systems:**

  Systems with Gyrocompasses: Military and commercial vehicle navigation systems often use gyrocompasses. Instead of relying on magnetic fields for directional, they send precise information based on the Earth's rotation. Increasing precision and dependability in harsh environments—like those military tanks face—has been the goal of research and development in this field. Research indicates that compact, reliable navigation systems appropriate for cars with high vibration levels and tight spaces can make use of MEMS based gyrocompasses.

2.2 Conclusion:

The military has improved its tanks by changing the traditional to new digital control platforms: a milestone. Utilizing advanced end-user interfaces (e.g. Nextion display); powerful microcontrollers (e.g. ESP32) contemporary sensing methods aim at enhancing the precision, reliability as well as quality of essential operational data in this study. In examining the literature it is noted that significant benefits outweigh minor drawbacks.

Chapter 3

Materials and components used

A microcontroller, a display unit, and several sensors are integrated into the system architecture and design of a digital driver panel for a tank in order to monitor and display real-time data. The architecture, design, and interactions between parts that result in the intended functionality are described in this section. The digital driver panel comprises the following main components:

3.1 Hardware

3.1.1 Hall Effect Sensor

The hall effect sensor is a type of magnetic sensor which can be used for detecting the strength and direction of a magnetic field produced from a permanent magnet or an electromagnet with its output varying in proportion to the strength of the magnetic field being detected.

Magnetic sensors convert magnetic or magnetically encoded information into electrical signals for processing by electronic circuits.

Magnetic sensors are solid state devices that are becoming more and more popular because they can be used in many different types of application such as sensing position, velocity or directional movement.

They are also a popular choice of sensor for the electronics designer due to their non-contact wear free operation, their low maintenance, robust design and as sealed hall effect devices are immune to vibration, dust and water.One of the main uses of magnetic sensors is in automotive systems for the sensing of position, distance and speed. For example, the angular position of the crank shaft for the firing angle of the spark plugs, the position of the car seats and seat belts for air-bag control or wheel speed detection for the anti-lock braking system, (ABS). [2]

Figure 3.1: Hall Effect Sensor

Figure 3.2: Hall Effect Sensor Mounted on tyre

3.1.2 Gyrocompass Sensor

MPU6050 is a MEMS-based 6-axis motion tracking device. It has an on-chip gyroscope and accelerometer sensors along with temperature sensor. MPU6050 is a digital device. This module is of very small in size, has low power consumption requirements, highly accurate, has high repeatability, high shock tolerance, it has application-specific performance programmable and low consumer price points. MPU6050 can be easily interfaced with other sensors such as magnetometers and microcontrollers. The gyroscope present in MPU6050 can detect rotation about the three axis's X, Y, Z. Coriolis effect causes a vibration when the gyros are rotated about any of the axes. These vibrations are picked up by the capacitor. The signal produced is then amplified, demodulated and filtered to produce a voltage that is proportional to the angular rate. This voltage is then digitized using ADC's.[3]

Figure 3.3: Gyrocompass Sensor

Figure 3.4: MEMS Accelerometer

3.1.3 ESP32 Microcontroller

ESP32 is a series of low-cost, low-power system on a chip microcontrollers with integrated Wi-Fi and dual-mode Bluetooth. The ESP32 series employs either a Ten silica Xtensa LX6 microprocessor in both dual-core and single-core variations, Xtensa LX7 dual-core microprocessor or a single-core RISC-V microprocessor and includes built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power-management modules. ESP32 is created and developed by Espressif Systems, a Chinese company based in Shanghai, and is manufactured by TSMC using their 40 nm process. It is a successor to the ESP8266 microcontroller.[4]

3.1.4 Nextion Display

Nextion is a Human Machine Interface HMI solution combining an onboard processor and memory touch display with Nextion Editor software for HMI GUI project development.

Using the Nextion Editor software, you can quickly develop the HMI GUI by drag-and-drop components (graphics, text, button, slider, etc.) and ASCII text-based instructions for coding how components interact on the display side.

Nextion HMI display connects to peripheral MCU (Micro controller unit) via TTL (transistor-transistor logic) Serial (5V, TX, RX, GND) to provide event notifications that peripheral MCU can act on, the peripheral MCU can easily update progress, and status back to Nextion display utilizing simple ASCII text-based instructions.[5]

Figure 3.5: Nextion Display

3.1.5 Power Supply

Gives stable power to all the components.

3.2 Components and their roles

3.2.1 Hall effect sensor

Using this sensor Distance and rotational speed are measured, RPM (Rotation Per Minute) corresponding pulses are the output. Interface is Connected to the GPIO pin of an ESP32.

3.2.2 Sensor Gyrocompass

Determines the orientation of the tank and the degrees of orientation data are the output. Interface is made by Connecting to the ESP32 by I2C.

3.2.3 ESP32 Microcontroller Function

The function of this microcontroller is to read the sensor data and display it over the screen or over the HMI that we have used.

3.3 Interface

Uses I2C (a two-wire serial communication protocol using a serial data line (SDA) and a serial clock line (SCL)) interface to communicate with the ESP32. Power Supply is also connected to it. Giving the sensors, ESP32, and displaying the voltages they require.

3.4 Software used

3.4.1 Arduino IDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.rograms written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor. [6]

3.4.2 Nextion editor

This document goes through various features of the current Nextion Editor. The Nextion Editor is used to rapidly create Human Machine Interface GUIs for Nextion HMI devices. As such the GUI can be created within Hours instead of Weeks, and Days instead of Months. So while we won't be covering basics such as opening a file, we will point out somethings that might prove helpful to know, or reminders need be made.

Note: Nextion Editor has indeed evolved since its early beginnings, so I would like to take a moment for a quick review. As time has passed, many additional features and bug fixes were incorporated. The Nextion Editor is not expected to retain every previous behaviours between versions exactly. With the new, then there are indeed new behaviours and new possibilities. Features:

3.4.2.1 Editor

It is a kind of designing platform where we can create GUI's by designing thinking dragging and dropping and giving accurate sizes or ides to respective text boxes and gauges.

3.4.2.2 Component Library

The program comes with a large library of pre-made widgets and components that you can modify to fit your project's requirements. Basic components (buttons, text), sophisticated components (gauges, sliders), and even timers are examples of components.

3.4.2.3 Event Handling

Event handlers can be added to any component, enabling the development of dynamic and interactive interfaces. Certain commands or actions can be triggered by events like button clicks

3.4.2.4 Code Generation

The code required to operate the interface on the Nextion display can be generated by Nextion Editor. It allows for straightforward scripting within the editor to support more sophisticated behaviors and logic.

3.4.2.5 Simulation

Nextion Editor is a powerful development tool that's designed to create interactive user interfaces for Nextion screens. Human-machine interface (HMI) displays from Nextion are widely used in many embedded systems, including Internet of Things (IoT) initiatives, home. Before deploying the GUI to the actual hardware, developers can use the integrated simulator to test and debug it. It saves time because content doesn't need to be regularly uploaded to the device.
Firmware Upload: The software allows you to transfer the completed graphical user interface (GUI) to the Nextion display via an SD card or USB.
Data Visualization: Because the editor supports a variety of data visualization approaches, it is suitable for dashboards and real-time monitoring systems,industrial control, automation, and other things that developers could design. [7]

3.5 Solid works

Solid Works is a solid modeler, and utilizes a parametric feature-based approach which was initially developed by PTC (Creo/Pro-Engineer) to create 3D CAD models and assemblies. The software uses the Parasolid modeling kernel.[33]

Parameters refer to constraints whose values determine the shape or geometry of the model or assembly. Parameters can be either numeric parameters, such as line lengths or circle diameters, or geometric parameters, such as tangent, parallel, concentric, horizontal or vertical, etc. Numeric parameters can be associated with each other through the use of relations, which allows them to capture design intent.[8]

3.6 Proteus

We have used Proteus software for making of PCB Design from our hardware circuit and its schematics.Proteus ISIS is used by Engineering students and professionals to create schematics and simulations of different electronic circuits.[9]

3.7 Conclusion

A micro controller, display unit and a variety of sensors necessary for monitoring and displaying live data are integrated into the digital driver panel of a tank. Major parts used in its production are as follows: Hall Effect sensor for speed and distance measurement, gyrocompass sensor to identify orientation, ESP32 micro controller for processing data, Nextion display for user interface as well as power supply. In addition, this architecture is used to link its components via interfaces such as GPIO or I2C.The Arduino IDE is necessary for programming robots. You use the Nextion Editor for constructing and testing graphical user interfaces (GUI's). When developing casings, Solid Works proves the best package as it assists in modelling cases in 3D. This software is used when designing PCBs (Printed Ciruit Boards) as well as coming up with diagrams electronically. This combination helps ensure better control systems during operations of tanks.

Chapter 4

Methodology

As part of the system architecture, an ESP32 microcontroller is incorporated with three sensors: a Nextion display for user interface, a Hall effect sensor for speed and distance measurement, and a gyrocompass sensor for orientation. Real-time numbers are shown on the Nextion display once the ESP32 has processed the sensor input. A specifically designed PCB housing the ESP32 and the sensors ensures proper power control and signal routing. The entire assembly, which consists of the PCB and Nextion display, is secured in a laser-cut acrylic enclosure designed specifically for that purpose. The digital driver panel on this glass enclosure is dependable and efficient for the tank while using the Arduino IDE to integrate Sensor Data.

4.1 Hardware Configuration

4.1.1 Speedometer Sensor

A speedometer uses a hall effect sensor. After determining whether a magnetic field is there, it generates a pulse for every revolution.

```
unsigned long lastTime = 0;
const unsigned long interval = 1000; // Interval in milliseconds (1 second)
float totalDistance = 0; // Total distance traveled

// Define the pin where the Hall effect sensor is connected
#define HALL_SENSOR_PIN 15

void IRAM_ATTR handlePulse() {
  // Increment the pulse count
  pulseCount++;
}

// Variables for angle calculations
float prevTime = 0.0;
float angleZ = 0.0;
float gyroOffsetZ = 0.0;  // Offset to correct drift

void setup() {
  // Initialize serial communication
  Serial.begin(115200);
  nextionSerial.begin(9600, SERIAL_8N1, 16, 17);  // Start serial communication with the Nextion display

  // Setup MPU6050
  while (!Serial)
    delay(10);

  Serial.println("Initializing sensors...");

  if (!mpu.begin()) {
    Serial.println("Failed to find MPU6050 chip");
```

Figure 4.1: Initialization code

Speed Calculation: The handle Pulse ISR increments the pulse Count every time the Hall effect sensor detects a pulse (e.g., a wheel rotation). Every interval (1 second), the frequency of the pulses is calculated: **float frequency = pulse Count / (interval / 1000.0)**; This converts the pulse count into a frequency in Hertz (pulses per second). Using the frequency, the speed is calculated with the given formula:

int speed = (int)(5.9645 * frequency);

Odometer Sensor: By counting the number of pulses over time, the hall effect. Sensor may also be used to determine the distance traveled.

```
// Speed and distance calculations
unsigned long millisCurrentTime = millis();
unsigned long elapsedTime = millisCurrentTime - lastTime;

if (elapsedTime >= interval) {
  // Calculate the frequency in Hz as a float to allow decimal places
  float frequency = pulseCount / (interval / 1000.0);

  // Calculate the speed using the given formula
  int speed = (int)(5.9645 * frequency); // Convert to whole number

  // Calculate the distance traveled in this interval (speed * time in hours)
  float distance = speed * (elapsedTime / 3600000.0); // elapsedTime is in milliseconds, convert to hours

  // Update the total distance
  totalDistance += distance;

  // Print the speed, distance, and angle to the Serial Monitor
  Serial.print("Speed: ");
  Serial.print(speed);
  Serial.print(" km/h, Distance: ");
  Serial.print(totalDistance, 1);
  Serial.print(" km, Angle: ");
  Serial.println(angleZ, 1);
```

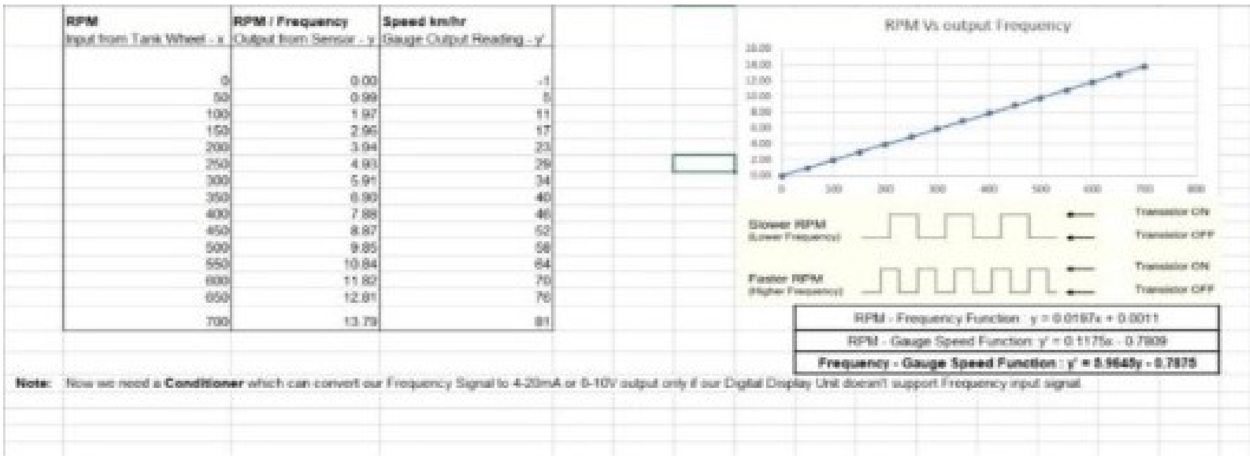
Figure 4.2: Speed and Distance Calculation

Distance Calculation: The elapsed time for the interval is measured in milliseconds: unsigned long elapsed Time = millisCurrentTime - last Time; The distance traveled during this interval is then calculated using the speed and the elapsed time:

float distance = speed * (elapsedTime / 3600000.0);

// elapsedTime is in milliseconds, convert to hours Here, elapsedTime / 3600000.0 converts the elapsed time from milliseconds to hours, since speed is in km/h. The calculated distance for the current interval is added to the totalDistance:

totalDistance += distance;



RPM Input from Tank Wheel - x	RPM / Frequency Output from Sensor - y	Speed km/hr Gauge Output Reading - y'
0	0.00	-1
50	0.99	5
100	1.97	11
150	2.96	17
200	3.94	23
250	4.93	29
300	5.91	34
350	6.90	40
400	7.88	46
450	8.87	52
500	9.85	58
550	10.84	64
600	11.82	70
650	12.81	76
700	13.79	81

Figure 4.3: Speed Frequency Graph

4.1.2 Gyrocompass

To ascertain the orientation and heading of the tank, an IMU (Inertial Measurement Unit) sensor is utilized, such as the MPU-6050. Linking Sensors with the ESP32:

- Attach the VCC of the Hall effect sensor to the ESP32's 3.3V pin .
- Attach the sensor's GND pin to the ESP32's GND
- Attach the sensor's output pin to one of the ESP32's GPIO pins (such as GPIO 2).

```
}

void loop() {
  // MPU6050 sensor data
  sensors_event_t a, g, temp;
  mpu.getEvent(&a, &g, &temp);

  float gyroZ = g.gyro.z * (180.0 / M_PI) - gyroOffsetZ; // Apply offset

  // Calculate time difference
  float currentTime = millis() / 1000.0; // Convert to seconds
  float deltaTime = currentTime - prevTime;
  prevTime = currentTime;

  // Calculate change in angle only when there is rotation
  if (abs(gyroZ) > 0.7) { // Adjust threshold as needed
    float deltaAngleZ = gyroZ * deltaTime;
    angleZ += deltaAngleZ;

    if (angleZ >= 360.0) {
      angleZ -= 360.0;
    } else if (angleZ < 0) {
      angleZ += 360.0;
    }
  }
```

Figure 4.4: Gyro calculation

Gyro Calculation:

Offset initialization:

• The offset is initially set to 0.0

float gyroOffsetZ = 0.0; // Offset to correct drift

Applying Offset:

The offset is subtracted from the gyroscope's Z-axis reading before further calculations.

float gyroZ = g.gyro.z * (180.0 / MPI) - gyroOffsetZ; // Apply offset

Periodic Adjustment:

The offset is periodically adjusted to correct for any drift. In this code, it is adjusted every 10 seconds.

if (millis() **gyroOffsetZ += gyroZ * 0.01**; // Simple adjustment based on current reading

This adjustment is a simple method that gradually corrects the offset based on the current reading, helping to reduce long-term drift.

Gyro Threshold: The threshold for the gyroscope's Z-axis reading is set to 0.7 degrees per second.

```
if (abs(gyroZ) > 0.7)

// Adjust threshold as needed
```

Applying the Threshold:

If the absolute value of the gyroZ reading is greater than the threshold, it is considered significant, and the angle is updated.

```
float deltaAngleZ = gyroZ * deltaTime;

angleZ += deltaAngleZ;

if (angleZ >= 360.0)  angleZ -= 360.0;  else if (angleZ < 0)

angleZ += 360.0;
```

Purpose and Impact:

• **Offset:**: Helps to maintain accuracy over time by compensating for slow drift in the gyroscope readings.

• **Threshold:**: Filters out minor noise and small movements to avoid false readings and ensure that only meaningful rotations are considered.

4.1.3 Software Setup

Install the Arduino IDE and configure the ESP32 board by installing the board package and adding the ESP32 board URL to the settings. To read the pulses from the Hall effect sensor, write code.

We are using **I2C** interface for sesnors integration with micrcontroller and **UART** (Universal Asynchronous Receiver Transmitter) for integration with Nextion display as both the hardwares are different so they are communicating at different frequencies (baud rates). Sensors are communicating at the baud rate of **115200**, while our HMI is communicating at the baud rate of **9600**.

4.1.4 Gyrocompass Sensor Integration

Attach the SDA and SCL of the gyrocompass sensor to GPIO 21 and GPIO 22, respectively, on the ESP32.

4.2 Designing the GUI

Download and Install Nextion Editor:

Install the Nextion Editor on your PC after downloading it from the official Nextion website.

Figure 4.5: GUI Design Made on Inkspace

Use a USB-to-TTL (transistor-transistor logic) converter to connect your computer to the Nextion display. Using the Nextion Editor, upload the project to the Nextion display.

4.3 Software Setup

Lay out an association between the ESP32 and the Corridor impact and gyrocompass sensors involving the Arduino IDE for sensor combination. Make a GUI on the Nextion Editor by composing code to peruse sensor information. Make the showcase's components and format. Transfer the point of interaction that has been worked to the Nextion show. Laying out the equipment associations between Nextion Show and ESP32 is the most vital phase in coordinating them. To alter the showcase using sensor information, compose the code. By following this thorough interaction, you can be sure that your task will precisely assemble sensor information and show it on a Nextion screen, creating a far-reaching computerized driver Panel for the tank.

4.3.1 ESP32

This microcontroller was chosen because it has a large number of GPIO pins and WiFi capabilities and also it is easily available Connect the digital input pins of the ESP32 to the hall effect sensors by wires . The IMU 6050 sensor is connected to the I2C interface through the SDA and SCL pins. Programming with the Arduino IDE:
Install and downloads the compulsory libraries (Ada fruit MP6050, , if you're using an) for the sensors.

4.3.2 Code Development

Write the respective code to read sensor data. Use interrupt service routines (ISRs) to count pulses in hall effect sensors. Read the orientation data for the IMU.
Information Processing: Transform pulse counts into distance and speed. Utilize the MPU6050 information to determine the heading.

```
    // Convert totalDistance to an integer value for the gauge object "odog"
    int distanceInt = static_cast<int>(totalDistance);
    int distanceGaugeValue = static_cast<int>((distanceInt / 500.0) * 180.0); // Assuming total distance range is 0-500 km
    nextionSerial.print("odog.val=");
    nextionSerial.print(distanceGaugeValue);
    nextionSerial.write(0xFF);
    nextionSerial.write(0xFF);
    nextionSerial.write(0xFF);
  }

  // Send angleZ value to Nextion display number object "gyron" and gauge object "gyrog"
  nextionSerial.print("gyron.val=");
  nextionSerial.print(static_cast<int>(angleZ)); // Cast to int to ensure it's sent correctly
  nextionSerial.write(0xFF);
  nextionSerial.write(0xFF);
  nextionSerial.write(0xFF);

  int angleGaugeValue = static_cast<int>((angleZ / 360.0) * 360.0); // Normalize angleZ to full circle
  nextionSerial.print("gyrog.val=");
  nextionSerial.print(angleGaugeValue);
  nextionSerial.write(0xFF);
  nextionSerial.write(0xFF);
  nextionSerial.write(0xFF);

  // Periodically adjust gyro offset (you can adjust timing as needed)
  if (millis() % 10000 == 0) {  // Adjust every 10 seconds
```

Figure 4.6: Integration with Nextion

Once we had information about the ESP32 sensors in the serial monitor, we connected the Nextiom by transmitting the analyzed values through UART. The ESP32 then communicates with the Nextion display to refresh the GUI in real time because what is shown on that display will be relevant to the data being captured by the sensors. Having established the data of the sensor on the serial monitor of the ESP32, we then relay these values to particularNextion display objects via UART(Universal Asynchronous Receiver Transmitter) transmission. For example, the gyroscope data is written to the Nextion GUI object labeled as **"gyron"** and the gauge object labeled as **"gyrog"** and similarly total distance is written to the Nextion GUI object labeled as **"odog"** with formatted command sent from the ESP32 and received by the Nextion which writes the most updated sensor reading to the that particular object.

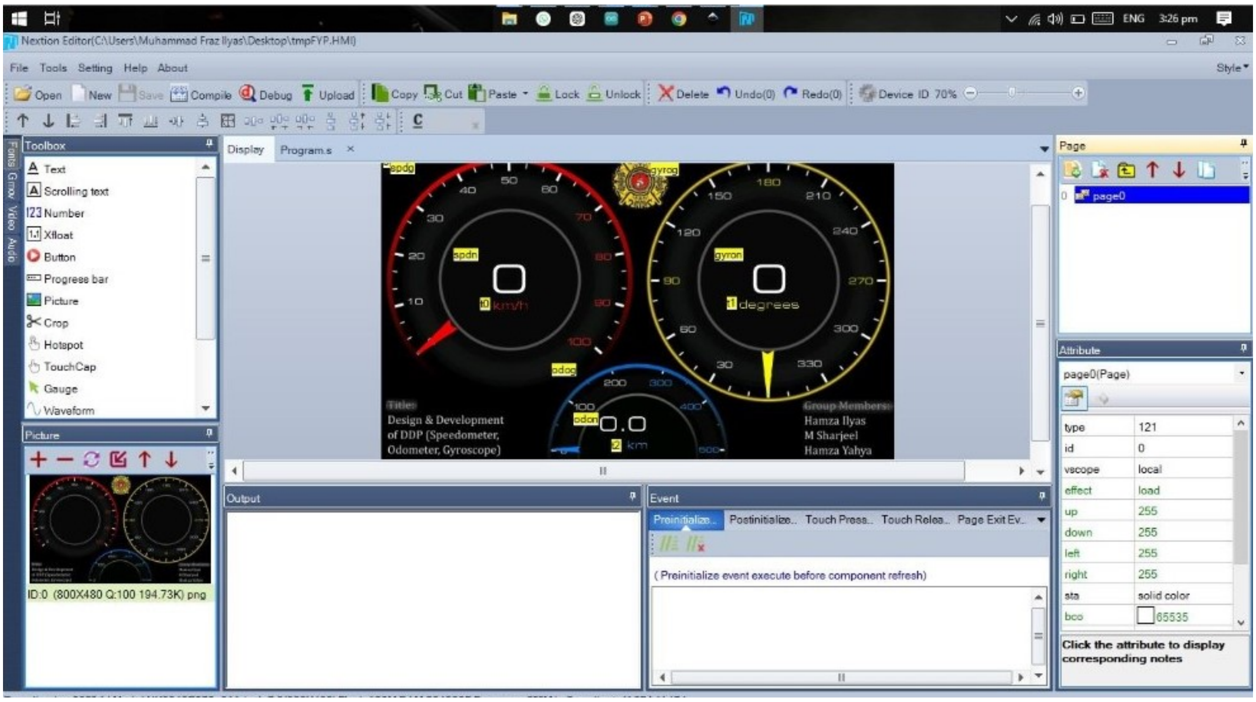


Figure 4.7: Using Nextion Editor to Create a GUI

4.4 Modeling and PCB Design

4.4.1 PCB Design

4.4.1.1 Software

To design the PCB layout, use Proteus

Components: Arrange the ESP32, sensors, voltage regulators, and connectors among other components on the PCB layout.

Routing: Use the proper tracks to connect the components.

4.4.2 3D Modeling

Software: To design the enclosure, use Solid Works.

4.4.2.1 Dimensions

Build a 3D model of the enclosure that is exactly the right size to accommodate the Nextion display and PCB. To communicate between the ESP32 and Nextion display, use the Software Serial library. Sending Data: To communicate sensor data to the Nextion display, write code in the Arduino IDE.

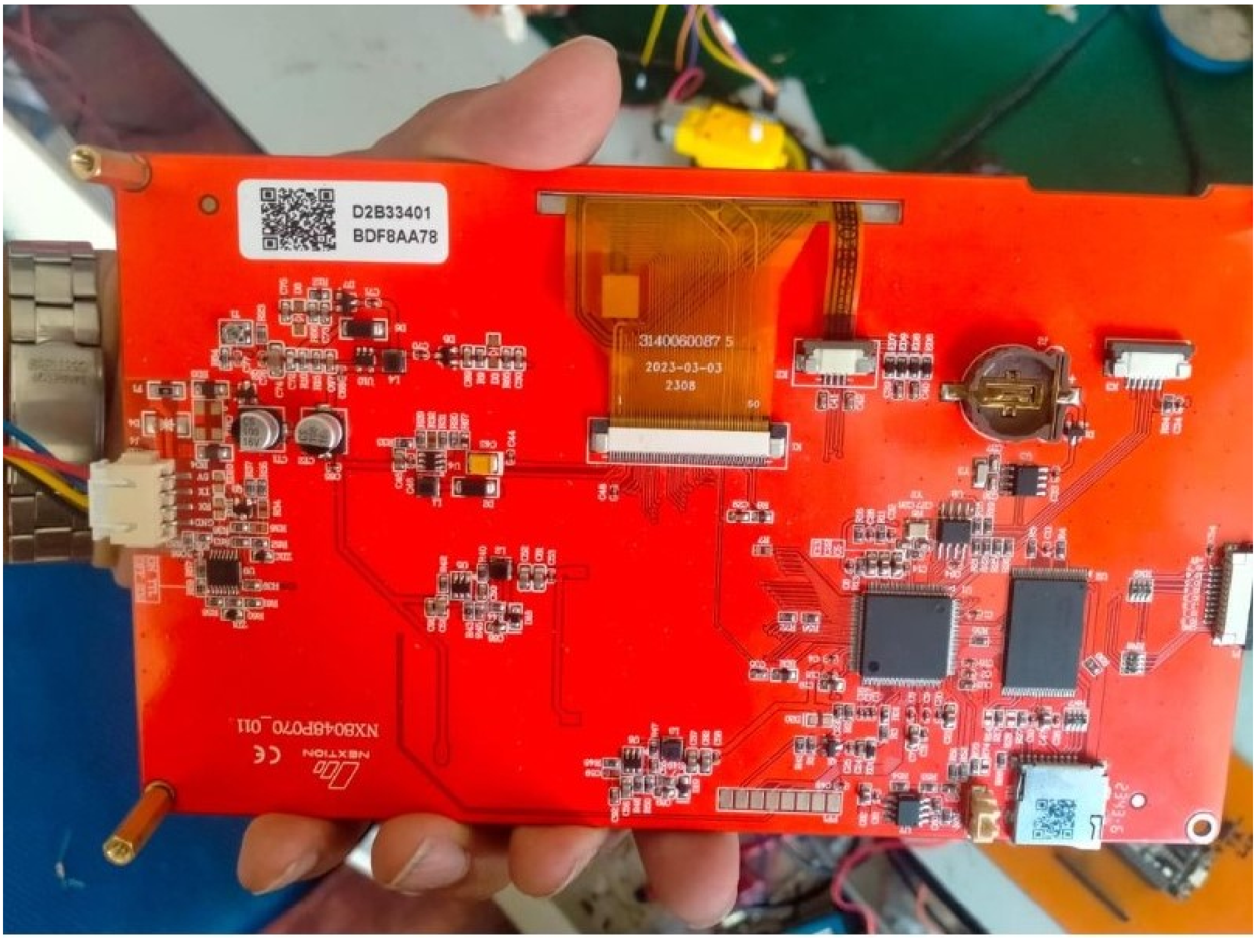


Figure 4.8: Nextion Display

4.5 PCB Schematic

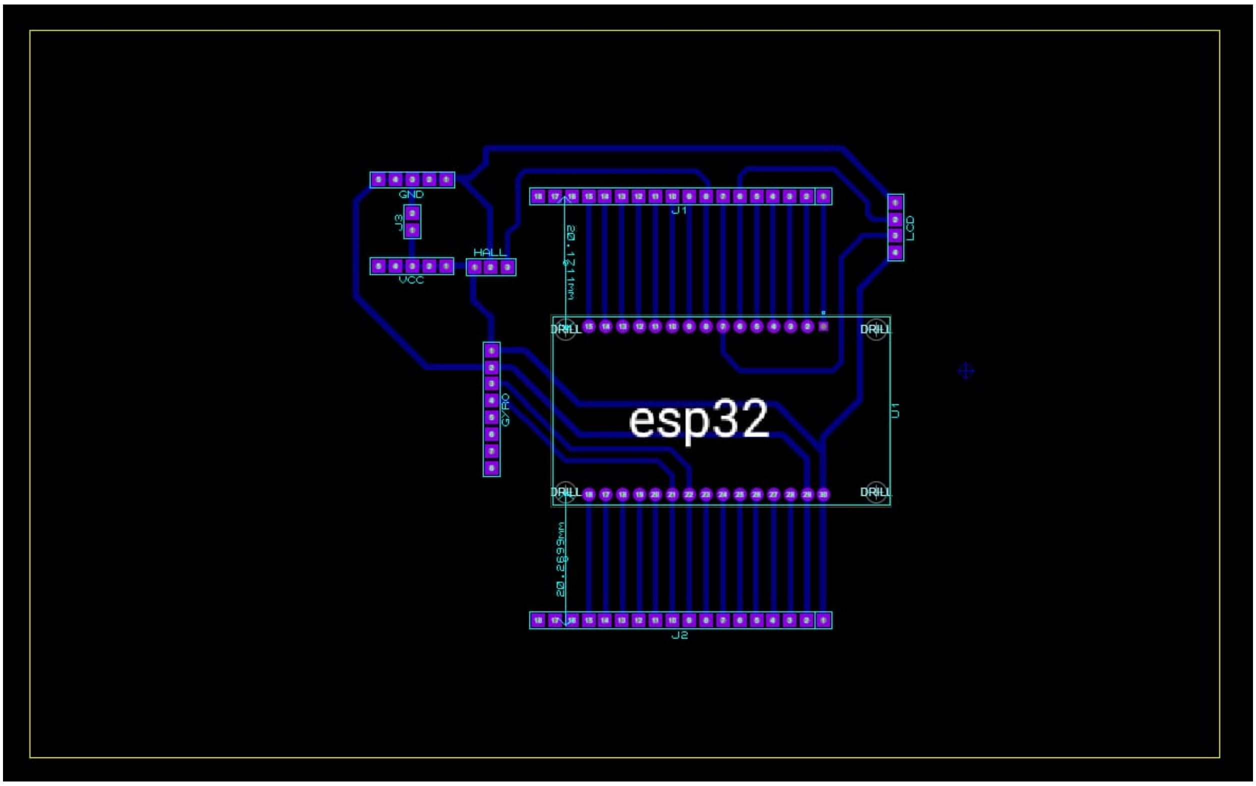


Figure 4.9: PCB Schematic

As you can see, the pcb includes , ESP32 module, Hall effect sensor, gyrocompass sensor, and any passive parts that are required (resistors, capacitors). As required by the ESP32 and sensors, the input voltage is regulated from 24V to either 3.3V or 5V. To measure distance and speed, connect the Hall effect sensor to one of the ESP32's GPIO pins. The item To obtain orientation data, connect the gyrocompass sensor to the ESP32's I2C pins. To safely link high voltage components with the ESP32, include an buck converter.

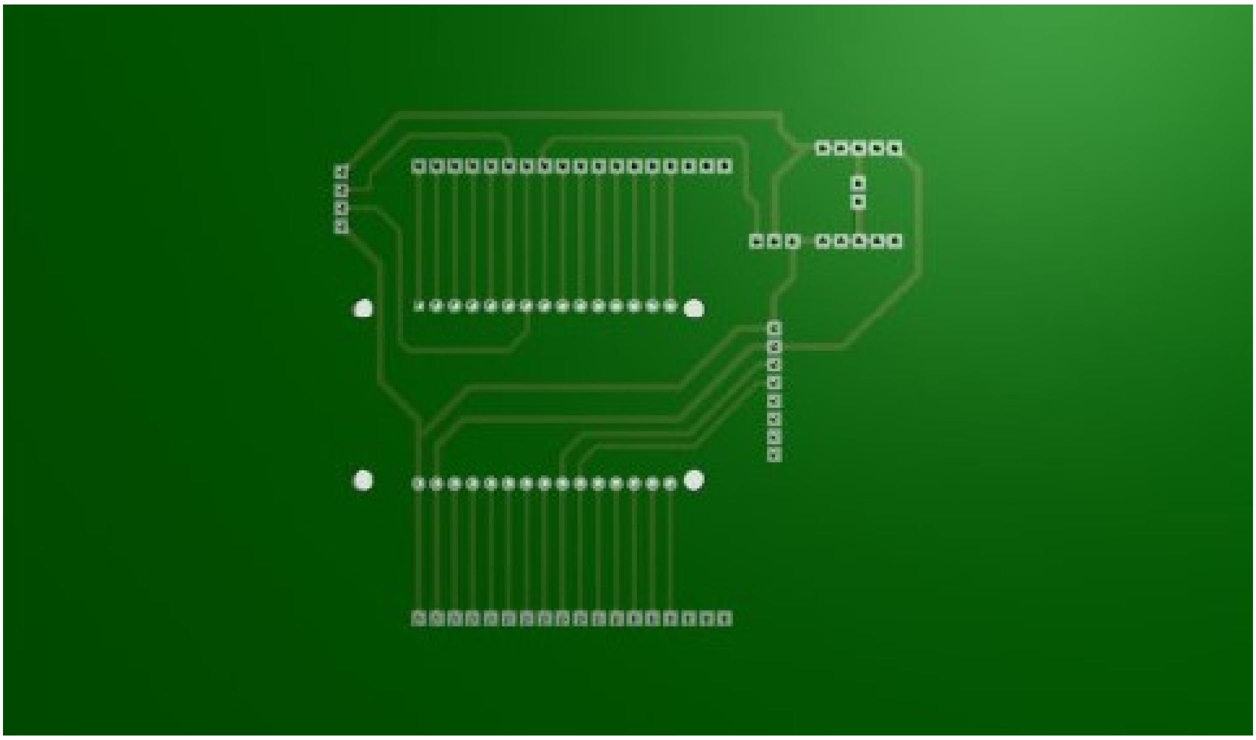


Figure 4.10: Schematics on Proteus

4.6 PCB Layout

4.6.1 Component Placement

Assemble the parts in a small, well-organized package, making sure that the trace lengths are as short as possible to maintain signal integrity. To minimize noise, maintain ground planes and power continuously.

4.6.2 Routing

As you route the traces, be sure that the power lines have enough trace widths and the signal lines have little interference. If necessary, use many layers and Visa in intricate designs.

4.6.3 Gerber Files

After the design is finished, produce Gerber files for the production process. The item Copper layers, solder mask, silkscreen, and drill files are among these files.

4.6.4 Component Soldering

Solder the components onto the PCB as soon as the PCB is received. The item For SMD components, use a refile oven or soldering tools. **Testing:** Test the PCB after assembly to make sure all connections are made correctly and all parts are working. The item Check that the sensor readings and the Nextion display are communicating.

4.7 3D Model Design

4.7.1 Software

We have Used CAD software such as Solid Works, to make our 3d model.

4.8 Laser Cutting

Export the 3D model to a laser-cutting format (for 2D profiles, DXF, for example). Cut the acrylic sheets into the desired shapes using a laser cutter. Put the sliced acrylic pieces together to create the enclosure. Use screws or adhesive to hold the PCB and LCD in place inside the enclosure. Assemble every part, making sure the sensor connections are tight and the PCB is firmly affixed to the Nextion display.

4.9 Conclusion

If you are planning to have a digitized panel for driving a tank, here are the things that you will need. First, get an ESP32 microcontroller as shown below; This little gadget here will assist to filter the signals originating from the sensors we are going to discuss later on. In addition to convenience, of course, speaking of sensors, you also might need a Hall effect sensor. One of them calculates the speed of your tank and the distance to be covered. But wait, there's one more component that must not be omitted, the gyrocompass sensor! I wonder if this can be marketed as compass that resembles the shape of the tank and also helps to determine in which direction it is aimed.

Well, friends, now it is high time to discuss how everything fits into the picture. From the sensors, the ESP32 receives signals and then provides real-time information on an Industrial Nextion display. Mary looks at it and said: It is as if I have a view through to the inner control mechanism of the tank. Overview At the core of it, to set it all up, you will be needing to create settings for the sensors and calibrate the ESP32 through the Arduino IDE. For power management and to keep all the components clean and in order, you will need to design PCB which stands for printed circuit board.

Now, let's move to the third key area of operation – the software. To work with this data, you will have to create an appropriate script that would read out the data from each of the sensors. It will also ensure all is well since timing is very important when preparing meals especially when people are hungry. Well, do not panic, they will not be asking you to become a coding whiz for that. After that is in place, the next thing, if you haven't done so already, is to design a graphical user interface on the Nextion display. In this manner when the user wishes to perform a certain task the user can easily interact with the system.

Finally, let me tell you how all the pieces of the pyramid fit into a cohesive conceptual framework. You will enclose all the components with a protective acrylic casing made from solid works. Actually it is about how to make ends meet that means how they must fit and connect in a tight and sound manner. And regarding the design of the PCB, well it has to be taken into consideration again. There is also a program to which, after getting into the labyrinth of cables and connectors, you can check that everything is grouped in the best way and does not take up extra space, and this program is called Proteus. After that, you can always run it to check for any problem that might be occurring within the computer system.

There you have it! By making this kind of setup, you will have an interface to control your tank via your digital driver panel and this will not just be efficient but also stylish.

Chapter 5

Testing and Validation

To achieve the data visualization objectives set in the project, we require the creation of GUIs for data acquisition from the Firebase Database, as described in Chapter 4. This means creating an app that is usable on a mobile device as well as an app that can be used on a PC.

The development of the user interfaces for both the mobile and web applications is described extensively in this chapter, as well as the architecture of their integration with Firebase Database and the approaches employed in data acquisition.

5.1 Component Testing:

5.1.1 Sensors:

5.1.1.1 Hall Effect Sensor:

Cut the pulse frequency by using oscilloscope and turning the magnet at given phase.

RPMs to ensure that the pulse is produced We also ensure that checksums are correct to validate RPMs.

5.1.1.2 Sensor MPU-6050 (IMU):

Make sure that the measured acceleration and gyroscope values by sensor readings are correct or not comparing with the known angle and motion.

5.1.1.3 ESP32:

Connect a microcontroller to a computer and use a serial monitor to display the raw values for the evaluation of the data sheets for the sensor.

5.2 ESP32 with Sensors:

Take the code and successfully interface it with the ESP32 along with the sensors. However, the printer itself can issue the sensor data to the

As you will see, serial monitor lets you check if it is properly read

Link the Nextion display to the ESP32 and make sure that the sensor data is correctly delivered and displayed on the Nextion interface. Testing for functionality:

- **Measuring Speed:** A known speed should be compared to the ESP32 output to validate the speed calculation. Compare the speed determined by the ESP32 with a reading from a tachometer by rotating the hall effect sensor at various rates.
- **Odometer:**

  Functionality involves simulating distance by operating the sensor at a steady speed for a predetermined amount of time, then confirming that the resultant distance matches the predicted figure.
- **Gyrocompass Accuracy:**

  Turn the tank in predetermined directions, then use a compass to compare the output of the gyrocompass with its heading.

5.3 Testing the System:

5.3.1 Complete Integration of Systems:

With every sensor linked to the ESP32 and real-time data shown on the Nextion display, test the entire system. Verify the accuracy of the data and the display refresh rates.

5.3.1.1 Testing in the field:

Test the system under real operating conditions after installing it in the tank. Monitor the stability of the system over time and the accuracy of the sensor data.

5.3.1.2 Verification of Correctness:

Align each sensor to guarantee that the results are inside allowed mistake edges. Change the product's alignment factors as the need might arise.

5.3.1.3 Comparative analysis:

Contrast the sensor results and those from normal hardware (such a tachometer, GPS for distance, and computerized compass) to guarantee precision.

5.3.1.4 Consistency of Data:

Verify that the data is consistent throughout multiple test runs carried out under the same conditions to confirm the repeatability of measurements.

5.3.1.5 Latency:

Decide the framework's reaction time from sensor contribution to the result that is displayed on the Nextion screen. Guarantee that the postponement is just about as negligible as possible and stays inside OK boundaries.

5.3.1.6 Reliability:

Do broad testing to approve the framework's trustworthiness. Confirm that there hasn't been any critical information misfortune or float over the long haul.

5.3.1.7 Ecological Testing:

Check the framework's exhibition in various natural circumstances, like fluctuating temperatures and moistness, to guarantee heartiness.

5.3.1.8 User Validation:

Testing for convenience: Test the point of interaction with genuine clients, such tank administrators, to guarantee that it is instinctive and simple to utilize. Acquire input regarding the framework's responsiveness and clarity of the showcase.

5.3.1.9 Feedback Iteration:

Utilize user input to improve the system's operation and usability.

5.3.1.10 Standards for Safety:

Verify whether the framework consents to all pertinent well-being and working rules for electronic gadgets utilized in tanks.

Ensure the framework consents to all regulations and rules relating to electronic and correspondence hardware utilized in military or business conditions. We call this administrative consistence.

Figure 5.1: Digital Driver Panel

5.4 Conclusion

When it comes to testing and validating the digital driver panel, we go through several stages to make sure everything works smoothly, accurately, and reliably. First, we test the different components. We use an oscilloscope to check the pulse frequency of the Hall effect sensor and compare it to known references. We also check the acceleration and gyroscope values of the MPU-6050 sensor against these references. To see if the ESP32 can read the sensor data correctly, we use a serial monitor.

After that, we proceed to the integration test step. This is where we check that ESP32, sensors, and Nextion display function jointly, and the collected information is shown properly. There are certain definitive tests that are applicable in an assessment of speed, testing of theodolite capability, and determination of the proper functioning of the odometer and the gyrocompass likewise.

System testing, the fourth stage, follows integration testing. This is the step where the all gadgets undergo tests to confirm that all are o.k., and all are in harmony. Those scenarios can be applied to the information presented in the paper be brought into the real world and apply the system to the field. We also ensure that the data is accurate and consistent, there is hardly any delay when processing the data or when the system is used, and that the system is stable hence not failing at any time.

This is also true with Environmental testing that comes as the next step. Since the system is designed to work as a facilitator in a wireless environment, we check on the performance of the system when subjected to different conditions. And of course, we also want to focus on the fact that it is easy to use for regular people who go to sites. They include things like user validation testing, where actual users go through the software depending on their preferences and give feedback.

Finally, ensuring the electronic devices positioned in tanks meet the protection and functionality requirements. This meant that every component has to be in check so that every element of safety is as perfect as anyone could make it.

Thus, passing through these different stages of testing and validation, it is possible to be certain that this digital driver panel is working in the correct manner and possess all the features it should have.

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

So, thanks to that, we were able to create a really cool digital driver panel for a tank. In other words, we linked an ESP32 micro controller to some sensors and a Nextion as a display to monitor and record working details in real time. To ensure that the structure was strong and durable, we incorporated a Hall effect sensor for the determination of speed and distance, a gyrocompass sensor for recording the orientation of the system, and a custom PCB design for power management and signal distribution. We then placed everything to protect them in a strong acrylic housing that we made using a laser cutter. What we have here is a perfect tank monitoring system that will work like charm.

Okay, let's switch to thinking about things from a software perceptive. For controlling the ESP32, we used the Arduino Integrated Development Environment and for the Nextion, we used Nextion Editor for Easy Viewing. We decided to give it the real pressure the first time to test whether it would be able to read and display sensors accurately or not and trust me it stunned me the most for doing it so perfectly. Regardless of the conditions that we put it through, it remained stable and kept good accuracy. We even went to the field to test the equipment for real and we also had some of the users to try the equipment and they agreed it was easy to use and manipulate as well as being safe and sensitive to operations.

This article perfectly demonstrates how to design and develop a safe and efficient embedded system for potentially life-critical applications. It just underscore the need for hardware and software components to be properly integrated in order to fetch and process that data in real-time and have a neat, friendly user interface. It's all about having it running effectively and this project was an ideal example of functionality.

6.2 Future Work

Thus, as seen from the angles below, the project was reasonably good in meeting up its main objective, though there still remained some areas that needed some improvement and additions. Here are the things we've identified:Here are the things we've identified:

6.2.1 Enhanced Sensor Integration:

More Sensors: Now, let's add some more features to our system, such as temperature, humidity, pressures, and similar sensors. This will enhance the general monitoring system for SEBI thereby making better reception of data and handling of crashed exchange relations.

Sensor Fusion: We should also incorporate some complicated technique which can integrate data from numerous sensors. In this case, we get more accurate studies, and this reduces the chances of getting wrong studies thereby making the studies more reliable.

6.2.2 Connectivity and Communication:

Wireless Communication: It would be awesome if we could integrate some advanced wireless communication modules, like LoRa or Zigbee. This would allow us to monitor and control the system remotely.

IoT Integration: It is necessary that we integrate our system to an IoT platform. This shall give actual time logging, remote control, and superior usability of procedures.

6.2.3 User Interface Improvements:

Mobile and Web Applications: To put it in simpler terms, let's make some nice mobile and web applications for specific niches. They can use the data from the office, home, while traveling and other related circumstances do not influence their interaction with the system.

Customize UI: The ability of users to say that everything in the environment should be arranged as they wish is also important. That will improve their experience and make (him|them) able to choose the best option easily and quickly.

6.2.4 Power Management:

Energy Efficiency: It is also important to simulate the power consumption of the system and determine how the consumption can be reduced or optimized. This effort will prolong the battery duration and guarantee that it will function optimally in the confines of areas that are far or in lifting of places that are unforgiving.

Alternative Power Sources: They also need to consider which other sources of power could work; for instance, solar power can be effective. This will help the system to be more sustainable and will cause less need for using existing power sources

6.2.5 Robustness and Durability:

Environmental Testing: That is why we should try putting the system through some drastic tests under certain stress inducement conditions. Consider high temperatures, humidity and vibration When it comes to the environments where the robots will be operating. This will in return assure it is capable of taking on anything that comes with it

Material Improvements: t is proposed to apply some of the actively used modern materials for the construction of the mentioned enclosure. This will make the system to be hard coded and thus be resistant to technical derogation through physical contact

6.2.6 Advanced Data Processing:

Machine Learning: We should try to incorporate some of the flexible control strategies into the system for analysis data from the sensors. This way it will make possible to carry out predictive maintenance and detection of unusual patterns

Real-Time Analytics: It is amazing to think if the system is capable to do data analytics in real time process. This will help us get more specific and gain better insights on the problem and possible solutions to it.

6.2.7 Regulatory Compliance and Certification:

Standards Adherence: Safety and Standards Compliance: It is important to ensure that the system complies with all proper standards as required under the military or industrial standards for electronics and communication equipment.

Certification: We also need to obtain licenses to demonstrate the system credibility to users stating that it is safe and efficient for intelligent applications. In other words, if these mentioned points are tackled, then the digital driver panel could be improved and it should be ready to address the varying demands concerning tank monitoring systems. This will make the panel more accurate, utilitarian, and pleasurable to use for the target users. As we go on in the healthy process of developing the sophisticated type of technologies and incorporating them in more complex ways in the future, so will we be leaning towards more sophisticated and sturdier type of solutions than before.

Bibliography

[1] “Literature review and background 2019 by emma bassier,”

[2] “Electronics tutorials / hall effect sensor https://www.electronics-tutorials.ws/electromagnetism/hall-effect.html,”

[3] “Elecronics projects focus / mpu6050 https://www.elprocus.com/mpu6050-pin-diagram-circuit-and-applications/: :text=mpu6050

[4] “Wikipedia / esp32 https://en.wikipedia.org/wiki/esp32,”

[5] “Nextion https://nextion.tech/,”

[6] “Arduino integrated development environment (ide) v1 https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics/,”

[7] “Nextion editor guide https://nextion.tech/editorguide/,”

[8] “Wikipedia solidworks https://en.wikipedia.org/wiki/solidworks: :text=the

[9] “The engineering project / introduction to proteus https://www.theengineeringprojects.com/2020/01/introduction-to-proteus.html,”