Department of Electronic & Telecommunication Engineering
University of Moratuwa, Sri Lanka.

Design Methodology:
RFID based asset tracking device for forklift

210610H Sirimanna N.T.W.

Submitted in partial fulfillment of the requirements for the module
EN 2160 Engineering Design Realization

06/07/2024

Overview

In this report, I have documented the methodology that was followed by the me in this project, step by step. This includes details of the early proposal, the review progress, early planning progress, stakeholder analysis, and design revisions.

Contents

1 Objective, Problem Identification (10/02/2024)

1.1 Objective

The objective of this project is to develop an RFID-based asset tracking system specifically designed for forklifts, aiming to enhance asset management efficiency and streamline logistics processes in warehouses. The system will utilize RFID technology, integrating RFID cards with assets, door-mounted RFID card tracking systems, and an online platform for real-time asset movement visibility. By implementing engineering principles related to System Integration, Power Management, and Embedded Systems Design, the system aims to solve the industrial problem of inefficient asset tracking, achieve the learning outcomes of the EDR course, and operate within a specified budget.

1.1.1 User Interfaces

The user interfaces for the RFID-based asset tracking system will be designed to be intuitive and user-friendly, catering to both warehouse staff and management. The interfaces will include a web-based platform accessible from desktops and mobile devices, providing real-time asset tracking information and allowing users to manage assets efficiently. Additionally, RFID card readers mounted on forklifts will have simple interfaces to enable easy scanning of RFID cards attached to assets, ensuring seamless integration into daily warehouse operations.

1.1.2 Data Logging

The RFID-based asset tracking system will incorporate robust data logging capabilities to ensure accurate and reliable record-keeping of asset movements. Each RFID card scan event will be logged with a timestamp and asset ID information, which will be stored securely in a database. The system will also support the logging of maintenance records, including service dates and performed actions, to enable proactive maintenance scheduling and asset lifecycle management. Data will be logged both locally on the forklift-mounted RFID reader and on the central server, ensuring redundancy and data integrity in case of network disruptions.

1.2 Procedure in a Situation

The Procedure in a Situation section outlines the detailed steps and actions required to complete a specific task or scenario. It serves as a guide for individuals involved, ensuring consistency and efficiency in task execution.

1. **Asset Tagging:** Each asset in the warehouse is tagged with a unique RFID card containing relevant information such as asset ID, description, and maintenance history.
2. **RFID Reader Installation:** RFID card readers are installed at key locations in the warehouse, including on forklifts and at entry/exit points, to enable asset tracking.
3. **Asset Movement Detection:** When a forklift carrying tagged assets passes by a reader, the RFID reader detects the RFID cards and records the asset movement event.
4. **Data Transmission:** The RFID reader transmits the asset movement data to a central server in real-time using a wireless communication protocol (e.g., Wi-Fi or Bluetooth).
5. **Centralized Database:** The central server logs the asset movement data, including timestamps, asset IDs, and locations, in a centralized database for future reference.
6. **User Interface:** Warehouse staff and management can access a web-based platform from desktop or mobile devices to view real-time asset movement information, track asset locations, and manage asset maintenance schedules.
7. **Alerts and Notifications:** The system can send alerts and notifications to relevant stakeholders in case of unauthorized asset movements, maintenance due dates, or other predefined events.

8. **Data Analysis and Reporting:** The system provides tools for analyzing asset movement patterns, identifying inefficiencies, and generating reports to optimize warehouse operations and asset management strategies.
9. **Maintenance Integration:** The system integrates with existing maintenance management systems to schedule preventive maintenance based on asset usage and condition data.
10. **Scalability and Flexibility:** The system is designed to be scalable and flexible, allowing for easy expansion to accommodate growing warehouse needs and future technological advancements.

2 Review Progress (12/02/2024)

2.1 Present Progress of RFID-based Warehouse Management Systems

Companies like "IBCS Poland" specialize in manufacturing RFID gates designed for specific purposes, including those suitable for people, baskets, and pallets or other large items. These gates are tailored to manage packaging and resources that are sent to contractors, providing efficient solutions where necessary. In contrast, companies such as "Zebra" focus on creating a range of RFID devices, including fixed and handheld readers, antennas, and infrastructure. These devices have found success in various industries, including retail, logistics, construction, fashion, and healthcare. Furthermore, software companies like "Cybra" produce RFID tracking software, such as the Edgefinity IoT platform, which allows for powerful tracking solutions. For real-time and fixed tracking applications, Edgefinity IoT offers capabilities to track assets and equipment, locate inventory, and monitor employees effortlessly.

In warehouses, RFID technology is extensively employed for multiple processes, including monitoring truck and handling equipment movements, managing stock levels, and maintaining proper storage conditions. This technology significantly improves safety and security by minimizing errors, delays, and theft risks.

Despite the many benefits of RFID technology, there are challenges to consider. Initial implementation costs can be high, and ongoing maintenance and updates are crucial for ensuring their effectiveness. Moreover, compatibility issues with different systems and tags, particularly for international companies, may arise.

2.2 Companies With Their Products

Company	Devices	Images
IBCS Poland	• **ibcsGate:** RFID gates are based on a dedicated design and come in three sizes: for people, baskets, and pallets (or other large goods).	
	• **ibcsTracker:** The system is applicable wherever it is necessary to manage packaging or other resources that are sent to contractors.	

	• **RFID Label Printer:** It is a solution that combines two functions – print automation and labelling of items. We offer RFID applicators.	
Impinj	• **Impinj R700 series readers :** Industry-leading performance for enterprisegrade IoT solution development	
	• **Impinj Speedway readers :** High-performance for flexible solution development	
Zebra	• Fixed RFID Readers and Infrastructure (Fixed Readers)	
	• Handheld RFID Readers and Scanners	
	• RFID Antennas	
Datalogics	• UHF RFID reader	
Honeywell	• IF2B Fixed RFID Reader	
	• IH40 RFID Handheld Reader	

Alien Technology	• Enterprise RFID Reader and Edge Service Device	
	• RFID Reader	
JADAK	• ThingMagic® M7e UHF RAIN® RFID Module Series	
	• ThingMagic® IZAR 4-Port UHF / RAIN Fixed Mount RFID Reader	

Table 1: Company Devices

2.3 Stakeholder Analysis

A stakeholder map is a visual representation of the various parties involved in a project, arranged according to their degree of influence and interest. Stakeholders are positioned strategically according to their degree of engagement and influence over project outcomes using a quadrant-based approach.

Through the provision of advance notice of stakeholder needs, thorough engagement with important stakeholders, and regular communication channels with the most important ones, this classification facilitates effective stakeholder management. Project stakeholders may be effectively managed to improve project outcomes by matching stakeholder engagement tactics to the corresponding degrees of interest and influence of the stakeholders.

2.4 Stakeholder Map

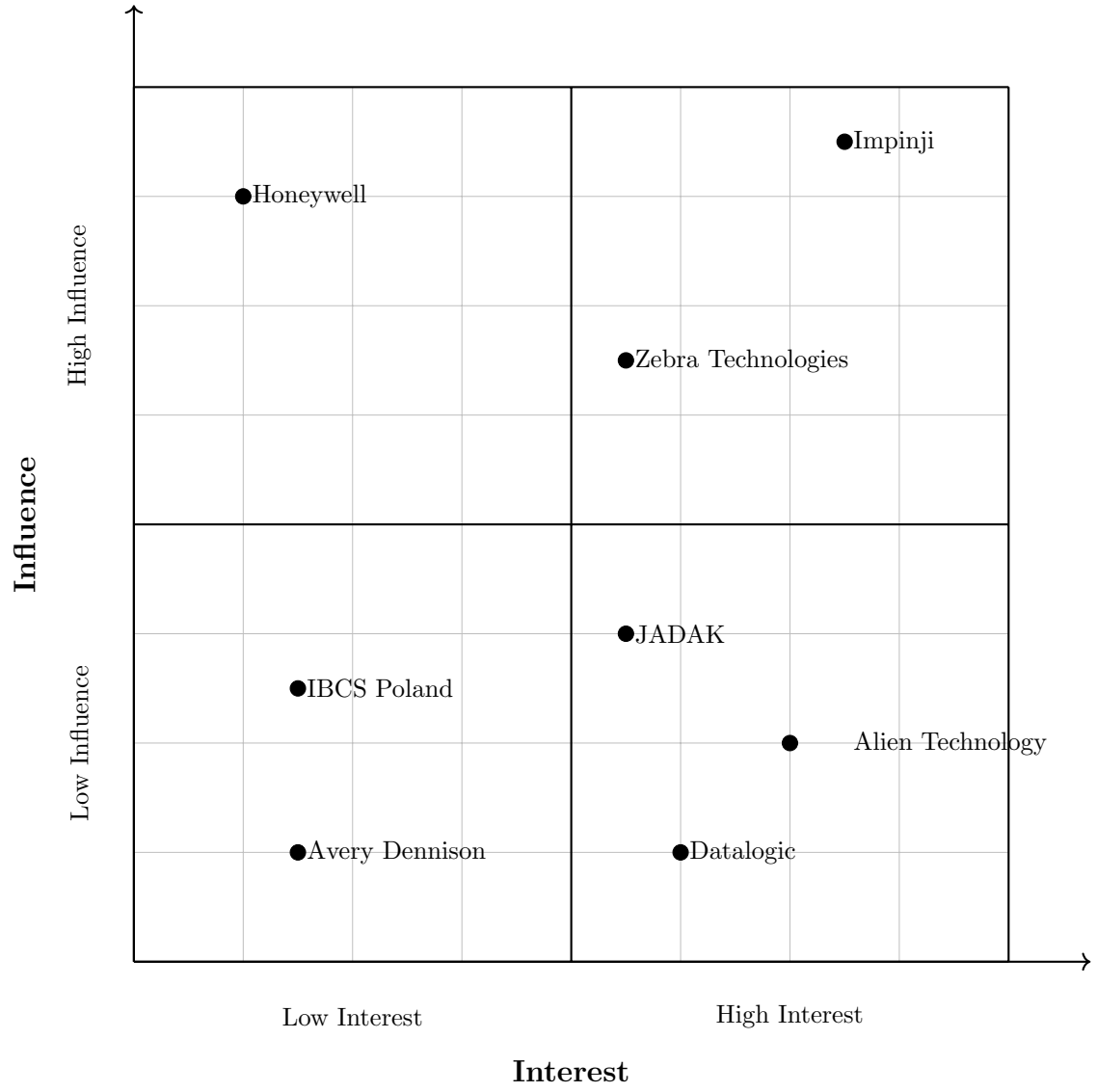


Figure 1: Mendelow Matrix for the stakeholder analysis

2.5 Main Stakeholder Heat Map

RESPONSIBLE PARTY NAME	ANTICIPATED INVOLVEMENT What level of involvement is expected?	ANTICIPATED ISSUES Known or potential issues	MOTIVATION / DRIVERS Why is the stakeholder invested in the project's success?	EXPECTATIONS OF EXCHANGE What is the stakeholder's predicted input?	MILESTONES At what point in the change effort is this stakeholder's involvement required?	ACTIVITIES What activities directly involve or impact the stakeholder?
Zebra Technologies	**HIGH**	**AVERAGE**	**ABOVE AVERAGE**	**ABOVE AVERAGE**	**HIGH**	**HIGH**
	Notes:	Notes:	Notes: Opportunity to showcase RFID technology, potential for long-term partnership.	Notes: Technical expertise, product recommendations, possibly customized solutions.	Notes: Critical at various design and implementation phases.	Notes: Providing RFID hardware, software, support services.
IBCS Poland	**AVERAGE**	**AVERAGE**	**ABOVE AVERAGE**	**AVERAGE**	**AVERAGE**	**AVERAGE**
	Notes:	Notes: Potential challenges in collaboration and communication	Notes: Opportunity to showcase expertise, secure future business opportunities	Notes: Contributions related to expertise but not critical components	Notes: Required at various milestones related to expertise	Notes: Providing expertise, recommendations, solutions related to their field
Datalogic	**AVERAGE**	**AVERAGE**	**ABOVE AVERAGE**	**ABOVE AVERAGE**	**AVERAGE**	**AVERAGE**
	Notes:	Notes:	Notes: Opportunity to demonstrate RFID solutions, potential for business growth.	Notes:	Notes: Involvement may be required at various stages, particularly related to their expertise.	Notes: Providing RFID solutions, support, and expertise.
Impinj	**HIGH**	**AVERAGE**	**ABOVE AVERAGE**	**ABOVE AVERAGE**	**HIGH**	**HIGH**
	Notes:	Notes:	Notes: Opportunity to showcase RFID technology, potential for long-term partnership.	Notes: Technical expertise, product recommendations, possibly customized solutions.	Notes: Critical at various design and implementation phases.	Notes: Providing RFID hardware, software, support services.

HEAT MAP LEVELS					
VERY LOW	LOW	AVERAGE	ABOVE AVERAGE	HIGH	CRITICAL

Figure 2: A rough sketch of a Machine-side console and a Management console

2.6 User Observation

This section will discuss possible use cases and user scenarios for a RFID based warehouse management system.

- **Warehouse Managers:**
  - **Impressions:** Current systems are often manual, leading to errors in inventory tracking and inefficient asset management.
  - **Expectations:** Desire for a more automated system that can provide real-time tracking and accurate inventory data.
- **Logistics Coordinators:**
  - **Impressions:** Existing some systems lack integration and are unable to provide comprehensive visibility into asset movement.
  - **Expectations:** : Expect a system that can streamline logistics processes, improve efficiency, and reduce errors.
- **Maintenance Personnel:**
  - **Impressions:** Limited visibility into asset status and maintenance history, leading to inefficiencies in maintenance operations.
  - **Expectations:** Seek a system that can provide detailed asset information and maintenance schedules to optimize maintenance activities.
- **Procurement Teams:**
  - **Impressions:** Difficulty in tracking asset procurement and utilization, resulting in challenges in asset allocation and budget planning.
  - **Expectations:** Look for a system that can provide insights into asset utilization and facilitate informed procurement decisions.
- **Overall Impressions:**
  - Users across different roles express a need for a more integrated, automated, and user-friendly asset tracking system.
  - Common expectations include real-time tracking, accurate inventory data, comprehensive visibility, and improved efficiency in asset management and logistics operations.

2.7 Conclusion

In conclusion, the Wireless Re-configurable Andon System with Maintenance Prediction offers a promising solution for enhancing operational efficiency across industries. Through thorough review and analysis, I have gained insights into current methodologies and innovations. Leveraging wireless technology and reconfigurable features, our system provides versatility for users in manufacturing, facility management, and logistics.

Integration of maintenance prediction enables proactive scheduling, minimizing downtime. Strategic stakeholder engagement is crucial for project success. Overall, our system holds potential to revolutionize operations in the modern industrial landscape.

3 Needs List (19/02/2024)

The needs list is a concise compilation of essential requirements and desires that the design solution must fulfill to meet the needs of stakeholders. It serves as a guiding document throughout the design process, ensuring that the final product addresses the diverse needs of users and other stakeholders. The needs list is derived from various research activities conducted during the Explore phase, such as stakeholder mapping, user observation, and persona creation, and evolves as new insights are gained.

1. **Accurate Asset Tracking:**
   - The device should accurately track the location and movement of assets within a warehouse environment.
   - It should provide real-time updates to ensure the current location of assets is always known.
2. **Compatibility and Integration:**
   - The device should be compatible with existing RFID systems and infrastructure.
   - It should integrate seamlessly with other warehouse management systems for data sharing and analysis.
3. **Durability and Reliability:**
   - The device should be durable enough to withstand the rigors of a warehouse environment, including temperature variations and physical impacts.
   - It should be reliable, with minimal downtime for maintenance or repairs.
4. **Battery Life and Power Management:**
   - The device should have a long battery life to minimize the need for frequent recharging or replacement.
   - It should have efficient power management features to optimize battery usage.
5. **Data Security and Privacy:**
   - The device should ensure the security and privacy of asset tracking data, complying with relevant regulations and standards.
   - It should provide secure access controls to prevent unauthorized access to sensitive information.
6. **User-Friendly Interface:**
   - The device should have a user-friendly interface for easy operation and configuration.
   - It should provide clear and intuitive feedback to users, such as LED indicators for status updates.
7. **Scalability and Flexibility:**
   - The device should be scalable to accommodate varying numbers of assets and locations.
   - It should be flexible enough to adapt to changing business requirements and technological advancements.
8. **Cost-Effectiveness:**
   - The device should be cost-effective, providing a high return on investment for asset tracking and management.
   - It should offer competitive pricing compared to other RFID tracking solutions.
9. **Environmental Impact:**

- The device should have minimal environmental impact, using sustainable materials and energy-efficient components.
- It should comply with relevant environmental regulations and standards.

10. **Support and Maintenance:**
    - The device should be supported by a reliable customer service team for troubleshooting and maintenance.
    - It should have easily accessible documentation and resources for users to resolve common issues.

4 Stimulate Ideas (21/02/2024)

This section is dedicated to encouraging the generation of new and innovative concepts relevant to the RFID-based asset tracking system project. It serves as a platform for brainstorming sessions, research, and discussions aimed at inspiring creative solutions to project challenges or goals. This section aims to foster a collaborative and innovative environment, driving the project team to explore unconventional approaches and think outside the box.

1. **Enhanced Asset Visibility:** Exploration of advanced RFID tags with long-range capabilities to improve asset tracking even in large warehouses or outdoor environments.
2. **Integration with IoT Devices:** Investigation of integrating RFID technology with Internet of Things (IoT) devices to enable real-time monitoring of asset movement and condition.
3. **Automated Inventory Management:** Implementation of automated inventory management features using RFID technology, such as automatic stock replenishment notifications or inventory discrepancy alerts.
4. **Energy-Efficient Solutions:** Exploration of energy-efficient RFID tag designs or power management techniques to prolong the lifespan of RFID tags and reduce the need for frequent battery replacements.
5. **Enhanced Data Analytics:** Use of data analytics tools to derive valuable insights from RFID data, such as identifying patterns in asset movement or optimizing warehouse layout for improved efficiency.
6. **Security and Access Control:** Investigation of using RFID technology for enhancing security and access control measures, such as tracking access to restricted areas or securing valuable assets.
7. **Collaboration with Suppliers:** Collaboration with RFID technology suppliers to explore the latest advancements in RFID technology and potential custom solutions tailored to the project's needs.
8. **User-Friendly Interface:** Exploration of user-friendly interface designs for the RFID asset tracking system, such as mobile applications or web interfaces, to facilitate ease of use for warehouse personnel.
9. **Scalability and Future Expansion:** Ensuring that the RFID asset tracking system is designed with scalability in mind, allowing for future expansion and integration with other systems or technologies.
10. **Sustainability Considerations:** Exploration of sustainable RFID tag materials or recycling options to reduce the environmental impact of the RFID asset tracking system.

5 Conceptual Designs (06/03/2024)

5.1 User Requirements

The user requirements for the RFID-based asset tracking device have been developed to ensure that the device meets the specific needs and expectations of its users. These requirements are based on extensive research and analysis of user needs, industry standards, and technological capabilities. By clearly defining the user requirements, I aim to create a device that provides accurate and efficient asset tracking, seamless integration with existing systems, and a user-friendly experience. The following section outlines the key user requirements that will guide the design and development of the RFID-based asset tracking device.

- **Asset Tracking:** The device must accurately track the location of assets in real-time within the warehouse or during transportation.
- **RFID Compatibility:** The device should be compatible with commonly used RFID tags and readers in the industry for seamless integration with existing systems.
- **Data Management:** The device must have a user-friendly interface for managing asset data, including adding new assets, updating information, and generating reports.
- **Security:** The device should have built-in security features to prevent unauthorized access and ensure data integrity.
- **Durability:** The device should be durable and able to withstand the rigors of industrial environments.
- **Cost-Effectiveness:** The device should be cost-effective, offering a good balance between price and features.
- **Power Efficiency:** The device should be designed to operate efficiently from the forklift battery, minimizing power consumption to avoid excessive drain on the forklift's electrical system.

5.2 Conceptual Designs and Functional Block Diagrams

5.2.1 Design 1

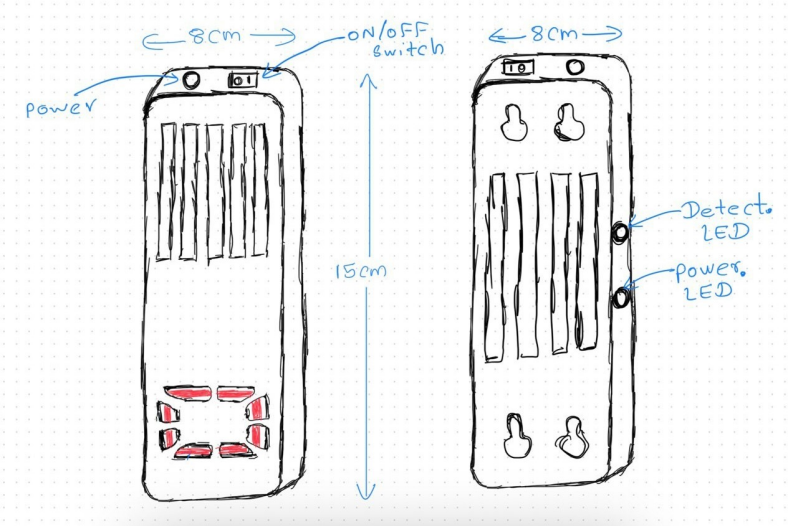


Figure 3: A rough sketch of a Machine-side console and a Management console

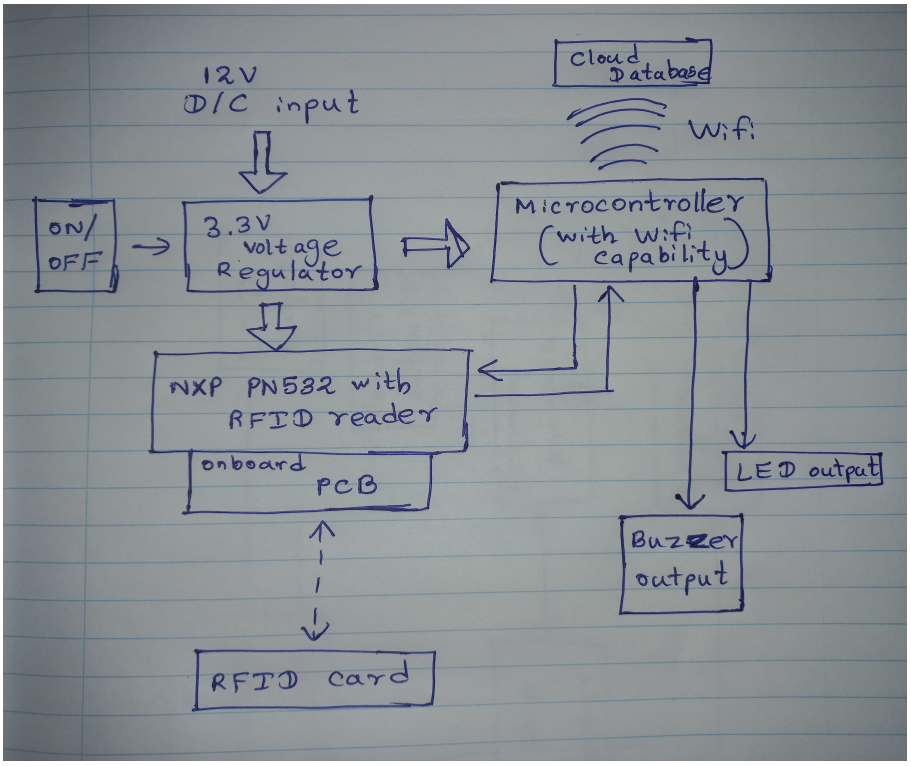


Figure 4: Functional block diagrams of the Machine side console (left) and the Management console (right)

The first design concept proposes an enclosure for the RFID-based asset management system, which is designed to be attached to the two arms of a forklift. This enclosure utilizes WiFi connectivity to enable real-time updates to a global database, ensuring accurate and up-to-date asset tracking information. The enclosure is powered by the forklift's 12V battery, eliminating the need for additional power sources.

It features power and detect LEDs for status indication, an external power supply for flexibility in power management, WiFi connectivity for seamless data transmission, and compatibility with global cloud storage for efficient data management. This design aims to enhance asset tracking capabilities in warehouse environments by providing a reliable and efficient solution for tracking assets attached to forklifts.

By leveraging WiFi connectivity and the forklift's 12V battery, this design offers a cost-effective and efficient solution for asset tracking. It ensures that asset tracking information is always up-to-date and easily accessible, enhancing overall efficiency and productivity in warehouse operations.

5.2.2 Design 2

The second design concept focuses on redundancy and reliability by incorporating two RFID antennas and one RFID reader. This configuration ensures that assets are accurately detected even if one antenna fails to detect them. The design features power and detect LEDs for status indication, an internal power supply that is charged through the forklift for uninterrupted operation, Bluetooth connectivity for data transmission, and a local database inside the forklift for storing asset tracking information.

By combining redundancy with advanced connectivity options, this design aims to provide a robust and reliable asset tracking solution for warehouse environments. The internal power supply ensures continuous operation, while Bluetooth connectivity enables seamless data transmission. The local database inside the forklift provides a secure storage solution for asset tracking information.

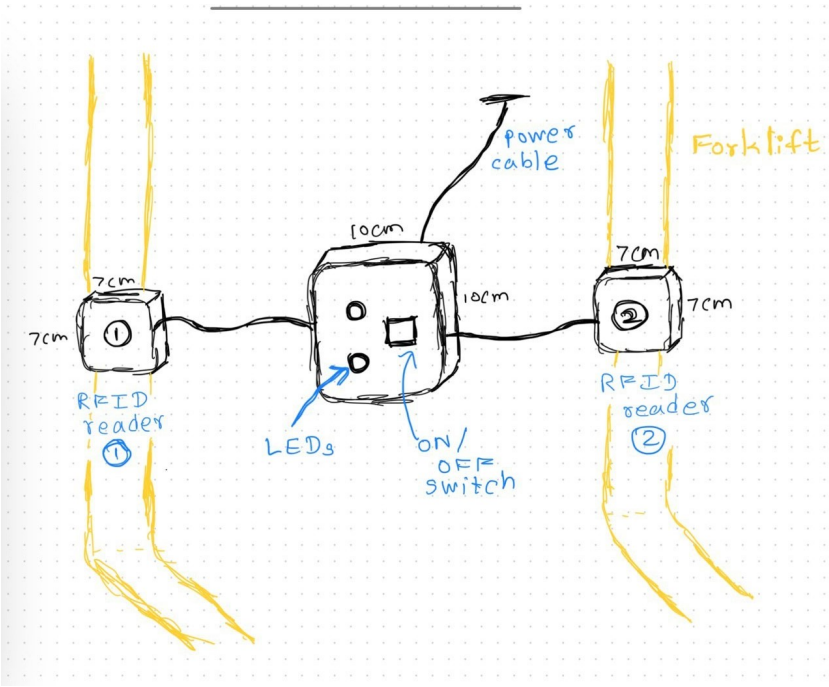


Figure 5: A rough sketch of a Machine-side console

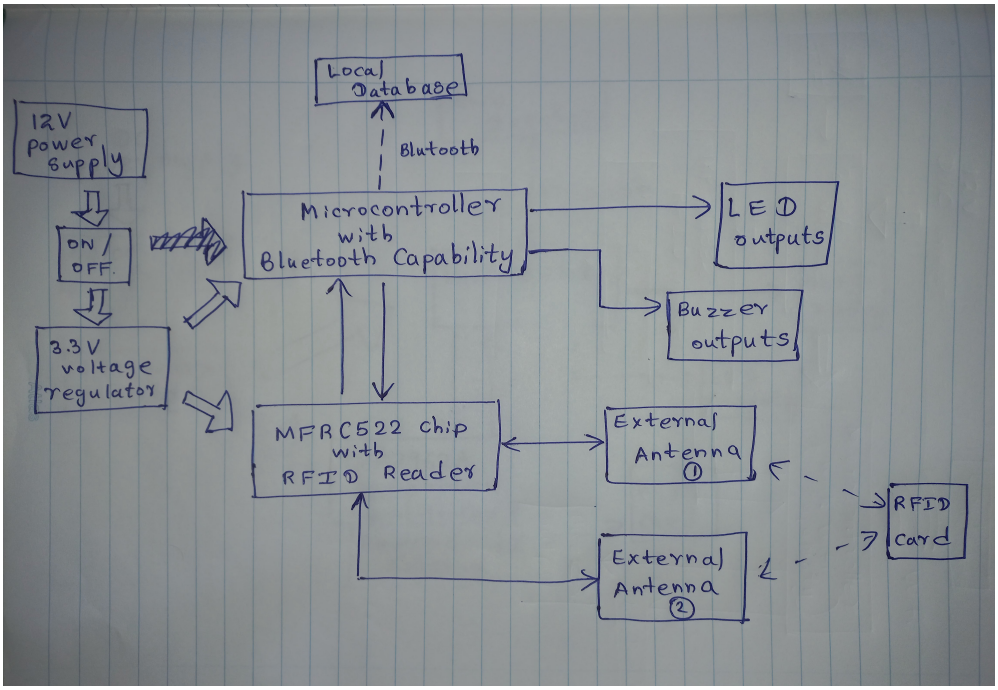


Figure 6: Functional block diagram of the Machine-side Console, and the Software Architecture

This design offers a cost-effective and reliable solution for asset tracking in warehouse environments. By incorporating redundancy and advanced connectivity options, it ensures that assets are accurately tracked and managed, leading to improved efficiency and productivity in warehouse operations.

5.2.3 Design 3

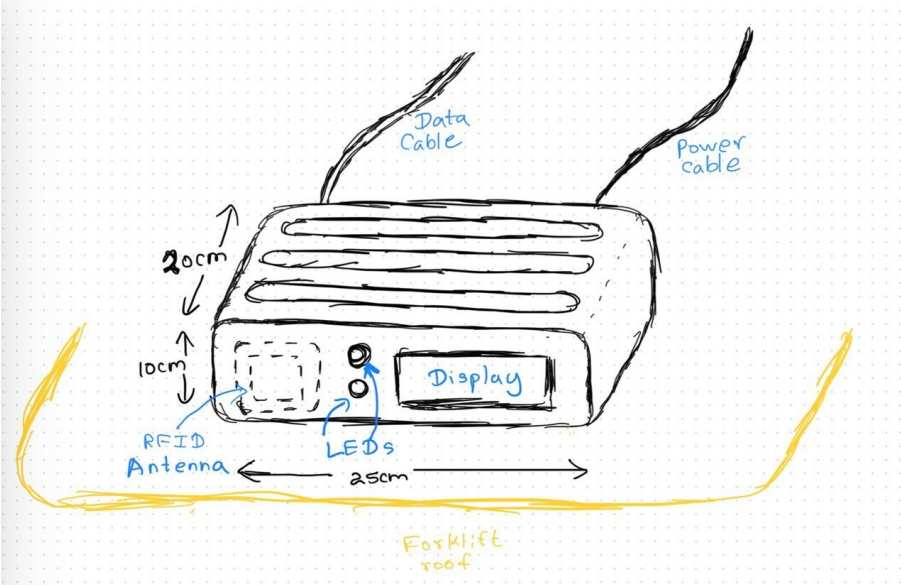


Figure 7: A rough sketch of a Machine-side console

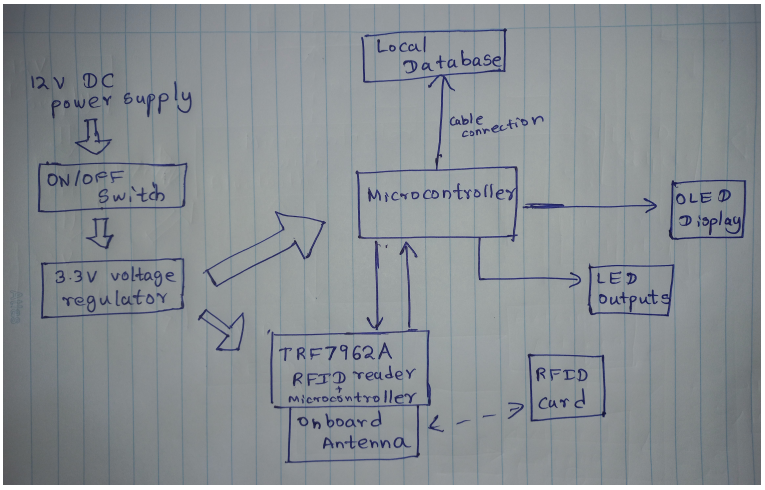


Figure 8: Functional block diagram of the Machine-side console

The third design concept introduces a novel approach to asset tracking by mounting the RFID reader on the roof of the forklift. This design requires an RFID reader with a higher distance reading capability to effectively track assets from a greater distance. The design features two displays for information display, an external power supply sourced from the forklift for continuous operation, cable connectivity for data transfer, and local data storage for storing asset tracking information.

By leveraging the height advantage of the forklift, this design aims to improve asset tracking accuracy and efficiency in warehouse environments. The external power supply ensures continuous operation, while cable connectivity enables data transfer. The local data storage provides a secure and reliable storage solution for asset tracking information.

This design offers an innovative and efficient solution for asset tracking in warehouse environments. By mounting the RFID reader on the roof of the forklift, it improves asset tracking accuracy and efficiency, leading to improved productivity and efficiency in warehouse operations.

5.3 Evaluation Criteria

It is required to choose the best suitable design out of the above three for the enclosure and the function of the product. For this, each of the three designs have been analysed and evaluated against the following criteria.

5.3.1 Enclosure Design Criteria

- **Performance** - Success rate of capturing information.
- **Compatibility** - How well does the enclosure integrate with the forklift's existing structure and components?
- **Protection** - How effectively does the enclosure protect the RFID reader from environmental factors such as dust, water, and impact?
- **Accessibility** - How easy is it to access the RFID reader for maintenance or replacement within the enclosure?
- **Durability** - How well does the enclosure withstand wear and tear, including vibrations from the forklift's operation?
- **Aesthetics** - How visually appealing is the enclosure, considering its design and finish?
- **Cost** - What is the overall cost of manufacturing and implementing the enclosure compared to its benefits?
- **Ease of Installation** - How easy is it to install the enclosure on the forklift without requiring extensive modifications?
- **Space Utilization** - How efficiently does the enclosure use space on the forklift while not obstructing other components or operations?

5.3.2 Functional block diagram criteria

- **Functionality** - How well does the block diagram illustrate the functional requirements of the RFID asset tracking device, including the interaction between different modules?
- **User Experience** - How intuitive is the design of the block diagram in terms of understanding the flow of data and operations within the device?
- **Manufacturing Feasibility** - Evaluate the feasibility of manufacturing the device based on the block diagram, considering factors such as component availability, assembly complexity, and production cost.
- **Cost** - Evaluate the overall cost-effectiveness of the design based on the components and modules depicted in the block diagram.
- **Performance** - Evaluate the performance of the device based on the block diagram, considering factors such as signal quality, and bandwidth range.
- **Future Proofing** - To what extent does the block diagram allow for easy replacement or upgrade of individual components in the device?
- **Power Efficiency** - The management of power consumption in the device?

5.4 Comparison of Designs

Evaluating the criteria given above, the designs have been given a score from 0 to 10, with 0 being the least desirable and 10 being the most desirable. The design with the highest total score will be chosen to continue the development process.

		Conceptual Design 1	Conceptual Design 2	Conceptual Design 3
Newly added features		Power LED and detect LED for information , External power supply (from forklift) , WIFI connectivity , Global Cloud Storage	Power LED and detect LED for information , Internal power supply and charge through forklift , Bluetooth Connectivity , Local data base (Locate inside the Forklift)	Two displays for information , External Power Supply (from forklift) , Cable connectivity for data transferring , Local data storage
Removed features		Bluetooth connectivity , Internal power supply	WIFI connectivity , Cloud data storage	WIFI connectivity , Bluetooth connectivity , Internal power supply
Enclosure Design Criteria Comparison	Performance	8	9	6
	Compatibility	9	7	8
	Space Utilization	9	6	9
	Protection	8	6	9
	Accessibility	8	7	8
	Durability	7	7	9
	Aesthetics	8	7	6
	Cost	8	6	7
	Ease of Installation	8	6	9
Functional Block Diagram Criteria Comparison	Functionality	8	7	8
	User Experience	8	7	7
	Manufacturing Fesibility	9	6	7
	Cost	8	6	8
	Performance	8	9	6
	Future Proofing	7	7	6
	Power	6	8	6
Total Score		127	111	119

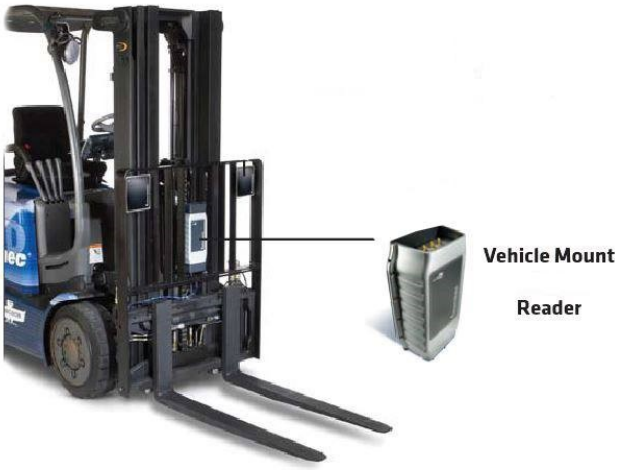
Table 2: Comparision of designs and scoring

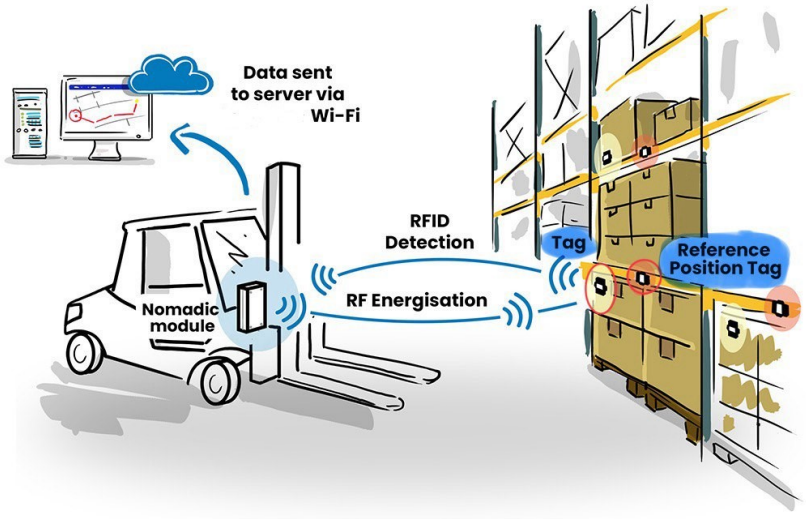
5.5 Selected Design

After careful evaluation based on the specified criteria, Design 1 has been selected as the preferred conceptual design for the RFID-based asset tracking device. Design 1 offers a comprehensive set of features, including power and detect LEDs for information, external power supply from the forklift, and WiFi connectivity for global cloud storage. This design excelled in several key criteria, including performance, compatibility, and space utilization, making it the optimal choice for meeting the project requirements. Additionally, Design 1

scored well in terms of protection, accessibility, and cost-effectiveness. Overall, Design 1 provides a robust and efficient solution for tracking assets in real-time, making it the ideal choice for implementation in the project.

5.5.1 Conceptual Final Assembly



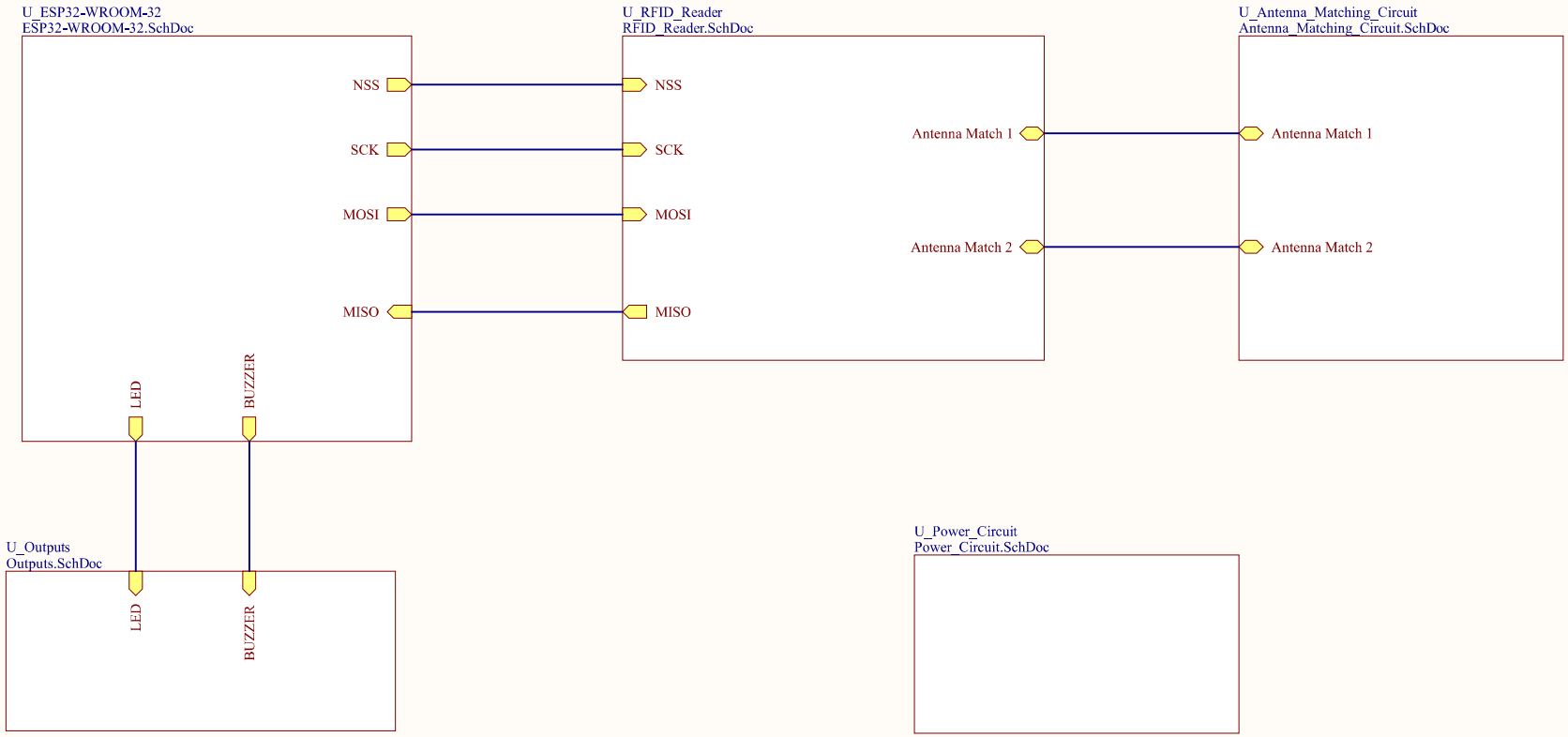


6 Schematic And PCB Design (09/03/2024)

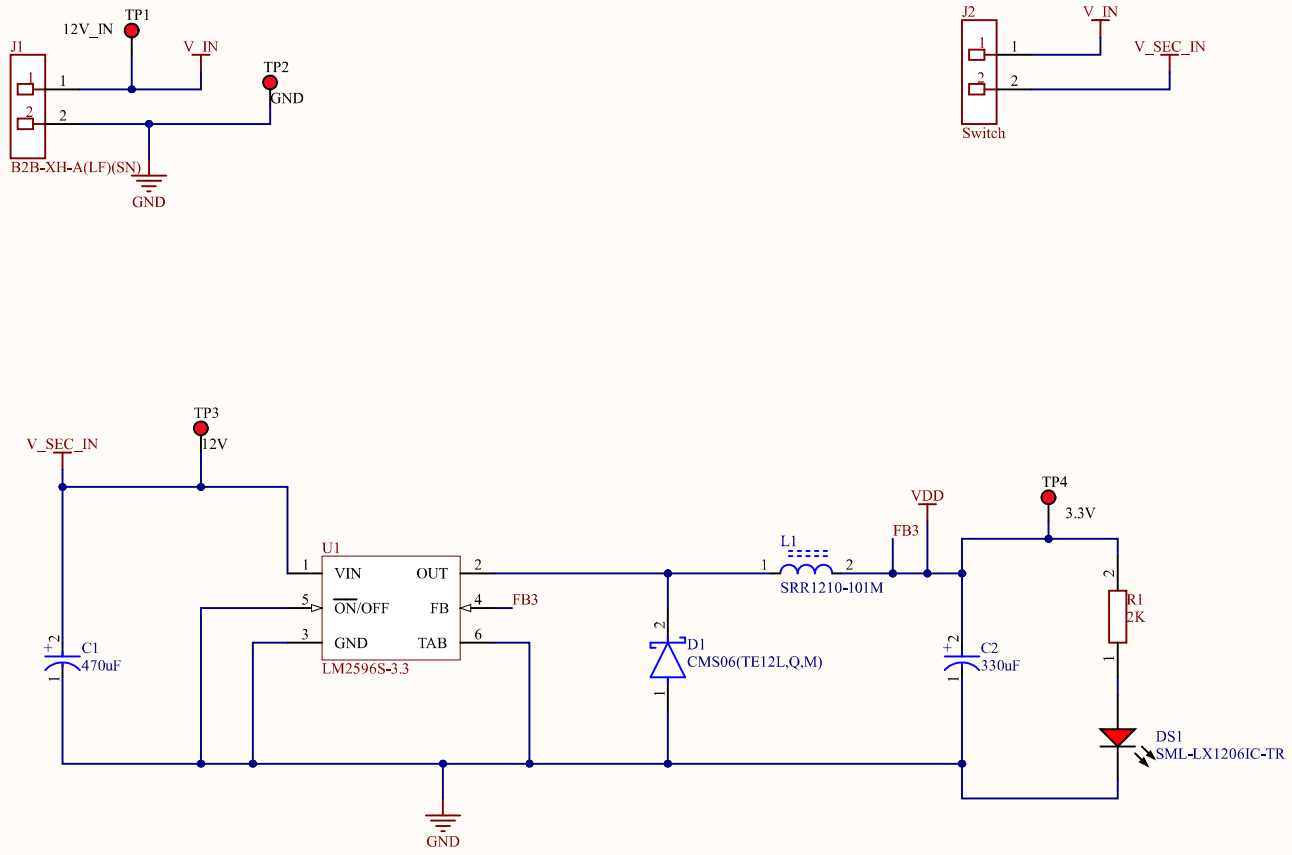
6.1 Schematic Design

This subsection features schematic images of the RFID-based asset tracking device, illustrating the electrical connections and component relationships in the circuit design. These images provide a clear overview of the circuitry and aid in understanding the functionality and complexity of the system.

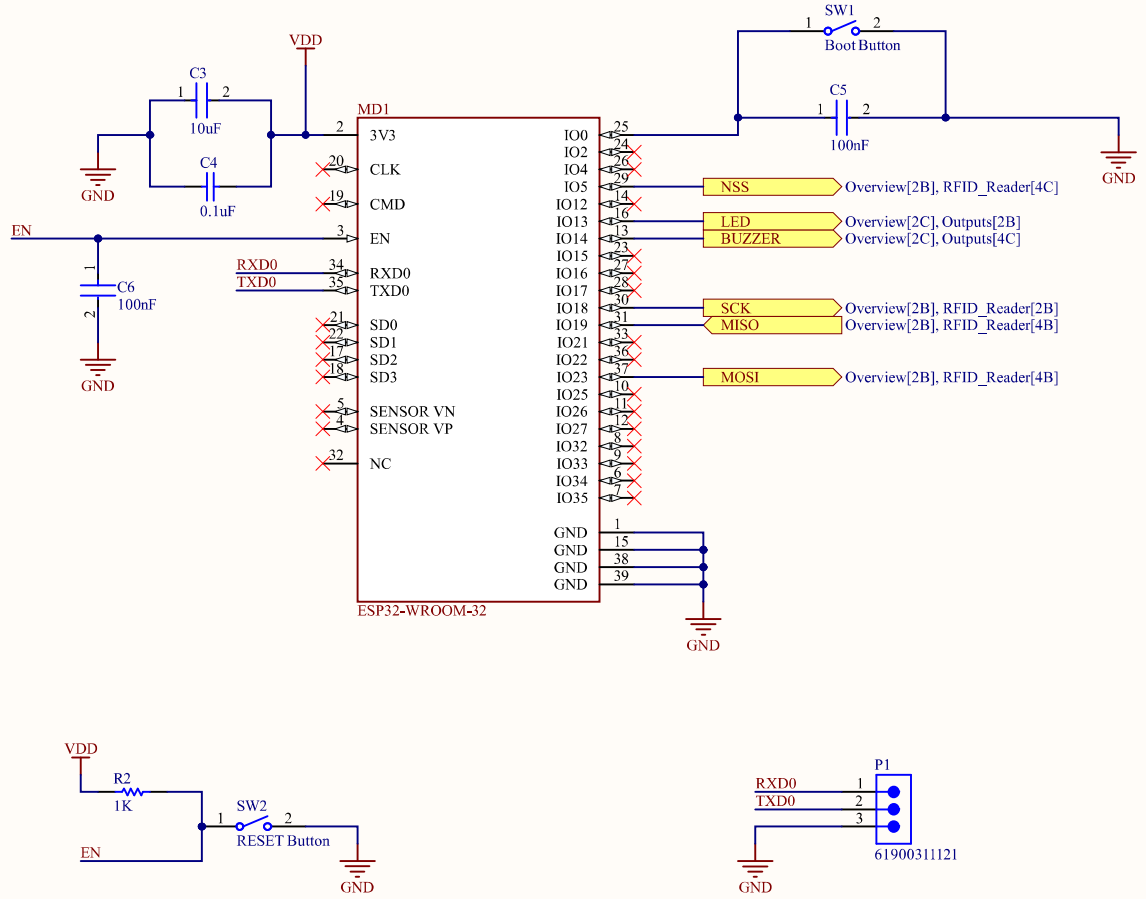
TRACKING SYSTEM PCB



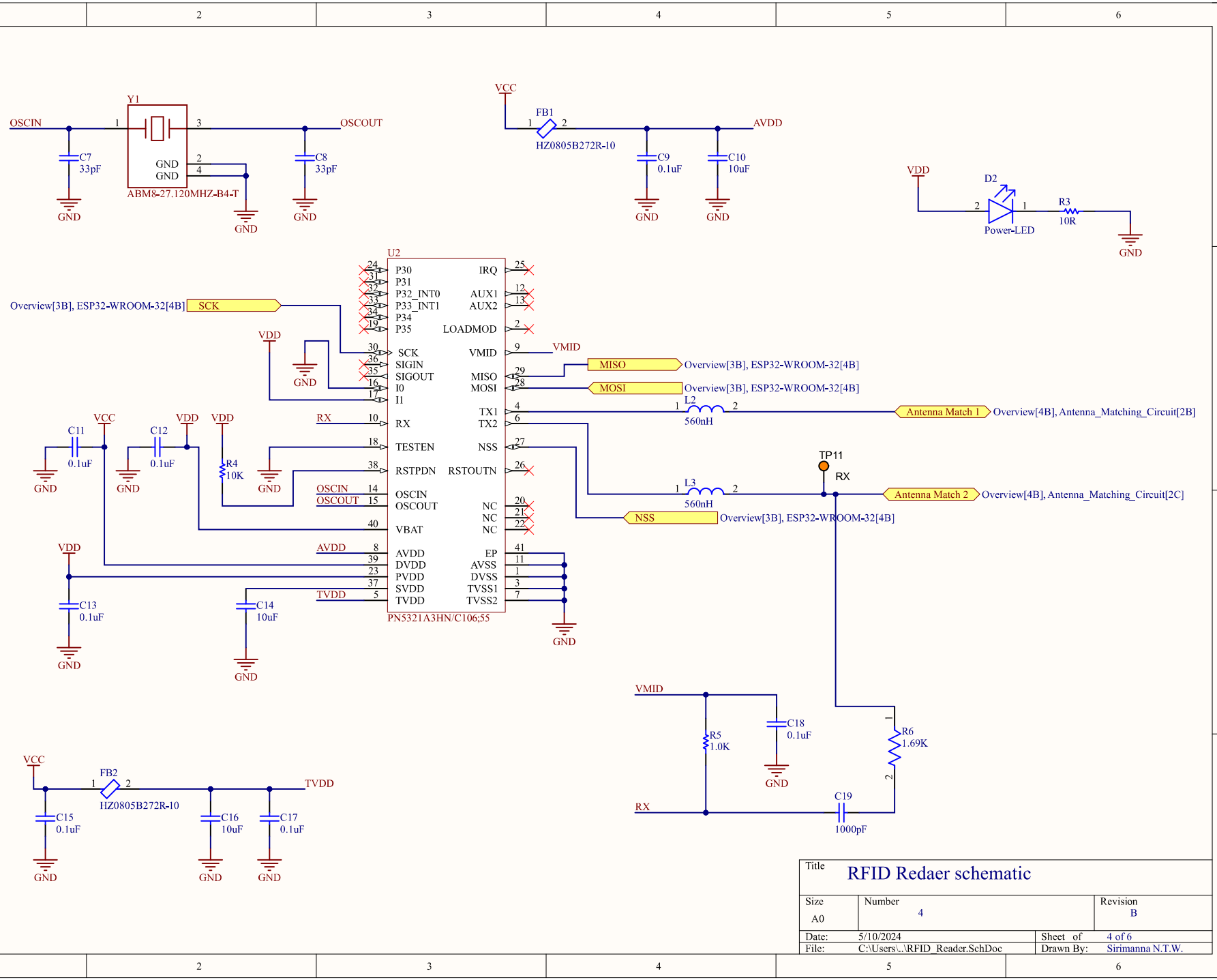
Title	Overview Schematic		
Size: A	Number: 1	Revision: B	
Date: 5/10/2024		Sheet of: 1 of 6	
File: C:\Users\...\Overview.SchDoc		Drawn By: Sirimanna N.T.W.	



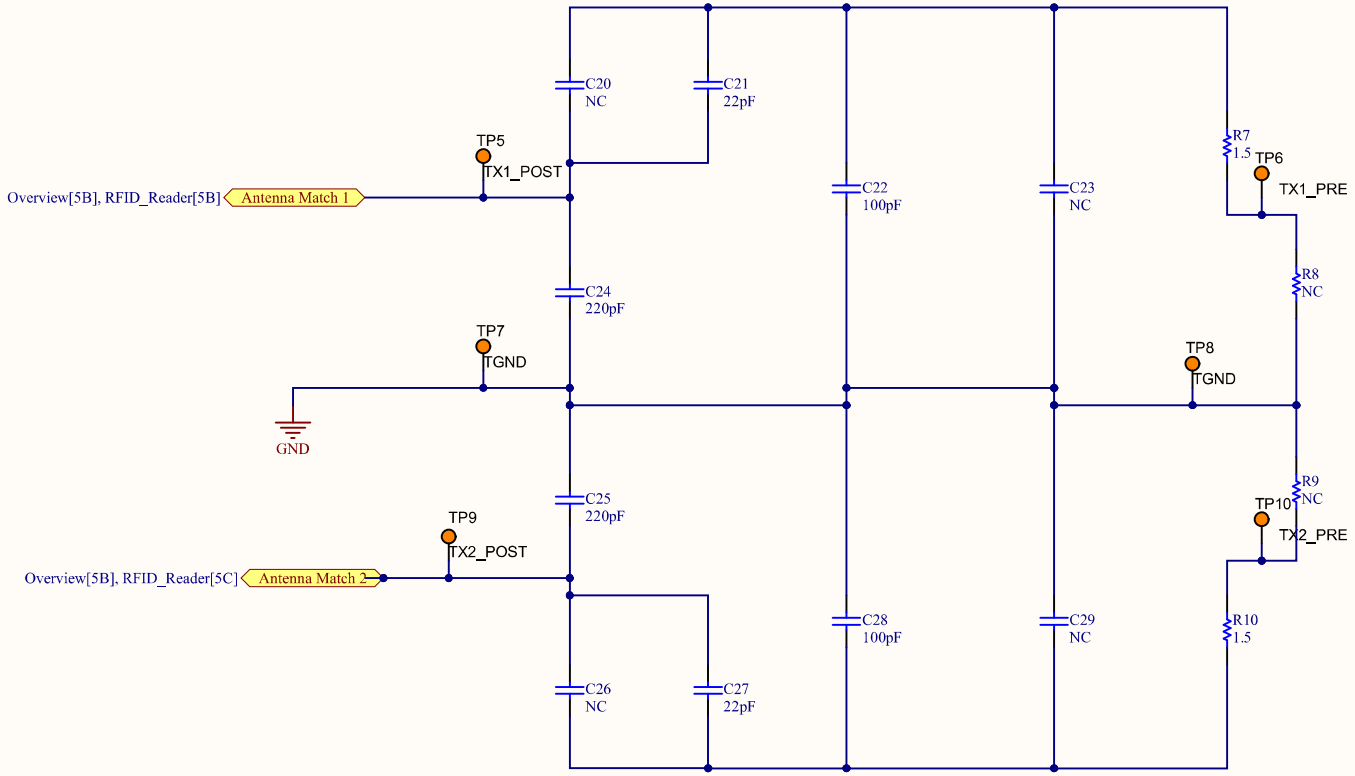
Title	Power Circuit Schematic		
Size: A	Number: 2		Revision: B
Date: 5/10/2024		Sheet of: 2 of 6	
File: C:\Users\..\Power_Circuit.SchDoc		Drawn By: Sirimanna N.T.W.	



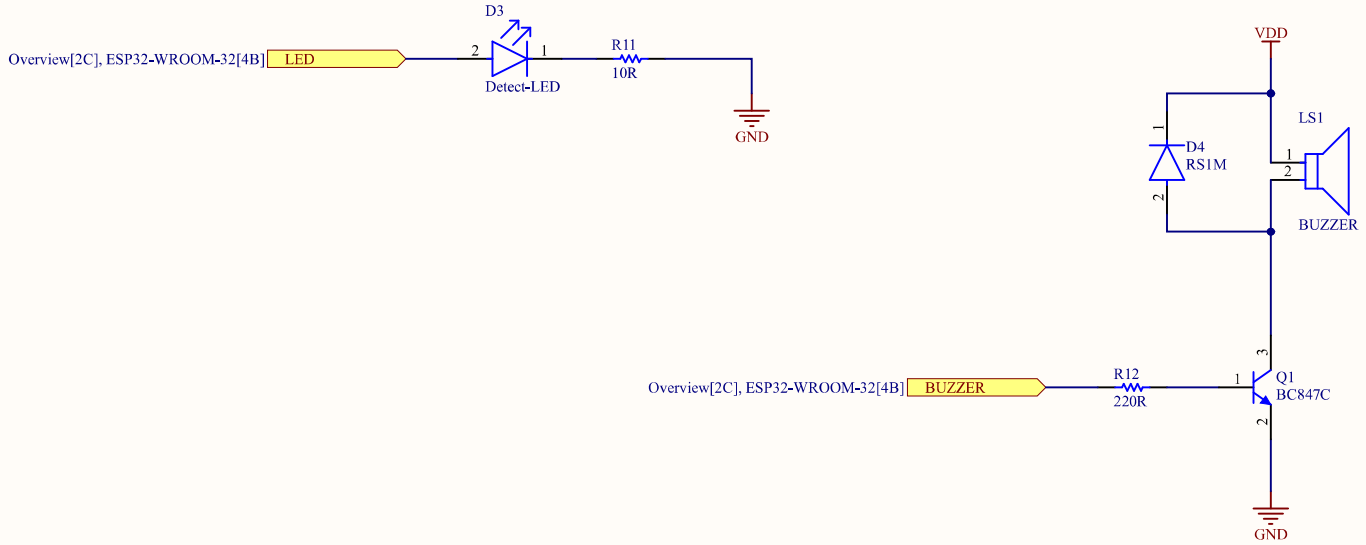
Title	ESP32 Schematic		
Size: A	Number: 3		Revision: B
Date: 5/10/2024		Sheet of: 2 of 6	
File: C:\Users\..\ESP32-WROOM-32.SchDoc		Drawn By: Sirimanna N.T.W.	



Title	RFID Redaer schematic		
Size: A0	Number: 4	Revision: B	
Date: 5/10/2024		Sheet of: 4 of 6	
File: C:\Users\...\RFID_Reader.SchDoc		Drawn By: Sirimanna N.T.W.	



Title	Antenna Matching Circuit Schematic		
Size: A	Number: 5		Revision: B
Date: 5/10/2024		Sheet of: 5 of 6	
File: C:\Users\...\Antenna_Matching_Circuit.SchDoc		Drawn By: Sirimanna N.T.W.	



Title	Buzzer and LED Schematic		
Size: A	Number: 6		Revision: B
Date: 5/10/2024		Sheet of 6 of 6	
File: C:\Users\...\Outputs.SchDoc		Drawn By: Sirimanna N.T.W.	

6.2 PCB Design

Within this subsection, detailed images of the PCB design are presented, showcasing the layout, component placement, and routing strategies employed in the development of the RFID-based asset tracking device. These images provide a visual representation of the circuit design and serve to illustrate the meticulous planning and implementation of the project.

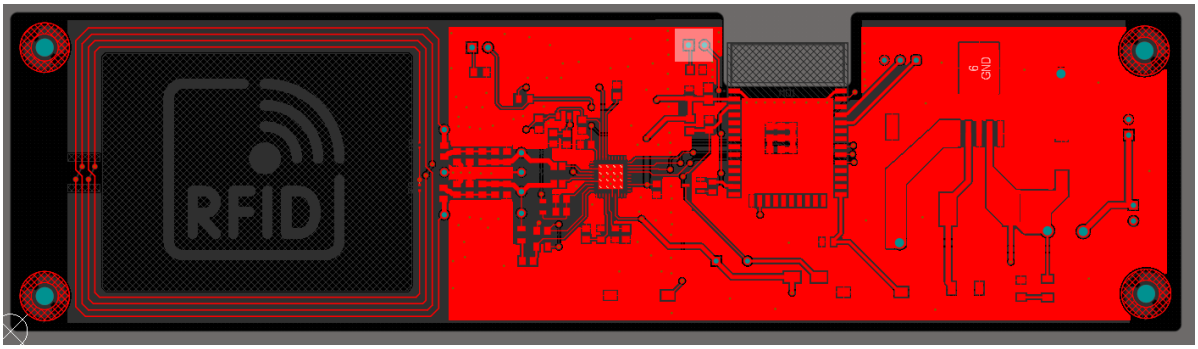


Figure 9: Top Layer

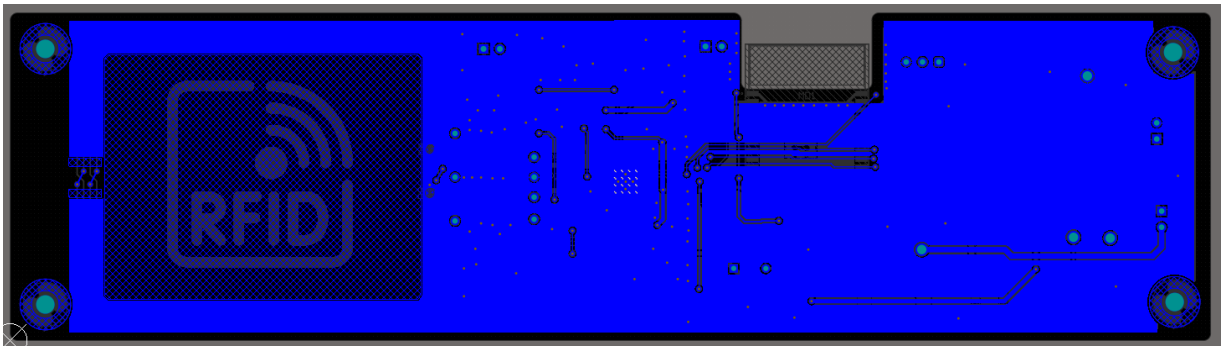


Figure 10: Bottom Layer

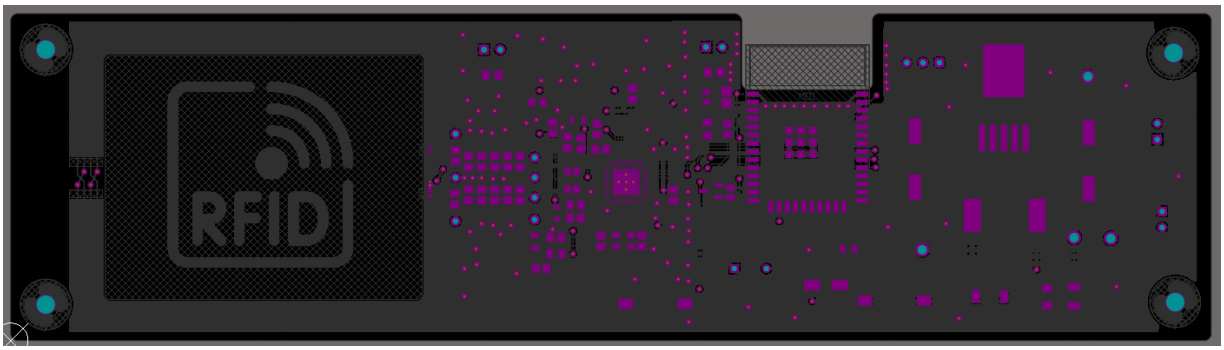
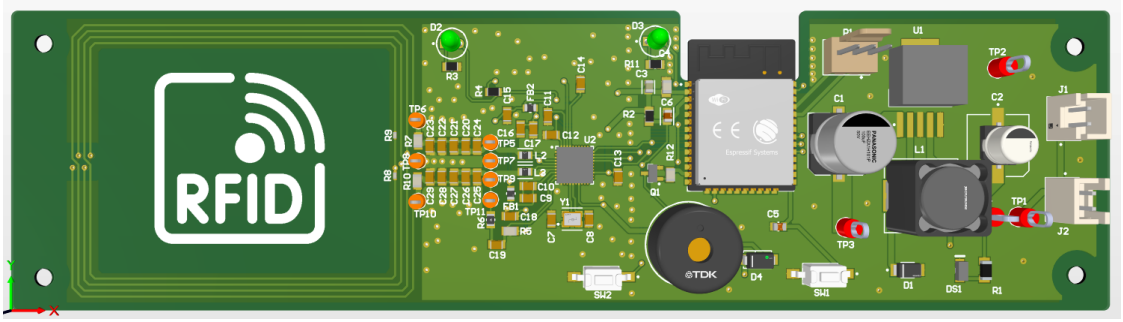


Figure 11: Soldermask



7 Firmware Implementation (12/04/2024)

Firmware implementation is a crucial aspect of embedded systems development, involving the creation and deployment of software that operates directly on hardware devices like microcontrollers or system-on-chips (SoCs). In the context of this introduction, we will focus on the implementation of firmware using the ESP-IDF framework for the ESP32-WROOM32 IC, leveraging UART communication for uploading firmware.

The ESP32-WROOM32 IC is based on the Xtensa LX6 architecture developed by Tensilica (now owned by Cadence Design Systems). The Xtensa architecture is known for its flexibility and configurability, allowing for efficient customization of the processor's instruction set and hardware features to meet specific application requirements.

ESP-IDF (Espressif IoT Development Framework) is a comprehensive platform for developing applications for ESP32 and ESP32-S series SoCs. It provides a rich set of libraries, tools, and APIs tailored for efficient and scalable firmware development.

7.1 Importance of Register Programming

In firmware development, register programming plays a vital role in configuring and controlling hardware peripherals directly at the register level. This approach offers several key advantages:

- **Performance Optimization:** Register programming allows developers to fine-tune hardware settings for optimal performance, often achieving faster data processing and reduced latency compared to higher-level APIs.
- **Resource Efficiency:** By accessing hardware registers directly, firmware can conserve system resources such as memory and processing power, crucial in embedded systems where resource constraints are common.
- **Low-Level Access:** It facilitates precise control over timing, interrupts, and low-level hardware features, essential for tasks like real-time data acquisition, signal processing, and device synchronization.

```
#include <stdio.h>
#include <string.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/event_groups.h"
#include "nvs_flash.h"
#include "esp_wifi.h"
#include "esp_event.h"
#include "mqtt_client.h"
#include "driver/spi_master.h"
```

```
#include "pn532.h"

#define PN532_SCK 5
#define PN532_MISO 19
#define PN532_MOSI 23
#define PN532_SS 18

#define BUZZER_PIN 14
#define INDICATOR_PIN 13

#define WIFI_SSID "NTWS-4G"
#define WIFI_PASS "XXXXXXXXXXXX"
#define MQTT_BROKER "mqtt://test.mosquitto.org"

static const char *TAG = "RFID_TRACKER";

static EventGroupHandle_t wifi_event_group;
const int WIFI_CONNECTED_BIT = BIT0;

esp_mqtt_client_handle_t mqtt_client;
pn532_t pn532;

char details[50];
char details_long[200];
char idcard[50];
char idcard_old[50];

// Function prototypes
static void wifi_event_handler(void *arg, esp_event_base_t event_base,
                               int32_t event_id, void *event_data);
static void wifi_init(void);
static void mqtt_event_handler(void *handler_args, esp_event_base_t base,
                               int32_t event_id, void *event_data);
static void mqtt_app_start(void);
static void setup_pn532(void);
static void read_pn532(void);
static void indicate(void);

// Macros for direct register access
#define REG_WRITE(addr, val) (*((volatile uint32_t *)(addr)) = (val))
#define REG_SET_BIT(addr, bit) REG_WRITE((addr), (REG_READ(addr) | (1ULL << (bit))))
#define REG_CLEAR_BIT(addr, bit) REG_WRITE((addr), (REG_READ(addr) & ~(1ULL << (bit))))
#define REG_READ(addr) (*((volatile uint32_t *)(addr)))

// GPIO register addresses and bit masks for ESP32
#define GPIO_OUT_REG(pin)   (GPIO_REG_BASE + 0x0C + (4 * ((pin) % 32)))
#define GPIO_ENABLE_W1TC_REG(pin) (GPIO_REG_BASE + 0x88 + (4 * ((pin) / 32)))
#define GPIO_ENABLE_W1TS_REG(pin) (GPIO_REG_BASE + 0x78 + (4 * ((pin) / 32)))

// Function to set GPIO pin level
```

```
61 static inline void gpio_set_level(uint32_t pin, uint32_t level) {
62     if (level)
63         REG_SET_BIT(GPIO_OUT_REG(pin), pin % 32);
64     else
65         REG_CLEAR_BIT(GPIO_OUT_REG(pin), pin % 32);
66 }
67
68 // Function to set GPIO pin direction
69 static inline void gpio_set_direction(uint32_t pin, uint32_t mode) {
70     if (mode == GPIO_MODE_OUTPUT) {
71         REG_SET_BIT(GPIO_ENABLE_W1TS_REG(pin), pin % 32);
72     } else {
73         REG_CLEAR_BIT(GPIO_ENABLE_W1TC_REG(pin), pin % 32);
74     }
75 }
76
77 // WiFi event handler
78 static void wifi_event_handler(void *arg, esp_event_base_t event_base,
79                                int32_t event_id, void *event_data) {
80     if (event_base == WIFI_EVENT && event_id == WIFI_EVENT_STA_START) {
81         esp_wifi_connect();
82     } else if (event_base == WIFI_EVENT && event_id ==
       WIFI_EVENT_STA_DISCONNECTED) {
83         esp_wifi_connect();
84         xEventGroupClearBits(wifi_event_group, WIFI_CONNECTED_BIT);
85     } else if (event_base == IP_EVENT && event_id == IP_EVENT_STA_GOT_IP)
       {
86         xEventGroupSetBits(wifi_event_group, WIFI_CONNECTED_BIT);
87     }
88 }
89
90 // Initialize WiFi
91 static void wifi_init(void) {
92     esp_err_t ret = nvs_flash_init();
93     if (ret == ESP_ERR_NVS_NO_FREE_PAGES || ret ==
       ESP_ERR_NVS_NEW_VERSION_FOUND) {
94         ESP_ERROR_CHECK(nvs_flash_erase());
95         ret = nvs_flash_init();
96     }
97     ESP_ERROR_CHECK(ret);
98
99     wifi_event_group = xEventGroupCreate();
100
101    ESP_ERROR_CHECK(esp_netif_init());
102    ESP_ERROR_CHECK(esp_event_loop_create_default());
103    esp_netif_create_default_wifi_sta();
104
105    wifi_init_config_t cfg = WIFI_INIT_CONFIG_DEFAULT();
106    ESP_ERROR_CHECK(esp_wifi_init(&cfg));
107
108    esp_event_handler_instance_t instance_any_id;
109    esp_event_handler_instance_t instance_got_ip;
110    ESP_ERROR_CHECK(esp_event_handler_instance_register(WIFI_EVENT,
       ESP_EVENT_ANY_ID,
```

```
111                                                         &
                                                                wifi_event_handler
                                                                , NULL, &
                                                                instance_any_id
                                                                ));
112     ESP_ERROR_CHECK(esp_event_handler_instance_register(IP_EVENT,
            IP_EVENT_STA_GOT_IP,
113                                                         &
                                                                wifi_event_handler
                                                                , NULL, &
                                                                instance_got_ip
                                                                ));
114
115     wifi_config_t wifi_config = {
116         .sta = {
117             .ssid = WIFI_SSID,
118             .password = WIFI_PASS,
119         },
120     };
121     ESP_ERROR_CHECK(esp_wifi_set_mode(WIFI_MODE_STA));
122     ESP_ERROR_CHECK(esp_wifi_set_config(ESP_IF_WIFI_STA, &wifi_config));
123     ESP_ERROR_CHECK(esp_wifi_start());
124
125     xEventGroupWaitBits(wifi_event_group, WIFI_CONNECTED_BIT, pdFALSE,
            pdTRUE, portMAX_DELAY);
126 }
127
128 // MQTT event handler
129 static void mqtt_event_handler(void *handler_args, esp_event_base_t base,
130                                int32_t event_id, void *event_data) {
131     esp_mqtt_event_handle_t event = event_data;
132     switch (event->event_id) {
133         case MQTT_EVENT_CONNECTED:
134             esp_mqtt_client_subscribe(mqtt_client, "210610H-CardDetails",
                    0);
135             break;
136         case MQTT_EVENT_DISCONNECTED:
137             break;
138         default:
139             break;
140     }
141 }
142
143 // Start MQTT client
144 static void mqtt_app_start(void) {
145     esp_mqtt_client_config_t mqtt_cfg = {
146         .uri = MQTT_BROKER,
147     };
148     mqtt_client = esp_mqtt_client_init(&mqtt_cfg);
149     esp_mqtt_client_register_event(mqtt_client, ESP_EVENT_ANY_ID,
            mqtt_event_handler, NULL);
150     esp_mqtt_client_start(mqtt_client);
151 }
152
```

```
153 // Setup PN532 module
154 static void setup_pn532(void) {
155     spi_bus_config_t buscfg = {
156         .miso_io_num = PN532_MISO,
157         .mosi_io_num = PN532_MOSI,
158         .sclk_io_num = PN532_SCK,
159         .quadwp_io_num = -1,
160         .quadhd_io_num = -1,
161         .max_transfer_sz = 0,
162     };
163     spi_device_interface_config_t devcfg = {
164         .clock_speed_hz = 1 * 1000 * 1000, // 1 MHz
165         .mode = 0,
166         .spics_io_num = PN532_SS,
167         .queue_size = 7,
168     };
169 
170     ESP_ERROR_CHECK(spi_bus_initialize(HSPI_HOST, &buscfg, 1));
171     ESP_ERROR_CHECK(spi_bus_add_device(HSPI_HOST, &devcfg, &pn532.
        spi_handle));
172 
173     pn532_init(&pn532);
174     uint32_t versiondata = pn532_get_firmware_version(&pn532);
175     if (!versiondata) {
176         ESP_LOGE(TAG, "Didn't find PN53x board");
177         while (1);
178     }
179 
180     pn532_set_passive_activation_retries(&pn532, 0xFF);
181     ESP_LOGI(TAG, "PN532 setup complete");
182 }
183 
184 // Read data from PN532
185 static void read_pn532(void) {
186     uint8_t success;
187     uint8_t uid[7]; // Buffer to store the returned UID
188     uint8_t uidLength;
189     success = pn532_read_passive_target(&pn532, PN532_MIFARE_ISO14443A,
        uid, &uidLength, 0);
190     if (success) {
191         idcard_old[0] = '\0';
192         strcpy(idcard_old, idcard);
193 
194         idcard[0] = '\0';
195         for (uint8_t i = 0; i < uidLength; i++) {
196             char hex[3];
197             sprintf(hex, "%02X", uid[i]);
198             strcat(idcard, hex);
199         }
200 
201         if (strcmp(idcard, idcard_old) != 0) {
202             snprintf(details, sizeof(details), "%s", idcard);
203             esp_mqtt_client_publish(mqtt_client, "210610H-CardDetails",
                details, 0, 1, 0);
```

```
            indicate();
        }
    }
}

// Indicate action with LEDs and buzzer
static void indicate(void) {
    gpio_set_level(BUZZER_PIN, 1);
    gpio_set_level(INDICATOR_PIN, 1);
    vTaskDelay(400 / portTICK_PERIOD_MS);
    gpio_set_level(BUZZER_PIN, 0);
    gpio_set_level(INDICATOR_PIN, 0);
}

// Main application entry point
void app_main(void) {
    esp_log_level_set("*", ESP_LOG_INFO);

    // Initialize GPIO pins using direct register access
    gpio_set_direction(BUZZER_PIN, GPIO_MODE_OUTPUT);
    gpio_set_direction(INDICATOR_PIN, GPIO_MODE_OUTPUT);

    // Initialize WiFi, MQTT, and PN532 module
    wifi_init();
    mqtt_app_start();
    setup_pn532();

    // Main loop to read from PN532 and handle MQTT communication
    while (1) {
        read_pn532();
        vTaskDelay(1000 / portTICK_PERIOD_MS);
    }
}
```

Listing 1: C Code for programming ESP32 IC - main.c

The CMakeLists.txt file is a configuration file used by CMake, a cross-platform build system generator. CMake automates the process of configuring build environments, making it easier to manage large projects and their dependencies. When working with the ESP-IDF (Espressif IoT Development Framework) for ESP32, CMakeLists.txt is used to specify how the project should be built.

```
# In main/CMakeLists.txt

idf_component_register(SRCS "main.c" "pn532.c"
                    INCLUDE_DIRS "."
                    REQUIRES esp_netif esp_wifi mqtt spi_master)
```

```
                   /your_project
                             CMakeLists.txt
                             sdkconfig
                             main
                                 CMakeLists.txt
                                 main.c
                                 pn532.c
```

```
        pn532.h
    components
        Adafruit-PN532
            Adafruit_PN532.cpp
            Adafruit_PN532.h
            ...
            library.properties
```

Listing 2: Project Directory Structure in ESP-IDF

1. `setupWifi()`: This function is responsible for setting up the WiFi connection. It attempts to connect to a WiFi network with the SSID "NTWS-4G" and the password "XXXXXXXXXXXXX". It waits in a loop until the connection is established.
2. `setupMqtt()`: Sets up the MQTT client with the MQTT broker's address and port. It initializes the `PubSubClient` object `mqttClient` to communicate with the MQTT broker at "test.mosquitto.org" on port 1883.
3. `setupPn532()`: Initializes the PN532 NFC module. It starts the PN532 module and checks the firmware version. If the firmware version cannot be retrieved, the function enters an infinite loop, halting further execution.
4. `readPn532()`: Reads the NFC tag using the PN532 module. It checks if a card is present and reads its unique identifier (UID). If a new card is detected (UID is different from the previous one), it converts the UID to a string and publishes it to the MQTT topic "210610H-CardDetails". It also calls the `Indicate()` function to indicate a successful read.
5. `connectToBroker()`: Attempts to connect to the MQTT broker. If the connection is successful, it publishes a message to the MQTT topic "210610H-Details" indicating that the RFID-based asset tracking device is connected. If the connection fails, it retries after a delay of 5 seconds.
6. `Indicate()`: This function indicates a successful NFC tag read. It activates a buzzer and LED for a short duration to provide visual and auditory feedback.

7.2 Compatibility Issue Between the Adafruit PN532 library and ESP32 IC

When implementing the software component, I encountered an issue where the PN532 board was not being properly detected by the ESP32 board, despite correct wiring. Upon investigation, I identified this as a problem stemming from the Adafruit PN532 library.

The specific issue with the Adafruit PN532 library pertains to the communication bitrate configuration when used with an ESP32 board. By default, the library sets the bitrate to 1000000 (1 MHz), which typically functions adequately. However, when the ESP32 is operating at a clock speed of 240 MHz, communication becomes erratic and prone to glitches.

To mitigate this issue, it is necessary to adjust the bitrate in the Adafruit PN532 library to 100000 (100 kHz) when the ESP32 is running at 240 MHz. This lower bitrate stabilizes communication and prevents glitches from occurring. This indicates a potential need for optimization or adjustment in the library to ensure compatibility with higher clock speeds.

So you need to add the following code to the `Adafruit_PN532` library to resolve that problem.

```
Adafruit_PN532::Adafruit_PN532(uint8_t ss) {

    #ifdef ESP32
        uint32_t frequency = 100000;
    #else
```

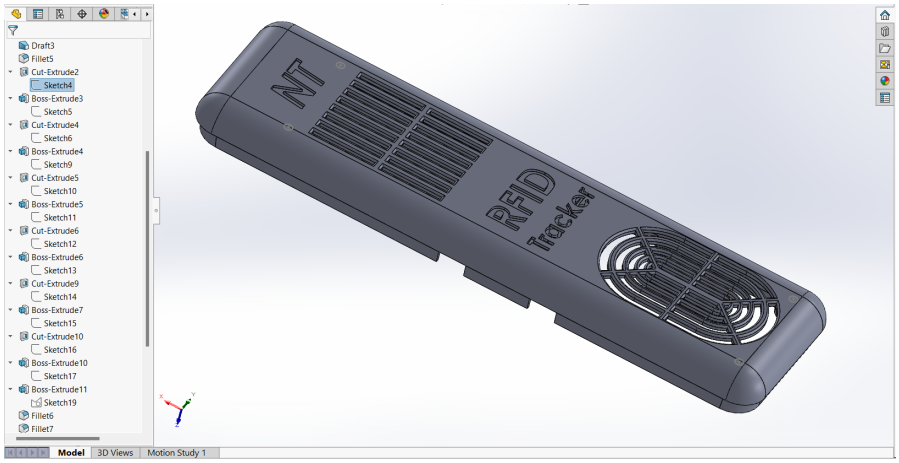
```
        uint32_t frequency = 1000000;
    #endif

    spi_dev = new Adafruit_SPIDevice(ss, frequency, SPI_BITORDER_LSBFIRST,
        SPI_MODE0);
}
```

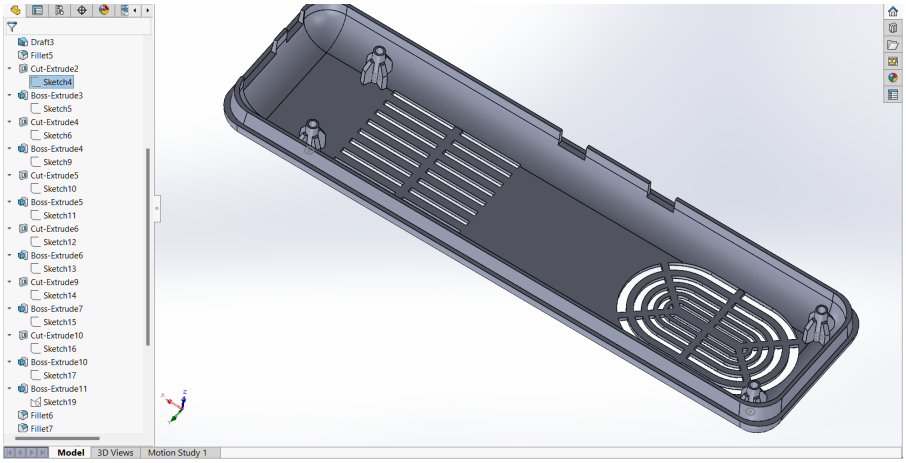
Listing 3: Additional code for Resolve the issue

8 Enclosure Design (08/03/2024)

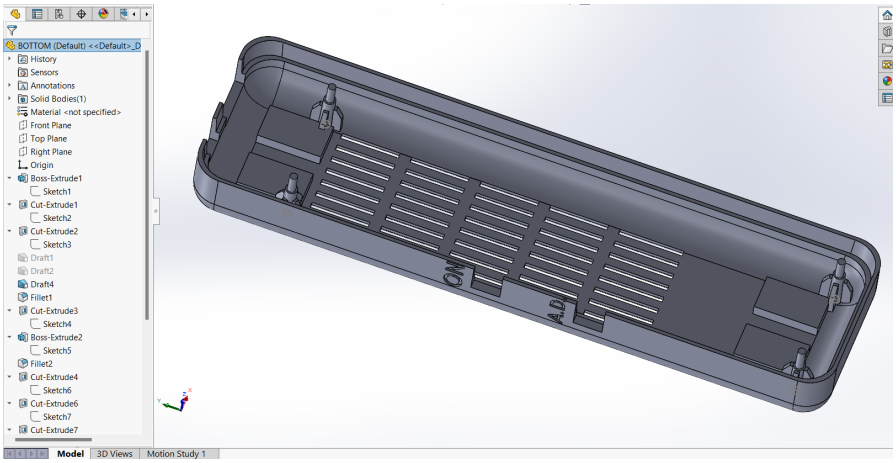
This section showcases the SolidWorks designs of the enclosure for the RFID-based asset tracking device, presenting detailed images of the top, bottom, and assembly views. These images highlight the design's aesthetics, functionality, and compatibility with the forklift, providing insights into the enclosure's robustness and suitability for industrial use.

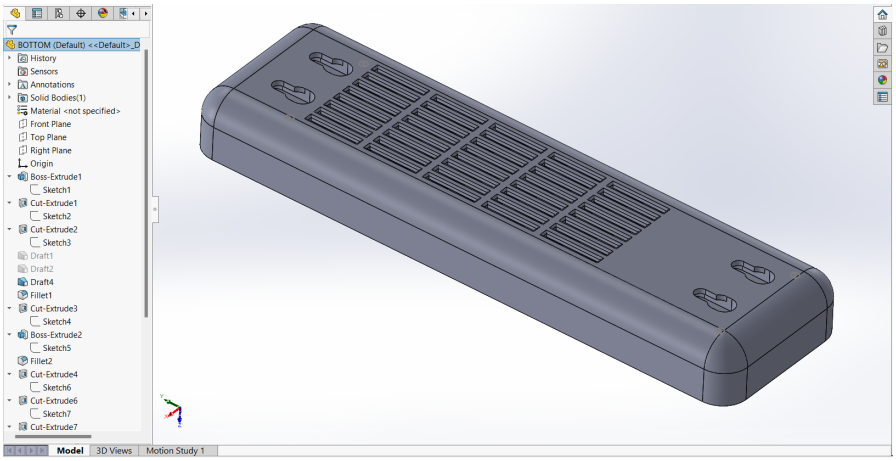
8.1 Upper Housing Component



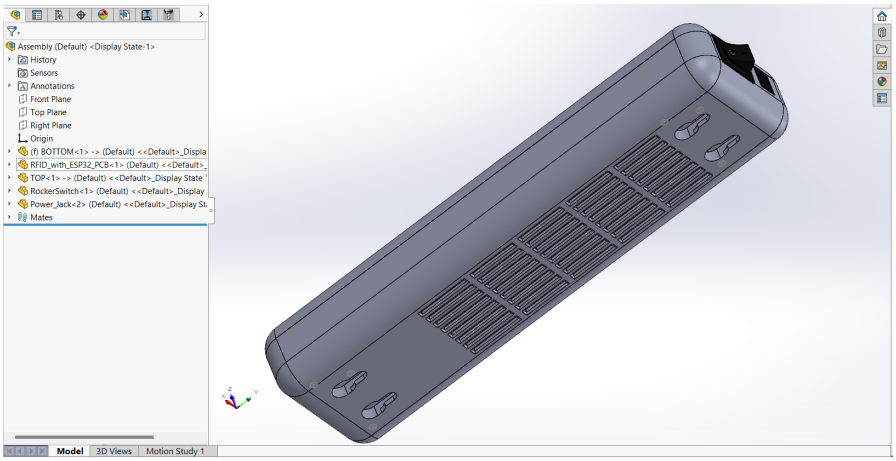


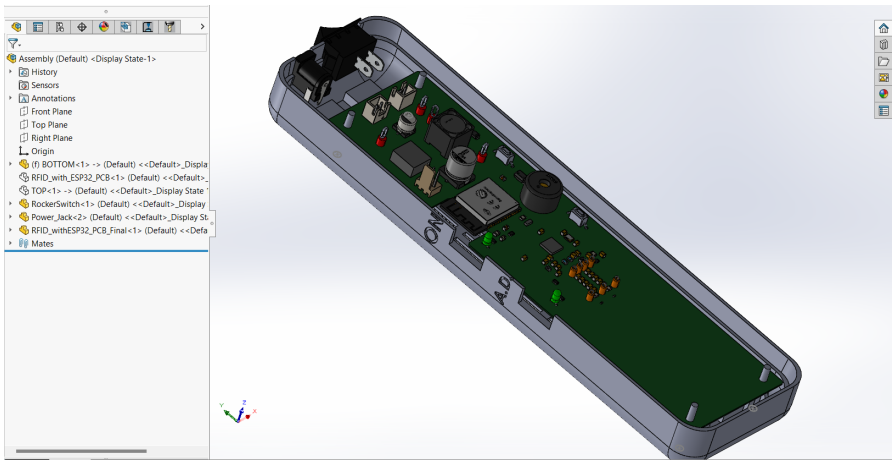
8.2 Lower Chassis Component





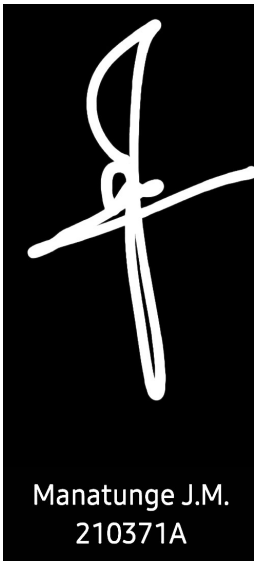
8.3 Enclosure Integration System





(Nethmal W.D.I.
210421E)

(Perera L.C.S
210463H)



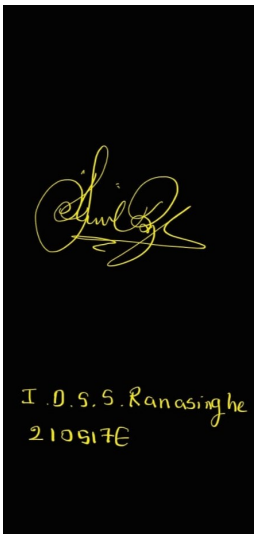


Figure 13: Signs Of Other Members Of Main Group