JOMO KENYATTA UNIVERSITY OF AGRICULTURE

AND TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

FINAL YEAR PROJECT

PROJECT TITLE:

DESIGN AND IMPLEMENTATION OF A WIRELESS PALLET TRUCK FOR AUTOMATED PALLET HANDLING IN LOGISTICAL OPERATIONS.

Submitted by:

APIYO FRANK - EN271-0484/2015

PROJECT SUPERVISOR

PROF. NYAKOE

A Final Year Project submitted to the Department of Electrical and Electronic Engineering in partial fulfillment of the requirements for the award of a Bachelor of Science Degree in Electrical and Electronic Engineering.

NOVEMBER 2020

DECLARATION

This final year project is my original work, except where due acknowledgement is made in the text, and to the best of my knowledge has not been previously submitted to Jomo Kenyatta University of Agriculture and Technology or any other institution for the Award of a degree or diploma.

SIGNATURE………………………………………… DATE ………………………………

NAME: APIYO FRANK **REG No.:** EN271-0484/2015

TITLE OF PROJECT:

DESIGN AND IMPLEMENTATION OF A WIRELESS PALLET TRUCK FOR AUTOMATED PALLET HANDLING IN LOGISTICAL OPERATIONS.

SUPERVISOR CONFIRMATION:

This project has been submitted to the Department of Electrical and Electronic Engineering, Jomo Kenyatta University of Agriculture and Technology, with my approval as the University supervisor:

NAME OF SUPERVISOR: PROF. NYAKOE

SIGNATURE: ……………………………… DATE: ……………………………………….

ABSTRACT

Retail and logistics operations require an active workforce with high mobility, accuracy, and pace in the location, picking, sorting and packaging of goods and customer orders. However, despite a competent labor pool, workforce safety is compromised due to factors including heavy loads and confined spaces in warehouses while productivity is reduced due to factors including time spent in locating goods as in the case of large warehouses.

This project seeks to ensure safety of the warehouse worker while increasing productivity by providing a system that will track inventory and ensure faster location of goods within a warehouse while also increasing the mobility of goods within the warehouse.

The solution will involve a wireless motion-controlled pallet truck that reads motion input from a user and has motorized wheels and a user-controlled steering system. A user interface synced to a warehouse management software will help guide workers through warehouse isles. The pallet truck will be able to support slightly heavy loads that don’t necessitate the use of forklifts ensuring worker safety as well and increasing mobility. The technology also guarantees job security in an already tight labor market in the logistics and distribution operations, where workers pulling carts or driving forklifts are increasingly working alongside machines built to keep goods moving at a rapid pace.

The solution will be offered through use of sensors, wireless transmitters, actuators, display systems and a microcontroller.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

ADC	Analog-to-Digital Converter
AGV	Automatic Guided Vehicles
AMR	Autonomous Mobile Robots
AS/RS	Automated Storage and Retrieval Systems
GND	Ground
GPIO	General Purpose Input-Output
JKUAT	Jomo Kenyatta University of Agriculture and Technology
I/O	Input/Output
IC	Integrated Circuit
IDE	Integrated Development Environment.
IoT	Internet of Things
UART	Universal Asynchronous Receiver Transmitter
PWM	Pulse Width Modulation
RX	Receiver
SoC	System on a Chip
TCP	Transport Control Protocol
TX	Transmitter

CHAPTER ONE: INTRODUCTION

1.1. BACKGROUND

Retail and logistics operations involve massive repetitive tasks in the movement and handling of goods. These can be classified as predictable physical work and, according to McKinsey and Company, have the highest susceptibility to automation [1]. Furthermore, the transportation and warehousing industry has the third highest automation potential of any sector in the global logistic chain [2]. There is therefore a need to develop new capabilities in the handling of goods in warehouses where picking, sorting, and palletizing of goods are the critical processes.

The warehousing sector grew massively in the last decade and with the soaring logistic volumes came an immense strain in the available warehousing capabilities. Increasingly, logistics companies are resorting to automation and AI powered capabilities to help move online orders at a faster pace. Many logistic operations in the warehousing sector will be automated by 2030 [1]. Currently the major players in the industry including Amazon, XPO logistics, DB Schenker, JD.com among others have developed advanced technologies that have reduced the manual handling of goods. Ocado, a British online grocery store, has fully automated warehouses that fulfill an estimated 190,000 orders per week while DHL has more than 100 automated parcel handling facilities in Germany that have greatly impacted their sorting and parcel handling ability. Such advances in technology include the use of Autonomous Guided Vehicles (AVG) such as Swarm robots which move cases and pallets on digital tracks based on warehouse management software [3].

Despite such novel technologies, there is still need for an active workforce with high mobility, accuracy, and pace in the location, picking, sorting and packing of goods at fulfillment centers. The option of automation is also not warranted in cases where there is abundant supply of workers and where these can be obtained at low costs [1].

Automating some warehouse and logistic processes also does not mean that all workers will be redundant. Many of these workers are reassigned to new tasks involving collaboration with the machines [1]. A modular goods-to-person approach where workers are increasingly working alongside machines is a welcome idea in the local warehousing and logistics sector.

Based on these considerations, there is therefore a need to innovatively develop retrofit equipment such as retrofit standard pallet trucks that are automated [2]. The proposed system will be able to track inventory while providing higher output levels, better quality and fewer errors in the movement and handling of goods [1].

1.2. PROBLEM STATEMENT

For the success of retail and logistics operations, an active workforce with high mobility, accuracy and pace in the location, picking, sorting and packaging of goods and customer orders is critical in order to guarantee higher output levels, better quality and fewer errors in the movement and handling of goods. However, despite a competent labor pool, there are gaps that ought to be addressed:

i. workforce safety is compromised due to factors including heavy loads and confined spaces in warehouses
ii. productivity is reduced due time spent in locating goods as is the case in large warehouses as well as due to errors in the movement and handling of goods.
iii. as logistics volumes increase especially in the peak holiday seasons, there is a need to automate some tasks and reassign workers to new roles while incurring as minimal labor substitutions as possible
iv. with the high demand and movement of goods occasioned by e-commerce, there is a need for a more robust inventory management system

1.3. PROJECT JUSTIFICATION

It has been shown that automated pallet handing systems cut shipment processing time by up to 50% [1]. The current warehouse capabilities have low output levels and are prone to many errors in the movement and handling of goods within the warehouse. Worker safety is also jeopardized in the case of heavy goods that must be retrieved and shipped.

The proposed solution is to develop retrofit standard pallet trucks that are automated in a modular goods-to-person approach where workers are increasingly working alongside machines. It guarantees the safety of the warehouse worker while increasing the mobility of goods within the warehouse. The system as well ensures that as minimal labor substitutions as possible are incurred by reassigning workers to new roles involving collaboration with the machines. There is also the ease of keeping track of inventory.

1.4. OBJECTIVES

1.4.1. MAIN OBJECTIVES

i. To develop, test and implement a wireless automated pallet truck for automated pallet handling in logistical operations.

1.4.2. SPECIFIC OBJECTIVES

a) To develop and implement a sensor circuit for driving directions for the pallet truck.
b) To develop and test a wireless system that transmits data read from the sensors to a webserver
c) To develop and implement a display system that will update the location of the pallet truck in the warehouse
d) To develop and implement an actuator circuit using servos that sets the pallet truck in motion according to sensor inputs

CHAPTER TWO: LITERATURE REVIEW

2.1. AUTOMATION IN THE LOGISTICS AND WAREHOUSING INDUSTRY

The logistics and warehousing industry has seen as an emergence in automation in various degrees. Traditional logistic operation involves mostly manual work done by workers who are involved in the picking, sorting, and packing of goods. Automation is on course to radically upset the global supply chain far beyond the warehouse and fulfillment center, affecting the way good and services move from one part of the supply chain to the next. Automation in warehousing has been on the rise due to three major factors; that is the shortage of labor, rise of the e-commerce sector which offers a huge demand from online retailers. Importantly, automation has been helped by the technical advancement in the field of robotics with the maturity of technologies including Artificial Intelligence (AI), Machine Learning (ML), IoT, and Cloud Computing among other novel technologies. The entrance of the established technology companies in the automation industry has further helped bolster the future of automation. Siemens for example took up a 50% stake in Magazino which is actively building automated picking solutions [6]. The transport-and-warehousing sector, according to an industry report by McKinsey & Company, has the third highest potential of any sector for automation, only bettered by delivery and customer services across the global logistics chain [2]. The sector has provided a market for autonomous mobile robots (AMR) or Commercial Service Robots which are intended to boost the output of human workers.

There has therefore been a need to innovate new handling devices to affect the whole supply chain in the warehousing sector. The devices are to solve one of these needs:

i. Enable safer and more enhanced physical collaboration between robots and humans in unpredictable environments
ii. Keep goods moving at a rapid pace
iii. Enable the few workers at the disposal of a company to focus on tasks that inherently require human interaction and emotional intelligence

There has been observed in rise in the use of robots in non-value adding movement of materials. As employed in the transport and warehousing sector, robots are increasingly being used to improve the jobs and output of already existing workers.

Much of the technologies and abilities developed have resulted in a sector where workers pulling carts or driving forklifts are increasingly working alongside machines [2].

2.2 CASE FOR AUTOMATION

There is a need to perform tasks that are oriented towards the customer needs. A constantly evolving demand on warehousing facilities where there is a need for efficiency, productivity, cost and scalability, calls for organizations within the warehousing sector to adapt and look for ways to accelerate delivery and manage product variety. According to IDC's John Santagate, the solution lies with the AMRs which shifts the focus of the workforce from mobility of goods to focus more on service delivery to the customer [4].

This has been enabled by collaborative robots among them the industrial robots and commercial service robots which are a part of a huge market straddling vision systems, intelligence software, motion control among a host of other industry players.

The uptake of robots has been helped hugely by the ability of the major players to deliver robots that are safer in dynamic and new environments meaning they can be relied upon to deliver reliable services in unpredictable systems. Robots also experience few disruptions. A WSJ article quotes a Gap executive saying that a single SORT robot, which uses AI to decide which item of clothing would be best to pick, is an equivalent of four workers working across four shifts [3].

Another reason for automation is that majority of the hours spent at the workplace involves physical activities including operation of machines in predictable environments making such processes highly susceptible to automation.

The relatively higher costs involved in substitution of labor with a view to achieve higher levels of output and fewer errors in the handling of goods, as opposed to reducing the labor costs seems to be a deterrence to automation but it has been shown that there are high levels of success. Some companies have seen the reduction in the shipment-processing time by up to 50% simply by deploying automated-pallet handling systems.

However, it is wise to carry out a technical feasibility in order to justify automation. The sector has huge potential, but it offers bigger uncertainty. If workers are abundant in supply, then automation becomes significantly expensive [1].

2.3 OPPORTUNITIES FOR AUTOMATION

The major technology in the commercial service robots market is the AMR. These have gained the ability to safely navigate aisles and populated spaces to take the low-value movement of material through the industrial process. E-commerce business models that favor omnichannel delivery strategies as well as the shift to B2C models has led to the development of more robust devices that can handle a variety of payloads.

Beside AMRs, the following are some of the technologies related to my area of interest and which are set to revolutionize warehouse operations:

While automation is poised to affect the labor market in terms of layoffs, few occupations in fact will feel the brunt of automation as was observed in the US labor market where an introduction of barcode readers actually led to an increase in the number of cashiers contrary to the earlier fears that it would disrupt the retail industry [5].

2.4 MATERIAL HANDLING IN A WAREHOUSE

A pallet rack (see *Figure 2-1*) is a material handling storage aid system that is designed to store materials on pallets. Most allow for the storage of material in horizontal rows with multiple levels. Forklift and pallet trucks are used to load and unload the pallet trucks.

Figure 2-1. A warehouse showing the handling of goods using forklifts

Pallet racks are a mainstay in modern warehouses and distribution facilities. They optimize storage density of warehouses. The layout of the warehouse needs to be done efficiently to maximize the storage space.

Ability to move material easily and smoothly around the warehouse is also a requisite.

Enough space to move around the material handling equipment and weight carrying machines without obstructions must be allowed. A right layout of the warehouse floor helps minimize safety hazards.

Creating traffic lanes and categorizing them as ingoing and outgoing lanes help maintain the flow of shipments. If space is limited in the warehouse, basic road traffic rules or consistent traveling rules can be made to prevent accidents.

Material handling is divided into:

a. **Manual handling -** Manual handling refers to the use of a warehouse personnel to move individual containers by lifting, lowering, filling, emptying, or carrying. It can expose workers to physical dangers that can lead to injuries involving strains and sprains to the lower back, shoulders, and upper limbs.[7]
b. **Automated handling -** Industrial robot are applied where it is technically and economically feasible. These equipment are used to reduce and sometimes replace the need to manually handle material. Most existing material handling equipment is only semi-automated because a human operator is needed for tasks like loading/unloading and driving that are difficult and/or too costly to fully automate. However, with sensing ability, machine intelligence, and robotics, it is becoming easier and possible to fully automate an increasing number of handling tasks.[8]

2.5 CURRENT TRENDS IN THE LOGISTICS INDUSTRY

There is a need for new warehouse devices that will aid the picking and palletizing of goods. These warehousing technologies can fall either into devices that assist the movement of goods or those that aid the handling of goods in such operations as picking, sorting, storage and packing.

Automatic Guided Vehicles (AGVs) fall in the first category and move cases as well as pallets. My project focuses on this category of warehousing technologies. For cases where the demand for goods undergoes peak-seasonal shifts, case in point to 2017 when Alibaba processed more than 800 million orders in a single day, and for the alternate case where labor is in abundance, then novel solutions can be developed that include retrofit equipment and software which remain manual when demand is slower and switched on whenever the demand peaks [2].

More recent technologies that have impacted pallet handling include swarm robots such as Kiva Robots (*See Figure 2-3* and *Figure 2-4*) employed by Amazon [9]. These move shelves with goods to picking stations or drop goods on advanced conveyors that are omnidirectional. Chuck units from 6 River Systems navigate warehouse aisles and notify the workers when they reach the exact aisle where goods are to be retrieved [6].

Advanced Automated Storage/Retrieval Systems (AS/RSs) (see *Figure 2-2*) provide large storage racks for goods which can then be retrieved by the help of robotic shuttles that move in 3D on rails attached to the structure.

Figure 2-2. Automated Storage/Retrieval Systems

The trend is shifting towards fully automated high-rack warehouses with robots navigating the aisles. These come complete with augmented reality goggles that enable the operators to have a full view of the warehouse floor thus aiding the coordination of both workers and robots in a modular goods-to-person system.

Warehouse management systems synced to collaborative robots help in real-time tracking of inventory to match the ordering system. In some instances, workers have Bluetooth badges that are linked to robots which display important information on items to be retrieved via warehouse inventory software [3].

Figure 2-3. Kiva Robots doing the heavy lifting at a fulfilment facility.

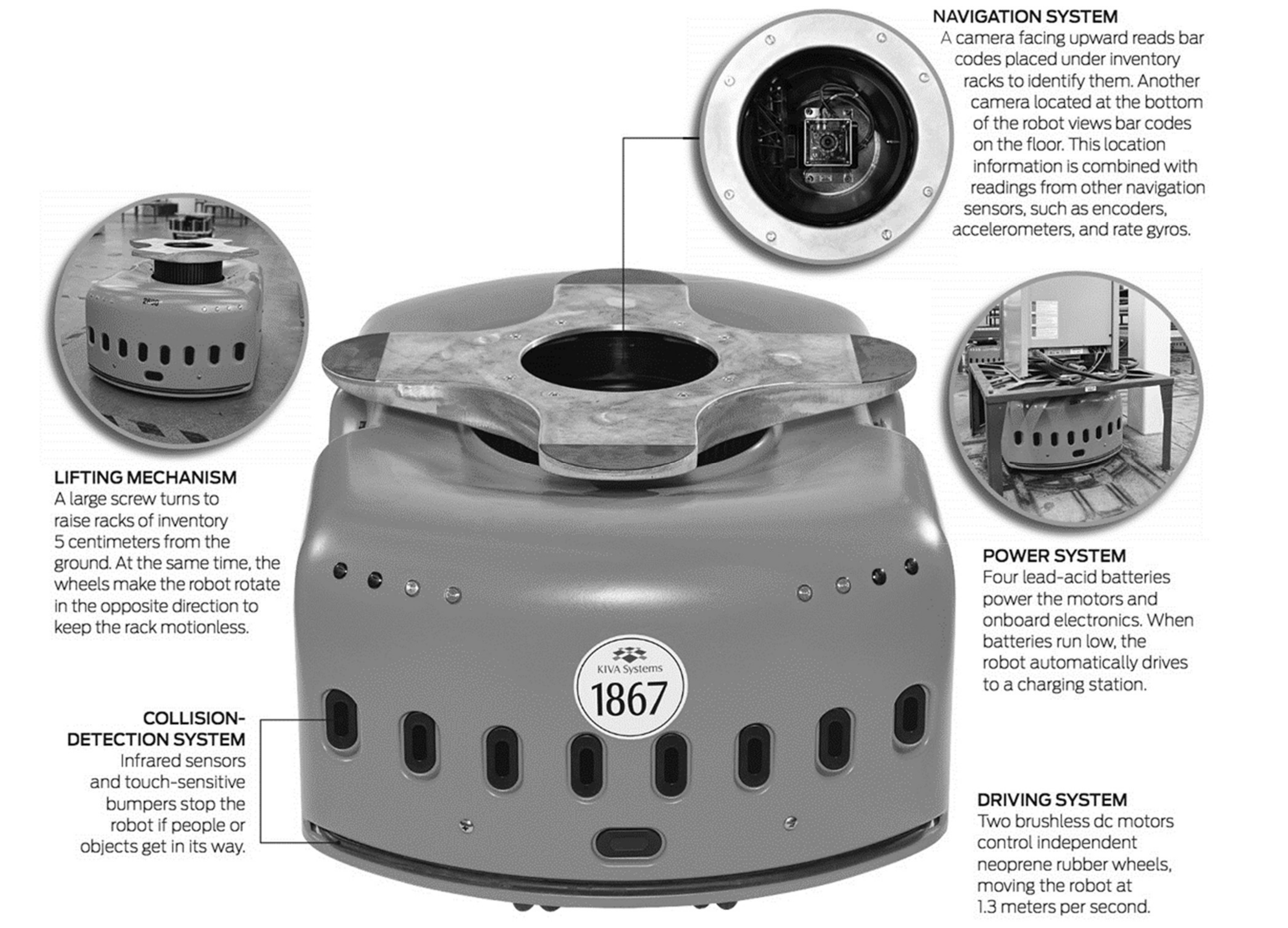

Figure 2-4. Kiva Robots' main systems

Major retailers that have demonstrated a number of technologies include the British online retail store that processes an estimated 190,000 orders per week via its fully automated warehouse in Andover, South East England (see *Figure 2-6*). Other major warehouse technology companies include CommonSense, GreyOrange and XPO logistics which has a fleet of more than 5000 collaborative robots operating in their logistics chain across Europe and North America. Geodi has deployed at least 150 robots from Locus Robotics to handle toys and apparel orders in North America [3].

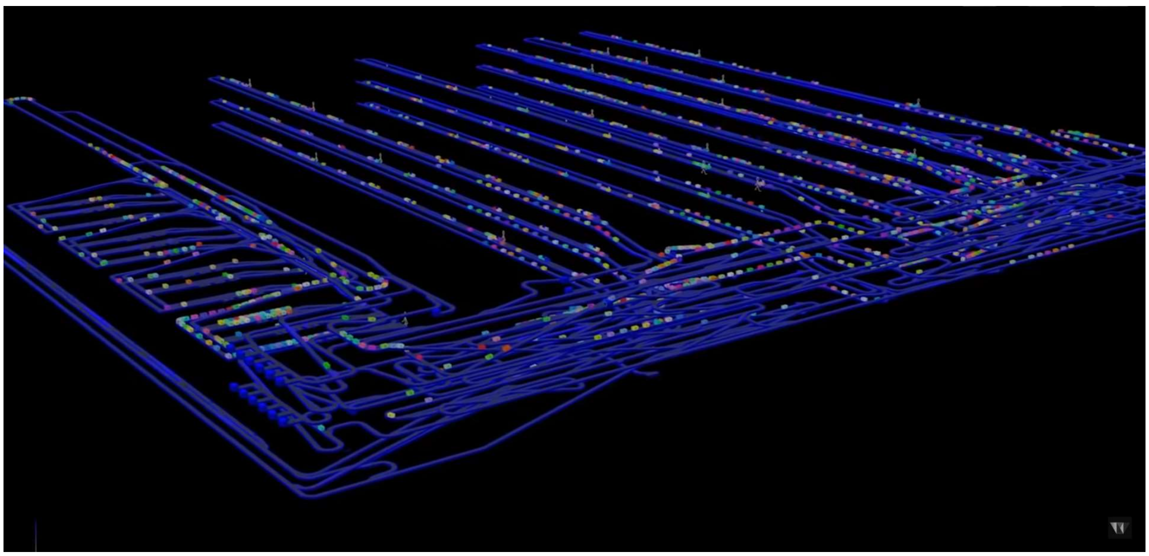

Figure 2-5. Ocado's warehouse management system showing over 35km of conveyor

Most devices are immobile and have goods brought to them while some like Magazino's TORU cube move and retrieve goods from warehouse shelves. (see *Figure 2-6).* My technology seeks to emulate the latter.

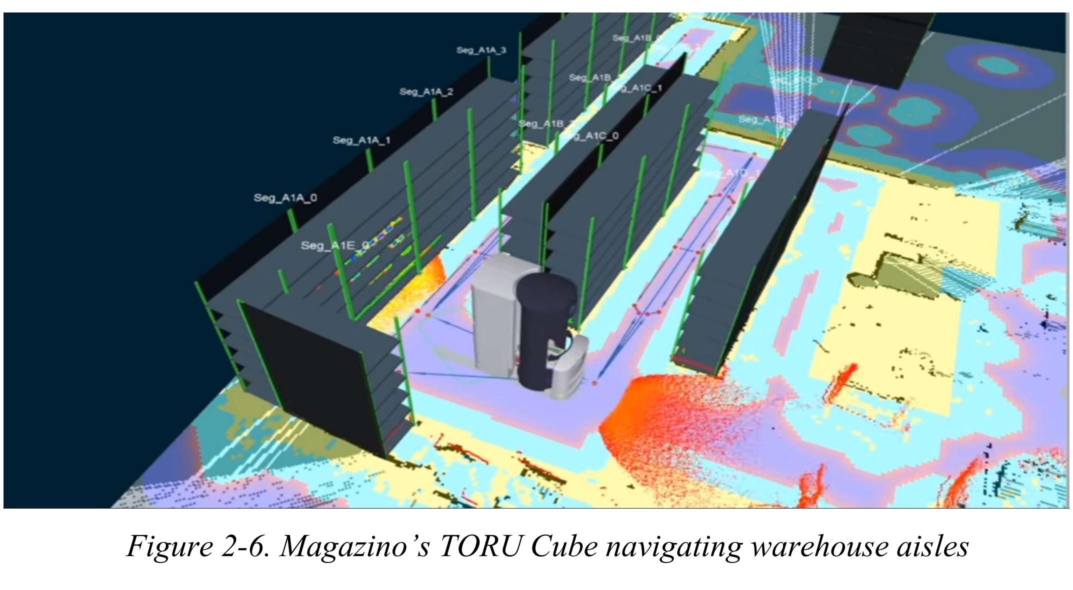

Figure 2-6. Magazino's TORU Cube navigating warehouse aisles

Current storage and retrieval systems including the motorized mobile pallet racks have several limitations including:

i. As the warehouse storage capacity is increased, costs go up
ii. Typical warehouse storage equipment have static parts and only a few forklifts are able to access the material
iii. Due to the stacking as space is maximized, the access speed of most goods is slowed due to the high storage density.
iv. Most AS/RS devices are considered expensive for smaller order fulfillment operations and retail warehouses.

2.6 ADVANTAGES OF MOTORIZED PALLET TRUCKS

The solution to be developed seeks to augment existing AS/RS devices complete with management software to help move goods at a rapid pace and track inventory.

The system will as well ensure the present workforce is redeployed to more productive tasks while using machines to augment human effort with a bid to raise quality of work, improve safety and increase throughput. The improved ergonomics and safety features will ensure fewer accidents.

The system will work safely alongside humans and can easily be deployed with minimal disruption to workflow.

2.7 ESP8266 NODEMCU

NodeMCU is an open-source firmware and development board for IoT Applications. The firmware is based on Espressif Systems' ESP8266 Wi-Fi SoC with hardware based on the ESP-12 module. It is a high performance, high integration wireless SOC, designed for space and power constrained mobile platform design with ability to embed Wi-Fi capabilities within other systems, or to function as a standalone application.

NodeMCU is applied in:

- Prototyping of IoT devices
- Low power battery operated applications
- Network applications
- Applications that requiring multiple I/O interfaces with Wi-Fi and Bluetooth functionalities

2.7.1. Features of ESP8266 NodeMCU

The NodeMCU ESP8266 development board comes with the ESP-12E module containing ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency.

NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs.

It has in-built Wi-Fi / Bluetooth and Deep Sleep Operating features that make it ideal for IoT applications.

It can be powered via a micro USB jack and V_{IN} pin.

It supports UART, SPI, and I2C communication protocols.

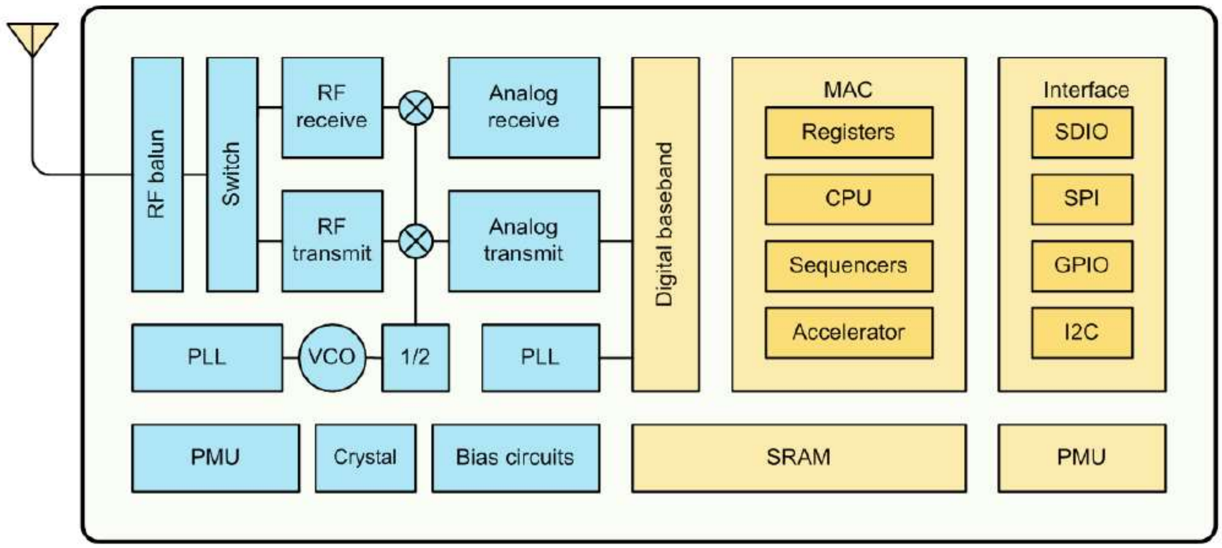

Figure 2-7. ESP8266EX Block Diagram

The ESP8266 NodeMCU supports a range of features including:

- 802.11 b/g/n
- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB

- SRAM: 64 KB
- Clock Speed: 80 - 160 MHz
- USB-TTL based on CP2102 is included onboard, Enabling Plug n Play
- PCB Antenna
- Integrated low power 32-bit MCU
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- WiFi 2.4 GHz, support WPA/WPA2
- Deep sleep power <10uA, Power down leakage current < 5uA
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)
- +20 dBm output power in 802.11b mode
- Operating temperature range -40C ~ 125C

2.7.1.2 ESP8266 NodeMCU Pinout and Ports

The NodeMCU ESP8266 development board comes with the ESP-12E module with 16 GPIO pins with pinout as shown below:

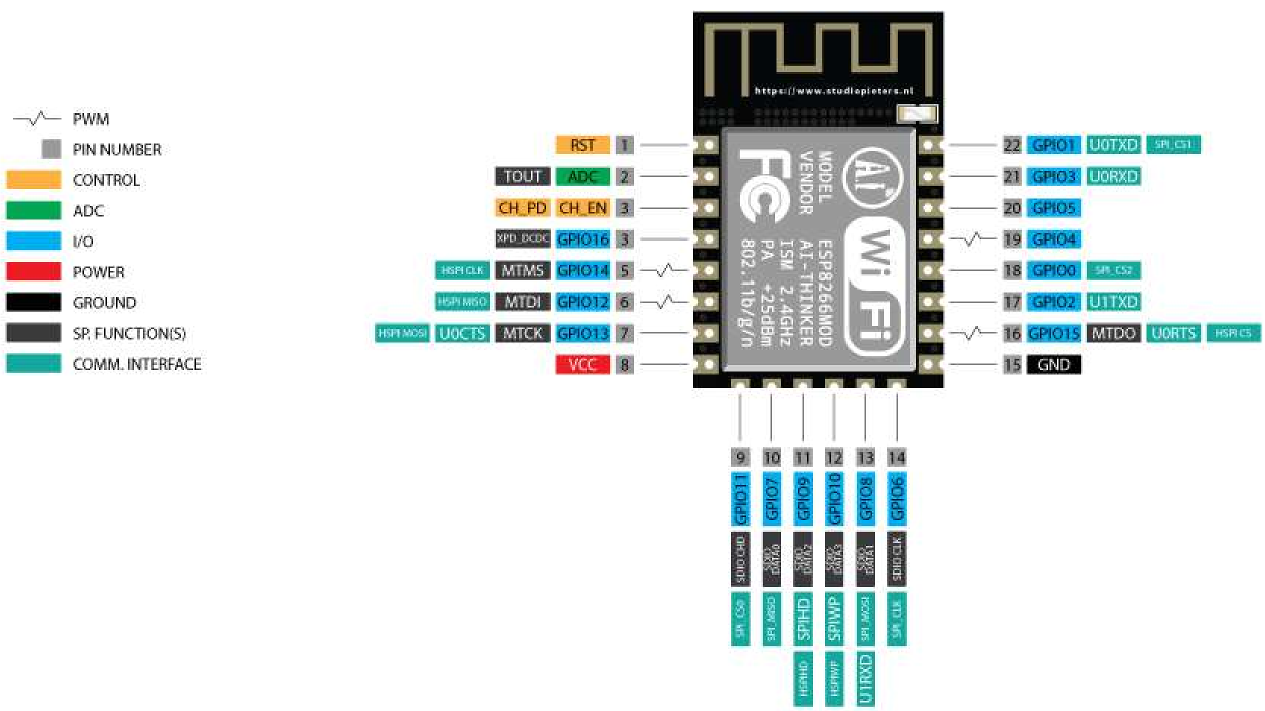

Figure 2-8. ESP8266 12-E chip pinout

Pin Category	Name	Description
Power	Micro-USB, 3.3V, GND, V_{in}	**Micro-USB:** NodeMCU can be powered through the USB port **3.3V:** 3.3V supplied via this pin to power the board **GND:** Ground **V_{in}:** External Power Supply
Control Pins	EN/RST FLASH/BOOT	Resets the microcontroller Sets the microcontroller into bootloader mode
Analog Pin	A0	Has internal voltage divider for voltage input range of 0 - 3.3V
GPIO Pins	GPIO1 to GPIO16	General purpose input-output pins
SPI Pins	SD1, CMD, SD0, CLK	SPI communication.
UART Pins	TXD0, RXD0, TXD2, RXD2	NodeMCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program.
I2C Pins		Implemented on software
PWM Pins	All I/O pins: GPIO0 to GPIO16.	Allows for software implementation of PWM PWM signals on ESP8266 have 10-bit resolution.
INTERRUPT pins	Any GPIO pin except GPIO16	

Table 2-1: NodeMCU Pinout

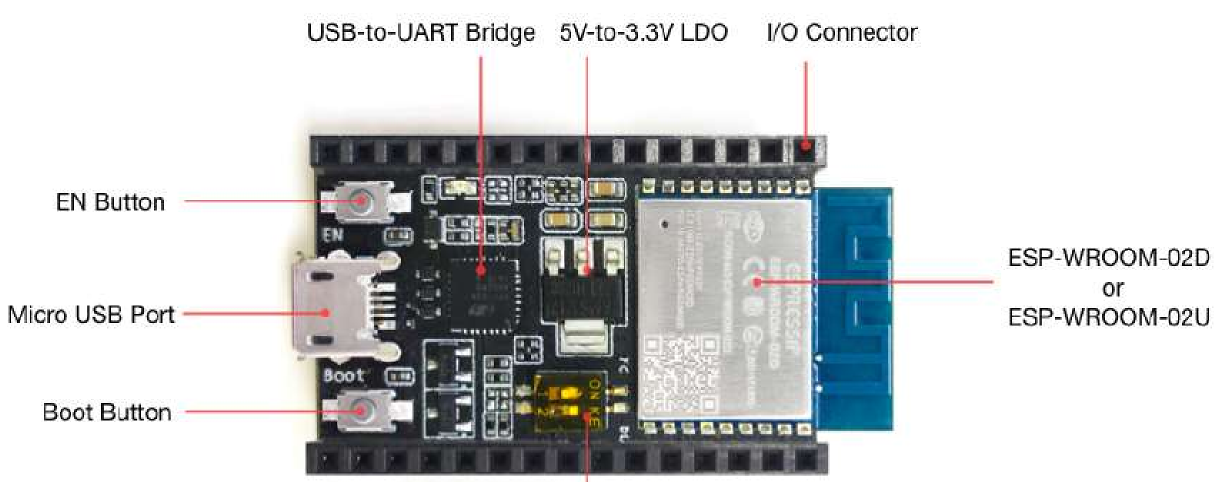

Figure 2-9. ESP8266-NodeMCU

The key components, interfaces, and controls of the ESP8266 NodeMCU include:

- **5V to 3.3V LDO**

A LDO regulator with a maximum current output of 800 mA provides power supply for the ESP8266 module and user peripherals.

- **USB-to-UART Bridge**

A single chip USB-UART bridge provides up to 3 Mbps transfers rates.

- **Boot/Flash Button**

Download button. Pressing the on-board FLASH/BOOT button sets the ESP8266 into bootloader mode. It is like when GPIO0 is pulled LOW. Holding down the FLASH button and pressing the RST button initiates the firmware download mode. Then users can download firmware through the serial port.

- **Micro USB Port**

USB interface. It functions as the power supply for the board and the communication interface between PC and the board.

- **EN/RST Button**

Reset button. When the RST pin is pulled LOW, the module resets.

- **I/O Connector**

All the pins on the ESP8266 module are broken out to breadboard compatible male headers.

2.7.1.3 ESP8266 NodeMCU Communication Protocols

2.7.1.3.1 I2C

The ESP8266 NodeMCU has no hardware pins for I2C which is instead implemented in software. Any GPIO pins can be used as I2C pins. The GPIO pins commonly used as I2C pins are:

GPIO5: SCL

GPIO4: SDA

2.7.1.3.2 SPI

The pins used as SPI in the ESP8266 NodeMCU are:

GPIO12: MOSI

GPIO13: MISO

GPIO14: SCLK

GPIO15: CS

2.7.2 Programming of the ESP8266 NodeMCU

The ESP8266 NodeMCU can be programmed to enable multiple functions. The NodeMCU Development Board is programmed using Arduino IDE via a USB cable. The program is written in Arduino/C++ which is an abstracted form of C++, a high-level language. The Arduino IDE version used is 1.8.13. To program the NodeMCU, IDE version 1.6.5 or higher is required.

Figure 2-10. Arduino IDE

2.8 RFID – RC522 Module

The RC522 is a 13.56MHz RFID module based on the MFRC522 controller from NXP semiconductors. The module supports I2C, SPI and UART protocols.

It is applied in automatic billing systems, attendance systems, verification and identification systems and access control systems.

The RC522 has the following features:

- Operating voltage: 2.5V to 3.3V
- Maximum Data Rate: 10Mbps
- Read Range: 5cm
- Current Consumption: 13-26mA
- Power down mode consumption: 10uA (min)

2.8.1. RC522 Pin Configurations

Pin Name	Description
V_{CC}	Supplies Power to the module (3.3V)
RST	Reset pin – used to reset or power down the module
GND	Ground
IRQ	Interrupt pin – used to wake up the module when a device comes into range
MISO/SCL/Tx	MISO pin (for SPI communication) or SCL(I2C) or as TX(UART).
MOSI	Master Out Slave In pin for SPI communication
SCK	Serial Clock pin – used for clock signal
SS/SDA/Rx	Serial Input pin (SS) (for SPI communication), SDA (I2C) or as RX(UART)

Table 2-2: RC522 Pinout

During application, the reader module waits for the tag to come into proximity. The Reader can be configured to Power Down mode to save power in battery operated applications using the IRQ pin. The minimum current consumed by the module during this mode will be 10uA.

The module is configured using the MFRC522.h Arduino library.

```
#include <MFRC522.h>

#define RFID_SS 10 //SD2
#define RFID_RST 9 //SD1
#define MINIMUM_TIME_BETWEEN_CARDS 2000
```

Figure 2-11. Configuring the RC522 using Arduino IDE

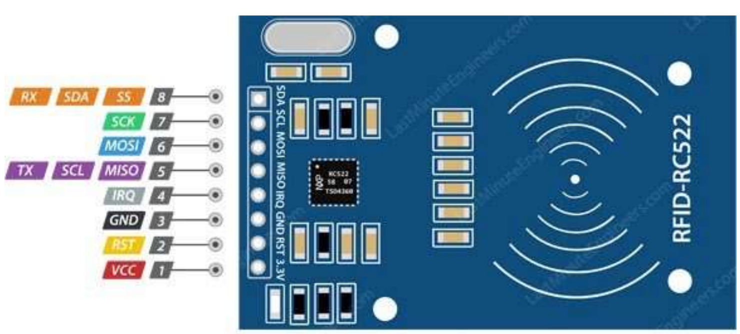

Figure 2-12. RC522 pin configuration

2.8 TCRT5000 IR Line Sensor Module

The line follower is an IR module that transmits Infrared light to a photo diode. The TCRT5000 is a black line follower and follows a black line. It is applied in robots for applications like factory floor management robots or warehouse robots.

It applies the behavior of light at the black and white surfaces. When light falls on a white surface it is almost fully reflected and in the case of a black surface light is completely absorbed.

The TCRT5000 uses IR Transmitters and IR receivers also called photodiodes which transmit and receive Infrared light. When infrared rays fall on the white surface, it’s reflected back and captured by the RX photodiode which generate some voltage changes. When IR light falls on a black surface, light is absorbed by the black surface and no rays are reflected, thus photo diode does not receive any light or rays.

The NodeMCU drives the motor according to the sensor output. When the TCRT5000 sensor senses a reflective surface then the NodeMCU receives a LOW input while for the black line Arduino it receives a HIGH input.

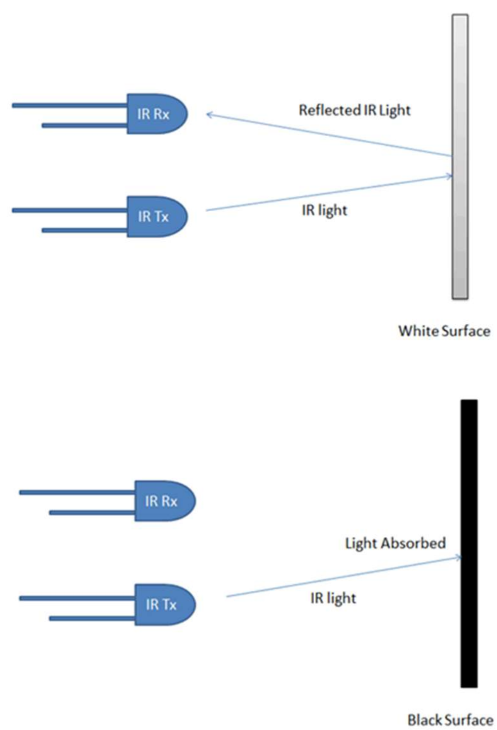

Figure 2-12. Working of an IR sensor

Two IR sensor modules, the left sensor and the right sensor are applied.

When both the left and right sensor sense reflective surface then the robot stops.

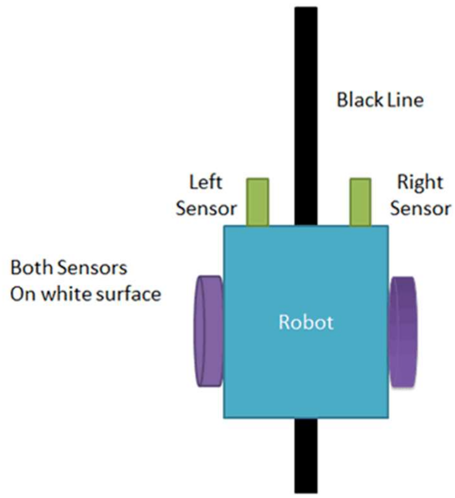

Figure 2-13. Robot on a reflective surface

If only the left sensor is on the black line, then the robot turns right while if only the right sensor is on the black line then robot turns right until both sensors are on the black line and the robot moves forward.

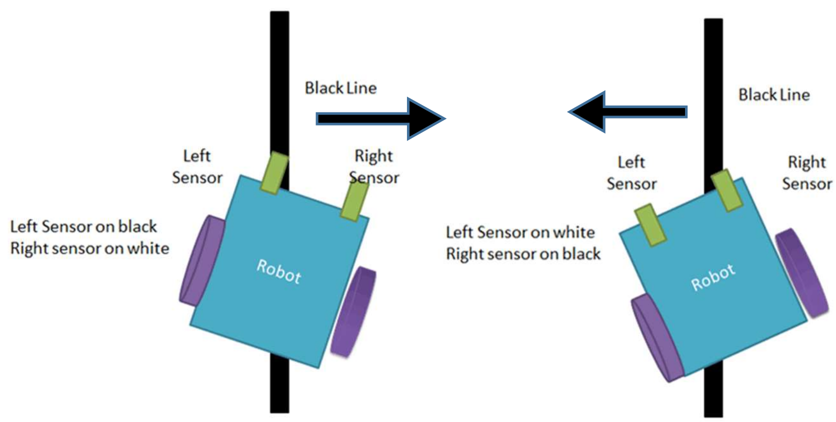

Figure 2-14. Either sensor on the black line

If both sensors come on the black line, the moves forward.

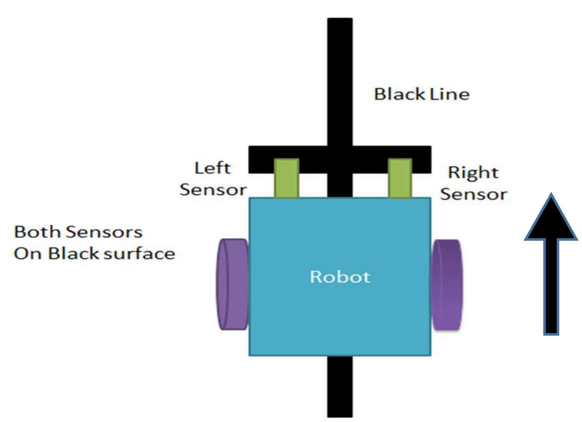

Figure 2-15. Robot on a black surface

2.9 MQTT

MQTT is a client server publish/subscribe messaging transport protocol that uses a publish and subscribe model to send messages to multiple clients.

MQTT is ideal for applications that involve constrained environments such as for communication in Machine to Machine (M2M) and IoT contexts where a small code footprint is required and/or network bandwidth is at a premium as it is light weight, open, simple, and designed for easy implementation. Its light weight and binary protocol, and minimal packet overhead means MQTT excels when transferring data over the wire in comparison to protocols like HTTP.

MQTT protocol is used in connected cars, manufacturing systems, logistics, military, enterprise chat applications, and mobile apps.

In MQTT a publisher publishes messages on a topic and a subscriber must subscribe to that topic to view the message. A central broker is required as shown:

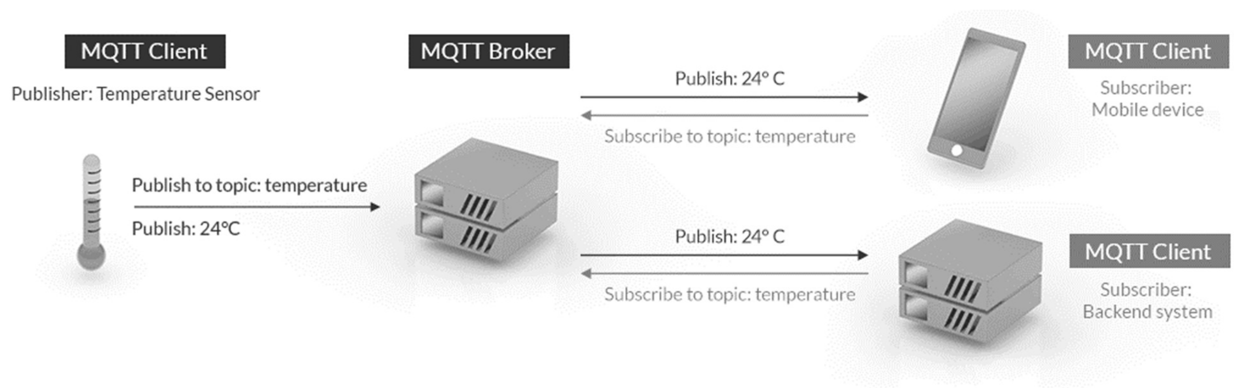

Figure 2-16. MQTT Publish / Subscribe Architecture (Source: https://mttq.org)

2.8.1. History of MQTT

The MQTT (MQ Telemetry Transport) protocol was invented in 1999 by Andy Stanford-Clark (IBM) and Arlen Nipper (Cirrus Link). They needed a protocol for minimal battery loss and minimal bandwidth to connect with oil pipelines via satellite. “MQ” refers to the MQ Series, which is an IBM product that supports MQ telemetry transport. When Andy and Arlen created their protocol in 1999, they named it after the IBM product. IBM used the protocol internally before releasing the free MQTT v3.1 in 2010.

MQTT is an officially approved OASIS Standard. The new MQTT 5 specification was ratified by OASIS in 2019 to include features required for IoT applications deployed on cloud platforms, and those that require more reliability and error handling to implement mission-critical messaging [11].

2.8.2. MQTT Client-Broker Connections

MQTT uses TCP/IP to connect to the broker. TCP is a connection orientated protocol with error correction and guarantees that packets are received sequentially. MQTT clients will remain connected to the broker even when not sending data. Connections are acknowledged by the broker using a Connection ACK message since MQTT is a command-response protocol.

MQTT clients publish a keepalive message at regular intervals typically 60s which tells the broker that the client is still connected.

Figure 2-17. MQTT client to broker protocol

2.8.3. Client ID

All MQTT clients must have a unique client name or ID otherwise the existing client connection is dropped. The client name is used by the MQTT broker to track subscriptions.

Topic names, Client ID, Usernames and Passwords are encoded as UTF-8 strings. The Payload excluding MQTT protocol information is binary data and the content and format is application specific.

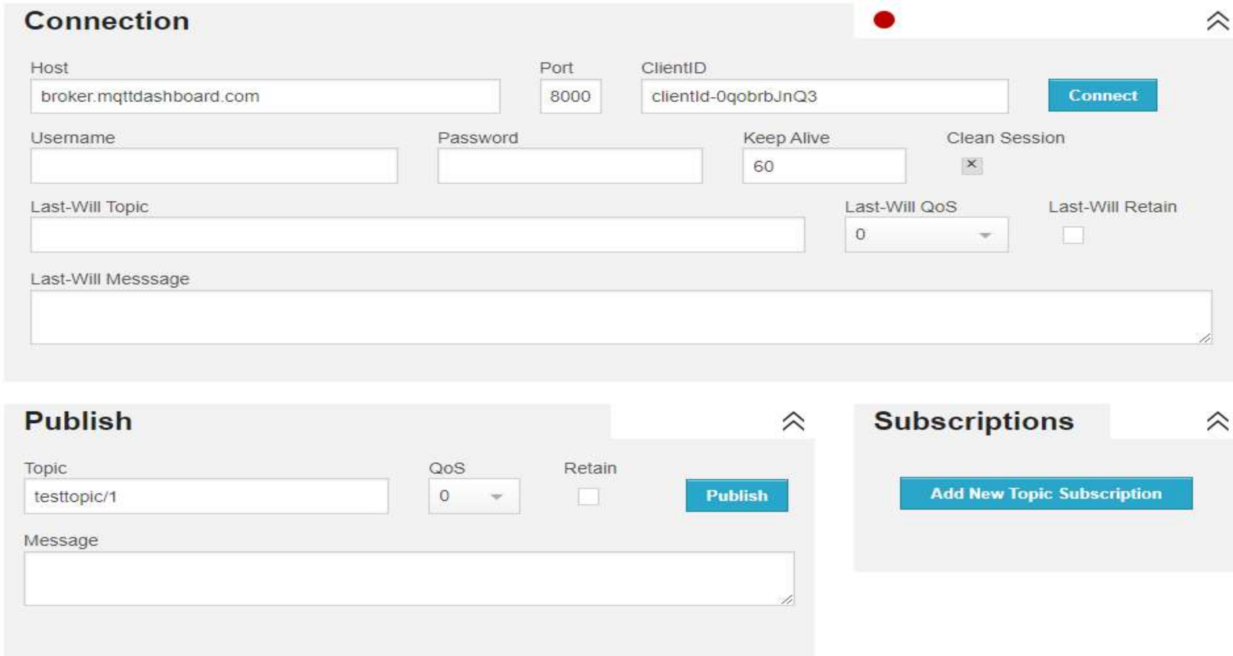

Figure 2-18. MQTT browser client dashboard

2.8.4. MQTT Topics

MQTT topics are a type of addressing that allows MQTT clients to share information. MQTT Topics are structured in a hierarchy using the forward slash (/) as a delimiter. This allows for a user friendly and self-descriptive naming structures.

Topic names are:

- Case sensitive
- use UTF-8 strings.

A client can subscribe to individual or multiple topics using wildcard characters. # allows for multi-level wildcard while + allows for single level wildcard.

A client can only publish to an individual topic. Using wildcards when publishing is not allowed. To publish a message to two topics the client will need to publish the message twice.

2.8.6. Clean Session

MQTT clients by default establish a clean session with a broker. A clean session is one in which the broker is not expected to remember anything about the client when it disconnects. With a non-clean session, the broker remembers client subscriptions and may hold undelivered messages for the client. However, this depends on the Quality of service used when subscribing to topics, and the quality of service used when publishing to those topics.

2.8.7. Arduino Client for MQTT

This Arduino library allows a ESP8266 client to do publish/subscribe messaging with a server that supports MQTT. It can only publish QoS 0 messages and subscribe at QoS 0 or QoS 1 [12].

The maximum message size, including header, can be configured via QTT_MAX_PACKET_SIZE in PubSubClient.h and has a default value 256 bytes.

The keepalive interval for the library is set to 15 seconds by default and can be configured via MQTT_KEEPALIVE in PubSubClient.h.

The client uses MQTT 3.1.1 by default. It can be configured to MQTT 3.1 by changing value of MQTT_VERSION in PubSubClient.h.

2.8.8. MQTT Programming

The important constructors and functions required to enable MQTT are:

- **PubSubClient ()**

Creates an uninitialised client instance. It must be initialized with the property setters.

- **PubSubClient (*client*)**

Creates a partially initialised client instance. The server details must be configured during initialization. *client* is the network client to use, in this case, WiFiClient

- **PubSubClient (*server, port*)**

Creates a fully configured client instance.

server is the IPAddress and is either a uint8_t[] or const char[] denoting the address of the server

port is the port to connect to. For webclients, the websocket port is 8000 while the TCP Port is 1883.

- **boolean connect (*clientID, [username, password]*)**

Connects the client.

clientID is the client ID used when connecting to the server

Credentials these are optional and include: *username* which is the username used. If the value is NULL, no username or password is required, and *password*.

- **boolean beginPublish (*topic, length, retained*)**

Begins sending a publish message. The payload of the message is provided by one or more calls to write followed by a call to endPublish.

topic is the topic to publish to

length is the length of the payload to be sent

retained a Boolean that indicates whether the message should be retained

- **boolean subscribe (topic)**

Subscribes to messages published to the specified topic. *topic* defines the topic to subscribe to

- **boolean loop ()**

This is called regularly to allow the client to process incoming messages and maintain its connection to the server.

- **boolean connected ()**

Checks whether the client is connected to the server.

- **int state ()**

Returns the current state of the client. If a connection attempt fails, this can be used to get more information about the failure. The values have corresponding constants defined in PubSubClient.h.

Value	**State**	**Description**
-4	MQTT_CONNECTION_TIMEOUT	The server did not respond within the keepalive time
-3	MQTT_CONNECTION_LOST	The network connection was broken
-2	MQTT_CONNECT_FAILED	The network connection failed
-1	MQTT_DISCONNECTED	The client is disconnected cleanly
0	MQTT_CONNECTED	The client is connected
1	MQTT_CONNECT_BAD_PROTOCOL	The server does not support the requested version of MQTT
2	MQTT_CONNECT_BAD_CLIENT_ID	The server rejected the client identifier
3	MQTT_CONNECT_UNAVAILABLE	The server was unable to accept the connection
4	MQTT_CONNECT_BAD_CREDENTIALS	The username/password were rejected
5	MQTT_CONNECT_UNAUTHORIZED	The client was not authorized to connect

Table 2-3: MQTT states

CHAPTER THREE: METHODOLOGY

The project was executed as follows:

a) Implementation of sensor circuits for driving directions for the pallet truck
b) Implementation of a wireless system that transmits data read from the sensors to a webserver
c) Implementation of a display system that updates the location of the pallet truck in the warehouse
d) Design and implementation of an actuator circuit using geared dc motors that sets the pallet truck in motion according to sensor inputs

3.1 IR Line Follower and RFID Reader.

The line follower IR module enables the pallet truck to move along a line or demarcations on the warehouse floor.

The circuit is controlled by an ESP8266 NodeMCU breakout board that is connected to a breadboard.

To enable autonomous control, we have two TCRT5000 IR modules fastened to the chassis using M3 screws and an RFID-RC522 module to detect RFID tags placed along the lines or demarcations on the surface.

The TCRT5000 IR sensor is used as a digital sensor to detect the reflective surface that form the demarcations.

The RFID module is powered using 3.3V from the 3v3 pin of the NodeMCU. The RC522 RFID reader operates at 13.56MHz which is a free industrial (ISM) band and stores information of stations identified by a unique ID into memory elements. The stations are represented by passive tags that operate at 13.56MHz. SPI is used for a maximum data rate of 10Mbps.

An ultrasonic sensor attached to the chassis detects obstacles.

The IR Line follower, Ultrasonic sensor and RFID tag were interfaced to the microcontroller as follows:

Line Follower Silkscreen	RFID Silkscreen	Ultrasonic Sensor	NodeMCU Silkscreen
Anode (5V) Collector (5V)	3.3V	5V	5V
Cathode (GND) Emitter (GND)	GND	GND	GND
-	MOSI	-	GPIO13(D7)
-	MISO (SPI communication)/SCL (I2C)/Tx (UART)	-	GPIO12(D6)
-	IRQ	-	-
-	SS (SPI communication), SDA (I2C)/Rx (UART)	-	GPIO4(D2)
-	SCK	-	GPIO14(D5)
	RST	-	GPIO5(D1)
-	-	ECHO	A0
-	-	TRIG	GPIO16(D0)
S (RIGHT)			GPIO3 (RX)
S (LEFT)			GPIO1(TX)

Table 3-1: NodeMCU and Sensor Interfacing

The schematic for the circuit layout and design files is shown in figure **below:**

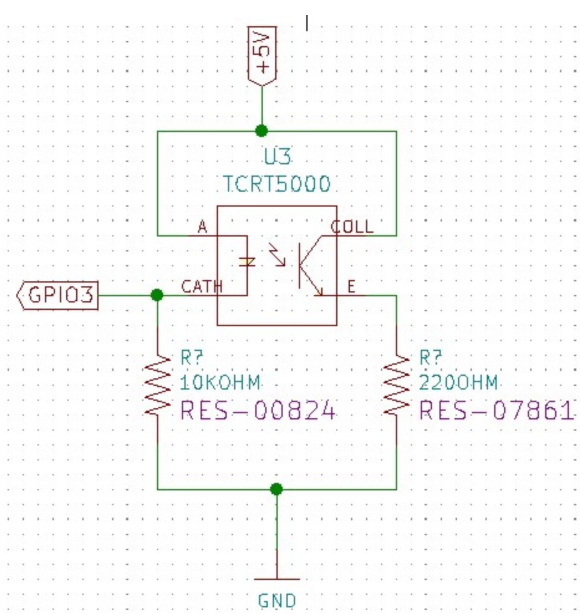

Figure 3-1. RCRT5000 schematic

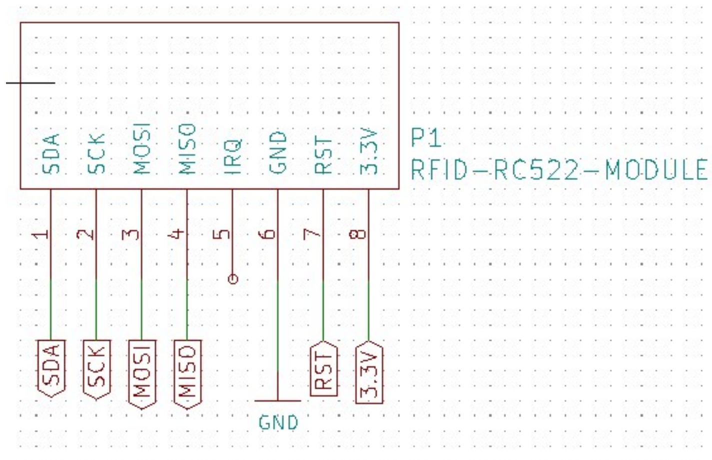

Figure 3-2. RFID-RC522 schematic

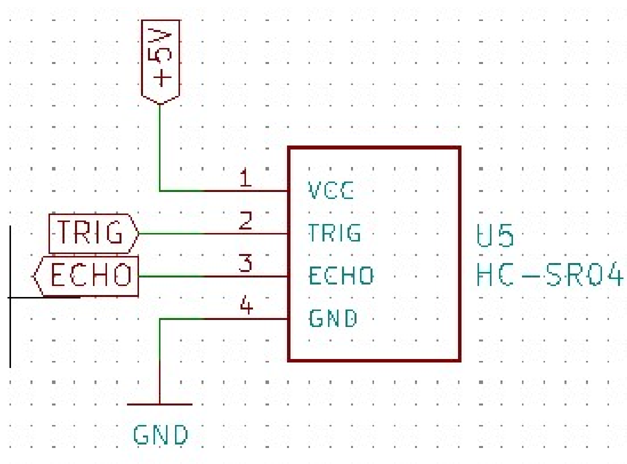

Figure 3-3. HC-SR04 Ultrasonic sensor schematic

The sensors were programmed using Arduino IDE and the code uploaded to the NodeMCU from the computer using a USB cable. The sample code below shows the initialization and interfacing of the sensors.

```
#include<NewPing.h>
#include <SPI.h>

int ENA = 8;  //ENA connected to digital pin 3
int ENB = 9;  //ENB connected to digital pin 9
int MOTOR_A1 = 2; // MOTOR_A1 connected to digital pin 4
int MOTOR_A2 = 3; // MOTOR_A2 connected to digital pin 5
int MOTOR_B1 = 4; // MOTOR_B1 connected to digital pin 6
int MOTOR_B2 = 5; // MOTOR_B2 connected to digital pin 7

int RIGHT = A2; // RIGHT sensor connected to analog pin A0
int LEFT = A3;  // LEFT sensor connected to analog pin A0

#define TRIG A0 // TRIG PIN connected to analog pin A2
#define ECHO A1 // ECHO PIN connected to analog pin A3
#define MAX_DISTANCE 100 // Define Maximum Distance

NewPing sonar(TRIG, ECHO, MAX_DISTANCE);

void setup() {
  // put your setup code here, to run once:

pinMode(ENA, OUTPUT); // initialize ENA pin as an output
pinMode(ENB, OUTPUT); // initialize ENB pin as an output
pinMode(MOTOR_A1, OUTPUT); // initialize MOTOR_A1 pin as an output
pinMode(MOTOR_A2, OUTPUT); // initialize MOTOR_A2 pin as an output
pinMode(MOTOR_B1, OUTPUT); // initialize MOTOR_B1 pin as an output
pinMode(MOTOR_B2, OUTPUT); // initialize MOTOR_B2 pin as an output
pinMode(RIGHT, INPUT); // initialize RIGHT pin as an input
pinMode(LEFT, INPUT);  // initialize LEFT pin as an input

Serial.begin(9600);


}
```

Figure 3-4. Initialization and interfacing of the sensors.

3.2 ESP8266 WiFi Communication.

ESP8266 Wi-Fi SoC is based on the ESP-12 module designed with ability to embed Wi-Fi capabilities within other systems.

The module is powered using 5V to the V_{IN} pin of the NodeMCU.

The schematic for the circuit layout and design files is shown in figure **below:**

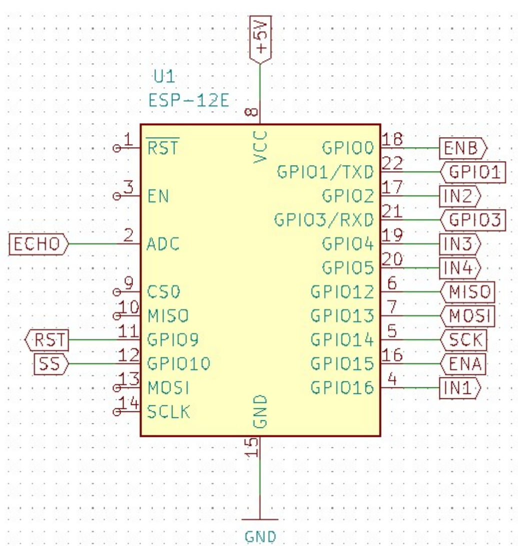

Figure 3-5. ESP8266 NodeMCU schematic

The microcontroller is programmed using Arduino IDE and the code uploaded to the NodeMCU from the computer using a USB cable. Flashing a program to the ESP8266 means resetting the microcontroller and starting it in flashing mode by setting GPIO0-to-GND.

Scanned information of stations which are unique IDs from the passive RFID tags are relayed via MQTT to a webserver hosted at broker.mqttdashboard.com.

Sample code for initializing the microcontroller in MQTT protocol is shown below:

```
// RFID setup
#include <SPI.h>
#include<NewPing.h>
#include <MFRC522.h>
#include <PubSubClient.h>
#include <ESP8266WiFi.h>
#include <time.h>

#define RFID_SS 10 //SD2
#define RFID_RST 9 //SD1
#define MINIMUM_TIME_BETWEEN_CARDS 2000

int ENA = 15;  //ENA connected to D8
//int ENB = 9;  //ENB connected to digital pin 9
int MOTOR_A1 = 16; // MOTOR_A1 connected to D0
int MOTOR_A2 = 5; // MOTOR_A2 connected to D1
int MOTOR_B1 = 4; // MOTOR_B1 connected to digital D2
int MOTOR_B2 = 2; // MOTOR_B2 connected to digital D4

int RIGHT = 3; // RIGHT sensor connected to RX
int LEFT = 1;  // LEFT sensor connected to TX

const char* WIFI_SSID = "SVEN";
const char* WIFI_PASSWORD = "_R___!_9S";
const char* mqttServer = "broker.mqttdashboard.com";
const int mqttPort = 1883;
//const char* mqttUser = "xtqtt";
//const char* mqttPassword = "f4ank";
const char* pubTopic = "apiyo/arduino";
```

Figure 3-6. Initializing the MQTT protocol

3.3 Display system

This is the user interface for the system, and it consists of a web application that updates the location of the pallet truck within the warehouse. The web application runs on the computer with the WebSocket at port 8000 being the subscriber in an implementation of the MQTT protocol.

The web application has a frontend built on JavaScript as shown below:

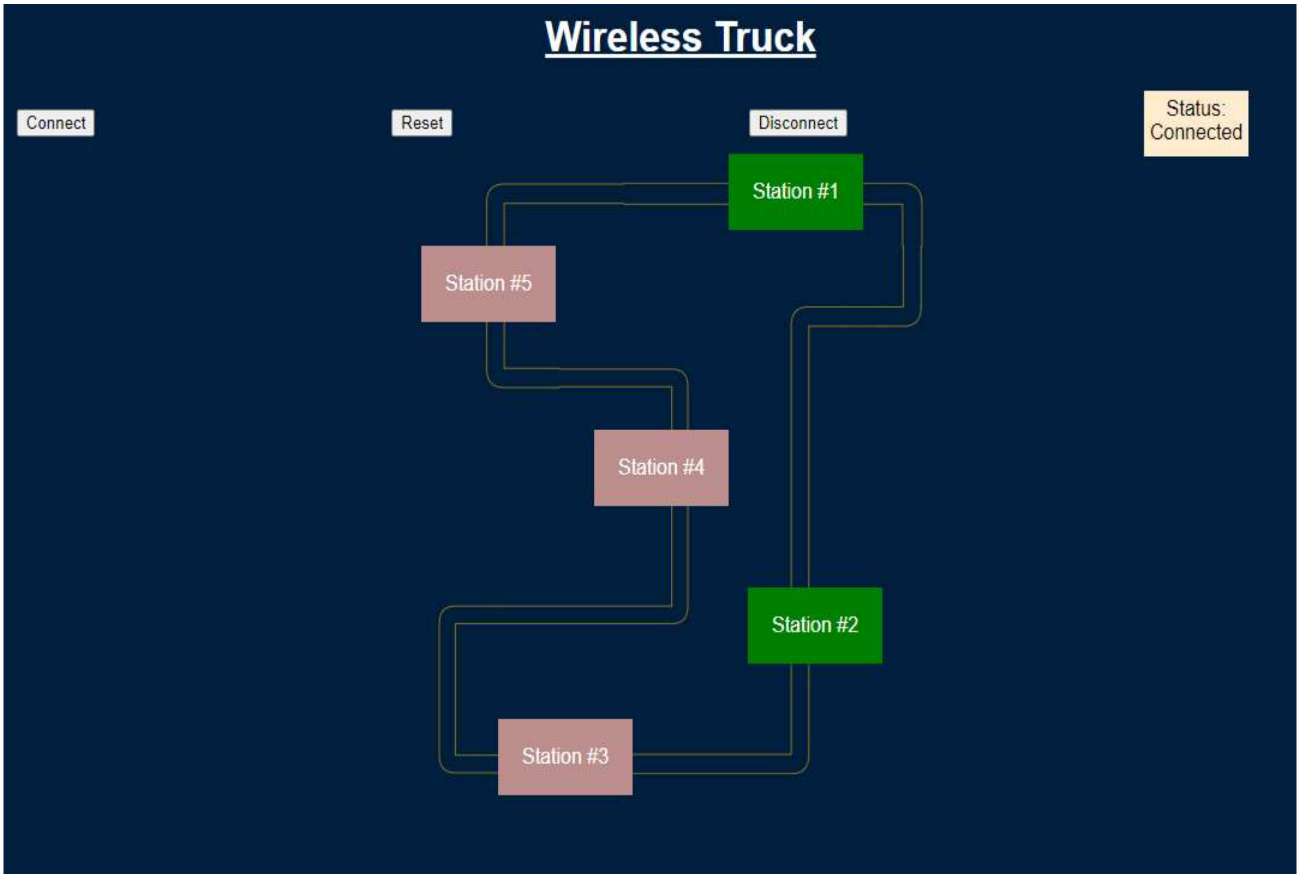

Figure 3-7. Web application for the system

The web application receives a payload of the stations represented by unique IDs from the passive RFID tags from the ESP8266 NodeMCU via MQTT protocol, built on TCP, and updates it on the display.

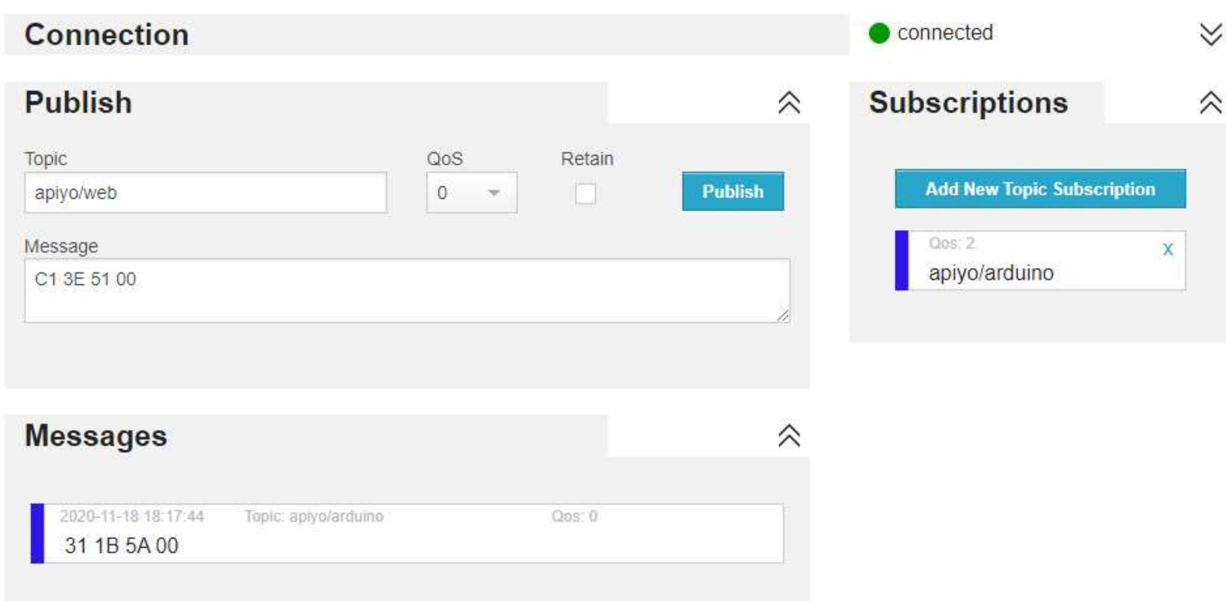

Figure 3-8. MQTT Websockets Client.

Sample code for the webpage is shown below:

```
var clientId = 'apiyo-web';

/*********** public broker ************/
// var host = 'broker.mqttdashboard.com';
var host = 'broker.hivemq.com'; // https://www.hivemq.com/public-mqtt-broker/
var port = 8000;
/*************************************** */

var subTopic = 'apiyo/web';
var pubTopic = 'apiyo/arduino';
var client = new Paho.MQTT.Client(host, port, clientId);

var st1Elem = document.getElementById('st1');
var st2Elem = document.getElementById('st2');
var st3Elem = document.getElementById('st3');
var st4Elem = document.getElementById('st4');
var st5Elem = document.getElementById('st5');

var stations = {
    '12 14 B1 2F': st1Elem,
    'F6 34 F9 25': st2Elem,
    'ED B9 E0 2B': st3Elem,
    '83 87 3B 2E': st4Elem,
    'F6 34 F9 27': st5Elem,
}
var connStatElem = document.getElementById('conn-stat');

function startSession(){
    client.onConnectionLost = onConnectionLost;
    client.onMessageArrived = onMessageArrived;
    client.connect({
        onSuccess: onConnect,
        onFailure: onFailure
    });
    // reset the colors
    // resetColors();
}
```

Figure 3-9. JavaScript code for the webpage.

3.4 Actuator Circuit

A motor driver circuit was implemented using the L298N H-bridge based driver module to control the speed and direction of DC geared motors that steer the pallet truck for precise movement.

The Enable A and Enable B pins are used for enabling and controlling the speed of the motor. A jumper on these pins enable the motors to work at maximum speed. Replacing the jumper by connecting a PWM input to the pins controls the speed of the motors.

The Input 1 and Input 2 pins are used for controlling the rotation direction of motor A while inputs 3 and 4 are for motor B. Using these pins, one controls the switches of the H-Bridge inside the L298N IC. The circuit is powered using a 2200mmAh 11.1V Li-Poly battery.

The schematic for the circuit layout and design files is shown in figure **below:**

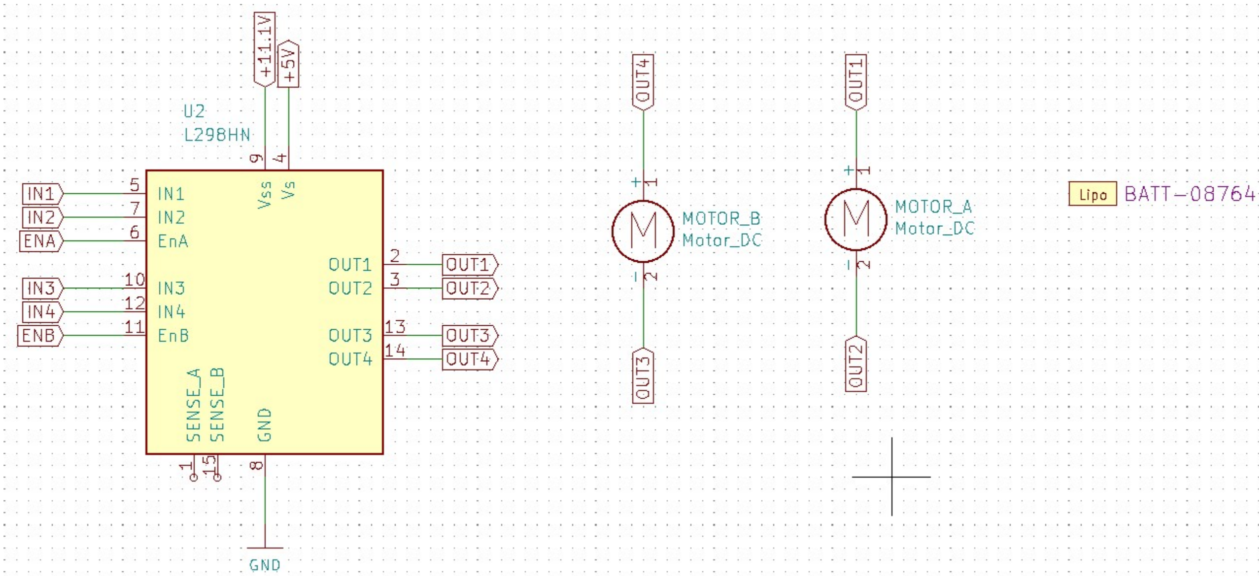

Figure 3-9. L298N interfacing with DC Motors

If input 1 is LOW and input 2 is HIGH the motor moves forward, and vice versa, if input 1 is HIGH and input 2 is LOW the motor moves backward. In case both inputs are similar, either LOW or HIGH the motor stops. The same applies for the inputs 3 and 4 and the motor B.

The control truth table is shown below:

ENA	INPUT 1	INPUT 2	Description
0	N/A	N/A	Motor A OFF
1	0	0	Motor A stops
1	0	1	Motor A ON and moving BACKWARDS
1	1	0	Motor A ON and moving FORWARD
1	1	1	Motor A stops

Table 3-2: L298N Truth Table

The speed of the DC geared motors is controlled by Pulse Width Modulation (PWM) feature of the microcontroller.

Sample code for the motor circuit is shown below:

```
//MOVING MOTORS FOR LINE FOLLOWING
if(digitalRead(right_sensor_pin) == HIGH && digitalRead(left_sensor_pin) == LOW){
  Serial.println("turning right");
  Right();
 }
if(digitalRead(right_sensor_pin) == LOW && digitalRead(left_sensor_pin) == HIGH){
  Serial.println("turning left");
  Left();
}
if(digitalRead(right_sensor_pin) == LOW && digitalRead(left_sensor_pin) == LOW){
  Serial.println("going forward");
  Forward();
}
if(digitalRead(right_sensor_pin) == HIGH && digitalRead(left_sensor_pin) == HIGH)
{
  Serial.println("stopping");
  Stop();

  //analogWrite (motorAspeed, 0);
  //analogWrite (motorBspeed, 0);
  while(true){}
  }
```

Figure 3-10. Sample code for the motor circuit

3.5. System Flowchart.

The flow-chart below in *Figure 3-11* below shows the overall flow of the project:

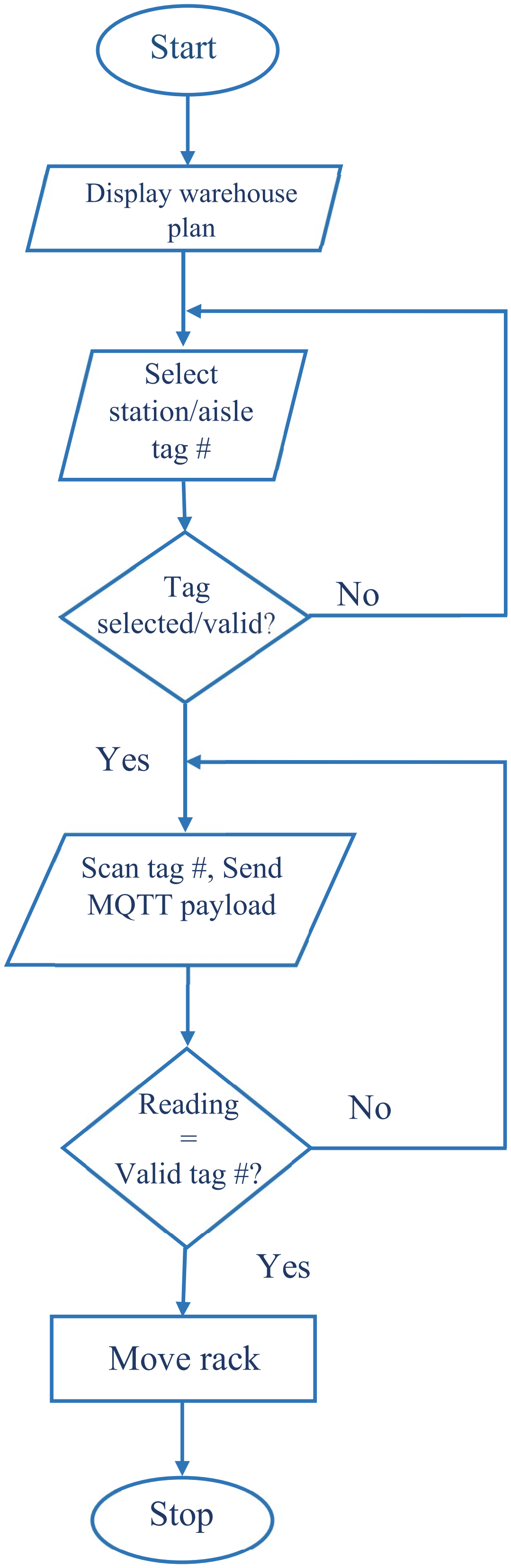

Figure 3-11. Overall flowchart for the pallet truck

3.6. Circuit Design and Schematic.

The system circuit schematic was designed using KiCad software which is a free open source circuit design software.

The figures below show the schematic design:

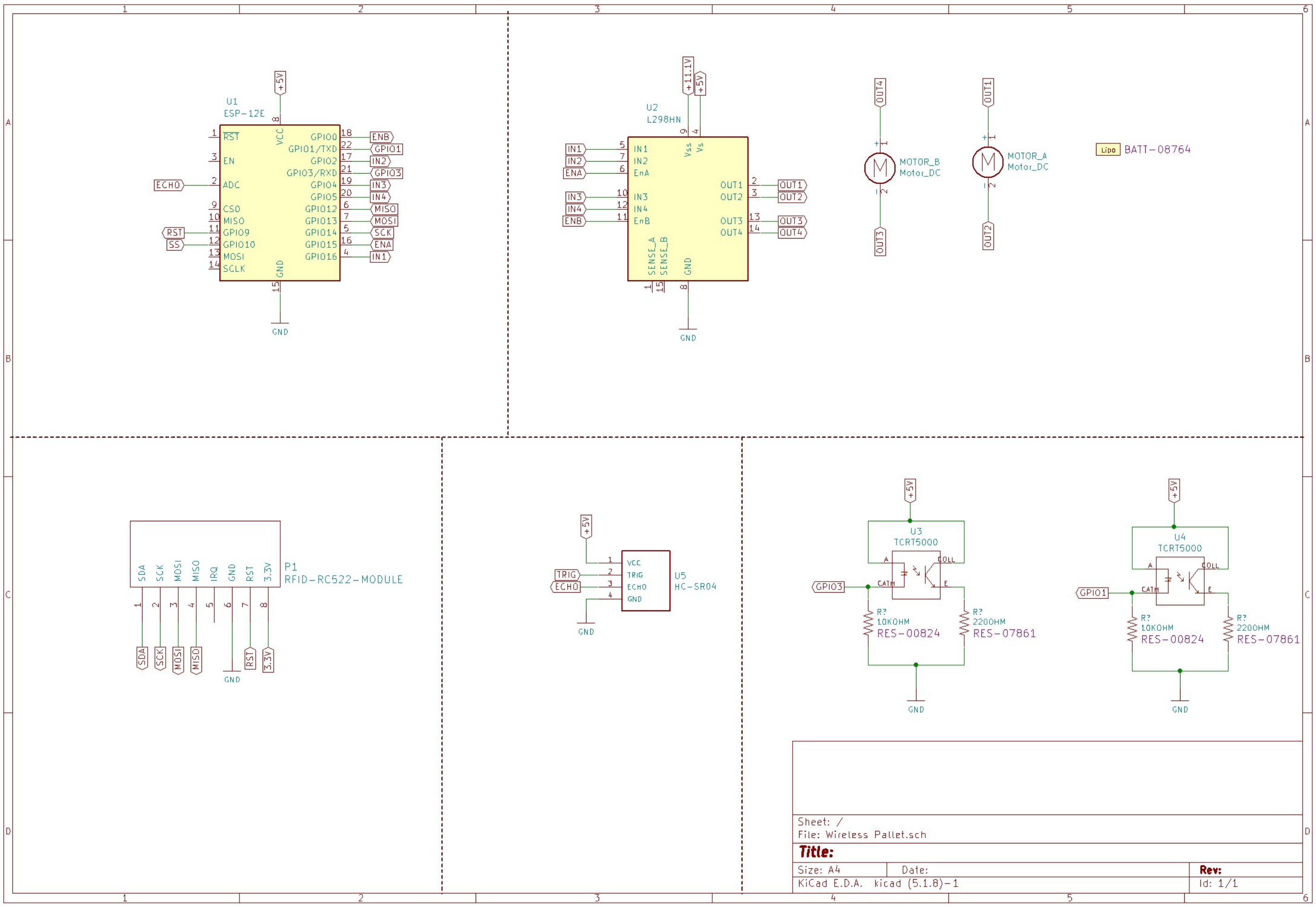

Figure 3-12. KiCad circuit design schematic

CHAPTER FOUR: RESULTS

Upon successful implementation of this project the following were results:

i). A prototype of a pallet truck that can be controlled by sensor and user inputs to automatically navigate through a warehouse was successfully implemented.

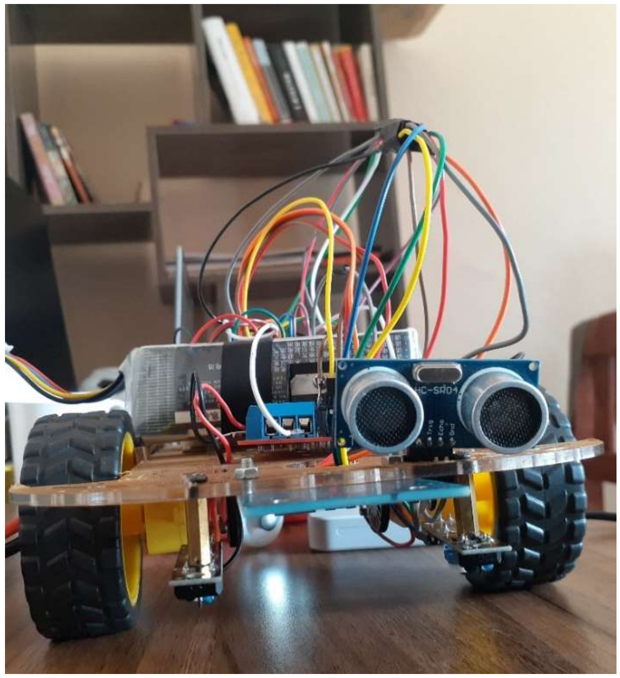

Figure 4-1. Assembled system prototype

ii). A warehouse management system that can track the movement of inventory was successfully implemented.

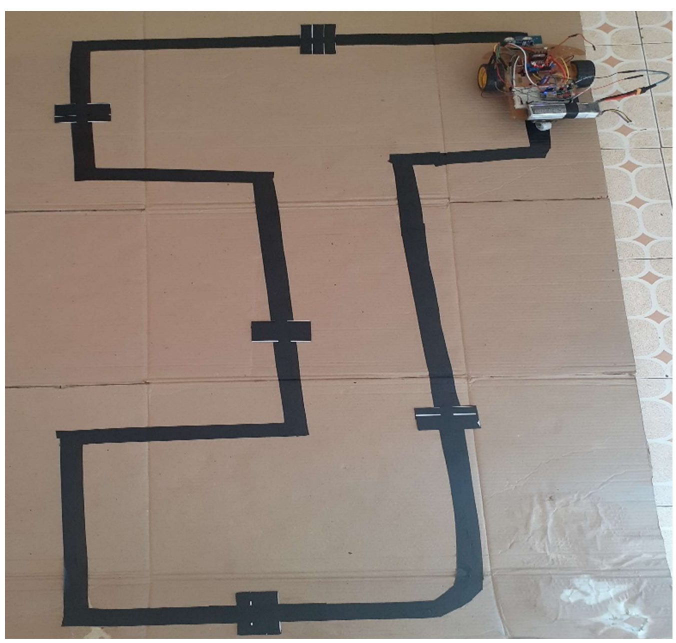

Figure 4-2. System implementation

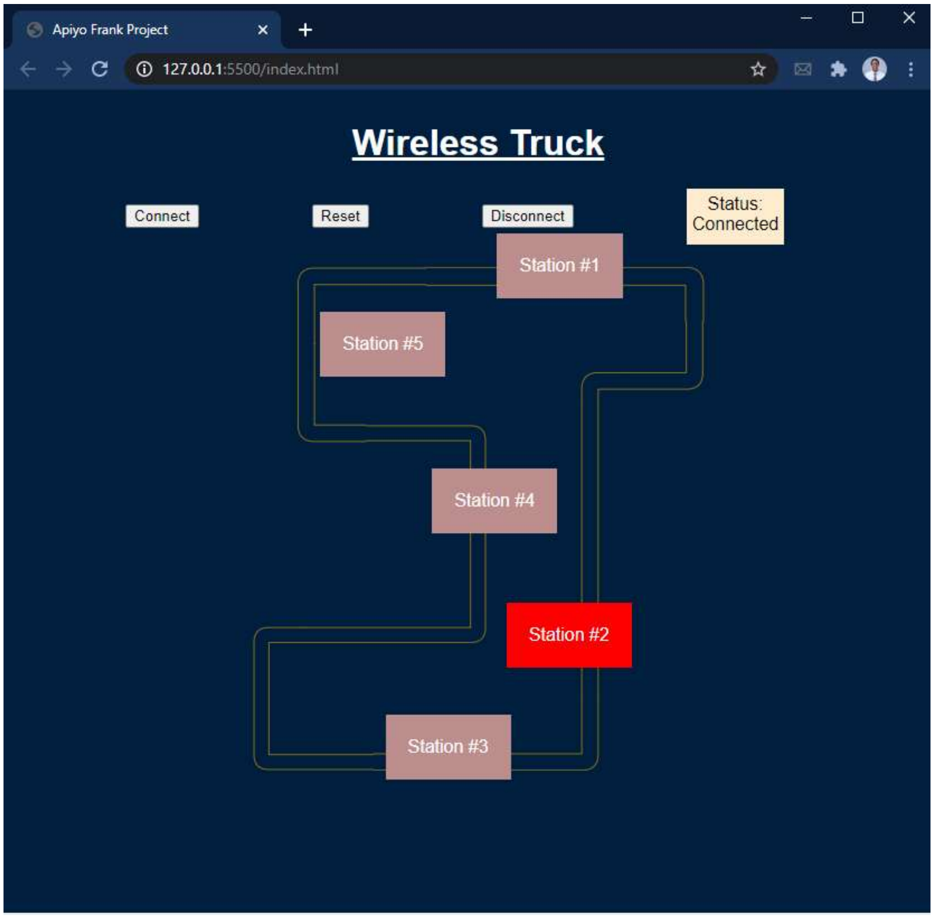

Figure 4-2. Warehouse Management System Web Interface

iii). A cheaper and mobile pallet tracking system than current AS/RS systems was successfully implemented.

Feature	**Wireless Automated Pallet Track**	**Kiva Robots**	**6River Chuck Unit**
Maneuvering	Optical navigation – IR sensors	Cameras, encoders, accelerometers, and rate gyros	LiDAR Sensors
Obstacle Detection	Ultrasonic sensor	Touch sensitive bumpers and infrared sensors	Motion sensors
Payload	**-**	~ 320 kg	90.7 kg
Tasks	Picking	Picking	Put-away, picking, counting, replenishment and sorting
Cost	~ Ksh. 11000*	Developed by Amazon for inhouse use	$400,000 (8 chuck units)

Table 4-1: Comparing Wireless Pallet Truck to other AGVs

CHAPTER FIVE: RESULTS

5.1. Challenges.

i. The design of the schematic on software was challenging due to some components lacking footprints and symbols while finding libraries for specific components was tedious and expensive.

ii. System Complexities during the implementation of live location for the pallet truck meant this was not possible due to expensive industrial sensors and navigation technologies that required very precise calibration, layout mapping and odometry which were not available to me. The other option was GPS which is only accurate to 1.11m which results in grave inaccuracies.

iii. Challenges procuring components due to restrictions and measures imposed during the COVID-19 pandemic. Limited or restricted access to research papers and components normally availed by the Electronics lab due to inaccessibility of the school during the lockdown.

iv. Financial challenges in the acquisition of components and lack of software licenses for simulation software that made the testing and circuit design a huge setback.

v. Challenges in breadboard implementation because of many jumper wires that introduced noise signals. This was a major setback during implementation.

vi. Problem obtaining standard fit components/custom parts that prompted use of alternative components.

vii. Challenges implementing PWM for motor speed control

5.2. Recommendations.

i. Provision of modern and high technology and software for calibration, PCB fabrication, simulation and testing which will improve the quality of the projects and eliminate hinderances to implementation that are currently being experienced.

ii. Implement AI and ML to achieve natural navigation for the AGV

iii. Implement the display on the robot to enable workers to work alongside the robot in the tracking of inventory

iv. Conduct frequent mini presentations to check on progress and ensure students keep track of their objectives and get a constant feedback loop during the implementation phase.

5.3. Conclusions.

The project herewith explored existing technologies in the warehousing industry and sought to explore alternative technologies. The objectives were achieved in the following areas. I was able to develop and implement a sensor circuit using IR sensors which provided driving directions for the pallet truck. The challenge was in the implementation of PWM for motor speed control.

The wireless circuit was implemented successfully, and data could be transmitted between the web server and the robot. A web interface was preferred due to the quality of display. An embedded display would be ideal for workers in a collaborative environment and this is a recommendation for future work in this area.

REFERENCES

[1] M. Chui, J. Manyika and M. Miremadi, "Where machines could replace humans--and where they can't (yet)", *McKinsey & Company*, 2016. [Online]. Available: https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/where-machines-could-replace-humans-and-where-they-cant-yet. [Accessed: 25- Feb- 2020].

[2] A. Dekhne, G. Hastings, J. Murnane and F. Neuhaus, "Automation in logistics: Big opportunity, bigger uncertainty", *McKinsey & Company*, 2019. [Online]. Available: https://www.mckinsey.com/industries/travel-transport-and-logistics/our-insights/automation-in-logistics-big-opportunity-bigger-uncertainty. [Accessed: 25- Feb- 2020].

[3] J. Smith, "From Reindeer to Robots, Automation Set to Deliver This Holiday Season", WSJ, 2019. [Online]. Available: https://www.wsj.com/articles/from-reindeer-to-robots-automation-set-to-deliver-this-holiday-season-1538569800. [Accessed: 13- Jan- 2020].

[4] J. Santagate, "Worldwide Commercial Service Robotics Forecast, 2019–2022", IDC: The premier global market intelligence company, 2020. [Online]. Available: https://www.idc.com/getdoc.jsp?containerId=US44771619. [Accessed: 26- Jan- 2020].

[5] E. Basker, "Change at the Checkout: Tracing the Impact of a Process Innovation", The Journal of Industrial Economics, vol. 63, no. 2, pp. 339-370, 2015. Available: 10.1111/joie.12073.

[6] T. Andersson, "Store of the Future 1Q18, 2018", Styleintelligence, 2018. [Online]. Available: https://www.styleintelligence.com/products/store-of-the-future-1q18-2018. [Accessed: 05- Feb- 2020].

[7] Cheung, Z., Feletto, M., Galante, J., Waters, T., & Centers for Disease Control and Prevention. (2007). Ergonomic guidelines for manual material handling. DHHS (NIOSH) Publication, (2007-131). [Accessed: 21- Jan- 2020].

[8] Gue, K., Akcali, E., Erera, A., Ferrell, B., & Forger, G. (2014). Material Handling & Logistics: US Roadmap. [Accessed: 22- Oct- 2020].

[9] E. Guizzo, "Kiva Systems: Three Engineers, Hundreds of Robots, One Warehouse", IEEE Spectrum: Technology, Engineering, and Science News, 2008. [Online]. Available: https://spectrum.ieee.org/robotics/robotics-software/three-engineers-hundreds-of-robots-one-warehouse. [Accessed: 24- Jan- 2020].

[10] DATASHEET, Atmel®. "8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash." (2015).

[11] MQTT Version 3.1.1 Plus Errata 01, OASIS Standard Incorporating Approved Errata 01. December 2015. Available: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html. [Accessed: 10- November- 2020].

[12] API documentation for Arduino Client Library. Available: https://pubsubclient.knolleary.net/ [Accessed: 16- November- 2020].