

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**“Jnana Sangama", Belagavi, Karnataka, INDIA 2024-2025**

**A Major Project Report on**

**“DESIGN AND DEVELOPMENT OF SMART BOREWELL CHILD RESCUE SYSTEM”**

***Submitted in partial fulfillment of the requirement for the award of the degree of***

**Bachelor of Engineering**

**in**

**Electronics and Communication Engineering**

*Submitted By*

###### NATARAJ K B (1GA20EC103) RAGHAVENDRA C S (1GA20EC144)

**GIRISH S N (1GA21EC407)**

**PRIYA K M (1GA20EC111)**

**Under the Guidance of Dr. RAVI J**

**Professor ,Dept Of ECE, GAT**

##### Department of Electronics and Communication Engineering

**GLOBAL ACADEMY OF TECHNOLOGY**

(An Autonomous Institute Affiliated to VTU)

Rajarajeshwarinagar, Bengaluru - 560 098

**2024 – 2025**

**Department of Electronics and Communication Engineering**

# CERTIFICATE

Certified that the Project Entitled **“DESIGN AND DEVELOPMENT OF SMART BOREWELL CHILD RESCUE SYSTEM”** carried out by **NATARAJ KB (1GA20EC103), RAGHAVENDRA CS (1GA20EC114), GIRISH S N (1GA21EC407), PRIYA K M (1GA20EC111),**bonafide students of Global Academy of Technology, is in partial fulfillment for the award of the **BACHELOR OF ENGINEERING** in Electronics and Communication Engineering from **Visvesvaraya Technological University, Belagavi** during the year **2024-2025**. It is certified that all the corrections/suggestions indicated for Internal Assessment have been incorporated in the report submitted in the department library.

Dr.Ravi J Professor, Project Guide Dept. of ECE GAT,

Bengaluru.

Dr.Manjunatha Reddy H S Professor& HOD Dept.of ECE GAT, Bengaluru.

External Viva

Dr. N. Ranapratap Reddy Principal

GAT, Bengaluru.

Name of the Examiners Signature with date



1. ……………………… ………………………….
2. ……………………… .…………………………

Rajarajeshwarinagar, Bengaluru – 560 098



**Department of Electronics and Communication Engineering**

# DECLARATION

We, **NATARAJ K B** (1GA20EC103), RAGHAVENDRA C S (1GA20EC114),GIRISH S N

**(1GA21EC407),PRIYA K M(1GA20EC111)** students of Eighth Semester B.E, Department of Electronics and Communication Engineering, Global Academy of Technology, Rajarajeshwari Nagar, Bengaluru, declare that the Project Work entitled **“DESIGN AND DEVELOPMENT OF SMART BOREWELL CHILD RESCUE SYSTEM”** has been carried out by us and submitted in partial fulfillment of the course requirements for the award of degree in **Bachelor of Engineering in Electronics and Communication Engineering** from **Visvesvaraya Technological University, Belagavi** during the academic year **2024-25**. The matter embodied in this report has not been submitted to any other university or institution for the award of any other degree.

###### NATARAJ K B (1GA20EC103)

###### RAGHAVENDRA C S (1GA20EC114)

###### **GIRISH S N (1GA21EC407)**

**PRIYA K M (1GA20EC111)**

Place: Bengaluru Date:

**ABSTRACT**

The borewell rescue robotic device for which using ESP 32 is a technology solution designed to improve the safety and efficiency of rescues in cases where people fall into open bore wells, a common problem in rural areas. This system utilizes the ESP32 micro controller and Internet of Things (IOT) technology to monitor borewell, collect data, transmit it to a central platform, and provide real-time alerts, location tracking, and automation for faster and more effective rescue operations. It is energy-efficient and scalable for monitoring multiple borewell, contributing to enhanced safety and potentially saving lives The traditional way to rescue the child is to dig a parallel pit t adjacent to the bore well. This method is difficult, lengthy and also risky to rescue the trapped child. In the proposed method mechanical system moves inside the borewell channel and moves its gripper arm in accordance with the user commands given.

### ACKNOWLEDGEMENT

The fulfillment and rapture that go with the fruitful finishing of any assignment would be inadequate without the specifying the people who made it conceivable, whose steady direction and support delegated the endeavors with success. Although it is not possible to thank all the members who helped for the completion of the Project Work individually, we take this opportunity to express our gratitude to one and all.

We are grateful to management and our institute **GLOBAL ACADEMY OF TECHNOLOGY** with its very ideals and inspiration for having provided us with the facilities, which made this, Project Work a success.

We express our sincere gratitude to **Dr. N. Rana Pratap Reddy,** Principal, Global Academy of Technology for the support and encouragement.

We wish to place on record, our grateful thanks to **Dr. Manjunath Reddy H S**HOD, Department of ECE, Global Academy of Technology, for the constant encouragement provided to us.

We are indebted with a deep sense of gratitude for the constant inspiration, encouragement, timely guidance and valid suggestion given to us by our guide **Dr. RAVI J , Professor** , Department of ECE, Global Academy of Technology.

We are thankful to all the staff members of the department for providing relevant information and helped in different capacities in carrying out this Project Work.

Last, but not least, we owe our debts to our parents, friends and also those who directly or indirectly have helped us to make the Project Work a success.

###### NATARAJ K B (1GA20EC103) RAGHAVENDRA C S (1GA20EC114)

**GIRISH S N (1GA21EC407)**

**PRIYA K M**   **(1GA20EC111)**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Chapter Name** | **Page No.** |
|  | **ABSTRACT** | i |
|  | **ACKNOWLEDGEMENT** | ii |
|  | **LIST OF FIGURES** | vi |
|  | **LIST OF ABBREVIATIONS (if any)** | vii |
| **1** | **INTRODUCTION** |  |
| 1.1 Description | 8 |
| 1.2 Project Outline | 9 |
| **2** | **LITERATURE REVIEW** |  |
|  | 2.1Proposed work and existing literature | 10-11 |
|  | 2.2 Scope of the project | 11-12 |
| **3** | **SYSTEM REQUIREMENTS& SPECIFICATION** |  |
| 3.1Hardware requirements | 13 |
| 3.1.1 Components with description | 14-17 |
| 3.2Software requirements | 18 |
| 3.3Specifications | 19-20 |
| **4** | **DESIGN AND DEVELOPMENT OF SMART BOREWELL CHILD RESCUE SYSTEM** |  |
|  | 4.1 Block diagram | 21-22 |
|  | 4.2 Coding Algorithm | 23-24 |
|  | 4.3 Implementation | 25-26 |
|  | 4.4 Outlook of the Project | 27 |

|  |  |  |
| --- | --- | --- |
|  | 4.5Result | 28-29 |
| **5** | **CONCLUSION** | 30 |
|  | **APPLICATIONS&FUTURE SCOPE** | 31-32 |
|  | **REFERENCES** | 33 |
|  | **APPENDICES** | 34-63 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Description** | **Page. No** |
| Figure1 | Arduino UNO | 13 |
| Figure2 | Liquid Crystal Display | 13 |
| Figure 3 | EM-18 Reader Module | 14 |
| Figure4 | RFID tags | 14 |
| Figure 5 | GSM Module | 14 |
| Figure 6 | Sealed Rechargeable Battery | 15 |
| Figure 7 | LED | 15 |
| Figure 8 | Buzzer | 15 |
| Figure 9 | Push Button | 16 |
| Figure 10 | Switch | 16 |
| Figure 11 | Arduino IDE | 17 |
| Figure 12 | Messaging App | 17 |
| Figure 13 | Block Diagram for EV Installation | 20 |
| Figure 14 | Block Diagram for Billing System | 21 |
| Figure 15 | Billing System | 26 |
| Figure 16 | Trolley | 26 |

**ABBREVIATIONS USED**

|  |  |
| --- | --- |
| RFID | Radio Frequency Identification |
| GSM | Global System for Mobile communication |
| LCD | Liquid Crystal Display |

## Chapter 1

## INDROTUCTION

##### 1.1Description:

The "Design And Development Of Smart Borewell Child Rescue System" The issue of children getting trapped in abandoned bore wells has become increasingly concerning due to the risk of fatalities. Traditional rescue methods involving parallel pit digging are complex, lengthy, and risky. This project proposes a simple yet effective solution using a robotic arm mechanism to rescue trapped children directly from the borewell. The mechanical system moves within the borewell channel and employs a gripper mechanism to safely retrieve the child.The development of surveillance robots has been a growing area of interest due to its applications in various fields such as security, industrial monitoring, and home automation. The surveillance robot using the ESP32-CAM is a project that focuses on developing a low-cost and efficient rescue robot.

Surveillance robots are becoming increasingly popular in modern times due to their ability to monitor and collect information from a remote location. The use of the ESP32 CAM module, which is an integrated camera module with Wi-Fi and Bluetooth connectivity, has made it possible to create surveillance robots that can be controlled remotely.

The ESP32 CAM module is a small-sized camera module that can be easily integrated into a robot. The module provides high-quality images and can be controlled using the ESP32 micro controller. The ESP32 CAM module also has Wi-Fi and Bluetooth connectivity, which allows for easy communication with remote control station or a computer. With the ESP32 CAM module, it is possible to create a surveillance robot that can be controlled remotely from a computer or a mobile device. The robot can be programmed to move around a specific area and capture images or videos of the surroundings. The images and videos can be transmitted wireless to the remote-control station, where they can be viewed and analyzed in real-time.

The use of the ESP32 CAM module in surveillance robots has many advantages, including the ability to capture high quality images and videos, the ability to control the robot remotely, and the ability to transmit data wireless. Additionally, the ESP32 CAM module is easy to use and can be integrated into a robot without requiring any additional hardware.Overall, the ESP32 CAM module is an excellent tool for creating surveillance robots that can be used in a variety of applications, including security, monitoring, and inspection.

**1.2 Project Outline:**

The "Design And Development Of Smart Borewell Child Rescue System" encompasses a comprehensive road map for the rescue child from borewell by measuring distance of child from ground level.

This initiative aims to seamlessly blend state-of-the-art hardware and software, Smart Borewell Child Rescue System is to prevent accidents involving children falling into bore wells and to enhance the efficiency and effectiveness of Rescue operations in case such incidents occur.

The project's foundation rests on a thorough literature review, which illuminates the existing potential challenges in designing and developing smart boards and telemetry systems.

It defines clear objectives and scope, acknowledging potential limitations while emphasizing the transformation how smart boards and telemetry systems integrate with other components of the smart borewell child rescue system.

In subsequent sections, the system's intricate design, hardware and software components, data flow, user interaction, testing procedures, and anticipated future enhancements will be explored, providing a holistic perspective on this cutting-edge retail solution.

**Chapter 2**

**LITERATURE REVIEW**

##### 1.3 Proposed work and existing literature

The proposed project, titled "Design Development of Smart Borewell Child Rescue system," aims to encompasses a thorough exploration of existing research, studies, and technologies pertinent to borewell rescue operations, robotic systems, remote sensing, and safety measures. This involves delving into past borewell rescue efforts to understand common challenges, employed techniques, and lessons learned. Concurrently, an examination of robotic systems tailored for search and rescue operations, particularly in confined spaces like bore wells, aids in identifying key features necessary for successful deployment. Additionally, a review of remote sensing technologies, including RF communication, wireless cameras, and gas sensors, provides insight into real-time monitoring and situational awareness during rescue missions. Integral to the survey is an investigation into safety measures and risk mitigation strategies, focusing on protecting both victims and rescue teams. Moreover, exploring humanitarian engineering principles and societal impacts underscores the importance of developing effective borewell retrieval solutions. Lastly, staying abreast of emerging technologies and innovations, such as drones and AI-based systems, informs potential collaboration opportunities and ensures the project remains at the forefront of advancements in the field.

[1]. The research paper discusses the design and construction of a portable Borewell Rescue Robot aimed at providing a cost-effective, quick, and accurate solution to the common problem of children falling into bore wells. The robot is equipped with a robotic arm, circular disk, airbag, and IR device to ensure the safety and successful rescue of the child.The paper highlights the lack of efficient and reliable instruments for rescuing children stuck in bore wells and proposes the use of the Borewell Rescue Robot as a solution.

[2] The research paper discusses the design and fabrication of a Mini-Size Borewell Rescue Robot to address the critical issue of children falling into abandoned bore wells in India. The robot aims to provide a safe and efficient rescue operation by autonomously navigating through the bore well and retrieving the trapped victim without causing harm. The paper highlights the challenges of current rescue methods and emphasizes the need for a technical solution. It also mentions the use of pneumatic cylinders, grippers, and batteries in the robot's design.

1. The Bore well Rescue Robot project addresses the distressing issue of children falling into uncovered bore holes. The robot is designed to swiftly and economically rescue the trapped child, providing continuous monitoring, necessary supplies for survival, and a safe handling system to ensure the child's well-being. The use of a robotic arm and fordable seat allows for the safe extraction of the child without causing harm, making it an innovative and life-saving solution to a critical problem.

##### **1.4 Research and technical papers:**

##### 2.2.1 Smart Rescue System for Borewell Accidents By- Christopher Adams, Sarah Miller (February 2021 “Journal of Engineering Sciences.”):This paper proposes an implementation introduces a comprehensive smart rescue system that integrates drones for aerial assessments and GPS tracking for precise location identification during bore well accidents.

2.2.2 Wireless Sensor Network for Bore Well Child Safety By-Laura Davis, Michael Johnson

(June 2017 "An Integrated Approach to Borewell Safety: Sensor Network Implementation."):

The goal is to minimize response time and improve the chances of successful child rescues from bore wells.

2.2.3 Io T-Based Smart Borewell Child Rescue System By-A. Kumar, B. Sharma

Singh, (August 2020 "Design and Implementation of IOT-Based Smart Borewell Child Rescue System."):It employs sensors to monitor borewell conditions, and a wireless communication system to transmit real-time data to a central server.

2.2.4 Smart Bore Well Child Rescue System Author(s): By-John Doe, Jane Smith:( "An Integrated Approach to Borewell Safety: Sensor Network Implementation.")The goal is to minimize response time and improve the chances of successful child rescues from bore wells.

**1.5 Scope of the project**

* Develop software and hardware components that automatically scan and track items placed in the trolley, calculate the total cost, and generate an electronic bill.
* Integrate the system with the store's inventory database to ensure real-time updates on product availability and pricing.
* Create a user-friendly interface for shoppers to view the list of scanned items, make adjustments, and finalize their purchases.
* Implement various payment methods, including card payments, mobile wallets, and cash, to accommodate different customer preferences.

I

**Chapter 3**

#### SYSTEM REQUIREMENTS & SPECIFICATION

**3.1:HARDWARE REQUIREMENTS:**

|  |  |  |
| --- | --- | --- |
| **SL.NO** | **COMPONENTS REQUIRED** | **QUANTITY** |
| 1 | ESP32CAM | 1 |
| 2 | MOTOR DRIVER | 1 |
| 3 | FDTI Programmer | 1 |
| 4 | Cables and connectors | As per requirement |
| 5 | Gear Motor | 1 |
| 6 | LED | 2 |
| 7 | Adapter | 1 |
| 8 | GSM Module | 1 |
| 9 | Rechargable battery | 1 |
| 10 | Push button | 2 |
| 11 | Buzzer | 1 |
| 12 | Switch | 1 |
| 13 | Jumper wires | As per requirement |
| 14 | 48v 1000W BLDC Motor | 1 |
| 15 | 48V 1000W controller | 1 |
| 16 | 48V Lithium ion battery | 1 |
| 17 | Throttle | 1 |

###### COMPONENTS WITH DESCRIPTION:

###### **3.1.20 ESP32cam**

The ESP32-CAM shown in Fig.3.1.1 is a small-sized camera module that is based on the ESP 32 micro controller and OV2640 sensor. It is capable of capturing images, streaming video, and performing various image processing tasks. The ESP32CAM module also features Wi-Fi and Bluetooth connectivity, making it ideal for 10T and surveillance applications.

Here are some key features of the ESP32-CAM:

1. ESP32 micro controller

2. OV2640 2 MP camera sensor

3. Wi-Fi and Bluetooth connectivity

4. GPIO pins for interfacing with other devices

5. Micro SD card slot for storage

6. 5 V DC power supply

To use the ESP32-CAM module, you need to have some knowledge of programming, especially with the Arduino IDE, as it is commonly used to program the ESP32-CAM. You will also need to have some basic electronics knowledge, such as how to connect wires and components to the GPIO pins of the module shown



Fig 3 .1: ESP32CAM

**3.1.21:MOTOR DRIVER**

The L298N is a popular motor driver IC (integrated circuit) that can control the direction and speed of DC motors and stepper motors. It can handle up to 2 amps of continuous current per channel, and has a wide input voltage range of 5V to 35V.

The L298N has two H-bridge circuits, which are used to control the direction of the motor. Each H-bridge consists of four transistors that can be controlled independently. By switching these transistors on and off in the correct sequence, the motor can be driven in either direction.

To use the L298N, you need to connect it to your micro controller or other control circuitry. There are several pins on the L298N shown in Fig 3.2.1 that you need to connect to:

You can connect these to digital output pins on your micro controller to enable

or disable the motor channels. IN 1, IN2, IN3, and IN4: These are the control pins for the H-bridge circuits.

By setting the correct combination of high and low signals on these pins, you can

control the direction and speed of the motor. OUTI, OUT2, OUT3, and OUT4: These are the output pins that connect to the motor. To connect the L298N to a motor, you need to connect the motor to the OUT

pins, and also connect the motor power supply to the L298N's V+ and GND pins.

It's important to note that the L298N can get quite hot when driving high-current motors, so you should use a heat sink to dissipate the heat. You should also be careful not to exceed the maximum current ratings of the L298N, or you may damage the IC

To use the L298N, you need to connect it to your micro controller or other control circuitry. There are several pins on the L298N shown in Fig 3.2.1 that you need to connect to:

In this LCD each character is displayed in 5x7 pixel matrix.

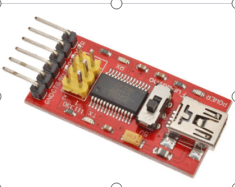
**Fig 2:** **MOTOR DRIVER**

**3.1.22 FDTI programmer**

FTDI stands for Future Technology Devices International, and an FTDI programmer is a device used to program and communicate with electronic devices that use FTDI chips for their USB communication. An FTDI programmer in Fig 3.3.1 typically consists of a USB-to serial converter chip from FTDI and a connector that can be used to interface with the target device. The programmer provides a simple and reliable way to communicate with the device, allowing for the transfer of data and programming instructions between the device and a computer. FTDI programmers are commonly used to program and debug micro controllers, such as those based on the atmal AVR and Microchip PIC architectures. They

are also used in other applications where a USB-to-serial interface is needed, such as in GPS receivers, data loggers, and other electronic devices. In addition to providing a USB-to-serial interface, FTDI programmers often include additional features such as voltage regulation, programmable clock sources, and other features that make them useful for a wide range of application.

**FDTI programmer**



Overall, FTDI programmers are an essential tool for many electronics designers anddevelopers providing a reliable and easy-to-use method for programming and communicating with a wide range of electronic devices.

i) Pin configuration of FDTI Programmer

FTDI chips and programmers typically use a 6-pin connector called a "FTDI

cable" or "FTDI header". The pinout of the FTDI cable or header is as follows:

• DTR (Data Terminal Ready)

• RXD (Receive Data)

• TXD (Transmit Data)

• VCC (Voltage Supply)

• CTS (Clear to send)

• GND (Ground)

**Description of each pin:** This pin is used to reset the microcontroller or target device. The programmer sends a signal to this pin to reset the device before uploading new code.

RXD - This pin is used to receive data from the target device. Data is transmitted from the target device to the programmer through this pin.

TXD - This pin is used to transmit data to the target device. Data is transmitted from the programmer to the target device through this pin.

VCC - This pin provides power to the target device. The voltage level can vary depending on the programmer and the target device, but it's typically 3.3 V or 5V.

CTS - This pin is used for flow control. It tells the target device when it's safe to

transmit data.

GND - This pin provides a common ground reference for the programmer and

the target device.

It's important to note that the pinout may vary depending on the specific FTDI programmer or cable being used. It's always a good idea to consult the documentation for the specific device you're using to ensure you're connecting it correctly.

**3.4 The interface between esp32 cam and FDTI programmer**

The ESP32-CAM is a small camera module based on the ESP 32 micro controller, which includes Wi-Fi and Bluetooth connectivity. The module features an FTDI chip that allows it to communicate with a computer through a USB port using an FTDI programmer.

To interface an ESP32-CAM with an FTDI programmer, you need to connect the FTDI programmer to the module's UART pins. Here's how to do it:

• Connect the FTDI programmer's GND pin to the ESP32-CAM's GND pin.

• Connect the FTDI programmer's TX pin to the ESP32-CAM's RX pin.

• Connect the FTDI programmer's RX pin to the ESP32-CAM's TX pin.

• Connect the FTDI programmer's DTR pin to the ESP32-CAM's GPIOO pin.

• Connect the FTDI programmer's VCC pin to the ESP32-CAM's 3.3V pin. Once the connections are made, you can use a program like the Arduino IDE or the ESP-IDF (ESP32 10T Development Framework) to upload code to the ESP32CAM module. To do this, you'll need to put the module in programming

mode by pulling its GPIOO pin low while resetting it. The FTDI programmer's DTR pin can be used to reset the module automatically by pulling the GPIOO pin low for a short time when programming begins.

It's worth noting that the specific pin out and programming process may vary depending on the ESP32CAM module and FTDI programmer you're using. Always consult the documentation for your specific hardware to ensure you're connecting it correctly and following the appropriate programming steps.

**3.5 Interface between esp32 cam and 1298N**

The L298N is a dual H-bridge motor driver that can be used to control two DC motors or a single bipolar stepper motor. It can also be used to drive other types of loads that require bidirectional current control. To interface an ESP 32-CAM with an L298N, you can use the following steps: Connect the ESP32-CAM's 5V pin to the L298N's VCC pin. Connect the ESP32-CAM's GND pin to the L298N's GND pin. Connect the ESP32-CAM's GPIO pins to the L298N's control pins as

follows:

i. GPIO x to INI

ii. GPIO y to IN2

iii. GPIO z to ENA

The specific GPIO pins used will depend on the code you're using to control the L298N. For example, if you're using the Arduino IDE, you can use the pin-mode() and digital-Write() functions to control the GPIO pins Connect the L298N's motor terminals to the DC motor or bipolar stepper motor you want to

control.

**3.6 The power supply to be provided**

The power supply required for an ESP32-CAM and an L298N motor driver will depend on the specific motors you're using and their voltage and current requirements. However, here are some general guidelines to help you determine the appropriate power supply:

The ESP32-CAM can be powered by a 5 V power supply. The recommended operating voltage range is 2.7 V to 3.6V, but it can be powered up to 5.5 V. You can use a USB cable connected to a computer or a USB power adapter to power the ESP32-CAM.

The L298N motor driver can be powered by a separate power supply from the ESP32-CAM. The recommended operating voltage range is 5V to 35 V, and the maximum current per channel is 2A. You should choose a power supply with a voltage and current rating that is appropriate for the motors you're using.

If you're using a high-power motor, you may need to use an external power supply with a higher voltage and current rating to ensure the motor operates correctly. In this case, you'll need to ensure that the ground of the external power supply is connected to the ground of the ESP32-CAM and L298N to ensure they share a common ground.

Be sure to check the power consumption of your circuit and the power rating of your power supply to ensure it can supply enough power to all components.

It's important to note that incorrect power supply connections or voltage/current ratings can damage the ESP32-CAM and L298N or even cause a safety hazard. Always refer to the data sheets of your components and consult an expert if you're unsure about the appropriate power supply.

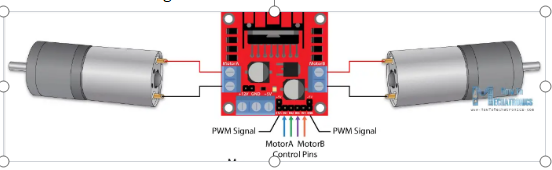
**3.1.23 : GEAR MOTOR**

A gear motor is a type of motor that incorporates a gear head, which is a gear assembly that is designed to reduce the speed of the motor while increasing its torque. Gear motors are widely used in a variety of industrial, commercial, and consumer applications that require precise and controlled movement.

The motor in a gear motor can be an AC motor or a DC motor, and the gear head can consist of various types of gears, such as spur gears, worm gears, helical gears, or planetary gears. The gear head is typically attached to the motor shaft and is designed to convert the high-speed, low torque output of the motor into a lower-speed, higher-torque output.

Gear motors are used in many applications, such as robotics, conveyor systems, industrial machinery, automotive equipment, and many more. They are preferred over regular motors because they provide greater control, accuracy, and efficiency. In addition, gear motors are generally more reliable and have a longer lifespan than regular motors.

**3.7. Interface between gear motor and L298N**



The L298N is a dual H-bridge motor driver IC that can be used to drive DC motors, stepper motors, and other types of motors. A gear motor is a type of motor that has a gearbox attached to it, which allows for higher torque and slower speed.

To interface a gear motor with the L298N, you will need to connect the motor to the output pins of the L298N and provide power to the L298N to control the motor. Here are the steps you can follow:

Connect the motor to the output pins of the L298N. The L298N has two output channels, so you can connect one motor to each channel. Each output channel has two pins: an enable pin and two direction control pins. Connect the motor leads to the two output pins of one channel, and repeat for the other channel if you are using two motors.

Provide power to the L298N. The L298N requires a separate power supply to drive the motors. Connect a 5 V or 12V power supply to the VCC pin of the L298N, and connect the ground of the power supply to the ground pin of the

Connect the control signals to the L298N. The L298N has two input pins for each motor channel: an enable pin and two direction control pins. Connect the enable pin to a PWM output pin on your micro controller or other control circuit. The direction control pins can be connected to any digital output pins on your micro controller or control circuit. Write code to control the motor. To control the motor, you will need to write code to generate PWM signals for the enable pins and set the direction control pins to control the direction of the motor. The specific code will depend on your micro controller or control circuit, and the programming language you are using.

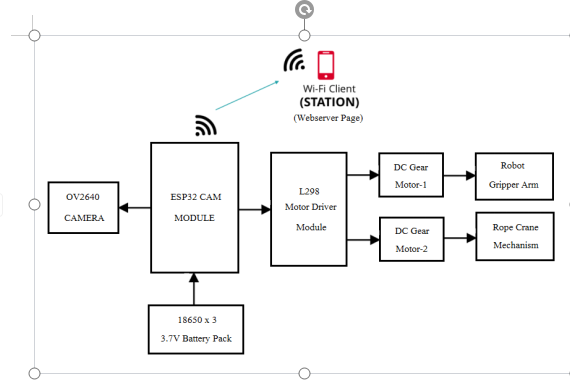
Overall, interfacing a gear motor with the L298N is a fairly simple process that involves connecting the motor to the output pins of the L298N and providing power and control signals to the L298N. With the right code, you can control the speed and direction of the motor to meet your specific needs.

### CHAPTER 4

###### 

**SYSTEM DESIGN OF “DESIGN AND DEVELOPMENT OF SMART BOREWELL CHILD RESCUE SYSTEM”**

**4.0: Block Diagram**



**4.1 BLOCK DIAGRAM DESCRIPTION**

**1 .Power Supply**

The power supply block provides the necessary voltage and current to all the components of the system. Here a 18650 3.7 battery pack is used. It typically includes a power source, voltage regulators, and any necessary conditioning circuitry to ensure stable power delivery.

**2ESP32 Module**

The ESP32-CAM is a small size, low power consumption camera module based on ESP32. It comes with an OV2640 camera and provides onboard TF card slot.

The system can be controlled through a web interface hosted on the ESP32-CAM board. The web interface allows the user to control the robot's movement, view live video streams, and take snapshots of the video feed.

**3.WiFi Client**

WiFi client is a web server page which connects to the ESP32 board through the server IP address and displays real time video from the ESP cameras. And also sends commands to the ESP board to control the motors of the robot rescue mechanism.

**4. L298N Motor Driver Module**

The L298N has two H-bridge circuits, which are used to control the direction of the motor. Each H-bridge consists of four transistors that can be controlled independently. By switching these transistors on and off in the correct sequence, the motor can be driven in either direction.

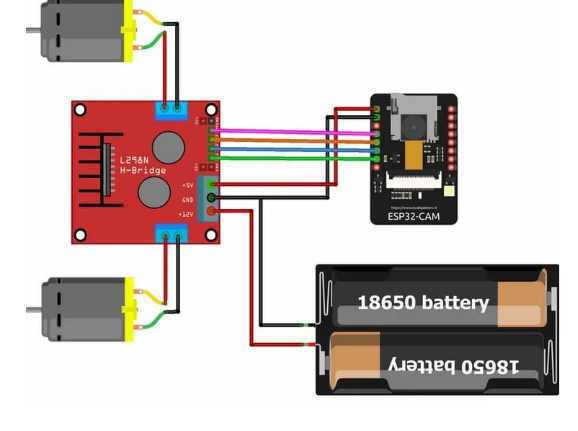
1. **Robotic Gripper Arm**

A robotic gripper arm, also known as a robot manipulator, is a mechanical device designed to imitate the movements and functions of a human arm. It can hold any objects by using open and closing of the gripper arm

1. **Rope Crane Mechanism**

The entire robotic mechanism is suspended through the rope crane mechanism. It uses a pulley attached to the shaft of robot motor which winds and unwinds the rope. By rotation of the pulley the rope can move the robot arm upward or downward.

**CIRCUIT DIAGRAM**



**CIRCUIT Description**

The rescue robot will be controlled via Wi-Fi using your ESP32-CAM. We’ll create a web-based interface to control the robot, that can be accessed in any device inside your local network. The web page also shows a video streaming of what the robot “sees”.

ESP32-camera module works on Wi-Fi. So it can be given commands wireless via IP address. As the command is given from an internet browser with the same IP address, the ESP32 then gives the command to the motor driver.

**Uploading Code to ESP32-CAM Module**

* Plug in the camera to the connector of ESP32-CAM board and close the pin.
* Programming of ESP32-CAM is done via CP2102 USB to TTL UART serial converter Module. First connect the USB to TTL module with ESP32-CAM and short IOO pin of ESP32-CAM to GND according to diagram.
* Open files menu in Arduino IDE. and then open preference and paste the link in additional boards managers URL and press OK:
* After that install the ESP32 library from include library menu.
* Open the tools menu and make changes to parameters according to shown above in images.
* In the code give name of the access point to be connected and SSID of that access point

L298N motor driver is perfect for driving DC motors and Stepper motors. It can control 4 DC motors or 2 DC motors with direction and speed control. This motor driver is perfect for robotics projects and perfect for controlling motors from micro controllers. Perfect for driving DC stepper motors for micro line-follower robots, robot arms, etc.

To keep the circuitry simple, we’ll power the robot (motors) and the ESP32 using the same power source. We used 18650 batteries which is a single cell, compact and powerful cell with a 2200 mAh capacity. It is very convenient to fulfill the 3.7 Volt requirement with this cell. The battery terminals can use in any compatible battery adapter/holder or they can be permanently soldered to our application's power source wires. Here 4 x 3.7v batteries are connected in series which gives a output 12v.

After uploading code disconnect USB to TTL Module completely from ESP32CAMand connect jumper wires to L298N module as shown in diagram :Now for powering the car we will use 12v Li-ion battery, connect accordingly

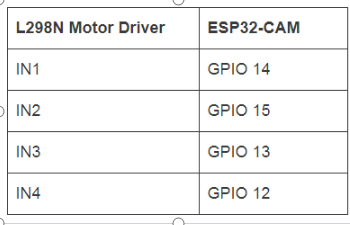
+Ve of battery to ---- +12 of L298N

-Ve of battery to ----- GND of L298N

For powering up the ESP32-CAM module we will use L298N's +5v as source to do this Place the power jumper which is left on board.

**Connection of L298N**

The following table shows the connections between the ESP32-CAM and the L298N Motor Driver.



Since we are not going to control the speed of car and will run at max speed place the ENA and ENB jumpers in place.

After everything is ready and powered up. The ESP-32 will connect to Wi-Fi AP provided by you to it in the code.Open your mobile hotpots setting and turn hotpots on. Wait until ESP-32 to connect, then open the internet browser and enter the IP address provided to you prior in the uploading part.

**CHAPTER-5**

**HARDWARE DETAILS**

**ESP32-CAM MODULE**



The ESP32CAM is a tiny module based on ESP32 chip and OV2640. You can even program the ESP32CAM through the ESP-ID by installing the ESP32 Core

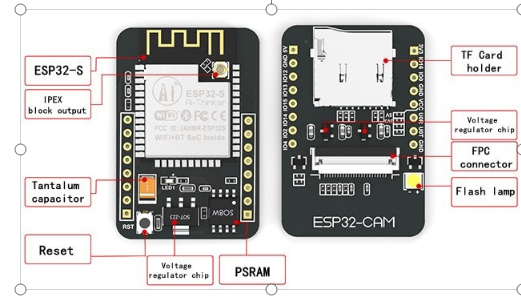
The ESP32CAM equips the ESP32 with everything necessary to program, run and develop on the wonder chip. Additionally, the board reserved the MPU6050,BME280 and an analog MIC.

ESP integrates WiFi, traditional Bluetooth and BLE Beacon, with 2 high-performance 32-bit LX6 CPU, 7-stage pipeline architecture, main frequency adjustment range 80MHz to 240MHz, on-chip sensor, Hall sensor, temperature sensor, etc.

Fully compliant with WiFi 802.11b/g/n/e/i and Bluetooth 4.2 standards, it can be used as a master mode to build an independent network controller, or as a slave to other host MCUs to add networking capabilities to existing devices.

ESP32-CAM can be widely used in various IoT applications. It is suitable for home smart devices, industrial wireless control, wireless monitoring, QR wireless identification, wireless positioning system signals and other IoT applications. It is an ideal solution for IoT applications.

The ESP32-CAM is a very small camera module with the ESP32-S chip that costs approximately $10. Besides the OV2640 camera, and several GPIOs to connect peripherals, it also features a micro SD card slot that can be useful to store images taken with the camera or to store files to serve to clients.



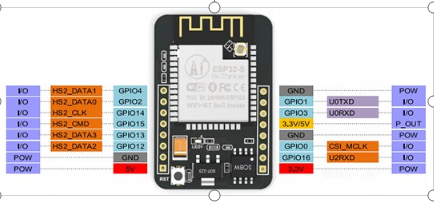
The ESP32-CAM doesn’t come with a USB connector, so you need an FTDI programmer to upload code through the U0R and U0T pins (serial pins).

**Features**

Here is a list with the ESP32-CAM features:

* The smallest 802.11b/g/n Wi-Fi BT Soc module
* Low power 32-bit CPU,can also serve the application processor
* Up to 160MHz clock speed, summary computing power up to 600 DMIPS
* Built-in 520 KB SRAM, external 4MPSRAM
* Supports UART/SPI/I2C/PWM/ADC/DAC
* Support OV2640 and OV7670 cameras, built-in flash lamp
* Support image WiFi upload
* Support TF card
* Supports multiple sleep modes
* Embedded Wipe and Freepost
* Supports STA/AP/STA+AP operation mode
* Support Smart Configure/Air Kiss technology
* Support for serial port local and remote firmware upgrades (FOTA)

**ESP32-CAM Pin out**



The following figure shows the ESP32-CAM pin out (AI-Thinker module).There are three GND pins and two pins for power: either 3.3V or 5V.GPIO 1 and GPIO 3 are the serial pins. You need these pins to upload code to your board. Additionally, GPIO 0 also plays an important role, since it determines whether the ESP32 is in flashing mode or not. When GPIO 0 is connected to GND, the ESP32 is in flashing mode.

The following pins are internally connected to the micro SD card reader:

GPIO 14: CLK

GPIO 15: CMD

GPIO 2: Data 0

GPIO 4: Data 1

GPIO 12: Data 2

GPIO 13: Data 3

**Power Pins:**  There are two power pins: 5V and 3V3. The ESP32-CAM can be powered via the 3.3V or 5V pins. Since many users have reported problems when powering the device with 3.3V, it is advised that the ESP32-CAM always be powered via the 5V pin. The VCC pin normally outputs 3.3V from the on-board voltage regulator. It can, however, be configured to output 5V by using the Zero-ohm link near the VCC pin. GND is the ground pin.

**GPIO Pins**: The ESP32-S chip has 32 GPIO pins in total, but because many of them are used internally for the camera and the PSRAM, the ESP32-CAM only has 10 GPIO pins available. These pins can be assigned a variety of peripheral duties, such as UART, SPI, ADC, and Touch.

**UART Pins**: The ESP32-S chip actually has two UART interfaces, UART0 and UART2. However, only the RX pin (GPIO 16) of UART2 is broken out, making UART0 the only usable UART on the ESP32-CAM (GPIO 1 and GPIO 3). Also, because the ESP32-CAM lacks a USB port, these pins must be used for flashing as well as connecting to UART-devices such as GPS, fingerprint sensors, distance sensors, and so on.

**Micro SD** **Card Pins**: Micros D Card Pins are used for interfacing the micro SD card. If you aren’t using a micros D card, you can use these pins as regular inputs and outputs.

**ADC Pins**: On the ESP32-CAM, only ADC2 pins are broken out. However, because ADC2 pins are used internally by the WiFi driver, they cannot be used when Wi-Fi is enabled.

**SPI Pins**: The ESP32-CAM features only one SPI (VSPI) in slave and master modes.

**PWM Pins**: The ESP32-CAM has 10 channels (all GPIO pins) of PWM pins controlled by a PWM controller. The PWM output can be used for driving digital motors and LED's.

**ESP32-CAM Hardware Overview**

The heart of the ESP32-CAM is an ESP32-S System-on-Chip (Soc) from AI-Thinker. Being an Soc, the ESP32-S chip contains an entire computer—the microprocessor, RAM, storage, and peripherals—on a single chip. While the chip’s capabilities are quite impressive, the ESP32-CAM development board adds even more features to the mix. Let’s take a look at each component one by one.

**The ESP32-S Processor**

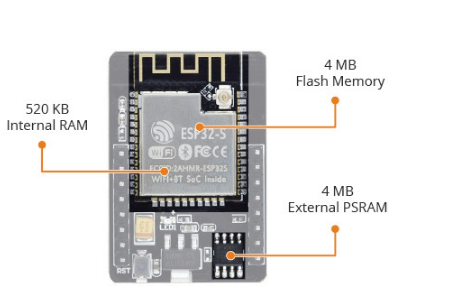
The ESP32-CAM equips the ESP32-S surface-mount printed circuit board module from Ai-Thinker. It is equivalent to Espresso's ESP-WROOM-32 module (same form factor and general specifications).



The ESP32-S contains a Ten silica Xtensa® LX6 microprocessor with two 32-bit cores operating at a staggering 240 MHz! This is what makes the ESP32-S suitable for intensive tasks like video processing, facial recognition, and even artificial intelligence.

**The Memory**

Memory is paramount for complex tasks, so the ESP32-S has a full 520 kilobytes of internal RAM, which resides on the same die as the rest of the chip’s components.

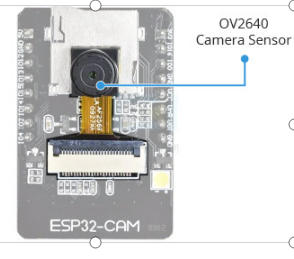


It may be inadequate for RAM-intensive tasks, so ESP32-CAM includes 4 MB of external PSRAM (Pseudo-Static RAM) to expand the memory capacity. This is plenty of RAM, especially for intensive audio or graphics processing.

All these features amount to nothing if you don’t have enough storage for your programs and data. The ESP32-S chip shines here as well, as it contains 4 MB of on-chip flash memory.

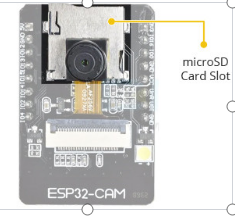
**The Camera**

The OV2640 camera sensor on the ESP32-CAM is what sets it apart from other ESP32 development boards and makes it ideal for use in video projects like a video doorbell or nanny cam.

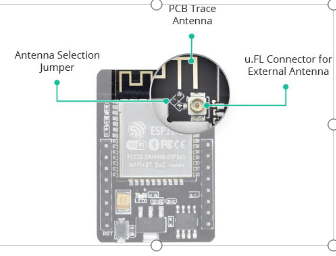
.

**The Storage**

The addition of a microSD card slot on the ESP32-CAM is a nice bonus. This allows for limitless expansion, making it a great little board for data loggers or image capture.



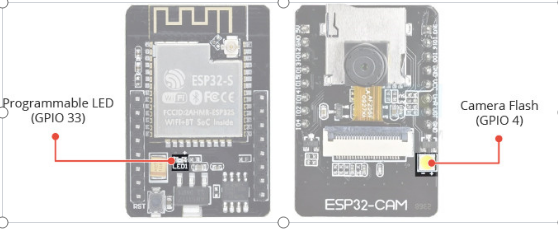
**The Antenna**



The ESP32-CAM comes with an on-board PCB trace antenna as well as a u.FL connector for connecting an external antenna. An Antenna Selection jumper (zero-ohm resistor) allows you to choose between the two options.

**LED's**

The ESP32-CAM has a white square LED. It is intended to be used as a camera flash, but it can also be used for general illumination.



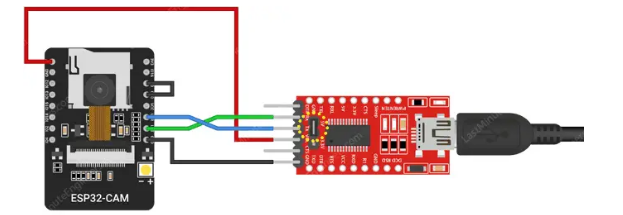
There is a small red LED on the back that can be used as a status indicator. It is user-programmable and connected to GPIO33.

**Programming the ESP32-CAM**

Programming the ESP32-CAM can be a bit of a pain as it lacks a built-in USB port. Because of that design decision, users require additional hardware in order to upload programs from the Arduino IDE. None of that is terribly complex, but it is inconvenient.

To program this device, you’ll need either a USB-to-serial adapter (an FTDI adapter) or an ESP32-CAM-MB programmer adapter.

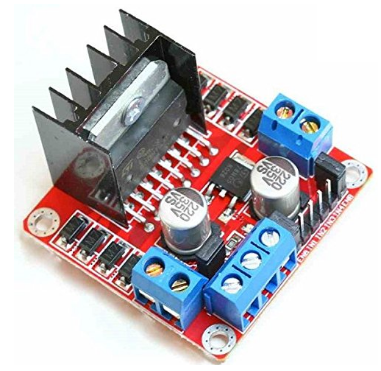
**Using the FTDI Adapter**



If you’ve decided to use the FTDI adapter, here’s how you connect it to the ESP32-CAM module.

Many FTDI programmers have a jumper that lets you choose between 3.3V and 5V. As we are powering the ESP32-CAM with 5V, make sure the jumper is set to 5V.

**L298N DC MOTOR DRIVER MODULE**



**Figure: L298N DC Motor Driver Module**

This L298N Motor Driver Module is a high power motor driver module for driving DC and Stepper Motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator. L298N Module can control up to 4 DC motors, or 2 DC motors with directional and speed control.

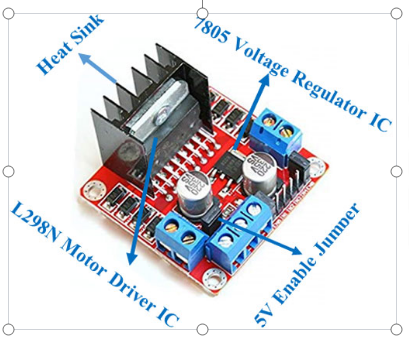
**Features & Specifications**

* Driver Model: L298N 2A
* Driver Chip: Double H Bridge L298N
* Motor Supply Voltage (Maximum): 46V
* Motor Supply Current (Maximum): 2A
* Logic Voltage: 5V
* Driver Voltage: 5-35V
* Driver Current:2A
* Logical Current:0-36mA

**Description of L298N Module**

The L298N Motor Driver module consists of an L298 Motor Driver IC, 78M05 Voltage Regulator, resistors, capacitor, Power LED, 5V jumper in an integrated circuit

**Figure: Pin Description of L298N Module**

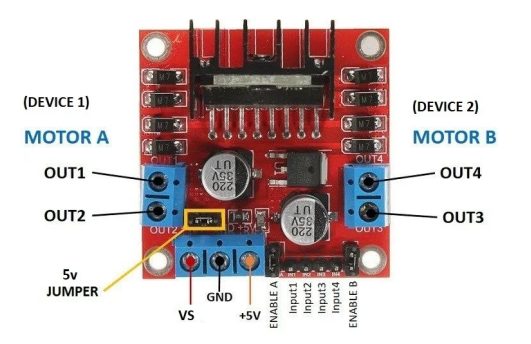


78M05 Voltage regulator will be enabled only when the jumper is placed. When the power supply is less than or equal to 12V, then the internal circuitry will be powered by the voltage regulator and the 5V pin can be used as an output pin to power the micro controller. The jumper should not be placed when the power supply is greater than 12V and separate 5V should be given through 5V terminal to power the internal circuitry.

ENA & ENB pins are speed control pins for Motor A and Motor B while IN1& IN2 and IN3 & IN4 are direction control pins for Motor A and Motor B.

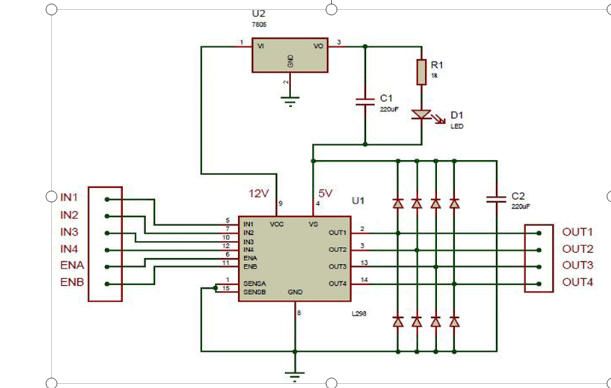
**PIN DESCRIPTION OF L298N MOTOR DRIVER**

**Figure: Pin Description of L298N Motor Driver**



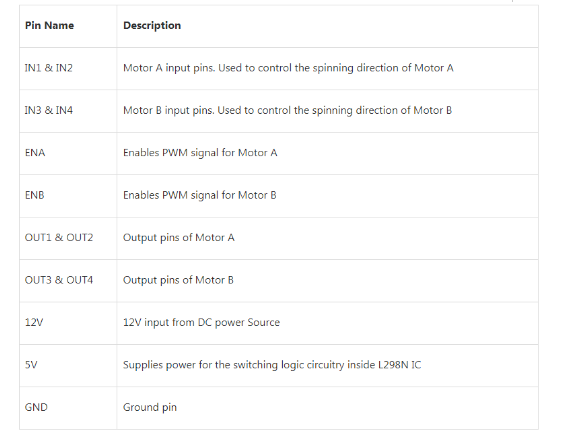
**INTERNAL CIRCUIT DIAGRAM OF L298N MOTOR DRIVER**

Internal circuit diagram of L298N Motor Driver module is given below:



**L298N MODULE PINOUT CONFIGURATION**

**Table: L298N Module Pin out Configuration**



**Power Pins**

* The L298N motor driver module is powered through 3-pin 3.5mm-pitch screw terminal.
* The L298N motor driver actually has two input power pins – VS and VSS.
* VS pin gives power to the internal H-Bridge of the IC to drive the motors. You can connect an input voltage anywhere between 5 to 12V to this pin.
* VSS is used to drive the logic circuitry inside the L298N IC which can be 5 to 7V.
* GND is the common ground pin.

**Output Pins**

The L298N motor driver’s output channels OUT1 and OUT2 for motor A and OUT3 and OUT4 for motor B are broken out to the edge of the module with two 3.5mm-pitch screw terminals. You can connect two 5-12V DC motors to these terminals.

Each channel of the module can deliver up to 2A to the DC motor. However the amount of current supplied to the motor depends on the power supply of the system.

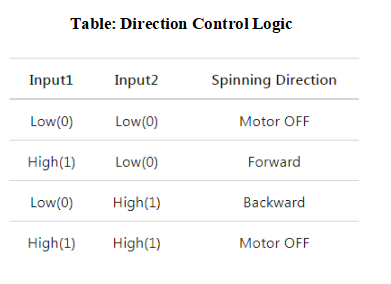
##### 

**Direction Control Pins**

By using the direction control pins, you can control whether the motor rotates forward or backward. These pins actually control the switches of the H-Bridge circuit inside the L298N chip.

The module has two direction control pins for each channel. The IN1 and IN2 pins control the spinning direction of motor A; While IN3 and IN4 control the spinning direction of motor B.

The spinning direction of the motor can be controlled by applying logic HIGH (5V) or logic LOW (Ground) to these inputs. The chart below shows how this is done.

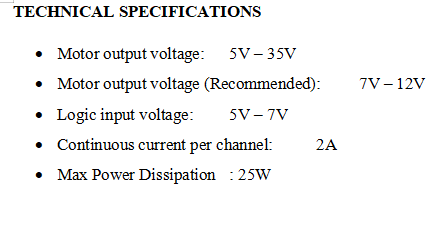


**Speed Control Pins**

The speed control pins ENA and ENB are used to turn on/off the motors and control its speed.

Pulling these pins HIGH will cause the motors to spin, while pulling it LOW will stop them. But, with Pulse Width Modulation (PWM), you can actually control the speed of the motors.

The module usually comes with a jumper on these pins. When this jumper is in place, the motor spins at maximum speed. If you want to control the speed of the motors programmatically, you need to remove the jumpers and connect them to the PWM-enabled pins on the Arduino.



**DC GEAR MOTOR**

Geared DC motors can be defined as an extension of DC motor which already had its Insight details demystified before. A geared DC Motor has a gear assembly attached to the motor. The speed of motor is counted in terms of rotations of the shaft per minute and is termed as RPM. The gear assembly helps in increasing the torque and reducing the speed. Using the correct combination of gears in a gear motor, its speed can be reduced to any desirable figure. This concept where gears reduce the speed of the vehicle but increase its torque is known as gear reduction. This Insight will explore all the minor and major details that make the gear head and hence the working of geared DC motor.

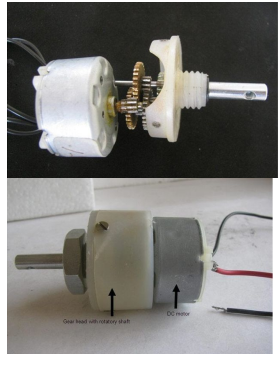


**Working of the DC Geared Motor**

The DC motor works over a fair range of voltage. The higher the input voltage more is the RPM (rotations per minute) of the motor. For example, if the motor works in the range of 6-12V, it will have the least RPM at 6V and maximum at 12 V. In terms of voltage, we can put the equation as: RPM= K1 \* V, where, K1= induced voltage constant V=voltage applied.

The working of the gears is very interesting to know. It can be explained by the principle of conservation of angular momentum.

The gear having smaller radius will cover more RPM than the one with larger radius. However, the larger gear will give more torque to the smaller gear than vice versa. The comparison of angular velocity between input gear (the one that transfers energy) to output gear gives the gear ratio. When multiple gears are connected together, conservation of energy is also followed. The direction in which the other gear rotates is always the opposite of the gear adjacent to it. In any DC motor, RPM and torque are inversely proportional. Hence the gear having more torque will provide a lesser RPM and converse. In a geared DC motor, the concept of pulse width modulation is applied.



**Parts of DC Gear Motor**

For example, an unloaded DC motor might spin at 12000 rpm and provide 0.1 kg-cm of torque. A 225:1 gear down is added to proportionally reduce the speed and increase the torque: 12000 rpm / 225 = 53.3 rpm and 0.1 x 225 = 22.5 kg-cm. The motor will now be able to move significantly more weight at a more reasonable speed. In a geared DC motor, the gear connecting the motor and the gear head is quite small, hence it transfers more speed to the larger teeth part of the gear head and makes it rotate.

**Controlling of Geared Motor**

DC gear motor can be controlled exactly the same way DC motor control.

**Applications:**

Robot Drive TrainsRadio Control VehiclesCordless Tools

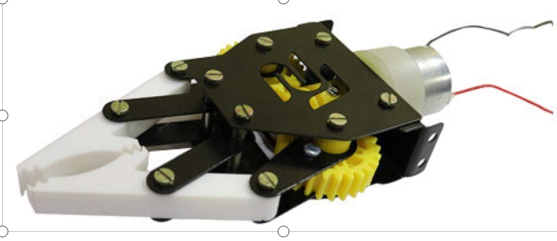
**Advantages:**

Speed Reduction - Many DC motors simply run too fast to be useful in direct-drive applications.Increased Torque - A lot of work can be coaxed from a relatively small motor if fitted with a suitable gear train.

**Limitations:**

This is especially a problem with low-cost plastic gear trains used with low-voltage motors. The extra resistance canmake these gear-trains balky at low speeds. For Controller & Motor browse:

**MECHANICAL ROBOTIC GRIPPER ARM**



**Mechanical Robotic Gripper Arm**

The Gripper module is state of the art robotic arm designed indigenously by Robomart. It can be used in various ‘pick and place’ kind of robots. It works on DC Motor (9 to 12V DC). Change in rotation direction of the DC Motor, generates Jaw Open & Close Action.

The DC motor can be easily be controlled with the help of DPDT Switch (manual mode) or with the help of any microcontroller along with L293D Motor Driver module. Give an extra functionality to your robots by adding a fully functional Robot gripper to them.

**Gripper Arm:** The kit provides a gripper arm or claw that can open and close to grasp objects. The design and size of the gripper arm can vary, but it’s usually adjustable and capable of gripping a wide range of objects.

**DC Motor:** The gripper arm is powered by a DC (Direct Current) motor. The motor provides the necessary mechanical force to open and close the gripper. The kit may include a pre-wired motor, or you might need to connect it to a power source and control circuitry.

**Control Mechanism:** Some kits include a control mechanism such as a microcontroller or remote control that allows you to operate the gripper arm. This can involve controlling the motor’s direction and speed to precisely manipulate objects.

**Mounting Hardware:** The kit often includes mounting hardware and brackets to attach the gripper arm to a robotic platform or other structure. This ensures stability and ease of integration into your project.

**Power Supply:** Depending on the kit, you might need to provide your own power supply, or the kit may include a power source, such as a battery pack or an adapter.

**PARTS SPECIFICATIONS**

* Gripper assembly Plates.
* 2x (Fiber Grippers).
* 1x (45 RPM DC Metal Geared Motor).
* 1x (Worm Gear).
* 2x (Spur Gear).

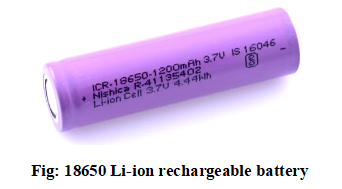
**Motor Type:** DC Gear Motor (45RPM)

**Gripper Type:** Mechanical, typically with adjustable jaws

**Control Method:** Typically includes a control board for motor control Material: Usually made of lightweight and durable materials such as plastic or aluminum

**Power Supply**: Requires a DC power source, commonly in the range of 5-12 volts

**18650 LI-ION RECHARGEABLE BATTERY**



18650 battery is a Li-ion rechargeable battery with a 1200 mAh Battery Capacity. This is not a standard AA or AAA battery but is very useful for applications that require continuous high current or high current in short bursts like in cameras, DVD players, iPod, etc. A 18650 cell can be charged and discharged up to 1000 cycles without much loss in battery capacity. They are safe to use, environment friendly and have long battery life. It comes with high energy density and provides excellent continuous power sources to your device. It should be used with a protection circuit board that guards the battery against over-charge, over-discharge of the pack, and avoid over-current drawn.

**Specifications**

* Voltage: 3.7 Volts
* Capacity: 3200 mAh
* Rechargeable: Yes
* Battery Size: Diameter- 18mm x Length- 67mm
* Charging Method CC-CV
* Model: 18650

**CHAPTER-6**

**SOFTWARE DESCRIPTION**

**INTRODUCTION TO ARDUINO IDE**

The Arduino Integrated Development Environment (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

IDE stands for “Integrated Development Environment” :it is an official software introduced by Arduino.cc, that is mainly used for editing, compiling and uploading the code in the Arduino Device.

**ARDUINO IDE DEFINITION**

1. Arduino IDE is an open source software that is mainly used for writing and compiling the code into the Arduino Module.

2. It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.

3. It is easily available for operating systems like MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment.

4. A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro and many more.

5. Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code.

**How to get Arduino IDE**

we can download the Software from Arduino main website. As I said earlier, the software is available for common operating systems like Linux, Windows, and MACos,we select to download the correct software version that is easily compatible with our operating system.

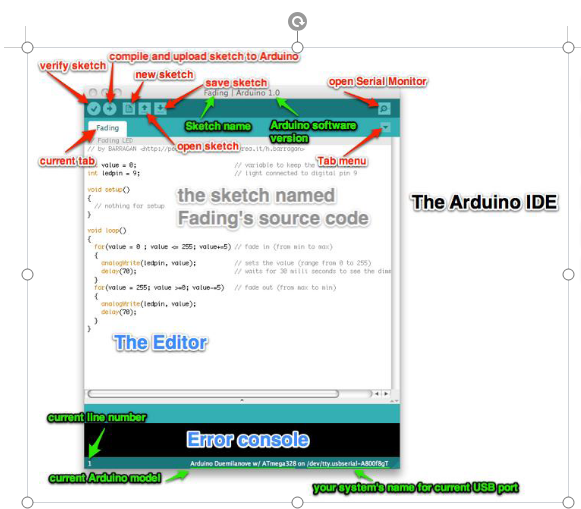
**Details on IDE**: The IDE environment is mainly distributed into three sections

**1. Menu Bar**

**2. Text Editor**

**3. Output Pane**

As we download and open the IDE software, it will appear like an image below.

 Arduino IDE Main Window

The bar appearing on the top is called **Menu Bar** that comes with five different options as follow

**File** – You can open a new window for writing the code or open an existing one. Following table shows the number of further subdivisions the file option is categorized into.

And at the end of compilation, it will show you the hex file it has generated for the recent sketch that will send to the Arduino Board for the specific task you aim to achieve.

##### 

##### Arduino IDE Code Window

##### 

**Edit** – Used for copying and pasting the code with further modification for font

**Sketch** – For compiling and programming

**Tools** – Mainly used for testing projects. The Programmer section in this panel is used for burning a bootloader to the new microcontroller.

**Help** – In case you are feeling skeptical about software, complete help is available from getting started to troubleshooting.

The **Six Buttons** appearing under the Menu tab are connected with the running program as follow.

##### 

.

* The check mark appearing in the circular button is used to verify the code. Click this once you have writtenyour code.
* The arrow key will upload and transfer the required code to the Arduino board.
* The dotted paper is used for creating a new file.

The upward arrow is reserved for opening an existing Arduino project.

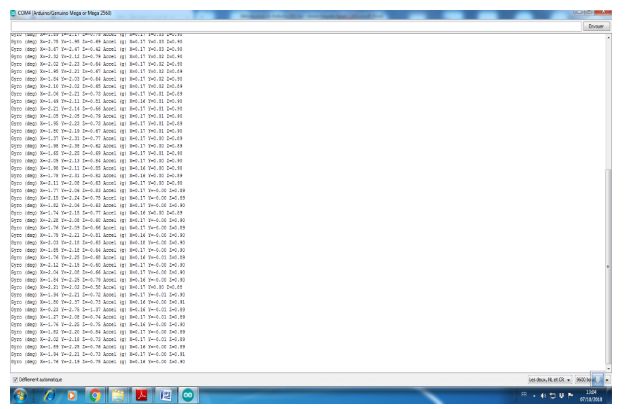
* The downward arrow is used to save the current running code.

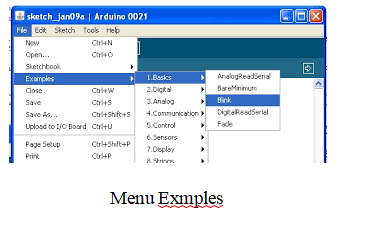
F THE

The button appearing on the top right corner is a **Serial Monitor** – A separate pop-up window that acts as an independent terminal and plays a vital role for sending and receiving the Serial Data. You can also go to the Tools panel and select Serial Monitor pressing Ctrl+Shift+M all at once will open the Serial Monitor. The Serial Monitor will actually help to debug the written Sketches where you can get a hold of how your program is operating.

* You need to select the baud rate of the Arduino Board you are using right now. For my Arduino Uno Baud Rate is 9600, as you right the following code and click the Serial Monitor, the output will show as the image below.

The main screen below show how to select a program from examples ie: Bink.cc







The bottom of the main screen is described as an Output Pane that mainly highlights the compilation status of the running code: the memory used by the code, and errors occurred in the program. You need to fix those errors before you intend to upload the hex file into your Arduino Module.More or less, Arduino C language works similar to the regular C language used for any embedded system microcontroller,

**Libraries**

Libraries are very useful for adding the extra functionality into the Arduino Module. There is a list of libraries you can add by clicking the Sketch button in the menu bar and going to Include Library.

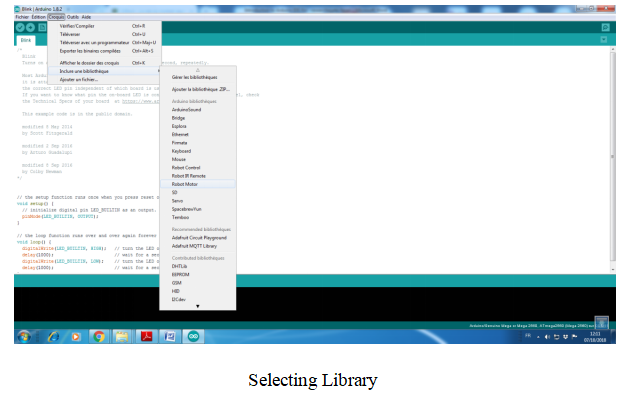
As you click the Include Library and Add the respective library it will on the top of the sketch with a #include sign. Suppose, I Include the EEPROM library, Temperature sensors DHT11/22, LCD or I2C library it will appear on the text editor as

**#include <EEPROM.h>.**

**#include <dht.h>**

**#include <I2Cdev.h>**

Most of the libraries are preinstalled and come with the Arduino software. However, we can also download them from the external sources.

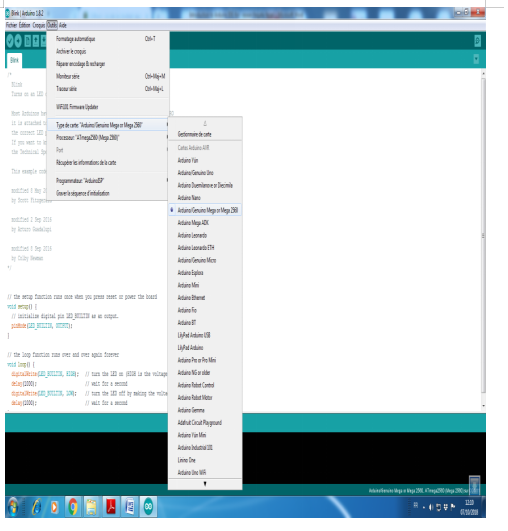


**Making Pins As Input or Output**

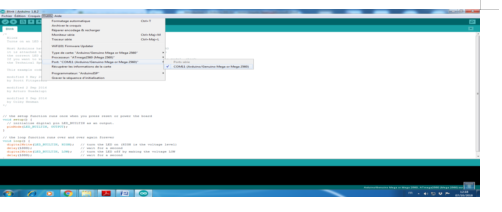
* The digitalRead and digitalWrite commands are used for addressing and making the Arduino pins as an input and output respectively.
* These commands are text sensitive i.e. you need to write them down the exact way they are given like digitalWrite starting with small “d” and write with capital “W”. Writing it down with Digitalwrite or digitalwritewon’t be calling or addressing any function.
* Example : if we want to use Pin D13 as output , the code will be; pinMode(13, OUTPUT); followed by digitalWrite(13,HIGH);
* If we want to use Pin D13 as input, the code will be : pinMode(13, INPUT);followed by x=digitalRead(13);

**Selecting Board of Arduino**

In order to upload the sketch, we need to select the relevant board we are using and the ports for that operating system. As we click the Tools on the Menu, it will open like the figure below.



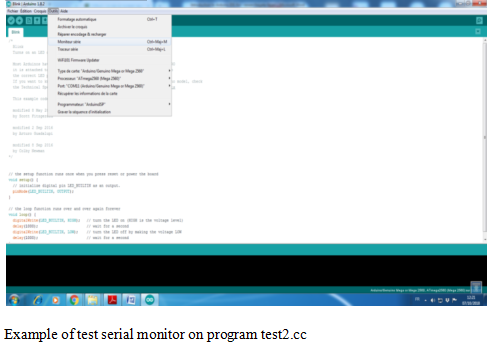
Just we go to the “Board” section and select the board wewould like to work on. Similarly, COM1, COM2, COM4, COM5, COM7 or higher are reserved for the serial and USB board. we can look for the USB serial device in the ports section of the Windows Device Manager.



After correct selection of both Board and Serial Port, click the verify and then upload button appearing in the upper left corner of the six button section or you can go to the Sketch section and press verify/compile and then upload.

* The sketch is written in the text editor and is then saved with the file extension .ino.
* It is important to note that the recent Arduino Modules will reset automatically as you compile and press the upload button the IDE software, however, older version may require the physical reset on the board.
* As we upload the code, TX and RX LEDs will blink on the board, indicating the desired program is running successfully.

**Using Serial Monitor**



**WRITING SKETCHES**

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

NB: Versions of the Arduino Software (IDE) prior to 1.0 saved sketches with the extension .pde. It is possible to open these files with version 1.0, you will be prompted to save the sketch with the .ino extension on save.

Additional commands are found within the five menus: File, Edit, Sketch, Tools, Help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available. Since version 1.0.1, the Arduino Software (IDE) has been translated into 30+ different languages. By default, the IDE loads in the language selected by your operating system. (Note: on Windows and possibly Linux, this is determined by the locale setting which controls currency and date formats, not by the language the operating system is displayed in.)

If you would like to change the language manually, start the Arduino Software (IDE) and open the Preferences window. Next to the Editor Language there is a dropdown menu of currently supported languages. Select your preferred language from the menu, and restart the software to use the selected language. If your operating system language is not supported, the Arduino Software (IDE) will default to English. You can return the software to its default setting of selecting its language based on your operating system by selecting System Default from the Editor Language drop-down. This setting will take effect when you restart the Arduino Software (IDE)

**CHAPTER-7**

**SOFTWARE PROGRAM**

## Chapter 5 CONCLUSION

In conclusion, the introduction of the smart shopping trolley with automated billing and EV installation presents a transformative shift in the retail and transportation industries. This innovation seamlessly combines convenience, efficiency, and sustainability, offering numerous benefits to both retailers and consumers. Firstly, the automated billing system integrated into the shopping trolley streamlines the shopping experience. Customers no longer need to wait in long checkout lines, scan items individually, or manually input payment information. Instead, they can simply place items in their cart, and the trolley's technology automatically keeps track of their purchases. This not only saves time but also reduces the potential for errors in billing, enhancing overall customer satisfaction. Furthermore, the incorporation of EV (Electric Vehicle) charging capabilities into the shopping trolley aligns with the growing emphasis on sustainability and eco-consciousness. Shoppers who own electric vehicles can conveniently charge their cars while shopping, promoting the use of cleaner transportation options. This can contribute to a reduction in greenhouse gas emissions and aligns with global efforts to combat climate change. Retailers also stand to benefit significantly from this innovation. Automated billing reduces the need for multiple cashier lanes and minimizes the risk of theft or shoplifting, improving operational efficiency and security. Additionally, offering EV charging facilities can attract environmentally conscious customers and serve as a competitive advantage for retailers. In summary, the smart shopping trolley with automated billing and EV installation represents a win-win solution for retailers and consumers alike. It offers a more efficient and enjoyable shopping experience while promoting sustainability and supporting the transition to cleaner transportation options. As technology continues to evolve, this innovation holds promise for reshaping the future of retail and transportation.

# APPLICATIONS & FUTURE SCOPE

#### APPLICATIONS:

* **Retail Transactions:** Smart billing systems streamline the payment process at retail stores within the mall. They enable quick and convenient payments through various methods, such as contactless payments, mobile wallets, and digital receipts.
* **Queue Management:** In busy times, smart billing can help manage queues by enabling mobile payments, reducing wait times, and improving customer satisfaction.
* **Security and Fraud Prevention:** Smart billing systems offer security features like encrypted transactions and fraud detection mechanisms, ensuring the safety of both customers and businesses.
* **Mall Memberships:** If the mall offers membership programs with special discounts or benefits, a smart billing system can manage membership fees and provide personalized offers to members.
* **Tenant Billing:** For retailers and businesses leasing space in the mall, smart billing systems can manage rental payments, common area maintenance charges, and other associated fees.

#### FUTURE SCOPE:

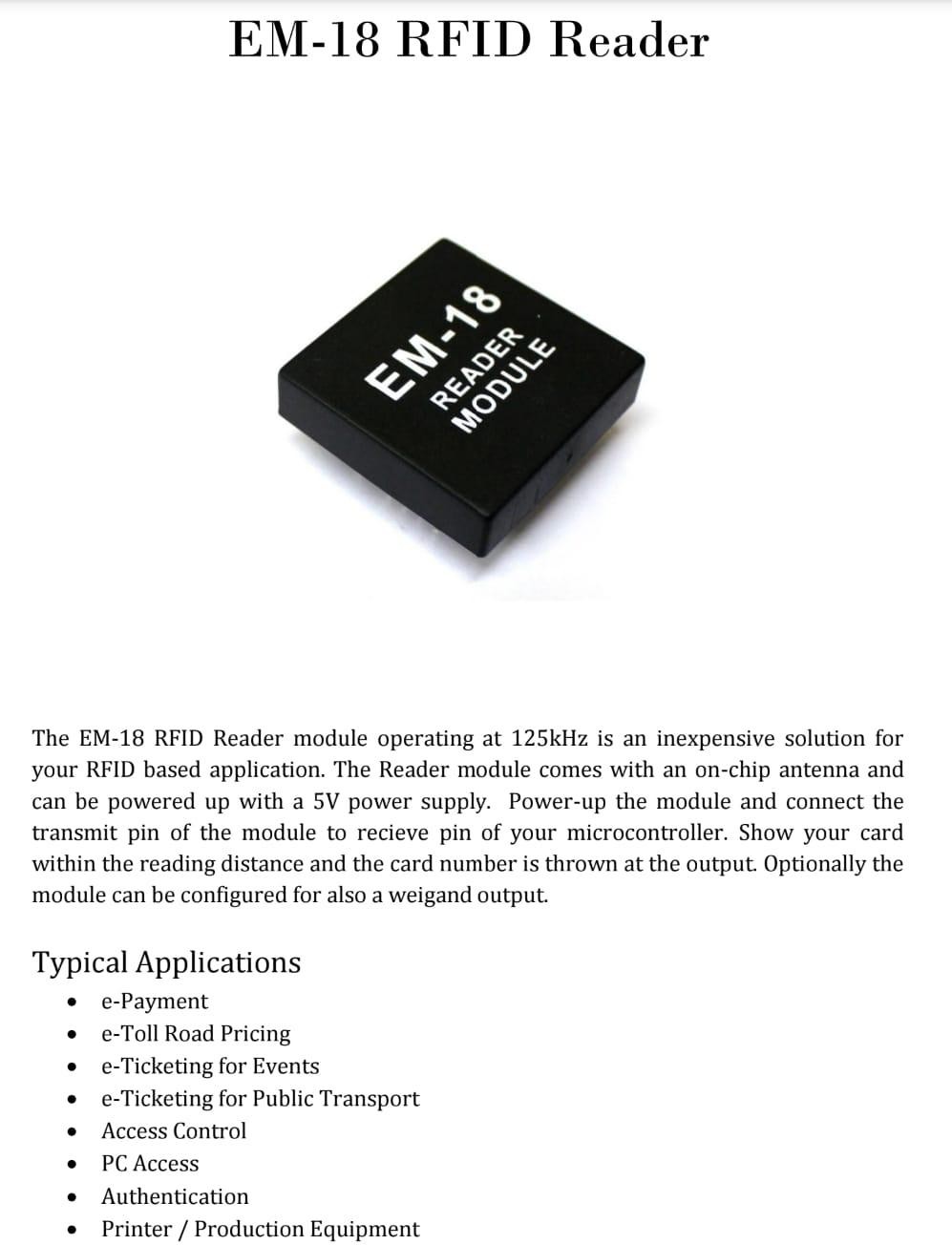
* **Enhanced Shopping Experience:** Smart shopping trolleys can provide personalized recommendations, product information, and navigation assistance within the store. This can enhance the overall shopping experience for customers.
* **Efficient Checkout:** Automated billing and payment processing can significantly reduce checkout times and queues, making shopping more convenient for customers and improving operational efficiency for retailers.
* **Inventory Management:** These trolleys can track inventory levels in real-time, helping retailers restock shelves more efficiently and reduce out-of-stock situations.
* **Data Analytics:** The data collected from smart trolleys can be analyzed to gain insights into customer behavior, preferences, and shopping trends. Retailers can use this information for targeted marketing and inventory management.
* **Integration with Mobile Apps:** Seamless integration with mobile apps can allow customers to create shopping lists, receive promotions, and access loyalty programs, enhancing customer engagement.
* **Environmental Benefits:** EV installation on trolleys can contribute to sustainability efforts by reducing carbon emissions associated with traditional gasoline-powered vehicles for restocking shelves.
* **Retailer Efficiency:** Retailers can optimize their operations by using smart trolleys to automate tasks like restocking, price tagging, and store layout adjustments.
* **AI and Machine Learning:** Continued advancements in AI and machine learning can enable smart trolleys to provide more accurate product recommendations and personalized offers to shoppers.
* **Security and Privacy:** As these trolleys collect a significant amount of data, ensuring robust security and privacy measures will be crucial to gaining and maintaining customer trust.
* **Market Expansion:** The adoption of smart shopping trolleys may extend beyond traditional grocery stores to other retail segments such as electronics, clothing, and home improvement.
* **Global Adoption:** The concept of smart shopping trolleys may spread to markets around the world, offering opportunities for companies to expand internationally.
* **Regulatory Considerations:** As with any technology that handles customer data and integrates with payment systems, compliance with data protection and financial regulations will be essential.

# REFERENCES

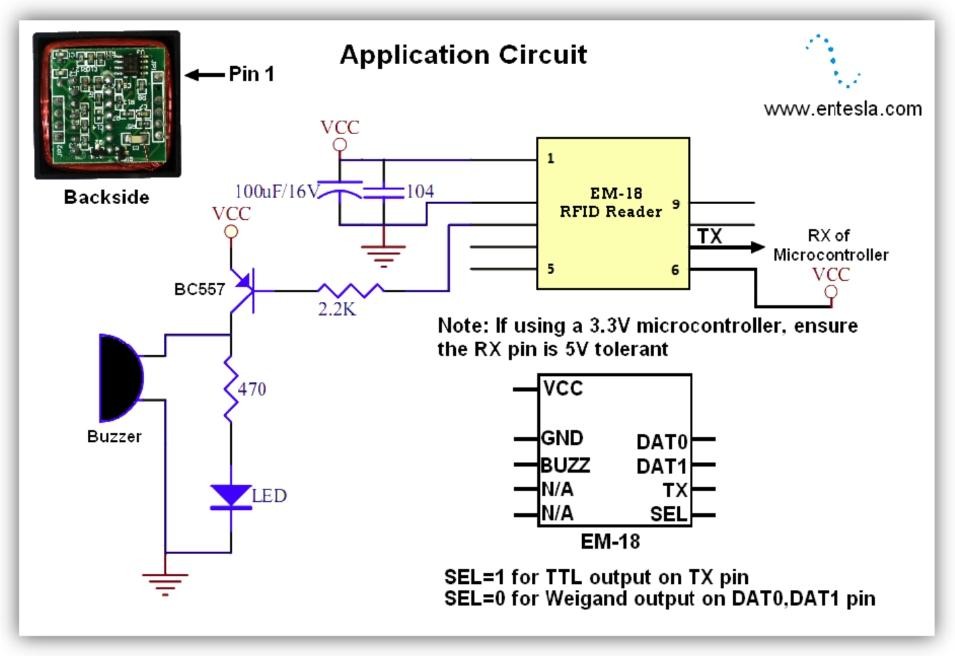
* Shridhar Mathad,”Smart Trolley with Automated Billing Systems Using Arduino”, February 2022 IAES(International Journal of Artifical Intelligence).
* Al Sult Al Kharusi , Dinesh Keloth kaithari, Muhammad Mumtaz Mirza , Parimal S. Bhambare, ”Electrically Operated Multipurpose Trolley”, June 2018 IOP Conference Series Materials Science and Engineering.
* Pavni Swaroop , Akshita Parasari , Mansi Singh, Shobha Rajput, “ Smart Trolley with Automated billing Systems Using Arduino”, August 2022 International Journal Of Creative Research Thoughts (IJCRT).
* Snehal Kulkarni , Dr. Supriya Shanbhag,” Smart Shopping Cart System”, December 2022 (IJARSCT).
* Rajini.H , Sandeep Jaiswal , Shyam Sunder Prasad , Kushboo , Anjela Kadiem,” Development of Automatic Shopping Trolley in Supermarkets”, November2022 (IJAER)
* https://how2electronics.com/smart-shopping-cart-with-automatic-billing-system-using- rfid-arduino/ (Smart Shopping cart)
* **https://circuitdigest.com/microcontroller-projects/smart-shopping-cart-with-automatic- billing-system-using-raspberry-pi**
* **https://youtu.be/2klCHdtFDg8** (design process of Multipurpose Trolley)

### APPENDICES

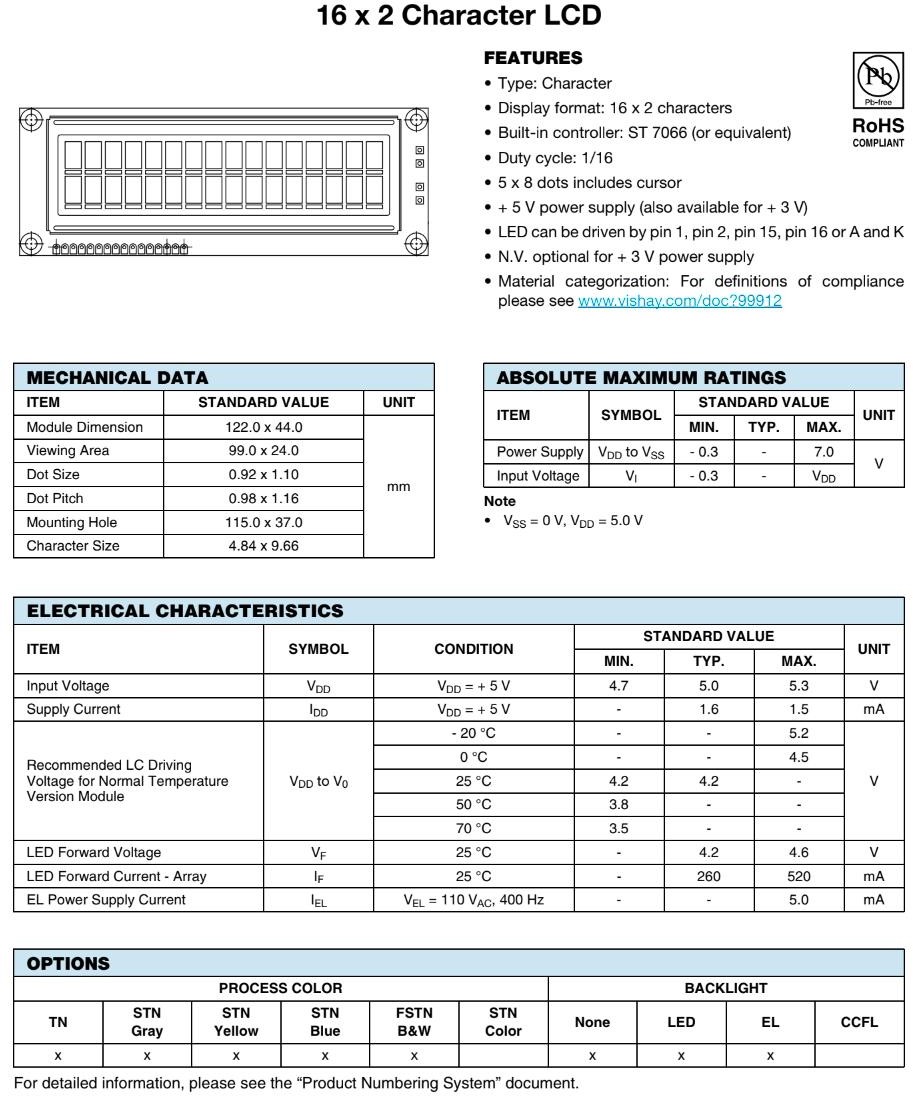
**EM18 RFID Module**







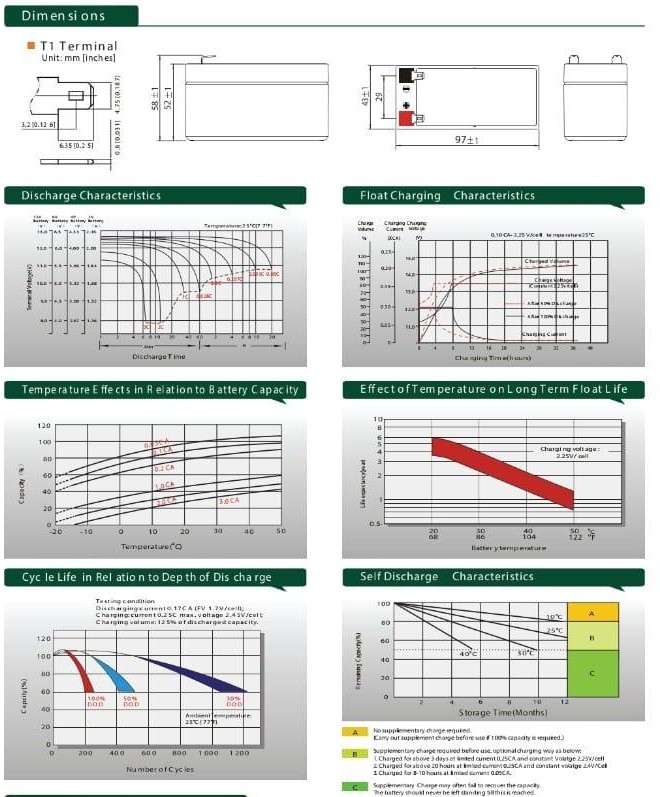
**LCD Display**



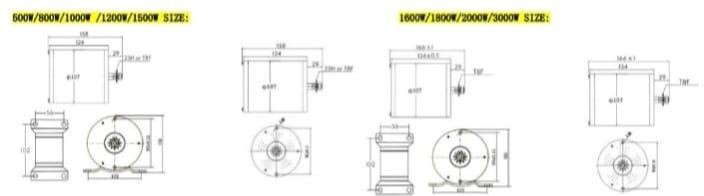
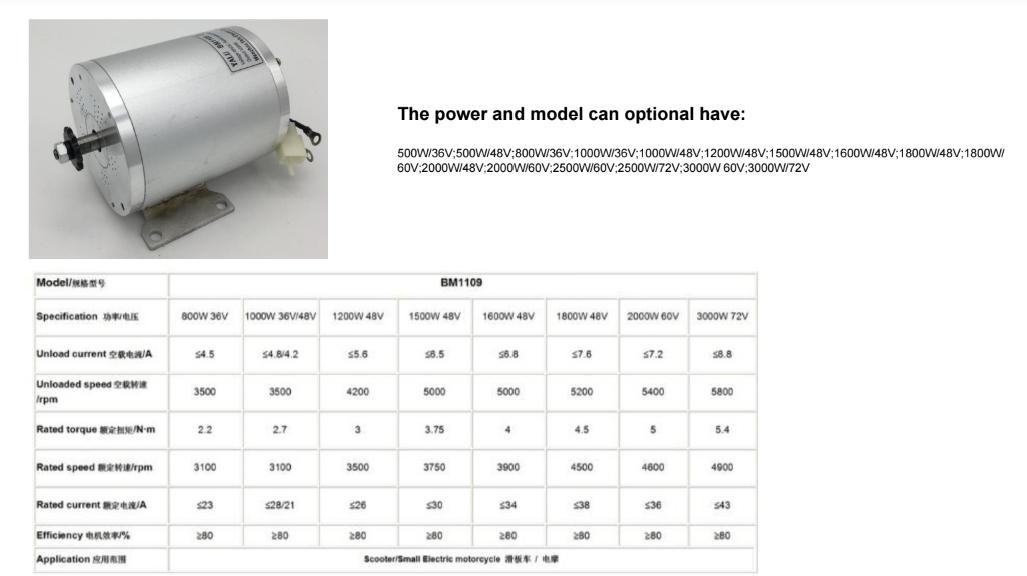


**Sealed Rechargable Battery**

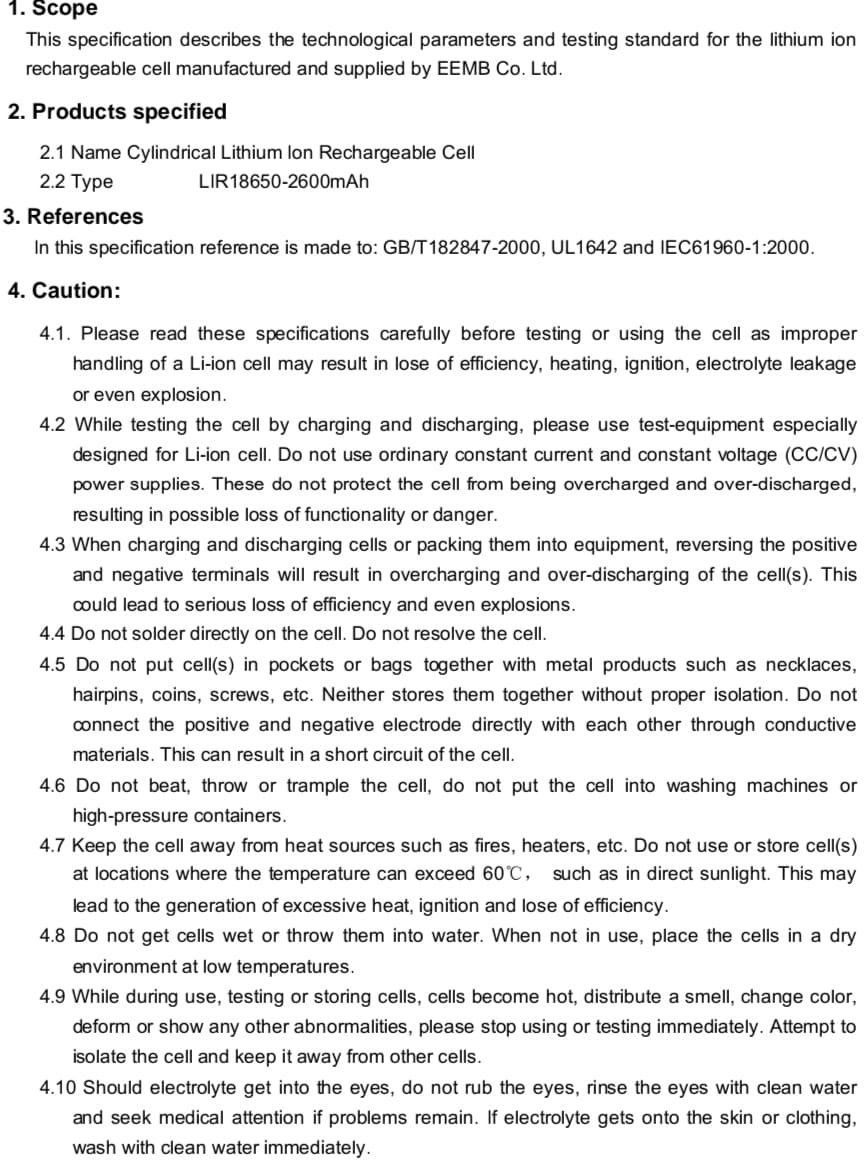


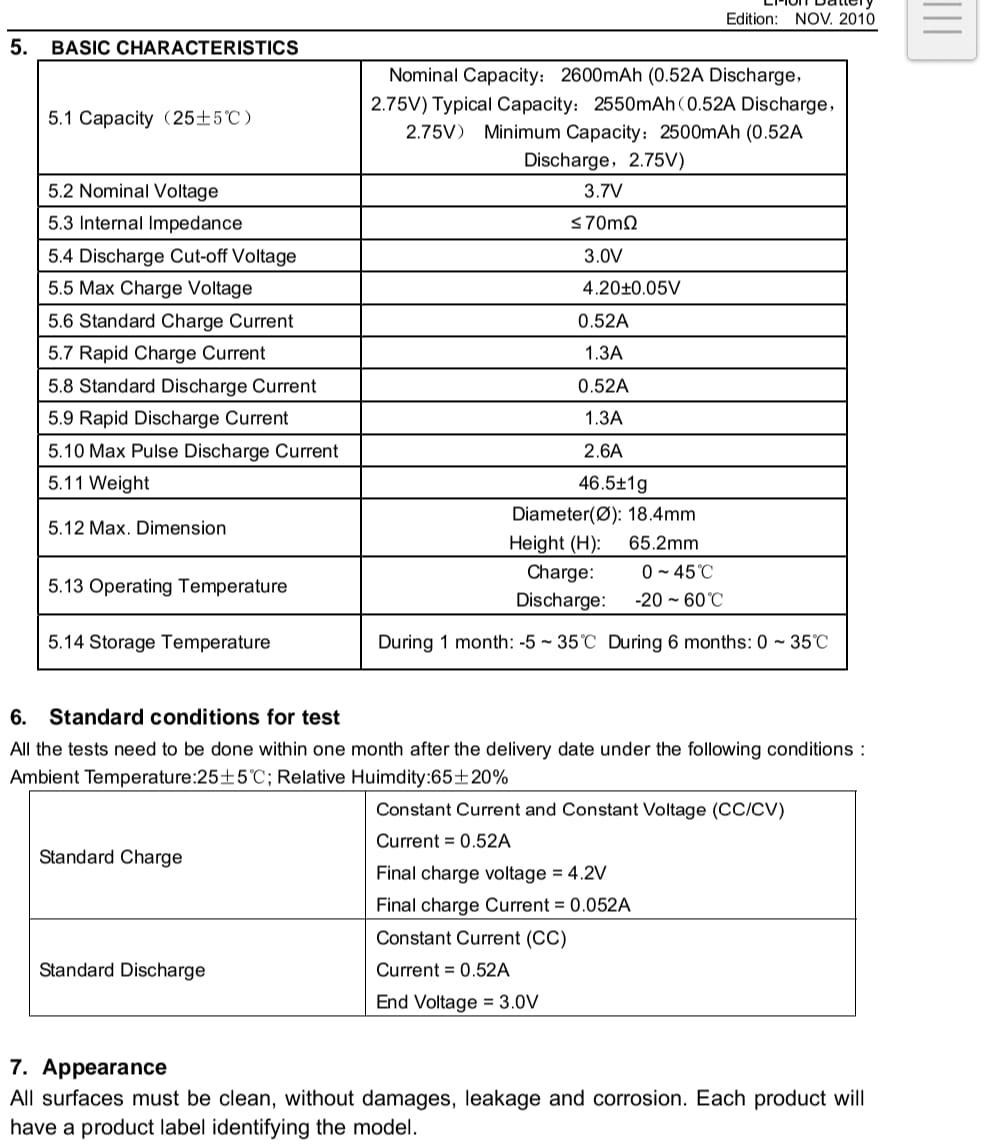


**BLDC1109 Motor**

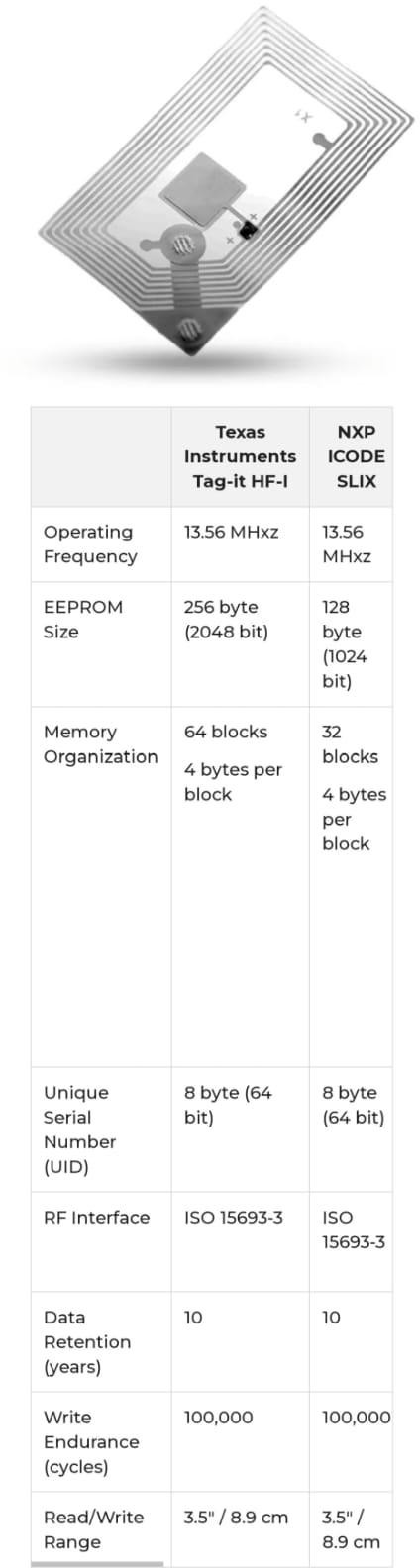


**Lithium ion battery**





**RFID TAGS**



**Code**

#include <LiquidCrystal\_I2C.h> #include <SoftwareSerial.h>

#include <stdio.h> #include <stdlib.h> #include <string.h>

void SEND\_SMS( char \*num1, char \* str1 ); void rfid1(void);

LiquidCrystal\_I2C lcd(0x27,16,2);

SoftwareSerial RfidSerial(6, 7); // RX, TX SoftwareSerial Serial1(2, 3); // RX, TX

int BILL=4;

int REMOVE=8; int BUZZER=5; int RFID\_LED=9;

String message=""; String str;

void rfid1(void);

char total1[4]; int L=0,M=0;

int item1=25,item2=35,item3=50,item4=30,total; int exceed;

void setup()

{

Serial.begin(9600); Serial1.begin(9600); RfidSerial.begin(9600); lcd.init(); lcd.backlight();

pinMode(BILL,INPUT\_PULLUP); pinMode(REMOVE,INPUT\_PULLUP); pinMode(BUZZER,OUTPUT); pinMode(RFID\_LED,OUTPUT);

lcd.begin(16,2);

lcd.clear();

lcd.print("SMART TROLLEY"); Serial.println("$SMART TROLLEY#");

SEND\_SMS("9113854195","WELCOME TO SHOPPING");

delay(2000); digitalWrite(BUZZER,LOW); digitalWrite(RFID\_LED,LOW);

}

void loop()

{

if(digitalRead(BILL)==LOW)

{

str = String(total); str.toCharArray(total1,4); lcd.clear(); lcd.setCursor(0,0);

lcd.print("TOTAL BILL IS:"); lcd.setCursor(0,1); lcd.print(total);

Serial.print("$TOTAL BILL IS:"); Serial.print(total); Serial.println("#");

SEND\_SMS("9113854195","TOTAL BILL IS:");

delay(2000); SEND\_SMS("9113854195",total1);

delay(2000);

SEND\_SMS("9113854195","THANK FOR SHOPPING VISIT AGAIN");

lcd.clear(); lcd.setCursor(0,0); lcd.print("THANK YOU"); lcd.setCursor(0,1); lcd.print("VISIT AGAIN"); delay(2000);

delay(2000); while(1);

// if(total>200)

// {

// lcd.clear();

// lcd.setCursor(0,0);

// lcd.print("MORE BILL ");

//

// digitalWrite(BUZZER,HIGH);

// delay(1000);

// digitalWrite(BUZZER,LOW);

//

// }

// total=0;

// rfid1();

// digitalWrite(BUZZER,LOW);

}

if(digitalRead(REMOVE)==LOW)

{

lcd.clear(); lcd.setCursor(0,0); lcd.print("REMOVE"); rfid2();

}

rfid1();

}

void rfid1(void)

{

lcd.clear(); lcd.setCursor(0,0);

lcd.print("SCAN THE PRODUCT");

delay(1000);

while(1)

{

if(digitalRead(BILL)==LOW)

{

str = String(total); str.toCharArray(total1,4); lcd.clear(); lcd.setCursor(0,0);

lcd.print("TOTAL BILL IS:"); lcd.setCursor(0,1);

lcd.print(total); Serial.print("$TOTAL BILL IS:"); Serial.println(total); Serial.println("#");

SEND\_SMS("9113854195","TOTAL BILL IS:");

delay(2000); SEND\_SMS("9113854195",total1);

delay(2000);

SEND\_SMS("9113854195","THANK FOR SHOPPING VISIT AGAIN");

lcd.clear(); lcd.setCursor(0,0); lcd.print("THANK YOU"); lcd.setCursor(0,1); lcd.print("VISIT AGAIN"); delay(2000);

delay(2000); while(1);

// if(L>M)

// {

// Serial.println("111111111");

//

// }

// if(L<M)

// {

// Serial.println("000000000");

//

// }

// if(total>200)

// {

// lcd.clear();

// lcd.setCursor(0,0);

// lcd.print("MORE BILL ");

// digitalWrite(BUZZER,HIGH);

// delay(1000);

// digitalWrite(BUZZER,LOW);

//

// }

// delay(5000);

// total=0;

// L=0;

// M=0;

// rfid1();

}

if(digitalRead(REMOVE)==LOW)

{

lcd.clear(); lcd.setCursor(0,0); lcd.print("REMOVE"); rfid2();

}

if(RfidSerial.available()>0)

{

message=RfidSerial.readString(); int str\_len = message.length() + 1; char textmessage[12];

message.toCharArray(textmessage,str\_len); Serial.println(textmessage); textmessage[12]='\0';

if((strcmp(textmessage,"17003B656B22"))==0)

lcd.setCursor(0,0); lcd.print("SUGAR ,ID:001 "); lcd.setCursor(0,1); lcd.print("AMOUNT:25Rs. "); total=total+item1; digitalWrite(RFID\_LED,HIGH); delay(3000); digitalWrite(RFID\_LED,LOW);

lcd.clear(); lcd.setCursor(0,0); lcd.print("BILL"); lcd.setCursor(0,1); lcd.print(total); Serial.println("SUGAR 25Rs");

//Serial.print(total); delay(5000);

L=L+1;

if(total>200)

{

lcd.clear(); lcd.setCursor(0,0); lcd.print("MORE BILL ");

digitalWrite(BUZZER,HIGH); delay(10000); digitalWrite(BUZZER,LOW);

}

rfid1();

}

if((strcmp(textmessage,"190083E1552E"))==0)

lcd.setCursor(0,0); lcd.print("RICE ,ID:002"); lcd.setCursor(0,1); lcd.print("AMOUNT:35Rs. "); total=total+item2; digitalWrite(RFID\_LED,HIGH); delay(3000); digitalWrite(RFID\_LED,LOW); lcd.clear();

lcd.setCursor(0,0); lcd.print("BILL"); lcd.setCursor(0,1); lcd.print(total); Serial.println("RICE 35Rs."); delay(5000);

M=M+1;

if(total>200)

{

lcd.clear(); lcd.setCursor(0,0); lcd.print("MORE BILL ");

digitalWrite(BUZZER,HIGH); delay(1000); digitalWrite(BUZZER,LOW);

}

rfid1();

}

if((strcmp(textmessage,"19006A21C290"))==0)

{

lcd.clear();

lcd.setCursor(0,0); lcd.print("OIL ,ID:003 "); lcd.setCursor(0,1); lcd.print("AMOUNT:50Rs. "); total=total+item3;

digitalWrite(RFID\_LED,HIGH); delay(3000); digitalWrite(RFID\_LED,LOW); lcd.clear();

lcd.setCursor(0,0); lcd.print("BILL"); lcd.setCursor(0,1); lcd.print(total); Serial.println("OIL 50Rs. "); delay(5000);

if(total>200)

{

lcd.clear(); lcd.setCursor(0,0); lcd.print("MORE BILL ");

digitalWrite(BUZZER,HIGH); delay(1000); digitalWrite(BUZZER,LOW);

}

rfid1();

}

if((strcmp(textmessage,"19006E963EDF"))==0)

{

lcd.clear(); lcd.setCursor(0,0);

lcd.print("SALT ,ID:004 "); lcd.setCursor(0,1); lcd.print("AMOUNT:30Rs. "); total=total+item4;

digitalWrite(RFID\_LED,HIGH); delay(3000); digitalWrite(RFID\_LED,LOW); lcd.clear();

lcd.setCursor(0,0); lcd.print("BILL"); lcd.setCursor(0,1); lcd.print(total); Serial.println("SALT 30Rs. "); delay(5000);

if(total>200)

{

lcd.clear(); lcd.setCursor(0,0); lcd.print("MORE BILL ");

digitalWrite(BUZZER,HIGH); delay(1000); digitalWrite(BUZZER,LOW);

}

rfid1();

}

if((strcmp(textmessage,"170073BB7AA5"))==0)

{

lcd.clear(); lcd.setCursor(0,0);

lcd.print("CURRENTLY "); lcd.setCursor(0,1); lcd.print("UNAVAILABLE ");

Serial.println("CURRENTLY UNAVAILABLE"); total=total;

digitalWrite(RFID\_LED,HIGH); delay(3000); digitalWrite(RFID\_LED,LOW); lcd.clear();

lcd.setCursor(0,0); lcd.print("BILL"); lcd.setCursor(0,1); lcd.print(total);

// Serial.println("OIL 50Rs. "); delay(5000);

// if(total>200)

// {

// lcd.clear();

// lcd.setCursor(0,0);

// lcd.print("MORE BILL ");

// digitalWrite(BUZZER,HIGH);

// delay(1000);

// digitalWrite(BUZZER,LOW);

//

//

// }

rfid1();

}

if(digitalRead(BILL)==LOW)

{

str = String(total); str.toCharArray(total1,4); lcd.clear(); lcd.setCursor(0,0);

lcd.print("TOTAL BILL IS:"); lcd.setCursor(0,1); lcd.print(total);

Serial.print("$TOTAL BILL IS:"); Serial.println(total); Serial.println("#");

SEND\_SMS("9113854195","TOTAL BILL IS:");

delay(2000); SEND\_SMS("9113854195",total1);

delay(2000);

SEND\_SMS("9113854195","THANK FOR SHOPPING VISIT AGAIN");

lcd.clear(); lcd.setCursor(0,0); lcd.print("THANK YOU"); lcd.setCursor(0,1); lcd.print("VISIT AGAIN"); delay(2000);

delay(2000); while(1);

// if(L>M)

// {

// Serial.println("111111111");

//

// }

// if(L<M)

// {

// Serial.println("000000000");

//

// }

// if(total>200)

// {

// lcd.clear();

// lcd.setCursor(0,0);

// lcd.print("MORE BILL ");

// digitalWrite(BUZZER,HIGH);

// delay(1000);

// digitalWrite(BUZZER,LOW);

//

//

// }

// total=0;

// rfid1();

//

//

}

if(digitalRead(REMOVE)==LOW)

{

lcd.clear(); lcd.setCursor(0,0); lcd.print("REMOVE"); rfid2();

}

}

}

}

void rfid2(void)

{

lcd.clear(); lcd.setCursor(0,0);

lcd.print("SCAN THE PRODUCT");

// digitalWrite(Voice1,LOW); delay(1000);

// digitalWrite(Voice1,HIGH); while(1)

{

if(digitalRead(BILL)==LOW)

{

str = String(total); str.toCharArray(total1,4); lcd.clear(); lcd.setCursor(0,0);

lcd.print("TOTAL BILL IS:"); lcd.setCursor(0,1); lcd.print(total);

Serial.print("$TOTAL BILL IS:"); Serial.println(total); Serial.println("#");

SEND\_SMS("9113854195","TOTAL BILL IS:");

delay(2000); SEND\_SMS("9113854195",total1);

delay(2000);

SEND\_SMS("9113854195","THANK FOR SHOPPING VISIT AGAIN");

lcd.clear(); lcd.setCursor(0,0); lcd.print("THANK YOU"); lcd.setCursor(0,1); lcd.print("VISIT AGAIN"); delay(2000);

delay(2000); while(1);

// if(L>M)

// {

// Serial.println("111111111");

|  |  |
| --- | --- |
| //  // | } |
| // | if(L<M) |
| // | { |
| // | Serial.println("000000000"); |
| // |  |
| // | } |
| // | total=0; |
| // | rfid1(); |

}

if(RfidSerial.available()>0)

{

message=RfidSerial.readString(); int str\_len = message.length() + 1; char textmessage[12];

message.toCharArray(textmessage,str\_len); Serial.println(textmessage); textmessage[12]='\0';

delay(1000);

if((strcmp(textmessage,"17003B656B22"))==0)

{

lcd.clear(); lcd.setCursor(0,0); lcd.print("SUGAR ,ID:001 "); lcd.setCursor(0,1);

lcd.print("REMOVED 25Rs "); total=total-item1; digitalWrite(RFID\_LED,HIGH); delay(3000); digitalWrite(RFID\_LED,LOW);

lcd.clear(); lcd.setCursor(0,0); lcd.print("BILL"); lcd.setCursor(0,1); lcd.print(total);

// Serial.println("SUGAR -25Rs");

//Serial.print(total); delay(5000);

L=L-1;

rfid1();

}

if((strcmp(textmessage,"190083E1552E"))==0)

{

lcd.clear(); lcd.setCursor(0,0); lcd.print("RICE ,ID:002"); lcd.setCursor(0,1);

lcd.print("REMOVED 35Rs. "); total=total-item2; digitalWrite(RFID\_LED,HIGH); delay(3000); digitalWrite(RFID\_LED,LOW); lcd.clear();

lcd.setCursor(0,0); lcd.print("BILL"); lcd.setCursor(0,1); lcd.print(total);

// Serial.println("RICE 35Rs."); delay(5000);

M=M-1;

rfid1();

}

if((strcmp(textmessage,"19006A21C290"))==0)

{

lcd.clear(); lcd.setCursor(0,0); lcd.print("OIL ,ID:003 "); lcd.setCursor(0,1);

lcd.print("REMOVED 50Rs. ");

total=total-item3; digitalWrite(RFID\_LED,HIGH); delay(3000); digitalWrite(RFID\_LED,LOW); lcd.clear();

lcd.setCursor(0,0); lcd.print("BILL"); lcd.setCursor(0,1); lcd.print(total);

// Serial.println("OIL 50Rs. "); delay(5000);

rfid1();

}

if((strcmp(textmessage,"19006E963EDF"))==0)

{

lcd.clear(); lcd.setCursor(0,0); lcd.print("SALT ,ID:004 "); lcd.setCursor(0,1);

lcd.print("REMOVED 30Rs. "); total=total-item4;

digitalWrite(RFID\_LED,HIGH); delay(3000); digitalWrite(RFID\_LED,LOW);

lcd.clear(); lcd.setCursor(0,0); lcd.print("BILL"); lcd.setCursor(0,1); lcd.print(total);

// Serial.println("OIL 50Rs. "); delay(5000);

rfid1();

}

if(digitalRead(BILL)==LOW)

{

str =String(total); str.toCharArray(total1,4); lcd.clear(); lcd.setCursor(0,0);

lcd.print("TOTAL BILL IS:"); lcd.setCursor(0,1); lcd.print(total);

Serial.print("$TOTAL BILL IS:"); Serial.println(total); Serial.println("#");

SEND\_SMS("9113854195","TOTAL BILL IS:");

delay(2000); SEND\_SMS("9113854195",total1);

delay(2000);

SEND\_SMS("9113854195","THANK FOR SHOPPING VISIT AGAIN");

lcd.clear(); lcd.setCursor(0,0); lcd.print("THANK YOU"); lcd.setCursor(0,1); lcd.print("VISIT AGAIN"); delay(2000);

while(1);

// if(L>M)

// {

// Serial.println("111111111");

//

// }

// if(L<M)

// {

// Serial.println("000000000");

//

// }

// total=0;

// rfid1();

}

}

}

}

void SEND\_SMS( char \*num1, char \* str1 )

{

char buff[10],i=0;

//while(i<3)

//{

Serial1.write('A'); delay(100); Serial1.write('T'); delay(100); Serial1.write('E'); delay(100); Serial1.write('0'); delay(100); Serial1.write('\r');

//i++;

// recvResponse(buff);

//Serial.print("AT sent");

//\*\*\*\*\*

Serial1.write("AT+CMGF=1\r"); //Initialize GSM For mobile delay(2000);

//Serial.print("cmgf sent");

// sendSMS("9071295134","hiiiii");

// recvResponse(buff); delay(2000);

//Serial.print("ATcmgf sent"); Serial1.write("AT+CMGS=\""); delay(2000); Serial1.write(num1); delay(2000); Serial1.write("\"\r");

// gsmSerial.write("AT+CNMI=0,0,0,0\r"); //Disabling unsolicited sms indication.

// recvResponse(buff);

//recvResponse(buff); Serial1.write(str1); delay(2000); Serial1.write(26); delay(2000); Serial.print("sms sent");

//}

}