# RFID CARD TAG PROJECT:
# IOT BASED RFID OBJECT TRACKER SYSTEM

*FINAL REPORT SUBMITTED FOR*

**ELECTRICAL AND ELECTRONICS ENGINEERING SOCIETY**
**SUMMER MENTORSHIP PROGRAMME (SMP)**

*By*


**Rijul Jana (BTECH/10663/21)**
**Amiya Jha (BTECH/10216/21)**

**DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING**
**BIRLA INSTITUTE OF TECHNOLOGY**
**MESRA, RANCHI-835215(INDIA)**

**DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING**
**BIRLA INSTITUTE OF TECHNOLOGY**
**MESRA, RANCHI-835215(INDIA)**

# CERTIFICATE

**This is to certify that the contents of the report entitled "RFID CARD TAG PROJECT" is a bonafide work carried out by the team of Mr. Rijul Jana and Mr. Amiya Jha. The contents of the report have not been submitted earlier for the award of any other degree or certificates and we hereby commend the work done by the team in this connection.**

**Dr. S Shiva Kumar**
Assistant Professor,
Dept. Of Electrical and Electronics Engineering,
Birla Institute of Technology,
Mesra, Ranchi

# ACKNOWLEDGEMENT

We would like to extend our heartfelt appreciation to Sauhardya Chatterjee and Aditya Raj, our project mentors, whose guidance and support have been instrumental in our learning and successful implementation of the concepts presented in this project. Their consistent supervision has served as a constant source of inspiration, encouraging us to delve deeper into the subject matter and accomplish the project with utmost success.

We are immensely grateful to Dr. S Shiva Kumar, the Advisor of the Electrical and Electronics Engineering Society (EEESoc) at Birla Institute of Technology. Through the Summer Mentorship Programme organized by EEESoc, BIT Mesra, he provided us with an invaluable platform to acquire knowledge and apply the learned concepts towards building something innovative.

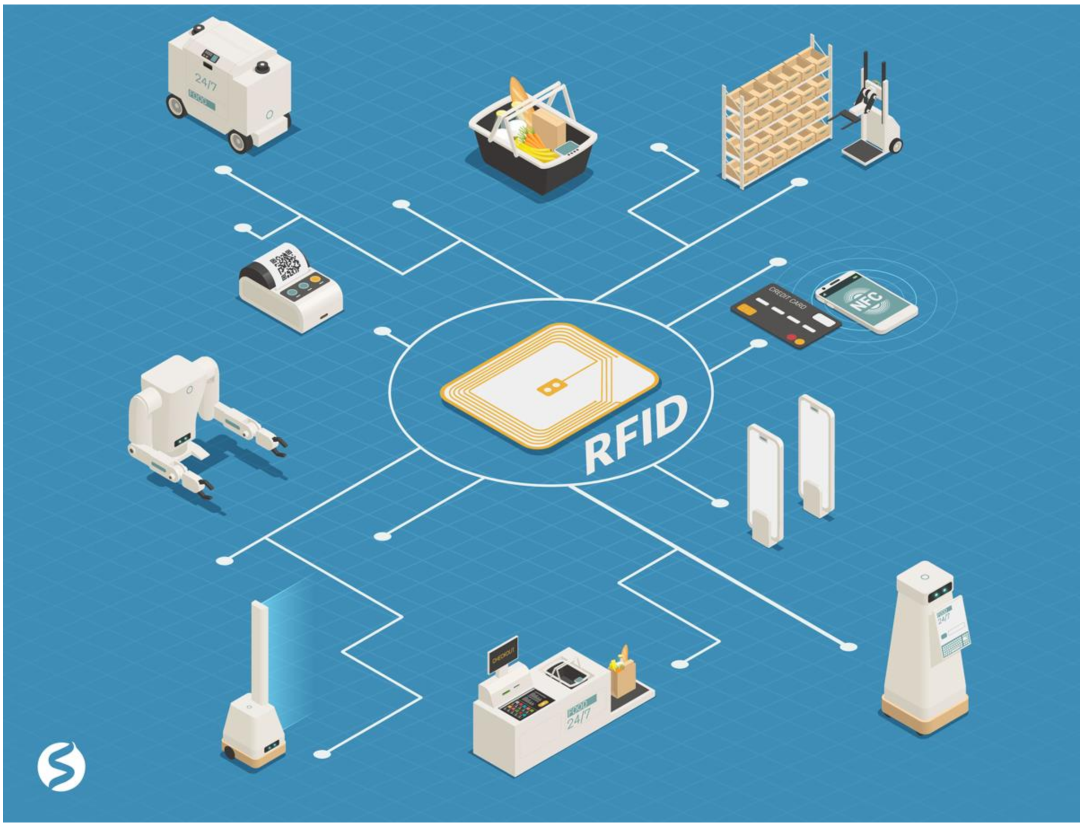
Our heartfelt gratitude also goes to Dr. Tirthadip Ghosh, Professor and Head of the Department of Electrical and Electronics Engineering at Birla Institute of Technology. His unwavering support and provision of necessary resources and opportunities have greatly contributed to our participation and growth during the Summer Mentorship Programme.

We would like to express our sincere thanks to Prof. (Dr.) Indranil Manna, the Vice Chancellor of Birla Institute of Technology, Mesra, Ranchi. His provision of essential facilities has been instrumental in the smooth execution of this project.

Rijul Jana (BTECH/10663/21)
Amiya Jha (BTECH/10216/21)

# TABLE OF CONTENTS





# ABSTRACT

RFID technology has gained significant attention in recent years due to its ability to provide real-time tracking, efficient data capture, and improved security. The objective of this project is to leverage the potential of RFID technology to enhance identification processes, streamline operations, and strengthen security measures.

The RFID tag card project involves the integration of RFID tags into traditional identification cards, transforming them into smart cards capable of transmitting and receiving information wirelessly. These RFID tag cards utilize electromagnetic fields to communicate with RFID readers, enabling quick and accurate identification of individuals or assets within a designated range. The tags are embedded with unique identifiers, enabling seamless and reliable tracking and authentication.

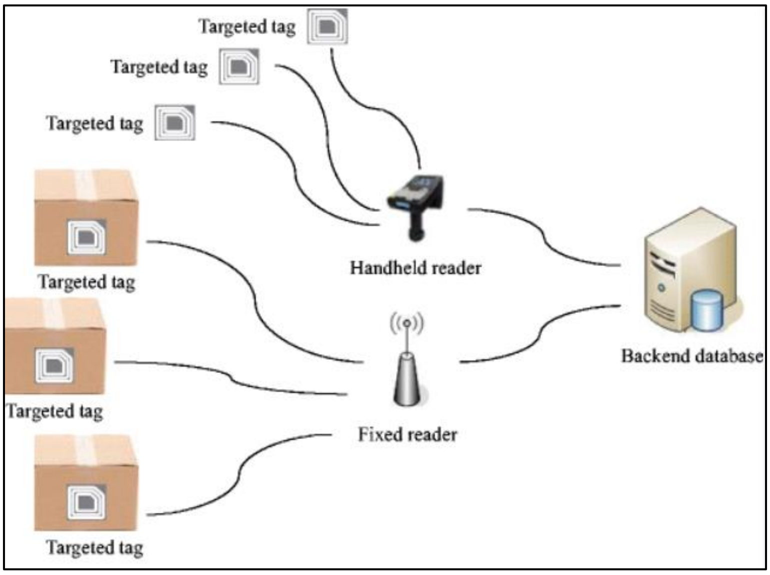
One of the primary advantages of RFID tag cards is their ability to enable swift and automated identification processes. By simply waving or tapping the card near an RFID reader, individuals can swiftly gain access to secured areas, make transactions, or check in/out of various facilities. This significantly reduces waiting times and enhances overall operational efficiency in environments such as offices, airports, hospitals, and educational institutions.

Moreover, the integration of RFID technology offers enhanced security features compared to traditional identification systems. The unique identifiers encoded within the RFID tags ensure that each cardholder is uniquely identified, minimizing the risk of identity theft and unauthorized access. Additionally, the real-time tracking capabilities of RFID tag cards allow for better asset management, preventing loss, theft, or misplacement of valuable resources.

# INTRODUCTION

This report presents an overview of an IoT-based RFID (Radio Frequency Identification) object tracker system that integrates with Google Sheets. The system combines RFID technology, Internet of Things (IoT) connectivity, and cloud-based data management to enable real-time tracking and monitoring of objects equipped with RFID tags. By utilizing Google Sheets as a central data repository, users can access and analyze the collected data efficiently. This report outlines the key components, functionalities, and benefits of the system.

The RFID Card Tag Project was implemented in both hardware and software modes. For hardware components, a breadboard was used to connect the microprocessor and the necessary components to the computer. This connection was essential for uploading the code and ensuring successful interactions with the Google sheet. The RFID card itself is embedded with a microchip, which works in conjunction with RFID readers and a backend system to enable identification and data transfer. The RFID system is connected to the NodeMCU board, and the NodeMCU is further connected to the computer system. This setup facilitates the uploading of code onto the microprocessor, which is connected to the RFID system. By establishing this connection, the microprocessor can detect the RFID card tag. Once the tag is detected, the microprocessor initiates the process of uploading the corresponding data entry onto the Excel sheet.



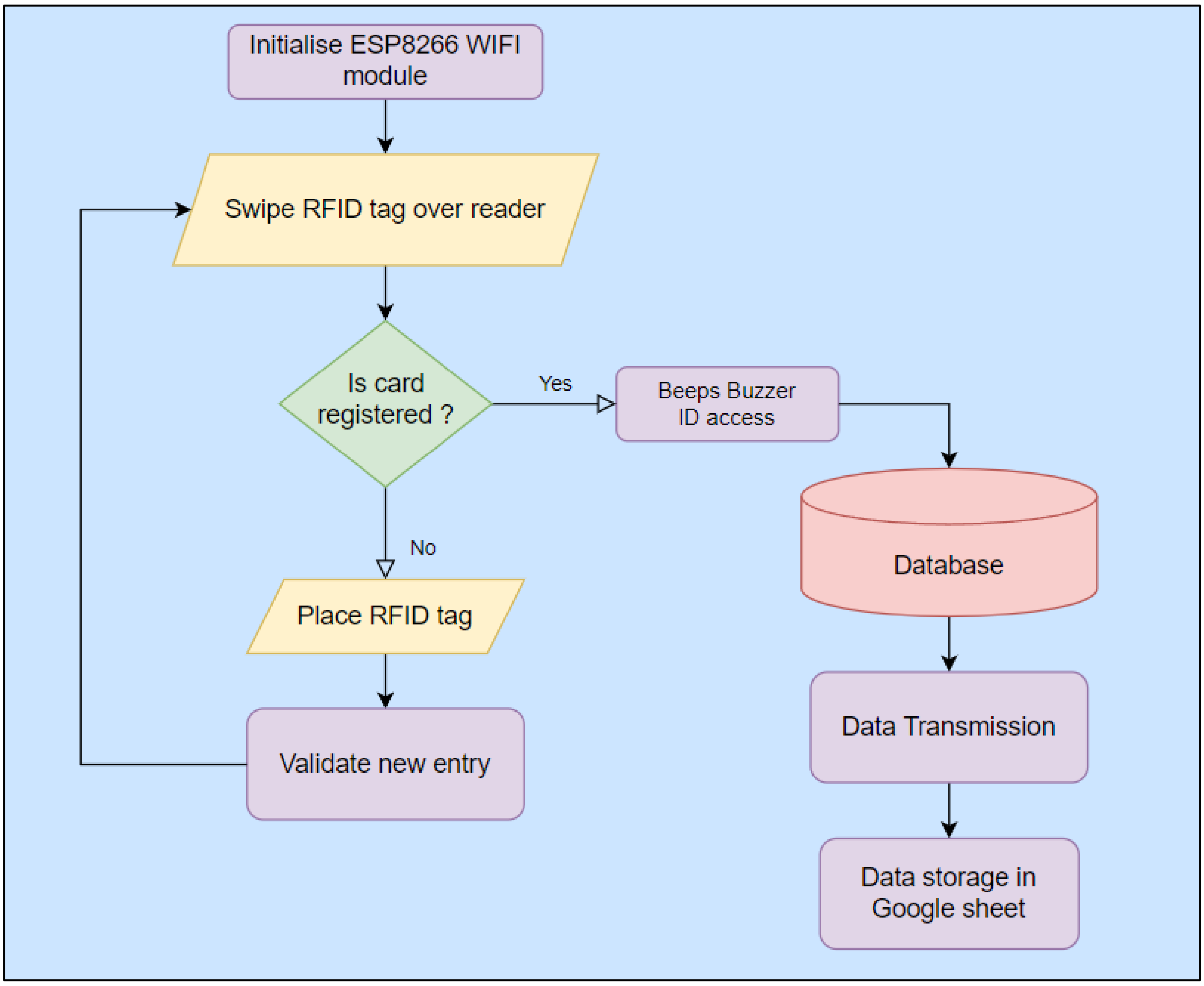

## IoT Concept

IoT stands for Internet of Things. It is a technology for a system of interconnected smart devices over a wireless network without human involvement. The IoT devices maybe sensors, actuators, computer devices or software. IoT enables all devices to collect, transfer the data to the cloud, processes that data on a certain software program, and automate the tasks. So, it is a network of physical connected devices for exchanging data after being connected over the internet. So, it operates remotely at remote place. There are 4 key components to integrate into the complete IoT:

I. Devices or sensors: It maybe RFID, temperature Sensor.
II. Connectivity: The sensors or devices connected to the cloud. It can be connected to the internet via cellular, WIFI, Bluetooth.
III. Data processing: It processes the sensor's data in the cloud using certain software program.
IV. User interface: This will be the end user. It will work like an alert about the status or the situation of the devices or give the notification through texts or emails. So, in this way, IoT will work.

# ANALYSIS

The working of an RFID tag card system can be understood by the following flowchart:



The workflow of the IoT-based RFID object tracker system can be summarized as follows:

1. Tag Initialization: Each RFID tag card is uniquely identified and initialized with a specific identifier or serial number during the manufacturing process. This identifier is stored in the microchip's memory and cannot be altered.

2. Electromagnetic Field Activation: When an RFID tag card comes within the range of an RFID reader, the reader emits a radio frequency electromagnetic field.

3. Powering the Tag: The electromagnetic field emitted by the RFID reader induces a small electric current in the antenna of the RFID tag card. This current provides the necessary power to the microchip embedded in the card.

4. Data Transmission: Once powered, the microchip becomes active and starts transmitting the stored data back to the RFID reader. The data can include the unique identifier, additional information about the cardholder, or any other relevant data stored on the card's memory.

5. Reception and Processing: The RFID reader receives the transmitted data from the tag card's microchip. It decodes the information and forwards it to the backend system for further processing and analysis.

6. Authentication and Response: The backend system compares the received data with the stored records in its database. It authenticates the cardholder's identity and triggers an appropriate response, such as granting access, initiating a transaction, or recording attendance.

7. Communication Feedback: The RFID reader sends a feedback signal to the RFID tag card, indicating that the data transmission was successful. This feedback can be used to trigger visual or auditory indicators, such as a green light or a beep, to inform the cardholder that the identification process is complete.

Overall, RFID tag cards simplify and streamline identification processes by wirelessly transmitting data to RFID readers. Their efficiency, speed, and ability to work in various environments make them suitable for a wide range of applications, including access control, asset tracking, inventory management, and contactless payment systems.

# COMPONENTS

## Hardware:

### 1. Nodemcu:

The hardware of NodeMCU consists of the following key components and features:

1. ESP8266 Wi-Fi Module: The core component of NodeMCU is the ESP8266 Wi-Fi module. It integrates a microcontroller, Wi-Fi capabilities, and necessary peripherals onto a single chip. The module supports 802.11 b/g/n Wi-Fi standards, offering wireless connectivity for Internet of Things (IoT) applications.

2. Microcontroller: The ESP8266 module houses a powerful 32-bit microcontroller with a clock speed of 80 MHz. It is based on the Tensilica Xtensa LX106 architecture and provides sufficient processing power for running applications and handling Wi-Fi communication.

3. Flash Memory: NodeMCU comes with varying amounts of flash memory, typically ranging from 4MB to 16MB. This non-volatile memory is used for storing the firmware, program code, and data.

4. GPIO Pins: NodeMCU features a number of General Purpose Input/Output (GPIO) pins, which allow the connection of external devices and components. These pins can be used as digital input/output, PWM (Pulse-Width Modulation) outputs, or for specialized functions such as I2C or SPI communication.
   Some GPIO pins are:

3V3: This pin provides a regulated 3.3V output voltage, which can be used to power external components that require 3.3V.

GND: These pins are ground connections and are used as the reference for the circuit.

A0: This pin can be used as an analog input to read analog voltage values from external sensors or devices.

D0 to D8: These pins are GPIO pins and can be used for general digital input/output purposes. They support both digital input and output operations.

D9: This pin is a GPIO pin, but it also serves as the default hardware SPI (Serial Peripheral Interface) MOSI (Master Out Slave In) pin.

D10: This pin is a GPIO pin and serves as the default hardware SPI clock (SCLK) pin.

D11: This pin is a GPIO pin and serves as the default hardware SPI MISO (Master In Slave Out) pin.

D12: This pin is a GPIO pin and can be used for general digital input/output purposes.

D13: This pin is a GPIO pin and serves as the default hardware SPI chip select (CS) pin.

D14 (SCL): This pin is a GPIO pin and serves as the clock line for I2C (Inter-Integrated Circuit) communication.

D15 (SDA): This pin is a GPIO pin and serves as the data line for I2C communication.

RX: This pin is used for serial communication and serves as the receive pin.

TX: This pin is used for serial communication and serves as the transmit pin.

5. Analog Inputs: NodeMCU provides a few analog input pins, typically labeled as A0, A1, etc. These pins can read analog voltage values, enabling the interfacing of sensors and other analog devices.

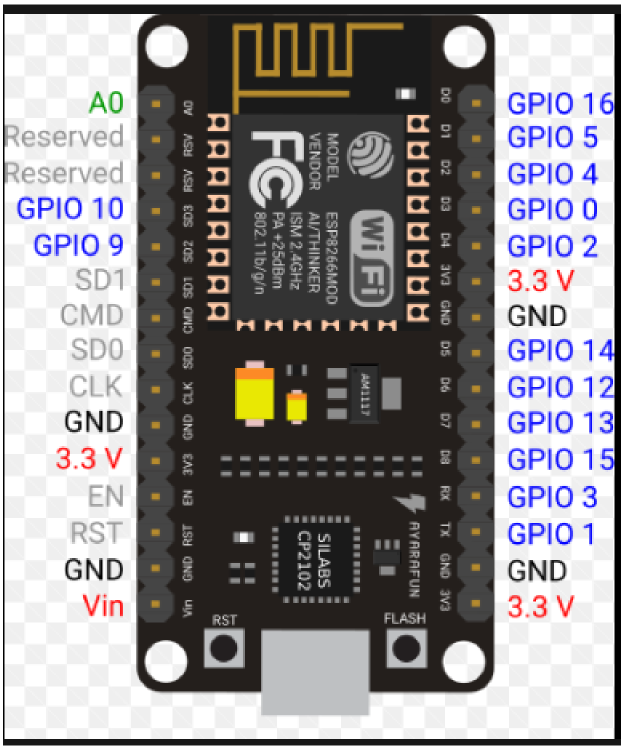
6. USB-to-Serial Converter: The NodeMCU development board incorporates a USB-to-serial converter, such as the popular CH340 or CP2102 chip. This converter facilitates easy programming and debugging of the board by connecting it to a computer via USB.

7. Power Regulator: NodeMCU includes a voltage regulator circuit that regulates the input power supply to the required voltage level for the ESP8266 module and other components. It usually operates on a 3.3V supply voltage.

8. Reset and Flash Buttons: NodeMCU boards often have reset and flash buttons for resetting the microcontroller and initiating firmware flashing or programming modes, respectively.

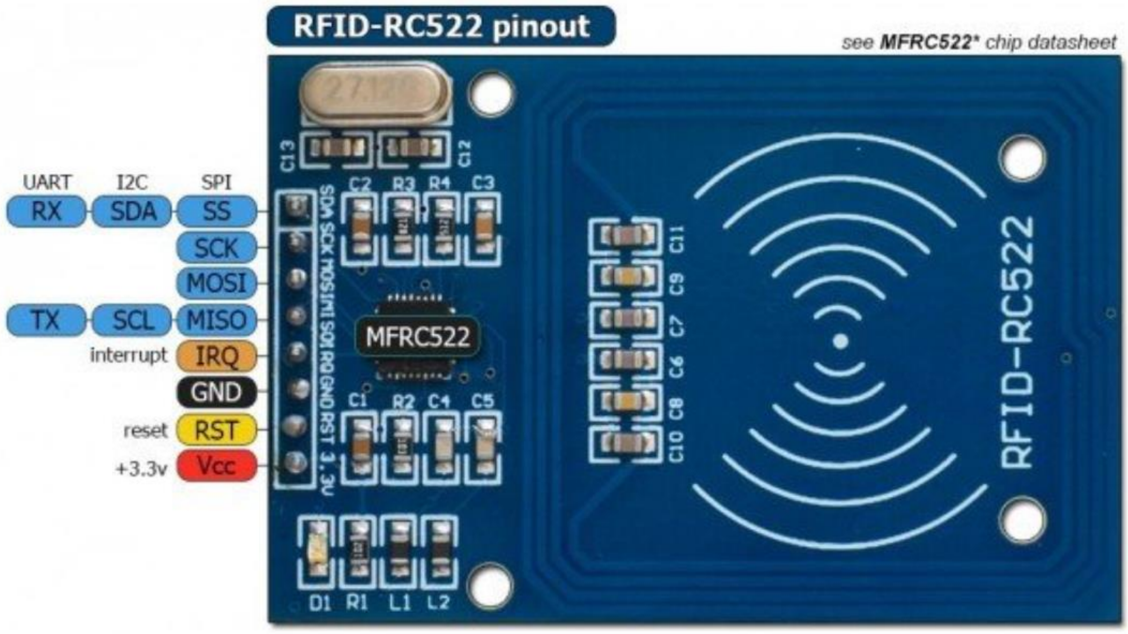
9. Wi-Fi Antenna: NodeMCU boards have an integrated antenna or an antenna connector for connecting an external antenna. This allows for better Wi-Fi signal reception and transmission range.

10. LEDs: Some NodeMCU boards have built-in LEDs that can be controlled by the microcontroller. These LEDs are often used for indicating power status, Wi-Fi connectivity, or user-defined purposes.




## 2. **RFID Reader:**

1. Antenna: The RFID reader contains an antenna that emits radio frequency signals and receives responses from RFID tags. The antenna's design and specifications determine the reading range and coverage area.
2. Transceiver: The transceiver module in the RFID reader is responsible for transmitting and receiving signals to and from the RFID tags. It handles the modulation and demodulation of the radio frequency signals.
3. Control Unit: The control unit manages the overall operations of the RFID reader. It includes a microcontroller or a dedicated circuitry to process data, control the antenna, and interface with other systems.
4. Power Supply: The RFID reader requires a power source, which can be a direct power connection or a battery. The power supply ensures that the reader operates reliably and provides sufficient power to the antenna and control unit.
5. Communication Interface: The RFID reader may have various communication interfaces, such as USB, Ethernet, RS-232, or Wi-Fi, enabling it to connect to external devices and transmit data to the system or host computer.
6. Additional Features: Depending on the specific model and application, RFID readers may include additional features like LCD displays, keypad interfaces, or LED indicators for user interaction and feedback.

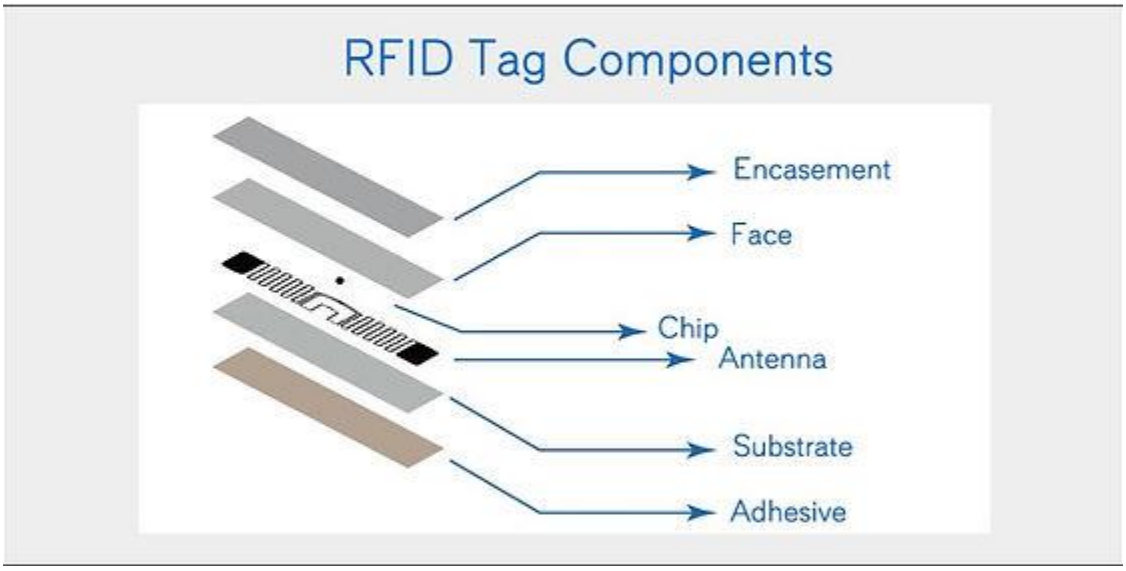



## 3.**RFID Tag:**


1. Microchip: The RFID tag contains a microchip or integrated circuit that stores data and performs communication functions. The microchip typically consists of a memory unit to store information and a small processor for processing and responding to signals.
2. Antenna: The RFID tag has an antenna, which is usually a small coil or printed circuit, connected to the microchip. The antenna receives radio frequency signals from the RFID reader and transmits the tag's response back to the reader.
3. Encapsulation: The microchip and antenna are encapsulated within a protective material, such as plastic or laminated paper, forming the

physical structure of the RFID tag. The encapsulation provides durability and protects the internal components from environmental factors.

4. Unique Identifier: Each RFID tag has a unique identifier or serial number programmed into the microchip's memory during manufacturing. This identifier allows for the individual identification and tracking of each tag.
5. Passive or Active Operation: RFID tags can be either passive or active. Passive tags do not have an internal power source and rely on the energy harvested from the RFID reader's electromagnetic field to operate. Active tags, on the other hand, have their power source, typically a battery, enabling them to transmit signals over longer distances.




## Software :

## 1. Arduino IDE:

The Arduino IDE (Integrated Development Environment) is a software platform used for programming Arduino boards. It provides a user-friendly interface and a simplified programming language based on C/C++ to develop and upload code to Arduino-compatible microcontrollers. In the IoT-based RFID object tracker system using NodeMCU and Google Sheets, the Arduino IDE plays a vital role in coding the NodeMCU board. It allows developers to write the necessary code to interface with RFID readers, establish Wi-Fi connectivity, and transmit data to Google Sheets. The Arduino IDE offers a wide range of libraries and examples that simplify the development process, making it easier to integrate the NodeMCU with the RFID readers, handle data processing, and implement the communication protocol with Google Sheets API. By utilizing the Arduino IDE, developers can write efficient and reliable code for the NodeMCU board, enabling seamless data transfer from the RFID object tracker system to the Google Sheets cloud-based application.

# METHODOLOGY AND IMPLEMENTATION

The methodology and implementation of the IoT-based RFID object tracker system involve the following steps:

1. Hardware Setup:

- NodeMCU Installation: The Arduino IDE is configured for NodeMCU development. The NodeMCU board is connected to the computer via a USB cable, and the required drivers are installed.
- RFID Reader Connection: The connections between the RFID readers and the NodeMCU board is established using appropriate interfaces such as UART or SPI. Proper power supply and configuration of the RFID readers are ensured.
- Wi-Fi Connectivity: The NodeMCU board is configured to connect to a Wi-Fi network. The necessary network credentials, such as SSID and password are provided, to enable seamless data transmission.

2. Software Development:

- Library Installation: The required libraries are installed in the Arduino IDE to interface with the RFID readers and facilitate communication with Google Sheets. Libraries such as MFRC522 and Google Sheets API are utilized.
- RFID Reader Integration: The user develops code to initialize and configure the RFID readers. The relevant library is utilized to read the UID of the RFID tags when objects are detected within range.
- Data Processing: Code is written to process the captured UID data from the RFID readers. This may involve converting the data into a suitable format, such as JSON or CSV, for efficient transmission.
- Google Sheets Integration: The user integrates the Google Sheets API into the code to enable data transmission and storage. Authentication of the NodeMCU with the Google Sheets API using

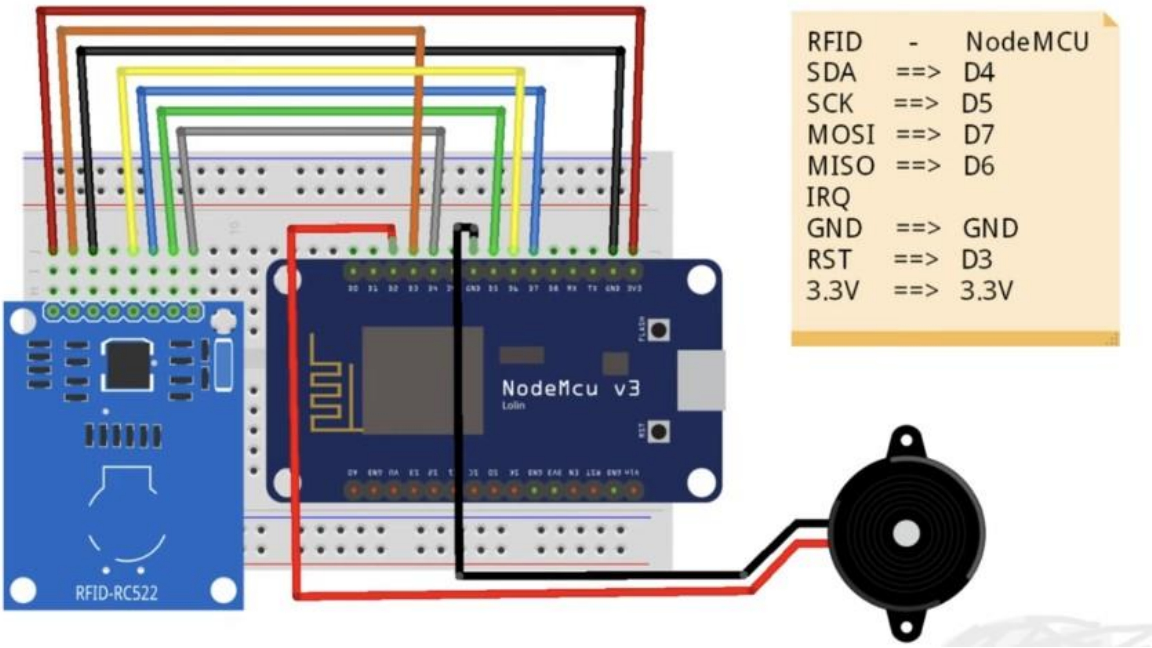
OAuth 2.0 credentials is performed, and the necessary access tokens are obtained.

- Data Transmission: The user implements code to establish a Wi-Fi connection with Google Sheets. The processed RFID tag data is sent to the designated Google Sheets spreadsheet using the API.

3. Google Sheets Configuration:

- Spreadsheet Creation: A new spreadsheet is created in Google Sheets to serve as the central database for storing the RFID tag data. The spreadsheet is customized with relevant columns to capture information such as date, time, object descriptions and location details.
- API Credential Generation: API credentials are generated within the Google Cloud Platform Console. The user obtains the required credentials, including the client ID, client secret, and access token, to authorize the NodeMCU's access to the Google Sheets API.

## Circuit Diagram



# Code:

## *CODE for Nodemcu on Arduino IDE*

```
#include <SPI.h>
#include <MFRC522.h>
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <WiFiClientSecureBearSSL.h>

// Fingerprint for demo URL, expires on September 11, 2023
const uint8_t fingerprint[20] = {0x56, 0xED, 0xC7, 0xDA, 0xBF, 0x51, 0x12,
0xC2, 0x79, 0x43, 0xC6, 0x01, 0xAB, 0xF7, 0x88, 0x98, 0x0F, 0x97, 0xB2, 0xB8};
// 56 ED C7 DA BF 51 12 C2 79 43 C6 01 AB F7 88 98 0F 97 B2 B8

#define RST_PIN  D3     // Configurable, see typical pin layout above
#define SS_PIN   D4     // Configurable, see typical pin layout above
#define BUZZER   D2     // Configurable, see typical pin layout above

MFRC522 mfrc522(SS_PIN, RST_PIN);  // Instance of the class
MFRC522::MIFARE_Key key;
ESP8266WiFiMulti WiFiMulti;
MFRC522::StatusCode status;

int blockNum = 2;

/* Create another array to read data from Block */
/* Length of buffer should be 4 Bytes more than the size of Block (16 Bytes)
*/
byte bufferLen = 20;
byte readBlockData[20];

String data2;
const String data1 =
"https://script.google.com/macros/s/AKfycbxwFt2CFAerv2hQxS-_u2CBjW0wuhAqUEB-
dnqZMLkfOsi7VGS2fq68cAk1vKls4xtE/exec?name=";

void setup()
{
  /* Initialize serial communications with the PC */
  Serial.begin(9600);
  // Serial.setDebugOutput(true);
```

```
  Serial.println();
  Serial.println();
  Serial.println();

  for (uint8_t t = 4; t > 0; t--)
  {
    Serial.printf("[SETUP] WAIT %d...\n", t);
    Serial.flush();
    delay(1000);
  }

  WiFi.mode(WIFI_STA);

  /* Put your WIFI Name and Password here */
  WiFiMulti.addAP("JioFiber-5Gh9w", "1234");

  /* Set BUZZER as OUTPUT */
  pinMode(BUZZER, OUTPUT);
  /* Initialize SPI bus */
  SPI.begin();
}

void loop()
{
  /* Initialize MFRC522 Module */
  mfrc522.PCD_Init();
  /* Look for new cards */
  /* Reset the loop if no new card is present on RC522 Reader */
  if ( ! mfrc522.PICC_IsNewCardPresent())
  {
    return;
  }
  /* Select one of the cards */
  if ( ! mfrc522.PICC_ReadCardSerial())
  {
    return;
  }
  /* Read data from the same block */
  Serial.println();
  Serial.println(F("Reading last data from RFID..."));
  ReadDataFromBlock(blockNum, readBlockData);
  /* If you want to print the full memory dump, uncomment the next line */
  //mfrc522.PICC_DumpToSerial(&(mfrc522.uid));

  /* Print the data read from block */
  Serial.println();
  Serial.print(F("Last data in RFID:"));
```

```
  Serial.print(blockNum);
  Serial.print(F(" --> "));
  for (int j=0 ; j<16 ; j++)
  {
    Serial.write(readBlockData[j]);
  }
  Serial.println();
  digitalWrite(BUZZER, HIGH);
  delay(200);
  digitalWrite(BUZZER, LOW);
  delay(200);
  digitalWrite(BUZZER, HIGH);
  delay(200);
  digitalWrite(BUZZER, LOW);

  // wait for WiFi connection
  if ((WiFiMulti.run() == WL_CONNECTED))
  {
    std::unique_ptr<BearSSL::WiFiClientSecure>client(new
BearSSL::WiFiClientSecure);

    client->setFingerprint(fingerprint);

    data2 = data1 + String((char*)readBlockData);
    data2.trim();
    Serial.println(data2);

    HTTPClient https;
    Serial.print(F("[HTTPS] begin...\n"));
    if (https.begin(*client, (String)data2))
    {
      // HTTP
      Serial.print(F("[HTTPS] GET...\n"));
      // start connection and send HTTP header
      int httpCode = https.GET();

      // httpCode will be negative on error
      if (httpCode > 0)
      {
        // HTTP header has been send and Server response header has been
handled
        Serial.printf("[HTTPS] GET... code: %d\n", httpCode);
        // file found at server
      }
      else
      {
        Serial.printf("[HTTPS] GET... failed, error: %s\n",
https.errorToString(httpCode).c_str());
```

```
      }
      https.end();
      delay(1000);
    }
    else
    {
      Serial.printf("[HTTPS} Unable to connect\n");
    }
  }
}

void ReadDataFromBlock(int blockNum, byte readBlockData[])
{
  /* Prepare the key for authentication */

  for (byte i = 0; i < 6; i++)
  {
    key.keyByte[i] = 0xFF;
  }
  /* Authenticating the desired data block for Read access using Key A */
  status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, blockNum,
&key, &(mfrc522.uid));

  if (status != MFRC522::STATUS_OK)
  {
     Serial.print("Authentication failed for Read: ");
     Serial.println(mfrc522.GetStatusCodeName(status));
     return;
  }
  else
  {
    Serial.println("Authentication success");
  }

  /* Reading data from the Block */
  status = mfrc522.MIFARE_Read(blockNum, readBlockData, &bufferLen);
  if (status != MFRC522::STATUS_OK)
  {
    Serial.print("Reading failed: ");
    Serial.println(mfrc522.GetStatusCodeName(status));
    return;
  }
  else
  {
    Serial.println("Block was read successfully");
  }
}
```

## ***Google Appscript Code:***

```
function doGet(e) {
  Logger.log( JSON.stringify(e) );
  var result = 'Ok';
  if (e.parameter == 'undefined') {
    result = 'No Parameters';
  }
  else {
    var sheet_id = '1A4D4pxzIDoms_OTQDwWXduZnHQT7B0mv_Fosg0b4D44';  // Spreadsheet ID
    var sheet = SpreadsheetApp.openById(sheet_id).getActiveSheet();
    var newRow = sheet.getLastRow() + 1;
    var rowData = [];

    var Curr_Date = new Date();
    rowData[0] = Curr_Date; // Date in column A

    var Curr_Time = Utilities.formatDate(Curr_Date, "Asia/Kolkata", 'HH:mm:ss');
    rowData[1] = Curr_Time; // Time in column B

    for (var param in e.parameter) {
      Logger.log('In for loop, param=' + param);
      var value = stripQuotes(e.parameter[param]);
      Logger.log(param + ':' + e.parameter[param]);
      switch (param) {
        case 'name':
          rowData[2] = value; // Object Name in column C
          result = 'Object Name Written on column C';
          break;
        default:
          result = "unsupported parameter";
      }
    }
    Logger.log(JSON.stringify(rowData));
    var newRange = sheet.getRange(newRow, 1, 1, rowData.length);
    newRange.setValues([rowData]);
  }
  return ContentService.createTextOutput(result);
}
function stripQuotes( value ) {
  return value.replace(/^["']|['"]$/g, "");
}
```

# Conclusion:

The IoT-based RFID object tracker system using Google Sheets and NodeMCU provides a reliable and scalable solution for tracking and monitoring objects in real-time. By leveraging RFID technology and cloud-based storage, the system offers benefits such as accurate inventory management, enhanced asset security, and data-driven insights. This system can find applications in various industries, including logistics, warehousing, healthcare, and retail, to improve operational efficiency and asset utilization.

## Benefits and Applications:

a. Inventory Management: Real-time tracking of objects helps streamline inventory management processes, reducing errors and improving efficiency.
b. Supply Chain Optimization: The system enables accurate tracking of objects throughout the supply chain, enhancing visibility and reducing the risk of loss or theft.
c. Asset Tracking: Valuable assets can be monitored and tracked, ensuring their proper utilization and minimizing the chances of misplacement.
d. Research and Data Collection: The system can be used in research studies or data collection projects where tracking and analysis of objects are required.

## References:

https://ieeexplore.ieee.org/document/9616967
https://www.researchgate.net/publication/315808980_RFID-based_Object_Detection_System_in_IoT
https://www.irjet.net/archives/V6/i2/IRJET-V6I2180.pdf