Radio Frequency Identification

Introduction to Wireless Communication

- **Transmission**: The sender or transmitter encodes data into electromagnetic waves. This process can involve various modulation techniques where the properties of the wave, such as amplitude, frequency, or phase, are varied in accordance with the data signal.
- **Propagation**: These electromagnetic waves travel through the air (or even through vacuum) from the transmitter to the receiver. The waves can pass through obstacles, though they may weaken or be reflected depending on the material they encounter.
- **Reception**: The receiver captures the electromagnetic waves using an antenna and decodes the data from the waves. This involves demodulation, where the received signal is converted back into its original data form.

Inside the Transmitter

Oscillator: Generates a carrier wave at a specific frequency.

Modulator: Imprints the data onto the carrier wave by varying its amplitude, frequency, or phase.

Amplifier: Boosts the modulated signal's power to ensure it can travel the required distance.

Antenna: Converts the electrical signal into electromagnetic waves and radiates them into the air.

Inside the Receiver

Antenna: Captures the electromagnetic waves from the air and converts them back into electrical signals.

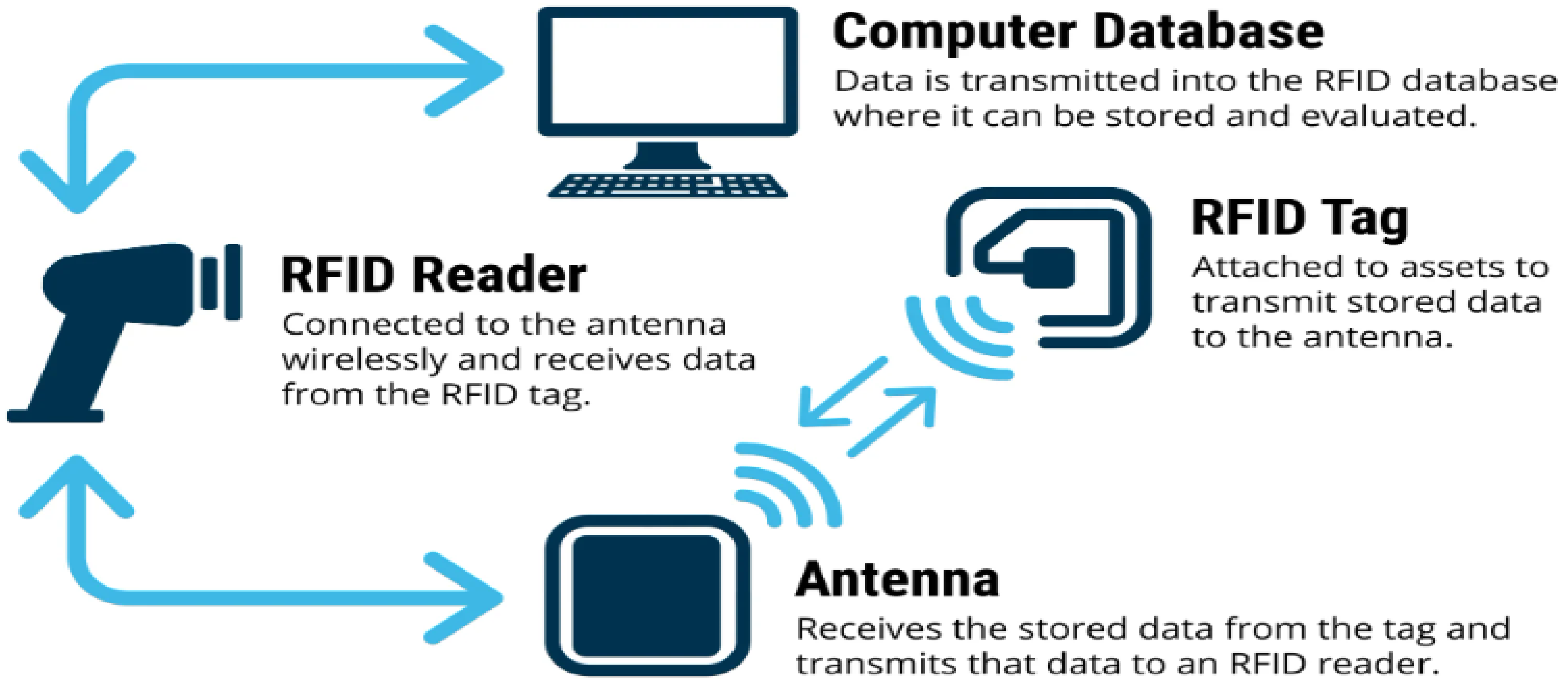
Amplifier: Boosts the weak received signal to a usable level.

Demodulator: Extracts the data from the modulated carrier wave by reversing the modulation process.

Filter: Removes any noise or unwanted signals, ensuring a clean data signal.

Decoder: Interprets the extracted data into a format that can be understood by the receiving device.

Basic RFID System



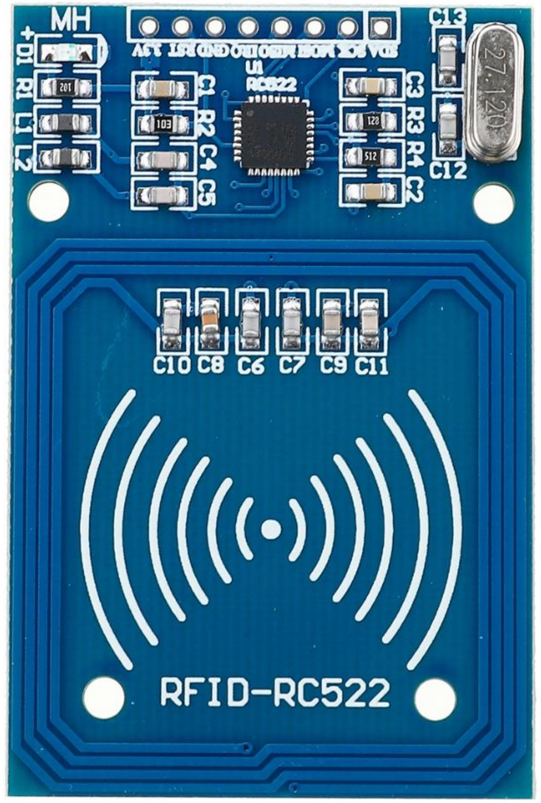
Computer Database
Data is transmitted into the RFID database where it can be stored and evaluated.

RFID Reader
Connected to the antenna wirelessly and receives data from the RFID tag.

RFID Tag
Attached to assets to transmit stored data to the antenna.

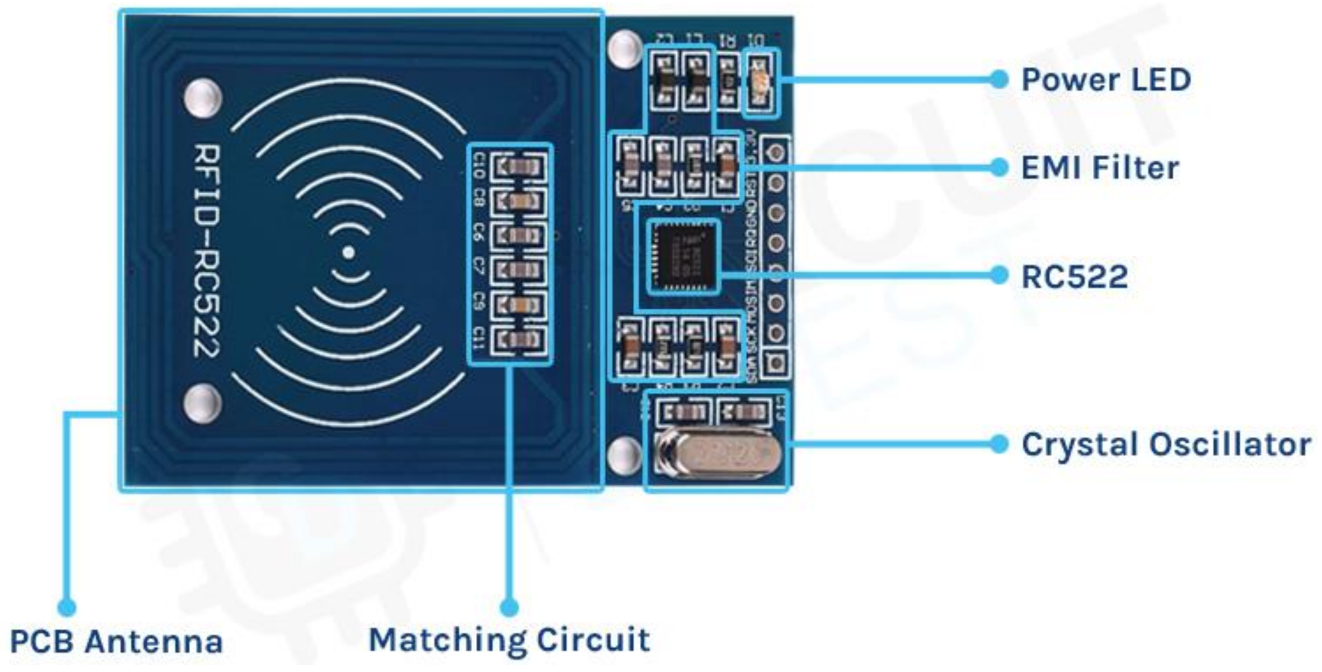
Antenna
Receives the stored data from the tag and transmits that data to an RFID reader.

RFID Reader and Tags



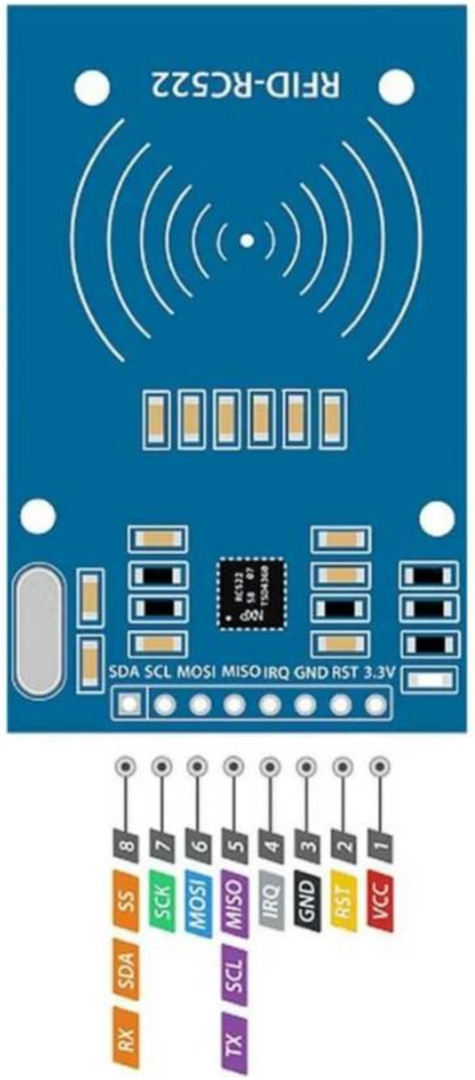
Inside the Reader

RFID Antenna: Emits and receives radio waves for communication with RFID tags.
RFID Reader Module: Circuitry for generating and processing RF signals, often including microcontrollers.
Microcontroller/Processor: Controls reader operation, manages data processing, and interfaces with external devices.
Memory: Stores temporary data, configuration settings, or logged information.
Communication Interfaces: Connects to external devices via UART, USB, Ethernet, or Wi-Fi.
Power Supply Circuitry: Regulates power from batteries, AC mains, or PoE.
Control and Indicator Interfaces: Buttons, LEDs, displays for user interaction and status indication.



RC522

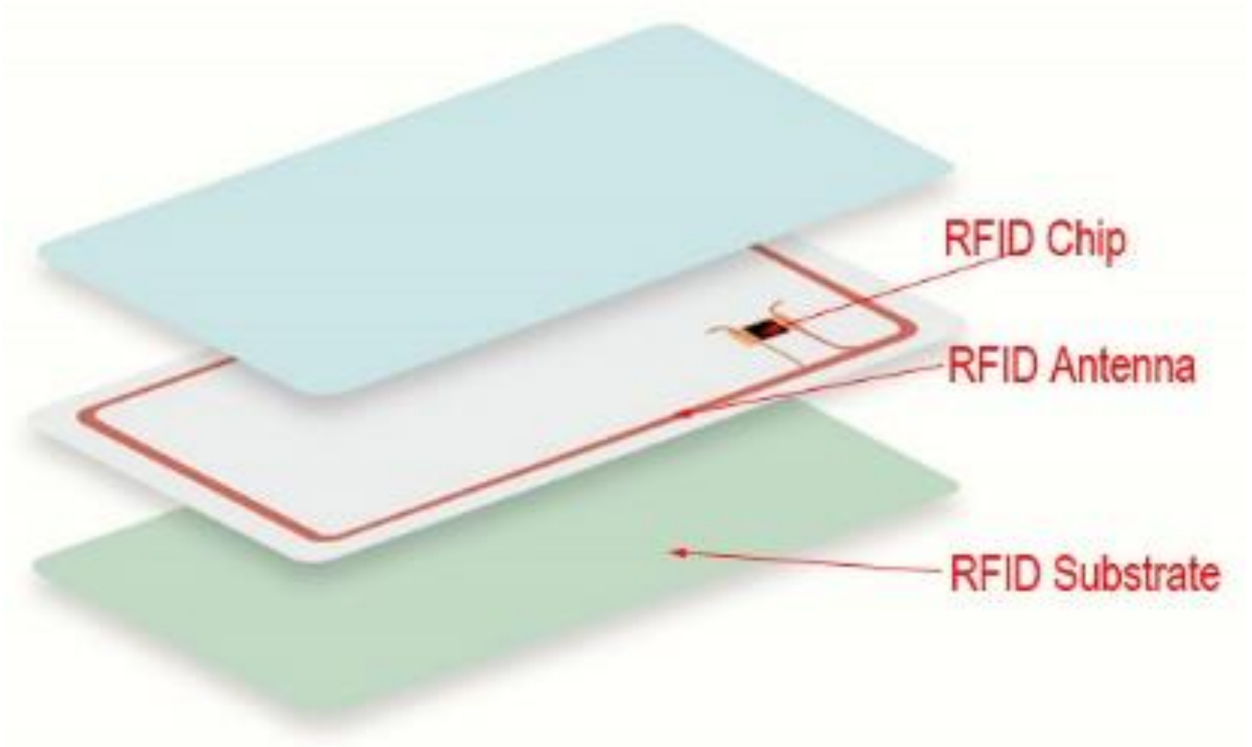
Pin Name	Details
1 - VCC	Power supply (3.3V)
2 - RST	Reset signal - used to reset the device in case of an error.
3 - GND	Ground
4 - IRQ	Interrupt pin. The device can go into sleep mode to save power. So, the IRQ helps to wake it.
5 - MISO	Master In Slave Out. It sends data from the RFID reader to the SPI master device.
6 - MOSI	Master Out Slave In. It receives data from the SPI master device.
7 - SCK	Serial Clock. This is the SPI clock pin used to provide clock signal where needed.
8 - SS	Slave Select - used to select the RC522 module as the active slave device.



Inside the Tag

RFID CHIP (IC)
The IC has a logic unit that makes decisions and provides memory to store data. A part of the IC is dedicated to controlling power. The processing logic implements the communication protocol. It also is used to modulate/demodulate signals and to encode/decode digital bits during the communication between the interrogator and the tag. The memory on the IC may be divided in different blocks, called banks.

TAG ANTENNA
The antenna is the largest part of the tag and is connected to the tag IC. It receives the signals from the reader and, depending on tag type, it either transmits or reflects the received signal back. For active tags, it transmits the signals, and for semi-passive and passive tags, it reflects the signals. For passive tags, the antenna also collects power from the radio waves and supplies it to the IC.



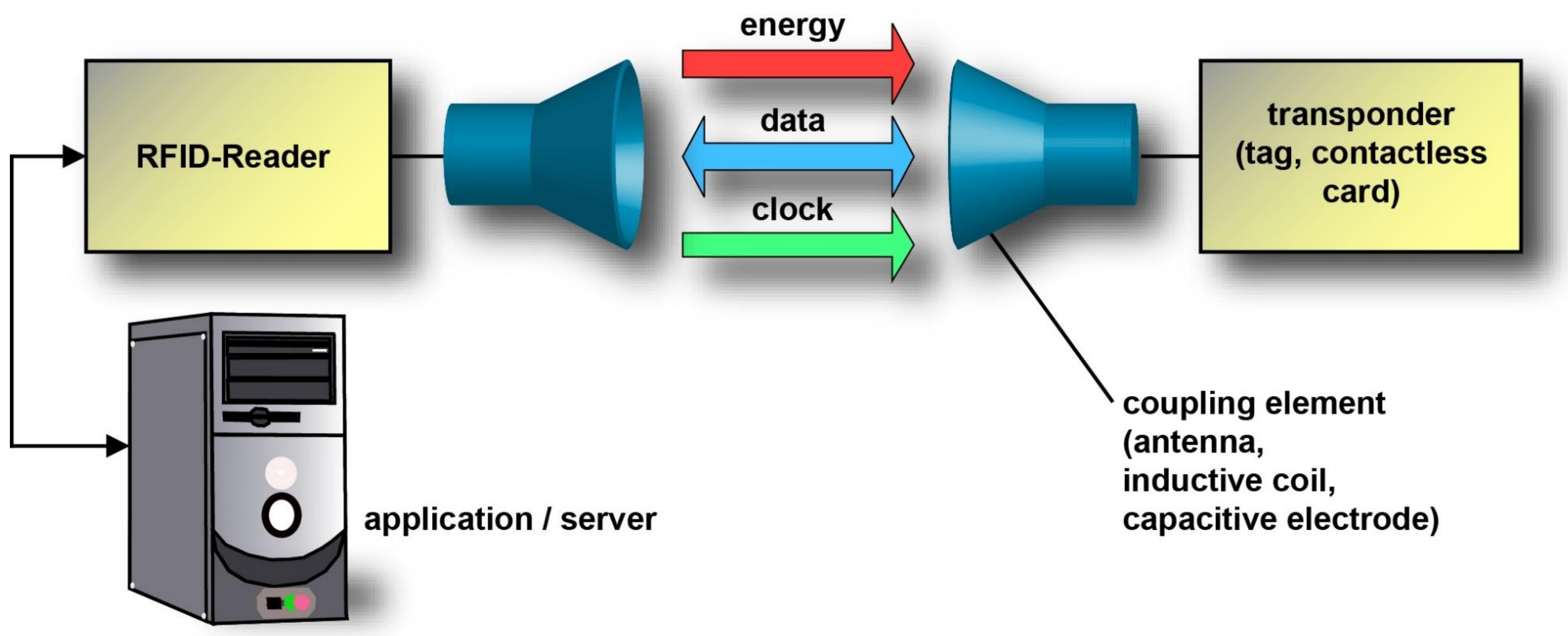
Memory

Inside an RFID tag's memory, information is organized into sectors and blocks. Each sector consists of several blocks, and each block contains a set amount of data storage capacity, typically 16 bytes.

Sectors and Blocks: The memory is divided into sectors, each containing several blocks (for example, four blocks per sector in many common RFID tags like the MIFARE Classic). Each block can store data, and the organization into sectors and blocks allows for efficient data management and access control. A block may be a read-only type, a write-only-one-time type, or a write-many-times type.

Data Storage and Access Control: The blocks within a sector can store different types of data, such as user information, configuration settings, or identification numbers. Additionally, each sector has its own access keys and permissions, allowing for secure read and write operations. This means that certain sectors can be locked or made read-only, depending on the application's requirements.

Working Principle of RFID


energy
data
clock
RFID-Reader
transponder (tag, contactless card)
application / server
coupling element (antenna, inductive coil, capacitive electrode)

Interfacing RFID with Arduino

For **interfacing the RC522 RFID module with the Arduino**, we will be using the SPI.

Libraries Required: `SPI.h` and **`MFRC522.h` -** The MFRC522 library helps to decode and encode the incoming data from the RFID module and SPI helps to establish the SPI communication. These two libraries are dependent on each other.

Circuit Initialization: `MFRC522 mfrc522(SS_PIN, RST_PIN);`

SPI Initialization: `SPI.begin();`

Module Initialization: `mfrc522.PCD_Init();`

Reading from a Tag:

```
#include <SPI.h>
#include <MFRC522.h>
#define RST_PIN         5       // Reset pin for MFRC522 module
#define SS_PIN          53      // SS pin for MFRC522 module

MFRC522 mfrc522(SS_PIN, RST_PIN);  // Create MFRC522 instance
void setup() {
  Serial.begin(9600);   // Initialize serial communication
  SPI.begin();          // Initialize SPI bus
  mfrc522.PCD_Init();   // Initialize MFRC522 RFID module
}
void loop() {
  if (!mfrc522.PICC_IsNewCardPresent()) {
    return;}
  if (!mfrc522.PICC_ReadCardSerial()) {
    return;}
  // Print card UID
  Serial.print("Card UID:");
  for (byte i = 0; i < mfrc522.uid.size; i++) {
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
    Serial.print(mfrc522.uid.uidByte[i], HEX);}
  Serial.println();
  // Halt PICC
  mfrc522.PICC_HaltA();
  // Stop encryption on PCD
  mfrc522.PCD_StopCrypto1();
}
```

Writing to a Tag:

```
#include <SPI.h>
#include <MFRC522.h>

#define RST_PIN         5       // Reset pin for MFRC522 module
#define SS_PIN          53      // SS pin for MFRC522 module

MFRC522 mfrc522(SS_PIN, RST_PIN);  // Create MFRC522 instance

void setup() {
  Serial.begin(9600);   // Initialize serial communication
  SPI.begin();          // Initialize SPI bus
  mfrc522.PCD_Init();   // Initialize MFRC522 RFID module

  Serial.println("Place your card near the reader to store the name.");
}

void loop() {
  if (!mfrc522.PICC_IsNewCardPresent())
    return;

  if (!mfrc522.PICC_ReadCardSerial())
    return;

  String name;
  Serial.println("Card detected. Enter the name:");
  while (Serial.available()) {
    char c = Serial.read();
    if (c != '\n' && c != '\r')
      name += c;
    else
      break;
  }

  mfrc522.PICC_HaltA();

  if (name.length() > 0) {
    byte buffer[16];   // Create a buffer to store data in the card

    name.toCharArray((char *)buffer, sizeof(buffer));

    if (mfrc522.MIFARE_Write(1, buffer, sizeof(buffer))) {
      Serial.println("Name stored successfully.");
    } else {
      Serial.println("Failed to store name. Please try again.");
    }
  } else {
    Serial.println("Invalid name. Please try again.");
  }

  delay(2000);  // Delay to avoid repeated reads
}
```

RFID Implants