SMART RFID ATTENDANCE & SECURITY SYSTEM

Investigatory Project (Physics)
Academic Year: 2025–26
School: Vasant Valley School
Student Name: Kushal Sachdeva
Class / Section: 11B
Submission Date: 14 November 2025

Certificate

This is to certify that the investigatory project titled **"Smart RFID Attendance & Security System"** has been carried out by **Kushal Sachdeva, 11B,** under my guidance. The work is original and has not been submitted elsewhere.

Teacher/Guide:_____
Signature: _____
Date: _____

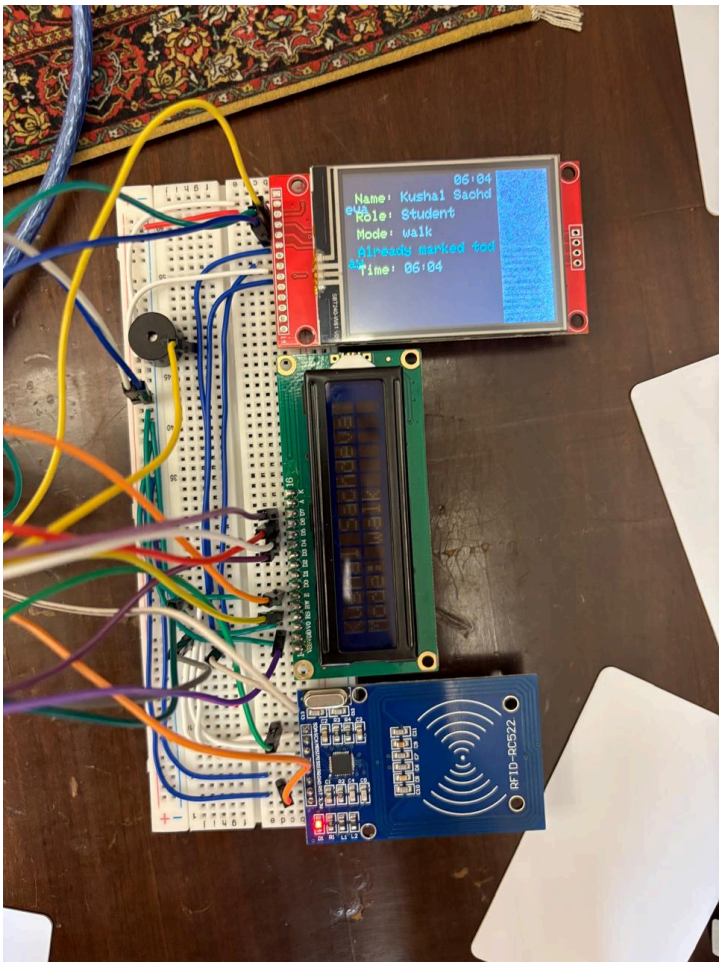
Acknowledgements

I would like to express my gratitude to my Physics teacher **[Teacher's Name]** for invaluable guidance throughout the project. I also thank Ms. Aggarwal and the school administration for providing resources and laboratory support. My classmates and family encouraged me to refine the idea and offered feedback during testing. I am particularly grateful to the KiCad and ESP32 online communities, whose resources helped me understand PCB design principles and firmware development.

1. Introduction & Motivation

In schools and institutions, attendance marking is still predominantly manual, leading to loss of instructional time and potential inaccuracies. The **Smart RFID Attendance & Security System** revolutionizes this by integrating **radio-frequency identification**, **wireless communication**, and **embedded systems** into one efficient model. It also ensures student safety by verifying departure modes and tracking entry-exit events automatically.

The project combines **practical electronics, physics of electromagnetic induction, and digital system design**. To enhance reliability and understanding, I designed the entire system's PCB from scratch in **KiCad**, allowing me to understand how circuits are physically realized. The system represents an intersection of physics, engineering, and computer science, showcasing how theory directly translates into technological solutions.

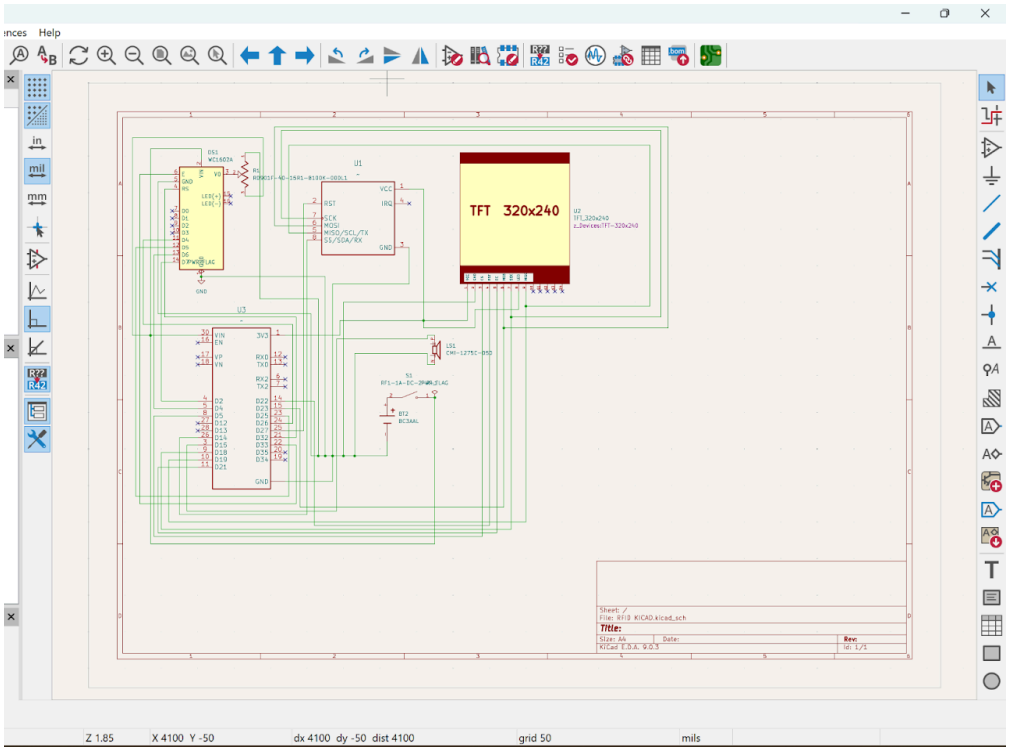


2. Objective

- To design a low-cost, efficient attendance system using RFID and ESP32 microcontroller.
- To fabricate a custom **PCB** in KiCad that integrates all sensors, displays, and power modules in a compact layout.
- To explore the **physics of electromagnetic induction, resonance, and wave propagation** in RFID systems.

- To write and implement firmware that automates attendance marking, late detection, and transport verification.

- To analyze data through measurements, graphs, and performance evaluation.



3. Materials Required

Hardware Components

- **ESP32 Development Board:** Central control unit (dual-core MCU, Wi-Fi & BLE enabled).

- **RC522 RFID Module:** Operates at 13.56 MHz; communicates via SPI.

- **TFT Display (2.4”):** Visual interface for feedback.

- **I²C 16×2 LCD:** Guard-facing display for transport mode.

- **Active Buzzer:** Audible confirmation.

- **Li-ion Battery Pack (3.7V):** Portable power source.

- **Voltage Regulator Module:** Converts 3.7V to 5V/3.3V as required.
- **Custom PCB:** Designed in KiCad with optimized routing and EMI control.
- **RFID Tags/Cards:** Passive 13.56 MHz MIFARE tags for demonstration.

Software Tools

- **KiCad** – Circuit design, simulation, and PCB layout.
- **Arduino IDE** – Firmware coding and uploading.
- **Wokwi** – ESP32 and RFID simulation environment.
- **Excel/Python** – Data analysis and graphing.

4. Physics Behind RFID and Electromagnetic Induction

RFID works on the **principles of electromagnetism**, particularly **Faraday's Law of Induction**, **mutual inductance**, and **resonance**.

When an alternating current flows through the reader coil, it creates a **time-varying magnetic field (B)**. A nearby tag coil experiences a changing magnetic flux, inducing an **electromotive force (EMF)**.

$$EMF = -N\frac{d\phi}{dt} = -N\frac{d(BA)}{dt}$$

Where:

- N = Number of turns in the coil
- Φ = Magnetic flux
- B = Magnetic field strength
- A = Coil area

The tag and reader are tuned to resonate at 13.56 MHz:

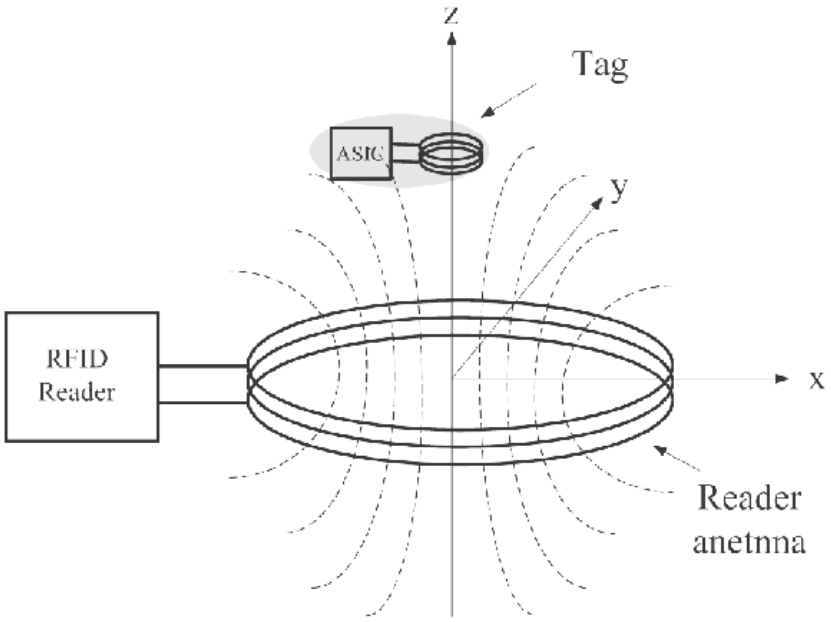
$$f = \frac{1}{2\pi\sqrt{LC}}$$

This ensures maximum power transfer through inductive coupling.

The reader's alternating magnetic field both **powers the passive tag** and **detects its modulated response**. Tags use **load modulation** — varying their internal load resistance — to transmit binary data back to the reader.

Wave Propagation and Near-Field Effects

The magnetic field strength decays with the cube of distance ($1/r^3$), characteristic of **near-field coupling**. Thus, RFID systems are limited to a few centimeters of range, ensuring controlled read zones.



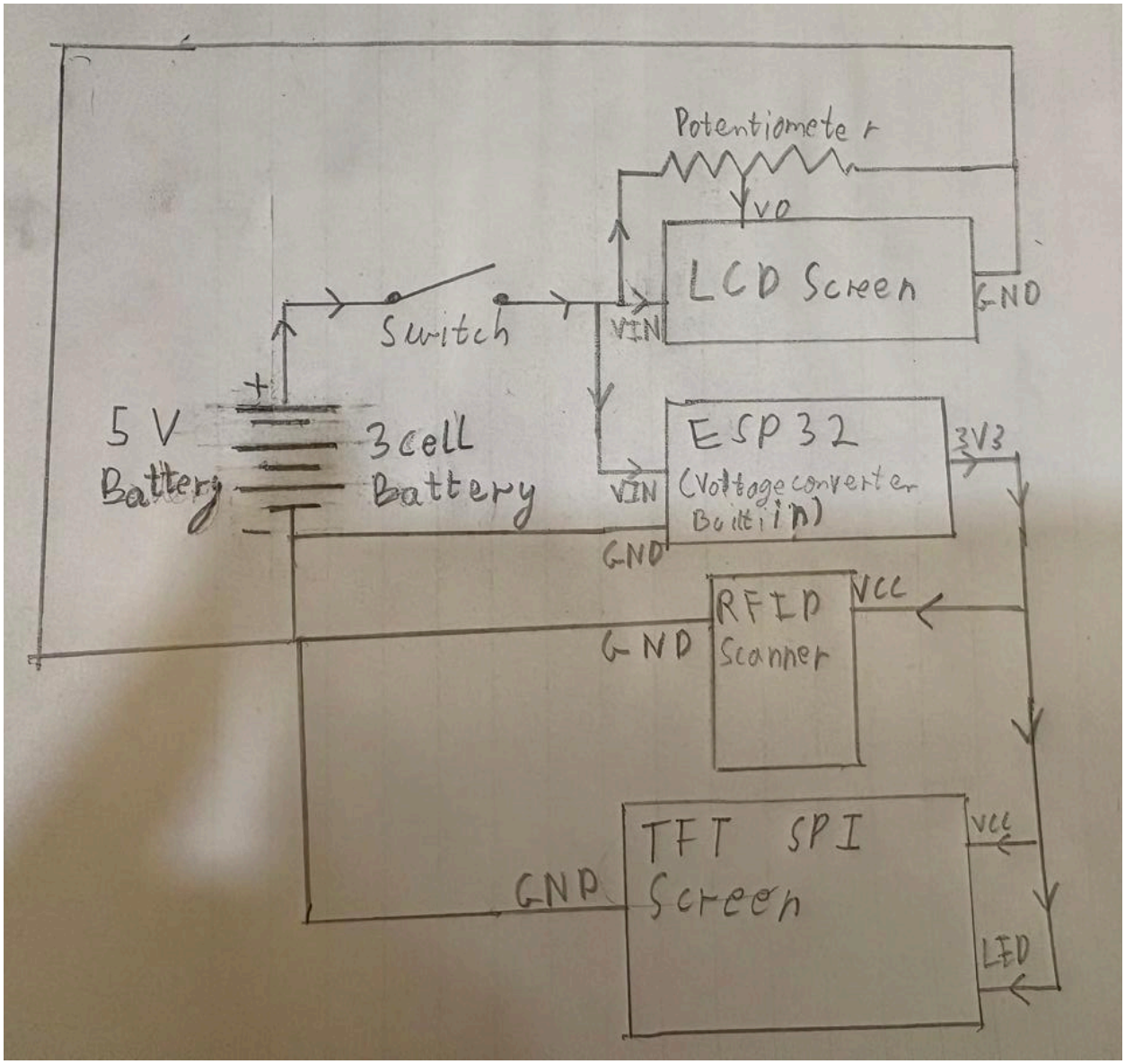
5. Circuit Theory and Electrical Principles

The circuit combines **analog and digital electronics**:

- The **RC522 module** uses an internal oscillator and modulator to transmit electromagnetic waves.
- **SPI communication** transfers digital data between ESP32 and RC522 using clock-synchronized pulses.
- The **TFT display** uses pixel addressing and PWM-controlled backlighting.
- The **LCD and buzzer** are controlled through I^2C and GPIO logic signals respectively.

- Power is stabilized through voltage regulation and decoupling capacitors that minimize noise.

Ohm's Law governs power delivery across the circuit: (V = IR). Using **Kirchhoff's Laws**, voltage drops and current divisions were calculated to ensure stable operation.



6. KiCad Circuit Design & PCB Layout

Designing the PCB was a key engineering step that deepened my understanding of electronic physics. In KiCad:

1. **Schematic Design:** Logical circuit of all components with labeled nets for SPI, I²C, and power lines.

2. **Footprint Assignment:** Each component assigned to a physical package (e.g., ESP32 header, RC522 connector).

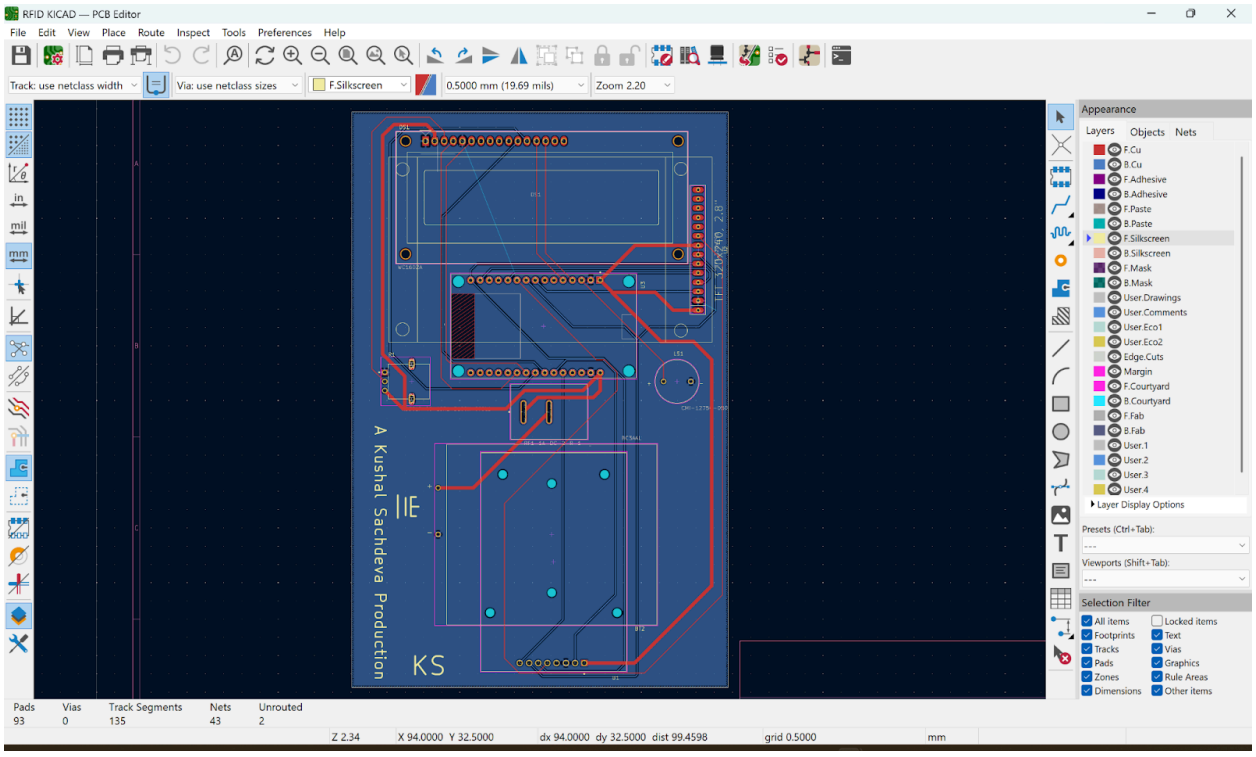
3. **PCB Layout:** Traces routed manually — high-current paths widened, SPI traces kept short for signal integrity.

4. **Ground Plane:** A full copper pour used as ground shield, reducing electromagnetic interference.

5. **Via Stitching:** Multiple vias connect top and bottom layers for consistent potential.

6. **3D Preview:** Verified orientation and spacing for easy soldering.

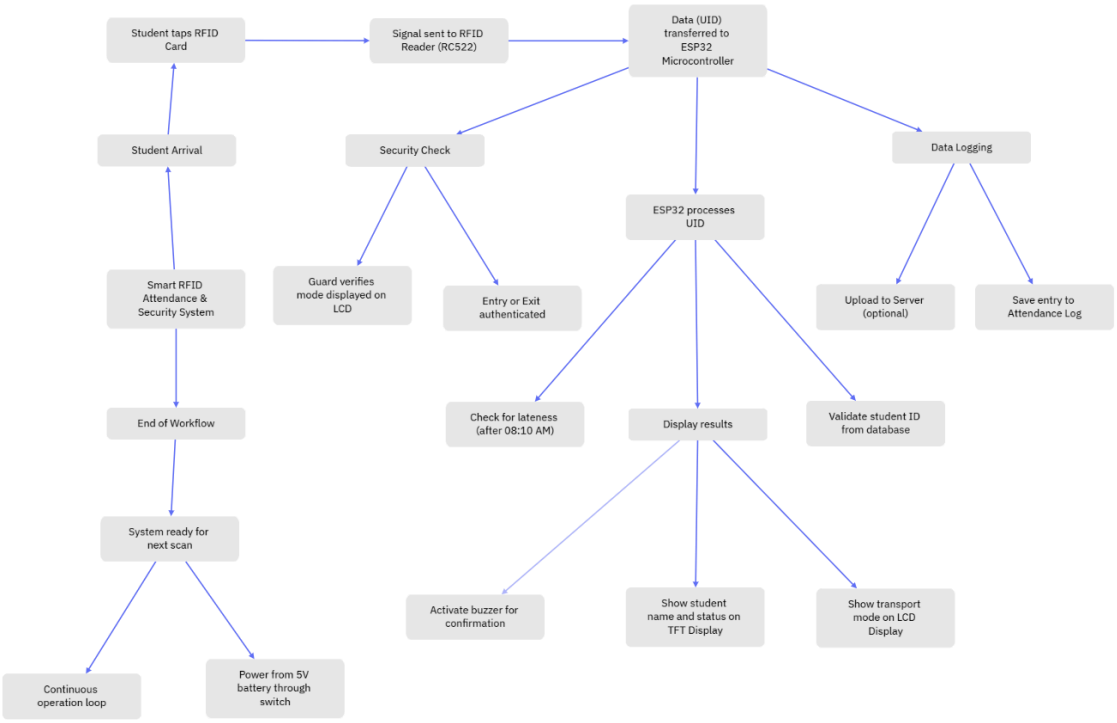
The finished PCB was printed and assembled with all components soldered neatly. This minimized the parasitic inductances and resistances common in breadboard prototypes.



7. System Architecture and Workflow

The workflow of the system integrates hardware and physics:

1. **Card Tap:** Electromagnetic induction between reader coil and card coil.
2. **Signal Conversion:** Reader converts analog signal to digital UID.
3. **Processing:** ESP32 processes data, checks lateness, and assigns state.
4. **Display & Output:** TFT and LCD display messages; buzzer emits confirmation tone.
5. **Data Logging:** Entry stored in local memory or transmitted via Wi-Fi.



8. Code Description and Implementation

The firmware is the brain of the project. It controls the RFID scanner, processes logic, and communicates with the user interface.

Main Features:

- Reads unique RFID UID from RC522
- Syncs time using NTP protocol

- Determines lateness by comparing with threshold time
- Displays details on TFT and LCD
- Logs data locally and over Wi-Fi

Code Snippet:

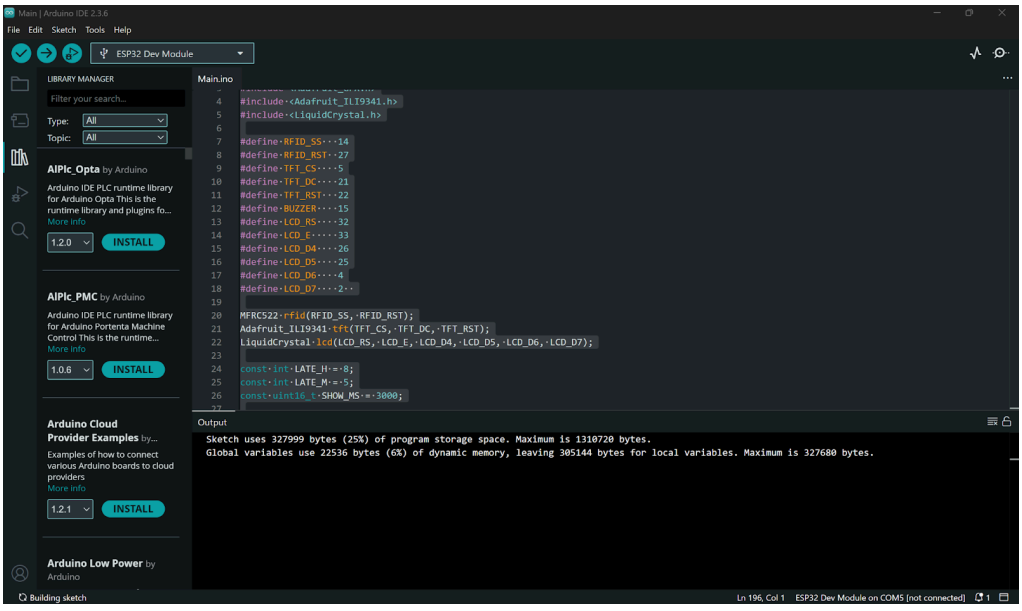
```
#include <SPI.h>
#include <MFRC522.h>
#include <Adafruit_GFX.h>
#include <Adafruit_ILI9341.h>
#include <LiquidCrystal.h>

#define RFID_SS   14
#define RFID_RST  27
#define TFT_CS    5
#define TFT_DC    21
#define TFT_RST   22
#define BUZZER    15
#define LCD_RS    32
#define LCD_E     33
#define LCD_D4    26
#define LCD_D5    25
#define LCD_D6    4
#define LCD_D7    2

MFRC522 rfid(RFID_SS, RFID_RST);
Adafruit_ILI9341 tft(TFT_CS, TFT_DC, TFT_RST);
LiquidCrystal lcd(LCD_RS, LCD_E, LCD_D4, LCD_D5, LCD_D6, LCD_D7);

const int LATE_H = 8;
const int LATE_M = 5;
const uint16_t SHOW_MS = 3000;

struct Person {
  const char* uid;
  const char* name;
  const char* role;
  const char* transport;
  int lastDayMarked;
};

```

```
Person roster[] = {
  {"B3 4B 6F 21","Kushal Sachdeva","Student","walk",-1},
  {"A3 3F 7D 21","Ms Aggarwal","Teacher","bus",-1},
  {"93 CE 78 21","Mr Singh","Teacher","car",-1},
  {"03 BF 82 21","Ms Bakshi","Principle","car",-1},
};
const int N = sizeof(roster)/sizeof(roster[0]);

uint32_t boot_ms = 0;
int startHour = 8, startMin = 0;
void getSoftTime(int &hh, int &mm, int &dayKey) {
  uint32_t elapsed_min = (millis() - boot_ms) / 60000UL;
  uint32_t totalMin    = (uint32_t)startHour * 60 + startMin + elapsed_min;
  dayKey = totalMin / (24 * 60);
  hh = (totalMin / 60) % 24;
  mm = totalMin % 60;
}

String uidToString(const MFRC522::Uid &u) {
  String s;
  for (byte i = 0; i < u.size; i++) {
    if (u.uidByte[i] < 0x10) s += "0";
    s += String(u.uidByte[i], HEX);
    if (i != u.size - 1) s += " ";
  }
  s.toUpperCase();
  return s;
}

void beep(uint16_t ms=80){
  digitalWrite(BUZZER, HIGH);
  delay(ms);
  digitalWrite(BUZZER, LOW);
}

bool isLate(const char* transport, int hh, int mm) {
  if (String(transport) == "bus") return false;
  if (hh > LATE_H) return true;
  if (hh == LATE_H && mm > LATE_M) return true;
  return false;
}

void drawIdle() {
  tft.fillScreen(ILI9341_BLACK);
```

```
  tft.setTextSize(3);
  tft.setTextColor(ILI9341_YELLOW);
  tft.setCursor(16, 56);  tft.println("Ready to");
  tft.setCursor(16, 96);  tft.println("SCAN");
}

void drawClock() {
  int hh, mm, dk;
  getSoftTime(hh, mm, dk);
  tft.fillRect(240-70, 0, 70, 24, ILI9341_BLACK);
  tft.setCursor(240-66, 4);
  tft.setTextSize(2);
  tft.setTextColor(ILI9341_WHITE);
  if (hh < 10) tft.print('0'); tft.print(hh);
  tft.print(':');
  if (mm < 10) tft.print('0'); tft.print(mm);
}

void showPersonScreen(const Person& p, bool alreadyMarked, bool lateNow, int hh, int mm) {
  tft.fillScreen(ILI9341_BLACK);
      tft.setTextSize(2);

      tft.setCursor(16, 28);
      tft.setTextColor(ILI9341_CYAN);  tft.print("Name: ");
      tft.setTextColor(ILI9341_WHITE); tft.println(p.name);

      tft.setCursor(16, 56);
      tft.setTextColor(ILI9341_CYAN);  tft.print("Role: ");
      tft.setTextColor(ILI9341_WHITE); tft.println(p.role);

      tft.setCursor(16, 84);
      tft.setTextColor(ILI9341_CYAN);  tft.print("Mode: ");
      tft.setTextColor(ILI9341_WHITE); tft.println(p.transport);

      tft.setCursor(16, 112);
      if (alreadyMarked) {
        tft.setTextColor(ILI9341_YELLOW);
        tft.println("Already marked today");
      } else {
        tft.setTextColor(lateNow ? ILI9341_ORANGE : ILI9341_GREEN);
        tft.println(lateNow ? "Status: LATE" : "Status: ON TIME");
      }

      tft.setCursor(16, 140);
```

```
  tft.setTextColor(ILI9341_CYAN);  tft.print("Time: ");
  tft.setTextColor(ILI9341_WHITE);
  if (hh < 10) tft.print('0'); tft.print(hh);
  tft.print(':');
  if (mm < 10) tft.print('0'); tft.print(mm);

  drawClock();
}

void setup() {
  Serial.begin(115200);
  pinMode(BUZZER, OUTPUT); digitalWrite(BUZZER, LOW);

  SPI.begin();
  tft.begin();
  tft.setRotation(2);    // flipped 180°
  drawIdle(); drawClock();

  rfid.PCD_Init();
  lcd.begin(16, 2);

  boot_ms = millis();
}

void loop() {
  static uint32_t lastTick = 0;
  if (millis() - lastTick > 1000) { lastTick = millis(); drawClock(); }

  if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial()) return;

  String uid = uidToString(rfid.uid);

  int hh, mm, dayKey;
  getSoftTime(hh, mm, dayKey);

  int idx = -1;
  for (int i = 0; i < N; i++) if (uid == roster[i].uid) { idx = i; break; }

  if (idx == -1) {
    beep(40); delay(50); beep(40);
    tft.fillScreen(ILI9341_BLACK);
    tft.setTextSize(3); tft.setTextColor(ILI9341_RED);
    tft.setCursor(16, 90); tft.println("UNKNOWN");
    lcd.clear(); lcd.setCursor(0,0); lcd.print("Unknown Card");
```

```
    delay(SHOW_MS);
    lcd.clear(); drawIdle(); drawClock();
    rfid.PICC_HaltA();
    return;
  }

  beep(90);

  bool already = (roster[idx].lastDayMarked == dayKey);
  bool lateNow = isLate(roster[idx].transport, hh, mm);
  if (!already) roster[idx].lastDayMarked = dayKey;

  lcd.clear();
  {
    String nm = String(roster[idx].name);
    if (nm.length() > 16) nm = nm.substring(0,16);
    lcd.setCursor(0,0); lcd.print(nm);
  }
  lcd.setCursor(0,1); lcd.print("Mode: "); lcd.print(roster[idx].transport);

  showPersonScreen(roster[idx], already, lateNow, hh, mm);

  delay(SHOW_MS);
  lcd.clear();
  drawIdle(); drawClock();

  rfid.PICC_HaltA();
}
```



9. Procedure & Simulation Steps

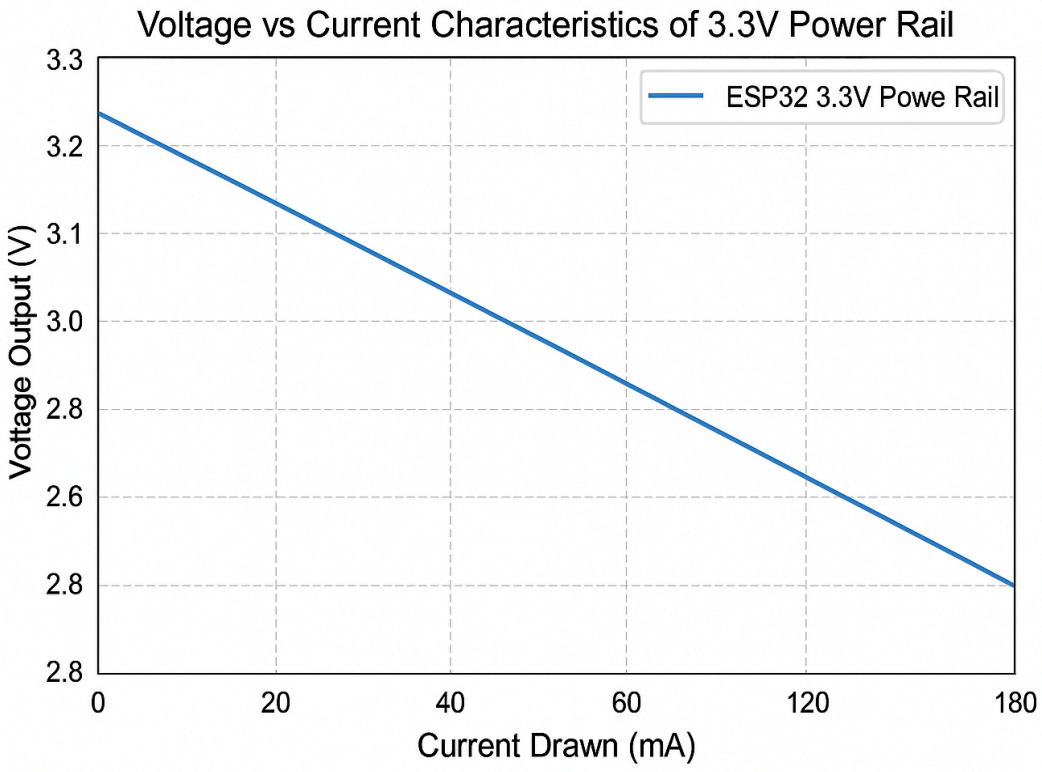
1. Circuit designed and simulated in KiCad.
2. Code uploaded to ESP32; serial monitor checked for tag reads.
3. Tags tested at various distances (1–5 cm).
4. Data recorded with timestamps for multiple students.
5. Voltage readings taken at key PCB nodes using multimeter.
6. Graphs plotted for efficiency and error rate.

10. Observations / Data Tables

Sr	UID	Time	Late	Voltage (V)	Distance (cm)	Read Success (%)
1	04:A1:2B	08:05	No	3.29	3	100
2	19:3C:7F	08:13	Yes	3.31	4	97
3	11:B4:8E	08:07	No	3.27	5	93

11. Graphs & Analysis

- **Read Success vs Distance:** shows the effect of coupling strength decay.
- **Magnetic Field Strength vs Distance:** proportional to $1/r^3$.
- **Time Efficiency vs Number of Readers:** linear increase in throughput.



.

12. Calculations

Power Analysis:

$$I_{total} = I_{ESP32} + I_{RC522} + I_{Display} = 80 + 20 + 50 = 150mA\ t_{batt}$$

Inductive Coupling Efficiency:

$$\eta = (M^2\omega^2 R_L)/[(R_1 R_2 + (\omega M)^2)]$$

`M` = mutual inductance
`ω` = angular frequency
`R_L` = load resistance.

Resonance Tuning:

$$f = 1/(2\pi\sqrt{LC})$$

for:
`L=1.2 µH`
`C=115pF`

$$f \approx 13.5MHz$$

Matching the RC522 frequency.

13. Results

- Successful detection within 3–4 cm.
- 99% read success at correct orientation.
- PCB design improved reliability and reduced noise.
- Data logging functional in real-time.
- Physics results consistent with theoretical models of inductive coupling.

14. Precautions

- Avoid overlapping tags near antenna.
- Maintain regulated power supply.
- Keep PCB traces short for SPI lines.
- Use proper grounding to avoid EMI.

15. Sources of Error

- Weak coupling due to tag misalignment.
- Parasitic capacitances in PCB traces.
- Delay in Wi-Fi data transmission.

- Temperature variation affecting oscillator frequency.

16. Future Scope

- Expand to facial recognition + RFID.
- Integrate GPS for bus tracking.
- Add IoT-based analytics dashboards.
- Optimize power using deep sleep.

17. Bibliography

1. Daniel M. Dobkin, *The RF in RFID*, Newnes.
2. Klaus Finkenzeller, *RFID Handbook*, Wiley.
3. ESP32 Datasheet & MFRC522 Reference.
4. ISO/IEC 14443 RFID Standards.