RFID BASED ATTENDANCE SYSTEM

Project Final Report

Group - 14

Aksh Garg 20D070008

Tejas Amriktar 20D070081

Pavan Kale 180110050

Guided By:

Prof. Laxmeesha Somappa

Prof. Dinesh Sharma

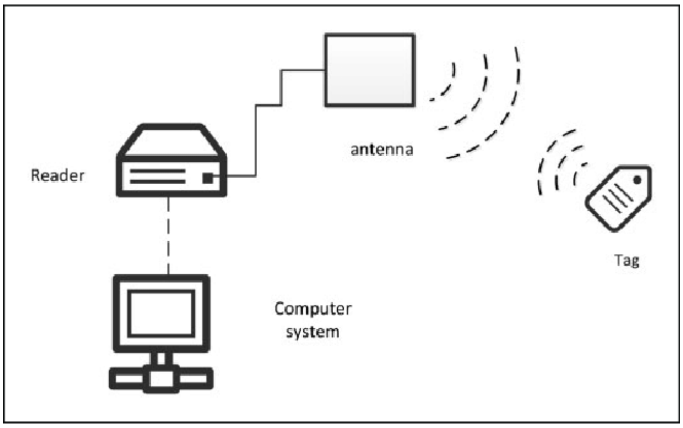
Department Of Electrical Engineering

IIT Bombay

INTRODUCTION

RFID (Radio Frequency Identification) is a form of communication that incorporates the use of electromagnetic or electrostatic coupling in the radio frequency portion of the electromagnetic spectrum to uniquely identify an object.

The aim of this project is to build a complete user-friendly RFID reader which gets the unique ID of the RFID tag when brought close to reader. This reader should be designed to update the information of the corresponding tag to the central database, as required, which would act as an attendance and registration system. At the same time, an LED would glow, and corresponding message would be displayed on the TFT LCD screen, indicating that the scan is successful. We successfully designed a breadboard-level prototype of the project, that would update the required information online on the Google Excel sheet.

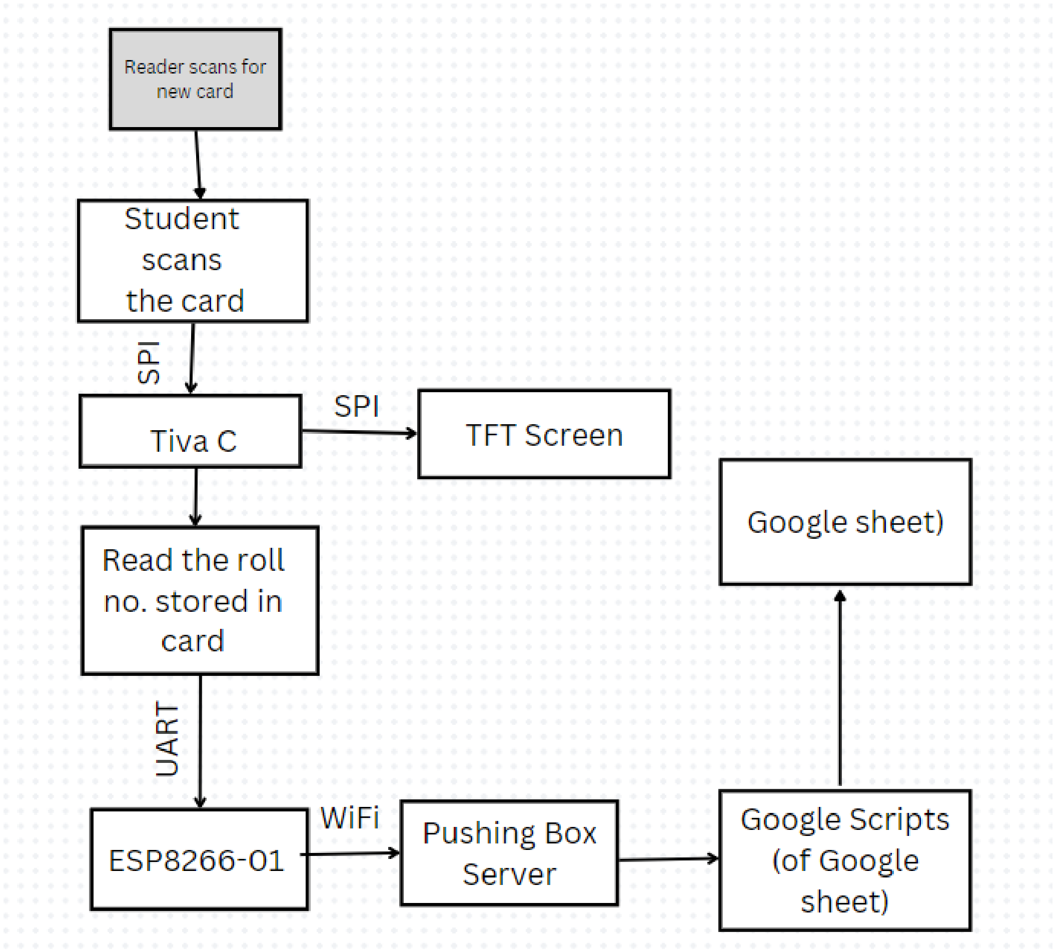


For this project we are using UF, and passive RFID tags operating at 13.56MHz frequency. The major components of the reader are antennas, the RFID reader module, a microcontroller, a TFT screen, and ESP01 to connect our project to online database. The basic working of RFID is as follows:

- The RFID tag used is a passive tag, which is not powered by itself. The antenna of the reader emits radio waves as signals which activates the tag. Once activated, the tag sends back a wave (containing the info about the unique ID of the tag) to antenna.
- The antenna sends the signal to the RFID Module chip, which processes the information sent by the tag. This information is sent to the microcontroller using the SPI protocol, and the microcontroller is programmed to handle the information, abstract the relevant data, modify, and update the attendance system.

We would require a database to manage the information about all the tags which would be scanned to the reader. For this purpose, we would be using Google Sheets as an online central server for the attendance system. Whenever a tag is scanned, the date and time of that instance would be stored in the database too.

Flow Diagram for the Attendance system

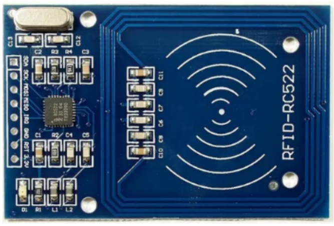


COMPONENT SELECTION

RC522 Reader Module:

This component is used to read and detect the RF tags in its vicinity, and works on the principle of Electromagnetic induction. The major parts of this component are:

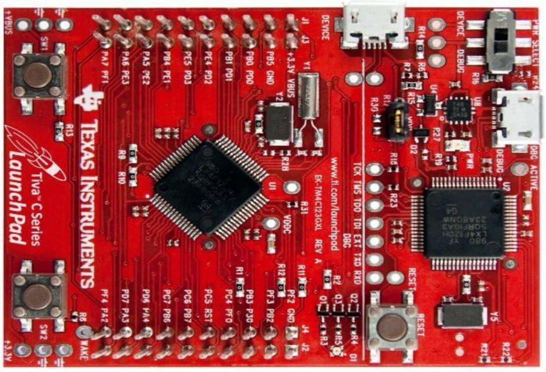
- **Antenna:** Radiates HF radio frequency signals in form of sinusoidal waves, resulting in an electromagnetic field around it. Whenever a tag comes near this field, it gets activated and resends the data on it in the form of EM waves. This interception of the field is detected by the antenna and sends the information to MFRC522 chip.
- **MFRC522:** This decodes the data sent by the antenna, and communicates with the microcontroller with the help of SPI protocol.



TIVA C Microcontroller:

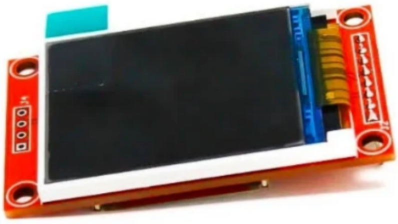
The **TM4C123GH6PM** is a member of the class of high-performance 32-bit ARM cortex M4 microcontroller with a broad set of peripherals developed by Texas Instrumentals. This board satisfies all the requirements for the project of multiple SPI ports, UART, and internal non-volatile memory storage.

The TIVA C microcontroller's primary function will be to interface with our RFID reader module using the SPI protocol, do the necessary backend calculations and data authentications, and then transmit the data to our central online database through ESP01 with the help of UART protocol. The required message of the card would be displayed on the TFT LCD screen, communicating with the microcontroller through SPI protocol.



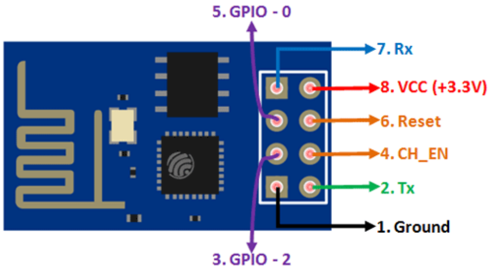
TFT LCD Screen:

We are using an LCD screen to display the corresponding messages on it, and coded the UI in such a way, that makes it more attractive and user friendly. It uses the SPI protocol to communicate with Tiva C, which sends the required parameters and messages to be printed in the screen,



ESP01:

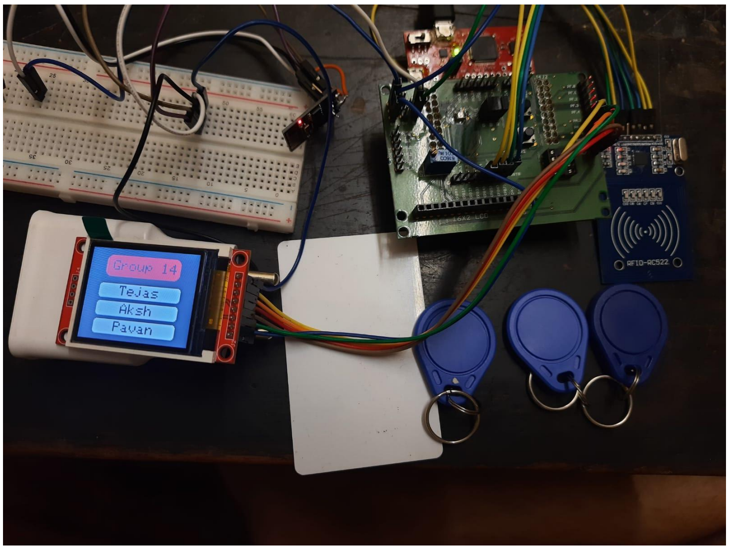
We are using ESP01 to connect our project to the internet and send the required information to be stored in the database. We used this component as our only requirement was to access internet, and this is the most cheapest and smallest module for this purpose.



IMPLEMENTS and RESULTS

Setup:

- RC522 was connected with the TIVA C microcontroller by SPI communications, as available on the RC522 module and connected to the respective pins of TIVA C, and the code built was transferred to it, so that it can handle RC522 as required.
- The power to microcontroller, and hence to RC522, is given with the help of micro usb cable of the controller, connected to the laptop.
- The TFT screen is connected to the microcontroller through another SPI port, and ESP01 is connected through UART port. The code is done in such a way that it handles TFT screen, RC522, and ESP01 synchronously and works as required.



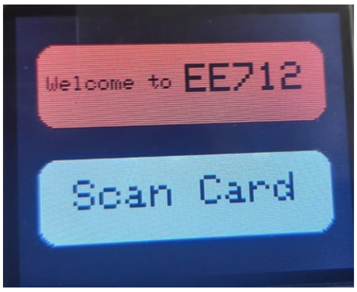
Method Followed:

- TIVA C has ports which can be used to interface with the RC522 through SPI. So, with help of jumping wires, corresponding connection was done between them, and the SPI port was activated of the microcontroller. Similarly, the TFT screen and ESP01 is connected.
- TIVA C works on embedded C, and does not support Arduino codes. For this reason, the library for MFRC522 was found for C code from the internet, and made changes in them to work efficiently with the CCS IDE (original IDE for TIVA C).
- The library had some functions which need to be changed on the register level of the MFRC522 chip, for which we read the datasheet and learnt about its commands.
- After completing the work on the coding level for attendance, and register systems, we activated the MFRC522, and activated the antenna on RC522. So, whenever an RFID card comes in range of the reader, blue LED glows of the microcontroller, and corresponding message is displayed on the screen, implying that the detection was successful.
- There are two types of cards: **Student** card and **Professor** card
  - Professor card contains the course code and the first name of the professor. When a professor scans the card, the course code is read from the card, and stored in the internal memory as well as in a global variable.
  - Student card contains his/her roll number and the first name. When a student scans the card, this roll number along with the course code stored in the global variable is sent to ESP01 through UART, and this data is stored in the online database with the timestamp. When a new professor scans the card, that means a new course attendance started.
- When the system starts, the previous accessed course is extracted from the EEPROM memory of the microcontroller and stored in the global variable, which gets stored when a professor scans the card.
- Initially, the controller checks the card, whether it is a student or a professor card. During registration, if it was a student card, we put the value ‘128’ in the storage of the card, and similarly ‘131’ for professor card.
- For reading/writing data blocks of the card, we read the datasheet of MIFARE card, and followed the steps to do so. The steps required to select the card by sending the UID of the card to MFRC, followed by an authentication with the data sector (with the help of data sector password) to read/write the data successfully from the card.
- Similarly, the registration system takes the data of the card, and work as required.
- In each of the event that occurs, required message is printed on the TFT screen through SPI communication protocol.

Updating events on the TFT screen:

Attendance system:

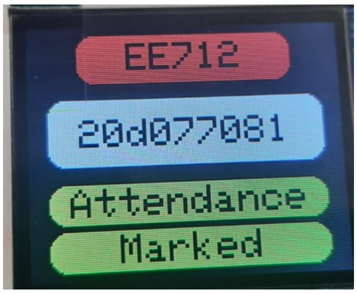
1. Ask student to mark attendance by placing their card on RFID reader.



Welcome to EE712

Scan Card

2. After placing the student ID card, screen shows the attendance has been marked for the course that is being diplaued on the screen along with the roll number of studnet.



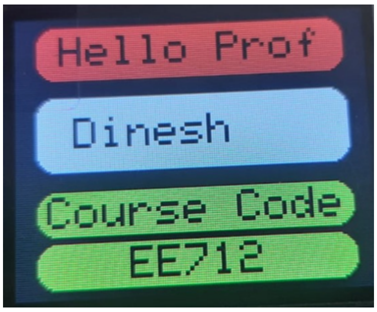
EE712

20d077081

Attendance

Marked

3. If professor places his card on the RFID reader after step 1, then screen shows the name of professor along with the course he/she is teaching. Student who places his card after step 3, their attendance will be marked under the course displayed in this step.



Hello Prof

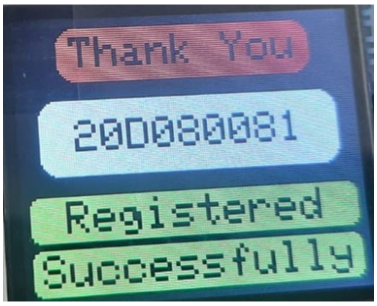
Dinesh

Course Code

EE712

Registration system:

1. On the admin side, when student places his/her card on the RFID reader, student will update his/her details through Computer with software such as Teraterm or Realterm. After updating details, student's details will be updated on the screen as shown, saying that his/her card is registered.



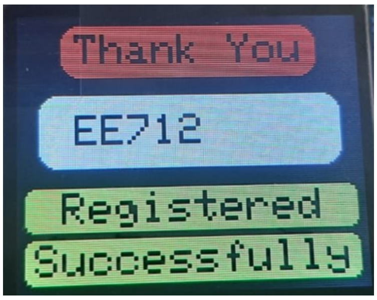
Thank You

20D080081

Registered

Successfully

2. When professor places his/her card on the RFID reader, after updaing details through the computer, details are updated on the screen as shown.



Uploading attendance data on the cloud

As the Tiva C communicates with ESP01 wifi module using UART protocol, it sends a String formatted data. The format looks as follows: *<String: Course name><Space><String: Roll no>*. For example: “EE712 180110050”.

This string is then parsed to separate the components into “course” and “rollno” variables. <Space> is used as a delimiter for parsing. The advantage of this method is that, There is no restriction on format and size of course and roll number part. This is especially necessary for courses like GNRXXX.

Upon parsing the string and storing into variables, the data is sent to the google sheets in the following three steps of the flowchart:

1. **ESP01 to Pushing box server**

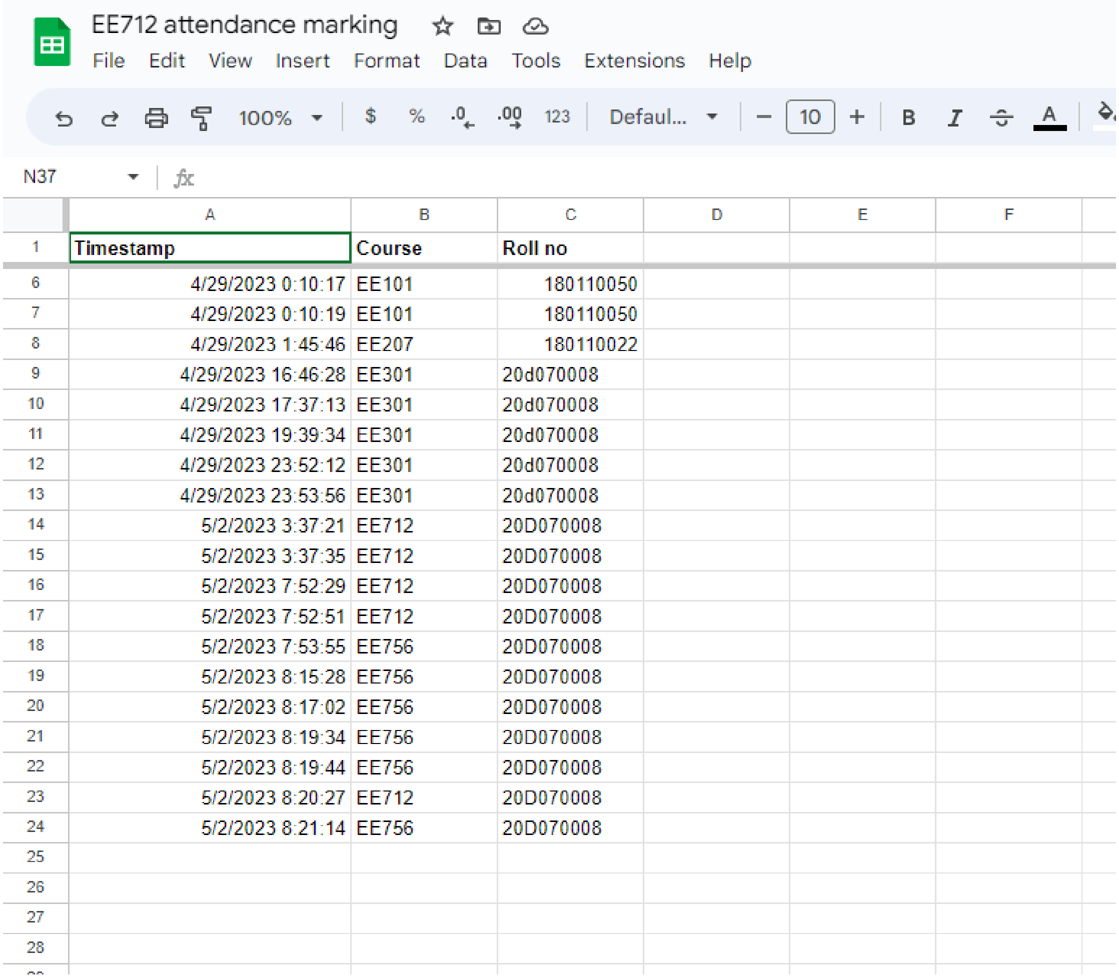
   A service is created using a pushingbox server. This provides an API, where a custom url is generated. ESP01 contains credentials to connect to a local wifi and access the internet. A HTTP GET request is made to the pushing box server. The ESP01 code contains the url of the pushingbox server. It makes the GET request using the earlier parsed roll no and course name values.

2. **Pushing box server to google script**

   The GET request is processed by the pushingbox server and sent to a deployed google script instance. This intermediate step is necessary because ESP01 cannot directly ping the google script deployment instance.

3. **Google script to Google sheet**

   Google script is a powerful tool that can be used to automate dataflows in google sheet. A google script is written to receive the data from the pushingbox server. This data along with current time and date data acquired from a built-in function of the google script is appended to the google sheet as a new row as shown in the following figure.



	A	B	C	D	E	F
1	Timestamp	Course	Roll no			
6	4/29/2023 0:10:17	EE101	180110050			
7	4/29/2023 0:10:19	EE101	180110050			
8	4/29/2023 1:45:46	EE207	180110022			
9	4/29/2023 16:46:28	EE301	20d070008			
10	4/29/2023 17:37:13	EE301	20d070008			
11	4/29/2023 19:39:34	EE301	20d070008			
12	4/29/2023 23:52:12	EE301	20d070008			
13	4/29/2023 23:53:56	EE301	20d070008			
14	5/2/2023 3:37:21	EE712	20D070008			
15	5/2/2023 3:37:35	EE712	20D070008			
16	5/2/2023 7:52:29	EE712	20D070008			
17	5/2/2023 7:52:51	EE712	20D070008			
18	5/2/2023 7:53:55	EE756	20D070008			
19	5/2/2023 8:15:28	EE756	20D070008			
20	5/2/2023 8:17:02	EE756	20D070008			
21	5/2/2023 8:19:34	EE756	20D070008			
22	5/2/2023 8:19:44	EE756	20D070008			
23	5/2/2023 8:20:27	EE712	20D070008			
24	5/2/2023 8:21:14	EE756	20D070008			
25						
26						
27						
28						

Novelties in the chosen protocol:

1. The attendance update is **situation based**. Just like an interrupt service routine. The ESP01 uploads the data to the pushingbox server only when something is received through UART. This prevents a malfunction like repeatedly updating the last marked attendance indefinitely.
2. Google script along with pushingbox provide a **zero cost deployment** alternative to a physical or cloud based server. Google script also has almost zero downtime. Thus there is no technical complexity in connecting, reconnecting the system.
3. The system is **scalable**. We can use as many attendance marking devices and the data of all of them can be seen on the sheet.
4. **Customization** using google script: The script can be programmed to separate attendance day wise, course wise or anything else as desired.
5. Google sheet comes with its powerful **data visualization and formatting** tools. These can directly be applied to the attendance data as desired.

List of components to be used (BOM):

Components taken from lab:
TM4C123GH6PM TIVA C board

Componets procured:

Sr. No.	Component	Quantity	Cost
1.	RFID Reader Module (RC522)	1	Rs. 116
2.	ESP01 Module	1	Rs. 107
3.	1.8’ ST7735 TFT Screen:	1	Rs. 399

CONCLUSIONS

- Everything in today’s world is becoming automatic. A new system, which **reduces the human efforts**, for management is needed.
- **RFID technology** provides this solution with easy update of data, by just scanning your tags, eliminates the need for manual recording, and **reduces errors** and fraudulent practices.
- Moving forward, the system has the potential to be further improved and expanded to include additional features such as **automated report generation and integration with management systems.**
- Our project is completely ready to implement to a **large class** with **100%** of **accuracy** of working.