

CS637 Course project

RFID Project

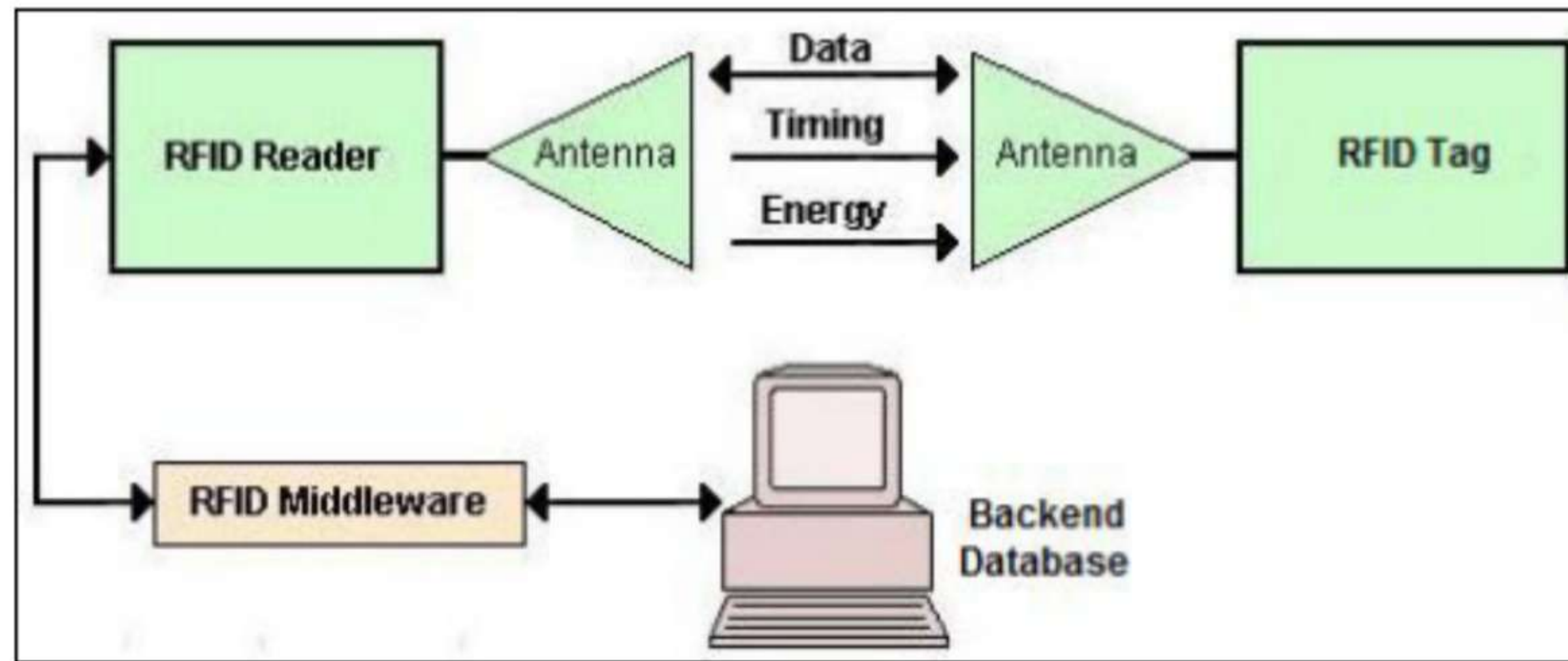
RADIO FREQUENCY IDENTIFICATION

Harsh Agrawal
200407

harsha20@iitk.ac.in

Components of an RFID system

An RFID system consists of various components that are connected to one another by a dedicated communication path. The individual components are integrated into the system to implement the benefits of RFID solution.



Tags

An object that is attached to any product and uses a unique sequence of characters to define it. It comprises of a chip and the antenna.

Antenna

It is responsible for the transmission of information between the reader and tag using radio waves..

Reader

A scanning device that uses the antenna to realise the tags that are in its vicinity. It transmits signals at a certain frequencies.

Middleware

A communication interface to interpret and process data being fed by the readers into information. It takes into account all relevant ports of communication and a software application to represent this information..

Backend database

A repository of information, which is designed specific to the application. The database stores records of data specific to individual tags

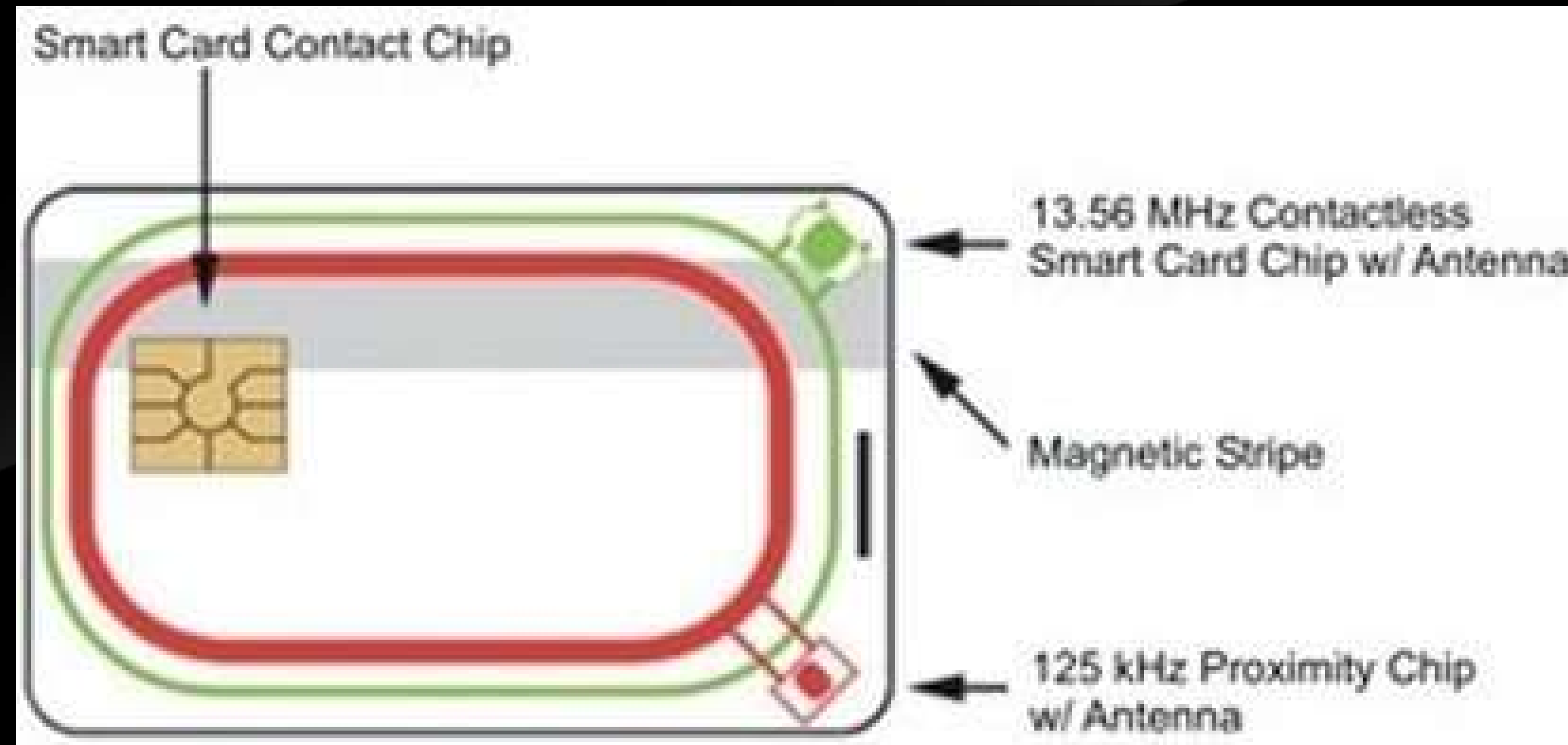
How does RFID works?



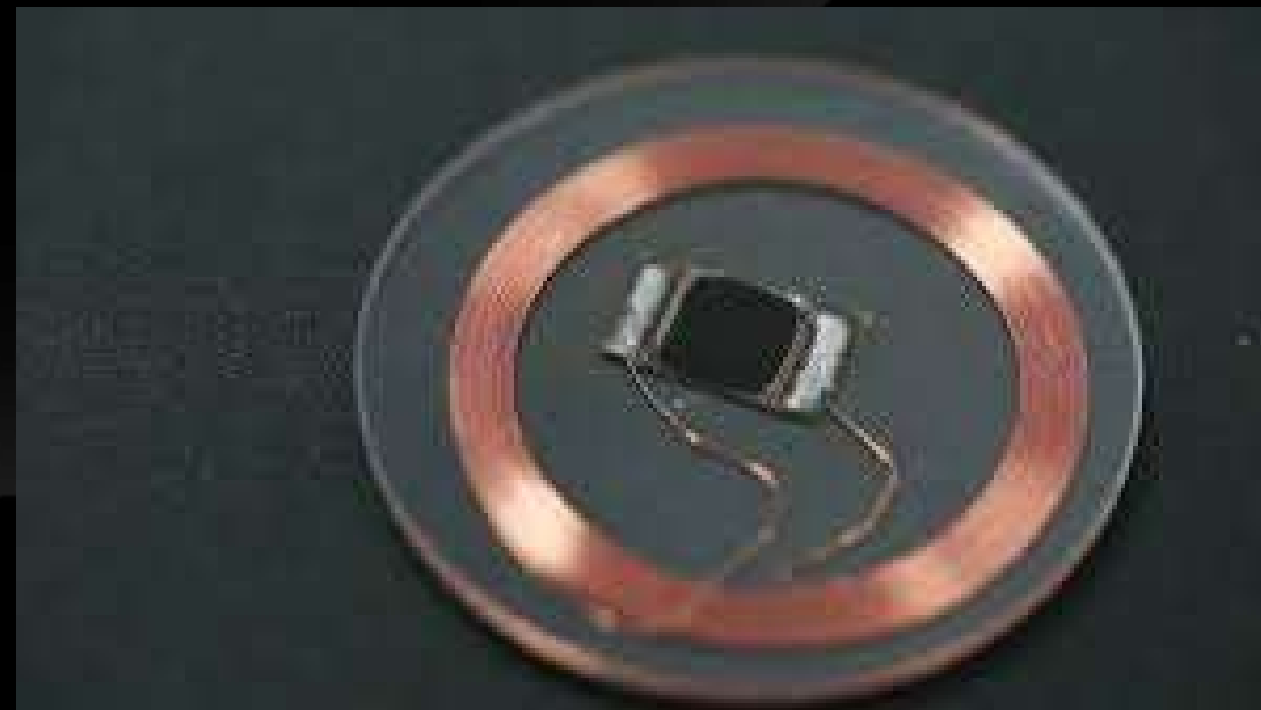
When the tag is brought close to the reader, the reader generates an electromagnetic field. This causes electrons to move through the tag's antenna and subsequently powers the chip.



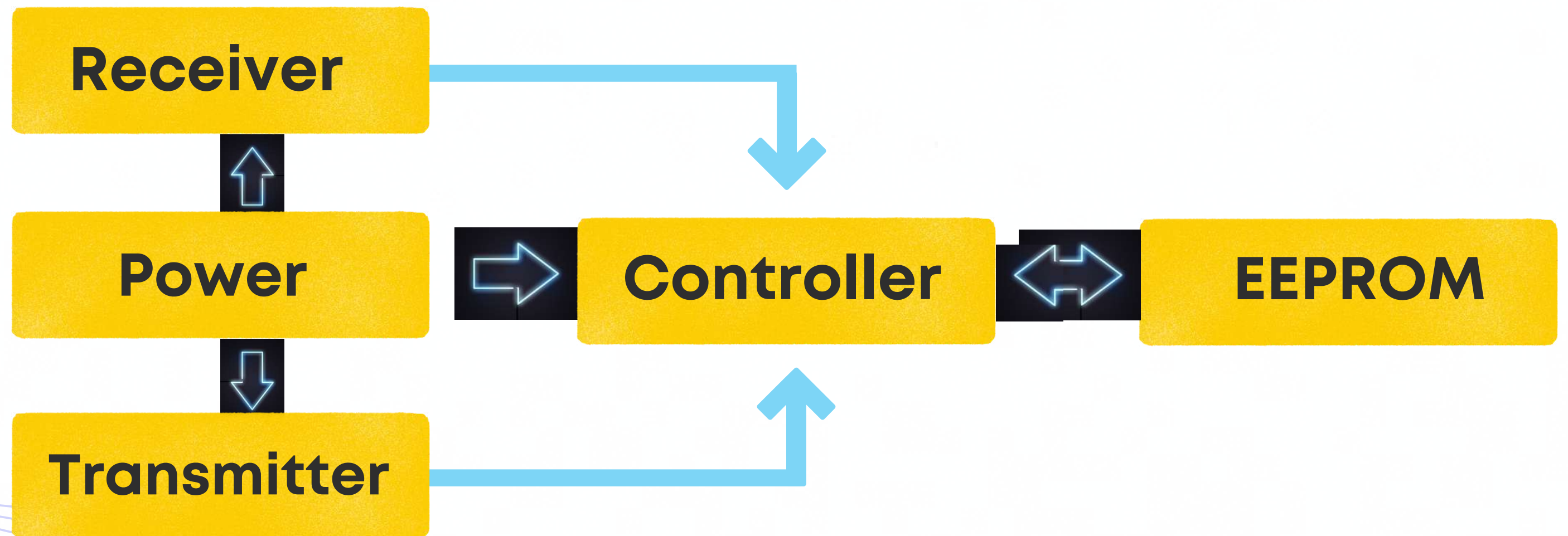
The chip then responds by sending its stored information back to the reader in the form of another radio signal. This is called a backscatter. The reader detects and interprets this backscatter and sends the data to a computer or microcontroller.



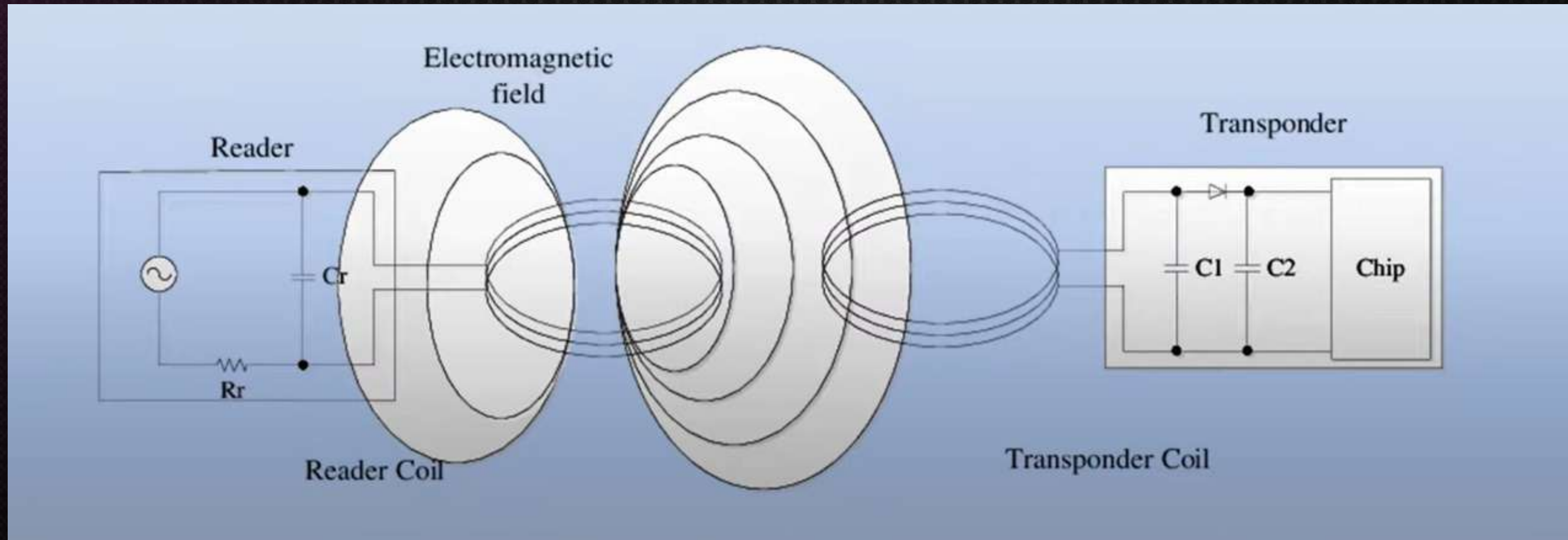
Inside RFID tag and Card



RFID CHIP



Basic Tag Operation Principle



Why only RFID?



Contactless

Quick

Simultaneous read of multiple items

Individual item data and tracking

No requirement of power source on tag so easy and long term usage



RFID Applications



Retail and Distribution



Contactless Payment



Keyless Entry



Livestock Tagging



Pharmaceuticals



Pet Identification

RFID Frequencies

RFID frequency bands^{[23][24]}

Band	Regulations	Range	Data speed	ISO/IEC 18000 section	Remarks	Approximate tag cost in volume (2006)
LF: 120–150 kHz	Unregulated	10 cm (4 in)	Low	Part 2 ↗	Animal identification, factory data collection	US\$1
HF: 13.56 MHz	ISM band worldwide	0.1–1 m (4 in – 3 ft 3 in)	Low to moderate	Part 3	Smart cards (ISO/IEC 15693, ISO/IEC 14443 A, B), ISO-non-compliant memory cards (Mifare Classic, iCLASS, Legic, FeliCa ...), ISO-compatible microprocessor cards (Desfire EV1, Seos)	US\$0.05 to US\$5
UHF: 433 MHz	Short range devices	1–100 m (3–300 ft)	Moderate	Part 7 ↗	Defense applications, Underground Miner Tracking with active tags	US\$5
UHF: 865–868 MHz (Europe) 902–928 MHz (North America)	ISM band	1–12 m (3–40 ft)	Moderate to high	Part 6 ↗	EAN, various standards; used by railroads ^[25]	US\$0.04 to US\$1.00 (passive tags)
microwave : 2450–5800 MHz	ISM band	1–2 m (3–7 ft)	High	Part 4 ↗	802.11 WLAN, Bluetooth standards	US\$25 (active tags)
microwave: 3.1–10 GHz	Ultra wide band	up to 200 m (700 ft)	High	Not defined	Requires semi-active or active tags	US\$5 projected
mm-wave: 24.125 GHz ^{[26][27][28]}	ISM band worldwide	10–200 m (30–700 ft)	High	Not defined	Requires semi-passive tags. Uses retrodirective backscatter approaches to achieve extended ranges	US\$10 projected

Aim

Using an RFID tag system to make a entry exit management system, to make this smooth and efficient a tag with low maintainence and easy to handle.

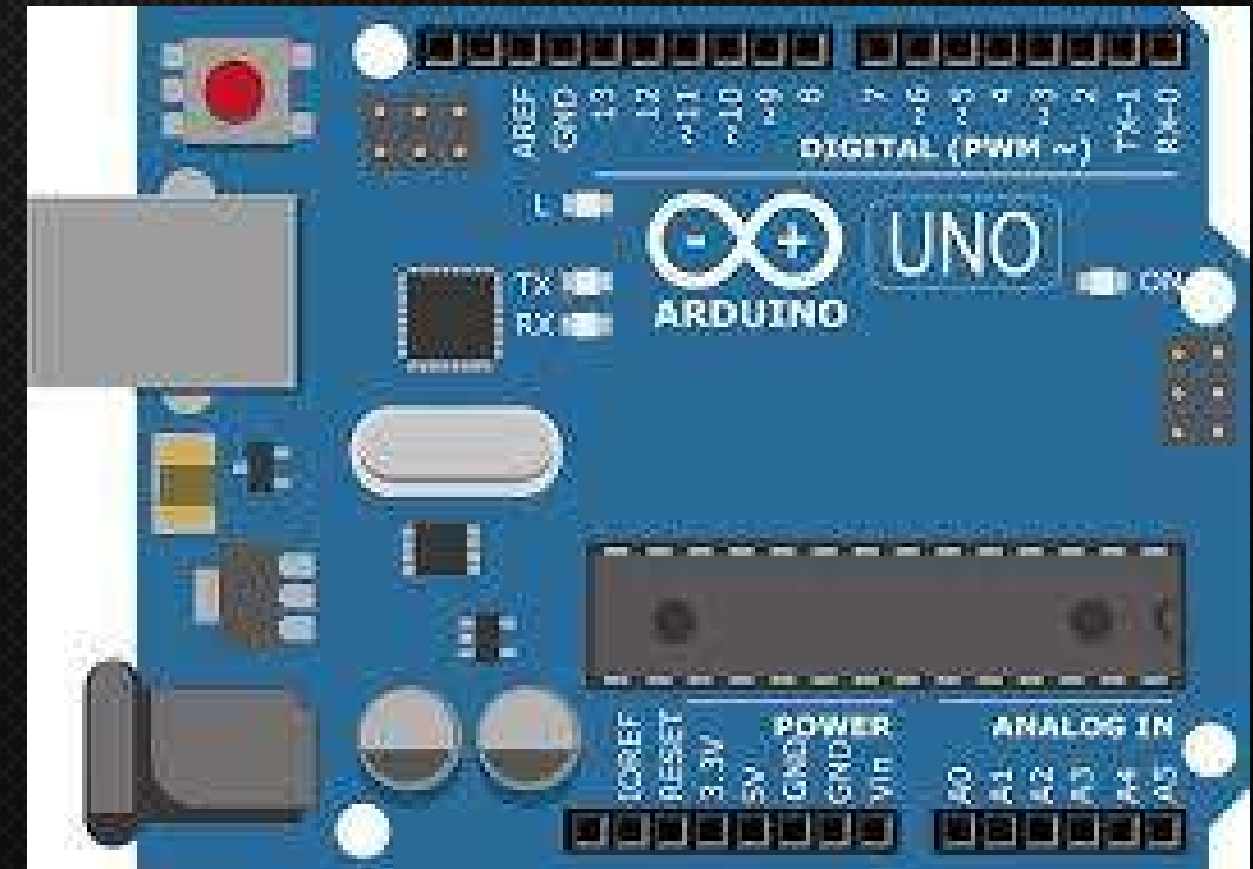
Components used

RFID Reader/Writer



The RC522 RFID reader module which is used to read and write RFID tags.

Arduino



Arduino here is used to do the processing work here and communicate data received from RFID reader to a local host.

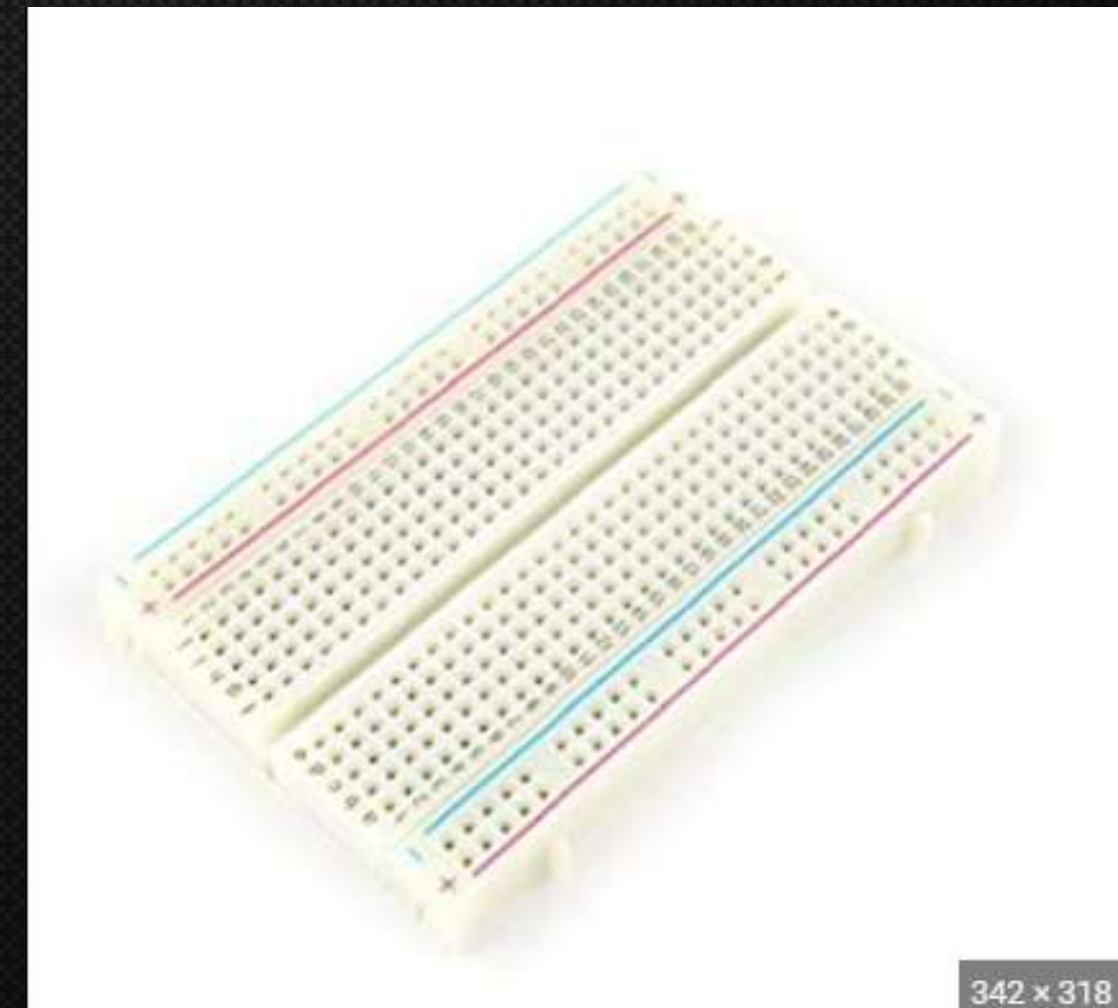
Components used

RFID tag



RFID stores data of a student and can be edited .

RFID tag

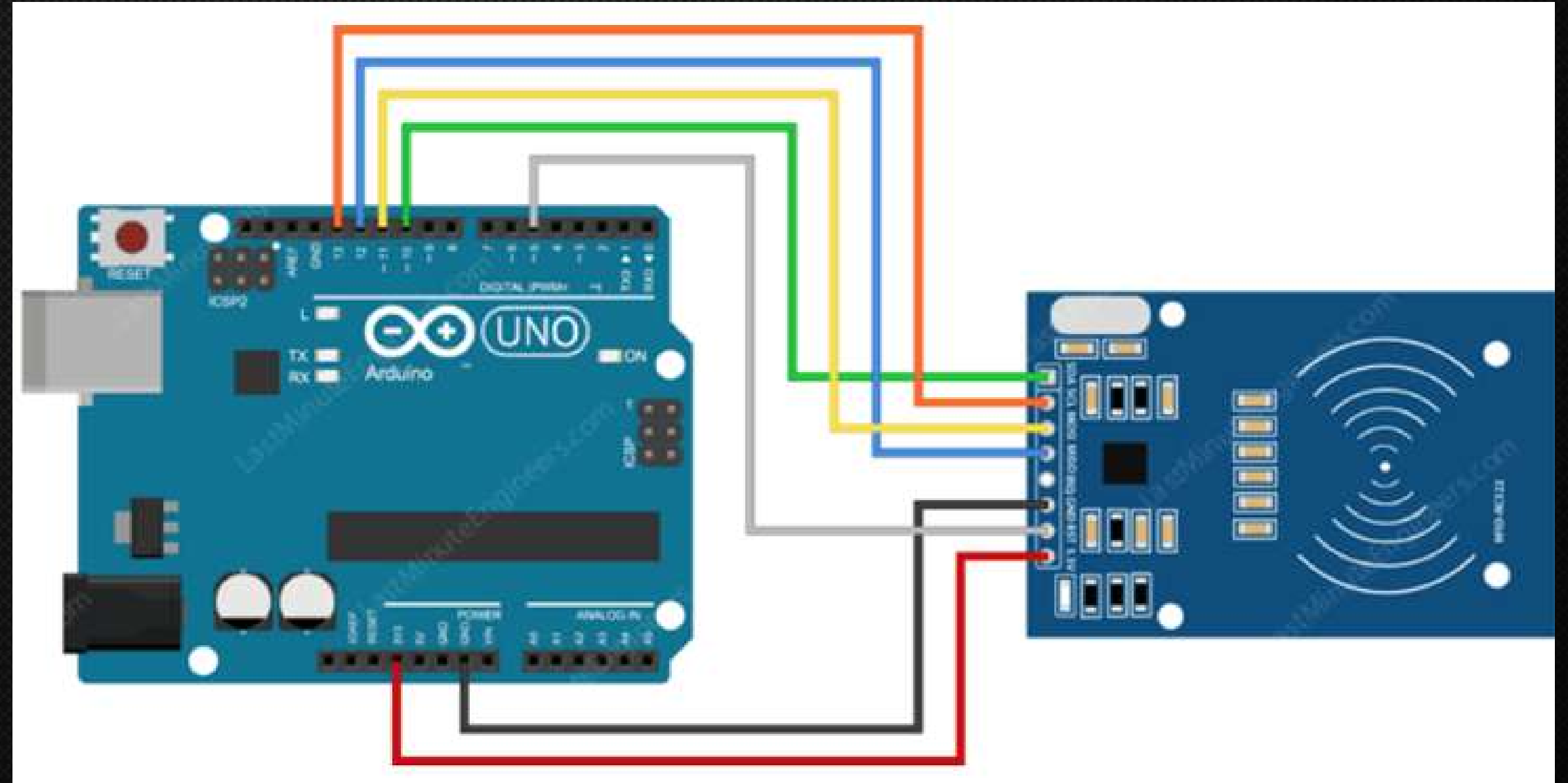
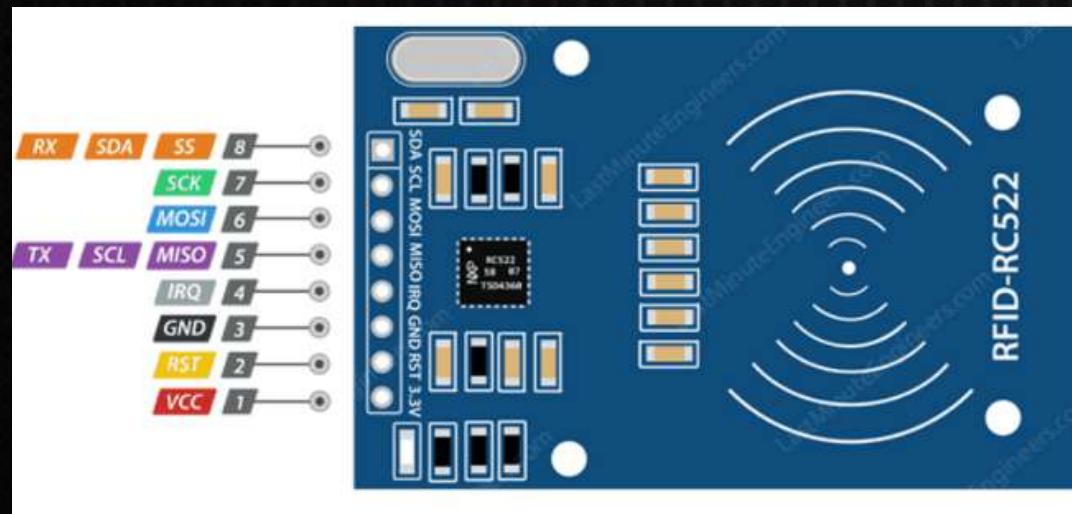


A bridgeboard is used to make connections to our model.

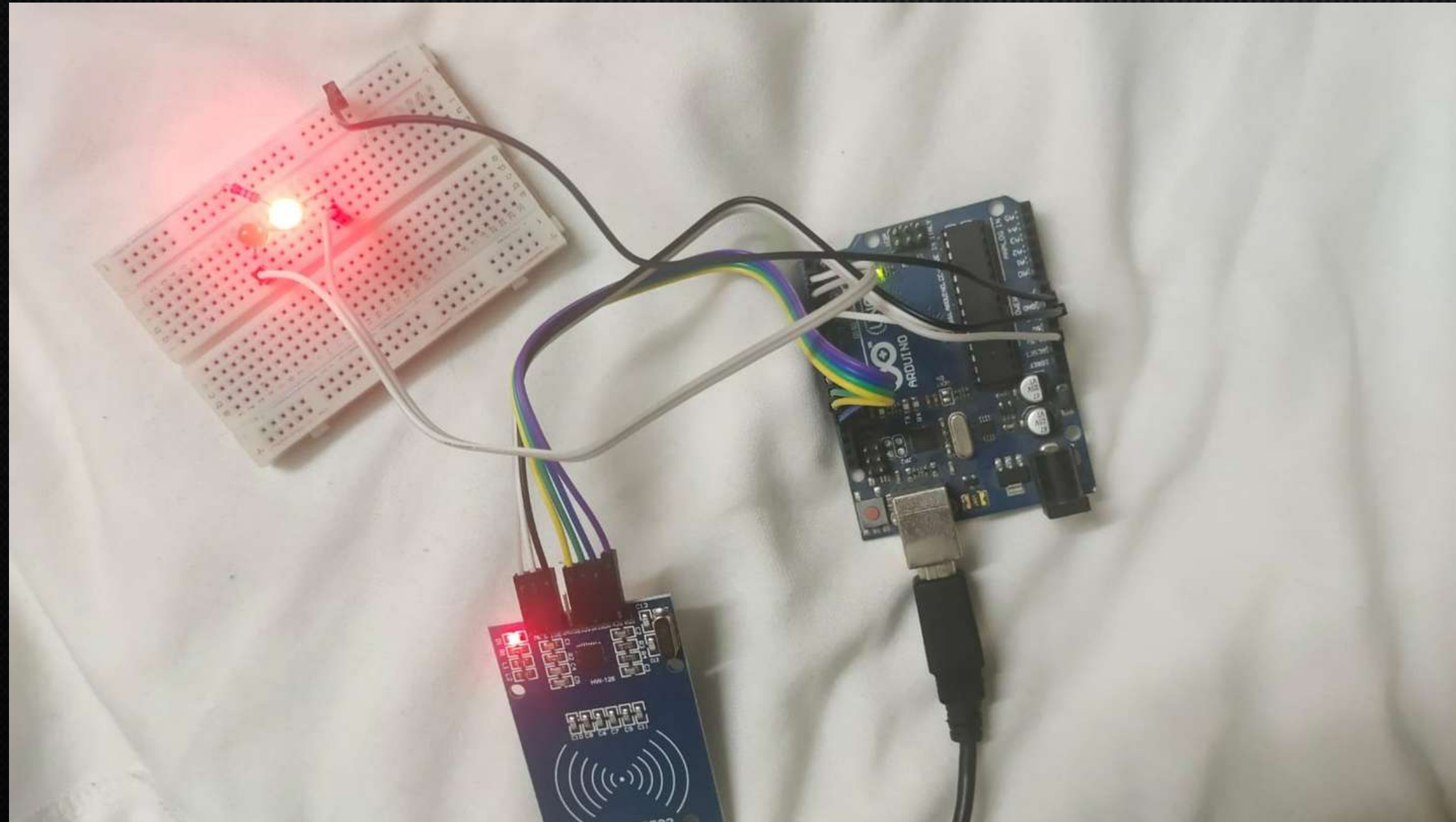
RFID Reader/Writer

The RC522 RFID reader module is designed to create a 13.56MHz electromagnetic field and communicate with RFID tags. The reader can communicate with a microcontroller over a 4-pin SPI with a maximum data rate of 10 Mbps. It also supports communication over I2C and UART protocols. The RC522 RFID module can be programmed to generate an interrupt.

Connections and Pinouts

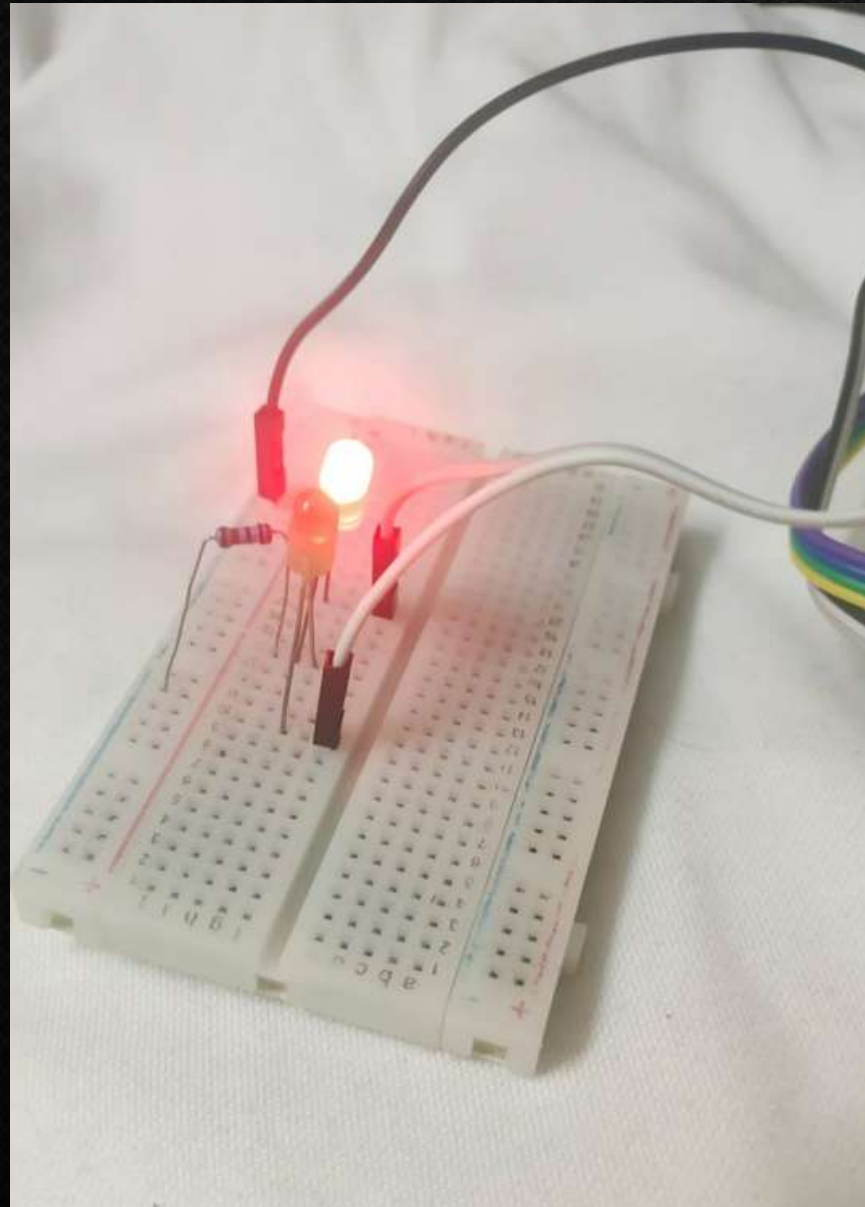


Working Model

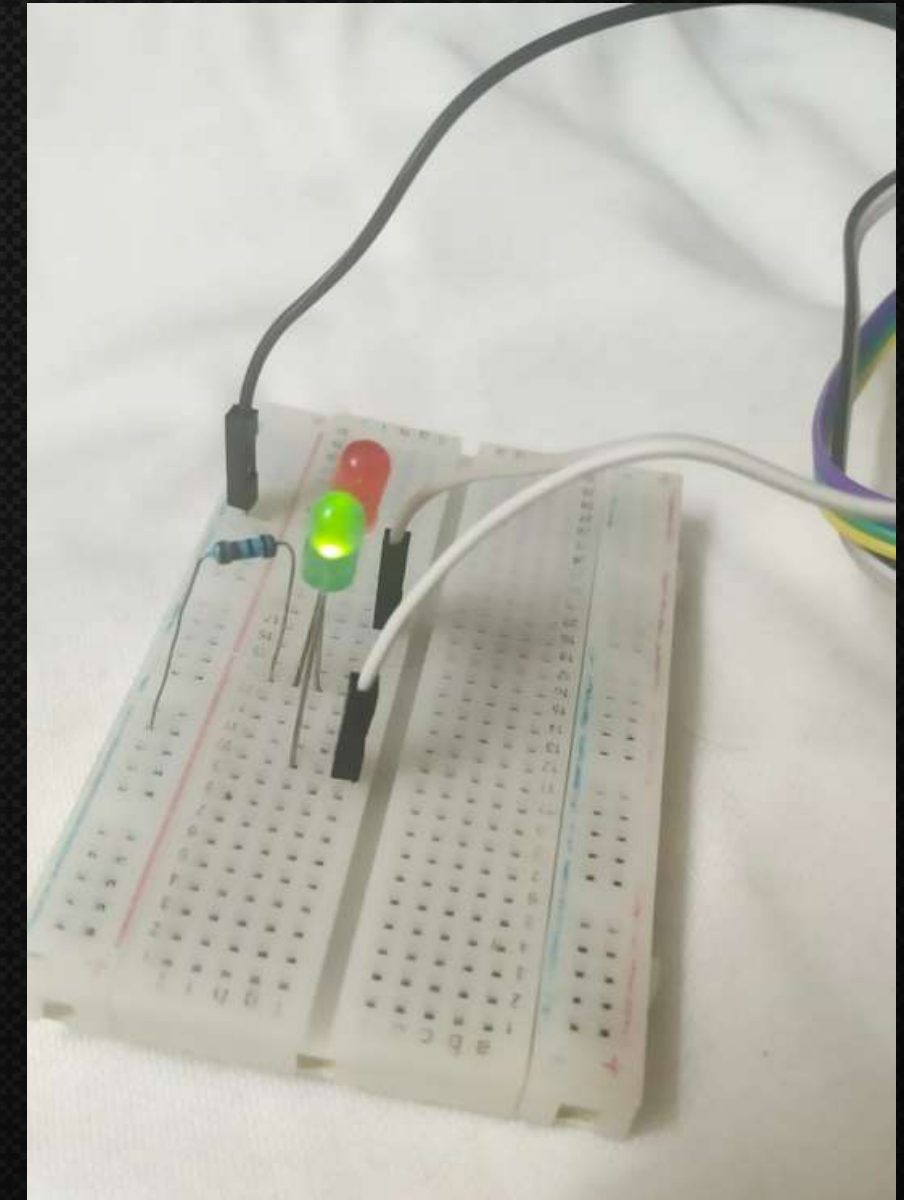


An Arduino connected to an RFID reader, and a bridgeboard, which whole is connected to a Laptop.

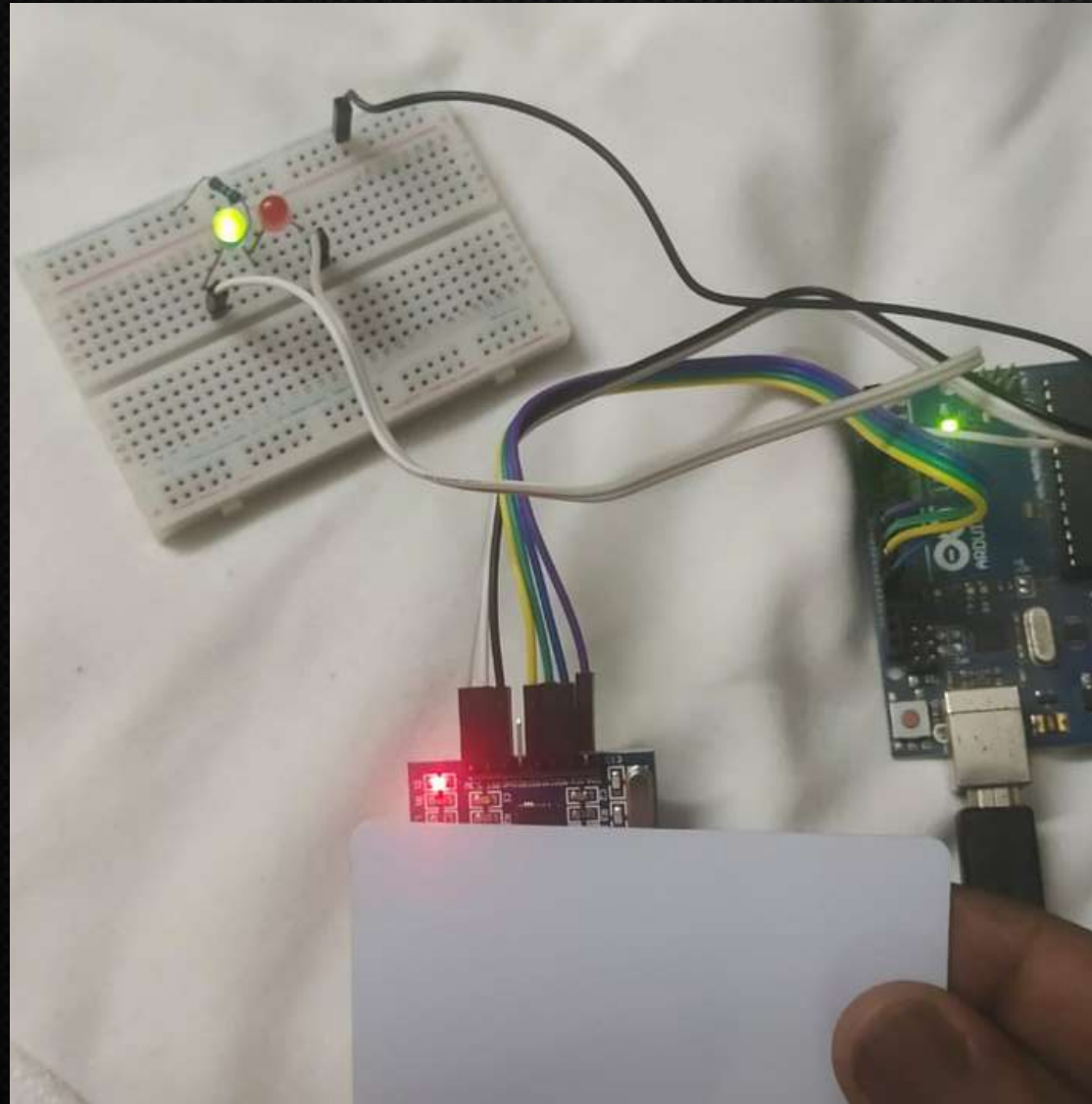
Bridgeboard



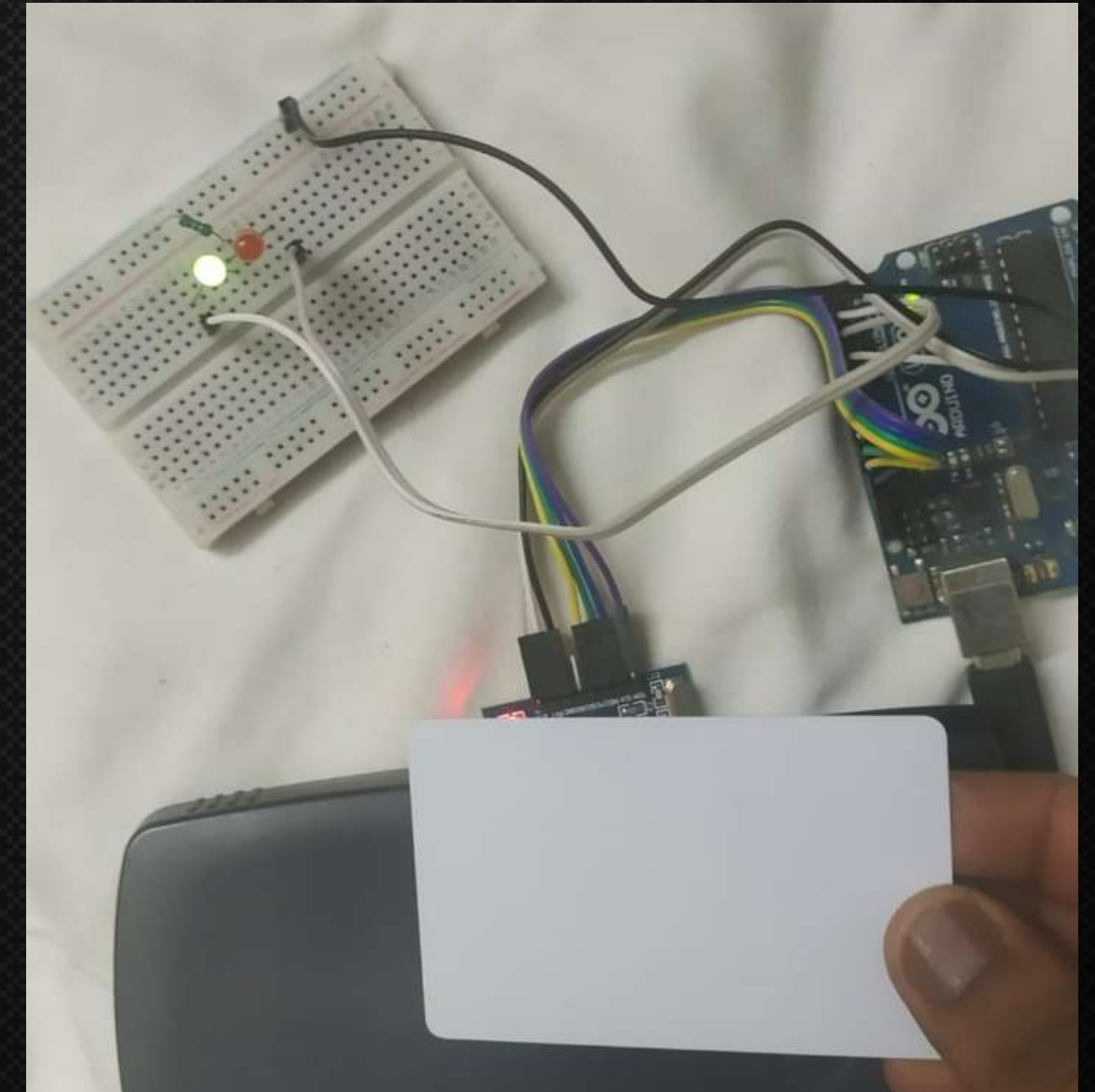
As a RFID tag/card is read by RFID reader successfully the Bridgeboard turns to green light.



Reading of a RFID Tag/card



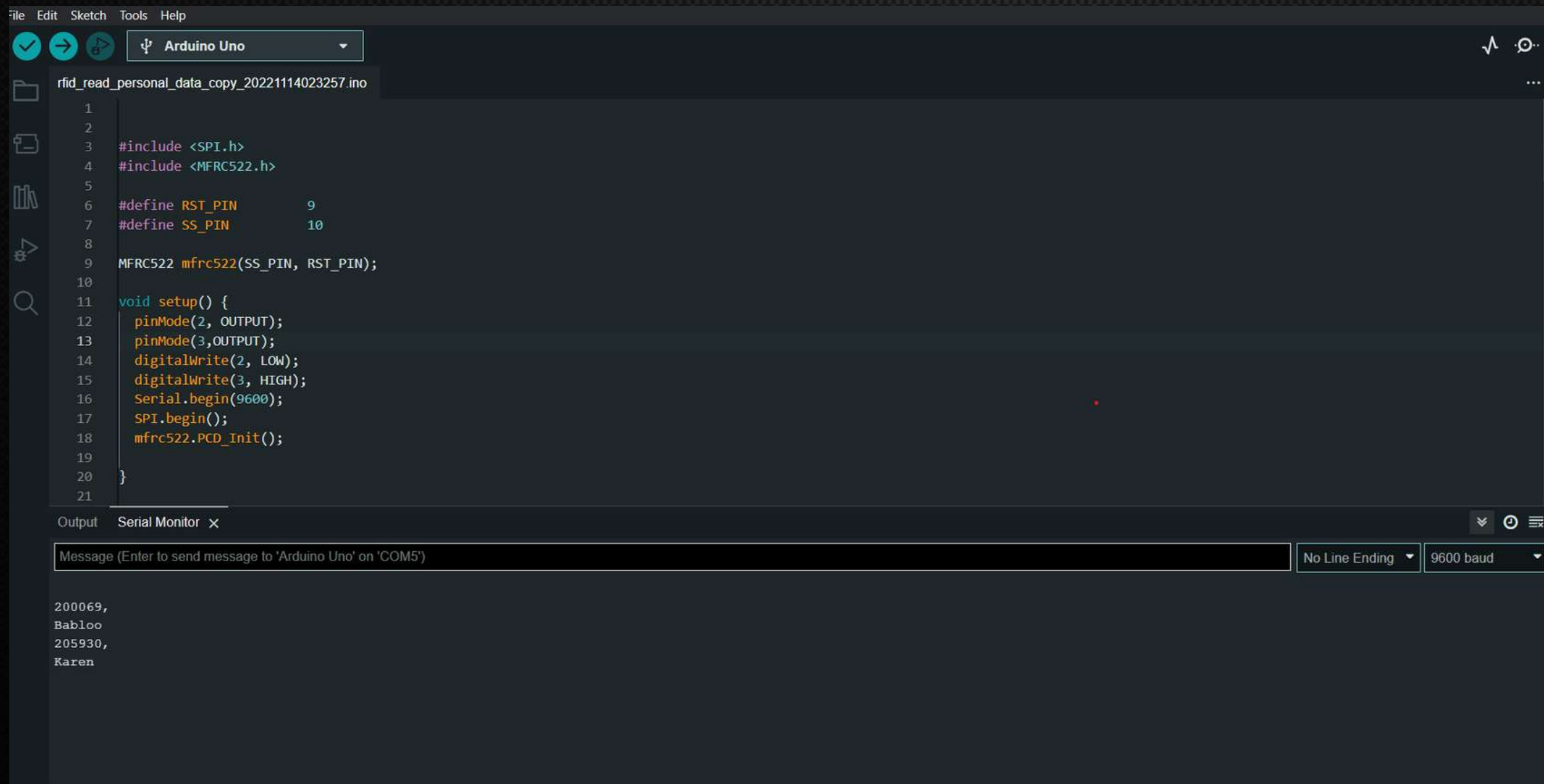
An RFID card is read by the reader at a distance of few centimeters.



An RFID card can also be read by the reader in presence of obstruction between them in this case a calculator.

Terminal Window

Reading an RFID card

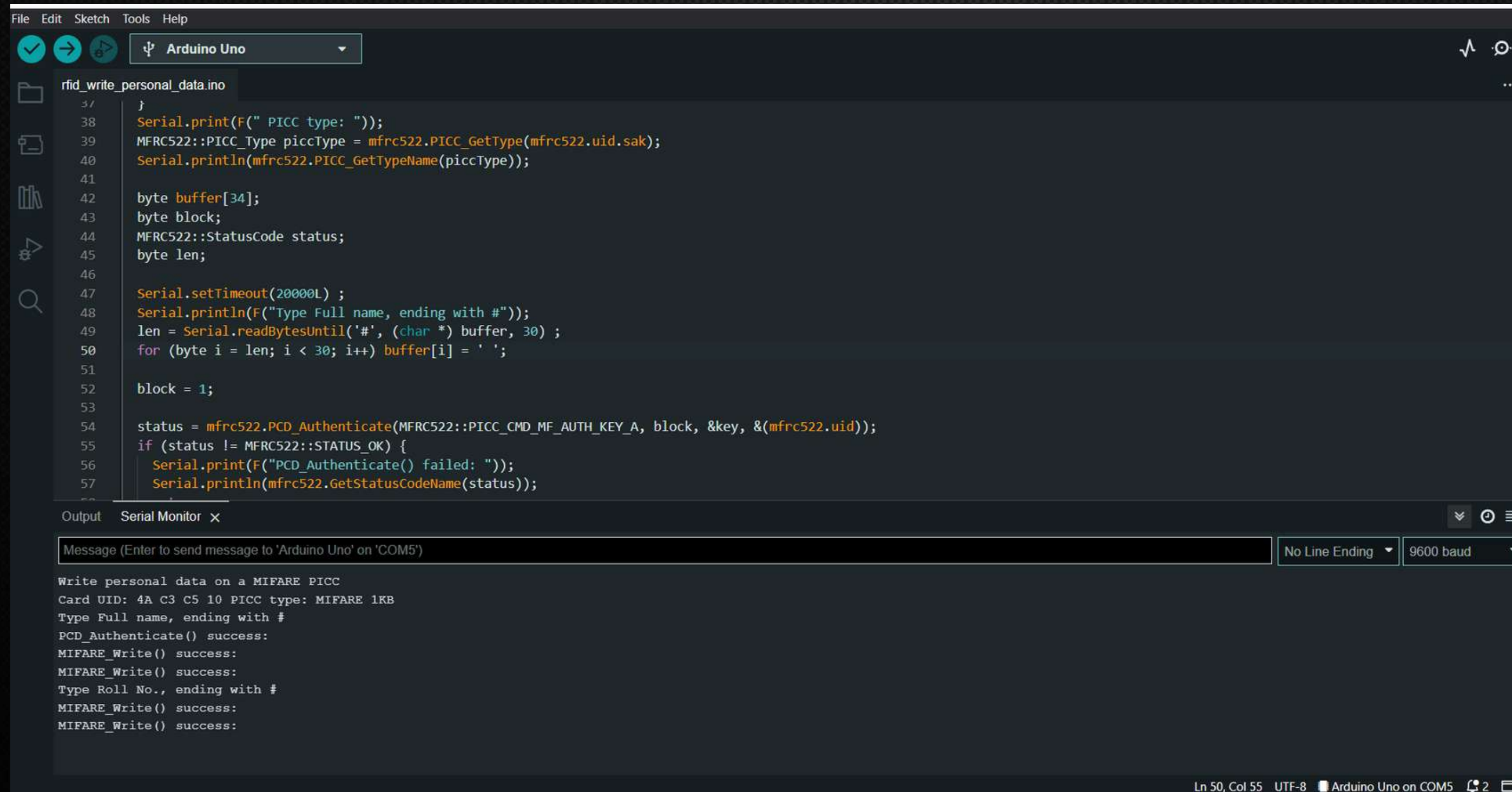


The screenshot shows the Arduino IDE interface. The top menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for checking, uploading, and debugging, along with a dropdown menu set to 'Arduino Uno'. The main editor area displays a sketch named 'rfid_read_personal_data_copy_20221114023257.ino'. The code is as follows:

```
1
2
3 #include <SPI.h>
4 #include <MFRC522.h>
5
6 #define RST_PIN      9
7 #define SS_PIN       10
8
9 MFRC522 mfrc522(SS_PIN, RST_PIN);
10
11 void setup() {
12     pinMode(2, OUTPUT);
13     pinMode(3, OUTPUT);
14     digitalWrite(2, LOW);
15     digitalWrite(3, HIGH);
16     Serial.begin(9600);
17     SPI.begin();
18     mfrc522.PCD_Init();
19 }
20
21
```

At the bottom, the 'Serial Monitor' window is open, showing the output of the sketch. The output consists of two lines of data: '200069, Babloo' and '205930, Karen'. The Serial Monitor window has a text input field for sending messages to the 'Arduino Uno' on 'COM5', and dropdown menus for 'No Line Ending' and '9600 baud'.

Writing on a RFID card



The screenshot displays the Arduino IDE interface. The top menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for checking, running, and uploading code, along with a dropdown menu set to 'Arduino Uno'. The main editor window shows the sketch 'rfid_write_personal_data.ino' with the following code:

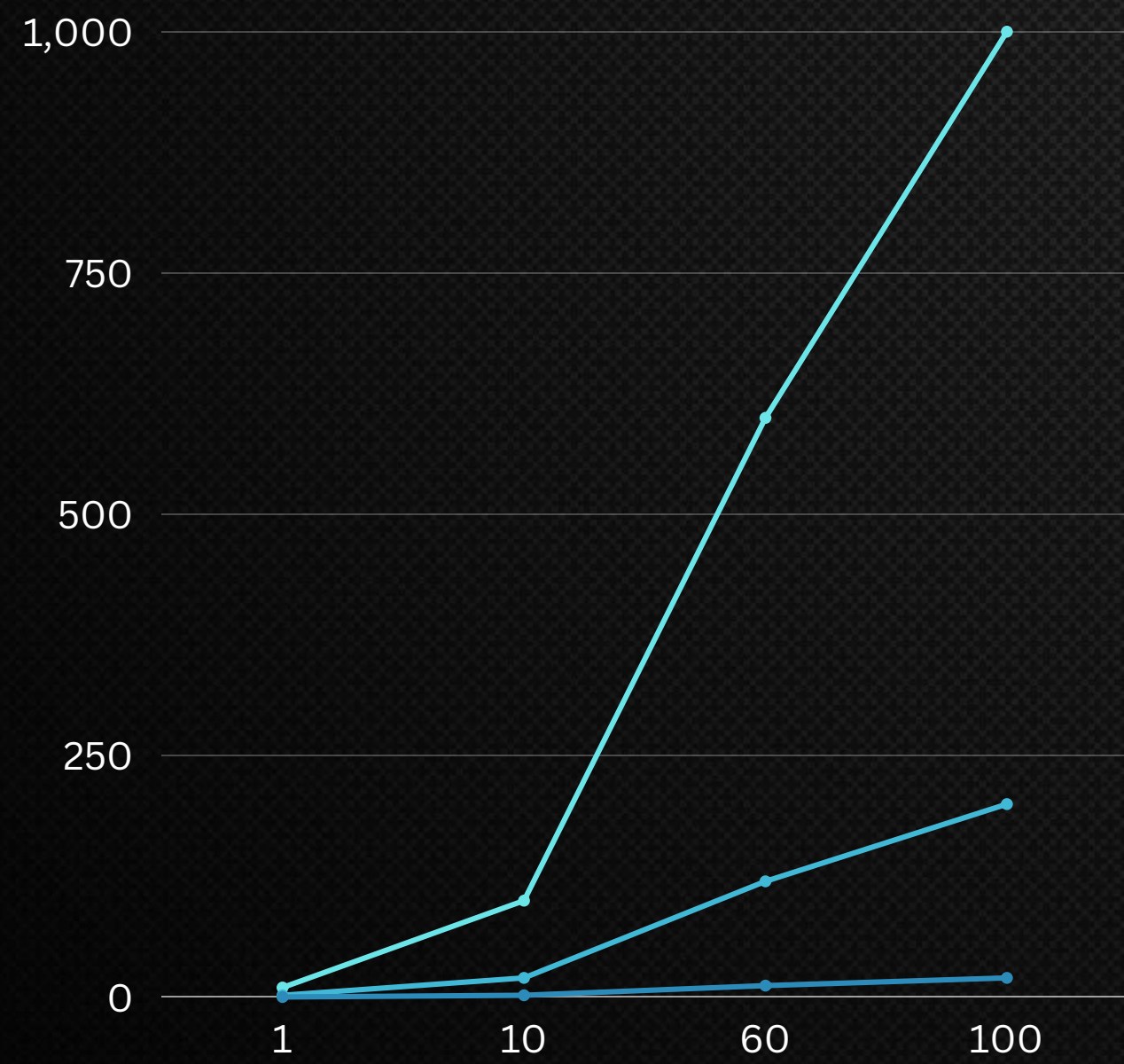
```
37 }
38 Serial.print(F(" PICC type: "));
39 MFRC522::PICC_Type piccType = mfrc522.PICC_GetType(mfrc522.uid.sak);
40 Serial.println(mfrc522.PICC_GetTypeName(piccType));
41
42 byte buffer[34];
43 byte block;
44 MFRC522::StatusCode status;
45 byte len;
46
47 Serial.setTimeout(20000L);
48 Serial.println(F("Type Full name, ending with #"));
49 len = Serial.readBytesUntil('#', (char *) buffer, 30);
50 for (byte i = len; i < 30; i++) buffer[i] = ' ';
51
52 block = 1;
53
54 status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, block, &key, &(mfrc522.uid));
55 if (status != MFRC522::STATUS_OK) {
56     Serial.print(F("PCD_Authenticate() failed: "));
57     Serial.println(mfrc522.GetStatusCodeName(status));
58 }
```

Below the editor is the 'Serial Monitor' tab, which is active. It shows the output of the sketch:

```
Write personal data on a MIFARE PICC
Card UID: 4A C3 C5 10 PICC type: MIFARE 1KB
Type Full name, ending with #
PCD_Authenticate() success:
MIFARE_Write() success:
MIFARE_Write() success:
Type Roll No., ending with #
MIFARE_Write() success:
MIFARE_Write() success:
```

The status bar at the bottom indicates 'Ln 50, Col 55', 'UTF-8', and 'Arduino Uno on COM5'.

Observations



Method	Total number of students			
	1	10	60	100
Manual Entry	10 sec	100 sec	600 sec	1000 sec
Bar Code	2 sec	20 sec	120 sec	200 sec
RFID technology	0.2 sec	2 sec	12 sec	20 sec

Working idea

To make an exit/entry from campus we do use a permission at portal. I am now working on to check the received information from a database with students requested for entry/exit, and checking a student's card if they have been permitted.

Thank You

Harsh Agrawal
200407

harsha20@iitk.ac.in

Implementations

1. Made a working model, and implemented it to store data from the tag.
2. We can write information on a card/tag through the same setup.
3. Storing this data on a CSV file.
4. In process of making a automated system for permission based entry/exit system.

Implementations

100% Project contribution

Harsh Agrawal(200407)