RFID employee tags

Seceleanu Ioan-Adrian

Introduction

I created a solution to monitoring employees' check in and check out of office using programmable RFID tags which contain the name of each employee, each employee having a unique tag with a UID (unique ID) and key (password of the tag/data encryption).

When employees come to the office, they scan their RFID tag, same as when they leave the office. It can be stored in a log that shows each employees' check in and check out dates and hours. **Great for monitoring emplyee activity**

General Description

We use the generic Arduino UNO paired with a RFID read/write module.

I was thinking how can I make a project that actually has a usage far away from personal use, so I came with the idea of making this. Each employee from a certain company has his own unique tag that contains his name. When someone gets employed, we use the MASTERwrite program to encrypt his data on a new tag, knowing the UID and the key of the tag, then giving him the tag. When he gets at the office he will scan the tag at check in and check out, which will be stored in a log accessible by the boss, making it easy for him to know how many hours an employee works a day, who was and who wasn't at the office at a certain day.

I made two programs. One is the **MASTERwrite**, which will be given to the boss of the company so he can program the tags for each employee with his personal details. The other one is the **reader**, which is the program that will read the data off a tag. The reader program can only show the name of the employees, not the UID and key of their tag. **(safety measures)**

Easy employee activity monitoring!

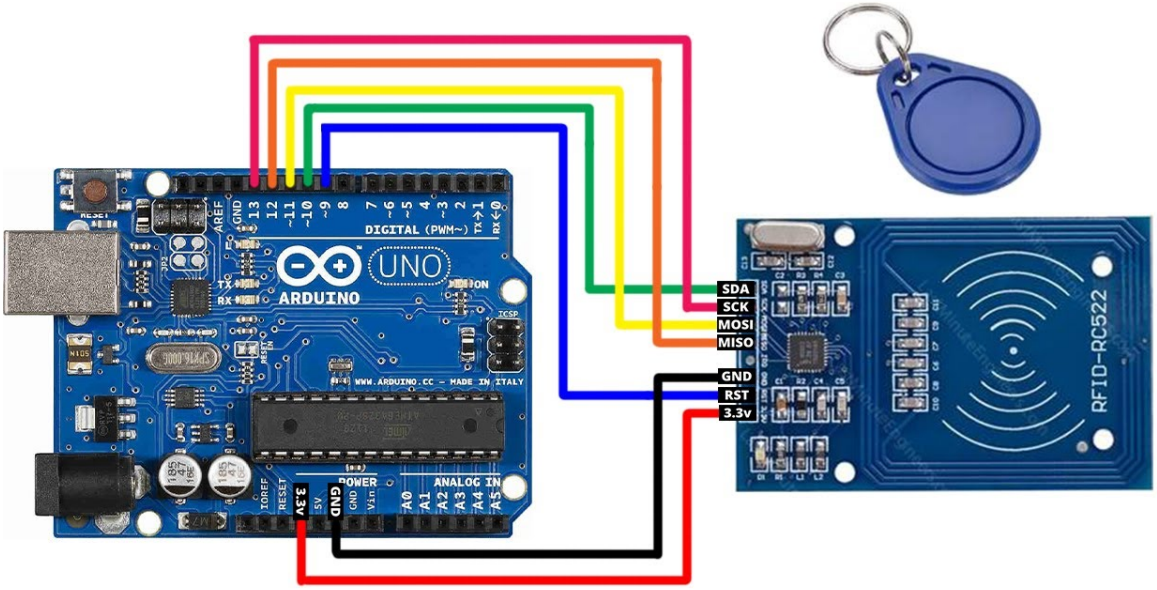
Hardware Design

Components used:

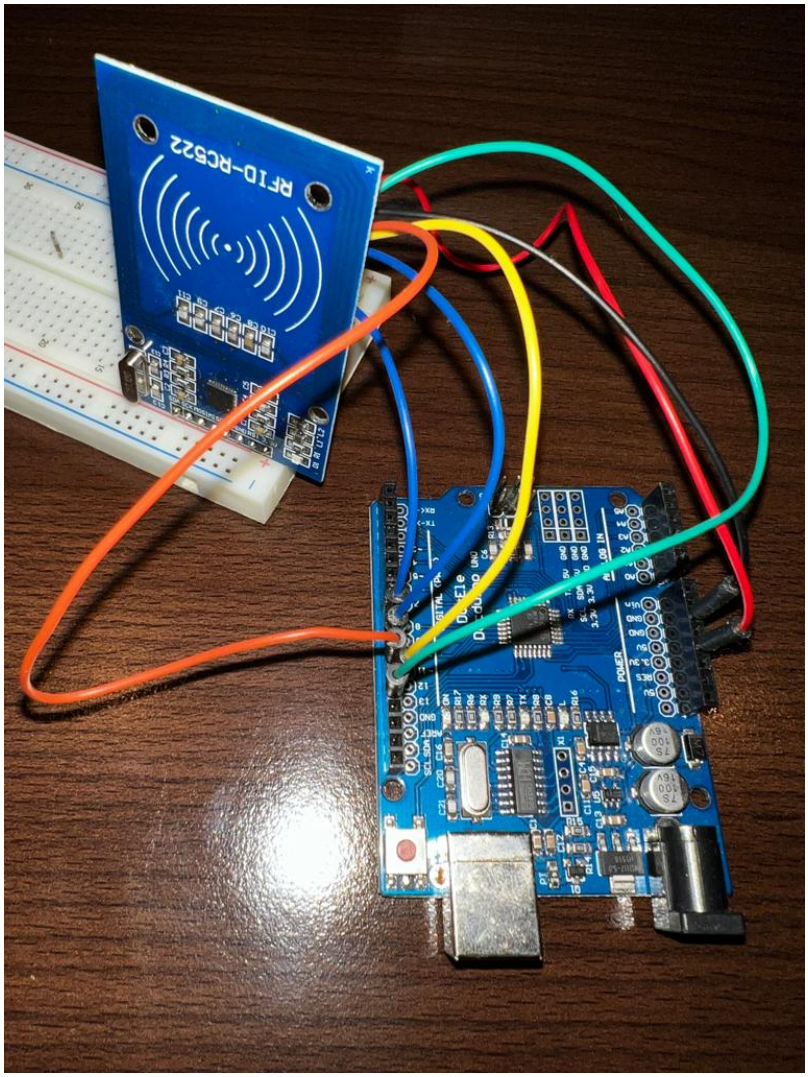
1. Arduino UNO

2. RC522 RFID reader/writer module

Arduino connection scheme:



The hardware:



Software Design

I made 2 programs, one is the MASTERwrite and the second is the reader:

1. MASTERwrite

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

#define RST_PIN 9 #define SS_PIN 10

MFRC522 mfrc522(SS_PIN, RST_PIN); *Create MFRC522 instance void setup() { Serial.begin(9600);* Initialize serial communications with the PC

```
SPI.begin();                // Init SPI bus
mfrc522.PCD_Init();         // Init MFRC522 card
Serial.println(F("Write personal data of the employee: "));
```

}

void loop() {

```
// Prepare key - all keys are set to FF FF FF FF at chip delivery from the
factory.
MFRC522::MIFARE_Key key;
for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;
```

```
// Reset the loop if no new card present on the sensor/reader. This saves
the entire process when idle.
if ( ! mfrc522.PICC_IsNewCardPresent()) {
  return;
}
```

```
// Select one of the cards
if ( ! mfrc522.PICC_ReadCardSerial()) {
  return;
}
```

```
Serial.print(F("Card UID:"));    //Dump 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);
}
```

```
byte buffer[34];
byte block;
MFRC522::StatusCode status;
byte len;
```

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

```
Serial.setTimeout(20000L) ;     // wait until 20 seconds for input from
serial
// Ask personal data: Family name
Serial.println(F("Type Family name, ending with #."));
len = Serial.readBytesUntil('#', (char *) buffer, 30) ; // read family name
from serial
for (byte i = len; i < 30; i++) buffer[i] = ' ';     // pad with spaces
```

```
block = 1;
//Serial.println(F("Authenticating using key A..."));
status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, block,
&key, &(mfrc522.uid));
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("PCD_Authenticate() failed: "));
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}
else Serial.println(F("PCD_Authenticate() success: "));
```

```
// Write block
status = mfrc522.MIFARE_Write(block, buffer, 16);
```

```
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("MIFARE_Write() failed: "));
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}
else Serial.println(F("MIFARE_Write() success: "));
```

```
block = 2;
//Serial.println(F("Authenticating using key A..."));
status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, block,
&key, &(mfrc522.uid));
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("PCD_Authenticate() failed: "));
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}
```

```
// Write block
status = mfrc522.MIFARE_Write(block, &buffer[16], 16);
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("MIFARE_Write() failed: "));
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}
else Serial.println(F("MIFARE_Write() success: "));
```

```
// Ask personal data: First name
Serial.println(F("Type First name, ending with #"));
len = Serial.readBytesUntil('#', (char *) buffer, 20) ; // read first name
from serial
for (byte i = len; i < 20; i++) buffer[i] = ' ';     // pad with spaces
```

```
block = 4;
//Serial.println(F("Authenticating using key A..."));
status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, block,
&key, &(mfrc522.uid));
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("PCD_Authenticate() failed: "));
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}
```

```
// Write block
status = mfrc522.MIFARE_Write(block, buffer, 16);
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("MIFARE_Write() failed: "));
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}
else Serial.println(F("MIFARE_Write() success: "));
```

```
block = 5;
//Serial.println(F("Authenticating using key A..."));
status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, block,
&key, &(mfrc522.uid));
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("PCD_Authenticate() failed: "));
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}
```

```
// Write block
status = mfrc522.MIFARE_Write(block, &buffer[16], 16);
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("MIFARE_Write() failed: "));
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}
else Serial.println(F("MIFARE_Write() success: "));
```

```
Serial.println(" ");
mfrc522.PICC_HaltA(); // Halt PICC
mfrc522.PCD_StopCrypto1();  // Stop encryption on PCD
```

}

2. reader

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

#define RST_PIN 9 #define SS_PIN 10

MFRC522 mfrc522(SS_PIN, RST_PIN); *Create MFRC522 instance void setup() { Serial.begin(9600);* Initialize serial communications with the PC

```
SPI.begin();                                                    // Init SPI
bus
mfrc522.PCD_Init();                                             // Init
MFRC522 card
Serial.println(F("Read personal data on a MIFARE PICC:"));    //shows in
serial that it is ready to read
```

}

void loop() {

```
// Prepare key - all keys are set to FFFFFFFFFFFFh at chip delivery from the
factory.
MFRC522::MIFARE_Key key;
for (byte i = 0; i < 6; i++) key.keyByte[i] = 0xFF;
```

```
//svariables we need
byte block;
byte len;
MFRC522::StatusCode status;
```

```
// Reset the loop if no new card present on the sensor/reader. This saves
the entire process when idle.
if ( ! mfrc522.PICC_IsNewCardPresent()) {
  return;
}
```

```
// Select one of the cards
if ( ! mfrc522.PICC_ReadCardSerial()) {
  return;
}
```

```
Serial.println(F("**Card Detected:**"));  // prints card detected if the
rfid can read any cards
```

```
Serial.print(F("Name: "));
```

```
byte buffer1[18];
```

```
block = 4;
len = 18;
//------------------------------------------- GET FIRST NAME
```

```
status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, 4, &key,
&(mfrc522.uid)); // if the card is encrypted with a password different than
the generic one FF FF FF FF, the program will display "autentification
failed"
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("Authentication failed: "));
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}
```

```
status = mfrc522.MIFARE_Read(block, buffer1, &len); // if the card is
encrypted and the program cannot read data from it, it will display "reading
failed"
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("Reading failed: "));
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}
```

```
//PRINT FIRST NAME
for (uint8_t i = 0; i < 16; i++)
{
  if (buffer1[i] != 32)
```

```
  {
    Serial.write(buffer1[i]);
  }
}
Serial.print(" ");
```

```
//---------------------------------------- GET LAST NAME
```

```
byte buffer2[18];
block = 1;
```

```
status = mfrc522.PCD_Authenticate(MFRC522::PICC_CMD_MF_AUTH_KEY_A, 1, &key,
&(mfrc522.uid)); //line 834
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("Authentication failed: "));
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}
```

```
status = mfrc522.MIFARE_Read(block, buffer2, &len);
if (status != MFRC522::STATUS_OK) {
  Serial.print(F("Reading failed: "));
  Serial.println(mfrc522.GetStatusCodeName(status));
  return;
}
```

```
//PRINT LAST NAME
for (uint8_t i = 0; i < 16; i++) {
  Serial.write(buffer2[i] );
}
```

```
//---------------------------------------
```

```
Serial.println(F("\n**End Reading**\n"));
```

```
delay(1000); //change value if you want to read cards faster
```

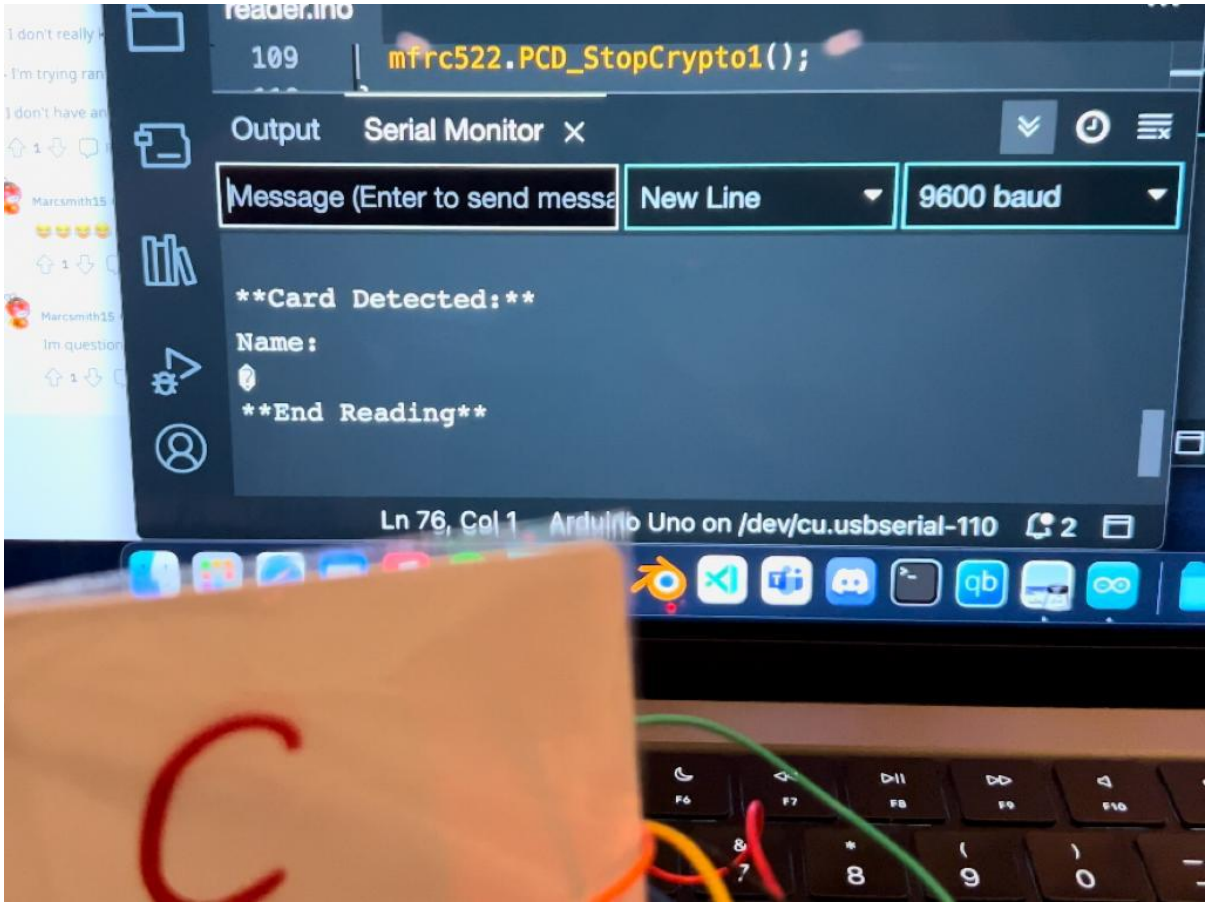
```
mfrc522.PICC_HaltA();
mfrc522.PCD_StopCrypto1();
```

```
}
```

Results Obtained

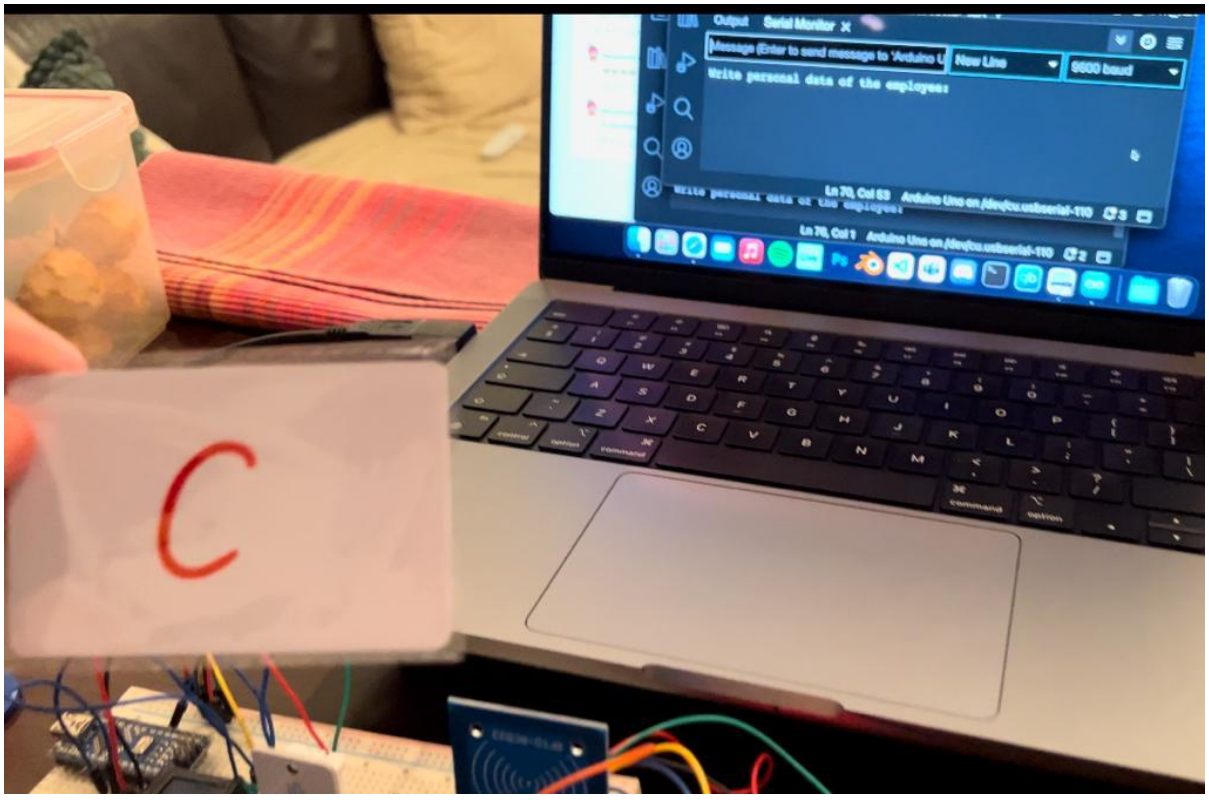
As for results, i took some photos of how the whole proccess works:

1. We scan a RFID tag using the reader program:



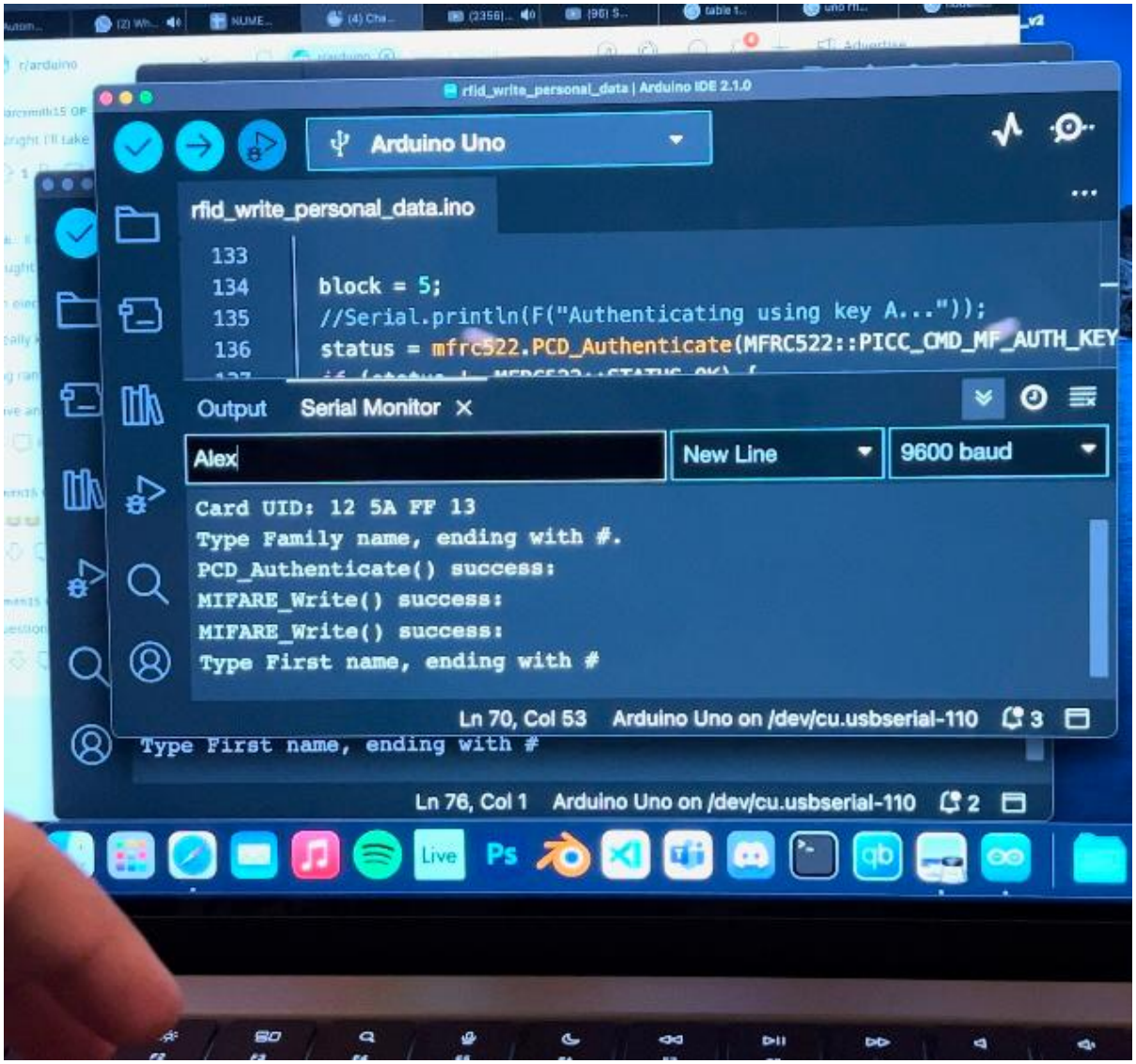
We see it has no name on it, it's empty.

2. It's time to write the name of a new employee on it:

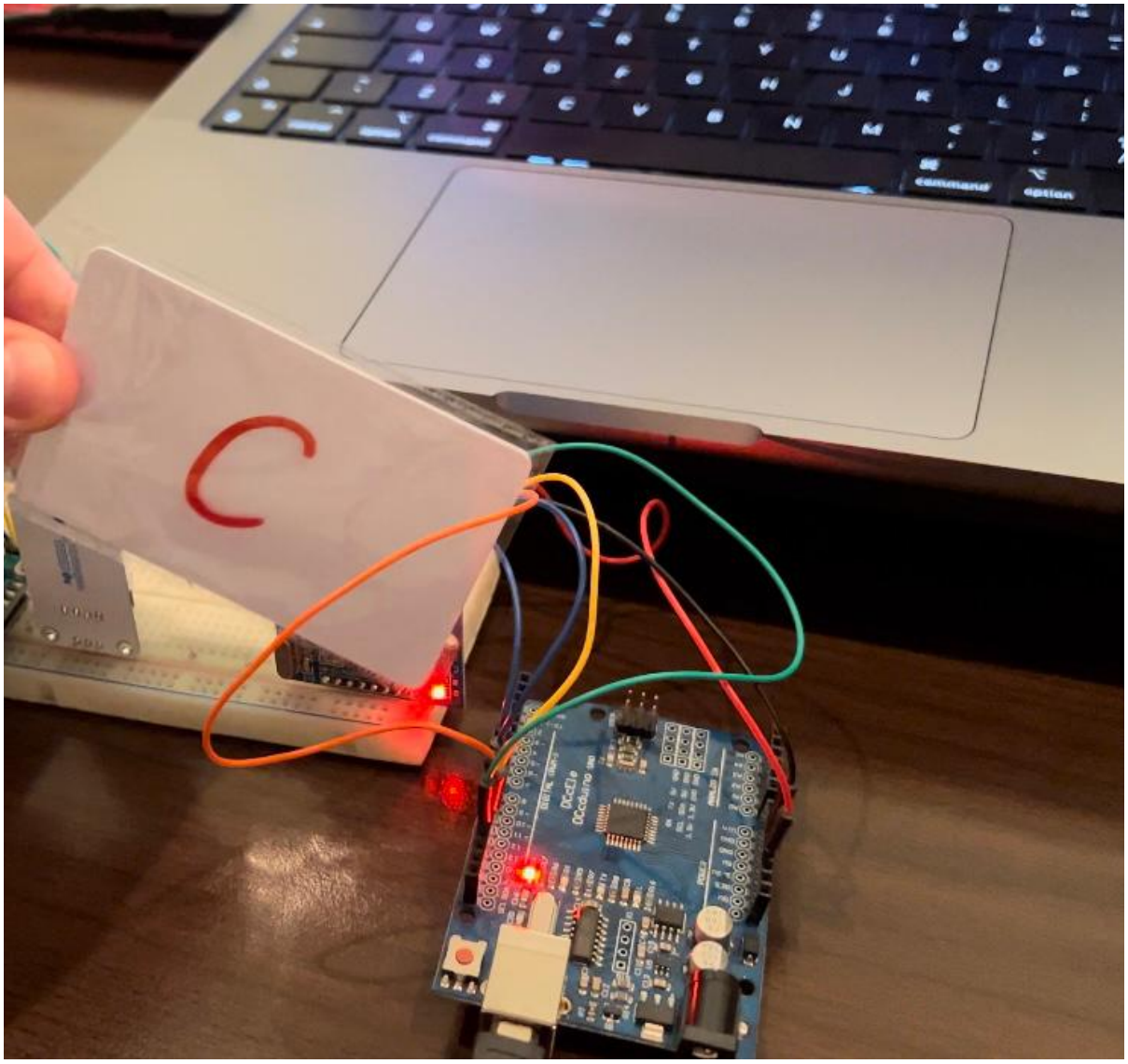


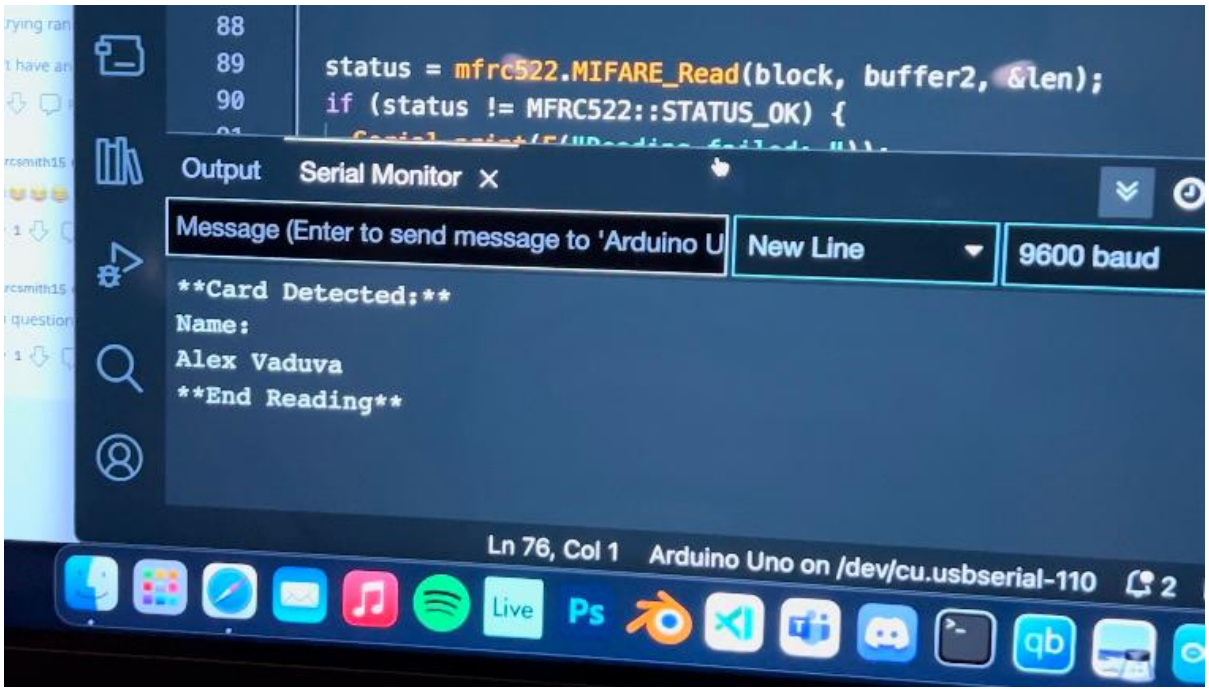
I started MASTERwrite program in the background, now we just have to put the tag on the RFID module;

3. After we put the tag on the RFID module, the MASTERwrite program will read it's UID, verify it's key and then it will ask us to put the new employee name on it:

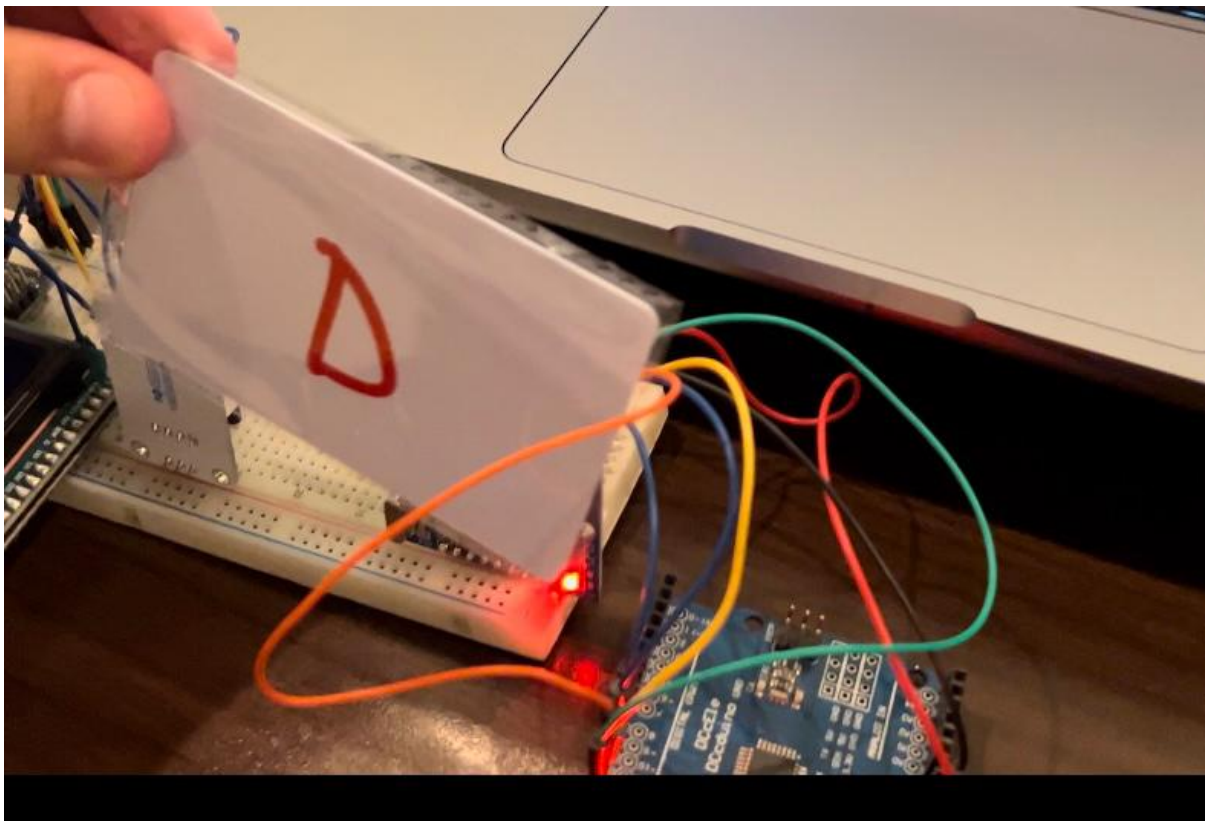


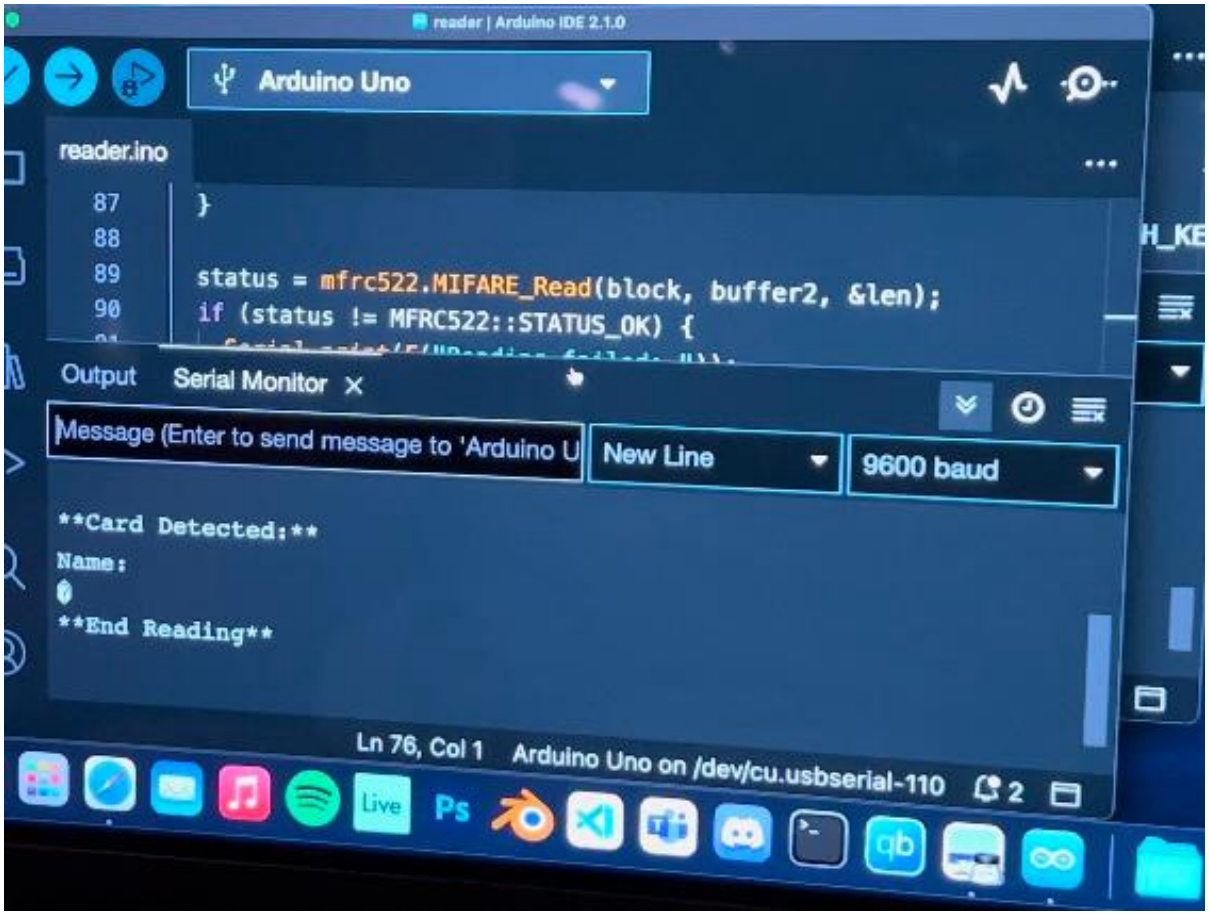
4. After populating the tag with the new employee name, we will use the reader program to check if it was done properly:



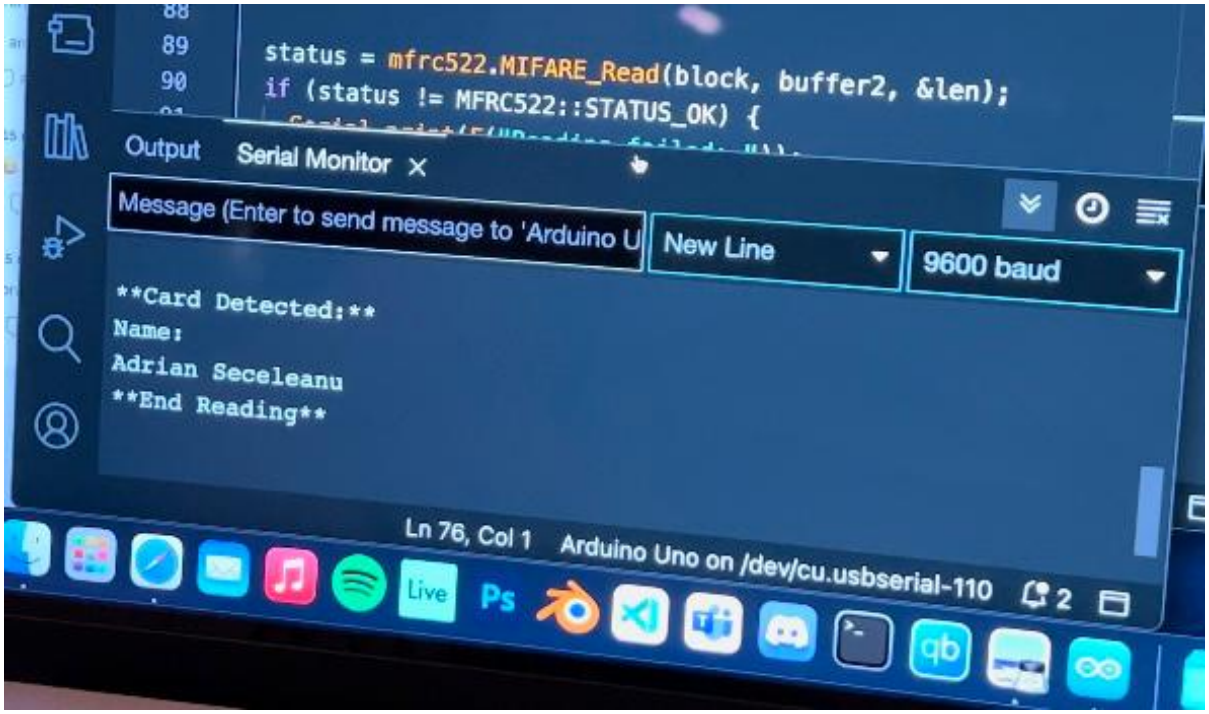


5. We check another empty RFID tag just to make sure the software works properly (after reading a card, it repeats the loop of waiting for another card to be read. When another card is read it will display the new card data) :





6. And after that we read another RFID tag, which happens to have my name on it.



Conclusions

Working with the RC522 RFID module is quite a headache because every time you want to make a change on the data of a tag you need to check if the key is okay. Tags usually have a generic key when they are new, which is FF FF FF FF, but you can change that key to make it encrypted (you will need to know the key to access the data on that certain tag) and if you do not note the key and the

UID of that certain tag, you will not be able to access the information on it (unless you can backdoor the tag using some sorts of hacking software).

The software was quite a pain because i had to get informed on how buffers work in arduino for data reading, I had to search for every commands RFID uses and they're not so common like an LCD screen commands or some sensor commands.

Every time i want to write data on a certain card i had to check the status of MIFARE, use a buffer and stop it after writing something.

The tags can only be programmed when they are directly on the RFID module and not moving. **If you start writing employee name on a tag and the tag doesnt stay put on the module, the whole operation will fail and you need to try again!**

Download

Jurnal

Bibliography/Resources

https://circuits4you.com/2018/10/03/interfacing-of-rfid-rc522-with-arduino-uno/

https://circuitdigest.com/microcontroller-projects/interfacing-rfid-reader-module-with-arduino

https://www.youtube.com/watch?v=pdBrvLGH0PE&themeRefresh=1

Export to PDF