

RFID Attendance System

Introducere

Scopul proiectului este de a monitoriza scanarile unor RFID tags si stocarea unor log-uri in Cloud cu ajutorul unui modul de Ethernet. Ar putea fi integrat in sistemele de acces din blocuri sau cladiri de birouri cu scopul de a pastra datele in mod persistent si de a putea analiza diverse metrice calculate pe baza datelor cumulate. De asemenea, poate fi util in cazul companiilor care monitorizeaza in detaliu activitatea angajatilor in birou, intrucat pot vedea zonele care sunt accesate si timpul petrecut in acele zone. In cazul acestui proiect datele sunt puse in Google Drive via Google Sheets, intr-o forma tabelara, dar se pot stoca in multe variante - bazele de date reprezinta o alta modalitate de a persista datele.

Descriere generală

Schema bloc care cuprinde toate componentele software, cat si hardware poate fi vazuta mai jos.



Descriere module si interactiunea dintre acestea:

Sistemul are 3 componente hardware - *RFID-RC522*, *Ethernet Shield 1* si *Arduino UNO* (cu microcontroller-ul integrat *ATMega328P*) - si 2 componente software - *PushingBox* si *Google Drive*. In cele ce urmeaza vom discuta despre fiecare in parte, cum interactioneaza intre ele si rolul lor in acest proiect.

RFID_RC522

Acest modul este un RFID reader generic, bazat pe *MFRC522*, un reader/writer integrat in IC pentru comunicarea contactless la frecventa de 13.56MHz. Transmisorul intern al *MFRC522* poate comunica doar cu carduri *MIFARE* (ISO/IEC 14443) si comunicarea cu microcontroller-ul este facuta prin Serial Peripheral Interface (SPI). Acesta trebuie alimentat intre 2.5V si 3.3V, conform datasheet-ului.

Ethernet Shield 1

Arduino Ethernet Shield V1 permite unei plăci Arduino să se conecteze la internet. Se bazează pe cipul ethernet Wiznet W5100. Wiznet W5100 oferă o stivă de rețea (IP) capabilă atât de TCP, cât și de UDP. Acesta suportă până la patru conexiuni socket simultane. Are conexiune standard RJ-45, cu un transformator de linie integrat și Power over Ethernet activat si comunica cu Arduino folosind bus-ul SPI. Din cauza ca avem 2 device-uri - *slaves* - care impart magistrala de SPI cu Arduino - *master* - sunt folositi 2 pini separati pentru *Slave Select*, deoarece numai unul poate fi activ la un moment dat.

Arduino UNO - ATMega328P

Reprezinta piesa centrala a sistemului avand rol de gestiune a celorlalte componente si prin acesta se vor primi si trimite comenzi catre celelalte module.

PushingBox - 3rd Party

Este un Cloud care poate trimite notificări pe baza apelurilor API. Dintr-o singură cerere, se pot trimite mai multe notificări, cum ar fi un Push, un Tweet, un Email, în timp real. Necesitatea acestui 3rd Party este datorată comunicării pe care o impune API-ul Google Sheets, i.e. HTTPS. Deoarece stiva TCP/IP pe care ne-o pune la dispoziție W5100 nu permite și un layer suplimentar de SSL/TLS, peste cel de TCP, atunci calea cea mai ușoară este de a folosi alt API pentru a putea urca datele în Google Drive via Google Sheets.

Google Drive

La fel ca PushingBox, este un serviciu în Cloud, care ne oferă o suită de alte servicii, cu ajutorul cărora putem salva date în mod persistent în Cloud. În cazul acestui proiect s-a folosit Google Sheets pentru a păstra datele sub formă tabelară, iar pentru a interacționa cu PushingBox am folosit un alt serviciu, numit Google Apps Script. Acesta din urmă este o platformă pe care se pot urca script-uri scrise în diverse limbaje, precum JavaScript, care pot interacționa cu stiva de servicii oferită de Google Workspace, prin intermediul unor biblioteci. Acestea sunt rulate în Cloud.

Componentele prezentate mai sus interacționează conform diagramei anterioare în următoarea manieră:

1. Tag-ul este citit de reader-ul RFID și sunt trimise metadate prin SPI microcontroller-ului ATmega328P.
2. Acesta din urmă verifică validitatea ID-ului asociat tag-ului (se cunosc în prealabil toate card-urile eliberate pentru sistemul nostru, la fel cum într-un bloc cu interfon se cunosc toate cartelele care au acces)
3. Dacă tag-ul este valid atunci se trimite prin intermediul controller-ului de Ethernet (comunica cu ATmega328P prin SPI), care acționează ca un *Client*, un request HTTP către PushingBox. În acest request HTTP se mai pun și 2 query parameters care reprezintă datele pe care le vom pune în tabel (allowed și id-ul cardului).
4. PushingBox pune la dispoziție o interfață web pe care se pot configura web-hooks și diverse alte scenarii. În cazul proiectului s-a creat un hook către script-ul care a fost deployed în Cloud (acesta din urmă are asociat un URL). Acest hook primește un ID, numit în cadrul PushingBox *devid*, prin care îl putem identifica și apela cu URL-ul de bază pus la dispoziție în API-ul lor.
5. Când PushingBox va primi request-ul, va identifica hook-ul cu *devid* și va face la rândul său un request **HTTPS**, pentru a putea fi acceptat de serviciile Google.
6. În cele din urmă, App Script va primi request-ul de la PushingBox și va rula un script scris în JavaScript care va interacționa cu Google Sheets și va putea adăuga o nouă linie în tabela activă.

Hardware Design

Lista de piese:

1. Ethernet controller - WIZnet 5100
2. Microcontroller - ATmega328P
3. Reader/Writer IC: MFRC522

Schema electrică:



Software Design

- Limbaj de programare: Arduino (C)
- Mediu de dezvoltare si compilare: Arduino 1.8.15
- Bibliotecile folosite:

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

- Aplicatii 3rd Party: PushingBox (<https://www.pushingbox.com/>), Google Drive (<https://www.google.com/drive/>)

Firmware-ul:

Pentru a putea comunica extern, placa are nevoie de o adresa IP. Aceasta este asignata automat prin *DHCP*, in codul:

```
if ( ! Ethernet.begin(mac) )
```

, insa am folosit si o adresa IP statica in caz ca nu se poate negocia cu DHCP. De asemenea, orice device are nevoie si de o adresa MAC, insa doar Shield-urile mai noi vin insotite de un MAC predefinit. Asadar, se poate defini in cod o adresa MAC arbitrara.

Se initializeaza si Seriala cu *baud* 9600, SPI-ul si RFID-ul.

```
Serial.begin(9600);
SPI.begin();
mfrc522.PCD_Init();
```

Pentru asteptarea unui card in proximitate, vom apela in loop:

```
if( ! mfrc522.PICC_IsNewCardPresent() )
    return;

if( ! mfrc522.PICC_ReadCardSerial() )
    return;
```

Doua define-uri foarte importante sunt cele de mai jos:

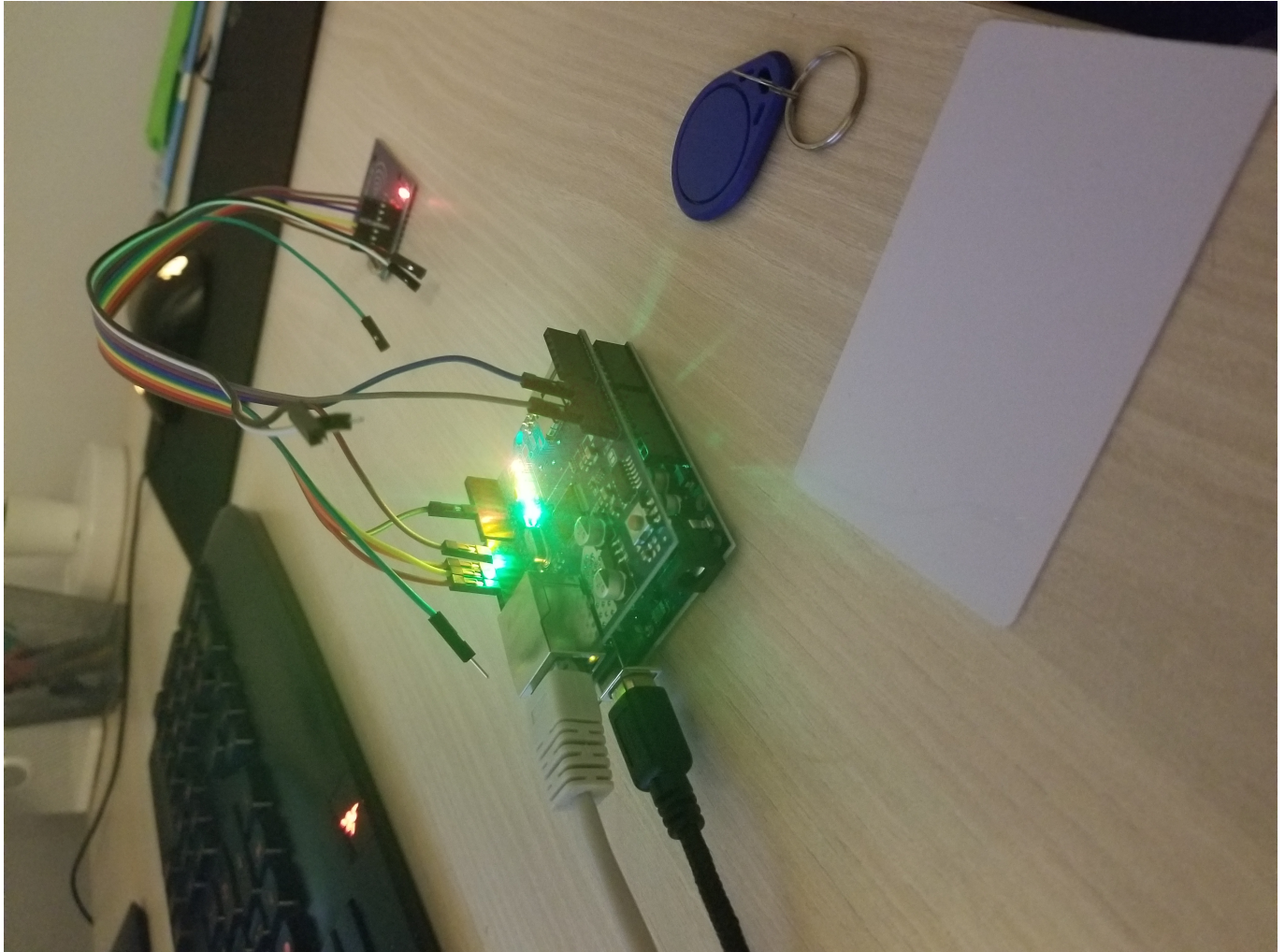
```
#define SS_PIN 4
#define RST_PIN 9
```

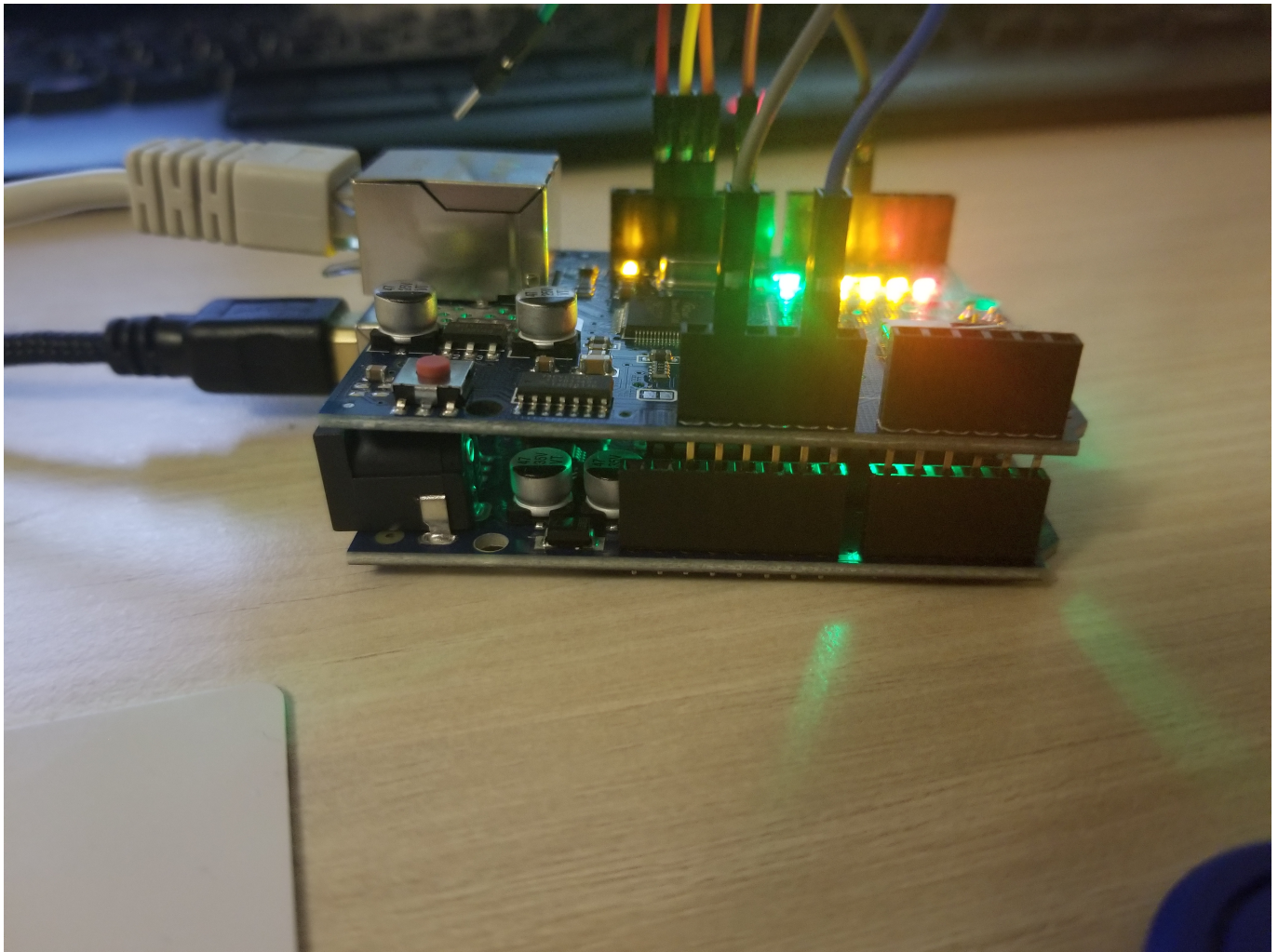
, fiindca ele definesc port-urile pe care le va folosi RFID-ul, in special *SS_PIN*. Acest pin este in mod obisnuit 10, insa din cauza ca avem 2 *slaves* care comunica prin SPI cu microcontroller-ul, va trebui sa avem un pin diferit pentru *Slave Select*. De aceea, Shield-ul va folosi pin-ul 10, iar reader-ul va folosi pin-ul 4.

In codul care urmeaza se fac verificari asupra id-ului citit, iar daca acesta este valid se va apeca

functia care face requestul HTTP cu valoarea 1, insemand Access Granted, iar 0 pentru Forbidden.

Rezultate Obținute





Concluzii

A fost un proiect foarte practic, care poate fi integrat in multe contexte si multe industrii. Pot fi aduse multe imbunatatiri si features.

Un demo al proiectului poate fi vazut la:

<https://drive.google.com/drive/folders/1VHvZFZ2Zlqa5S0xpfgaWkEoQl2FtF4mu?usp=sharing>.

Download

1. [Arhiva proiect](#)

Bibliografie/Resurse

Resurse hardware:

1. <https://www.arduino.cc/>

2. <https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf>
3. https://www.sparkfun.com/datasheets/DevTools/Arduino/W5100_Datasheet_v1_1_6.pdf

Resurse software:

1. <https://www.pushingbox.com/api.php>
2. <https://developers.google.com/apps-script/guides/sheets>
3. <https://www.lucidchart.com>

Documentatia in format PDF:

1. [Documentatie](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2021/amocanu/attendance-system>



Last update: **2021/06/03 02:30**