

Proiect Sisteme Incorporate

Camera de cămin inteligentă

Bogdan Bercea, 1.1

Bogdan Turdășan, 6.2

Bogdan Tomici, 6.1

Luigi Ionuț Bolovan, 1.2

Enunț

Realizarea unui sistem de tip smart home pentru camera de cămin cu următoarele componente:

- Kit RFID RC522 pe 13.56 MHz pentru citirea de carduri de acces
- Încuietoare electromagnetică
- Bandă LED
- Senzor de umiditate și temperatură de tip DHT22 pentru monitorizarea temperaturii și umidității din interiorul camerei
- Module wireless de tip ESP8266 responsabile cu comunicarea cu baza de date din cloud pentru acces remote la hardware
- Aplicație Android responsabilă cu controlul senzorilor, al instalației de iluminat și a senzorilor de umiditate și temperatură
- Bază de date Firebase
- Arduino UNO

Descrierea plăcii de dezvoltare utilizate

Arduino UNO

Arduino UNO este o placă de dezvoltare bazată pe microcontrolerul ATmega328p. Este echipată cu un set de pini de input/output digitali și analogi care pot fi conectați la interfețele diferitelor plăci de expansiune sau la alte circuite.

Specificații tehnice:

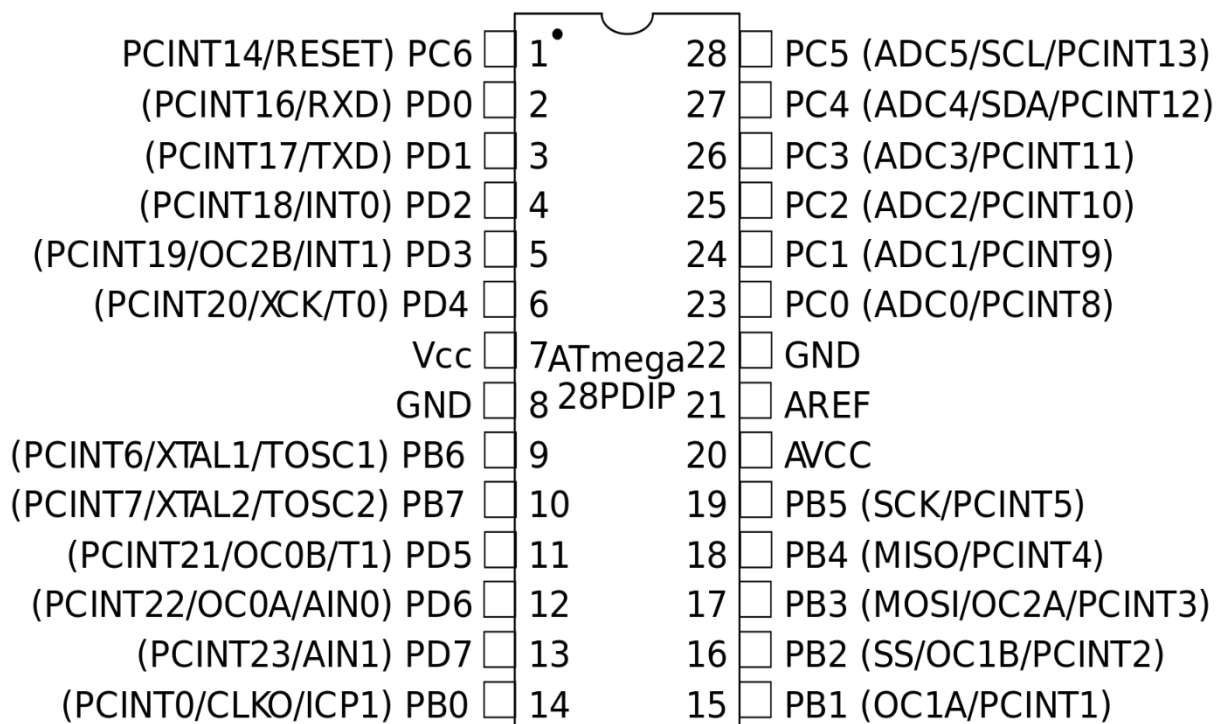
- Microcontroler: ATmega328P
- Operating voltage: 5 volți
- Input voltage: 7 – 20 volți
- Pini I/O digitali: 14

- UART: 1
- I2C: 1
- SPI: 1
- Analog input pins: 6
- Curent continuu per I/O pin: 20mA
- Curent continuu per 3.3V pin: 50mA
- Memorie flash: 32KB din care 0.5kb utilizați de bootloader
- SRAM: 2KB
- EEPROM: 1KB
- Clock speed: 16MHz
- Lungime: 68.6mm
- Lățime: 53.4mm
- Greutate: 25g

Microcontrolerul ATmega328

ATmega328 este un cip microcontroler creat de către Atmel și face parte din seria de megaAVR. Este un microcontroler de tip RISC ce combină 32 KB de memorie flash cu capabilități read-while-write, 1 KB EEPROM, 2 KB SRAM, 23 linii GPIO, 32 de regiștri, trei timere, USART, SPI serial port, convertor A/D, timer watchdog programabil cu oscilator intern și cinci moduri de economisire de energie. Device-ul poate opera între 1.8 și 5.5 volți.

Diagrama cu aranjamentul pinilor microcontroler-ului ATmega328



Descrierea pinilor VCC – Sursa de curent GND – Masa

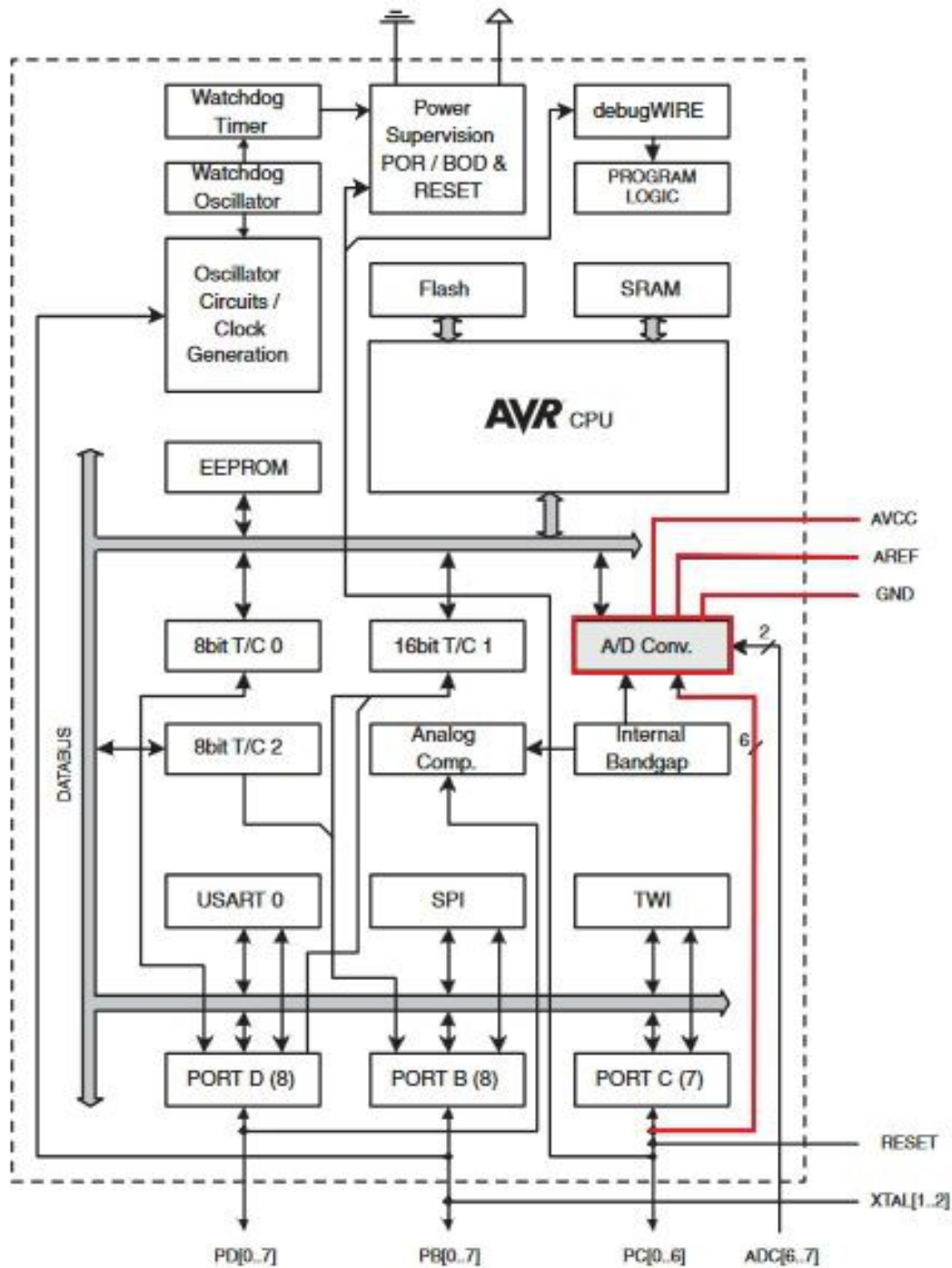
Port A (PA7 .. PA0) Port-ul A servește drept port de intrări analogice pentru Convertorul A/D. Port-ul A servește de asemenea și ca un port bidirecțional I/O de 8 biți, în cazul în care Convertorul A/D nu este folosit. Pinii de port pot fi conectați opțional la VCC prin rezistori interni, (selectați pentru fiecare bit). Buffer-ele de ieșire ale Portului A au caracteristici de amplificare .

Port B (PB7.. PB0) Portul B este un port I/O de 8 biți bidirecțional cu rezistori interni (opționali). Buffer-ele de ieșire ale Port-ului B au caracteristici de amplificare. Port-ul B îndeplinește de asemenea funcții speciale ale microcontrolerului ATmega 16

Port C (PC7...PC0) Portul C este un port I/O de 8 biți bidirecțional cu rezistori interni (opționali). Buffer-ele de ieșire ale Port-ului C au caracteristici de amplificare.

Port D (PD7...PD0) Portul D este un port I/O de 8 biti bidirectional cu rezistori interni conectați optional la VCC (selectați pentru fiecare bit). Buffer-ele de output ale Port-ului D au caracteristici de amplificare. Port-ul D îndeplinește de asemenea funcții speciale ale ATmega 16. Reset Un nivel scăzut la acest pin mai mare ca durata decât o valoare prestabilită, va genera o inițializare.

Schema bloc a microcontroler-ului ATmega328



Modulele utilizate ale microcontrolerului

1.SPI

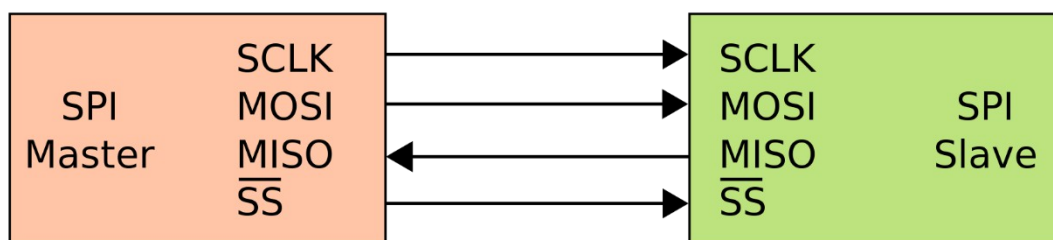
Interfața serială SPI este o interfață sincronă standard de mare viteză, ce operează în mod full duplex. Numele ei a fost dat de Motorola. Ea e folosită ca sistem de magistrală serială sincronă pentru transmiterea de date, unde circuitele digitale pot să fie interconectate pe principiul master-slave. Aici, modul master/slave înseamnă că dispozitivul (circuitul) digital master inițiază cuvântul de date. Mai multe dispozitive (circuite) digitale slave sunt permise cu slave select individual, adică cu selectare individuală.

De obicei, există trei linii comune pentru toate device-urile:

- MISO(Master In Slave Out) – linia prin care slave trimite date către master
- MOSI(Master Out Slave In) – linia prin care master trimite date către slave
- SCK(Serial Clock) – clock-ul ce sincronizează transmisia de date generată de master,

Și o linie specifică pentru fiecare device:

- SS(Slave Select) – un pin pe fiecare device pe care masterul îl folosește pentru a activa sau a dezactiva device-uri.



Un slave poate comunica cu masterul doar dacă SS pin este pe „0” logic. În caz contrar, îl ignoră.

Pentru a începe comunicarea, masterul mai întâi configurează ceasul, folosind o frecvență mai mică sau egală cu maximul frecvenței suportată de slave. Aceste frecvențe sunt de obicei în intervalul 1-70 MHz. Atunci masterul setează slave select-ul pe nivelul „0” pentru chip-ul dorit. În cazul în care este nevoie de o perioadă de așteptare, masterul va aștepta cel puțin acea perioadă înainte de a începe ciclurile de clock.

În timpul fiecărui ciclu de clock SPI, apare o transmisie full duplex:

- Masterul trimite un bit pe MOSI
- Slave-ul trimite un bit pe MISO

În mod normal, transmisia implică existența a doi regiștri de date de o lungime oarecare a cuvântului, cum ar fi opt biți, unul situat în dispozitivul master și celălalt în dispozitivul slave. Aceștia sunt conectați într-o configurație de tip inel. Informația este de obicei transferată începând cu cel mai semnificativ bit(MSB) și continuând bit cu bit până se transferă și cel mai nesemnificativ bit(LSB) pentru același registru. În această fază putem afirma că cele două dispozitive, master și slave, și-au schimbat valorile din regiștri. Imediat după, fiecare dispozitiv citește valoarea stocată în registrul de date și o prelucrează, cum ar fi scrierea într-o locație de memorie. Dacă sunt mai multe date de schimbat, regiștrii de schimb sunt încărcăți cu noi date și procesul se repetă.

Modulul SPI este folosit în cadrul proiectului pentru comunicarea senzorului RFID RC522(slave) cu Arduino UNO(master).

Review on midterm

Modulul RC522 este un modul ce folosește tehnologia RFID(Radio Frequency Identification).

Un sistem RFID este alcătuit din două componente principale, un tag atașat de un obiect pe care vrem să îl identificăm(e.g. card), și un transceiver cunoscut și ca cititor.

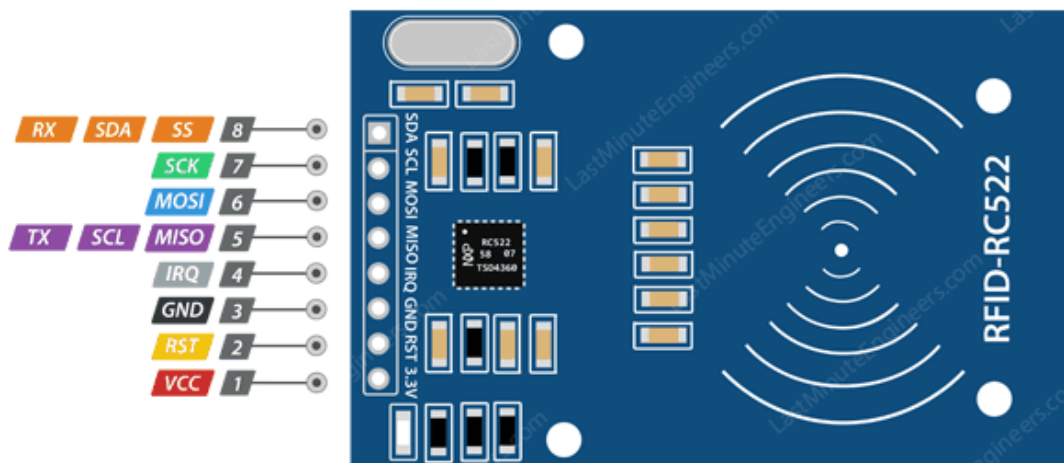
Cititorul constă într-un modul radio și o antenă ce generează un câmp electromagnetic de frecvență înaltă(RC522 are o frecvență de 13.56MHz).

Tag-ul este de obicei un dispozitiv pasiv, ce nu are alimentare. În schimb, acesta conține un microcip care reține și procesează informație și o antenă care primește și trimite semnale. Pentru a citi informația stocată pe un tag, acesta este plasat lângă un cititor. Cititorul generează un câmp electromagnetic care face ca electronii să pătrundă prin antena tag-ului și să îi alimenteze cipul acestuia. Cipul, la rândul său, odată alimentat, va răspunde cu datele stocate sub forma unui alt semnal radio ce poartă numele de backscatter.

Modulul poate comunica cu un microcontroler folosindu-se de cei 4 pini de SPI cu un data rate maxim de 10 Mbps. De asemenea, suportă și comunicare I2C și UART.

Modulul prezintă și un pin pentru întreruperi prin care modulul poate alerta microcontrolerul atunci când un tag apare în proximitatea cititorului.

Tensiune de operare: 2.5-3.3v, însă suportă și 5v



VCC – Recomandat între 2.5v și 3.3v

RST – Reset pin, activ pe '0' logic. Oprește alimentarea modulului.

GND

IRQ – Pin pentru cererile de întrerupere prin care modulul poate anunța microcontrolerul de apariția unui tag.

MISO – Master-In-Slave-Out pin pentru interfața SPI.

MOSI – Master-Out-Slave-In pin pentru interfața SPI. Reprezintă input-ul modulului.

SCK – Pin pentru clock. Acceptă impulsuri de pe bus-ul de SPI de la Arduino.

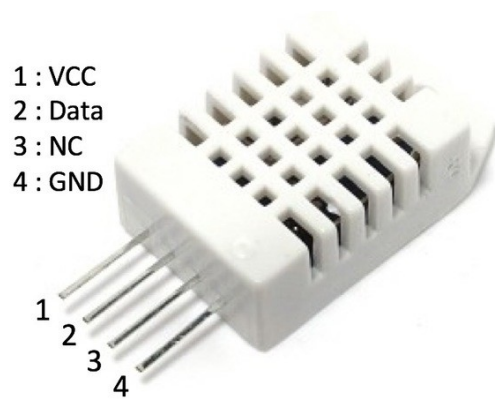
SS – Signal input atunci când SPI este activat.

Încuietoare electromagnetă.

O încuietoare solenoid este un mecanism remote de închidere a ușilor a cărui deschidere se face prin intermediul unui solenoid electromagnetic. De cele mai multe ori, mecanismul de închidere este identic cu cel al unei încuietori convenționale cu cheie. Singura diferență dintre cele două este includerea solenoidului în mecanism care trage înăuntru încuietorii zăvorul odată cu aplicarea unei tensiuni între 9-12 volți. Bobina din interiorul încuietorii este legată la un fir care, la rândul său, este legat la o sursă de tensiune. Odată cu aplicarea tensiunii de 9-12 volți, curentul va pătrunde în interiorul bobinei și va genera un câmp electromagnetic puternic ce va atrage zăvorul instant.

DHT22: The DHT22 is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed). It's fairly simple to use, but requires careful timing to grab data. Power supply pins(VDD GND) AM2302 supply voltage range 3.3V - 5.5V, recommended supply voltage is 5V. Serial data(SDA) SDA pin is tri structure for reading, writing sensor data. Specific communication timing, see the detailed description of the communication protocol.

MQ2: is one of the commonly used gas sensors in MQ sensor series. When tin dioxide (semiconductor particles) is heated in air at high temperature, oxygen is adsorbed on the surface. In clean air, donor electrons in tin dioxide are attracted toward oxygen which is adsorbed on the surface of the sensing material. This prevents electric current flow. The analog output voltage provided by the sensor changes in proportional to the concentration of smoke/ gas. The greater the gas concentration, the higher is the output voltage; while lesser gas concentration results in low output voltage. The following animation illustrates the relationship between gas concentration and output voltage.



Esp8266 → Firebase communication

What is the ESP8266?

ESP8266 is a wifi SOC (system on a chip) produced by [Espressif Systems](#) . It is an highly integrated chip designed to provide full internet connectivity in a small package.

ESP8266 can be used as an external Wifi module, using the standard AT Command set Firmware by connecting it to any microcontroller using the serial UART, or directly serve as a Wifi-enabled micro controller, by programming a new firmware using the provided SDK.

Technical Features

- 802.11 b / g / n
- Wi-Fi Direct (P2P), soft-AP
- Built-in TCP / IP protocol stack
- Built-in TR switch, balun, LNA, power amplifier and matching network
- Built-in PLL, voltage regulator and power management components
- 802.11b mode + 19.5dBm output power
- Built-in temperature sensor
- Support antenna diversity
- off leakage current is less than 10uA
- Built-in low-power 32-bit CPU: can double as an application processor
- SDIO 2.0, SPI, UART
- STBC, 1×1 MIMO, 2×1 MIMO
- A-MPDU, A-MSDU aggregation and the 0.4 Within wake
- 2ms, connect and transfer data packets
- standby power consumption of less than 1.0mW (DTIM3)

How to Communicate with ESP8266 via Arduino UNO

ESP 8266 works on 3.3v and not on 5v. So, we will try to talk to ESP8266 with the help of Arduino Uno.

Circuit Description:

Connect the Arduino's 3v3 (3.3V) output to ESP8266. The ESP8266 works with 3.3V and not 5V, so this is necessary.

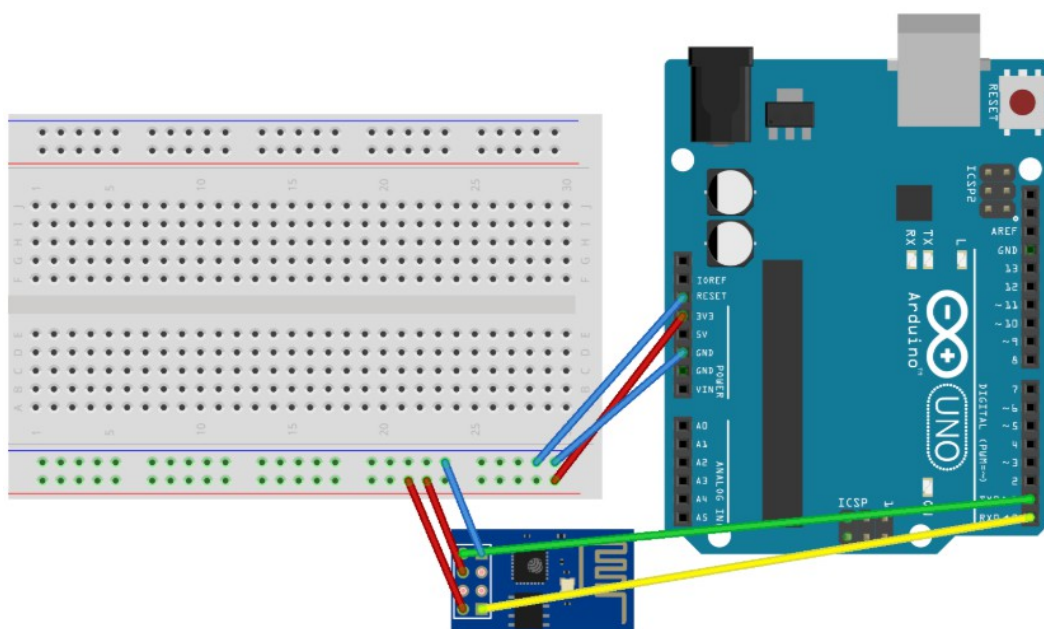
Connect the RES or RESET pin, When you ground the reset pin, the Arduino works as a dumb USB to serial connector, which is what we want to talk to the ESP8266.

Connect the RXD pin of the Arduino to the RX pin of the ESP8266.

Connect the TXD pin of the Arduino to the TX pin of the ESP. When we want two things to talk to each other over serial, we connect the TX pin of one to the RX of the other (send goes to receive and the opposite). Here we do not have the Arduino talk to the ESP8266 though, our computer is talking to it via the Arduino.

Connect GND and VCC.

Finally CH_PD connects.



Firestore Realtime Database

Firestore provides a real-time database and back-end as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firestore's cloud. The company provides client libraries that enable integration with Android, iOS, JavaScript, Java, Objective-C, Swift and Node.js applications. The database is also accessible through a REST API and bindings for several JavaScript frameworks such as AngularJS, React, Ember.js and Backbone.js. The REST API uses the Server-Sent Events protocol, which is an API for creating HTTP connections for receiving push notifications from a server. Developers using the realtime database can secure their data by using the company's server-side-enforced security rules.

Esp8266 → Firestore Communication

The main purpose of using a real-time database for our application is to communicate between the android application and hardware devices. but how is the communication done though using a real-time database?

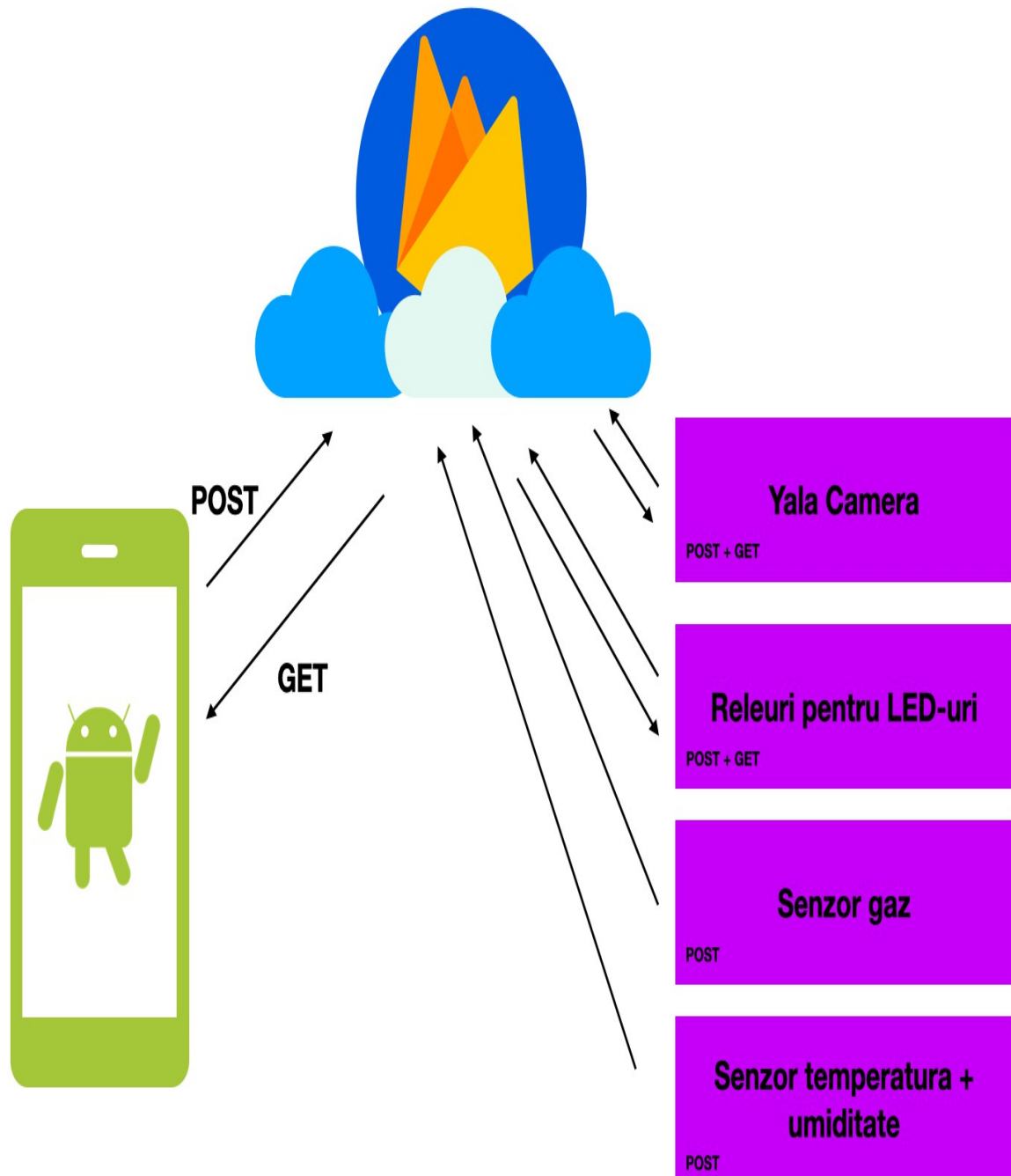
For connecting arduino with the database , we have used the Firebase Arduino Library which is a very easy to use library with a good documentation. To set it up , first we need to obtain the database authentication key and the database api url. These will be specified in the code that will then be loaded on esp8266 thus making the connection between hardware and firebase

As soon as a field in the database is updated , the applications which are connected to the database can be notified of those changes using the APIs provided by the database. In this case if A and B are two applications which is connected to C database , if A make a change to the database , B gets notified , if B makes a change to the database A gets notified. Unlike normal relational databases like MySQL , Oracle , Firebase is an NoSQL database. In here data is stored in JSON format . So it is very easy for applications to access data stored in this.

Via post and get methods, esp8266 has the ability to both read and write data to firebase.

Both the data received and transmitted by firebase and the data transmitted and received by esp866 are in JSON format. The data transmission between esp and firebase is done through the http protocol.

What's behind



Aici pot fi vizualizate mai multe detalii tehnice, precum și progresul proiectului și o scurtă descriere:

<https://github.com/luigibolovan/si-smart-room>



Bibliografie:

<https://medium.com/coinmonks/arduino-to-android-real-time-communication-for-iot-with-firebase-60df579f962>

<https://create.arduino.cc/projecthub/PatelDarshil/how-to-communicate-with-esp8266-via-arduino-uno-f6e92f>

<https://www.hackanons.com/2018/02/iot-with-firebase-connecting-arduino.html>

<https://en.wikipedia.org/wiki/Firebase>

<https://lastminuteengineers.com/how-rfid-works-rc522-arduino-tutorial/>

<https://www.wisegEEK.com/what-is-a-solenoid-door-lock.htm>

https://ro.wikipedia.org/wiki/Interfa%C8%9Ba_serial%C4%83_SPI

<https://en.wikipedia.org/wiki/ATmega328>

<https://cityos-air.readme.io/docs/4-dht22-digital-temperature-humidity-sensor>

https://wiki.seeedstudio.com/Grove-Gas_Sensor-MQ2/

<https://pythonhosted.org/pyserial/>