

UNIVERSITATEA POLITEHNICA BUCUREȘTI
FACULTATEA DE ELECTRONICĂ, TELECOMUNICAȚII ȘI TEHNOLOGIA INFORMAȚIEI
DEPARTAMENTUL INGINERIA INFORMAȚIEI

PROIECT DE DIPLOMĂ

Smart Home – Wireless Sensors for Home Automation

Robert-Valentin Ene

Coordonator științific:

Ș.l. dr. ing. Calin Bira

BUCUREȘTI

2022

UNIVERSITY POLITEHNICA OF BUCHAREST
FACULTY OF ELECTRONICS, TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY
INFORMATION ENGINEERING DEPARTMENT
DIPLOMA PROJECT

Smart Home – Wireless Sensors for Home Automation

Robert-Valentin Ene

Thesis advisor:

Prof. dr. ing. Calin Bira

BUCHAREST

2022

CUPRINS

SINOPSIS	1
ABSTRACT	1
1 INTRODUCERE	3
1.1 Context	3
1.2 Problema.....	4
1.3 Obiective.....	4
1.4 Soluția propusă	5
1.5 Structura lucrării	6
2 ANALIZA ȘI SPECIFICAREA CERINTELOR	7
3 STUDIU DE PIAȚĂ / SOLUȚII EXISTENTE	9
4 TEHNOLOGII FOLOSITE	13
4.1 Arduino IDE	13
4.2 Visual Studio Code.....	14
5 SOLUȚIA PROPUȘĂ	17
5.1 Aplicația server.....	18
5.2 Aplicația web.....	22
5.3 Soluția hardware.....	26
5.3.1 STM32 Nucleo-64 F103.....	28
5.3.2 ESP32 Lolin32.....	29
5.3.3 Senzor de temperatura și umiditate.....	30
5.3.4 Senzorul PIR (Passive Infrared Sensor)	31
5.3.5 Senzorul de gaz MQ – 2	32
5.3.6 Senzorul de lumină fotosensibil KY-018	33
5.3.7 Motorul Stepper.....	34
5.3.8 Senzorul IR transmițător și IR receptor	34
6 EVALUAREA REZULTATELOR	35
7 CONCLUZII	36
8 CONTRIBUȚII PERSONALE	37
8 BIBLIOGRAFIE	38
9 ANEXE	39

SINOPSIS

Automatizarea casei si intreaga idee despre case inteligente devine din ce in ce mai importanta si mai cautata in secolul 21. Obiectul principal al acestui proiect este acela de a dezvolta si a crea un sistem inteligent prin care putem acumula informatii de la diferiti senzori din casa pe telefon si de a controla functionalitati ale casei noastre direct de pe smartphone-ul nostru.

Tehnologiile folosite in acest proiect sunt protocolul de comunicatie wireless - Wi-Fi; ESP-NOW care este un protocol de transmisie wireless creat de Espressif intre mai multe placi de dezvoltare ESP32; diferite protocoale de transmisie seriala - uart, i2c, spi; baze de date pentru acumularea datelor si protectia acestora; o parte de front-end in care vom avea interfata utilizatorului.

Aplicatia propusa permite controlul facil al utilitatilor din casa: stingerea/aprinderea becurilor, pornirea/oprirea A/C, pornirea/oprirea TV. De asemenea nu se folosesc telecomandele aferente fiecarui device, ci sistemul este capabil sa invete si sa reproduca semnalele IR de la telecomanda fizica, si sa foloseasca in schimb butoane de pe aplicatia web, pe telefon. Toate acestea fiind implementate in reseaua proprie, folosindu-ne de ssid-ul router-ului Wi-Fi si de parola retelei pentru a crea un sistem inteligent, fara fire si retele fizice complexe.

ABSTRACT

Home automation and the whole idea of smart homes is becoming more and more important and wondered in the 21st century. The main goal of this project is to create and develop a smart system through which we can receive data from different sensors on our phone and control the functionality of our home directly from our smartphone.

The technologies used in this project are the wireless communication protocol - Wi-Fi; ESP-NOW which is a wireless transmission protocol created by Espressif between several ESP32 development boards; different serial transmission protocols - uart, i2c, spi; databases for data accumulation and protection; a front-end part where we will have the user interface.

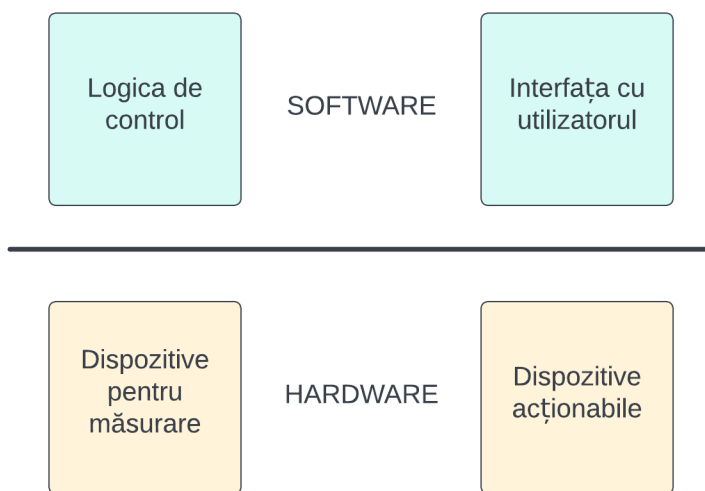
The proposed application allows easy control of household utilities: turning on / off light bulbs, turning on / off AC, turning on / off TV. Also, the remote controls for each device are not used because the system is able to learn and reproduce the IR signals from the physical remote control, and to use the buttons on the web application on the phone instead. All this things are implemented in our own network, using the ssid of the Wi-Fi router and the network password to create an intelligent system, without wires and complex physical networks.

1 INTRODUCERE

Oamenii, de multe ori, au tendința de a face lucrurile mai ușoare, și nu este un lucru de condamnat. Vedem cum lumea din zilele noastre se îndreaptă din ce în ce mai mult spre tehnologie, și acesta este un lucru extraordinar. Dacă tehnologia se dezvoltă, este bine să o aplicăm și să o folosim în mod direct în viețile noastre. Oamenii sunt destul de ocupați la locul de muncă încât nu mai doresc să aibă stresul provocat de anumite neajunsuri în propria casă. Din aceste motive (și din altele pe care le voi prezenta în continuare), aplicațiile pentru case inteligente au o importanță deosebită, acestea influențând în mod direct starea și emoțiile ființei umane.

1.1 Context

Proiectul propus are în vedere crearea unui sistem prin care putem controla propria casă doar din telefonul mobil, fiind mai ușor și mai la îndemână. Automatizarea casei este de multe ori numită pur și simplu „casă inteligentă”. Pentru realizarea acestui proiect se folosește tehnologie IoT. Aceste sisteme IoT sunt formate atât din componente hardware cât și din componente software.



Figură 1 Componente IoT

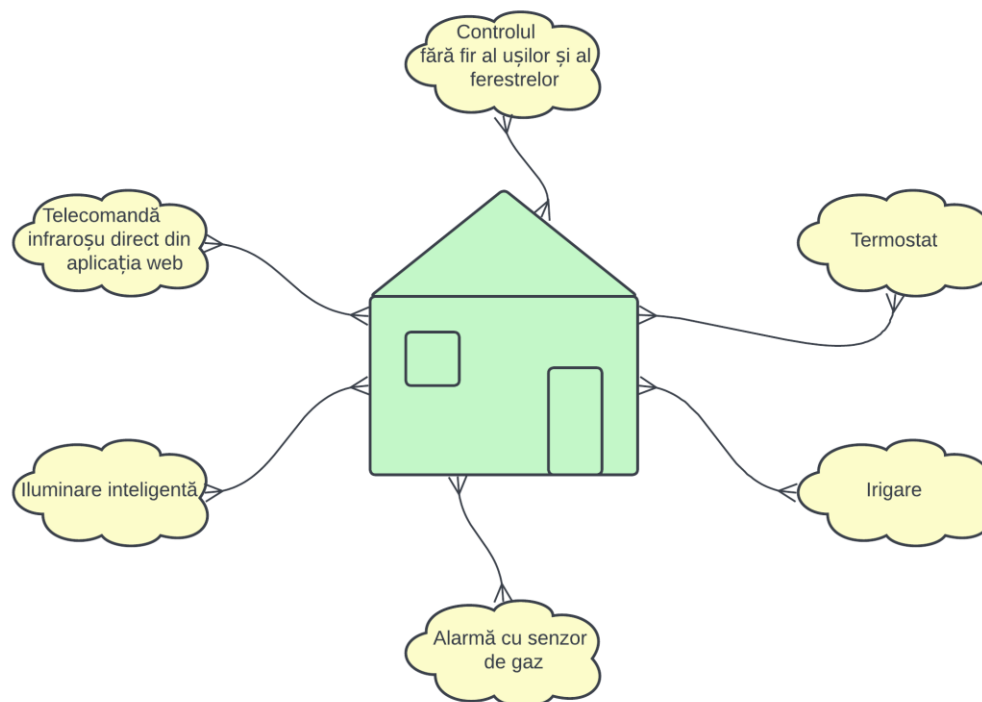
1.2 Problema

În România încă nu este atât de extins acest domeniu de case inteligente deoarece soluțiile apărute pe piață sunt destul de scumpe și anevoioase, mai ales dacă locuința este deja construită. Principalele probleme sunt lipsa banilor, fără de care nu prea poți construi un ecosistem prietenos prin care să introduci funcționalități inteligente la propria casă. Sistemele apărute pe piață, sunt într-adevăr inteligente, dar numai un Hub inteligent controlul luminilor costă destul de mult.

O altă problema este că atunci când vrei să implementezi un sistem inteligent pentru casa ta, trebuie să introduci tot felul de cabluri. Probabil este nevoie de canale de cabluri prin pereți și este greu să iei această decizie gândindu-te la implicații.

1.3 Obiective

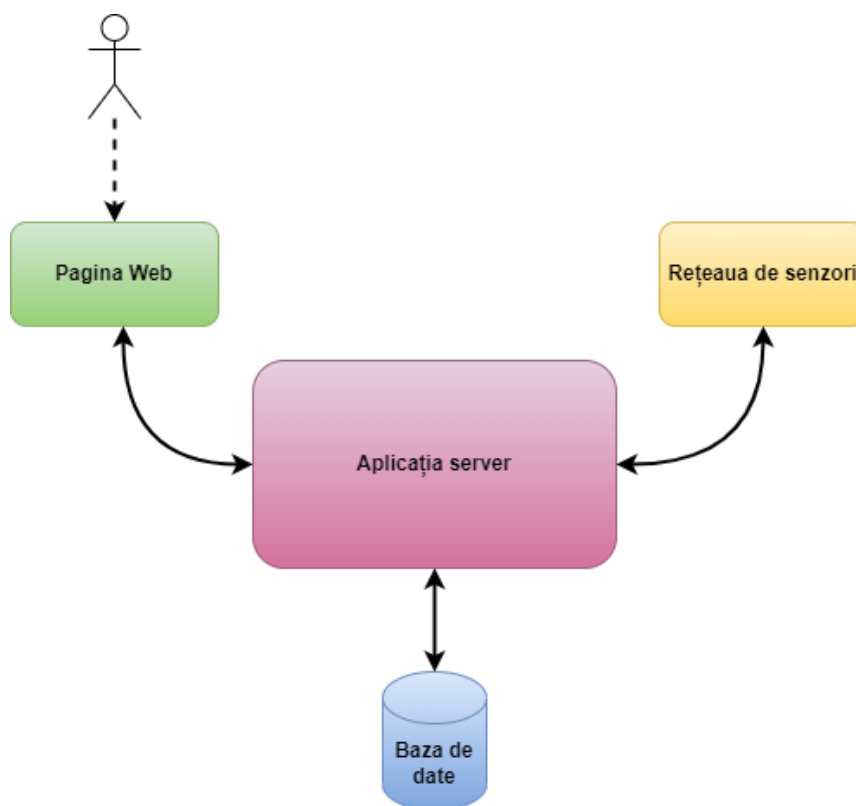
Obiectivul principal al acestui proiect este oferirea unei soluții complete pentru a fi disponibilă crearea unei case inteligente, cu nivel de finanțe redus, care să îndeplinească cerințele unui sistem automatizat și inteligent. Pornind de la configurarea unei camere cu diferiți senzori care transmit starea casei aproape în fiecare moment al zilei, această soluție se poate răspândi în întreaga casă. Implementarea proiectului și proiectarea aplicației își propune a fi ușor de accesat și folosit în viața de zi cu zi de către utilizatori, astfel devenind o soluție de luat în calcul când vine vorba de automatizarea unor funcționalități într-o casă inteligentă.



Figură 2 Utilități în casa inteligentă

1.4 Soluția propusă

Pentru a atinge obiectivele acestui proiect, soluția propusă este alcătuită din 3 componente: o aplicație web care are în vedere informarea utilizatorilor cu privire la starea casei și în același timp controlul diferitelor funcții pe care casa inteligentă le pune la dispoziție, o componentă hardware pentru măsurarea indicatorilor cu ajutorul senzorilor și o aplicație server pentru colectarea și gestionarea datelor.



Figură 3 Arhitectura soluției propuse

Prototipul hardware este un dispozitiv propus pentru măsurarea diferitelor informații și date provenite de la senzori. Acesta va fi construit dintr-o stație de bază – o plăcuță de dezvoltare Nucleo-64, senzori pentru măsurarea luminozității din încăpere (KY-018), măsurarea temperaturii și a umidității (ambele putând fi preluate de la un singur senzor – DHT11), măsurarea nivelului de gaz din încăpere (MQ-2).

Pe lângă senzorii care transmit date despre starea casei, avem și componente hardware care ajută la automatizarea locuinței – senzor PIR pentru facilitarea aprinderii luminii la detectarea prezenței, senzori IR care detectează prezența unor unde infraroșu, iar sistemul poate învăța coduri de la telecomanda TV/AC și poate transmite mai departe acele coduri direct din aplicația Web, motoare stepper care ajută la închiderea/deschiderea ușilor/ferestrelor în cazul în care utilizatorii pleacă de acasă, sau se detectează o cantitate semnificativă de gaz sau de fum.

Aplicația server, detaliată în capitolul 5 – Soluția propusă, este responsabilă pentru colectarea datelor. Aceasta primește în mod constant informații utile de la senzori și le publică la end-point-ul programat. Aplicația server interoghează prototipul hardware, acesta efectuând măsurări la intervale de timp bine stabilite, iar fiecare senzor ajunge să trimită datele către server.

Aplicația web este destinată oricărui utilizator, având în vedere un public larg. Aceasta îndeplinește mai multe roluri. Un prim rol este cel de informare: utilizatorii se pot documenta, într-o pagină dedicată, cu privire la starea locuinței, temperatura și umiditatea din încăpere și nivelul de gaz sau fum și impactul pe care acesta l-ar putea avea asupra sănătății acestora. Există o altă pagină dedicată în care utilizatorul poate urmări grafice cu valorile senzorilor în ultimele 2h, astfel putând să aibă o privire de ansamblu a istoricului casei.

1.5 Structura lucrării

În continuare vom aprofunda motivațiile acestui proiect în capitolul 2: vom răspunde la întrebări precum de ce este nevoie de o asemenea aplicație, ce valoare aduce utilizatorului aplicația, care este impactul pe care o astfel de implementare o are în viața de zi cu zi și de ce soluțiile existente nu sunt suficiente pentru împlinirea scopului acestui proiect.

În capitolul 3 vom analiza soluțiile deja existente pe piață în acest sens, care sunt avantajele și dezavantajele acestora, cu ce diferă soluțiile deja existente cu sistemul propus în această lucrare, apoi în capitolul 4 vom prezenta ce tehnologii au fost folosite în realizarea acestui proiect, urmând ca în capitolul 5 să fie prezentată soluția propusă pentru a acoperi neajunsurile soluțiilor existente prezentate în urmă cu 2 capitole. Vom vedea care este arhitectura soluției propuse, cum intercomunică componentele ce alcătuiesc întreg sistemul propus, cum sunt construite și ce rol îndeplinesc aceste componente și, în final, ce aduce în plus această soluție față de cele prezentate în capitolul 3.

În capitolul 6 vom evalua aplicația din mai multe perspective și vom vedea dacă aceasta și-a atins scopul și poate deservi utilizatorii în sensul anunțat anterior. La final, în urma acestor evaluări vom trasa concluziile acestui proiect (în capitolul 7).

2 ANALIZA ȘI SPECIFICAREA CERINȚELOR

Dezvoltarea prezentului proiect este motivată din mai multe direcții: din punct de vedere informatic, al integrării tehnologiei în viața de zi cu zi cu scopul de a îmbunătăți viața și de a ne-o face mai ușoară, de a transforma o casă obișnuită într-una inteligentă, cu diferite funcționalități controlabile din propriul telefon inteligent și din punct de vedere al finanțelor de nivel ridicat necesare pentru implementarea acestui gen de proiecte. În al doilea rând, o problemă este și cea medicală și psihologică din punct de vedere al stresului la care utilizatorul este supus atunci când nu este în controlul propriei locuințe. În continuare vom dezvolta pe rând aceste motivații.

Motivația din punct de vedere informatic este foarte importantă deoarece omul merită să fie la curent cu tot ce se întâmplă în casă. Tehnologia ne ușurează viața și în același timp o îmbunătățește. Având la dispoziție aplicația de informare cu privire la senzorii plasați în interiorul casei, utilizatorii vor fi mult mai liniștiți cu privire la ce se întâmplă în interiorul locuinței, stresul la nivel medical se va micșora, iar oamenii nu vor mai pune atât de mult accent pe panica creată de faptul că propria locuință nu este în siguranță.

Transformarea unei case obișnuite într-una inteligentă este necesară deoarece tehnologia evoluează pe zi ce trece și este păcat să nu o folosim pentru binele propriei ființe. Tehnologia ne-a ușurat și îmbunătățit viața și nivelul de trai și va continua să o facă, și aceasta ne este la îndemână. Deja folosim laptopuri, telefoane inteligente, mașini digitalizate, electrocasnice inteligente. Toate acestea ne-au impactat viața personală într-un fel sau altul. Este momentul să integrăm și la nivel de locuință această inteligență. Funcționalitățile pe care le poate oferi o astfel de aplicație sunt simple, dar pot aduce traiul de zi cu zi al vieții la un alt nivel, având siguranța ca totul este sub control.

Proiectul prezent propune o soluție pentru a îmbunătăți viața colectivă, în afara spațiului personal al casei, dar nu doar atât. Această soluție permite atât controlul luminilor și diferitelor funcții ale casei din telefon, cât și vizualizarea în timp real a anumitor parametri importanți: temperatura, umiditatea, nivelul de gaz din încăpere, nivelul de luminozitate. Având la dispoziție o aplicație în care informația este doar la un clic distanță, utilizatorul își poate pune încrederea în propria casă și poate pleca la drum fără a mai avea frică pentru ce se poate întâmpla în lipsa acestuia.

Din punct de vedere medical, stresul este unul din factorii care pot duce la îmbolnăvirea pacienților foarte grav. Acesta este un fenomen psihosocial complex care decurge din confruntarea persoanei la nivel psihologic cu diferite sarcini și situații pe care creierul uman le percepe ca fiind dificile, greu de rezolvat, provocând gândire excesivă asupra unor aspecte care, inițial nu par foarte grave, dar ajung să necesite atât de multă energie și efort depus.

3 STUDIUL DE PIAȚĂ / SOLUȚII EXISTENTE

Oamenii interacționează cu mediul din jurul lor prin diferite moduri. Dacă mediul cu care aceștia interacționează este unul prietenos și care răspunde înapoi cu informații, aceasta poate aduce un mare avantaj. O aplicație pentru o casă inteligentă introduce această comunicare a utilizatorilor cu propria locuință, și face din aceasta o proprietate cu un mediu prietenos, dând posibilitatea de control total și informare continuă.

Una din abordările uzuale cu privire la crearea unui mediu inteligent pentru locuințe este automatizarea funcționalităților prezente deja în interiorul casei. Pot fi utilizați anumiți senzori (senzori PIR) pentru a se reduce costurile privitoare la consumul de curent, sau senzori care să transmită informații utile proprietarilor (senzori de gaz, senzori de temperatura). În același timp, este nevoie de microcontrolere care facilitează comunicația senzorilor cu aplicația pentru casa inteligentă.

Comunicațiile între senzori și microcontroler sau microcontroler și server/aplicație se pot realiza prin mai multe protocoale prezente deja pe piață. Aceste protocoale au diferite avantaje și dezavantaje; unele pot fi comunicații seriale, care necesită prezența unui canal fizic de comunicație (cablu, fire), iar altele pot fi comunicații wireless (fără fir) și aici se folosesc, din nou, mai multe protocoale și tipuri de comunicație. În continuare vom prezenta diverse protocoale și tipuri de comunicație, și vom pune în lumină avantaje și dezavantaje.

- Ethernet – Protocol de comunicație standard, care utilizează transmiterea datelor prin cablu, permite transmiterea unor cantități mari de date și acestea pot fi transmise cu ușurință și destul de rapid. Tehnologia Ethernet permite pachetelor de date să fie transmise cu ajutorul cablurilor fizice, create din perechi de fire răsucite. Vitezele de transmisie atinse cu acest protocol sunt de 1Gbps la cablurile de tip CAT4, iar la cele de tip CAT6 se poate ajunge la viteze de 10Gbps.
- Wi-Fi – Protocol de comunicație standard, care utilizează transmiterea datelor prin aer, permite interoperabilitatea (abilitatea sistemelor sau proceselor de a lucra împreună pentru realizarea unui scop comun) și comunicarea facilă între mai multe dispozitive diferite. Pachetele transmise prin protocolul de comunicație Wi-Fi circulă în undă radio în spectrul de frecvențe de 2.4GHz sau 5 GHz. Vitezele atinse de acest protocol de transmisie de date sunt cuprinse între 10Mbps și 100Mbps. De asemenea, protocolul Wi-Fi este capabil să transmită elemente audio și video la calitate superioară.
- Zigbee – Protocol de comunicație wireless, dezvoltat pentru a face posibilă comunicația cu o viteză relativ redusă, dar, în același timp, folosind o bandă de frecvență redusă. Acest protocol de transmisie al datelor se folosește pentru comunicații pe distanțe reduse. Undele radio transmise prin acest protocol operează la frecvența de 2.4GHz. Vitezele de transmisie sunt de 250kbps și de obicei se transmit și se primesc diferite comenzi deoarece frecvența folosită este robustă.

- Bluetooth – Protocol de comunicație wireless, ce face posibilă comunicația între diferite dispozitive, pe o distanță redusă. Majoritatea dispozitivelor (în special telefoanele inteligente) au la dispoziție și includ acest protocol în arhitectură, ceea ce face din acest protocol unul facil când vine vorba de comunicarea între dispozitive într-o aplicație creată pentru o casă inteligentă. Și acest protocol folosește banda de frecvență de 2.4GHz. Protocolul funcționează pe o setare de Master/Slave. Variantele Bluetooth 3.0 și Bluetooth 4.0 permit transmiterea datelor la o viteză aproximativă de 24Mbps.

În continuare, voi prezenta avantajele și dezavantajele pe care le oferă fiecare din protocoalele de comunicație prezentate succint mai sus.

Protocol	Avantaje	Dezavantaje
Ethernet	<ul style="list-style-type: none"> - viteza de transmisie a datelor - distanța facilă pe care se poate face transmisia - securitatea de care dispune rețeaua fizica, pe cablu 	<ul style="list-style-type: none"> - costul ridicat necesar instalării rețelei de cabluri - necesitatea unor switch-uri - planificarea corectă și din timp a casei pentru a introduce rețeaua de cabluri prin perete, pentru a nu fi la vedere
Wi-Fi	<ul style="list-style-type: none"> - viteza de transmisie a datelor - raza de acoperire - tehnologia și informația este la îndemână - un router wireless este mult mai ieftin decât crearea unei întregi rețele pe cablu care conține switch-uri 	<ul style="list-style-type: none"> - competiția dispozitivelor este foarte mare în aceeași bandă de frecvență: telefoane, tablete, ceasuri inteligente, termostate - consumul de putere este mare la dispozitivele Wi-Fi - la nivel de securitate este necesară o setare corectă și atentă
Zigbee	<ul style="list-style-type: none"> - rețea flexibilă - necesită consum de putere mic - dispozitivele care folosesc Zigbee sunt pe baterii, ceea ce face ca rețeaua Zigbee să fie redusă din punct de vedere arhitectural și fizic ca dimensiune 	<ul style="list-style-type: none"> - limitat din punct de vedere al accesoriilor disponibile pe piață - nu este atât de popular - nu este sigur din punct de vedere al securității - transmiterea datelor cu viteză mică
Bluetooth	<ul style="list-style-type: none"> - metodă simplă de împerechere a dispozitivelor - utilizează comunicație care are un consum redus de putere - accesorii reduse din punct de vedere arhitectural și fizic ca dimensiune, majoritatea dispozitivelor utilizând baterii - implementarea este facilă din punct de vedere al costurilor materiale 	<ul style="list-style-type: none"> - utilizează banda de frecvență destul de ocupată: 2.4GHz - comunicațiile se fac pe o distanță redusă - poate pierde conexiunea destul de rapid în condiții precare

Tabel 1 Avantaje și dezavantaje protocoale de comunicație

După ce am studiat ce protocoale sunt pe piață, ce se poate folosi într-o aplicație de casă inteligentă, se poate face și o comparație cu ce soluții există pentru o anumită funcționalitate, ce avantaje și dezavantaje are, ce tipuri de comunicație oferă, durata de viață etc.

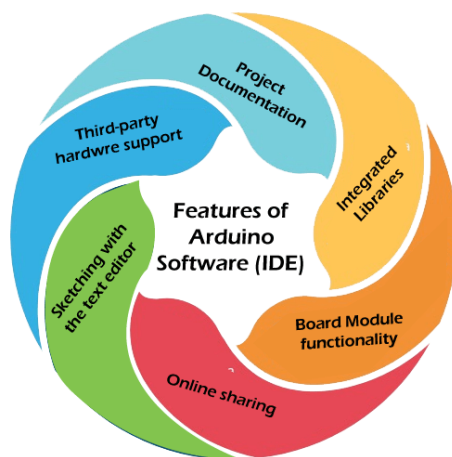
Parameter		Smart lighting	
Product	LIFX Mini White	LIFX A19+	Philips Hue White and Color + Bridge
Connectivity	Wi-Fi (no HUB needed), 802.11 b/g/n standards compliant	Wi-Fi (no HUB needed), 802.11 b/g/n standards compliant	Wi-Fi (requires bridge), Zigbee
Smart home integration	Home Kit, Amazon Alexa, Google Assistant	Home Kit, Amazon Alexa, Google Assistant	IFTT, Logitech, Amazon Alexa, Home Kit, Google Home and Assistant
Voice activation	Yes	Yes	Yes
Power Consumption	60W-equivalent, consumes 8W power	60W-equivalent, consumes 11W power	60W-equivalent, consumes 10W power
Life-span	22.8 years	22.8 years	20.000 hours ~ 2.3 years
Cost	\$34/Bulb	\$120	\$150

Tabel 2 Comparație soluții iluminare inteligentă

4 TEHNOLOGII FOLOSITE

4.1 Arduino IDE

Arduino IDE (Integrated Development Environment) este componenta software oficială introdusă de Arudino.cc și este folosită în principal pentru editarea, compilarea și încărcarea codului pe un dispozitiv Arduino. Majoritatea componentelor hardware Arduino sunt compatibile cu acest IDE, acesta având mai multe beneficii: software open-source, bine documentat și ușor de folosit și de înțeles.



Imagine 1 Funcționalități Arduino IDE

Codul în Arduino este scris în limbajul C++, dar pe lângă acesta, are biblioteci, metode și funcții dedicate. C++ este un limbaj de programare citibil de către om, iar atunci când sketch-ul (fișierele scrise în Arduino IDE) este creat, codul este compilat și procesat în cod mașină.

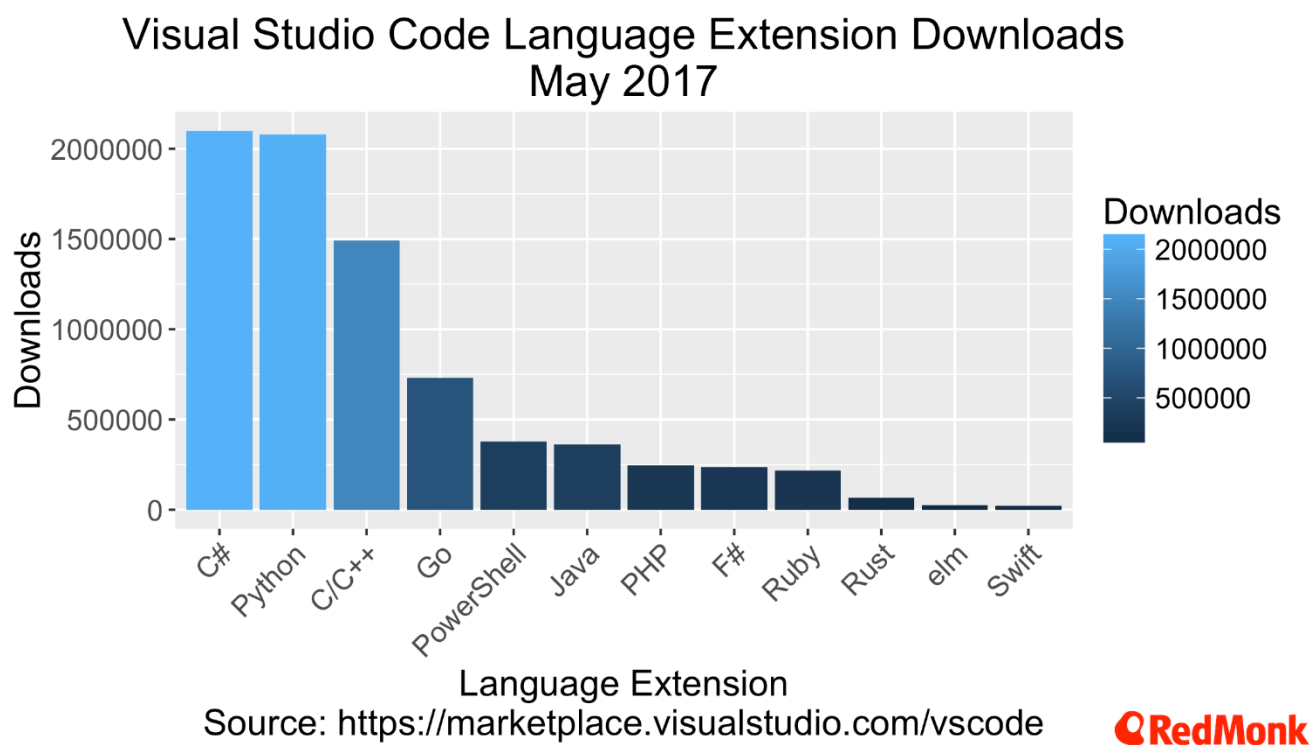
Structura de bază a fiecărui cod scris în Arduino IDE are 4 părți principale:

- Includerea bibliotecilor necesare parcurgerii codului fără posibile erori
- Definirea pinilor folosiți de partea hardware, declararea variabilelor și crearea obiectelor care sunt instanțe ale unor clase create anterior
- Apelarea funcției `setup()` – aceasta definește starea inițială a plăcii Arduino și se execută doar o singură dată, la inițializarea sistemului. În această funcție se definesc funcționalitățile pinilor declarați anterior utilizând funcția `pinMode()`, starea inițială a pinilor și inițializarea variabilelor
- Apelarea funcției `loop()` – aceasta este obligatorie în orice sketch Arduino (ca și funcția `Setup()`), și se execută imediat după ce se termină de executat funcția `Setup()`. Funcția `Loop()` este funcția main(principală) și, așa cum indică și numele acesteia, se execută continuu la nesfârșit.

Arduino este o platformă foarte versatilă. Conține suport pentru majoritatea plăcilor de dezvoltare (acestea trebuie instalate din Arduino IDE), o varietate mare de biblioteci ce fac viața dezvoltatorilor mai ușoară – acestea conținând multe funcții, metode și clase care ajută utilizatorii software-ului să ajungă rapid la rezultatul dorit. [4]

4.2 Visual Studio Code

Visual Studio Code, denumit și VS Code, este un editor de cod sursă realizat de Microsoft pentru Windows, Linux și macOS. Caracteristicile includ suport pentru depanare, evidențierea sintaxei, completarea inteligentă a codului, fragmente, refactorizarea codului și Git încorporat. [5]



Imagine 2 Limbaje folosite și dezvoltate în Visual Studio Code

În Visual Studio Code putem scrie mai multe limbaje de programare precum C/C++, JavaScript, Python și multe altele. Acesta are o interfață prietenoasă, ușor de înțeles și se dezvoltă în continuu din 2015 până în prezent. La acest proiect au participat mai mulți dezvoltatori dar principalul om care a lucrat a fost Erich Gamma, acesta creând și primul commit pentru acest proiect.

În centrul său, Visual Studio Code are un editor de cod sursă foarte rapid, perfect pentru utilizarea de zi cu zi. VS Code încurajează productivitatea cu evidențierea sintaxelor, potrivirea parantezelor, indentarea automată și multe altele. Comenzile rapide intuitive de la tastatură, personalizarea ușoară și mapările de comenzi rapide de la tastatură permit navigarea cu ușurință în cod.

Pentru o codificare serioasă, de asemenea veți beneficia de instrumente cu mai multă înțelegere a codului decât blocuri de text. Visual Studio Code include suport încorporat pentru completarea codului IntelliSense, înțelegerea și navigarea bogată a codului semantic și refactorizarea codului.

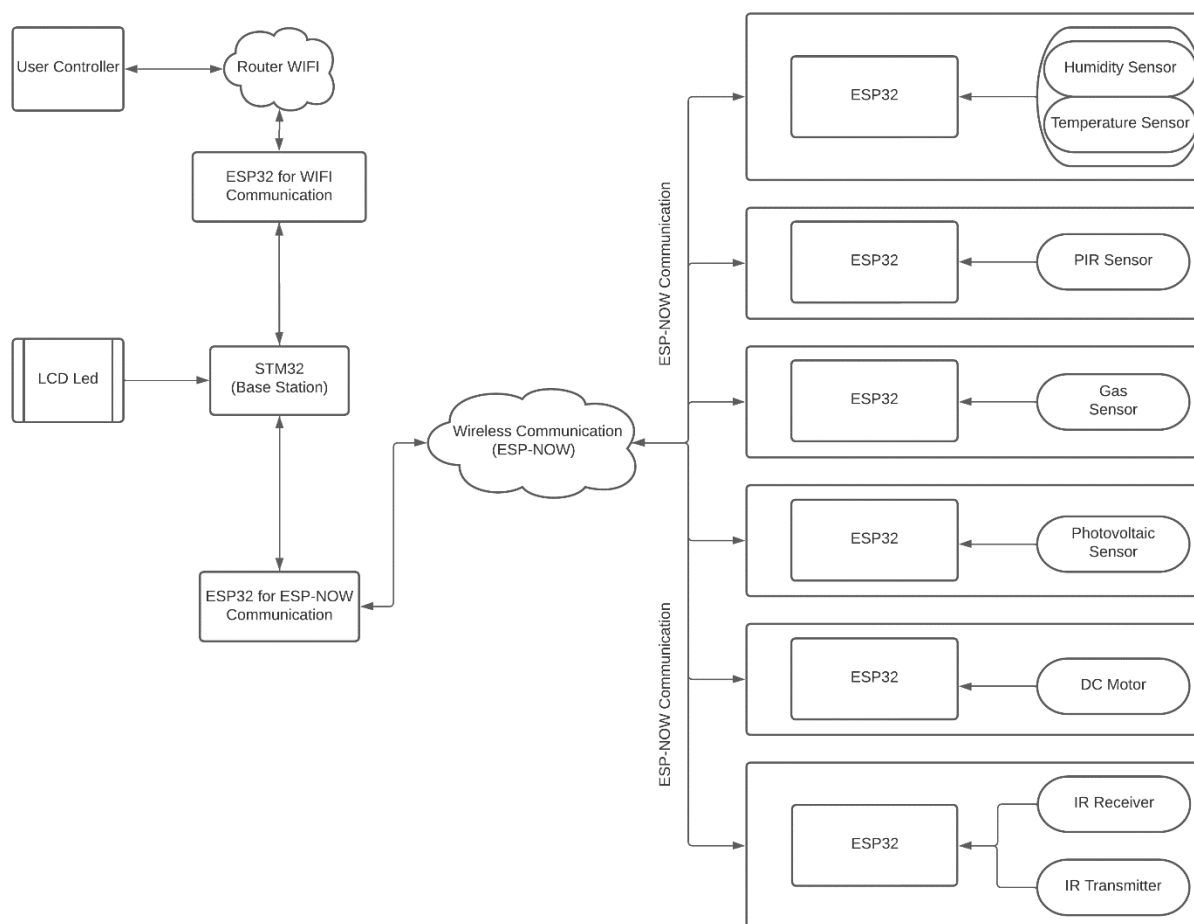
Acest software include un depanator interactiv, astfel încât utilizatorii să poată parcurge codul sursă, să inspecteze variabilele, să vizualizeze stivele de apeluri și să execute comenzi în consolă.

De asemenea, VS Code se integrează cu instrumente de creare și scriptare pentru a efectua sarcini obișnuite, făcând fluxurile de lucru de zi cu zi mai rapide. VS Code are suport pentru Git, astfel încât să puteți lucra cu controlul sursei fără a părăsi editorul, inclusiv vizualizarea diferențelor de modificări în așteptare. [6]

5 SOLUȚIA PROPUȘĂ

Soluția pe care o propunem este formată din 3 componente principale: aplicația server, aplicația web și o rețea hardware compusă din senzori care este distribuită în fiecare încăpere din locuință. Rețeaua hardware este alcătuită din componente individuale asemenea prototipului propus în continuare. Fiecare dintre aceste componente vor fi analizate, pe rând, în capitolul curent.

Schema bloc a soluției propuse este descrisă în următoarea figură:



Figură 4 Schema bloc a sistemului propus

Aplicația server va avea ca furnizor de date stația principală (microcontroller STM32) care va trimite informațiile printr-o comunicație serială (UART). Aceste date se transmit serverului la intervale de timp bine stabilite, iar acesta din urmă stochează datele într-o bază de date timp de X ore, acestea fiind accesibile utilizatorului prin clientul web.

Interfața cu utilizatorul va fi simplă, ușor de folosit și va beneficia de funcționalități precum prezentarea sub forma unor grafice a nivelului de gaz din încăperea, nivelului temperaturii și a umidității sau oferirea posibilității utilizatorului de a controla unități de iluminare din locuință.

5.1 Aplicația server

Aplicația server este construită cu ajutorul NodeJS. Node.js este un mediu de execuție JavaScript și este adoptat masiv în aplicațiile IoT (Internet of Things), deoarece JavaScript este un limbaj de programare rapid, înțeles de majoritatea dezvoltatorilor web și este potrivit pentru crearea unui server pentru o aplicație embedded. [1]

Avantajele framework-ului NodeJS:

- Acesta ajută un dezvoltator web să se concentreze mai mult pe front-end-ul aplicației decât pe partea script-ului de pe server – server-ul fiind destul de ușor de implementat, fiind ajutați și de framework-ul express.js
- Toate layer-urile și funcțiile conținute de acest framework folosesc același limbaj de programare, JavaScript
- Deoarece serverul de baze de date este implementat pe sistemul dezvoltatorilor, el are control deplin asupra confidențialității și securității acestuia. [2]

Express este cel mai popular framework Node, acesta fiind un cadru pentru aplicații back-end pentru Node.js. Cu ajutorul acestuia se poate crea un server foarte simplu în 3 etape:

- Importarea modulului express pentru crearea unei aplicații de tip Express. Obiectul app conține metode care pot fi apelate pentru crearea unor cereri HTTP (HTTP requests).

```
const express = require('express');
```

```
const app = express();
```

- Definirea rutei – se folosește metoda app.get() alături de 2 parametri: ruta care este pasată ca un string și o funcție de callback care este invocată mereu când se face un HTTP GET request pe ruta selectata

```
app.get('/temperature-sensor', (req, res) => {  
  res.send(dataFromSensor.get("0"));  
});
```

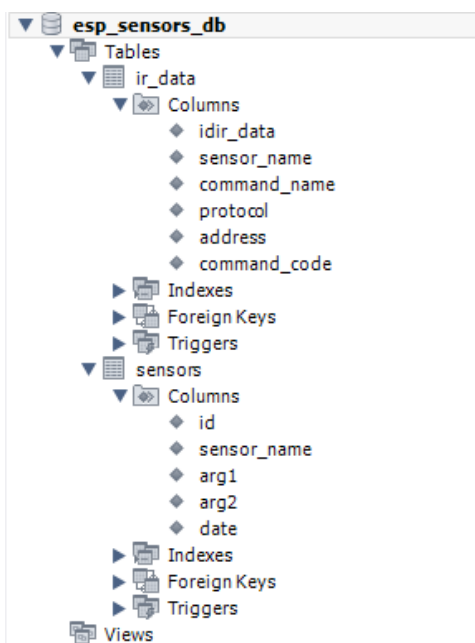
- Pornirea server-ului pe un port specificat de dezvoltator și afișarea datelor din funcția anterioara de callback, dacă este cazul

```
app.listen(3000, () =>  
  console.log('App listening on port 3000!'),  
);
```

Pentru a face posibilă comunicația cu stația principală a componentei hardware (STM32) am apelat la biblioteca serialport cu ajutorul căreia am creat un obiect și i-am pasat parametrii necesari pentru a prelua datele de pe port-ul serial al laptopului.

```
var port = new SerialPort("COM10", {  
    baudRate: 115200,  
    dataBits: 8,  
    parity: "none",  
    stopBits: 1,  
    flowControl: false,  
});
```

De asemenea, aplicația server comunică cu o bază de date creată în MySQL folosind instrumentul de proiectare al bazelor de date MySQL Workbench 8.0. Această bază de date conține 2 tabele pentru senzori. Primul tabel este folosit pentru stocarea informațiilor provenite de la senzorii de măsurare – temperatura, umiditate, nivel de gaz în încăpere, luminozitatea în cameră. În această tabelă sunt stocate informațiile senzorilor precum: id-ul intrării în baza de date, numele sensorului, câmpurile pe care acesta le măsoară și nu în ultimul rând data și ora la care intrarea în baza de date a fost făcută. Al doilea tabel este folosit pentru a stoca informațiile și comenzile senzorilor IR folosiți în aplicație. Utilizatorul va putea stoca în baza de date comenzile IR ale telecomenzii proprii iar apoi va putea folosi interfața creată (pagina web) pentru a interacționa direct cu dispozitivele IR de pe telefon, fără a mai avea nevoie de telecomanda clasică.



Imagine 3 Baza de date cu tabelele aferente

În aplicația server avem și câteva funcționalități ce pot controla baza de date fără a fi nevoie de intervenția utilizatorului direct din MySQL Workbench 8.0. În interiorul server-ului avem câteva rute definite ce pot crea tabele noi, pot insera rânduri noi în aceste tabele, pot șterge elemente sau chiar pot crea o nouă bază de date de la zero.

- Crearea unei noi baze de date:

```
app.get('/createdb', (req, res) => {  
  let sql = 'CREATE DATABASE esp_sensors_db'  
  db.query(sql, err => {  
    if(err) {  
      throw err;  
    }  
    res.send('Database Created');  
  })  
})
```

- Crearea unui tabel nou în baza de date:

```
app.get('/create-sensors-table', (req, res) => {  
  let sql = 'CREATE TABLE sensors(id int AUTO_INCREMENT, sensor_name  
  VARCHAR(255), arg1 VARCHAR(255), arg2 VARCHAR(255), PRIMARY KEY(id))'  
  db.query(sql, err => {  
    if(err) {  
      throw err;  
    }  
    res.send('Sensors table Created');  
  })  
})
```

- Inserarea unui rând nou în baza de date:

```
app.get('/insert-sensor', (req, res) => {
  var today = moment().toDate()
  today = moment(today).format('YYYY-MM-DD HH:mm:ss')

  let post = {sensor_name: `temp`, arg1: '25', arg2: ``, date:
`${today}`};

  let sql = 'INSERT INTO sensors SET ?'
  let query = db.query(sql, post, err => {
    if(err) {
      throw err
    }
    res.send('Sensor added');
  })
})
```

- Ștergerea unui rând din baza de date:

```
app.get('/delete-sensor/:id', (req, res) => {
  let newName = 'Updated name';
  let sql = `DELETE FROM sensors WHERE id = ${req.params.id}`
  let query = db.query(sql, err => {
    if(err) {
      throw err
    }
    res.send('Sensor deleted');
  })
})
```

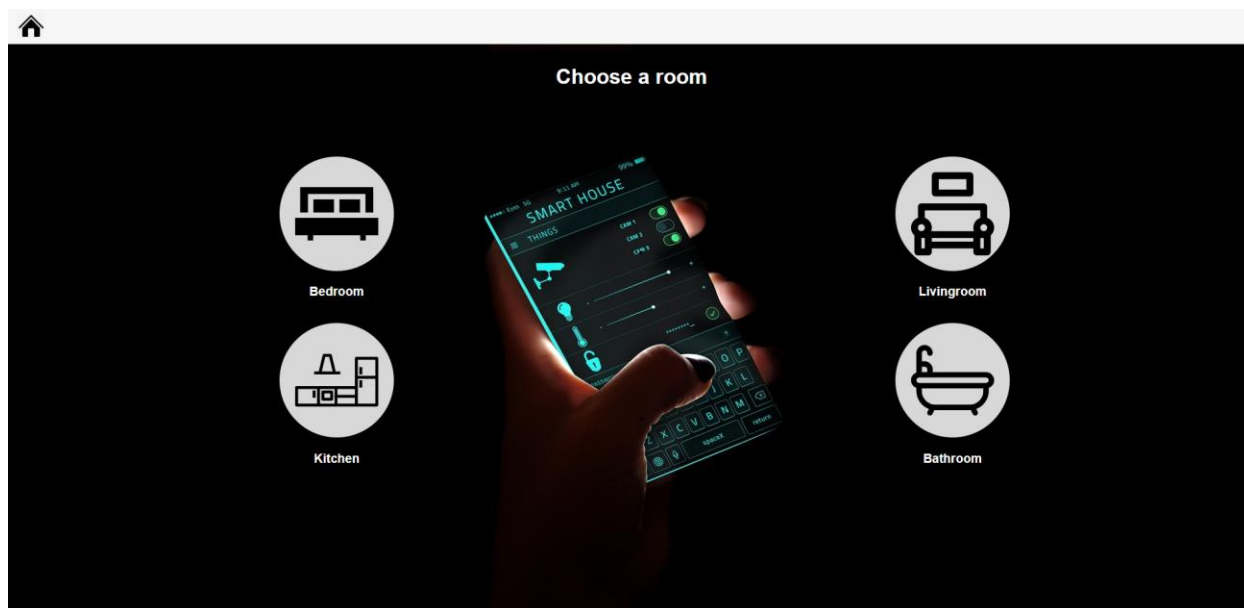

5.2 Aplicația web

Interacțiunea cu utilizatorul prezintă un obiectiv important al acestui obiect. Așa cum am amintit în primul capitol, soluția propusă este compusă din cele 3 componente: aplicația server, aplicația web și componenta hardware. În continuare vom prezenta a doua componentă din cele 3 menționate anterior și anume aplicația web. Utilitatea acestei aplicații constă în faptul că utilizatorii vor putea afla informații despre casă și vor putea controla mai multe dispozitive din casa inteligentă alcătuită din întregul sistem.

Tehnologiile folosite pentru dezvoltarea aplicației sunt: Javascript, HTML, CSS. Cele 3 tehnologii lucrează împreună pentru a aduce valoare aplicației finale. HTML este folosit pentru a introduce conținut în pagina web, în timp ce CSS ajută la stilizarea conținutului, făcând aplicația să fie plăcută ochiului uman și, de asemenea, intuitivă la folosire. În timp ce primele 2 tehnologii sunt statice, pe urmă intervine Javascript care aduce un plus aplicației prin faptul că dezvoltă interfața și pe parte dinamică.

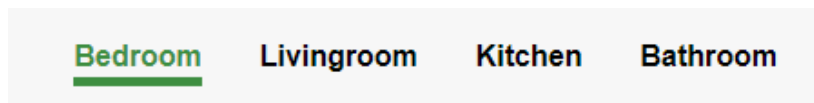
Aplicația este formată din trei pagini: Home, Room, Configuration. În continuare le vom prezenta pe rând, menționând funcționalitățile fiecăreia și cum se realizează legătura cu aplicația server sau alte aplicații externe.

Pagina Home este formată dintr-un antet care conține un buton ce accesează însăși această pagină. Acest buton cu o iconiță în formă de casă este activ și prezent pe fiecare pagină a aplicației pentru a naviga ușor la pagina de start a interfeței. Pe urmă, în cadranul principal al paginii, avem un fundal reprezentativ pentru aplicația creată și 4 butoane ce reprezintă camerele uzuale într-o casă. Utilizatorul poate accesa oricare din aceste camere apăsând pe butoanele rotunde din centrul paginii.



Imagine 4 Pagina principală a aplicației web

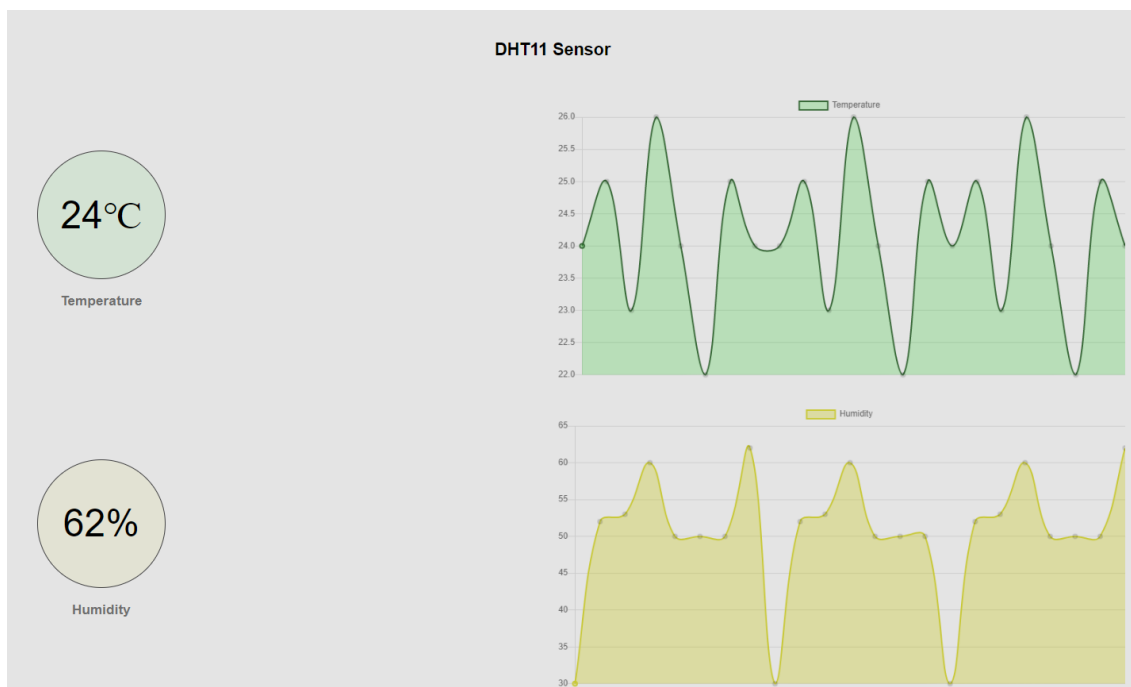
Navigarea în camera selectată se va face instant iar aplicația va face apelul către a doua pagină creată. Aceasta este împărțită în 3 categorii. Prima secțiune din această pagină este tot un meniu care poate accesa pagina Home, dar pe lângă aceasta, în pagina destinată camerei selectate mai apar încă 4 butoane ce permit navigarea utilizatorului la alte camere fără a se întoarce în pagina de start Home.



Imagine 5 Meniul de navigare în diferite camere din pagina Room

A doua categorie face referire la partea de citire a senzorilor. Se dorește a fi o pagină intuitivă și ușor de folosit, de aceea și această secțiune de afișare a informațiilor este divizată în mai multe părți. În această secțiune a paginii utilizatorul poate avea acces la informațiile provenite de la senzorii din camera accesată. Fiecare senzor are diviziunea lui în pagină, informațiile acestora fiind ușor de distins și de interpretat. De asemenea există un titlu cu senzorul care afișează date la fiecare diviziune a paginii. Pe lângă datele provenite de la senzori în format text, utilizatorul poate observa și câteva grafice reprezentative ce reprezintă istoricul pe un timp limitat al unor anumiți senzori.

Datele pe baza cărora se trasează aceste grafice sunt obținute prin apeluri la aplicația server. Pentru fiecare tip de metrică dorit se realizează un apel asincron către un endpoint care conține datele actualizate din baza de date. Un apel asincron inițiază o cerere HTTP către server, însă nu așteaptă ca acesta să răspundă. În schimb, se oferă o metodă care să fie apelată în momentul în care răspunsul server-ului este disponibil. Dacă din anumite motive server-ul nu răspunde în timp, celelalte funcționalități ale paginii vor fi în continuare disponibile utilizatorului, nealterând funcționarea aplicației.



Imagine 6 Informațiile curente provenite de la senzorul DHT11 și reprezentarea grafică a istoricului evoluției datelor în încăpere

O a treia categorie creată în pagina curentă este cea care face referire la controlul dinamic pus la dispoziție pentru utilizator. Această secțiune își propune să rezolve o problemă sau o cerință a utilizatorului și anume: Cum să controlez casa într-un mod inteligent, la o apăsare de pe telefonul mobil? Acesta poate controla cu ușurință cu ajutorul oricărui dispozitiv cu conexiune la internet mai multe funcționalități conținute de aplicație. Pentru a răspunde la întrebarea adresată anterior, putem spune că utilizatorul va putea folosi cele 2 butoane puse la dispoziție pentru a aprinde sau a stinge un LED.

Control LED



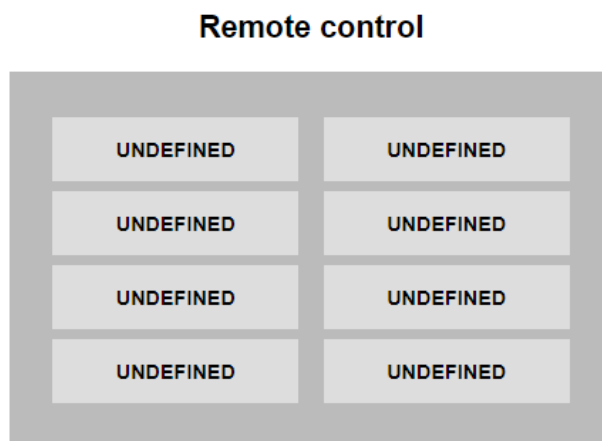
Imagine 7 Controlul LED-ului din aplicația web

Această funcționalitate pare simplă, însă este benefică atunci când totul se întâmplă rapid și fără fire. Prin simpla atingere a butonului destinat aprinderii sau stingerii LED-ului, aplicația web trimite o cerere la aplicația server și aceasta deservește rutina prin metode de tip get ce apelează diferite rute definite în aplicația server. La apelul rutei pentru aprinderea LED-ului, aplicația server știe că va trebui să trimită un obiect în format JSON pe port-ul USB, iar acest obiect, odată ajuns la plăcuța ESP32 este transmis mai departe plăcuței care este desemnată să aprindă LED-ul. Comunicația între cele 2 componente hardware (cele 2 plăcuțe ESP32) se realizează fără fir prin protocolul de transmisiune al datelor ESP-NOW.

Utilizatorul poate controla un motor pas-cu-pas ce controlează la rândul său o draperie sau o ușă din casa inteligentă. Aceasta funcționalitate este realizată la fel, prin cereri HTTP trimise către aplicația server, iar aceasta la rândul său apelează anumite rute ce conțin metode menite să rezolve problema pusă de utilizator.

TODO: Imagine cu controlul motorului pas-cu-pas

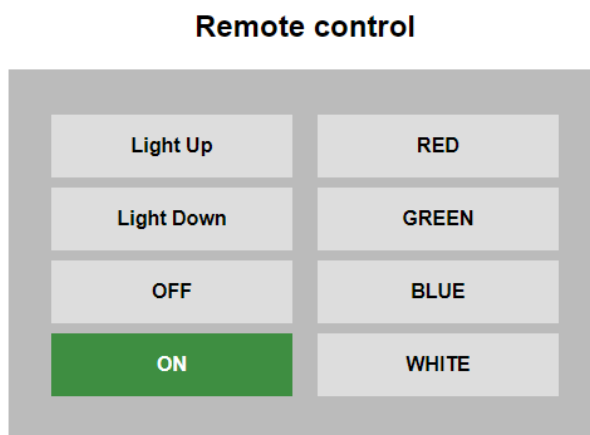
Nu în ultimul rând, poate controla butoane ce fac referire la comenzi infraroșu ce pot controla orice dispozitiv care primește astfel de comenzi. Mai jos se poate observa o captură de ecran în care avem afișată temperatura curentă ce se găsește în cameră, precum și un grafic cu istoricul temperaturii în ultima perioadă de timp.



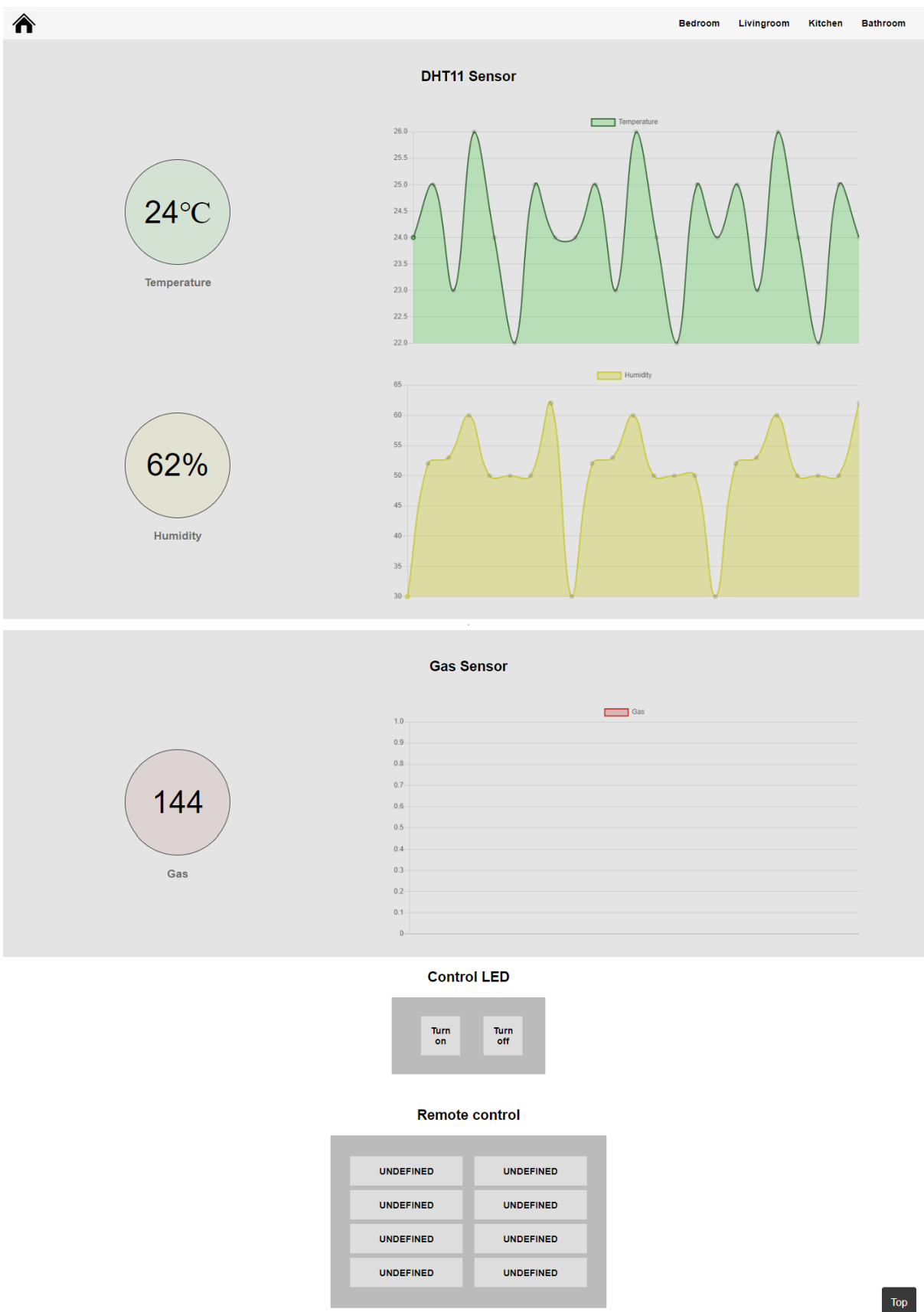
Figură 5 Secțiunea de control din aplicația web

Butoanele pentru funcțiile de infraroșu sunt fără funcționalitate atunci când aplicația este predată utilizatorului, dar acesta poate crea comenzile dorite de el pentru orice telecomandă infraroșu din casa acestuia.

Tot în pagina destinată fiecărei camere, utilizatorul va putea să acceseze o nouă rută care îl va duce pe pagina destinată stocării în baza de date a comenzilor infraroșu. Acesta va putea introduce de la tastatură un nume pentru comanda dorită (ex: Aprinde TV), va apăsa butonul de “Introducere” și apoi va apăsa butonul de Aprinde TV de pe telecomanda fizică îndreptată către senzorul IR pentru recepția datelor. În urma acestor operațiuni, utilizatorul se poate întoarce pe pagina senzorului și poate folosi noul buton atribuit funcției dorite de acesta.



Imagine 8 Butoane destinate comenzilor infraroșu pentru telecomandă OSRAM pentru becul LED inteligent



Imagine 9 întreaga pagină destinată fiecărei camere

5.3 Soluția hardware

Din punct de vedere hardware, prototipul propus este format din mai multe componente. Acesta va fi construit dintr-o stație de bază – o plăcuță de dezvoltare Nucleo-64, senzori pentru măsurarea luminozității din încăpere (KY-018), măsurarea temperaturii și a umidității (ambele putând fi preluate de la un singur senzor – DHT11), măsurarea nivelului de gaz din încăpere (MQ-2).

Pe lângă senzorii care transmit date despre starea casei, avem și componente hardware care ajută la automatizarea locuinței – senzor PIR pentru facilitarea aprinderii luminii la detectarea prezenței, senzori IR care detectează prezența unor unde infraroșu, iar sistemul poate învăța coduri de la telecomanda TV/AC și poate transmite mai departe acele coduri direct din aplicația Web, motoare stepper care ajută la închiderea/deschiderea ușilor/ferestrelor în cazul în care utilizatorii vin/pleacă de acasă, sau se detectează o cantitate semnificativă de gaz sau de fum.

Proiectul de față își propune crearea unui sistem bine definit ce poate duce la stabilitatea psihică din punct de vedere al stresului și al neliniștii utilizatorilor cu privire la propria casă. Acest proiect este menit să aducă în casa fiecărui om un sistem inteligent de control al locuinței și o aplicație în care utilizatorul să poată observa în timp real starea casei oferită de senzorii în cauză.

Soluția hardware este una simplă, care necesită costuri reduse de implementare și nu implică schimbări majore în planul casei sau în arhitectura pereților casei. Această soluție se poate implementa într-un timp relativ scurt și poate oferi informații prin comunicație wireless (fără fir) de la senzori la server.

Componentele hardware au fost gândite în așa fel încât utilizatorii doar să le pornească și acestea să ofere deja informații și date sistemului software. Soluția propusă este ușor scalabilă și poate fi implementată în fiecare încăpere din casă cu ușurință, fără a fi nevoie de mult prea multe schimbări și configurări.

Prin schema bloc prezentată la începutul acestui capitol putem vedea cum fiecare componentă este legată în sistem și comunică wireless prin protocolul de comunicație creat de Espressif – ESP-NOW.

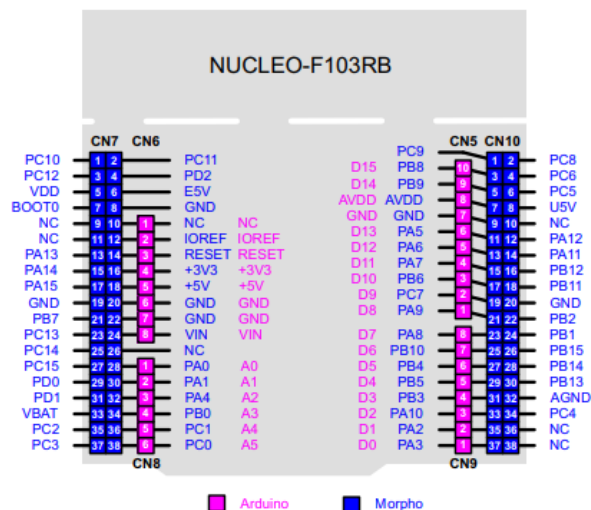
Cu ajutorul mai multor plăcuțe de dezvoltare ESP32 este posibilă comunicația wireless între senzori și stația principală (STM32) și astfel este o soluție ușor de implementat care nu necesită tragerea firelor prin casă. Prin faptul că avem și un LCD legat la plăcuța mamă principală a sistemului, se pot observa cu ușurință anumite informații și date provenite de la senzori cum ar fi temperatura sau umiditatea din încăpere.

În continuare voi prezenta fiecare componenta a schemei bloc în parte:

5.3.1 Plăcuțe folosite

5.3.1.1 STM32 Nucleo-64 F103

Plăcuța STM32 Nucleo este o variantă ieftină de dezvoltare a unor proiecte și are o platformă ușor de folosit, de codat și de înțeles. Alimentarea acesteia cu putere se poate fi oferită fie din partea computer-ului, folosind port-ul USB, fie de către o sursă de tensiune externă.



Figură 6 NUCLEO-F103RB

Aceasta este folosită în proiectul de față ca fiind stația principală prin care trec toate informațiile și datele. Nucleo-64 face posibilă comunicația server-ului cu plăcuțele ESP32 printr-o comunicație serială. Aceasta comunică prin interfața USB cu calculatorul pe care este găzduit server-ul și transmite datele către baza de date. În același timp, plăcuța de dezvoltare Nucleo-64 face posibilă comunicația cu master-ul ESP32 prin interfața UART. De asemenea, este conectat la aceasta și un LCD display pe care se pot citi diferite informații despre starea casei cum ar fi temperatura și umiditatea din încăpere. Protocolul de transmisie a datelor UART(Universal Asynchronous Receiver-Transmitter) este folosit de aproape toate microcontrolerele și toate calculatoarele.

Acesta folosește o interfață de comunicație serială care poate transmite și poate recepționa date. Pachetele de date trimise prin acest protocol sunt formate din biți de start și de stop, din biți de date și opțional din biți de paritate care pot fi folosiți pentru detectarea erorilor la transmisia de date. Majoritatea interfețelor UART folosesc 8 biți de date. [7]

Comunicația în acest protocol poate fi de 3 tipuri:

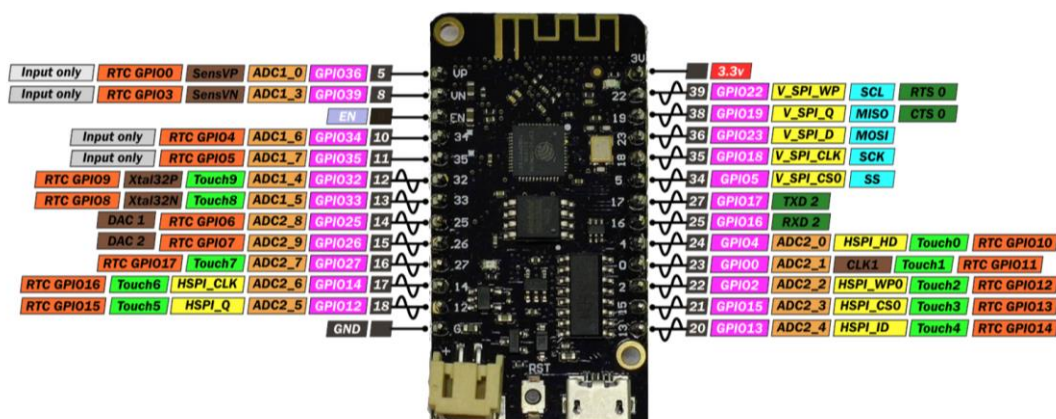
- simplex – transmisia se realizează doar într-o singură direcție, fără ca receptorul să poată transmite date înapoi către transmițător
- full duplex – ambele dispozitive (și receptorul și transmițătorul) pot transmite și primi date în același timp
- half duplex – dispozitivele transmit și primesc date pe rând

5.3.1.2 ESP32 Lolin32

ESP32 este un microcontroler puternic SoC (System on Chip) cu Wi-Fi 802.11 b/g/n integrat, versiunea Bluetooth 4.2 și o varietate de periferice. Este un succes avansat al cipului ESP-8266 în primul rând prin implementarea a două nuclee în versiuni diferite de până la 240 MHz. În comparație cu predecesorul său, cu excepția acestor caracteristici, extinde și numărul de pini GPIO (General Purpose Input Output) de la 17 la 36, numărul de canale PWM (Pulse-width Modulation) la 16 și este echipat cu 4MB de memorie flash.

Microcontrolerele se conectează de obicei cu module IoT și alți senzori inteligenți și oferă date sistemului superior. Plăcuța de dezvoltare folosită în proiect Lolin32 este dezvoltată de firma Wemos și este perfectă pentru a fi folosită în proiecte IoT. Aceasta are un consum redus de putere, dimensiuni reduse și poate integra o multitudine de senzori. Comunicația folosită este ESP-NOW.

ESP-NOW este un protocol de comunicație wireless care poate face transmisiune de date peer-to-peer sau broadcast. Acesta este un protocol care necesită un consum redus de putere și poate fi folosit alături de tehnologia deep-sleep. Împreună pot reduce consumul de putere semnificativ până la aproximativ 120mA în timpul transmisiei și 0.1mA în timp ce sistemul se află în deep-sleep. [8]

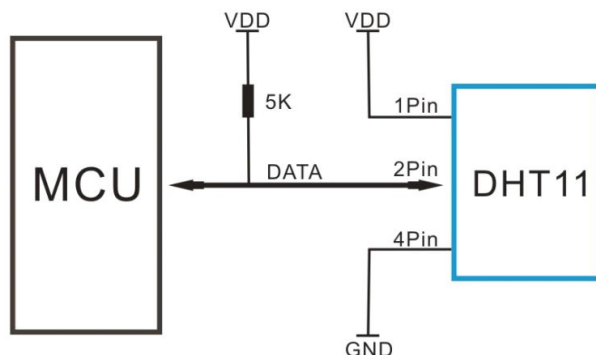


Figură 7 ESP32 Pinout

5.3.2 Senzori folosiți

5.3.2.1 Senzor de temperatură și umiditate

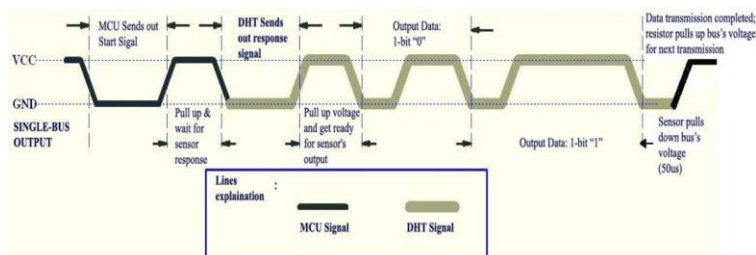
Senzorul de temperatură și umiditate DHT11 este unul complex cu o ieșire de semnal digital care este calibrată încă din fabrică. Acesta asigură o fiabilitate ridicată și stabilitate excelentă pe termen lung. Senzorul include o componentă de tip rezistiv ce măsoară umiditatea și o componentă NTC pentru a măsura temperatura. Acesta se conectează la un microcontroller pe 8 biți oferind astfel o calitate excelentă a măsurătorilor, un timp de răspuns redus și de asemenea un cost redus pentru ce oferă comparativ cu produsele de pe piață.



Figură 8 Conectarea senzorului DHT11 la microcontroler

Sursa de tensiune la care se poate alimenta senzorul DHT11 se află între 3 și 5.5V. Când se alimentează senzorul este recomandat ca timp de o secundă să nu se ceară date de la acesta pentru a nu primi o stare instabilă de la inițializare. Procesul de comunicare este pe magistrala simplă și este folosit pentru sincronizarea și comunicația dintre microcontroler și senzor. Un pachet de date complet se transmite în aproximativ 4ms și conține 40 biți de informație.

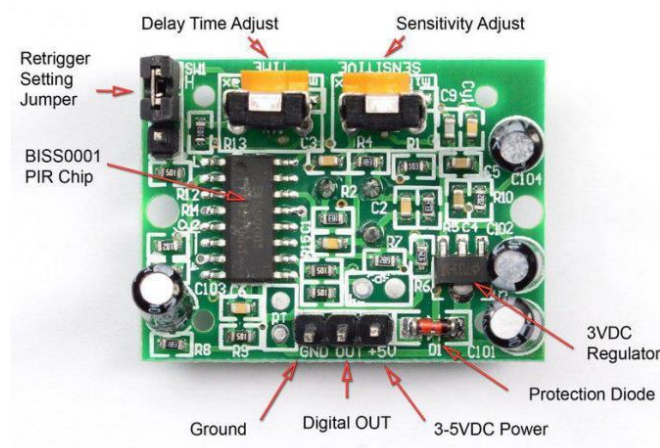
Când microcontroler-ul trimite un semnal puls de start, senzorul DHT11 își schimbă starea din modul low-power în modul running, așteptând ca microcontroler-ul să termine pulsul de start și acesta să înceapă transmisia celor 40 biți în care avem incluse informațiile despre temperatură și umiditate. Utilizatorul poate alege ce date preia de la senzor: fie umiditate, fie temperatură, fie ambele. Odată ce datele utile sunt transmise, senzorul intră din nou în modul low-power și așteaptă un nou semnal de start de la microcontroler. [9]



Figură 9 Comunicația MCU - DHT11

5.3.2.2 Senzorul PIR (Passive Infrared Sensor)

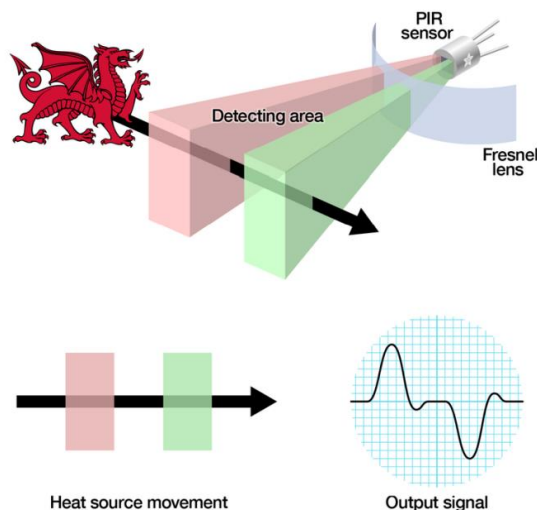
Acesta se folosește de la detecția mișcării unei persoane în sau în afara razei senzorului. Este un senzor de dimensiuni reduse, ușor de folosit, redus la capitoarele costuri și consum de putere. Acesta este folosit des în aplicații de case inteligente sau chiar în companii pentru a reduce consumul de energie electrică.



Figură 10 Senzorul PIR

Componenta hardware folosește o ieșire digitală (3V) care este asertată atunci când este detectată mișcare de către senzor și este 0V când nu se detectează mișcare. Marja de sensibilitate a senzorului poate ajunge până la 7 metri, >120 grade, iar timpul de întârziere poate fi ajustat între 0.8 și 18 secunde. [10]

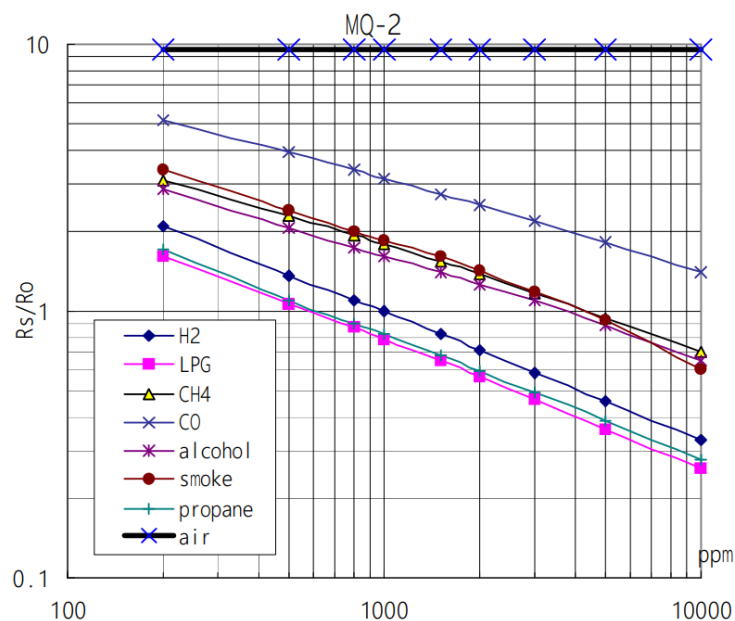
Tensiunea de alimentare a senzorului este între 5 și 12V dar se poate alimenta și la 3.3V cu ajutorul unui regulator de tensiune montat pe modulul senzorului. Senzorul PIR are în componența sa 2 sloturi ce conțin materiale diferite, ambele fiind sensibile la infraroșu. Atunci când senzorul este în repaus și nu detectează nimic, înseamnă ca cele 2 componente primesc aceeași cantitate de unde infraroșu. În momentul în care senzorul detectează mișcare și temperatura corpului uman în raza de mișcare se trimite un puls și astfel se detectează mișcarea și se transmite semnalul de ieșire către MCU.



Figură 11 Funcționarea senzorului PIR

5.3.2.3 Senzorul de gaz MQ – 2

Scurgerile de gaze sunt una dintre marile probleme din sectorul industrial, dar nu numai. Putem avea probleme majore și în propria casă dacă avem scurgeri de gaz. Una dintre soluțiile de reducere a pierderilor de gaz este detectarea precoce atunci când există o scurgere. Există deja multe tehnologii care ajută la prevenirea daunelor provocate de către scurgerile de gaze, una dintre ele fiind senzorul MQ – 2 . [11]

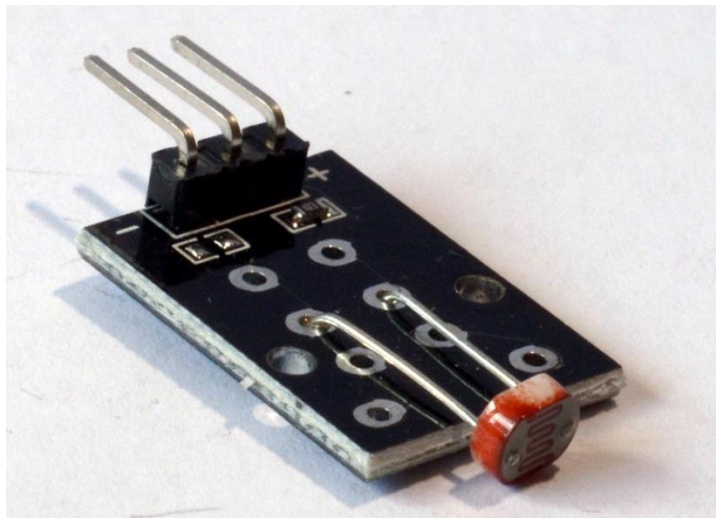


Figură 12 Caracteristica de sensibilitate a senzorului MQ - 2

Folosind un senzor de gaz de tip MQ – 2 ca detector, este de așteptat ca scurgerile de gaz să fie detectate în timp util, ulterior neavând impact mai larg asupra oamenilor. Senzorul MQ – 2 a fost ales deoarece are un preț scăzut și o durabilitate mare. MQ-2 este calibrat și configurat folosind limbajul C, care este implementat prin Arduino IDE. După configurarea senzorului de gaz, este de așteptat ca rezultatele de precizie să ajungă la 80% cu distanța de la senzorul de gaz până la punctul de scurgere de aproximativ 0-10 cm. [12]

5.3.2.4 *Senzorul de lumină fotosensibil KY-018*

Senzorul KY-018 este un modul cu ieșire de tip analog, și se poate citi informația oferită de acesta ca voltaj al pinului de semnal atribuit acestuia. Senzorul poate fi alimentat la 3.3V sau la 5V, depinde de aplicația la care este întrebuințat și de sursa de tensiune folosită. Acesta poate citi date pe 12 biți între valorile de referință 0 și 4095. Se poate calcula rezistența cu ajutorul voltajului măsurat pe pinul analog. În funcție de valoarea de threshold pe care o setăm din programul software, când valoarea rezistorului ajunge mai jos de acest prag, se declanșează un semnal prin care putem aprinde led-ul și astfel putem avea control asupra luminilor din casă.



Figură 13 Senzorul de lumină KY-018

Acest modul deja are integrat în componența sa hardware un rezistor SMD R103 care poate percepe schimbările de luminozitate și face ca senzorul să fie foarte senzitiv și precis în măsurarea cantității de lumină percepute. Cu cât rezistența rezistorului liniar este mai mică, cu atât avem senzitivitate și precizie mai bună în condiții de lumină ridicată.

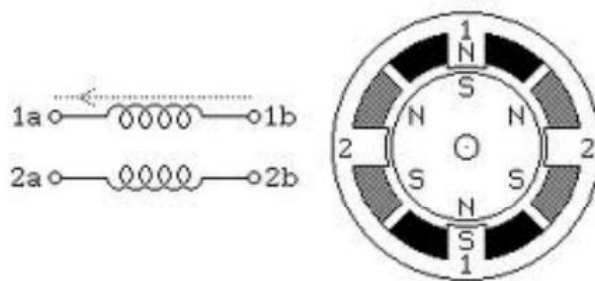
5.3.2.5 Motorul pas-cu-pas

Motorul stepper poate fi văzut ca un motor sincron AC dar care are numărul de poli (și pe rotor și pe stator) mai mare. Acesta este caracterizat de mai mulți parametri printre care se numara și:

- Voltajul: Motorul folosit în aplicația curentă lucrează între 5V și 12V. În funcție de tensiunea de alimentare la care este supus motorul cu pas, acesta poate produce o putere de rotație mai mare sau mai mică

- Rezistența: “Rezistența la rotație este o altă caracteristică a unui motor pas-cu-pas. Această rezistență va determina intensitatea curentului motorului, dar va afecta și curba cuplului motorului și viteza maximă de funcționare.” [13]

- Grade pe fiecare pas: Aceasta este probabil cea mai importantă caracteristică atunci când se alege un motor pas-cu-pas. Acest număr reprezintă numărul de grade cu care se va roti tija motorului și astfel putem calcula de câte rotații ale tijei este nevoie pentru a acoperi o rotație completă de 360 grade.



Figură 14 Funcționarea unui motor pas-cu-pas bipolar

5.3.2.6 Senzorul IR transmițător și IR receptor

6 EVALUAREA REZULTATELOR

7 CONCLUZII

8 CONTRIBUȚII PERSONALE

8 BIBLIOGRAFIE

- [1] Shah, Hezbullah & Soomro, Tariq. (2017). Node.js Challenges in Implementation. Global Journal of Computer Science and Technology. 17. 72-83.
- [2] Bangare, Sunil & Gupta, S & Dalal, M & Inamdar, A. (2016). Using Node.Js to Build High Speed and Scalable Backend Database Server. International Journal of Research in Advent Technology (E-ISSN: 2321-9637). 4. 19.
- [3] Express/node introduction - learn web development: MDN. Learn web development | MDN. (n.d.). Retrieved May 23, 2022, from https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction
- [4] Fezari, Mohamed & Al Dahoud, Ali. (2018). Integrated Development Environment "IDE" For Arduino.
- [5] Why Visual Studio Code? (2021, November 3). <https://code.visualstudio.com/>. Retrieved May 25, 2022, from <https://code.visualstudio.com/docs/editor/whyvscode>
- [6] Michael Plainer. (2020). Study of Visual Studio Code
- [7] Nanda, Umakanta & Pattnaik, Sushant. (2016). Universal Asynchronous Receiver and Transmitter (UART). 1-5. 10.1109/ICACCS.2016.7586376.
- [8] Babiuch, Marek & Foltýnek, Petr & Smutný, Pavel. (2019). Using the ESP32 Microcontroller for Data Processing. 1-6. 10.1109/CarpethianCC.2019.8765944.
- [9] DHT11-Technical-Data-Sheet. <https://www.mouser.com>. Retrieved June 7, 2022, from <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>
- [10] PIR Motion Sensor. <https://cdn-learn.adafruit.com>. Retrieved June 7, 2022, from <https://cdn-learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf>
- [11] Technical Data MQ-2 Gas Sensor. <http://www.hwsensor.com>. Retrieved June 7, 22, from <https://www.mouser.com/datasheet/2/321/605-00008-MQ-2-Datasheet-370464.pdf>
- [12] Trisnawan, I & Jati, Agung & Istiqomah, Novera & Wasisto, Isro. (2019). Detection of Gas Leaks Using The MQ-2 Gas Sensor on the Autonomous Mobile Sensor. 177-180. 10.1109/IC3INA48034.2019.8949597.
- [13] Roy, Tanu & Kabir, Humayun & Chowdhury, Md. (2014). Simple Discussion on Stepper Motors for the Development of Electronic Device. 5. 1089-1096.

9 ANEXE