



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

MONITORIZAREA CONSUMULUI DE ENERGIE ELECTRICĂ

LUCRARE DE DIPLOMĂ

Autor: **George-Sergiu Bălăucă**

Conducător științific: **Șl.dr.ing. Valentin SITA**

2017



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

Vizat,

DECAN
Prof.dr.ing. Liviu MICLEA

DIRECTOR DEPARTAMENT AUTOMATICĂ
Prof.dr.ing. Honoriu VĂLEAN

Autor: **George-Sergiu Bălăucă**

Monitorizarea consumului de energie electrică

1. **Enunțul temei:** *Monitorizarea online a consumului de curent prin intermediul unei prize inteligente și controlul acesteia prin acționarea unui releu.*
2. **Conținutul proiectului:** *(enumerarea părților componente) Pagina de prezentare, Declarație privind autenticitatea proiectului, Sinteza proiectului, Cuprins, Introducere, Studiu bibliografic, Analiză, proiectare și implementare, Concluzii , Bibliografie, Anexe.*
3. **Locul documentației:** *Universitatea Tehnică din Cluj-Napoca*
4. **Consultanți:**
5. **Data emiterii temei:** 24.11.2016
6. **Data predării:** 11.07.2017

Semnătura autorului _____

Semnătura conducătorului științific _____

**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

**Declarație pe proprie răspundere privind
autenticitatea lucrării de diplomă**

Subsemnatul(a) George-Sergiu BALĂUCĂ,
legitimat(ă) cu CI/BI seria XT nr. 543519, CNP 1940405070011,
autorul lucrării:

Monitorizare consumului de energie electrică

elaborată în vederea susținerii examenului de finalizare a studiilor de licență

la **Facultatea de Automatică și Calculatoare**,

specializarea **Automatică și Informatică Aplicată**,

din cadrul Universității Tehnice din Cluj-Napoca,

sesiunea Iulie 2017 a anului universitar 2016-2017,

declar pe proprie răspundere, că această lucrare este rezultatul propriei activități intelectuale, pe baza cercetărilor mele și pe baza informațiilor obținute din surse care au fost citate, în textul lucrării, și în bibliografie.

Declar, că această lucrare nu conține porțiuni plagiate, iar sursele bibliografice au fost folosite cu respectarea legislației române și a convențiilor internaționale privind drepturile de autor.

Declar, de asemenea, că această lucrare nu a mai fost prezentată în fața unei alte comisii de examen de licență.

În cazul constatării ulterioare a unor declarații false, voi suporta sancțiunile administrative, respectiv, *anularea examenului de licență*.

Data

George-Sergiu BALĂUCĂ

(semnătura)



SINTEZA

lucrării de diplomă cu titlul:

„Monitorizarea consumului de energie electrică”

Autor: George-Sergiu BĂLĂUCĂ

Conducător științific: **Sl.dr.ing. Valentin SITA**

1. Cerințele temei:

Monitorizarea online a consumului de energie electrică, tradus direct în cost, al unui consumator la alegere, cu scopul de a avertiza utilizatorul asupra consumului inutil de energie sau de informare.

2. Soluții alese:

Ca tehnologii și elemente de construcție s-au ales plăci de dezvoltare Arduino Uno R3, Nano, comunicație Bluetooth BLE, senzori de curent și conexiune Ethernet pentru comunicarea de la distanță cu sistemul, prin intermediul unei interfețe web. Gazda serviciilor web este reprezentată de un server de la distanță pe care lucrează o bază de date MySQL.

3. Rezultate obținute:

În urma implementării fizice și a realizării conexiunii cu interfața, s-a putut realiza citirea consumului de energie de la priză și stocarea acestuia în baza de date, în vederea procesărilor ulterioare. Totodată, s-a reușit acționarea prizei de la distanță prin acționarea unui releu.

4. Testări și verificări:

Sistemul fizic a fost supus unor serii de validări prin comparații cu citirile unui multimetru și prin măsurarea unor consumatorii cărora li s-a știut consumul. Totodată, s-a pus accentul pe siguranță, urmându-se pași pentru izolarea acestuia. Interfața web a fost supusă unor diferite scenarii de utilizare.

5. Contribuții personale:

Tușa personală este evidențiată în cadrul structurii interfeței web, cât și în arhitectura sistemului fizic ce are ca rezultat comunicația prin Bluetooth între două module separate, dintre care unul portabil.



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

6. Surse de documentare:

Sursele bibliografice sunt reprezentate prin cărți științifice, articole de pe internet, cât și secvențe de cod puse la dispoziție în medii de dezvoltare precum „Arduino IDE” sau librării pentru dezvoltare web deja existente și disponibile.

Semnătura autorului

Semnătura conducătorului științific

Cuprins

1	INTRODUCERE.....	3
1.1	CONTEXT GENERAL	3
1.2	OBIECTIVE.....	3
1.2.1	<i>Obiective hardware.....</i>	3
1.2.2	<i>Obiective software.....</i>	3
1.2.3	<i>Obiective Generale.....</i>	4
1.3	SPECIFICAȚII	4
2	STUDIU BIBLIOGRAFIC.....	6
2.1	PIERDERI DE ENERGIE	6
2.2	TARIFE.....	6
2.2.1	<i>Clasificarea tarifelor.....</i>	6
2.3	CONSUMATOR REZISTIV VERSUS CONSUMATOR INDUCTIV	7
2.4	STUDIU TEORETIC	8
2.5	INSTRUMENTE NUMERICE.....	10
2.5.1	<i>Clase de precizie.....</i>	11
2.5.2	<i>Consum propriu</i>	12
2.5.3	<i>Capacitatea de suprasarcină</i>	12
2.6	ANALIZA COMPONENTELOR UTILIZATE.....	12
2.6.1	<i>Senzorul de curent</i>	12
2.6.2	<i>Plăcile de dezvoltare Arduino</i>	13
2.6.3	<i>Arduino NANO</i>	13
2.6.4	<i>Arduino Uno REV3.....</i>	13
2.6.5	<i>Ethernet Shield.....</i>	14
2.6.6	<i>Releul</i>	14
2.7	APLICAȚII SIMILARE	14
3	ANALIZĂ, PROIECTARE, IMPLEMENTARE.....	16
3.1	ARHITECTURĂ ȘI CONFIGURARE	16
3.1.1	<i>Setări hardware</i>	16
3.1.1.1	<i>Configurare module Bluetooth</i>	17
3.1.1.2	<i>Alimentare și conectare la internet.....</i>	18
3.1.2	<i>Serverul.....</i>	19
3.1.2.1	<i>Găzduire online</i>	19
3.1.2.2	<i>Baza de date.....</i>	24
3.1.2.3	<i>Servicii PHP.....</i>	27
3.1.2.4	<i>Autentificare</i>	27
3.1.2.5	<i>Pagina principală</i>	28
3.1.2.6	<i>Pagina de citiri</i>	28
3.1.2.7	<i>Pagina de rezultate.....</i>	28
3.1.2.8	<i>Pagina de istoric</i>	28
3.2	IMPLEMENTARE.....	29
3.2.1	<i>Fișiere Arduino cu extensia .ino</i>	29
3.2.1.1	<i>Funcția de prelucrare a datelor de la senzor.....</i>	29
3.2.1.2	<i>Setarea displayului LCD</i>	31
3.2.1.3	<i>Comunicarea serială</i>	31
3.2.1.4	<i>Conectare la server</i>	32
3.2.1.5	<i>Acționarea Releului</i>	33

3.2.2	Fișiere PHP	33
3.2.2.1	Autentificare	33
3.2.2.2	Pagina principală	36
3.2.2.3	Pagina de citiri	37
3.2.2.4	Pagina de rapoarte	41
3.2.2.5	Script PHP pentru inserarea datelor achiziționate	42
3.3	TESTARE ȘI VALIDARE	42
3.3.1	Testarea conexiunilor fizice	42
3.3.2	Teste și măsuri de siguranță	42
3.3.3	Teste de consum	43
3.3.4	Teste de conexiune Bluetooth	43
3.3.5	Teste interfață web	43
3.3.6	Validare măsurători	43
4	CONCLUZII.....	43
4.1	REZULTATE OBTINUTE	44
4.2	DIRECȚII DE DEZVOLTARE	44
5	BIBLIOGRAFIE.....	46
6	ANEXE	48

1 Introducere

1.1 Context general

Într-o lume dependentă de acționarea aparatelor și dispozitivelor din jurul nostru prin intermediul curentului electric, dar și o lume afectată tot mai mult de un nivel tot mai ridicat al consumului de orice fel, este absolut necesară conștientizarea omului asupra nevoii de limitare, de economie. De foarte multe ori, atunci când în cutia poștală apare factura de curent, rămânem uimiți la vederea sumelor ridicate, a costurilor și a consumului de curent. Următorul pas este acela de a întocmi un plan în vederea diminuării consumului. Aici intervine necesitatea acestei lucrări, care materializează un sistem capabil să măsoare orice consumator din casă și să îi traducă direct în lei consumul. Desigur, reducerea costurilor nu va fi realizată prin scoaterea din priză a consumatorilor ce necesită conexiune permanentă la curent, precum frigiderul, însă prin monitorizarea consumului unui cuptor cu microunde ce este conectat la priză chiar și atunci când nu este folosit sau a monitorului extern ce stă peste noapte conectat sau a încărcătorului de telefon, se poate ajunge la concluzia că pe parcursul unei întregi luni se acumulează costuri inutile. Așadar, nu este de ajuns ca utilizatorul să urmărească acest consum online, ci să fie și avertizat asupra consumului diferitelor dispozitive pe parcursul unei perioade de timp alese și totodată să aibă posibilitatea de a controla acest consum.

Ideea există de mult timp și totodată se găsesc diferite implementări ale ei, însă varianta în care vine acest proiect a pornit de la necesitatea portabilității și a ușurinței de configurare a conexiunii la internet. Sistemul poate fi conectat printr-o conexiune hotspot de la telefon distribuită prin intermediul unui laptop, așa încât utilizatorul să poate monitoriza consumul de oriunde, atât timp cât există acoperire de semnal pentru telefon. Spre exemplu, se poate monitoriza consumul unui laptop lăsat peste noapte în priză, pornit fiind și în timp ce procesează niște informații grafice. Se poate ajunge la concluzia că această operațiune va genera un cost de până la un leu.

1.2 Obiective

În vederea implementării proiectului s-au definit o serie de obiective și cerințe, atât funcționale, cât și non-funcționale.

1.2.1 Obiective hardware

Componentele hardware trebuie să asigure o măsurare precisă a datelor și o consistență a lor. Acest lucru este posibil printr-o arhitectură bine pusă la punct, fără existența unor pericole la întrebuințare, precum cele de electrocutare. Sistemul trebuie să fie ușor configurabil și să asigure ușurință în utilizare.

1.2.2 Obiective software

- Interfața va cere utilizatorului autentificarea printr-un nume de utilizator și o parolă

- Pagina principală va oferi o perspectivă asupra proiectului, printr-un scurt text de prezentare și o secvență de imagini de ansamblu asupra componentelor hardware.
- Pagina de citiri va propune utilizatorului opțiunile de începere a înregistrării consumului pentru un dispozitiv, de încetare, de control al prizei prin activarea sau dezactivarea releului și va afișa ultimele cincisprezece citiri de la senzor, pentru o perspectivă asupra variației consumului.
- Pagina de costuri va putea fi accesată numai în momentul apăsării butonului de încetare a înregistrării și va afișa patru rubrici despre consum, cât și un tabel cu toate citirile din intervalul de timp cât s-au efectuat înregistrări.
- Ultima pagină oferă utilizatorului posibilitatea de a vedea un grafic cu istoricul citirilor efectuate.

1.2.3 Obiective Generale

Întreg ansamblul are ca țintă, pe lângă monitorizarea consumului de energie, atragerea atenției utilizatorului asupra consumului inutil și educarea acestuia, în sensul de a-l convinge să își controleze mai bine resursele.

1.3 Specificații

În vederea realizării obiectivelor ce definesc proiectul, s-a urmărit o serie de pași necesari.

Primul pas este definit prin configurarea senzorului de curent cu placa de dezvoltare Arduino Nano și crearea unei funcții de convertire a valorilor citite în curent alternativ.

Următorul pas este reprezentat prin realizarea legăturii dintre placa de dezvoltare Arduino Uno și Nano, prin intermediul conexiunii Bluetooth dintre două module, unul configurat Master și celălalt Slave.

Al treilea pas este acela de a configura un server de la distanță împreună cu o bază de date MySQL, în vederea stocării datelor venite de la senzor.

Al patrulea pas este definit prin realizarea conexiunii plăcii Arduino Uno R3 cu Shieldul Ethernet conectat prin cablu la laptop sau un router configurat la rețea.

Nu în ultimul rând, este necesară realizarea interfeței web pentru prelucrarea informațiilor din baza de date.

Sistemul este proiectat în vederea aplicabilității în cadrul locuințelor pentru monitorizarea consumului de energie, pentru uz propriu, însă nu este exclusă posibilitatea utilizării și la nivel mai ridicat, precum în fabrici. Se urmărește divizarea proiectului în două module: software și hardware, cel din urmă cuprinzând la rândul lui două module ce comunică între ele fără fir. Ca scop principal, se caută simplitate în vederea instalării componentelor pentru a fi pregătite de citire și ușurință în vederea utilizării interfeței web grafice. Componentele hardware sunt alese în așa fel încât să satisfacă nivelul minim de procesare a informațiilor de la senzor, să ocupe puțin spațiu în

cadrul ansamblului și să funcționeze cu un consum redus de energie. În consecință, s-a ales ca prim micro controller o placă de dezvoltare Arduino Nano cu dimensiuni reduse, consum redus de curent și număr redus de pini, însă cu caracteristici ce satisfac obiectivele și anume cele de procesare a informațiilor citite de la senzor și de acționare a releului prin comanda de la distanță prin interfață.

Senzorul utilizat este un senzor de curent cu efect Hall ACS712, capabil a măsura curenți de până la 30 de Amperi, fapt ce duce la o sensibilitate scăzută atunci când este necesară măsurarea consumului unui dispozitiv precum telefonul când este conectat la priză pentru reîncărcarea bateriei. De aceea, în cadrul acestui proiect, limita inferioară de curent măsurabil este de 0.02 Amperi.

Releul are ca scop simpla acționare a prizei, eveniment pus la dispoziția utilizatorului în cadrul paginii web, la rubrica „Vezi citiri live”. Ca limitare, releu folosit poate controla un curent de maxim 10 Amperi.

Cu siguranță se poate alege dintr-o gamă variată de micro-controllere existente pe piață, însă optarea pentru acest model are la bază o serie de motive, precum: posibilitatea de a folosi dispozitivul rapid, prin simpla conectare la portul USB al unui laptop, exemple de cod propuse de mediul de dezvoltare, rapiditatea în procesare, o comunitate răspândită în întreaga lume.

Cea de-a doua placă de dezvoltare aleasă este Arduino Uno R3, întrucât propune posibilitatea conectării cu un Shield ethernet existent pe piață. Astfel, se poate realiza cu ușurință conexiunea la internet, prin simpla asignare a unui IP plăcii, în funcție de tipul conexiunii realizare: router configurat în cadrul locuinței sau laptop folosit pe post de router.

Conexiunea dintre plăci este bazată pe comunicarea dintre două module Bluetooth HC-05, programate ca Master și Slave și configurate în așa fel încât să se conecteze automat la repornirea sistemului, pentru a evita pașii necesari reconectării lor. Ca nivel de performanță se urmărește atingerea unei distanțe de transmitere a datelor între module de minim 20 de metri, distanță fără obstacole, precum pereți sau alte element decorative din cadrul unei locuințe și de minim 10 metri ca distanță cu obstacole.

În ceea ce privește partea software, va fi folosit un server achiziționat, în cadrul căruia s-a efectuat instalarea tehnologiilor Apache, CentOS7, MySQL, PHP, cu scopul de a crea o bază de date în care vor fi stocate informațiile de la senzor, date despre utilizator, cât și date despre dispozitivele cărora le monitorizăm consumul. Capacitatea bazei de date este ridicată, aceasta depinzând de memoria serverului. Pentru a propune utilizatorului datele despre consum și dispozitiv, s-a realizat o interfață web accesibilă de oriunde și de pe oricare dispozitiv, fie el laptop sau smartphone. Aceasta propune un meniu cu 4 opțiuni: „Acasă”, „Vezi citiri”, „Calculează cost” și „Istoricul citirilor”. Toate operațiunile efectuate cu baza de date sunt bazate pe fișiere php și query-uri bazate pe sintaxă MySQL. În ceea ce privește securitatea, accesarea paginii principale va fi condiționată de un formular de log in, ce necesită un nume existent și o parolă validă din baza de date.

2 Studiu bibliografic

2.1 Pierderi de energie

Pe baza informațiilor găsite în cadrul [1], se poate trage un semnal de alarmă clar asupra cantității de energie nu doar consumată, ci și pierdută inutil. Livrarea energiei electrice către consumatori include pierderi majore ce apar în timpul transportului, fapt ce se revarsă puternic asupra factorilor economici. Astfel, cifre de 30-40% din costul transportului reprezentând cel al pierderilor, împreună cu cifre de 8-15% din energie reprezentând pierderi în unele țări, ridică preocupare în ceea ce privește reducerea consumului și a pierderilor. Tot conform [1], „[...]reducerea pierderilor în rețele este mult mai economică decât creșterea corespunzătoare a capacităților de generare.”

Mai mult decât atât, pierderile din sistem, raportate ca diferențe dintre energia generată la centrale și cea furnizată, pot fi împărțite în trei categorii:

- „*Consumul propriu* tehnologic aferent procesului de transport și distribuție[...]” [1]
- „*Pierderi tehnice* prin abateri de la regimul de funcționare proiectat, fie prin dezvoltarea incompletă a instalațiilor, fie printr-o funcționare necorespunzătoare.”
- „*Pierderi comerciale* rezultate din erorile introduse de calitatea grupurilor de măsură și organizarea evidenței energiei electrice, dar cuprinzând și unele consumuri nemăsurate ca cel al transformatoarelor de măsură sau contoarelor, precum și furturilor de energie electrică.”

Cel din urmă tip de pierdere reprezintă problema propusă în cadrul acestui proiect, iar implementările realizate au ca țintă îmbunătățirea evidenței energiei, în ceea ce privește fie administrarea unei locuințe sau chiar fabrici.

Cu toate acestea, dacă vrem să determinăm pierderile dintr-o rețea bazându-ne pe varianta măsurărilor, ajungem la concluzia că este dificil din punct de vedere, atât tehnic, cât și economic.

2.2 Tarife

Unul din principalii piloni ce stau la baza acestui proiect este bine evidențiat în [1], unde se definește următorul principiu: tariful aplicat unui consumator/utilizator trebuie să fie ușor de înțeles și aplicat, în sensul că nu trebuie să conțină ambiguități din punctul de vedere economic. Totodată, nu trebuie să apară alarmări în ceea ce privește tipul de aparat de măsură utilizat.

2.2.1 Clasificarea tarifelor

Conform [1], principalele criterii în care sunt împărțite tarifele pentru energia electrică sunt:

- Costurile pe baza cărora se proiectează (costuri medii și costuri marginale)
- Numărul componentelor de preț ce determină calculul facturii pentru energie (tarife monome, tarife binome, tarife trinome). Acestea sunt exprimate în unități monetare, mai exact în kWh.
- Diferențierea componentelor de preț pe perioadele de tarifare corespunzătoare intervalelor de variație a costurilor sistemului energetic. Astfel, există tarife nediferențiate și tarife diferențiate, care la rândul lor se împart în:
 - (a) Tarife în funcție de perioada din zi;
 - (b) Tarife zi-noapte;
 - (c) Tarife preferențiale de noapte;
 - (d) Tarife în funcție de perioada din săptămână;
 - (e) Tarife în funcție de perioada din an (sezoniere);
- Tarife cu destinații speciale (tarife pentru furnizări cu întreruperi – reduceri de cost pentru oprirea furnizării de energie, în cazul în care sistemul funcționează greu; tarife scop sau speciale – atragerea de noi clienți sau stimularea consumului în anumite zone geografice; tarife în timp real);

Este de așteptat ca o simplă factură de curent venită în cutia poștală să pună în dificultate o persoană, însă, cu siguranță, o clasificare cum este cea de mai sus va complica lucrurile și mai mult. De aceea este binevenit un sistem capabil a traduce direct și rapid costul unui consumator conectat la rețea și de oferi posibilitatea unui control de la distanță.

2.3 Consumator rezistiv versus consumator inductiv

Un alt criteriu demn de luat în considerare, ce are scop informativ este acela al tipului de consumator al cărui consum de energie dorim a-l monitoriza. Astfel, conform [2], există consumatori rezistivi și consumatori inductivi. Consumatorii rezistivi sunt caracterizați prin producerea de căldură și generare de mișcare, pe când consumatorii inductivi sunt descriși prin circulația curentului prin bobină. În această categorie intră majoritatea dispozitivelor din cadrul unei locuințe, cum ar fi televizorul, frigiderul sau calculatorul.

În termeni mai tehnici, pe baza informațiilor regăsite în cadrul [3], consumatorii rezistivi sunt descriși ca fiind cei mai simpli, formele de undă ale tensiunii și curentului potrivindu-se sau fiind în fază. În cazul consumatorilor inductivi, forma de undă a tensiunii este înaintea celei de curent, astfel producându-se o a doua formă de undă ce acționează împotriva aceleia din sursa de alimentare, fenomen cunoscut ca inductanță. Din cauza aceasta, în momentul în care fie oprim dispozitivul sau îl pornim, apar fenomene nedorite, precum cel de val de tensiune sau vârf de tensiune (spike).

2.4 Studiu teoretic

Pentru o abordare corectă a măsurării consumului și traducerii acestuia în cost, am pornit prin a studia bazele, în ceea ce privește circuitele electrice, cât și metodele de măsurare.

„Prin circuite de curent alternativ se înțeleg circuitele electrice alimentate cu tensiuni electromotoare alternative, adică tensiuni periodice de valoare medie nulă.[...]Cu foarte puține excepții, rețelele electrice pentru producerea, transmisiunea și distribuția energiei electromagnetice sînt, în prezent, rețele de curent alternativ sinusoidal cu frecvența standardizată de 50 Hz (în America 60 Hz), numită frecvență industrială.” [4].

Am pornit de la această idee, întrucât tehnologiile aplicate în cadrul acestui proiect sunt aplicate pentru măsurări aplicate în circuite alimentate cu curent alternativ. Cu siguranță senzorul de curent folosit este disponibil a fi folosit și în circuite de curent continuu, însă cum obiectivul proiectului este să atingă nevoile casnice ale consumatorilor uzuali, vom lucra în regim sinusoidal.

Totodată, citind [4], se poate face trimitere la noțiuni precum valoare instantanee și mărime periodică, unde o mărime instantanee este definită printr-o valoare fixată la un moment oarecare, pe când o mărime periodică este caracterizată printr-o succesiune de valori instantanee dintr-un interval de timp. O formulă ce definește această noțiune este:

$$i(t) = I(t + nT) \quad 2.1$$

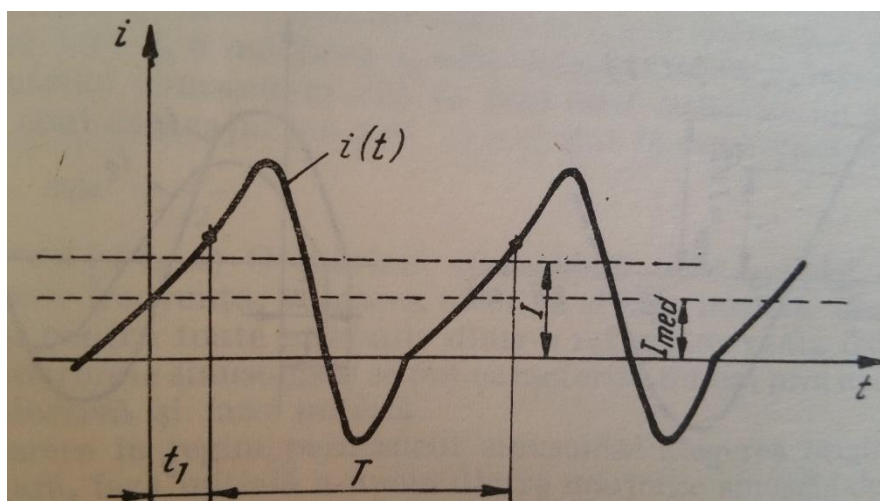
Unde n este un număr oarecare, T este perioada mărimii și i este valoare instantanee.

„Numărul de perioade cuprinse în unitatea de timp se numește frecvență, simbolizată cu litera f .”

$$f = \frac{1}{T} \quad 2.2$$

Unitatea de măsură a frecvenței se numește hertz (Hz).

Mărimile periodice care iau valori de un singur semn se numesc mărimi pulsatorii, iar cele care iau valori de ambele semne – mărimi alternative în sens larg. În ultimul caz, mulțimea valorilor pozitive pe care le ia mărimea într-o perioadă formează alternanța pozitivă, iar mulțimea valorilor negative – alternanța negativă. Valoarea maximă (pozitivă) pe care o ia mărimea în alternanța pozitivă se numește amplitudinea pozitivă, iar modulul celei mai mici valori luate (în alternanța negativă) – amplitudine negativă. La multe mărimi alternative cele două amplitudini sînt egale și atunci se vorbește simplu de amplitudinea mărimii.”



Figură 2.1 Reprezentarea grafică a unei funcții periodice de timp

[...]Valoarea efectivă (sau eficace) a unei mărimi periodice este rădăcina pătrată a mediei pătratelor valorilor ei instantanee într-un interval de timp egal cu o perioadă.” [4]

$$I = \sqrt{\frac{1}{T} \int_{t_1}^{t_1+T} i^2 dt} = \frac{1}{T} \quad 2.3$$

Pe baza ultimelor noțiuni definite, s-au efectuat măsurarea și calculul curentului într-un circuit alimentat de o sursă de tensiune alternativă. Totodată, în cadrul aceleiași referințe, s-a constatat următoarea afirmație: „O mărime sinusoidală este complet determinată când se cunosc frecvența, valoarea efectivă și faza inițială. Dacă frecvența este aceeași pentru toate mărimile dintr-o rețea (frecvența industrială, de exemplu), mărimile sinusoidale se pot caracteriza numai prin doi parametri: valoare efectivă și faza inițială.” [4], cea din urmă putând fi aleasă arbitrar, datorită regimului permanent sinusoidal. Pe baza acestei afirmații, s-au efectuat calcule și procesări asupra datelor venite de la senzor, cu scopul unei măsurări adecvate.

Totodată, datorită regimului permanent sinusoidal, se pot defini curentul și tensiunea prin următoarele formule:

$$i = I\sqrt{2} \sin(\omega t + \varphi) \quad 2.4$$

$$u = U\sqrt{2} \sin \omega t \quad 2.5$$

Unde $\omega t + \varphi$ este faza mărimii sinusoidale, ω este pulsația, iar φ este faza inițială.

De aici s-a mers mai departe la definirea noțiunii de putere activă în regim sinusoidal, mărime necesară la calculul consumului de energie a unui consumator și definită prin următoarea formulă:

$$P = UI \cos \varphi \quad 2.6$$

Conform [5], atunci când φ este 0, vom avea putere maximă atunci când curentul este în fază cu tensiunea, deoarece $\cos \varphi$ va fi egal cu 1.

Mai mult de atât, deoarece scopul final al proiectului este acela de a propune utilizatorului un cost direct al energiei consumate, se definește, tot conform [5], energia activă, ca fiind descrisă de următoarea formulă:

$$W = \int P dt \quad 2.7$$

Această mărime se măsoară în kilowatt-ore [kWh], iar dacă într-un interval de timp considerat puterea este constantă, energia activă poate fi descrisă de următoarea formulă:

$$W = Pt = U_b I \cos \varphi t \quad 2.8$$

Pentru măsurarea energiei electrice, se utilizează contoare de inducție, atât pentru puterea activă, cât și pentru puterea reactivă. Acestea se comportă ca un element motor analog wattmetrului, conform [4]. Se pot folosi wattmetre de tip electrodinamic, ferodinamic sau cu inducție, în cazul rețelelor monofazate [5]. Unitatea folosită în practică se numește kilowatt-oră, iar valoarea unui kWh este egală cu $3.6 * 10^6$ J

În ceea ce privește ideea propusă de acest proiect, se dorește a utiliza un senzor de curent cu efect Hall pentru determina fluctuațiile de consum, astfel făcându-se o diferențiere față de metodele clasice de măsurare.

2.5 Instrumente numerice

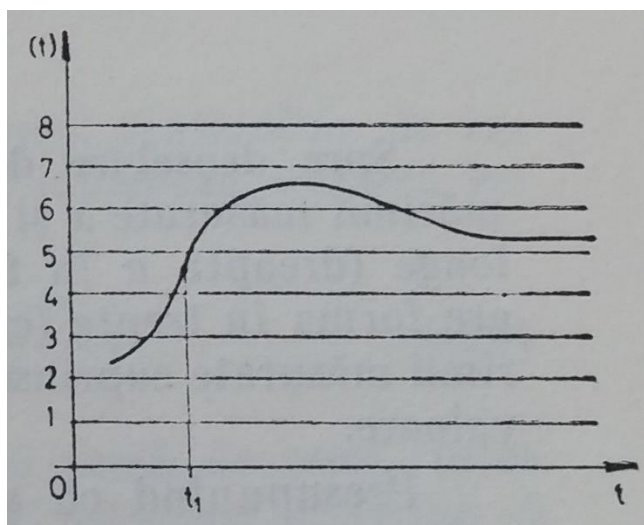
În ceea ce privește metodele de măsurare existente la dispoziția oricărui dintre noi, se poate alege dintr-o gamă de dispozitive împărțită în cele profesionale (multimetre) și cele de uz casnic, ce propun o interfațare mai bogată. În continuare vor fi menționate câteva noțiuni reprezentative pentru funcționarea dispozitivelor numerice, pentru a forma o privire de ansamblu.

În ceea ce privește modul în care funcționează un multimetru, pe baza articolului [6], se poate spune că, în funcție de producător, apar diferențe între multimetre. Principiul de bază rămâne totuși același și anume de măsurare a tensiunii. Celelalte mărimi dorite a fi măsurate sunt derivate din procesul principal. Mai mult decât atât, citind [7] se constată faptul că din cauza costurilor ridicate, nu s-a trecut mai departe la producția industrială a ampermetrelor digitale, preferându-se în acest fel utilizarea voltmetrelor digitale.

Funcționarea unui multimetru are la bază un convertor analog – digital (CAN), care, în cazul multimetrelor digitale, poartă denumirea de registru de aproximări succesive. Acest tip de convertor poate avea o rezoluție pornind de la 12 biți și poate depăși 16 biți, valoare utilizată în majoritatea cazurilor. Rezoluția voltmetrelor numerice se descrie ca

fiind valoarea cea mai mică ce poate fi măsurată pentru fiecare domeniu al instrumentului. Numărul de eşantioane pe secundă este de 100000, număr potrivit pentru cazul acestor tipuri de multimetre, unde este necesar un nivel ridicat al vitezei pentru citire.

În ceea ce priveşte aparatele numerice disponibile pe piaţă, se poate stabili principiul de bază utilizat în măsurare şi anumite faptul că mărimile supuse măsurării sunt în principal continue, iar informaţia dată de un dispozitiv numeric este categoric discontinuă. În acest fel, se defineşte procesul de discretizare sau cuantizare prin care o funcţie continuă $s(t)$ este aproximată printr-o mulţime discretă de valori, mai exact fiecărei valori a funcţiei i se atribuie cea mai apropiată valoare din mulţimea discretă.



Figură 2.2 Cuantizare semnal continuu [7] pg 79

Totodată, se defineşte teorema eşantionării Nyquist-Shannon, pe baza căreia se poate reconstitui un semnal din eşantioane, dacă eşantionarea se efectuează cu o frecvenţă dublă faţă de cea mai mare frecvenţă de semnal.

2.5.1 Clase de precizie

Ca urmare a informaţiilor asimilate din [5] pg(294), reiese faptul că există o serie de clase ce caracterizează instrumentele electrice de măsurare.

- Clasa 0.1 cu eroare admisă de $\pm 0.1\%$;
- Clasa 0.2 cu eroare admisă de $\pm 0.2\%$;
- Clasa 0.5 cu eroare admisă de $\pm 0.5\%$;
- Clasa 1 cu eroare admisă de $\pm 1\%$;
- Clasa 1.5 cu eroare admisă de $\pm 1.5\%$;
- Clasa 2.5 cu eroare admisă de $\pm 2.5\%$;
- Clasa 4 cu eroare admisă de $\pm 4\%$;

2.5.2 Consum propriu

Trebuie ținut cont de fapt că aceste dispozitive consumă energie electrică, datorită efectului Joule-Lenz [5] dezvoltat la trecerea curentului prin rezistența interioară a acestora. În următorul tabel sunt ilustrate câteva variante de consum.

Tabel 1 Consum instrumente de măsurare

Tipul instrumentului	Consum propriu	
	Ampermetru pentru 5 A	Voltmetru pentru 100 V
Magnetoelectric	0.2 ... 0.4	0.1 ... 1
Electromagnetic	2 ... 8	2 ... 5
Electrodinamic	3.5 ... 10	3 ... 6
C inducție	1 ... 4	2 ... 5

„Trebuie făcută o distincție netă între consumul de la obiectul măsurat și cel de la sursele auxiliare care alimentează instalația și care nu influențează condițiile de măsurare.” [7]

2.5.3 Capacitatea de suprasarcină

Un alt aspect demn de luat în considerare este modul de realizare a dispozitivelor de măsurare, astfel încât să suporte sarcini ce depășesc intervalul de măsurare [7]. Acest aspect este util în cazul proiectului, luând în considerare faptul că senzorul de curent utilizat poate măsura curenți de până la 30 de Amperi. Totuși, deși limita este ridicată, sensibilitatea va avea de suferit.

2.6 Analiza componentelor utilizate

Sistemul descris în această lucrare are în compunere componente fizice, precum micro controllere Arduino, module de conectivitate Bluetooth, senzor de curent, releu și o interfață web ce propune utilizatorului posibilitatea controlului prizei și prelucrării datelor venite de la senzor și stocate într-o bază de date pe un server. Aceasta arhitectură este propunerea personală ce se diferențiază de restul dispozitivelor și tehnologiile existente pe piață.

2.6.1 Senzorul de curent

În prezent, există numeroase tehnologii pe baza cărora se poate efectua citirea curentului electric. Varianta aleasă pentru proiect este un senzor invaziv, liniar, ACS712 cu efect Hall, capabil a citi atât curent continuu, cât și alternativ, în limita a 30 de Amperi, negativ sau pozitiv. Astfel, se pot face măsurări ale consumului de curent pentru orice tip de consumator, fie el rezistiv sau inductiv.

Citind [8] și [9] am ajuns să înțeleg principiul de funcționare al unui senzor de acest tip, mai exact să înțeleg definiția forței Lorentz ce face referință la diferența de potențial produsă de devierea electronilor la exercitarea asupra lor a unei forțe magnetice.

Totodată, am ajuns să constat că este unul din cei mai răspândiți senzori folosiți în industrie. Ca și caracteristici tehnice, senzorul poate fi alimentat cu o tensiune continuă cuprinsă între 4.5 V și 5.5 V, consumă 10 mA, are o sensibilitate de ieșire de 66 mV/A și poate avea erori de până la 1.5 %. Cu toate acestea, senzorul nu se va comporta ca un ampermetru digital, având la ieșire valoarea directă a curentului, ci va propaga o valoare numerică, urmând ca din cod să fie procesată așa încât să avem valoarea căutată. În mod cert, consumul ar putea fi măsurat direct cu ajutorul unui multimetru, dispozitiv capabil să afișeze direct mărimea electrică dorită, însă acest tip de citire este folosit doar în cazul validărilor și verificărilor, fără posibilitatea stocării de informații.

Conform [10], alte tehnologii folosite pentru citirea curentului sunt: sesizarea rezistivă a curentului, convertoarele curent-tensiune, senzorii cu buclă închisă și transformatoarele de curent. Totuși, din cercetările efectuate asupra pieței și posibilelor magazine de articole electronice, nu s-au regăsit senzori care să folosească aceste tehnologii și să se preteze conexiunii cu micro controllere Arduino.

2.6.2 Plăcile de dezvoltare Arduino

În ceea ce privește micro controllerele menite a procesa datele și efectua comenzile, am folosit plăci de dezvoltare NANO și UNO R3, întrucât se găsesc pe piață la un nivel scăzut de preț și nu necesită un nivel ridicat al cunoștințelor în ceea ce privește programarea lor.

2.6.3 Arduino NANO

Conform [11], această placă de dezvoltare are în arhitectura sa un microcontroller ATmega328, un ceas ce lucrează la frecvența de 16 MHz, 8 pini analogi de intrare sau ieșire, 6 pini de ieșire PWM, poate fi alimentat cu o tensiune continuă cuprinsă între 7 și 12 V și consumă 19 mA. Pe lângă acestea, prin intermediul celor 8 pini analogi se pot efectua măsurări, lucrând la o rezoluție de 10 biți, adică 1024 de valori distincte, în intervalul de 0 – 5 V. În cadrul proiectului, o parte din pinii digitali au fost folosiți pentru conectarea unui display LCD, cu scopul informativ de a afișa date despre puterea activă consumată, cât și curent. Doi din pinii analogi sunt folosiți în vederea conectării ieșirii senzorului de curent și a intrării releului pentru comanda prizei. Totodată, placa pune la dispoziție doi pini digitali pentru stabilirea comunicării UART TTL serială, mai exact pinii RX și TX. Prin intermediul lor se poate realiza conexiunea cu o altă placă sau, în cazul nostru, cu modulul Bluetooth, prin conexiunea încrucișată (RX -> TX, TX -> RX).

2.6.4 Arduino Uno REV3

Conform [12], placa UNO R3 se diferențiază de NANO printr-un număr mai ridicat de pini digitali și anume 14. Aceștia pot fi configurați ca intrare sau ieșire, iar 6 dintre ei pot fi configurați ca ieșire PWM. Numărul pinilor analogi este 6, iar frecvența de lucru a ceasului este aceeași, de 16 MHz. Alimentarea plăcii se poate face în intervalul limită 6 – 20 V, cu mențiunea limitelor recomandate de minim 7 V și maxim 12 V. Se păstrează

aceeași pini RX și TX necesari pentru conectarea modulului Bluetooth, în cazul nostru, cât și rezoluția de 10 biți propusă de pinii analogi.

2.6.5 Ethernet Shield

Spre deosebire de produsele menite a măsura consumul de curent de pe piață, acest proiect este proiectat a fi conectat prin intermediul cablului ethernet RJ45 la un router sau laptop, fiind capabil să trimită direct date la server. Acest lucru este totuși posibil numai prin intermediul conectării Shieldului Ethernet compatibil cu placa de dezvoltare Arduino UNO Rev3. Conectarea se face prin suprapunerea modului direct pe pinii plăcii de dezvoltare. Pe baza informațiilor regăsite în cadrul [13], aceasta lucrează la o tensiune continuă de 5 V provenită de placă, conține în arhitectura sa un controller Ethernet W5500 și suportă o viteză a conexiunii de 10/100 Mb. Totodată, pune la dispoziție o serie de LED-uri cu caracter informativ: ON – indică faptul că există alimentare; LINK – indică prezența unei conexiuni la rețea și se va aprinde intermitent în cazul transmițerilor de date; FDX – indică faptul că s-a realizat o conexiune la rețea full duplex; 100M – indică prezența unei conexiuni 100 Mb/s, iar ACT – se aprinde intermitent dacă este detectată activitatea pinilor RX, TX.

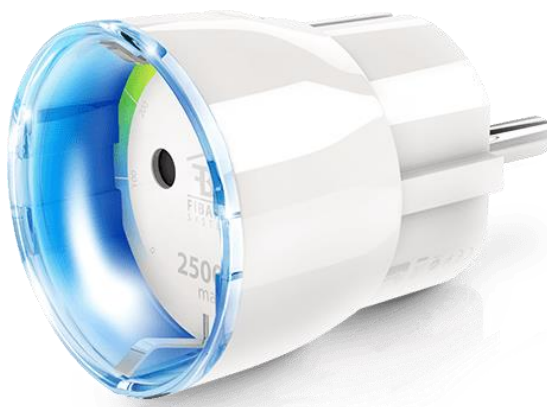
2.6.6 Releul

Pentru controlul de la distanță al prizei, s-a utilizat un modul releu, controlabil prin intermediul micro controllerului Arduino. Conform informațiilor regăsite la [14], modulul funcționează la o tensiune continuă de 5V, perfect pentru interacționarea cu placa de dezvoltare. Poate controla un curent de maxim 10 Amperi la tensiuni de 250VAC sau 30VDC, fapt ce intră într-un conflict cu senzorul de curent utilizat. Funcționarea lui se bazează pe o bobină, un terminal comun, un terminal normal închis și un terminal normal deschis. Când prin bobină trece curent, vor fi conectate terminalele comun și deschis, permițându-se astfel circulația curentului.

2.7 Aplicații similare

Pe lângă dispozitivele numerice ce se găsesc pe piață există și o altă categorie de dispozitive prin care utilizatorul poate măsura și monitoriza consumul de energie, prin intermediul interfețelor accesibile, fie ele sub formă de pagini web sau de aplicații mobile.

O aplicație similară demnă de luat în considerare este cea propusă de Fibaro, o priză inteligentă de dimensiuni mici, construită minuțios și capabilă a satisface aceleași obiective propuse în cadrul proiectului. Totuși, pentru a funcționa, are nevoie de o centrală denumită „Fibaro Home Center”, ce are menirea de a culege date de la priză. Costurile acesteia, conform [15], se ridică la 2.249 de lei. Astfel, pe lângă preț, diferența principală dintre această aplicație și proiectul dezvoltat de mine este aceea că priza Fibaro are nevoie de o conexiune fără fir la internet pentru a trimite datele către centrală, care în cazul nostru, poate fi considerată a fi modulul Arduino Uno Rev 3 conectat prin cablu ethernet, pe când priza din cadrul proiectului trimite datele prin intermediul protocolului Bluetooth.



Figură 2.3 Priză Fibaro de tip F

Interfațarea acestei prize este una de tip web, ce propune numeroase opțiuni de prelucrare a datelor, cât și controlul de la distanță al prizei.



Figură 2.4 Integrare web pentru controlul și monitorizarea prizei Fibaro

3 Analiză, proiectare, implementare

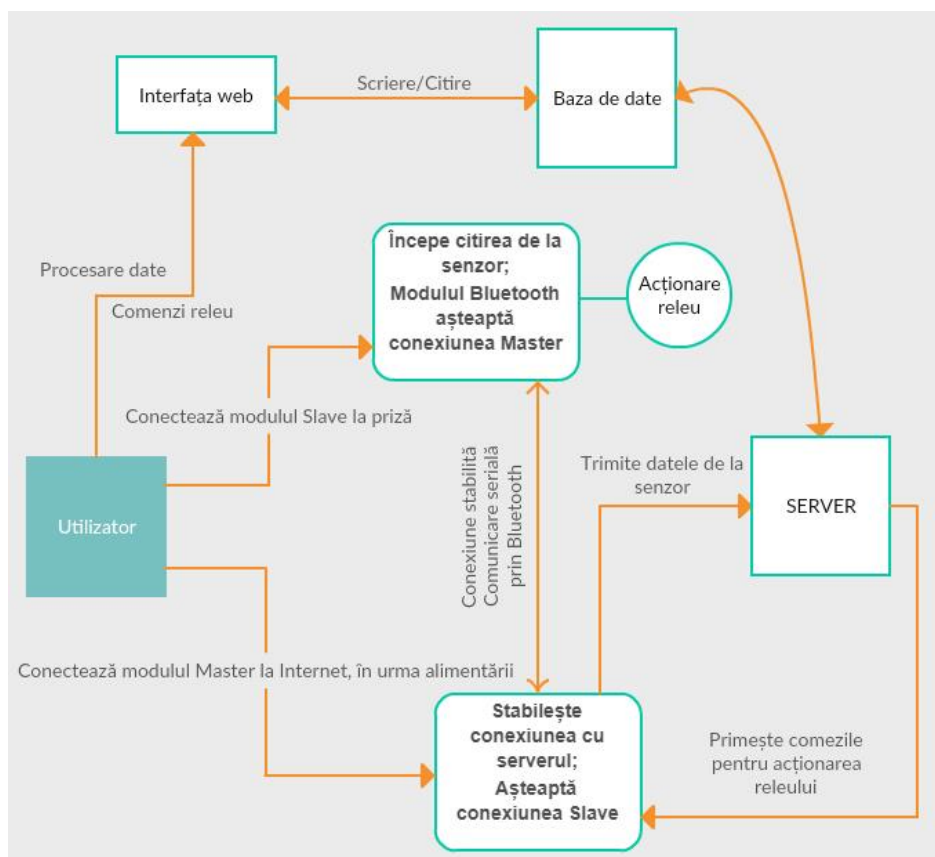
În continuare, voi descrie pașii urmăți în elaborarea proiectului. Mai exact, voi prezenta conexiunile făcute la nivel de componente hardware, detalii despre configurarea componentelor, cât și interfațarea lor cu componenta software (baza de date, fișiere PHP, fișiere cu extensia .ino în limbaj de programare C++).

3.1 Arhitectură și configurare

3.1.1 Setări hardware

Pentru început, voi descrie elementul de referință din cadrul ansamblului fizic și anume priza utilizată pentru măsurare și control. Aceasta priză simplă cu ramă, menită a fi montată pe perete, vine împreună cu un transformator ce va genera curent continuu pe portul USB de 2.1 Amperi încorporat. Pe baza acestuia, vom alimenta modulul Slave compus din placa Arduino NANO. Priza poate genera până la 16A la tensiuni de 220 – 250V și, conform specificațiilor, conține un chip inteligent ce va optimiza consumul echipamentelor conectate la USB, prin recunoașterea lor.

În continuare, voi descrie arhitectura ce stă la baza proiectului, modul în care componentele comunică între ele și procesează informații pe baza cererilor venite din partea utilizatorului. În vederea realizării acestui lucru, am creat următoarea diagramă ce ilustrează fluxul de date dintre componente.



Figură 3.1 Fluxul de date

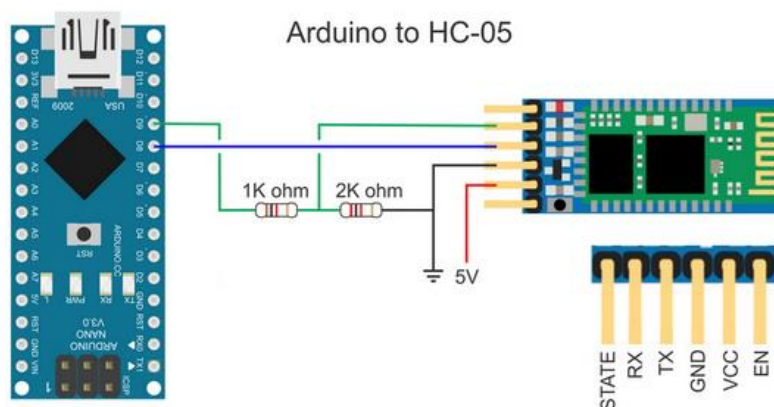
3.1.1.1 Configurare module Bluetooth

Pentru configurarea celor două module Bluetooth HC-05 a fost nevoie de o conectare a lor cu ieșirea USB a laptopului prin intermediul unui modul de comunicație FT232 ce realizează transmiterea serială a datelor USB – UART („Universal Asynchronous Receiver/Transmitter”). Prin simpla conectare a modulului prin intermediul plăcii de dezvoltare, nu s-a putut accesa modul AT (modul de comandă al dispozitivului). În consecință, s-a folosit următorul modul cu configurația următoare de conexiuni între pini:

Pe baza informațiilor găsite la [16], am conectat pinul TX de pe modul Bluetooth cu pinul RX de pe modulul FT232 și viceversa.

Pentru această comunicare, am utilizat mediul de programare Arduino IDE care pune la dispoziție o particularitate utilă și anume *Serial Monitor*. Prin intermediul acestuia se poate realiza conexiunea directă cu modulul Bluetooth, trimițându-se comanda „AT”. Dacă totul este în regulă, ar trebui ca în cadrul interfeței grafice Serial Monitor să apară mesajul „OK”. Mai departe, trimițând comanda „AT+HELP”, vom primi ca mesaj returnat o listă cu toate comenzile disponibile pentru programarea modulului Bluetooth. Pe baza acestora, am configurat modulele atribuindu-le roluri de Master și Slave și am setat viteza de comunicare BAUD la 9600. În cadrul proiectului, am programat modulele în așa fel încât să se conecteze automat atunci când primesc alimentare cu energie, pentru a evita necesitatea programării lor în repetate rânduri. Acest lucru a fost posibil prin utilizarea comenzii „AT+BAUD1”, care necesită o adresă a modulului a cărui conexiune este căutată, adresă găsită pentru fiecare modul în parte prin intermediul comenzii „AT+ADDR”. Modulele utilizate de mine au următoarele adrese asiguate: 98:84:E3:2D:6E:C0 (pentru modulul Master) și 98:84:E3:2D:59:69 (pentru modulul Slave).

Conectarea modulului Bluetooth la placa de dezvoltare se poate realiza direct prin legarea încrucișată a pinilor RX, respectiv TX, însă se specifică pe modul faptul că este nevoie de un nivel de tensiune de 3.3V pentru funcționarea optimă. În consecință, am realizat un divizor de tensiune pentru a crea un raport de reducere a tensiunii. Pe baza informațiilor asimilate din [17], am creat un divizor rezistiv format din conectarea în serie a două rezistențe de valori 1Kohm și 2Kohmi, conform următoarei scheme:



Figură 3.2 Divizor de tensiune pentru conectarea modulului Bluetooth

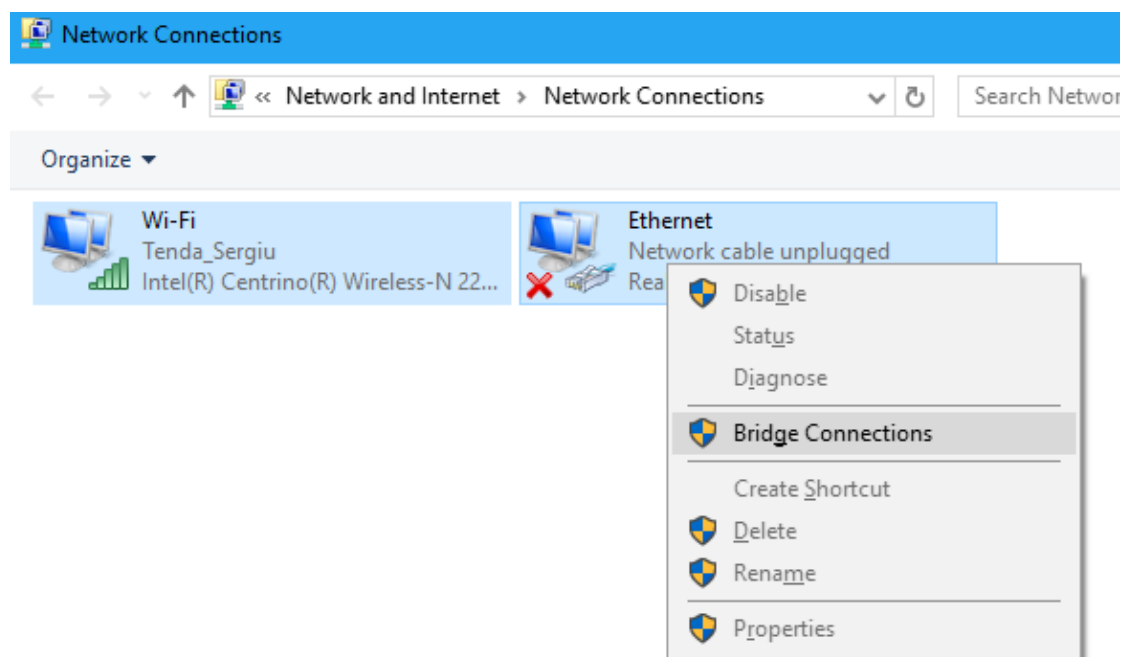
Astfel, tensiunea ce va ajunge trece prin pinul RX al modulului Bluetooth va fi redusă de la 5V, cât generează Arduino NANO, la aproximativ 3.3V prin intermediul formulei:

$$k = \frac{R2}{R1 + R2} \quad 3.1$$

Unde k reprezintă raportul de reducere a tensiunilor, R2 este rezistența de 2KOhm, iar R1 este rezistența de 1KOhm. Conform [7], un divizor de tensiune este compus din două sau mai multe rezistoare ce vor fi conectate în serie, iar ca raportul de reducere să nu fie influențat de temperatură, rezistențele sunt fabricate din materiale ce au coeficientul de variație a rezistivității cu temperatura neglijabilă.

3.1.1.2 Alimentare și conectare la internet

În momentul în care utilizatorul dorește să monitorizeze consumul unui dispozitiv sau să aibă control de la distanță asupra prizei și automat asupra alimentării dispozitivului, va alimenta cele două module hardware, neținându-se cont de ordine. Trebuie să se țină cont de faptul că modulul Slave poate fi alimentat direct de la priză prin intermediul ieșirii de 2.1 Amperi disponibilă. Acest lucru va duce la o eroare neglijabilă în măsurare, din cauza consumului adăugat al plăcii de dezvoltare care va alimenta la rândul ei modulul Bluetooth, displayul LCD și releul. Pentru a se realiza interfațarea cu pagina web de control și monitorizare, trebuie neapărat ca următorul pas să fie realizat și anume cel de conectare a modulului Ethernet fie la un router deja configurat la rețeaua de internet, fie la un laptop care va realiza o conexiune distribuită la ieșirea ethernet. Acest ultim aspect poate fi realizat în cadrul sistemului de operare Windows 10, urmărind următorii pași: se accesează meniul de setări pentru internet (Network & Internet Settings), iar din ramura „Status” se accesează opțiune de modificare a setărilor adaptorului („Change adapter option”). Acest pas va duce la deschiderea unei ferestre noi unde vor fi afișate conexiunile disponibile.



Figură 3.3 Distribuirea rețelei hotspot prin portul ethernet

În cazul de față, se vor selecta cele două conexiuni disponibile (conexiunea Wi-Fi printr-un routerul configurat și modulul Ethernet al laptopului interfațat prin driverul corespunzător), și se va apăsa butonul dreapta al mouse-ului. Astfel, vor apărea opțiunile ilustrate mai sus, dintre care o vom alege pe cea descrisă ca fiind „Bridge Connections”. În acest moment, o a treia pictogramă va apărea în cadrul ferestrei ce va reprezenta liantul dintre conexiunea la internet prin router și ieșirea ethernet corespunzătoare a laptopului. Astfel, modulul Master reprezentat prin placa de dezvoltare Arduino Uno Rev 3 va fi conectat la internet și va fi capabil să trimită datele venite de la modulul Slave mai departe către server.

În funcție de modalitatea de conectare la internet, fie prin conexiune directă la router, fie prin modul descris anterior, modulul Master va fi programat în mediul Arduino IDE, atribuindu-se o adresă IP specifică.

În acest moment, cele două module Bluetooth vor realiza o conexiune între ele automat și vor începe comunicarea cu serverul. În acest moment, utilizatorul poate decide dacă vrea să înceapă monitorizarea consumului unui dispozitiv, ori să aibă posibilitatea controlului de la distanță asupra prizei. În oricare dintre cazuri, va fi nevoie să acceseze pagina web care va găzdui toate opțiunile disponibile de informare și control. Mai exact, utilizatorul va accesa următorul IP: 92.188.164.20, la care se regăsește întreg setul de servicii PHP create și care comunică cu o bază de date.

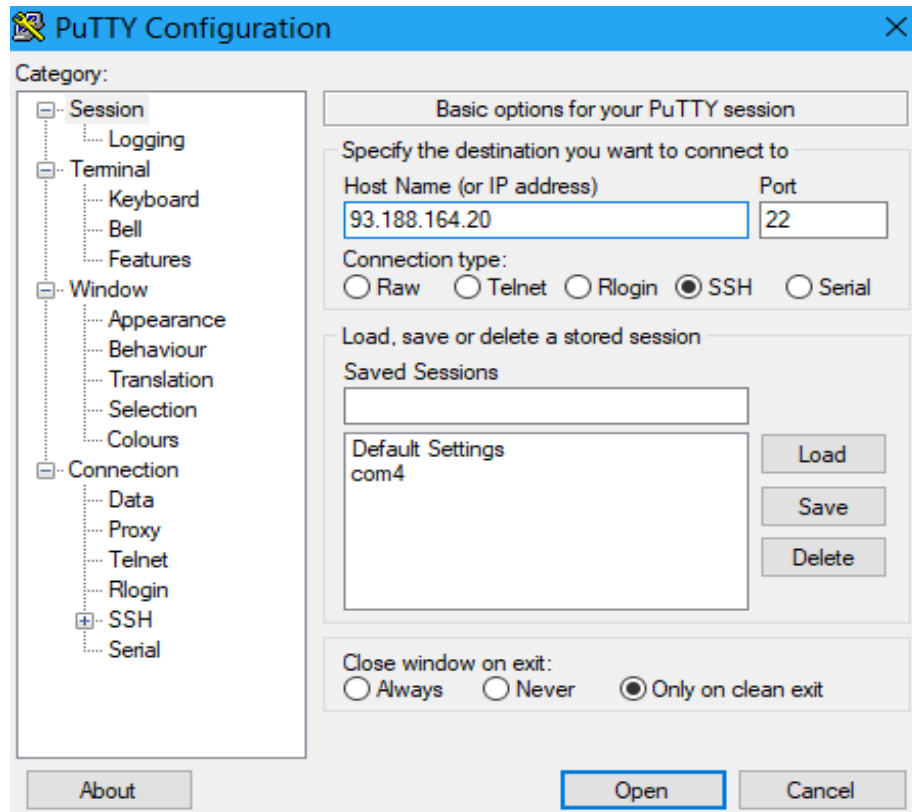
3.1.2 Serverul

3.1.2.1 Găzduire online

Pentru implementarea acestui proiect, am apelat la serviciile oferite de către Hostinger. Hostinger pune la dispoziție servicii online de găzduire, iar pentru necesitățile proiectului am achiziționat un server oferit la prețul de 5 dolari pe lună. Pe baza informațiilor regăsite în [18] (xiv), pentru cunoașterea funcționării proiectului, este nevoie de știut că server nostru este un calculator gazdă, capabil să trimită pagini web către utilizatorii care le solicită. Aceasta se realizează prin intermediul unui browser web. În ceea ce privește specificațiile serverului achiziționat, acesta este echipat cu un procesor single core ce lucrează la o viteză de 2.4Ghz, are o memorie de 1GB, un spațiu de lucru de 20Gb și o lățime de bandă de 1000Gb.

Totuși, doar având la dispoziție serverul, nu este de ajuns pentru a îndeplini funcționalitățile sistemului nostru. Pentru aceasta, urmând pașii descriși în [19], cei de la Hostinger pun la dispoziție toate cele necesare pentru a instala în cadrul serverului pachetul LAMP (Linux, Apache, MySQL și PHP).

Pentru început, este necesară accesarea serverului prin intermediul protocolului SSH [20]. Acest lucru este posibil folosind aplicația client „PuTTY”, despre care se găsesc mai multe informații accesând [21].



Figură 3.4 Client PuTTY

În cadrul acestei aplicații se va deschide fereastra de mai sus, unde va fi nevoie ca în câmpurile Host Name și Port să fie completate date 93.188.164.20 și 22, adică adresa IP asignată de către cei de la Hostinger serverului achiziționat și numărul portului prin care se realizează comunicarea. Toate aceste informații, incluzând detalii despre specificațiile și configurările serverului se găsesc pe pagina de utilizator a celor de la Hostinger.


SERVER MANAGEMENT

ACTIVITY LOG

STATISTICS

TUTORIALS

Server Status

 **Running**

Stop

Reboot

CPU Cores	1
Total CPU Speed	2.4Ghz
Memory	1Gb
Disk Space	20Gb
Bandwidth	1000Gb
Plan: Plan 1	<div>Upgrade</div>
Expires At: 2017-07-18	<div>Renew</div>

Recovery mode ?

Activate

Snapshot ?

Create

SSH Details

SSH IP: 93.188.164.20

SSH Port: 22

SSH Username: root

We recommend using dedicated offline SSH client to connect to your server. The most popular SSH client for windows is PuTTY. You may download PuTTY via this link:

[PuTTY](#)

If you are using MacOS or Linux, you do not need to download any additional software. You may connect to your Server via the Terminal with this command:

ssh root@93.188.164.20

Other settings

Hostname ?

sergiubalauca.xyz

Save changes

Operating system ?

CentOS 7 64bit

Save changes

Figură 3.5 Detalii SSH ale serverului

În urma deschiderii sesiunii în cadrul aplicației PuTTY, următoarea fereastră va apărea, unde vom fi nevoiți să introducem datele de autentificare pentru server, adică numele SSH ilustrat mai sus și parola setată de utilizator.



Figură 3.6 Autentificare PuTTY

În continuare, trebuie urmați cu rigurozitate pașii descriși în pagina [19], începând cu instalarea softului HTTP server Apache. În terminalul PuTTY vom introduce următoarele linii de comenzi:

```
yum install httpd -y  
systemctl start httpd.service  
systemctl start httpd.service  
systemctl start enable.service
```

Pasul următorul este reprezentat de instalarea serviciului de baze de date MariaDB, compatibil cu serviciul MySQL. Pentru aceasta, trebuie introduse pe rând următoarele linii de comandă în terminalul PuTTY:

```
yum install mariadb-server mariadb -y  
systemctl start mariadb  
systemctl enable mariadb
```

Următoarea linie de comandă este necesară pentru setarea unor măsuri de securitate, cerând utilizatorului setarea unei parole. Parola setată pentru acest proiect este „password”.

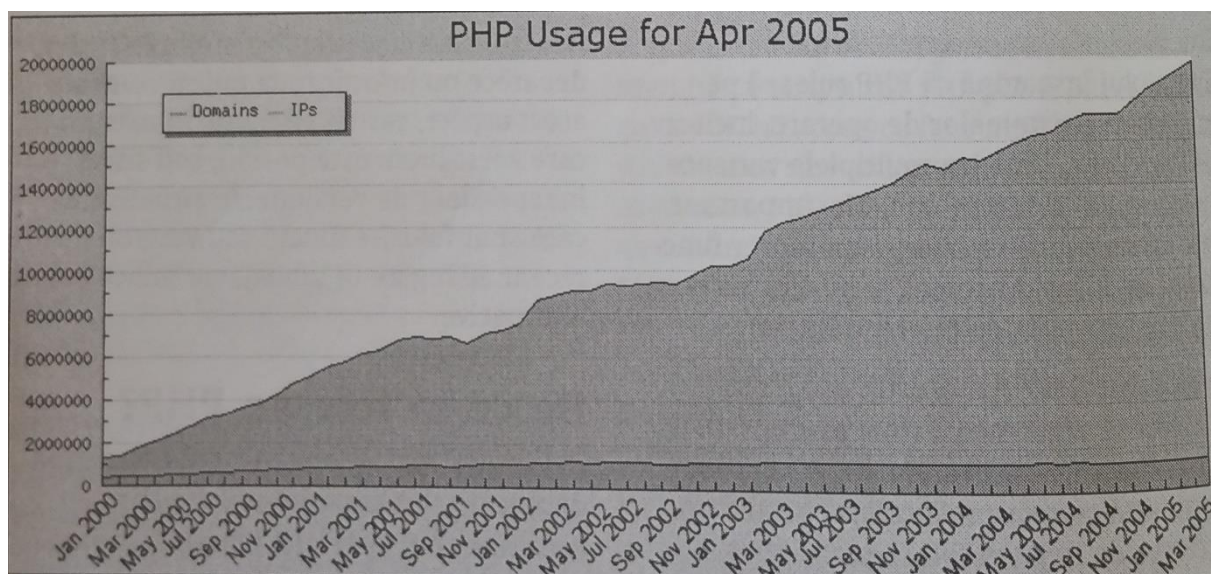
```
mysql_secure_installation
```

Ultimul pas de urmărit este acela de a instala PHP („Hypertext Preprocessor”), limbaj de scriptare datat încă din anul 1994.

```
yum install php php-mysql -y  
systemctl restart httpd.service
```

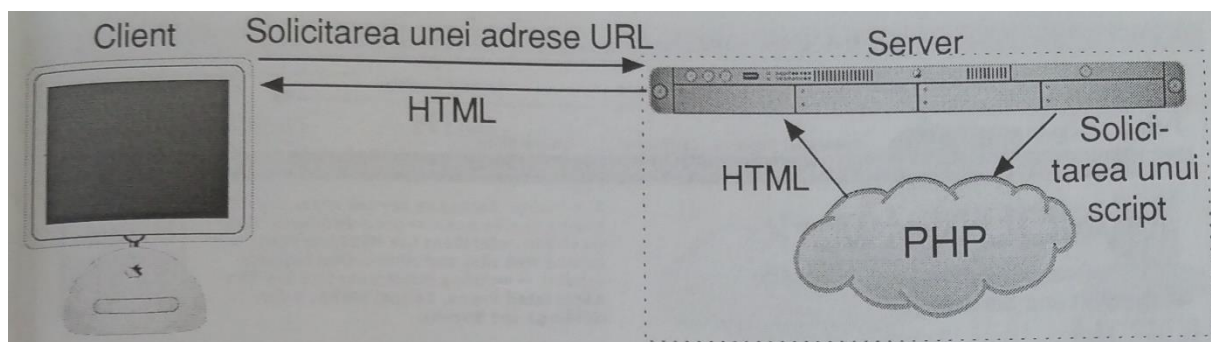
PHP este o tehnologie pe partea de server, un limbaj de scriptare înglobat în html, așa cum este descris în [18]. De aceea, în cadrul proiectului s-a ales această abordare, având în vedere necesitatea abordării unei tehnologii care să poată realiza conexiunea cu baza de date, dar să și permită editarea designului interfeței web. Datorii caracteristicii de lucru interplatformă a limbajului, scripturile PHP ce vor fi rulate pe serverul pus la dispoziție de Hostinger vor putea fi accesate de pe majoritatea sistemelor de operare și vor lucra și pe alte servere.

Există și alte variante similare acestei tehnologii, cum ar fi ASP, însă am ales această abordare datorită faptului că este ușor de învățat și oferă performanțe ridicate. Totodată, conform sursei [18](xiv), este unul din cele mai folosite limbaje de scriptare, această afirmație fiind însoțită de următoarea ilustrație:



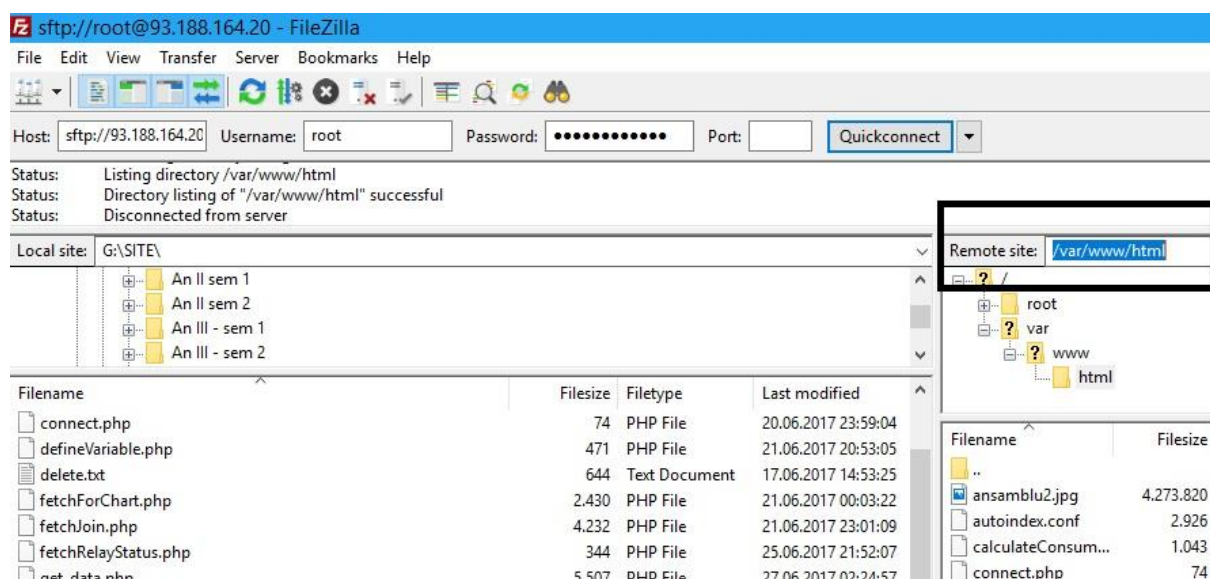
Figură 3.7 Rata utilizării PHP

Funcționarea acestui limbaj pe server, care trimite mai departe paginile web pe care un utilizator le cere. Când se realizează această solicitare, serverul va citi codul PHP, urmând prelucrarea corespunzătoare liniile de cod. Această tratare din partea serverului nu va ține cont de formatul fișierului.



Figură 3.8 Solicitări client – server

Pentru a putea realiza transferul fișierelor script ce vor sta la baza interfeței web, se va utiliza o aplicație FTP (File Transfer Protocol), ce va permite salvarea fișierelor la adresa serverului nostru. Pentru acest proiect, am folosit aplicația numită „Filezilla”. Pe baza informațiilor de la [19], se face realizarea accesului către directorul din care serverul va prelucra fișierele, mai exact la adresa „/var/www/html”.



Figură 3.9 Aplicație FTP (FileZilla)

Pentru scrierea liniilor de cod script am folosit editorul de text Notepad++, iar pentru editarea liniilor de cod în format C++ am folosit mediul de programare Arduino IDE ce pune la dispoziție o serie de exemple pe baza cărora pot fi programate plăcuțele de dezvoltare Arduino.

3.1.2.2 Baza de date

O importantă componentă a întregului ansamblu este baza de date, ca și componentă software. Aceasta se află pe serverul localizat la adresa 93.188.164.20 și poate fi accesată de la distanță, caracteristică folosită atât la citirea, scrierea și afișarea datelor în interfața grafică, cât și la scrierea datelor de către clientul web configurat pe placa de dezvoltare Uno Rev 3.

Baza de date de tip server utilizată este MariaDB, fiind gestionată prin intermediul sistemului MySQL, un sistem de gestionare a bazelor de date relaționale. În ceea ce privește arhitectura bazei de date utilizate în acest proiect, s-a optat pentru crearea a patru tabele, numite „RelayStatus”, „device”, „sensor” și „users”. Acestea vor stoca date referitoare la utilizatorii aplicației, consumatorii meniți a li se monitoriza consumul, date despre starea releului, cât și date achiziționate de la senzorul de curent. Dacă se opta pentru utilizarea unui server creat local prin intermediul aplicației XAMP, spre exemplu, se putea utiliza interfața grafică disponibilă local la adresa „localhost/phpmyadmin”. În cazul nostru, cum serverul este unul la distanță, accesarea lui se face, cum am spus mai sus, prin intermediul unui terminal de comandă, PuTTY, care va deschide o sesiune de lucru SSH. Programul nu are o interfață grafică bogată, însă, conform [18], însă este un instrument fiabil, ce nu creează dificultăți în utilizare. Pe lângă aceasta, intervine și motivul personal pentru care am aborda această variantă și anume acela că am exersat sintaxa MySQL.

După autentificarea la server pe baza datelor de autentificare, programul va accepta o listă de argumente pe baza cărora se poate face accesul la serviciul MySQL:

„mysql -u username -p -h hostname” [18]

Parametrul „-p” va determina cererea unei parole din partea utilizatorului, aceasta putând fi specificată direct în linia de comandă, după parametru. Totuși, în cazul acesta, parola va fi vizibilă, fapt nedorit. Parametrul „-h” este opțional și de aceea, în cazul nostru, este de ajuns să inserăm ca linie de comandă următoarea listă de parametri:

„mysql -u root -p”

Înainte de a crea tabelele necesare, primul pas este de a crea o bază de date, pe care am numit-o „test”. Acest lucru a fost posibil prin introducerea următoarei linii de comandă: create „database test;”. Pentru a putea săvârși operațiuni în cadrul acestei baze de date, mai întâi trebuie selectată prin comanda: „use test;”. În acest moment se poate începe procedura de creare a tabelelor necesare.

3.1.2.2.1 Tabela „RelayStatus”

Acest tabel va conține starea releului, variabilă necesară pentru accesarea ulterioară din interfața web pentru modificare în funcție de preferința utilizatorului de a închide sau deschide priza și pentru accesare de pe placa Arduino Uno ce va acționa ca și client. Crearea tabelului s-a efectuat prin utilizarea comenzii următoare:

```
„CREATE TABLE RelayStatus(
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
status VARCHAR(15) NOT NULL,
reg_date TIMESTAMP);
```

Câmpul „id” este reprezentativ tabelului, el ajută la identificarea acestuia prin atribuirea caracteristicii de cheie primară. Aceasta nu se va schimba niciodată și va fi unică pentru fiecare înregistrare din tabel. Câmpul „status” este cel necesar în cadrul tabelului, el permițând a i se atribui valori de tip varchar și nu poate fi nulă. Vom vedea mai departe că este necesară atribuirea numai a două valori și anume „A” sau „F”. Câmpul „reg_date” este adăugat cu scop informativ, în cazul în care se dorește o ulterioară dezvoltare a unei monitorizări a schimbărilor de stare a releului.

3.1.2.2.2 Tabele device

Acest tabel este utilizat pentru monitorizarea dispozitivelor cărora li se va măsura consumul de energie, el conținând o serie de parametri reprezentativi. Structura tabelului este prezentată în figura următoare:

Field	Type	Null	Key	Default	Extra
id	int(6) unsigned	NO	PRI	NULL	auto_increment
description	varchar(30)	NO		NULL	
consumption	varchar(30)	YES		NULL	
cost	varchar(30)	YES		NULL	
reg_date	timestamp	NO		CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP

Figură 3.10 Tabelul „device”

La fel ca în cazul tabelului RelayStatus, vom avea un câmp numit „id”, necesar identificării tabelului și un câmp numit „reg_date” necesar în cadrul implementării calculului costurilor. Un alt parametru creat este „description”, acesta purtând o scurtă descriere a consumatorului, asignată de utilizator în momentul începerii citirii de la senzor. Pe lângă aceștia, am creat încă doi parametri ce vor primi valorile calculate ale consumului și costului în momentul încetării înregistrării.

3.1.2.2.3 Tabela sensor

Acest tabel reprezintă legătura dintre server și clientul Arduino, el înmagazinând înregistrările venite de la senzorul de curent. Ar fi fost de ajuns crearea unui singur câmp reprezentativ, însă am optat pentru următoarea structură:

Field	Type	Null	Key	Default	Extra
id	int(6)	NO	PRI	NULL	auto_increment
time	timestamp	NO		CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP
value	double	YES		NULL	
value2	double	YES		NULL	
ID_Device	int(11)	YES		NULL	

Figură 3.11 Tabelul "sensor"

Același câmp numit „id” se păstrează ca și cheie primară, însă tipul TIMESTAMP l-am asignat de data aceasta unei variabile numite „time”. Totodată, am creat un câmp numit „ID_Device”, ca și cheie străină pentru tabelul „device”. Am făcut acest lucru în ideea monitorizării datelor din baza de date. Câmpurile principale din tabel sunt „value” și „value2”, care reprezintă valori venite de la senzor ale curentului, respectiv tensiunii.

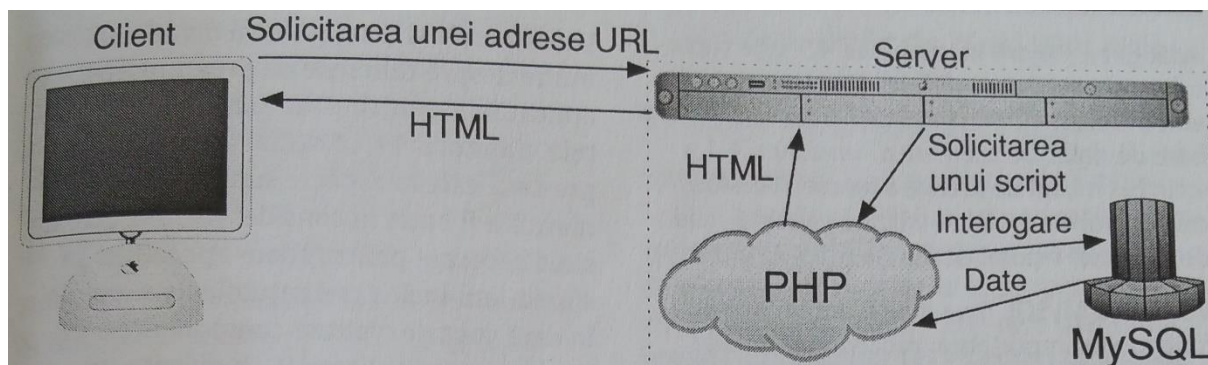
3.1.2.2.4 Tabela users

Această ultimă tabelă are rolul de a păstra date despre utilizatorii ce pot accesa pagina web, totodată creând posibilitatea realizării unui nivel minim de securitate prin dezvoltarea unui sistem de autentificare ce necesită aceste informații. Tabelul are o structură simplă, ca în figura următoare:

Field	Type	Null	Key	Default	Extra
id	int(11) unsigned	NO	PRI	NULL	auto_increment
username	varchar(100)	NO		NULL	
password	varchar(30)	NO		NULL	
reg_date	timestamp	NO		CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP

Figură 3.12 Tabelul "users"

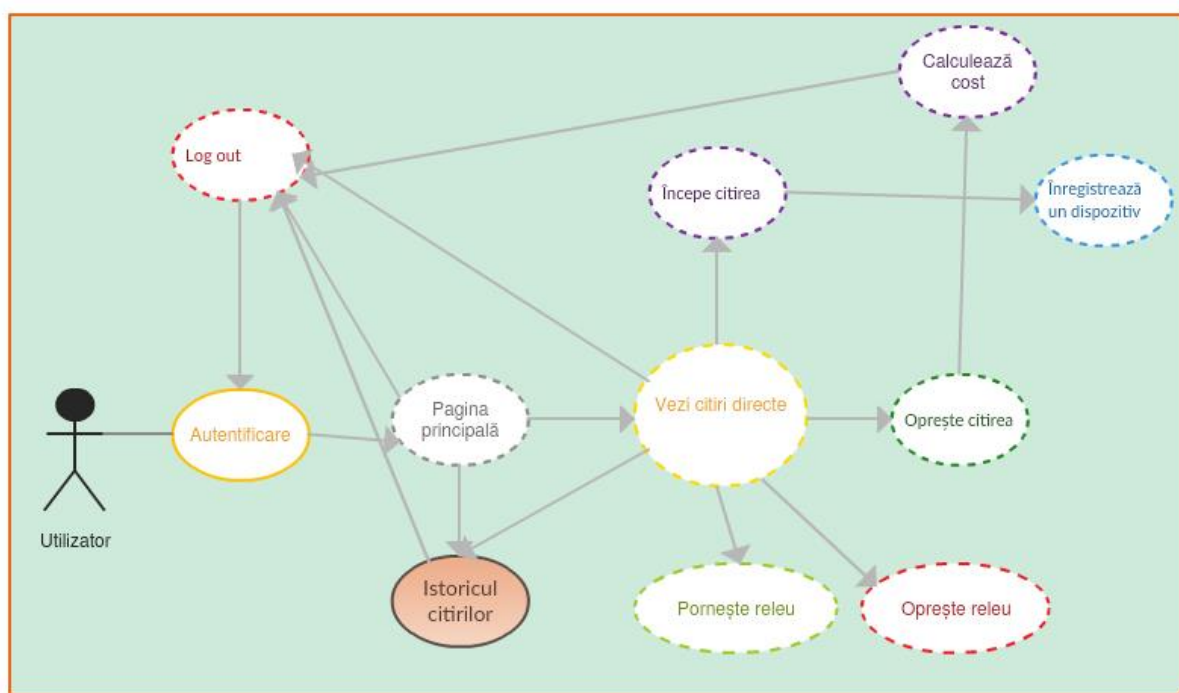
În continuare, este ilustrată o simplă manieră de comunicație dintre componentele unui sistem client-server bazat pe servicii PHP și un sistem MySQL.



Figură 3.13 Comunicare client - server bazată pe servicii PHP, MySQL

3.1.2.3 Servicii PHP

În diagrama de scenarii prezentată în continuare sunt înfățișate principalele funcții îndeplinite de fișierele PHP stocate pe server și modul în care interacționează.



Figură 3.14 Diagrama de scenarii

3.1.2.4 Autentificare

Pentru început, utilizatorului aplicației i se vor cere datele necesare autentificării, mai exact numele de utilizator și parola necesară. Dacă acestea sunt invalide, va fi afișat un prompt care îl va informa că încearcă să acceseze pagina cu un cont invalid. Altfel, va fi direcționat către pagina principală. Datele de autentificare nu sunt sensibile la majuscule, iar opțiune de înregistrare a unui nou client nu există, în acest fel întărindu-se nivelul de securitate. Singura posibilitate de creare a unui nou utilizator este prin intermediul terminalului PuTTY și a liniilor de comandă MySQL.

3.1.2.5 Pagina principală

În cadrul acestei pagini, utilizatorul este întâmpinat printr-o scurtă prezentare a proiectului, împreună cu un carusel de imagini descriptive. Această pagină reprezintă punctul de plecare al interfeței web din care se poate ajunge mai departe la pagina de citiri ale datelor de la senzor, la pagina cu istoricul citirilor sau se poate opta pentru deconectare.

3.1.2.6 Pagina de citiri

Aceasta este pagina care pune la dispoziția utilizatorului un set mai complex de opțiuni de prelucrare și control, mai exact de începere a citirii consumului unui dispozitiv, respectiv încetare și totodată de control asupra releului ce acționează priza. În cadrul paginii este afișat un tabel ce conține ultimele cincisprezece citiri de la senzor. Această caracteristică este una cu scop strict informativ, utilizatorul putând vedea în direct fluctuațiile de consum. Acest tabel se actualizează automat, fără a afecta conținutul întregii pagini. Tot în cadrul paginii sunt afișate două opțiuni de control al releului. Una dintre ele pune la dispoziție două butoane de tip comutator, iar cealaltă cere scrierea unui caracter „A” sau „F”, opțiuni ce vor activa sau dezactiva releul.

În momentul în care se dorește începerea înregistrării, utilizatorul trebuie să apese pe butonul de începere a citirii, acțiune ce va genera un prompt ce va cere atribuirea unui nume consumatorului, cu scopul unei bune gestionări a istoricului. Când se dorește încetarea înregistrării, este nevoie de apăsarea butonului corespunzător, acțiune ce va declanșa de asemenea un prompt ce va afișa numele ultimului dispozitiv înregistrat. După ce se acceptă acțiunea, utilizatorul va fi direcționat către următoarea pagină de calcul al costului și consumului. Nu în ultimul rând, din această pagină se poate ajunge înapoi către pagina principală, către pagina de istoric al citirilor sau se poate opta pentru deconectare.

3.1.2.7 Pagina de rezultate

Trebuie menționat faptul că aici se poate ajunge numai în urma încetării acțiunii de înregistrare a consumului, prin apăsarea butonului corespunzător și a confirmării acțiunii. Această pagină va afișa utilizatorului un set de date referitoare la consumului dispozitivului și costul acestuia direct tradus în lei. Totodată, sub acest set de date este afișat un tabel cu citirile de la senzor cu același scop informativ. Tabelul se extinde în josul paginii, în funcție de durata înregistrării. Totodată, același set de opțiuni de navigare este pus la dispoziția utilizator, în plus fiind butonul de navigare înapoi către pagina de citiri.

3.1.2.8 Pagina de istoric

Această pagină este una creată în scop pur informativ, în cadrul căreia este afișat un grafic ce conține ultimele seturi de înregistrări realizate. Totuși, tabelul este vizibil numai după ce se încetează înregistrarea pentru ultimul dispozitiv înregistrat. În aceeași manieră, se poate naviga către pagina principală, cea de citiri sau se poate alege varianta deconectării, acțiune ce va trimite către pagina de autentificare.

3.2 Implementare

Pentru a se ajunge la interacțiunea dintre componentele fizice și interfața grafică, plăcile de dezvoltare Arduino, inclusiv modulele adiacente au nevoie să ruleze persistent un cod dezvoltat în mediul Arduino IDE pe baza limbajului de programare C++.

3.2.1 Fișiere Arduino cu extensia .ino

3.2.1.1 Funcția de prelucrare a datelor de la senzor

Pentru început, s-a urmărit crearea unei funcții de prelucrare a informațiilor citite de la senzorul de curent, deoarece prin intermediul funcției simple `analogRead()` va rezulta o fluctuație în jurul valorii 508, declarată ca fiind variabila „`adc_zero`”. Pentru prelucrarea acestei valori, am creat un fișier nou prin intermediul mediului de programare Arduino IDE și am început prin a crea o funcție numită „`CurrentSense()`”. Am decis ca tipul returnat de această funcție „`float`”, tip care permite stocarea unui număr mare de zecimale, fapt ce conduce la o precizie mai mare a calculelor. Pentru implementarea funcției, am luat în calcul noțiunile dobândite citind [4] și utilizând informații găsite pe forumul deschis al celor de la Arduino [22]. Pentru translatarea valorii fluctuante de 510 într-o citire corespunzătoare a curentului alternativ, se realizează achiziția de un număr mare de date la intervale regulate și pe o perioadă de timp ce reprezintă un multiplu exact al perioadei. În continuare, se însumează citirile individuale, urmând ca din această sumă să obținem media și ulterior radicalul. Intervalele regulate pot fi obținute prin intermediul funcției „`micros()`” care nu face altceva decât să numere microsecundele din momentul începe execuția programului. Așa cum este menționat la [23], funcția va reveni în punctul de zero după trecerea a aproximativ 70 de minute, iar în cazul plăcii Nano, care se bazează pe funcționarea unui procesor la frecvența de 16Mhz, funcția va avea o rezoluție de patru microsecunde (valoarea rezultată va fi întotdeauna multiplu de patru).

```

float CurrentSense()
{
    unsigned long currentAcc = 0;
    unsigned int count = 0;
    unsigned long prevMicros = micros() - sampleInterval ;
    while (count < numSamples)
    {
        if (micros() - prevMicros >= sampleInterval)
        {
            int adc_raw = analogRead(currentPin) - adc_zero;
            currentAcc += (unsigned long) (adc_raw * adc_raw);
            ++count;
            prevMicros += sampleInterval;
        }
    }
    float rms = sqrt((float)currentAcc / (float)numSamples) * (75.75 / 1024.0);
    rms = rms - 0.05;
    if (rms < 0.01789)
    {
        rms = 0;
    }
    return rms;
}

```

Figură 3.15 Cod pentru funcția de prelucrare a datelor citite de sensor

Pentru funcționarea acestui cod, am definit global ca pin de intrare pentru ieșirea senzorului de curent pinul A5 prin sintaxa: „const int currentPin = A5;”. Totdată, a fost nevoie de definirea unui set de variabile, precum: „sampleTime” ca fiind 100000UL, ceea ce reprezintă numărul exact de cicluri pentru frecvența de 50Hz, „numSamples” ca fiind egală cu 250UL, valoare necesară a fi potrivită pentru împărțirea exactă a numărului de cicluri, dar mică îndeajuns pentru Convertorul Digital Analog, așa încât să țină pasul cu prelucrarea [22]. Valoarea 1024 reprezintă rezoluția de 2^{10} a convertorului analog-digital. Valoarea 75.75 este obținută prin calcularea unei ecuații simple ce implică sensibilitatea senzorului de 66mV/A, valoarea specificată în fișa tehnică a produsului și tensiunea de alimentare a senzorului de 5V, tensiune continuă. Cum 0.066V sunt specifici unui Amper, va reieși faptul că pentru 5V este specifică valoarea 75.75. În finalul secvenței de cod ilustrate mai sus este realizată o verificare bazată pe o condiție pentru calibrarea senzorului și evitarea afișării pe display a unei valori parazite fluctuante. Astfel, s-a creat valoarea limită inferioară de măsurare de 0.02 Amperi, îndeajuns de mică pentru ca ansamblul să detecteze consumatori mici, cum ar fi un telefon conectat pentru reîncărcarea bateriei.

Această funcție descrisă anterior a fost creată pentru a fi apelată în interiorul funcției „loop()”, a cărei prezență este specifică fișierelor dezvoltate în mediul Arduino IDE.

```

void loop() {
    Current = CurrentSense();
}

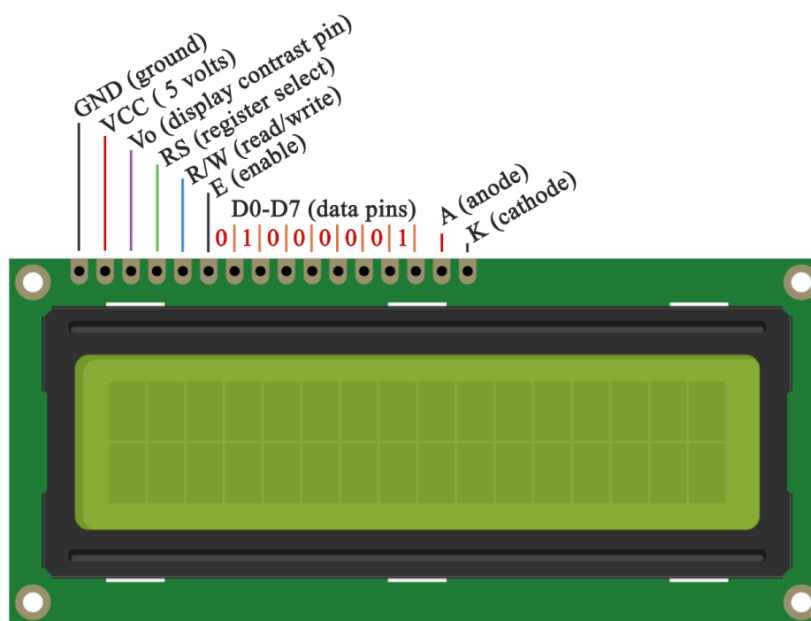
```

Figură 3.16 Apelul funcției de citire curent

Am asignat valoarea returnată de funcția `CurrentSense()` unei variabile declarate ca fiind „Current”, valoare pe care o vom folosi ulterior la afișarea pe ecranul LCD și pentru trimiterea prin intermediul modului Bluetooth către placa de dezvoltare Arduino Uno Rev3.

3.2.1.2 Setarea displayului LCD

Pentru afișarea directă a datelor ce vor fi trimise către server, am optat pentru conectarea unui modul LCD 16x2 la placa de dezvoltare Arduino NANO. Acesta permite o interfațare ușoară și rapidă, conform informațiilor ce se găsesc la [24]. Pentru aceasta, am alimentat modulul prin intermediul primilor doi pini, și ulterior am folosit șase din pinii digitali disponibili pe placă, 4 dintre care i-am conectat la regiștrii displayului. Pinul E (enable) l-am conectat la pinul 2 digital, pinul RS (selectare registru) la pinul 1, adică ieșirea TX, iar pinul R/W (citire/scriere) l-am conectat la masă.



Figură 3.17 Configurare Display LCD

În cadrul displayului vom afișa două valori și anume cea a curentului returnată de funcția de prelucrare `CurrentSense()` și cea a puterii, aproximată prin formula $\text{Current} \times 220$. În cadrul fișierului Arduino creat, pentru configurarea displayului s-a utilizat librăria `<LiquidCrystal.h>` accesibilă gratuit din mediul Arduino IDE și s-au efectuat setări conform configurărilor fizice prezentate anterior:

```
LiquidCrystal lcd(8, 2, 4, 5, 6, 7);
```

Figură 3.18 Setare pentru displayul LCD

3.2.1.3 Comunicarea serială

Pentru trimiterea datelor și comenzilor dintre cele două plăci de dezvoltare, am folosit funcția „`Serial.available()`”. Folosită în fișierul încărcat pe Arduino NANO, aceasta va ajuta la trimiterea datelor achiziționate de la senzor către modulul Bluetooth Master și

totodată va permite citirea comenzilor primite de la acesta pentru acționarea releului. Când este utilizată pe fișierul de pe placa Arduino UNO, această funcție este utilizată în următorul mod:

```
if (Serial.available())
{
  while (Serial.available())
  {
    commandbuffer[i++] = Serial.read();
  }
  a = atof(commandbuffer);
  if (i > 0)
    Serial.println((char*)commandbuffer);
}
```

Figură 3.19 Comunicare serială

Buclo „while(Serial.available())” este utilizată pentru citirea constantă a datelor ce sunt transmise de la modulul Bluetooth Slave, utilizându-se o variabilă de tip șir de caractere („char commandbuffer[100]”). Ulterior salvării datelor în cadrul acestei variabile, am folosit funcția „atoi”, convertind variabila într-o altă variabilă de tip „float”. Aceasta este varianta pe care am implementat-o, în urma analizării informațiilor găsite la [25].

3.2.1.4 Conectare la server

Conectarea la serverul achiziționat de la Hostinger presupune crearea unui client web. Acest lucru l-am realizat bazându-mă pe exemplele puse la dispoziție de mediul de programare Arduino IDE, urmând următoarea cale de acces: File->Examples->Ethernet->Web Client. Acest fișier l-am adaptat conform cerințelor funcționale ale proiectului.

Pentru început, am asignat o adresă IP de clasă C modulului Ethernet conectat la placă, în funcție de conexiunea la internet. Dacă este cazul conectării la un router configurat acasă, se folosește adresa 192.168.0.17, iar dacă utilizăm rețeaua „hotspot” creată de telefon, asignăm ca adresă IP 192.168.43.17. După săvârșirea acestei acțiuni, trebuie creată o variabilă pentru descrierea serverului la care dorim să ne conectăm. În cazul nostru, am definit o variabilă de tip șir de caractere „char server[]” căreia i-am asignat ca valoare adresa IP a serverului achiziționat de la Hostinger (93.188.164.20).

Următorii pași urmăți sunt declararea unui client Ethernet prin intermediul sintaxei „EthernetClient client” și pornirea serviciului Ethernet în cadrul funcției „void setup()” prin sintaxa „Ethernet.begin(mac, ip)”, unde ip reprezintă adresa asignată anterior, iar mac reprezintă o adresă de tip MAC, care în cazul nostru este furnizată de către fișierul exemplu: byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED}.

Cum toate variabilele și setările necesare sunt pregătite, singurul pas de îndeplinit pentru comunicarea cu serverul este acela de a realiza conexiunea dintre client și server, prin intermediul sintaxei următoare:

```

if (client.connect(server, 80)) {
    client.print("GET /write_data.php?");
    client.print("value=");
    client.print(AmpsRMS);
    client.print("&value2=");
    client.print(Voltage2);
    client.println(" HTTP/1.1");
    client.println("Host: 93.188.164.20");
    client.println("Connection: close");
    client.println();
    client.println();
    client.stop()
}

```

Figură 3.20 Conexiune client Arduino - server

Conexiunea dintre client și server se bazează pe sintaxa „client.connect(server, 80)”, unde 80 reprezintă numărul portului de conexiune. Pentru acest proiect, am adaptat codul așa încât datele venite de la senzorul de curent prin intermediul comunicației Bluetooth să ajungă la server prin intermediul unor cereri GET și a unor fișiere PHP, a căror dezvoltare va fi prezentată în continuare. Așa cum se observă, trimiterea datelor către server se realizează prin intermediul funcției „print”, iar pentru situația în care dorim să citim date de la server, am folosit sintaxa: „client.read()”.

3.2.1.5 Acționarea Releului

Pentru acționarea releului am creat două variabile globale „RELAY_ON” și „RELAY_OFF”, cărora le-am atribuit valorile 0, respectiv 1 și o variabilă cu denumirea Relay_1 atribuită pinului digital cu numărul 2 de pe placa de dezvoltare.

Acționarea ca atare se realizează bazându-ne pe o valoare citită de la server și trimisă mai departe prin intermediul comunicației Bluetooth către placa Arduino NANO, mai exact „A” sau „F”. În funcție de aceasta, prin intermediul funcției „digitalWrite”, vom trimite semnale logice de „1” sau „0”.

```

if(c == 'A'){digitalWrite(Relay_1, RELAY_ON);}
if(c == 'F'){digitalWrite(Relay_1, RELAY_OFF);}

```

Figură 3.21 Acționarea releului

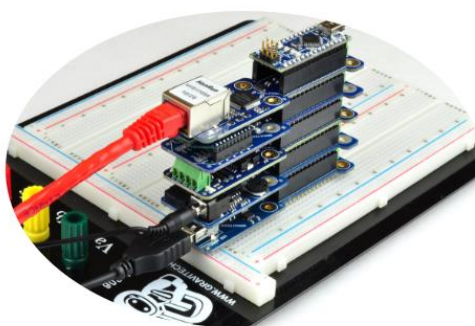
3.2.2 Fișiere PHP

Cealaltă componentă reprezentativă a proiectului este interfața grafică prin intermediul căreia se realizează interacțiunea cu componentele hardware și cu serverul unde sunt stocate toate datele relevante, precum citiri de la senzor, date despre utilizator, stările releului. Așa cum este prezentat în continuare, abordarea aleasă propune o compunere între limbaje web, de scriptare, precum PHP, JavaScript, CSS și HTML. Pentru parcurgerea tuturor serviciilor PHP, voi prezenta fișierele corespunzătoare în ordinea în care s-ar realiza navigarea în interfața web.

3.2.2.1 Autentificare

Prima pagină care îl întâmpină pe utilizator este cea de autentificare, unde îi sunt cerute datele necesare, mai exact numele de utilizator și parola.

Welcome



Username:

Password:

☐ Remember me

Figură 3.22 Pagina de autentificare

Trebuie menționat de la început că pentru stilizarea interfeței web am utilizat o bibliotecă Bootstrap ce vine împreună cu suport CSS și JavaScript. În cadrul fișierelor script am inclus-o prin intermediul unui CDN (Content Delivery Network).

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
```

Figură 3.23 Librării web

În cadrul acestei pagini am construi un formular HTML care va ilustra două câmpuri în care utilizatorul va introduce datele de autentificare. Pe baza celor regăsite în [18], se poate spune că gestionarea unui formular HTML în PHP descrie o acțiune structurată pe două etape. Mai întâi am creat formularul propriu-zis prin intermediul aplicației Notepad++ și ulterior am făcut prelucrarea datelor din formular prin intermediul scriptului PHP.

Un formular poate fi creat prin intermediul diferitelor controale, precum „action” sau „method”. În cazul de față, eticheta „action” este cea mai importantă, deoarece pe baza ei sunt transferate datele din formular către scriptul PHP. Al doilea atribut descrie felul în care se realizează transmiterea datelor, mai exact prin două opțiuni: GET sau POST. Mai precis, acestea din urmă se referă la cererea HTTP (Hypertext Transfer Protocol) efectuată. În cadrul aceleași referințe menționate anterior este descrisă diferența dintre metodele posibile, mai exact se spune că metoda GET ca avantaj posibilitatea de a marca adresa paginii în browser, putându-se astfel naviga înapoi sau a se reîmprospăta pagina. Dezavantajul este constituit în schimb prin limitarea cantității de date transmise și prin nivelul scăzut de securitate.

Între etichetele ce limitează formularul, am creat două suprafețe de text cu denumirile atent alese, în așa fel încât să nu existe ambiguități în momentul în care vor fi folosite mai departe în scriptul PHP. În interiorul suprafețelor sunt deja propuse niște

sugestii sub formă de text, acțiune cu scop informativ. Acest lucru a fost realizat prin intermediul atributului „placeholder” pus la dispoziție de biblioteca Bootstrap.

```
placeholder="Enter Username"
```

Figură 3.24 Sugestie în formular sub formă de text

Pentru a stiliza suplimentar pagina de autentificare, am utilizat o imagine pe care am modificat-o prin intermediul atributului „Avatar”.

O altă caracteristică implementată în cadrul interfeței web este modulul de sesiune. Acesta realizează suplinirea lipsei de informații referitoare la datele despre utilizator, în cazul de față, la efectuarea navigării între pagini. Pentru aceasta, între etichetele caracteristice unui script PHP am început cu apelarea funcției „session_start()”, care va efectua o cerere modulului de a crea o nouă sesiune. În continuare, am înregistrat valori în sesiune pentru numele de utilizator și parolă, conform denumirile date în formularul HTML, pentru verificarea existenței în baza de date și totodată pentru trimiterea mai departe a numelui utilizatorului în cadrul navigării pe paginile următoare.

```
$username = $_POST['username'];  
$password = $_POST['password'];
```

Figură 3.25 Setarea variabilelor de sesiune

Pentru verificarea existenței numelui de utilizator și a parolei în baza de date, este nevoie, pentru început, de o conexiune la baza de date. Acest lucru este posibil prin utilizarea funcției „mysqli_connect()” ce necesită ca parametri o listă ce include gazda unde este localizată baza de date, numele de utilizator și parola. În cazul nostru, gazda este 93.188.164.20, numele de utilizator „root”, parola „password”, însă am adăugat și numele bazei de date, lucru permis de către funcție. Altfel, selectarea bazei de date s-ar realiza prin intermediul funcției „mysqli_select_db(\$numele_bazei_de_date)”, care poate fi privită ca o alternativă a utilizării sintaxei „USE baza_de_date” [18], atunci când se lucrează în terminalul PuTTY. Această conexiune am realizat-o în fișierul „connect.php” pentru a separa entitățile și a reutiliza codul.

```
$con=mysqli_connect("93.188.164.20","root","password", "test");
```

Figură 3.26 Datele de conexiune pentru server și baza de date

După realizarea conexiunii, verificăm dacă în cadrul formularului HTML creat s-a efectuat apăsarea butonului „bttLogin” care este de tip „submit”, adică de trimitere a datelor trecute de utilizator în câmpurile de text. Aceasta se face prin intermediul sintaxei „if(isset(\$_POST['bttLogin']))”. Dacă se îndeplinește condiția, urmează verificarea existenței parametrilor de autentificare, acum că este stabilită și conexiunea la baza de date. Pentru aceasta, este folosită sintaxa MySQL, mai exact am executat o interogare prin funcția „mysqli_query()”, utilizând ca parametri conexiunea la baza de date declarată anterior și sintaxă MySQL sub forma:

```
$result = mysqli_query($con, 'select * from users where  
username = "'. $username. '" and password = "'. $password. '"');
```

Figură 3.27 Query pentru extragerea datelor de autentificare

Am selectat toate înregistrările din tabelul ce conține informațiile despre utilizatori și am comparat câmpurile „username” și „password” cu datele venite de la formular. Pentru verificarea unicității înregistrării am utilizat funcția „mysqli_num_rows(\$result)” [18], al cărui rezultat l-am comparat cu valoarea 1. Dacă această condiție este îndeplinită, atunci nu rămâne decât să se treacă la pasul următor, cel de navigare către următoarea pagină și totodată de trimitere a variabilei de sesiune descrisă prin numele de utilizator. Dacă datele de autentificare nu sunt corecte, va apărea un prompt în partea de sus a paginii care va informa utilizatorul asupra încercării de autentificare pe baza unor date invalide. Această funcționalitate este implementată utilizând limbajul JavaScript, mai exact o funcție simplă „alert('text')”. Partea atractivă și utilă a acestui limbaj este în special posibilitatea de a fi folosit în cadrul unui script PHP, fără a afecta funcționalitatea.

Trecerea către următoarea pagină este realizată prin următoarea sintaxă:

```
$_SESSION['username'] = $username;
header('Location: main2.php');
```

Figură 3.28 Sintaxă pentru trimiterea forțată către o adresă

3.2.2.2 Pagina principală

După autentificarea utilizatorului pe pagina index a serverului, acesta este întâmpinat cu un mesaj în următoarea pagină, numită „main2.php”. Pentru identificarea paginii în browser, am setat paginii titlul „Monitorizare consum”. Mesajul de întâmpinare reprezintă un mesaj simplu PHP compus din variabila de sesiune setată anterior, adică numele de utilizator.

```
session_start();
echo 'Welcome, ' . $_SESSION['username'];
```

Figură 3.29 Utilizarea variabilelor de sesiune

Tot în cadrul paginii am apelat la librăria Bootstrap despre care am precizat anterior, cu scopul de a crea un meniu de navigare stilizat. Navigarea se face simplu prin folosirea unui atribut HTML „href”, inclus într-o listă de valori, căruia îi este asignată pe rând adresa fișierului ce descrie o altă pagină. Pentru blocarea accesului la pagina de calcule a costului, am asignat atributului valoarea „#”. Pe lângă posibilitatea navigării între pagini, am adăugat și opțiunea deconectării, folosindu-mă de librăria Bootstrap. Atributului HTML i-am asignat următoarea valoare: . Astfel, se va face trimitere înapoi către pagina de autentificare. Dacă se optează pentru deconectare, pe pagina de autentificare se va realiza o altă verificare prin intermediul funcției „isset(\$_GET['logout'])”. Dacă se confirmă verificarea, vom șterge variabilele de sesiune și vom șterge și sesiunea [18].

Pe lângă meniul stilizat și mesajul de primire, utilizatorului îi sunt propuse încă două rubrici de prezentare. Prima este compusă dintr-un mesaj de tip text care descrie pe scurt proiectul și evidențiază necesitatea lui. Următoarea rubrică este reprezentată de un carusel de imagini reprezentative construit din librăria Bootstrap. Caruselul conține o serie de imagini descriptive cu componentele hardware peste care este suprapus un scurt text descriptiv. Navigarea între imagini se poate realiza manual prin apăsarea lateralelor imaginilor sau automat o dată la cinci secunde.

3.2.2.3 Pagina de citiri

Odată ajuns la această pagină, utilizatorul este întâmpinat cu același mesaj, dar are la dispoziție posibilitatea de a alege între o serie de comenzi posibile. Acum nu mai este informat cu privire la aspecte generale ale proiectului, ci este ilustrat pe pagină un tabel ce conține ultimele cincisprezece citiri venite de la senzor. Acest tabel este setat din cod sa se actualizeze automat, fără a afecta funcționarea componentelor din restul paginii. Implementarea acestei funcționalități a fost posibilă prin intermediul unui script JavaScript care va apela încărcarea unui fișier „load.php” la intervale de o secundă. Acest fișier este unui creat adițional, pentru a-l putea apela extern. În interiorul lui este implementat un cap de tabel simplu cu patru coloane(ID, TIMP, PUTERE, CURENT), acestea reprezentând datele culese din baza de date prin procedura prezentată în continuare.

Prin aceeași modalitate se realizează conexiunea la baza de date „test”, de unde se va realiza extragerea de informații din tabelul „sensor”, de această dată, prin intermediul sintaxei MySQL: 'SELECT id, time, value, value2 FROM test.sensor ORDER BY id DESC limit 15'. În acest tabel sunt memorate date referitoare la curent și putere. Rezultatul acestui query este salvat într-o variabilă „\$result”, care va servi drept parametru pentru următoarea construcție:

```
while($row = mysqli_fetch_array($result))
{
    echo "<tr>";
    echo "<td>" . $row['id'] . "</td>";
    echo "<td>" . $row['time'] . "</td>";
    echo "<td>" . $row['value'] . "</td>";
    echo "<td>" . $row['value2'] . "</td>";
    echo "</tr>";
}
```

Figură 3.30 Popularea unui tabel HTML cu un script PHP

Prin intermediul funcției „mysqli_fetch_array()” sunt aduse intrările din tabel rând cu rând, atâta timp cât ele există. Pe baza lor sunt extrase informațiile individual pe coloane. Corespunzător acestei operațiuni, datele sunt introduse în tabelul creat anterior prin funcția PHP „echo”.

Cum am spus anterior, pe lângă textul afișat în scop informativ, pentru vizualizarea eventualelor fluctuații de consum, utilizatorul are la dispoziție o serie de comenzi. Pentru început, el poate începe înregistrarea consumului pentru un anumit consumator. Pentru aceasta, va apăsa butonul verde „Începe citirea”. În acest moment, va fi declanșată lansarea unui prompt ce va cere introducerea unui nume dispozitivului. Acest prompt este creat tot prin intermediul librăriei Bootstrap și este adaptat, așa încât să cuprindă un câmp text de intrare. În cadrul acestuia, utilizatorul va introduce numele dispozitivului, care este trimis mai departe către un fișier numit „insert.php”, cu scopul de a introduce numele dispozitivului în tabelul corespunzător. Acest lucru este posibil prin aceeași sintaxă MySQL, însă sub următoarea formă:

```

$sql = "INSERT INTO device (description) VALUES ('".$_POST["name"]."');";
$sql .= "INSERT INTO sensor (ID_Device) SELECT id FROM device WHERE id = LAST_INSERT_ID()";

if (mysqli_multi_query($conn, $sql)) {
    header ("Location: get_data.php");
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
    header ("Location: get_data.php");
}

$conn->close();

```

Figură 3.31 Query multiplu pentru inserare în două tabele

Pe lângă inserarea numelui trimis din formular în baza de date, am introdus în tabelul corespunzător senzorului, în coloana „ID_Device”, id-ul ultimului dispozitiv creat de utilizator prin intermediul sintaxei „LAST_INSERT_ID()”, în ideea realizării monitorizării consumului. Ulterior, am utilizat ca parametru data și ora inserării. Ce este de evidențiat în cadrul scriptului ilustrat, este posibilitatea creării unui query multiplu prin concatenarea a doua variabile, utilizând „;” la finalul primului șir MySQL și „.” înainte de a atribui celei de-a doua variabile următorul șir. Apelarea acestui query este realizată într-un bloc „if”, al cărui condiție, dacă este satisfăcută, va trimite direcționa utilizatorul înapoi la pagina de citiri date, ca acesta să poată utiliza interfața în continuare. În caz contrar, locația este aceeași, dar va apărea și un mesaj de eroare.

Trimiterea numelui dispozitivului am implementat-o inițial prin adăugarea unui formular în prompt ce avea atributul „action” setat cu numele fișierului PHP ce realiza inserarea în tabel. Ulterior, am implementat o construcție mai avansată utilizând AJAX (Asynchronous JavaScript And XML). Scriptul prin care am implementat această funcționalitate este ilustrat în figura următoare:

```

<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
    var submitButton = $("#submitValue");

    submitButton.on("click", function(){
        $.ajax({
            url:"insert.php?",
            type:"POST",
            data: {
                name: $("#deviceName").val()
            }
        }).then(function(response){
            $("#stopRecName").val($("#deviceName").val());
            $("#deviceName").val('');
            $('#startRec').css('display', 'none');
            setTimeout(function(){
                $('#stopRec').css('display', 'block');
            }, 10000);
        }, function(error){
            console.log(error);
        });
    });

```

Figură 3.32 Script Ajax pentru scenariul butoanelor

Funcționarea logică a interfeței de citiri implică rularea celor două butoane de începere, respectiv încetare a înregistrării după un anumit scenariu. Mai exact, la intrarea în pagină trebuie să apară numai butonul verde. După apăsarea acestuia și prelucrarea informațiilor, trebuie să apară butonul roșu. Totuși, cum informațiile sunt trimise de la senzor la intervale de 5 secunde, este necesară implementarea unui întârzieri. După terminarea procesării informațiilor în cadrul promptului declanșat de butonul roșu, trebuie să reapară butonul verde. Această funcționalitate este implementată prin codul ilustrat anterior. Ce face per ansamblu această implementare este evitarea reîmprospătării paginii în momentul în care trimitem către serviciul PHP de inserare în tabela dispozitivelor numele dispozitivului. Acest lucru este posibil prin introducerea unui apel asincron la serviciu. Am atribuit unei variabile „submitButton” identificatorul butonului de validare a numelui dispozitivului din promptul Bootstrap, variabilă pe care am folosit-o ulterior pentru a-i atașa un „event handler” (procedură prin care se declanșează o acțiune urmată de apariția unui eveniment). În cazul de față, evenimentul este reprezentat de apăsarea butonului. În cadrul acesteia, am creat o funcție anonimă ajax căreia i-am dat ca parametru un obiect cu câmpuri de adresă, tip și date. Ce îndeplinește această funcție anonimă este de fapt înlocuirea atributelor „action”, „method” și „input text” din cadrul unui formular HTML. Așadar, câmpul de adresă va fi egal cu „insert.php”, tipul va fi „POST”, iar datele vor fi reprezentate de o singură valoare și anume numele consumatorului descris în prompt. Structura acestui script ajax implică acționarea unui răspuns dacă funcția anonimă este rulată cu succes. În cadrul blocului „then”, răspunsul este descris prin setarea unor acțiuni. Am setat în promptul de încetare a înregistrării numele dispozitivului, făcându-l totodată posibil a fi numai citit. Pe lângă aceasta, am setat butonul verde de începere a înregistrării să dispară după ce este confirmată apăsarea butonului „submit”. Butonul roșu va apărea pe ecran după zece secunde, întârziere necesară ca serverul să culeagă informații de la senzorul de curent. Butonul verde va reapărea după ce este confirmat butonul „submit” din promptul de încetare a citirii, deoarece nu am implementat funcție ajax și pentru butonul roșu.

O altă funcționalitate importantă implementată pe această pagină este acționarea releului. Aceasta a fost realizată în două moduri, propunându-se astfel două variante de acționare. Una este compusă dintr-o suprafață de text în care utilizatorul poate scrie literele „A” sau „F”, corespunzătoare stărilor de pornit sau oprit ale releului, iar cealaltă variantă este construită din două butoane radio intuitive cu descrierile „Pornit”, „Oprit”. Aceste variante fac parte dintr-un formular HTML, fiecare dintre ele atribuind unei variabile numite „status” starea releului. Orice variantă poate fi aleasă, însă trebuie validată prin apăsarea butonului albastru. Acțiunea formularului este bazată pe un serviciu PHP numit „updateRelay”. Acesta nu face altceva decât să realizeze conexiunile necesare cu baza de date și să modifice valoarea variabilei „status” din tabelul „RelayStatus”, în funcție de preferința utilizatorului. Totuși, nu este de ajuns pentru acționarea releului, întrucât tot ce se modifică este pe partea de sever, în baza de date. De aceea, am mai creat un fișier PHP cu denumirea „fetchRelayStatus.php” ce va selecta din tabelul corespunzător variabila „status” prin intermediul sintaxei mysqli. Ulterior, aceasta va fi transmisă mai departe prin intermediul funcției „echo”, pentru ca placa de dezvoltare Arduino Uno Rev3 să o citească prin configurarea sa de client web.

Așa cum am menționat în paginile anterioare, meniul de navigare propune o pagină de calcul al costului și consumului, însă navigarea către ea presupune urmarea secvenței de pași descriși tot anterior, referitoare la butoanele de începere, respectiv încetare a citirii de la senzor. Prin confirmarea ultimei etape, cea de validare a opririi înregistrării, se va face trimitere la pagina dorită de costuri. Aceasta este implementată prin serviciul „fetchJoin.php”. Pagina are același meniu stilizat, însă aranjarea elementelor în pagină și elementele diferă. Pentru început, va fi afișată o primă rubrică cu patru câmpuri, fiecare dintre ele reprezentând o valoare calculată pe baza informațiilor venite de la senzor. Prima valoare semnifică puterea medie consumată de dispozitiv și este calculată prin intermediul apelului către un alt serviciu PHP, numit „calculateConsumption.php”. În codul implementat în acest serviciu am folosit construcția ilustrată mai jos:

```
<?php
    require 'loadLastDevice.php';
    $id = $name['id'];
?>
```

Figură 3.33 Apel către un serviciu PHP

Scopul acesteia este de a încărca în cadrul serviciului o singură informație referitoare la ultimul dispozitiv inserat în baza de date și anume id-ul. Acest lucru a fost posibil prin apelul către un fișier php ce va returna din tabelul pentru dispozitiv ultima intrare adăugată și o face accesibilă coloană cu coloană prin construcția următoare, ce returnează un rând într-un șir asociativ [26]:

```
$result = mysqli_query($con,'SELECT * FROM test.device d ORDER BY d.id DESC LIMIT 1');
$name = (mysqli_num_rows($result)==1) ? mysqli_fetch_assoc($result) : null;
```

Figură 3.34 Selectarea ultimei intrări dintr-un tabel

Întorcându-ne la serviciul „calculateConsumption.php”, aici, pe baza sintexei MySQL am construit un query JOIN pentru a extrage din baza de date informațiile necesare referitoare la ultimul consumator înregistrat. Înainte de aceasta, am făcut o adaptare a unei funcții al cărei rezultat l-am atribuit unei variabile numite „stopTime”. Mai exact, serverul a necesitat adaptarea zonei de timp pentru România prin mutarea orei cu trei ore în plus față de valoarea inițială, pentru sincronizare. Această acțiune a dus la necesitatea adaptării funcției „date()”, în vederea utilizării variabilei „stopTime” în query-ul menționat anterior.

```
$stopTime = date('Y-m-d H:i:s', strtotime('+3 hour'));
```

Figură 3.35 Setarea variabilei de timp pentru oprirea înregistrării

Query-ul l-am construit bazându-mă pe orele exacte la care sunt inserate datele de la senzor în tabel, ora exactă la care este inserat un dispozitiv în tabel, numele atribuit acestuia în cadrul promptului de începere a citirii, respectiv id-ul acestuia, pentru a evita acumularea costului și consumului, în cazul inserării a două sau mai multe dispozitive cu același nume. Condiția de timp este ca ora la care a fost înregistrat dispozitivul să fie mai mică decât ora la care s-a înregistrat citirea de la senzor, iar citirea de la senzor să fie realizată înainte de „stopTime”, variabilă declarată anterior ca fiind ora actuală. Această

variabilă este declanșată în momentul apăsării butonului de validare a opririi înregistrării.

```
$result = mysqli_query($con,"SELECT time, value, reg_date, description FROM test.sensor s JOIN test.device d
ON s.time > d.reg_date AND s.time < '$stopTime' AND d.id = '$id' AND description = '$_POST["name"]."');"
```

Figură 3.36 Query JOIN pentru calcularea consumului și costului

Pentru calculul puterii medii consumate, am declarat o variabilă „sum” la care am adăugat toate valorile venite din coloana „value”, corespunzătoare puterii, conform query-ului anterior. Ulterior, am făcut media prin împărțirea la numărul de rânduri returnate din tabel, tot pe baza query-ului.

Tot în acest fișier „calculateConsumption.php” am definit încă o variabilă pe care am folosit-o la calculul energiei consumate. Pe baza informațiilor ce se găsesc pe pagina [27], am definit un cost pentru un KWh de 0.3741 lei.

Toate celelalte câmpuri ilustrare pe pagina de cost și consum sunt calculate pe baza funcționalităților despre care am discutat anterior. Mai precis, energia consumată și costul acesteia sunt calculate luând în calcul rata citirilor de 5 secunde, costul unui KWh și numărul de înregistrări generat de query. Toate aceste valori intră în următoarea formulă:

```
Cost: <?php echo $sum * ($num_rows * 5) / 720000 * $costKWh;
```

Figură 3.37 Calculul costului

Intervalul de timp ilustrat în minute cât care se face măsurarea este calculat împărțind numărul de rânduri general de query la numărul 12, adică numărul de înregistrări pe minut.

Pe lângă rubrica informativă cu datele despre cost și consum, sub aceasta este implementat un tabel în care sunt aduse toate înregistrările care aparțin ultimului dispozitiv descris de utilizator. Procedura de culegere a datelor este aceeași, sunt necesare datele despre ultimul dispozitiv inserat și ora exactă de oprire a monitorizării consumului pentru construirea query-ului. Acesta este la fel ca și cel prezentat anterior, însă de data aceasta nu este nevoie de procesarea datelor aduse, ci de afișarea lor în scop informativ. Pentru navigarea înapoi la pagina de citiri, am implementat și un buton ce are în spate un script JavaScript în care am creat o funcție „window.history.back()”.

3.2.2.4 Pagina de rapoarte

Pentru păstrarea unei evidențe a consumatorilor monitorizați și ilustrarea grafică a consumului, respectiv costului, am creat un grafic pe două axe ce conține date despre energia consumată și traducerea acesteia direct în cost. Implementarea acestui grafic am făcut-o folosindu-mă de un set de librării gratuite puse la dispoziție, pe lângă cea Bootstrap deja utilizată. Acestea sunt denumite „Morris” și „Raphael” și sunt disponibile în următoarea descriere, împreună cu sugestii despre implementare. După ce am adus datele din necesare din baza de date prin intermediul unui query MySQL, creat graficul prin construcția următoare:

```

Morris.Bar({
  element : 'chart',
  data:[<?php echo $chart_data; ?>],
  xkey:'Description',
  ykeys:['Cost', 'Consumption'],
  labels:['Cost', 'Consumption'],
  hideHover:'auto',
  stacked:false
});
</script>

```

Figură 3.38 Grafic pentru cost și consum

Pe axa orizontală a graficului am așezat numele dispozitivelor, iar pe verticală am alăturat două bare care descriu grafic consumul de energie și costul.

3.2.2.5 Script PHP pentru inserarea datelor achiziționate

Deși acest script nu face parte din interfața grafică, el este extrem de important, întrucât prin intermediul lui se realizează conexiunea dintre componenta hardware și serverul de la distanță. Codul implementat în acest fișier nu este unul complex, însă săvârșește o acțiune esențială. Prin intermediul construcției `"$_GET["value"]"` sunt preluate datele trimise de clientul Arduino prin metoda GET și ulterior sunt prelucrate în vederea inserării în baza de date, în tabelul senzorului, prin intermediul unui query MySQL.

3.3 Testare și validare

Aplicația dezvoltată este pregătită pentru utilizare, însă este necesară o verificare a funcționalității, atât din punctul de vedere al componentelor fizice, cât și software.

3.3.1 Testarea conexiunilor fizice

Conexiunile dintre componente sunt realizate prin intermediul unor fire subțiri, fie mamă-tată, tată-tată sau printr-un cablu mai gros în cazul conectării senzorului de curent în linie cu cablul de la priză. Nu sunt efectuate lipiri și de aceea este nevoie de o testare a stabilității și fixării componentelor, pentru a îndeplini cerința unor date consistente și a unor conexiuni stabile.

3.3.2 Teste și măsuri de siguranță

Luând în considerare faptul că se lucrează cu tensiuni directe de 220V, tensiune alternativă și curenți ridicați de până la 16 Amperi, cât poate genera priza, este obligatorie implementarea unor măsuri de siguranță de bază. Sistemul este izolat în locurile importante prin bandă sau sunt ascunse conexiunile riscante. Am efectuat teste prin intermediul unui multimetru, pentru a detecta eventuale riscuri.

3.3.3 Teste de consum

Modulul Arduino NANO este alimentat în situația de față prin intermediul portului USB de la priză, fapt ce nu afectează măsurătorile. Consumul este redus, de ordinul a 0.03 – 0.04 Amperi, curent continuu. În schimb modulul Uno Rev 3 consumă semnificativ mai mult, cifra ridicându-se la 0.19 – 0.20 Amperi. Aceste măsurători au fost efectuate prin intermediul unui aparat de măsură digital USB. Pentru a măsura efectiv cantitatea de curent consumat, am utilizat un acumulator extern cu capacitatea de 10000A cu care am alimentat modulul conectat la internet.

3.3.4 Teste de conexiune Bluetooth

Așa cum am precizat în obiectivele lucrării, se dorește atingerea unor distanțe minime pe care modulele Bluetooth să poată realiza transferul de date fără întreruperi. Am efectuat teste prin îndepărtarea modulelor, în încercarea de a afla distanța maximă de comunicație. Acest proces s-a efectuat în cadrul unei locuințe, unde, ca factori externi ce pot influența conexiunea, sunt pereți, elemente de mobilier sau electrocasnice.

3.3.5 Teste interfață web

Interfața web a fost testată prin navigarea pe toate paginile și prin toate funcționalitățile. Paginile web au fost create în așa fel încât să se adapteze formatului ecranului pe care se autentifică utilizatorul și de aceea testarea a fost efectuată pe mai multe dispozitive cu ecrane diferite (laptop, telefon).

Pe lângă aceasta, am efectuat un test pe un eșantion de 10 persoane pentru a observa cât de intuitivă este utilizarea interfeței.

3.3.6 Validare măsurători

Măsurarea prin intermediul senzorului de curent a necesitat o calibrare inițială, acțiune realizată prin intermediul unui multimetru. Tot pe baza lui, s-au comparat rezultatele măsurătorilor, pentru ajustarea constantă. Un alt test efectuat pentru validare s-a efectuat prin măsurarea consumului unor dispozitive cu consum fix, cum ar fi un bec de 65W sau un sandwichmaker. Ca urmare, prin comparația dintre rezultatul obținut și consumul atribuit consumatorului, s-a realizat validarea.

4 Concluzii

Această lucrare reprezintă o variantă simplă de măsurare a consumului și de control asupra acestuia prin acționarea directă asupra sursei de curent (priza). Totodată, prin construcția și arhitectura proiectului, lucrarea înglobează tehnologii recente, avansate, fiabile și ușor de manevrat, precum plăci de dezvoltare Arduino, componente adiacente compatibile, baze de date MySQL și tehnologii PHP, JavaScript.

Dezvoltarea proiectului a necesitat aproximativ patru luni, timp în care am efectuat o documentare potrivită, am efectuat verificări și teste pentru funcționarea corectă a aplicației și am propus o variantă prototip pe un stand de test, pentru o prezentare intuitivă. Costul acestui prototip a ajuns la aproximativ 200 de lei, însă acesta poate fi

reduc prin achiziționarea unor componente mai ieftine, cu capacități mai reduse, însă care ar suporta cerințele. Dezvoltarea s-ar putea face mai rapid, însă, pentru început, a fost nevoie de asimilarea cunoștințelor de programare web și de familiarizare cu componentele hardware. În acest fel, se reflectă următoarele contribuții personale în cadrul proiectului:

- Arhitectura componentelor fizice și comunicarea prin intermediul protocolului Bluetooth reprezintă o diferențiere față de restul dispozitivelor de pe piață care propun o comunicare prin wi-fi.
- Modul în care componentele comunică cu serverul prin intermediul bazei de date a cărei arhitectură a fost gândită în maniera în care să poată fi accesată cu ușurință. Baza de date poate fi accesată direct de către utilizator prin intermediul clientului SSH PuTTY.
- Configurarea interfeței web prin intermediul serviciilor PHP. Această caracteristică a fost proiectată cu scopul unei utilizări intuitive, ea propunând utilizatorului aspecte concrete, fără ambiguități, asupra consumului și costului energiei consumate.

4.1 Rezultate obținute

Pe baza testelor efectuate, s-a obținut o serie de rezultate prin care se poate spune că obiectivele inițiale au fost atinse. Pentru început, prin testarea conexiunilor fizice s-a dovedit că elementele hardware comunică fără probleme, însă este de menționat că pentru o mai bună stabilitate este nevoie de o mai bună arhitectură și distribuire a cablajelor. Următorul test, cel al măsurilor de siguranță, a scos la iveală faptul că sistemul este unul sigur, atât din punctul de vedere al conexiunilor, cât și al eventualelor puncte cu risc mai ridicat, cum ar fi microprocesoarele de pe plăcile de dezvoltare, care se încălzesc și prezintă riscuri. Cel de-al treilea test efectuat are ca rezultate un consum de 417A/h sau 7A/minut pentru modulul Master compus din placa de dezvoltare Arduino Uno Rev3 și componentele adiacente. Pentru modulul Slave, valorile sunt de aproximativ șase ori mai mici. Alimentarea acestui modul se poate face tot prin intermediul bateriei, pentru a exclude complet ca eroare de măsurare faptul că alimentarea vine prin intermediul prizei. Testarea interfeței web a decurs fără probleme, excluzând momentele în care încărcarea paginilor s-a făcut mai încet din cauza unor probleme de rețea. Aplicația funcționează bine și pe telefon, elementele grafice din pagină așezându-se corespunzător ecranului pe care sunt ilustrate. Testarea pentru verificarea nivelului intuitiv de utilizare a aplicației a obținut un rezultat de 7 din 10, adică șapte persoane din zece au reușit să navigheze prin paginile web și să testeze funcționalitățile, fără a li se explica anterior mersul firesc al procesului, acest lucru atribuind lucrării o notă pozitivă.

4.2 Direcții de dezvoltare

Varianta aceasta a lucrării este una prototip care pune la dispoziție o serie de facilități elementare, însă care își ating scopul. Există posibilitatea implementării ulterioare pentru diferite funcționalități și totodată, îmbunătățirea structurii fizice, a arhitecturii.

Un prim pas în dezvoltarea ulterioară ar fi curățarea structurii componentelor fizice prin realizarea lipirilor firelor, rezultând astfel contacte sigure și performante. Componentele ar putea fi așezate pe plăci PCB, rezultând înglobarea lor, fapt ce ar schimba caracterul de prototip într-unul de dispozitiv modernizat.

Tot referitor la ideea anterioară, sistemul poate fi închis în carcase printate cu ajutorul imprimantelor 3D, pentru a ascunde aspectele ce nu sunt de uz pentru utilizator. Aceștia îi poate fi ilustrate doar informații pe baza displayului LCD, restul conexiunilor fiind camuflate.

O altă idee este cea a realizării alimentării modului Slave de la priză, fără a mai fi nevoie de ieșirea USB 2.1. Am utilizat această priză, în ideea modularizării sistemului, însă transformatorul acesteia ar putea fi extras și adaptat nevoilor proiectului.

Tot în ceea ce privește aspectele fizice, în cadrul proiectului este posibilă și necesară conectarea a încă un senzor de curent cu limite mai mici de până la 5A, pentru a mări precizia de măsurare în cazul consumatorilor mai mici. Trecerea de la utilizarea unui senzor la celălalt se va face simplu din cod.

Ce s-ar putea implementa ulterior, ar fi adăugarea unui modul GSM prin intermediul căruia să se poată acționa priza în cazul unei probleme de conexiune la internet. Astfel, prin trimiterea unui singur mesaj, utilizatorul ar putea acționa de la distanță priza. Tot în cazul acestei situații, dacă avem o eroare de conexiune la internet, datele ar trebui salvate temporar într-o zonă de memorie, astfel încât în momentul reconectării, datele să poată fi trimise către server.

O altă dezvoltare ulterioară este posibilă pe partea web, unde, pe lângă nevoia unei stilizări avansate a paginilor, este nevoie de implementarea unui nivel mai ridicat al securității. În acest moment, accesul la interfață se realizează printr-o sesiune de autentificare simplă.

Tot pe partea web, este nevoie de implementarea a mai multe grafice referitoare la consum și cost, care să reflecte o mai bună monitorizare a dispozitivelor, cum ar fi ierarhizări, frecvența utilizării. Pe lângă acestea, este posibilă dezvoltarea unui algoritm de predicție a consumului pe parcursul unei perioade setate.

Nu în ultimul rând, sistemul poate fi dezvoltat pentru a fi capabil să citească date venite de la mai multe prize simultan. În momentul de față, conexiunea Bluetooth implementată suportă comunicarea între un modul Master și un singur modul Slave. Prin implementarea ulterioară a acestei facilități, utilizatorul ar fi capabil să monitorizeze și să controleze un întreg sistem electric de prize dintr-o locuință.

5 Bibliografie

- 1] [M. CHINDRIȘ, A. CZIKER, A. MIRON și B. TOMOIOAGĂ, Managementul energiei electrice. Aplicații, Cluj-Napoca: Casa Cărții de Știință, 2009.
- 2] [„ENERSAC,” [Interactiv]. Available: <http://www.enersac.com/energia-solar-faq-que-diferencia-hay-entre-cargas-inductivas-y-cargas-resistivas-in.php>. [Accesat 26 Mai 2017].
- 3] [„sciencing,” [Interactiv]. Available: <http://sciencing.com/difference-between-resistive-inductive-loads-12181159.html>. [Accesat 28 Mai 2017].
- 4] [A. MORARAU, V. HORTOPAN și I. CIRIC, Electrotehnică, măsurări și mașini electrice, București: Editura Didactică și Pedagogică, 1976.
- 5] [N. Bogoevici, Electrotehnică și măsurări electrice, București: Editura didactică și pedagogică, 1979.
- 6] [„radio-electronics.com,” [Interactiv]. Available: http://www.radio-electronics.com/info/t_and_m/digital-multimeter/how-a-dmm-works-operation.php. [Accesat 20 Mai 2017].
- 7] [E. POP, E. STOICA și S. CRIȘAN, Măsurări în energetică, Timișoara: Editura Facla, 1981.
- 8] [„Senzor Hall,” 2009. [Interactiv]. Available: <http://www.e-automobile.ro/categorie-electronica/106-senzor-hall.html>.
- 9] [„phys.utcluj.ro,” [Interactiv]. Available: <http://www.phys.utcluj.ro/resurse/Facultati/LucrariDeLaborator/Lupsa/STUDIUL%20EFFECTULUI%20HALL.pdf>.
- 10] [M. BARLEA, „5 Senzori pentru curent electric,” [Interactiv]. Available: <http://users.utcluj.ro/~mbirlea/a/06a.htm>.
- 11] [„Arduino NANO,” Arduino, [Interactiv]. Available: <https://store.arduino.cc/arduino-nano>. [Accesat 13 Iunie 2017].
- 12] [„Arduino Uno Rev3,” Arduino, [Interactiv]. Available: <https://store.arduino.cc/arduino-uno-rev3>. [Accesat 13 Iunie 2017].
- 13] [„Ethernet Shield 2,” Arduino, [Interactiv]. Available: <https://store.arduino.cc/arduino-ethernet-shield-2>. [Accesat 13 Iunie 2017].
- 14] [„hobbymarket,” [Interactiv]. Available: <http://www.hobbymarket.ro/module-atasabile/modul-1-releu-5v>. [Accesat 22 Martie 2017].

- [„eMag,” [Interactiv]. Available: <https://www.emag.ro/centrala-general-a-fibaro-home-center-2-fghc2/pd/DQ511BBBM/>.
- [ARduino, „Arduino Serial,” [Interactiv]. Available: <https://www.arduino.cc/en/Reference/Serial>. [Accesat 2 Mai 2017].
- [G. HORTOPAN și V. HORTOPAN, Șunturi și divizoare de tensiune, vol. 1, București: Editura tehnică, 1978, pp. 22-30.
- [L. ULLMAN, PHP și MySQL pentru site-uri web dinamice, București: Pearson Educational, Inc, Peachpit Press / Distribuție Editura Teora SRL, 2006.
- [Hostinger, „Hostinger Tutorial,” Hostinger, [Interactiv]. Available: <https://www.hostinger.com/tutorials/vps/how-to-install-lamp-stack-centos7>. [Accesat 18 Mai 2017].
- [„SSH Communication,” [Interactiv]. Available: <https://www.ssh.com/ssh/protocol/>. [Accesat 17 Mai 2017].
- [„PUTTY,” [Interactiv]. Available: <http://www.putty.org/>. [Accesat 14 Februarie 2017].
- [Arduino, „Arduino Forum Funcție,” [Interactiv]. Available: <https://forum.arduino.cc/index.php?topic=179541.0>. [Accesat 14 Aprilie 2017].
- [Arduino, „Arduino micros(),” [Interactiv]. Available: <https://www.arduino.cc/en/Reference/Micros>. [Accesat 19 Mai 2017].
- [howtomechatronics, „LCD setup,” [Interactiv]. Available: <http://howtomechatronics.com/tutorials/arduino/lcd-tutorial/>. [Accesat 5 Iunie 2017].
- [one2one, „linux-utils.blogspot.ro CommandBuffer,” 14 Octombrie 2010. [Interactiv]. Available: <http://linux-utils.blogspot.ro/2010/10/arduino-serial-read-line.html>. [Accesat 26 Mai 2017].
- [w3schools.com, „mysqli_fetch_assoc,” [Interactiv]. Available: https://www.w3schools.com/php/func_mysqli_fetch_assoc.asp. [Accesat 15 Mai 2017].
- [electricafurnizare, „electricafurnizare,” [Interactiv]. Available: <http://www.electricafurnizare.ro/clienti/clienti-casnici/tarife-reglementate-avizate-de-anre/tarife-avizate-anre/>. [Accesat 28 Iunie 2017].
- [J. Noble, în *Programming interactivity*, O'Reilly, 2009, p. 693.
- [webslesson, „Morris chart,” [Interactiv]. Available: <http://www.webslesson.info/2017/03/morrisjs-chart-with-php-mysql.html>. [Accesat 16 Iunie 2017].

6 Anexe

Figură 2.1 Reprezentarea grafică a unei funcții periodice de timp	9
Figură 2.2 Cuantizare semnal continuu [7] pg 79	11
Figură 2.3 Priză Fibaro de tip F.....	15
Figură 2.4 Integrare web pentru controlul și monitorizarea prizei Fibaro.....	15
Figură 3.1 Fluxul de date	16
Figură 3.2 Divizor de tensiune pentru conectarea modului Bluetooth.....	17
Figură 3.3 Distribuirea rețelei hotspot prin portul ethernet	18
Figură 3.4 Client PuTTY	20
Figură 3.5 Detalii SSH ale serverului	21
Figură 3.6 Autentificare PuTTY.....	21
Figură 3.7 Rata utilizării PHP	23
Figură 3.8 Solicități client – server.....	23
Figură 3.9 Aplicație FTP (FileZilla).....	24
Figură 3.10 Tabelul "device"	25
Figură 3.11 Tabelul "sensor"	26
Figură 3.12 Tabelul "users"	26
Figură 3.13 Comunicare client - server bazată pe servicii PHP, MySQL.....	27
Figură 3.14 Diagrama de scenarii	27
Figură 3.15 Cod pentru funcția de prelucrare a datelor citite de sensor	30
Figură 3.16 Apelul funcției de citire curent	30
Figură 3.17 Configurare Display LCD	31
Figură 3.18 Setare pentru displayul LCD	31
Figură 3.19 Comunicare serială.....	32
Figură 3.20 Conexiune client Arduino - server	33
Figură 3.21 Acționarea releului	33
Figură 3.22 Pagina de autentificare	34
Figură 3.23 Librării web.....	34
Figură 3.24 Sugestie în formular sub formă de text	35
Figură 3.25 Setarea variabilelor de sesiune.....	35
Figură 3.26 Datele de conexiune pentru server și baza de date	35
Figură 3.27 Query pentru extragerea datelor de autentificare	35
Figură 3.28 Sintaxă pentru trimiterea forțată către o adresă.....	36
Figură 3.29 Utilizarea variabilelor de sesiune	36
Figură 3.30 Popularea unui tabel HTML cu un script PHP.....	37
Figură 3.31 Query multiplu pentru inserare în două tabele.....	38
Figură 3.32 Script Ajax pentru scenariul butoanelor	38
Figură 3.33 Apel către un serviciu PHP	40
Figură 3.34 Selectarea ultimei intrări dintr-un tabel.....	40

Figură 3.35 Setarea variabilei de timp pentru oprirea înregistrării	40
Figură 3.36 Query JOIN pentru calcularea consumului și costului	41
Figură 3.37 Calculul costului.....	41
Figură 3.38 Grafic pentru cost și consum	42
Ecuția 2.1.....	8
Ecuția 2.2.....	8
Ecuția 2.3.....	9
Ecuția 2.4.....	9
Ecuția 2.5.....	9
Ecuția 2.6.....	10
Ecuția 2.7.....	10
Ecuția 2.8.....	10
Ecuția 3.1.....	18
Tabel 1 Consum instrumente de măsurare.....	12