Automatică și Calculatoare

Informatică

2022-2023

**SISTEM PENTRU STUDIUL**

**RENTABILITAȚII IMPLEMENTARII**

**METODELOR DE EFICIENTIZARE**

**A PRODUCȚIEI DE ENERGIE**

**ÎN SISTEMELE FOTOVOLTAICE**

**Candidat: Arnăutu Dumitru-Florin**

**Coordonator științific:** **Conf.dr.ing. Adrian Korodi**

Sesiunea: Septembrie 2023

# CUPRINS

# 

[CUPRINS 2](#_Toc144924849)

[**1 Introducere** 4](#_Toc144924850)

[**1.1** **Contextul actual** 4](#_Toc144924851)

[**1.2** **Tema lucrarii** 5](#_Toc144924852)

[**1.3** **Structura lucrarii** 6](#_Toc144924853)

[**2 Considerații teoretice și practice** 7](#_Toc144924854)

[**2.1 Echipamente hardware** 7](#_Toc144924855)

[**2.1.1 Placa de dezvoltare ESP32** 7](#_Toc144924856)

[**2.1.2 Clona a placii de dezvoltare Arduino Mega2560** 8](#_Toc144924857)

[**2.1.3 Senzorii de temperatură BMP280 si AHT21** 9](#_Toc144924858)

[**2.1.4 Senzorii pentru masurarea curentului INA219** 10](#_Toc144924859)

[**2.1.5 Puntile H** 12](#_Toc144924860)

[**2.1.6 Coborator de tesiune XL4015** 13](#_Toc144924861)

[**2.1.7 Modul microSD** 14](#_Toc144924862)

[**2.1.8 Module de control incarcare solara PWM , RBL-30A** 14](#_Toc144924863)

[**2.1.9 Panouri solare 20W** 15](#_Toc144924864)

[**2.1.10 Actuatoare liniare** 16](#_Toc144924865)

[**2.1.11 Modul SENZOR 3 AXE ADXL345** 17](#_Toc144924866)

[**2.1.13 TRANSLATOR NIVEL LOGIC I2C IIC BIDIRECTIONAL 8 CANALE 3.3V 5V TXS0108E** 17](#_Toc144924867)

[**2.2 Echipamente software** 20](#_Toc144924868)

[**2.2.1 Arduino IDE** 20](#_Toc144924869)

[**2.2.2 GitHub** 21](#_Toc144924870)

[**2.2.3 Limbajul de programare C++ în Arduino IDE** 23](#_Toc144924871)

[**2.2.4 I2C** 25](#_Toc144924872)

[**3 Arhitectura Sistemului** 27](#_Toc144924873)

[**3.1 Diagrama de stari** 27](#_Toc144924874)

[**3.2 Schema structurii sistemului** 28](#_Toc144924875)

[**3.3 Schema structurii sistemului** 29](#_Toc144924876)

[**4. Codul de la baza proiectului** 31](#_Toc144924877)

[**5. Rezultatele Obtinute** 38](#_Toc144924878)

[**6. Concluzii și direcții de dezvoltare** 39](#_Toc144924879)

[**6.1 Concluzii** 39](#_Toc144924880)

[**6.2 Direcții de dezvoltare** 40](#_Toc144924881)

[**FIGURI și TABELE** 41](#_Toc144924882)

[**BIBLIOGRAFIE** 42](#_Toc144924883)

# **1 Introducere**

## **Contextul actual**

La momentul de față, în contextul productiei de energie verde s-au facut numeroase studii, rezultatele obtinute fiind mult prea influentate de factori particulari.

Un studiu[1] relizat in 2017 pentru a fii prezentat in fata unei comisii ce s-a adunat pentru a discuta despre una din metodele de eficientizare a productiei de energie solara prin modificarea pozitiei panourilor dintr-un parc fotovoltaic a constatat chiar ca aceasta metoda este nefezabila, productia de energie in cadrul parcului fotovoltaic scazand randamentul anual de la 3.3% la 2.3%.

Randamentul anual se exprima procentual si reprezinta raportul dintre cantitatea de energie produsa de instalatia fotovoltaica si totalul energiei la care este expusa instalatia, scaderea randamentului unei instalatii mobile se poate atribuii chiar utilizarii mecanismelor pentru pozitionarea panourilor ce la randul lor consuma o parte din energia produsa.

In figura de mai jos se prezinta comparatia dintre instalatia mobila si instalatia fixa pe durata a 4 luni din cadrul experimentului realizat pe baza caruia s-au intocmit concluziile din cadrul experimentului .

A graph of a graph with red and blue lines

Description automatically generated

Fig. 1.1-1 Grafic comparatie instalatie fixa si mobila

Un alt studiu [2] realizat in coasta de esta a Maleziei a venit cu rezultate mai promitatoare , in cadrul acestuia vazandu-se clar avantajul montarii panourilor solare pe un cadru mobil . In mod normal panourile fixe au o perioada destul de restransa de cateva ore cand vor produce maxumul de energie posibil , atunci cand lumina solara produce un unghi cat mai aproiat de 90 de grade fata de suprafata panourilor.

In graficul de mai jos se poate observa cresterea eficientei panourilor la inceputul zilei si la sfarsitul acesteia la nivelul panourilor montate pe cadru mobil.

A graph showing the time and time

Description automatically generated with medium confidence

Fig. 1.1-2 Grafic de comparatie intre panouri fixe si mobile

## **Tema lucrarii**

Această lucrare de licență are drept scop crearea unui sistem ce poate ajuta la determinarea eficienței si rentabilitații instalarii diverselor echipamente în instalațiile fotovoltaice pentru a soporii eficiența acestora.

In urma cercetarii diverselor studii publicate ce nu se limiteaza doar la cele mentionate in contextul acestei lucrari am ajuns la concluzia ca dezvoltarea unui sistem , a unui punct de plecare pentru studiul rentabilitatii multiplelor metode de eficientizare a randamentului instalatiilor fotovoltaice este un proiect ce imi va pune la incercare aptitudinile, permitandu-mi sa aplic cunostintele si abilitatile capatate pe durata celor 3 ani de studiu din facultate.

## **Structura lucrarii**

Lucrare de licenta structurata in 6 sectiuni.Dupa introducerea ce prezinta contextul actual si scopul proiectului.

Capitolul 2 prezinta elementele hardware si software folosite in realizarea sistemului.  
 Capitolul 3 prezinta arhitectura sistemului, modulele atat software si hardware de la nivelul sistemului impreuna cu modul de interactiune de la nivelul acestora .

Capitolul 4 explica implementarea proiectului.  
 Capitolul 5 arata rezultatele obtinute in teste, scenariile de utilizare.

Capitolul 6 prezinta concluziile din cadrul proiectului si directiile de dezvoltare .

# **2 Considerații teoretice și practice**

## **2.1 Echipamente hardware**

### **2.1.1 Placa de dezvoltare ESP32**

A circuit board with many different colored buttons

Description automatically generated with medium confidencePlaca de dezvoltate WiFi bazată pe ESP32 [3], un circuit integrat care contine module GPIO, I²C, I²S, PWM, SDIO, SPI, UART si ADC toate pe o singură placă și facil de utilizat cu sintaxa gen Arduino pentru acces la hardware. In figura 2.1.1-1 se poate observa placa de dezvoltare.

Fig. 2.1.1-1 Pinout-ul microcontroller-ului esp32

Caracteristici tehnice:

* Tensiune de alimentare: 3.3V
* Driver: CP2102
* Wi-Fi Direct (P2P), soft-AP , Bluetooth
* Consum current: 10uA – 170 mA
* Memorie flash: 4 MB
* Protocol TCP/IP
* Procesor: dual-core Xtensa® 32-bit LX6

Wireless model: 802.11 b/g/n ( pana la 150Mb/s)

### **2.1.2 Clona a placii de dezvoltare Arduino Mega2560**

Placa de dezvoltare construită după schemele publice ale celebrei placi ArduinoMega [4], fundația ce se ocupă de dezvoltarea platformei Arduino incurajează crearea si comercializarea plăcilor după propriile scheme , acestia le fac publice in mod intenționat pentru a spori contribuția utilizatorilor în dezvoltarea viitoarelor modele , in prețul unei plăci originale se regaseste si o “taxă” folosita în dezvoltarea proiectului Arduino , “misiunea” fundației este : ” enable anyone to enhance their lives through accessible electronics and digital technologies” ( sa pună la dispoziție oricui mijloacele electronice necesare pentru a-și imbunatați viața) .   
In figura 2.1.2-1 se poate observa clona placii Mega2560.

A blue electronic board with black wires

Description automatically generated

Fig. 2.1.2-1 ATmega2560

* Driver: CH340
* Tensiune de lucru: 5V
* Tensiune de intrare (recomandat): 7-12V
* Pini digitali: 54 (14 PWM output)
* Pini analogici: 16
* Curent de iesire: 40 mA
* Flash Memory: 256 KB, 8 KB pentru bootloader
* SRAM: 8 KB
* EEPROM: 4 KB
* Clock: 16 MHz

### **2.1.3 Senzorii de temperatură BMP280 si AHT21**

Cei 2 senzori (BMP280 [5] si AHT21 [6] ) au fost adaugați in proiect pentru a extinde gama de date colectate și pentru a monitoriza temperatura din cutia de control ,aceștia folosesc tot magistrala de date I2C , adăugarea acestora presupunând efort minim , iar alegerea celor 2 modele diferite ofera o redundanță sporita in colectarea datelor.

Optandu-se pentru o cutie din plastic cu capac transparent , aceasta fiind umpluta cu componente ce disipa caldură am constatat ca este absolut necesară o imbunatațire a racirii sistemului , temperatura maxima masurată cu senzorii fiind de 75°C , prag la care presupun că sistemul a suferit avarii ale modulelor de masură a curentului INA219 ( 2 din 4 module au fost avariate iremediabil ).

A black and white electronic device

Description automatically generatedIn figurile 2.1.3-1 si 2.1.3-2 se pot observa senzorii BMP280, respectiv AHT21.

Fig. 2.1.3-1 BMP280

Tensiune de alimentare: 3.3V DC

Protocol de comunicare: I²C

Adrese I²C: 0x77(default - jumper deschis) sau 0x76(jumper inchis)

Tip conectori I²C 3V3: SH 4P

Pitch conectori I²C 3V3: 1mm

A black and white circuit board

Description automatically generated

Fig. 2.1.3-2. AHT21

Tensiune de alimentare: 3.3V DC

Temperatura masurata: -40℃ ~ 120℃

Umiditate: 0~100 %RH

Protocol de comunicare: I²C

Adresa I²C: 0x38

Tip conectori I²C 3V3: SH 4P

Pitch conectori I²C 3V3: 1mm

### **2.1.4 Senzorii pentru masurarea curentului INA219**

4 senzori INA219 [7] cu adresa configurabila prin lipirea a 2 perechi de pini aflati pe acestia, alesi chiar datorita faptului ca suporta legarea a 4 senzori cu 4 adrese diferite pe aceeasi magistrala I2C acestia au generat un numar de probleme si obstacole ce au necesitat modificari majore asupra sistemului, plecand de la rezistenta shunt de .1Ω ce este subdimensionata pentru puterea disipata in aceasta , conform specificatiilor acest senzor poate masura o intensitate a curentului de pana la 3.2A , in cadrul rezistentei de .1 ohm , aseasta intensitate va genera o cadere de tensiune conform legii I=U/R de .32V (3.2A=U/.1), prin rezistor trecand 3.2A cu o cadre de .32V va rezulta o putere disipata in energie termica de 1.024W ( P=U\*I -> P=.32V\*3.2A), dupa atingerea unei temperaturi excesive rezistorii de .1ohm au capatat valori noi , necesitand inlocuirea, am optat pentru rezistori de . 1Ω dar capabili sa disipe 5W ( mult peste cei ~1W ce ii va atinge sistemul).  
  
O noua problema ridicata de acesti senzori , in urma unei supraincalziri a sistemului insusi integratele INA219 au fost afectate, din 4 numai 2 mai raporteaza valori corecte, unul fiind complet inaccesibil pe linia de I2C , iar celalalt raportand valori constante de 320mV pe rezistenta de shunt cu tensiunea circuitului la 0V , caracteristicile senzorilor detaliaza temperaturi de operare cuprinse intre -40 si 125 °C , si de stocare pana la 150°C , exista posibilitatea avarierii acestora in timpul inlocuirii rezistorilor de .1ohm , insa defectul nu a fost sesizabil imediat dupa reparatie.

In figura 2.1.4-1 se poate observa in coltul din dreapta-sus un modul nou, in partea stanga unul avariat iar in coltul din dreapta jos module INA219 cu rezistori shunt noi.

**A group of electronic components

Description automatically generated**Fig. 2.1.4-1 Senzor de curent bidirectional INA219

Rezistor pentru măsurarea curentului de 0.1 ohmi, 2 W, 1%;

Poate măsura curenți ce dau o cădere de tensiune de maxim 26V;

Poate măsura curenți de ±3.2A cu rezoluție de ±0.8 mA;

Folosește adresele I2C 0x40, 0x41, 0x44, 0x45 selectabile prin jumperi;

Tensiune de alimentare: 3V - 5.5V.

### **2.1.5 Puntile H**

2 punti H de mare putere, bazate pe integratul BTS7960 [8] (fig. 2.1.5-1) au fost folosite pentru a controla cele 2 motoare liniare ce directioneaza cadrul mobil prezent in proiect, acestea se controleaza in mod clasic folosind semnale digitale si PWM. Acestea nu au prezentat dificultati de integrare in sistem , acest fapt se atribuie simplitatii ce sta la baza operarii unor astfel de module.

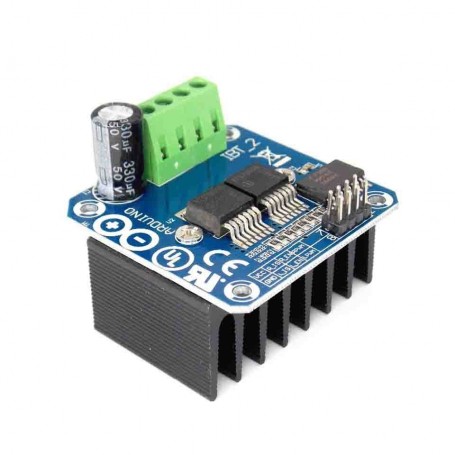


Fig. 2.1.5-1 Punte H de putere BTS7960

Tensiune alimentare: 5.5 - 27V

Tensiune logica: 5V

Protectie temperatura: Da

Protectie supra-tensiune: Da

Protectie curent: Da

Protectie scurt-circuit

Semnal: PWM sau 5V

Frecventa PWM: pana la 25KHz

Dimensiuni mm: 40 x 50 x 12mm

### **2.1.6 Coborator de tesiune XL4015**

A close-up of a circuit board

Description automatically generatedUn modul step down (fig. 2.1.6-1) bazat pe XL4015 [9] pentru coborarea tensiunii de la 12V( uneori chiar si 14V) provenite din circuitul realizat de panoul fotovoltaic la valori mai usor de gestionat , la bordul placilor de dezvoltare (Mega si ESP32) se regasesc regulatoare de tipul AMS1117 de 5V respectiv 3.3V , acestea disipand o cantitate considerabila de energie pentru a reduce tensiunea la valorile specificate, am optat in coborarea mai eficienta intr-un pas intermediar a tensiunii folosind acest modul din 2 motive :

Fig. 2.1.6-1 Coborator de tesiune XL4015

1. Regulatoarele AMS1117 disipa multa caldura punandu-le in pericol atat functionalitatea acestora cat si componentele ce depind de tensiunile stabile generate de ele
2. Pentru a beneficia de puterea maxima la tensiunile stabile de 5V , rspectiv 3.3 V , acestea se vor alimenta cat mai aproape de tensiunea finala+ caderea de tensiune din acestea (1V, maxim 1.3V). astfel XL4015 este setat sa converteasca cei 11.5V din acumulator ( tensunea minima, sub aceasta tensiune acumulatorul poate fi avariat daca se continua descarcarea) in 6.5V.

Tensiune intrare: 5 - 36VDC

Tensiune iesire: 1.25 - 32VDC, reglabila

Curent: intre 0 si 5 Ah

### **2.1.7 Modul microSD**

Modul microSD [10] (fig 2.1.7-1) folosit pentru colectarea datelor preluate de sistem :cantitatea de energie produsa , temperatura si umiditatea din cutia de control.

A black and silver microchip

Description automatically generated

Fig. 2.1.7-1 Modul microSD

Dimensiuni card microSD acceptate: 2GB ~ 32GB

Tensiune de alimentare: 3.3V sau 5V DC

Protocol de comunicare: SPI

### **2.1.8 Module de control incarcare solara PWM , RBL-30A**

Module RBL-30A [11] (fig. 2.1.8-1) folosite pentru controlul incarcarii acumulatorilor plumb acid AGM folositi in proiect , acest modul a fost ales datorita costurilor reduse si functionalitatilor multiple.

A blue and black solar charge controller

Description automatically generatedUn dezavantaj al acestor module fata de un modul ce foloseste MPPT (maximum power point tracking) este ca nu poate profita de toata energia produsa de panou , pentru funtionarea acestuia el necesita o tensiune de intrare de la panou mai mare decat cea a acumulatorului, un modul MPPT dispune de un convertor DC-DC de tip boost ce ridica in mod dinamic tensiunea livrata de panou peste cea a acumulatorului.

Fig. 2.1.8-1 Modul de control incarcare solara PWM

Model: RBL-30A

Tensiune baterie: 12 -24VDC

Curent descarcare: 10A maxim

Putere maxima: 360W la 12V, 720W la 24V

Tensiune panou maxima: 48V pentru baterie de 24V, 24V pentru baterie de 12V

Egalizare: 14.4V (Plumb Acid Sigilata), 14.2V (Baterie cu gel), 14.6V (baterie Plumb Acid nesigilata)

Tensiune absorbtie: 13.7V default, reglabila

Oprire descarcare: 10.7V default, reglabila

Reconectare la incarcare: 13V

Consum curent propriu modul: <10mA

Temperatura operare: -35 - +60 grade C

Protectii: Scurt circuit, circuit deschis, protectie inversa, protectie supra-sarcina

Protectie Curent Invers: Mosfet Dual

Dimensiuni mm: 133.5 x 70 x 35mm

### **2.1.9 Panouri solare 20W**

A solar panel with many black squares

Description automatically generated Panouri fotovoltaice [12] (fig. 2.1.9-1) monocristaline de 20W, rama de aluminiu , sticla de protectie securizata si cabluri de conectare atasate.

Fig. 2.1.9-1 Panou solar

Putere maxima:20W

Voltaj circuit deschis:22.3V

Voltaj maxim circuit inchis:18V

Curent de scurtciucuit: 1.2A

Curent maxim de lucru:1.11A

Dimensiuni: 450\*340\*20 mm

Greutate 1.55Kg

### **2.1.10 Actuatoare liniare**

A metal frame with metal tubes and wires

Description automatically generated with medium confidence 2 actuatoare liniare [13] (fig. 2.1.10-1) sunt folosite pentru a facilita miscarea de rotatie in jurul axelor X si Y.

Fig. 2.1.10-1 Actuatoare liniare

Cursa actuatorului:100mm

Tensiune alimentare: 12V

Forta maxima:1100N

Viteza de actiune: 13mm/s

Regim utilizare recomandat :10% (10 secunde ulilizare , 90 secunde idle)

### **2.1.11 Modul SENZOR 3 AXE ADXL345**

Modul accelerometru pe 3 axe [14] (fig. 2.1.11-1) pentru determinarea pozitiei panoului solar , aceasta abordare pune bazele unei implementari mult mai tehinice a pozitionarii panoului, in cazul actual va fi folosit pentru a reseta pozitia panoului de la o zi la alta si protectia impotiva grindinei.

Dat fiind faptul ca modulul va indeplinii functii critice s-a ales protejarea lui prin acoperirea cu lipici termic in lipsa unei metode de protectie mai adecvate,motoarele liniare nu au puncte de oprire reglabile, cadrul mobil isi va atinge limitele de miscare inaintea motoarelor, valori hardcodate ale inclinatiei panoului vor fi stabilite ca masura de protectie pentru a preveni avarierea motoarelor sau a cadrului.

A close-up of a blue circuit board

Description automatically generated

Fig. 2.1.11-1 Senzor ADXL345

Tensiune alimentare: 3.3 - 5VDC

Integrat: ADXL345

Integrat putere: RT9161

Interfata: I2C, SPI

Dimensiuni mm: 28 x 14mm

### **2.1.13 TRANSLATOR NIVEL LOGIC I2C IIC BIDIRECTIONAL 8 CANALE 3.3V 5V TXS0108E**

Modul bazat pe integratul TXS0108E [15] (fig. 2.1.13-1) ce permite trecerea de la nivelul logic de 3.3V la 5V , folosit in principal pentru protejarea componentelor mai sensibile aflate pe magistrala de I2C, in proiect se regasesc atat componente ce functioneaza la nivel logic de 5V cum ar fi : ADXL345 si ADS1015, acestea beneficiind chiar de voltajul mai mare ele aflandu-se la o distanta mai mare de controller .

A blue circuit board with black pins

Description automatically generated

Fig. 2.1.13-1 Translator nivel logic

-nivelul logic de 5V: INA219,ADXL345,Mega2560

-nivel logic de 3.3V : AHT21,BMP280,ecranul OLED, ESP32

VCCA se conecteaza la sursa 3.3V

VCCB se conecteaza la sursa 5V

GND se conecteaza in comun la cele 2 surse de 3.3 si 5V

Cand Ax are input TTL 3.3V, Bx are output TTL 5V

Cand Bx are input TTL 5V, Ax are output TTL 3.3V

NU necesita control de directie

Pentru aceasta am creat un ansamblu format din:

* 2 panouri fotovoltaice de 20W, unul montat intr-un cadru fix iar cel de-al doilea panou pe un cadru mobil pe 2 axe , acesta țintind sa creeze un unghi de 90 de grade între panou și razele soarelui constant.
* 2 acumulatori de tip AGM ,12V 9.1Ah 109Wh ,aceștia vor ajuta sistemul sa facă față cererilor de energie ce depașesc capacitatea panourilor și vor alimenta sistemul când energia solară este indisponibilă.
* 2 controlere PWM pentru incărcarea acumulatorilor și protecția acestora
* Arduino Mega 2560 pentru colectarea de date de la diverși senzori analogici și digitali, controlul deciziilor logice din cadrul ansamblului și acționarea diverselor actuatoare.
* un modul ESP32 pentru funcționalitațile de colectare de date si conectarea la internet a proiectului (îmbunatațirea conectivitații proiectului folosind acest modul)
* coborâtor de tensiune liniar reglabil ( pentru a nu suprasolicita regulatoarele de tensiune de pe arduino si ESP , acestea avand pierderi semnificative de energie la tensiuni ridicate )
* 2 punți H de mare putere pentru controlul motoarelor liniare de pe cadrul mobil.
* 2 senzori de temperatură ,BMP280 ( temperatură + presiune atmosferică) si AHT21 ( temperatură + umiditate aer)
* un stabilizator de tensiune bazat pe LM7805 ( 5V, 1.5A) folosit pentru alimentarea cu 5V a componentelor sensibile si ce pot induce perturbații ale tensiunii in sistem , ex.: releu , leduri )
* 2 ventilatoare cu turbină(12V, 130mA) pentru racirea componentelor (in mod special punțile H si stabilizatorul de tensiune LM7805)
* 2 module pentru măsurarea cantitații de energie produse de panouri
* un modul microSD pentru colectarea de date.

Drept urmare am obținut un sistem ce poate fi adaptat, dimensionat conform nevoilor,ușor de menținut si imbunatațit cu piese ușor de obținut.

## **2.2 Echipamente software**

### **2.2.1 Arduino IDE**

Mediu de programare Arduino IDE [16] (integrated development environment) ajuta la scriera usoara a codului si incarcarea acestuia in placile de dezvoltare , gandit pentru a fi folosit impreuna cu placile din gama Arduino, el nu se limiteaza la acestea, codul de la baza programului fiind opensource s-au integrat librarii ce permit programarea a mai multor placi de dezvoltare existente pe piata.

Arduino IDE pune la dispozitie unelte ce faciliteaza integrarea de librarii foarte usoara, realizand chiar si o distinctie intre librariile pentru programarea placilor de dezvoltare (fig. 2.2.1-1) si librariile pentru integrarea modulelor in proiecte (fig. 2.2.1-2) , astfel reducand riscul de a lua versiuni vechi de librarii sau programe malicioase in incercra de a procura resursele necesare integrarii modulelor in proiectele persoanale dezvoltate de pasionati de electronica de pe intreg globul.

A screenshot of a computer

Description automatically generated

Fig. 2.2.1-1 Meniul pentru managementul placilor de dezvoltare

A screenshot of a computer

Description automatically generated

Fig. 2.2.1-2 Meniul pentru managementul librariilor

### **2.2.2 GitHub**

GitHub [17] este o platforma ce gazduieste partea digitala a proiectelor si permite contolul detaliat al versiunilor realizate , colaborarea intre multiple entitati la realizarea proiectelor de ampoare fara a necesita contact fizic intre aceste entitati .

Unul dintre cele mai mari avantaje ale platformei este controlul versiunii si contributia sa majora in procesul de integrare si tetare continua , ajungand sa contribuie si la procesul de esec rapid ce presupune descoperirea in fazele incipiente ale problemelor existente.

Un aspect important este conceptul de “branch” ( ramura) , acesta presupune vizualizarea proiectului in module interconectate , ce pot fi dezvoltate separat si unite intr-o entitate principala (main) la data livrarii proiectelor (fig. 2.2.2-1) .

Alt aspect important este modul in care schimbarile se implementeaza in cadrul unui proiect, cu fiecare noua schimbare se creza o noua ramura (fig. 2.2.2-2) ce este dezvoltata si testata separat de proiectul principal , dupa finalizarea implementarii si testarii unei functionalitati ea este imbinata cu ramura principala.

Printre cele mai importante comenzi din git se numara :

* git init – initializeaza un fisier local de gestiune ce va fi sincronizat cu un server pentru a urmari schimbarile efectuate asupra unui set de fisiere
* git clone- aduce fisiere stocate intr-un repository aflat la o locatie la distanta
* git add – adauga fisiere noi in sistemul de urmarire a schimbarilor la nivel local
* git commit- adauga fisiere noi in sistemul de urmarire a schimbarilor pe serverul aflat la distanta accesibil tuturor participantilor la proiect
* git merge – imbina schimbarile facute cu ramura principala

A screenshot of a web page

Description automatically generated

Fig. 2.2.2-1 Management-ul branch-urilor in git

A diagram of a diagram

Description automatically generated

Fig. 2.2.2-2 Procesul de dezvoltare si integrare in Git

### **2.2.3 Limbajul de programare C++ în Arduino IDE**

Limbajul C++ este creaţia lui Bjarne Stroustrup şi reprezintă o extensie a limbajului C care permite programarea pe obiecte.Dezvoltarea acestuia a început in 1979 , cu denumirea inițiala de “C cu obiecte” acesta a ajuns să capete mult mai multe funcționalitați pe langă programarea orientată pe obiecte, ajungând sa fie redenumit ulterior C++ facând referire la operatorul de incrementare ++ acesta este un mod unic de a spune că C++ este o revizie incrementală a lui C.

Structura generală a unui program C++:

* un program C++ este constituit din funcţii, una dintre aceste funcţii este funcţia principală, denumită main()

int main()

{

cout<<"Hello World";

return 0;

}

* main() este o funcţie specială, care trebuie să apară obligatoriu o singură dată în orice program C++
* execuţia oricărui program începe cu funcţia main()
* o funcţii este constituită din antet şi corp antetul funcţiei conţine numele funcţiei, tipul rezultatului pe care îl calculează funcţia şi o listă de parametri prin care funcţia comunică cu exteriorul ei, încadrată între paranteze rotunde
* corpul funcţiei conține declarații și instrucțiuni care specifică prelucrările realizate de funcția respectivă

Instrucţiunea return este utilizată pentru a încheia execuţia unei funcţii şi a returna valoarea expresiei specificate în instrucţiunea return ca valoare a funcţiei.  
Vocabularul limbajului C++ este format din:

* setul de caractere
* identificatori
* cuvinte cheie
* comentarii
* separatori

Programarea C++ in Arduino Studio IDE este putin diferita fata de programarea C++ clasica. Diferenta intre cele doua este ca in Arduino Studio IDE nu mai avem clasica functie main, aceasta fiind inlocuita de doua funcții: setup și loop.

Funcția setup se executa mereu prima și aceasta se executa doar o dată, la inceputul programului, in schimb funcția loop se execută dupa funcția setup, însa aceasta se reapelează pe ea insași la infinit.

De obicei in funcția setup se fac inițializarile de variabile, clase, componente, etc., iar în funcția loop se execută codul care se doreste să fie executat mai mult de o singură dată.[3]

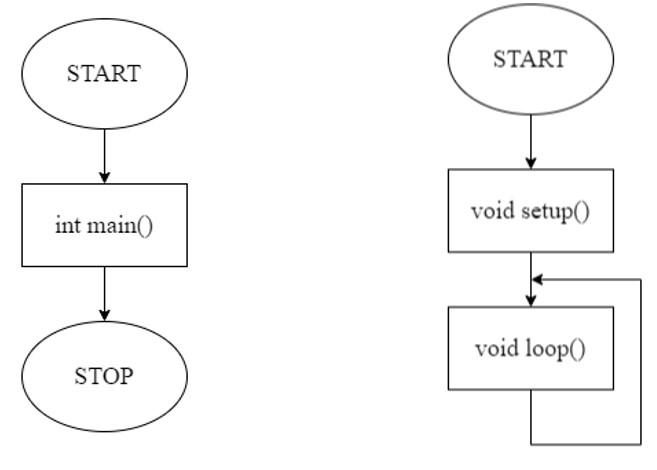


Fig. 2.2.3-1 Structura programului C++ Fig. 2.2.3-2 Structura programului în Arduino IDE

### **2.2.4 I2C**

Comunicarea prin interfața I2C [18] este o opțiune eficienta pentru transmiterea de date intre 2 microcontrolere , aceasta este și foarte practica punând la dispoziție o lațime de bandă de până la 3.4mb/s in modul de mare viteză, în cadrul acestui proiect s-a evitat folosirea unei viteze de transfer aproape de plafonul maxim pentru sporirea distanței la care este posibilă realizarea unei conexiuni stabile pe linia de comuicație.

Din cauza conductorilor folosiți in cadrul liniei de I2C odată cu cresterea lungimii acestora apare și efectul de inductor ce se opune schimbarilor ce țin de câmpul electromagnetic , în mod normal semanlele digitale pot calători pe distanțe de ordinul zecilor de metrii dar în cadrul unei linii I2C starea acestei linii se schimă de la HIGH la LOW chiar și de 400.000 de ori pe secundă astfel chiar și cea mai mică forța reactivă ( in cazul acesta inducția din cabluri) poate perturba interpretarea semnalelor între dispozitivul Master si Slave-urile acestuia.

Protocolul de comunicare I2C [18] (Inter-Integrated Circuit) este un protocol serial sincron folosit pentru comunicația între circuite integrate (ICs) pe o placă de bază (board) sau între placi de bază. Protocolul I2C utilizează doua linii pentru comunicație:

- SDA (Serial Data): Aceasta este linia de date serială bidirecțională pe care informațiile sunt transmise in forma seriala intre dispozitivele conectate. Linia SDA este controlată de dispozitivul master, care inițiaza și controlează comunicația.

- SCL (Serial Clock): Aceasta este linia de ceas folosită pentru a sincroniza transferul datelor între dispozitivele conectate. Semnalele de pe linia SCL sunt controlate de dispozitivul master si sunt utilizate pentru a sincroniza transferul de biți pe linia SDA.

Datorita faptului ca adresele I2C sunt in mod normal pe 7 biți , intr-o rețea se pot conecta pâna la 128 de dispozitive, dar unele adrese sunt rezervate coborând numarul de adrese disponibile sub 128 .

O mențiune importantă este existența integratelor cu adrese I2C pe 10 biți, ducând astfel numarul de componente suportate intr-o rețea aproape de 1024 , teoretic .

Protocolul funtioneaza intr-un mod simplu, liniile SDA si SCL sunt tinute la nivel high de catre rezistori de pull up, cu valori normale intre 4.7 si 10k ohm , atunci cand un dispozitiv doreste sa inceapa comunicatia acesta trage prima data linia de SDA in jos , urmand sa traga apoi linia de SCL in jos .Pentru oprirea comunicaitiei, in momentul cand linia de clock (SCL) se afla pe HIGH , linia de date (SDA) este trasa tot pe high de catre dispozitivul ce doreste oprirea comunicatiei, din acest motiv standardul I2C impune ca tranzitiile sa fie efectuate pe linia de date doar cand linia de clock este la nivel LOW comportament vizibil in figura 2.2.4-2 .

In figura 2.2.4-1 se poate observa comportamentul celor 2 linii la inceputul si la sfarsitul transmisiei de date intre 2 dispozitive.

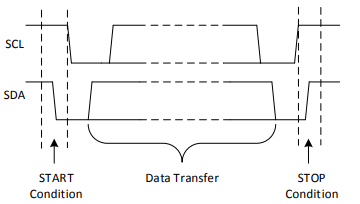


Fig. 2.2.4-1 Tranzitiile la inceputul si sfarsutul comunicatiei in I2C

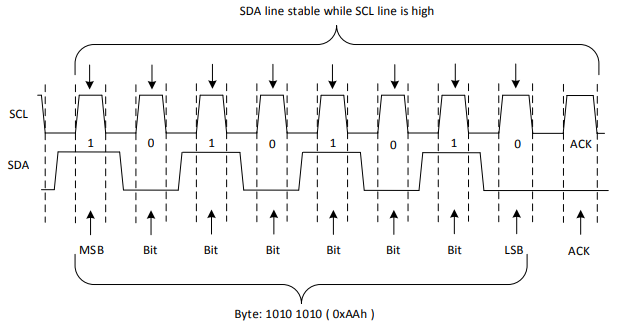


Fig. 2.2.4-1 Transmisie de date prin I2C

# **3 Arhitectura Sistemului**

## **3.1 Diagrama de stari**

O diagrama de stare [19] consta in stari,tranzitii intre acestea si evenimentele ce cauzeaza tranzitiile , diagramele sunt folosite pentru a ilustra comportamentul dinamic al sistemului.

Elementele importante ale unei diagrame de stare sunt :

* Stairle – reprezinta statusul sistemului intr-un moment si loc specific
* Evenimentele – semnaleaza momentul cand o tranzitie trebuie sa aiba loc
* Tranzitiile – semnifica actiunea de a schimba starea curenta

In figura3.1-1 se pot observa starile prin care trece sistemul in timpul functionarii normale, acestea descriu in linii mari comportamentul sistemului atunci cand diversi parametrii sunt atinsi, predominant fiind vorba de apelul functiilor recurente din interiorul buclei infinite (funtia loop) din arduino.

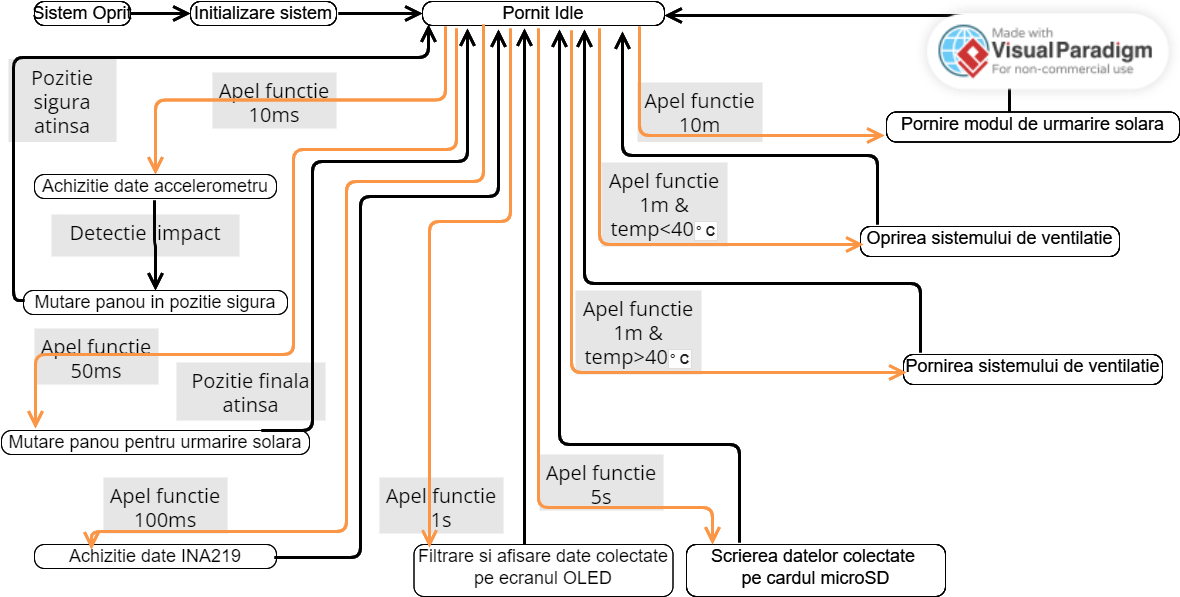


Fig. 3.1-1 Diagrama de stari

## **3.2 Schema structurii sistemului**

Figura 3.2-1 [20] prezinta schema structurii Hardware si Software, impreuna cu interactiunea dintre diversele module .

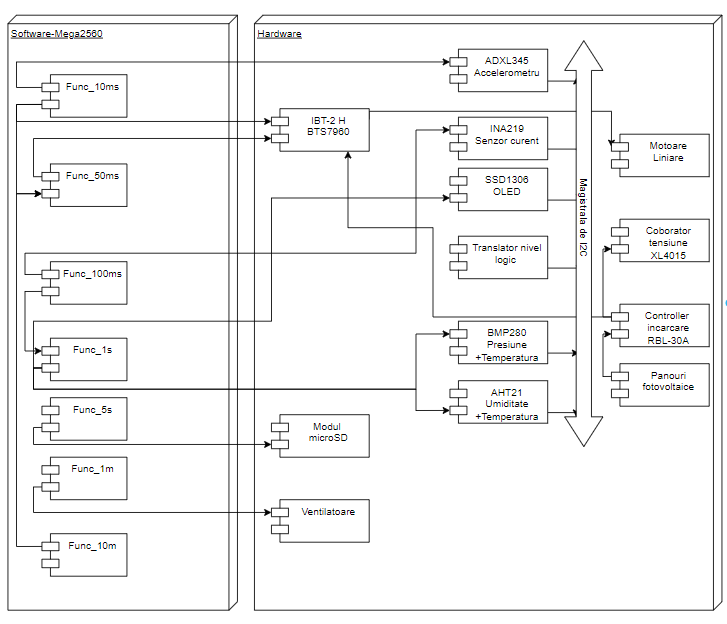


Fig. 3.2-1 Diagrama module

## **3.3 Schema structurii sistemului**

Figura 3.3-1 contine conexiunile din proiect. Am ales aceasta metoda in avantajul unei scheme electrice complete datorita limitarilor formatarii acestei lucrari , consider ca varianta aleasa este mai potrivita pentru contextul acestui document.

Figura 3.3-2 functioneaza ca si completare a datelor din tabelul din fig. 3.3-1.

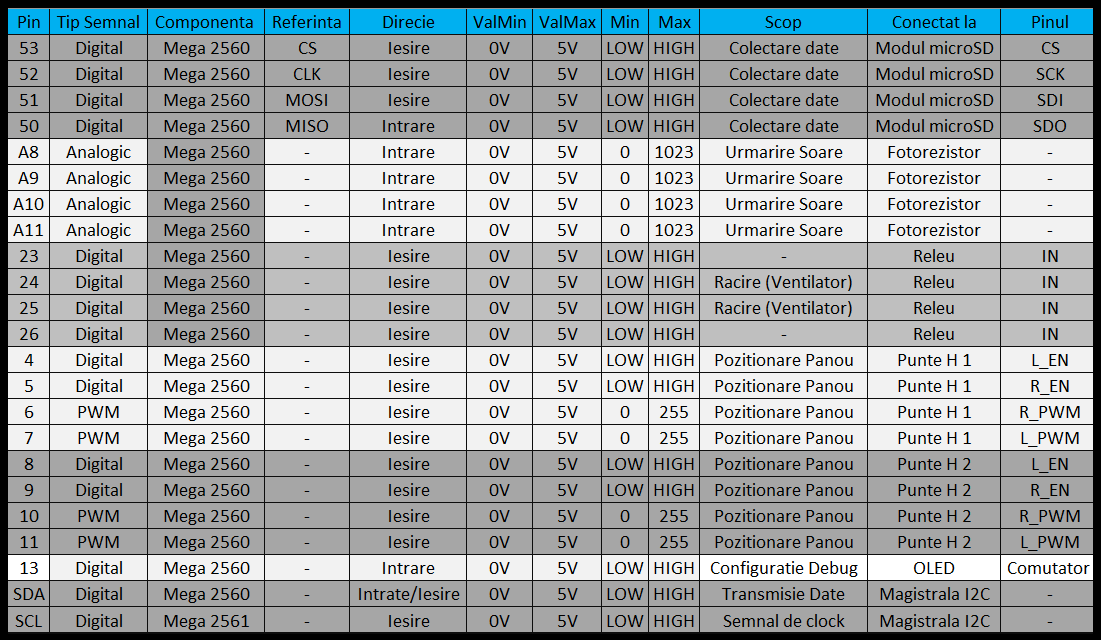


Fig. 3.3-1 Conexiuni sistem la nivel logic

A circuit board with many colored wires

Description automatically generated

Fig. 3.3-2 Pinout Mega 2560

# **4. Codul de la baza proiectului**

Pentru implementarea sistemului , am fractionat codul in mai multe fisiere, la baza regulii de impartire a codului fiind programarea in timp real , intreg sistemul bazandu-se pe acest concept prin care nu exista timpi in care microcontroller-ul sa astepte rezultate.

#include <Wire.h>// librarie ce faciliteaza comunicarea prin I2C

#include <Adafruit\_INA219.h>

#include <Adafruit\_Sensor.h>

#include "Adafruit\_BMP280.h"

#include <SPI.h>

#include <SD.h>

#include <Adafruit\_GFX.h

#include <Adafruit\_SSD1306.h

#include <Adafruit\_AHTX0.h>

#include <Adafruit\_ADXL345\_U.h>

**Wire.h este o librarie ce faciliteaza comunicarea prin I2C**

**Adafruit\_INA219.h pentu colectatrea datelor de la senzorii de intensitate**

**Adafruit\_Sensor.h folosita pentru senzorii AHT21/BMP280**

**Adafruit\_BMP280.h folosita pentru controlul BMP280**

**SPI.h protocolul SPI este folosit doar pentru modulul microSD**

**SD.h librarie folosita pentru a scrie pe cardul MicroSD**

Adafruit\_GFX.h ajuta la pozitionarea textului pe ecranul OLED

Adafruit\_SSD1306.h pentru controlul ecranului OLED

Adafruit\_AHTX0.h pentru prelucrarea datelor provenite de la AHT21

Adafruit\_ADXL345\_U.h pentru prelucrarea datelor provenite de la accelerometru

In interiorul buclei ce ruleaza in mod repetat dupa initializarea, microcontroller-ului se afla numai apelurile catre functiile recurente si codul aferent apelarii acestora la intervale regulate de timp , exista posiblitatea ca o sarcina de lucru sa interfereze cu apelul acesteia ( ex. in interiorul functiei cu recurenta la 10ms , codul rulat sa tina microcontroller-ul ocupat pentru 15ms, caz in care apelul urmatoarei functii se va face o data la 15ms in loc de 10ms) , acest comportament a fost realizat prin plasarea codului ce este costisitor din punct de vedere al timpului de executie in functii cu recurenta mai scazuta.

void loop(void)

{

  actual\_time=millis();

  if(actual\_time-prev\_time\_10ms>=task\_10ms){    func\_10ms();

    prev\_time\_10ms=actual\_time;

  }

  if(actual\_time-prev\_time\_50ms>=task\_50ms){    func\_50ms();

    prev\_time\_50ms=actual\_time;

  }

  if(actual\_time-prev\_time\_100ms>=task\_100ms){    func\_100ms();

    prev\_time\_100ms=actual\_time;

  }

  if(actual\_time-prev\_time\_1s>=task\_1s){    func\_1s();

    prev\_time\_1s=actual\_time;

  }

  if(actual\_time-prev\_time\_5s>=task\_5s){    func\_5s();

    prev\_time\_5s=actual\_time;

  }

  if(actual\_time-prev\_time\_1m>=task\_1m){    func\_1m();

    prev\_time\_1m=actual\_time;

  }

  if(actual\_time-prev\_time\_10m>=task\_10m){    func\_10m();

    prev\_time\_10m=actual\_time;

  }

  if(actual\_time-prev\_time\_1h>=task\_1h){    func\_1h();

    prev\_time\_1h=actual\_time;

  }

}

Un prim exemplu de cod folosit in evenimente recurente este cel pentru determinarea impactului in cazul grindinei ce poate distruge panourile , folosind datele de la senzorul ADXL345 sistemul poate detrmina in cazul unui impact schimbarea brusca a valorilor vectorilor de acceleratie masurati de senzori , caz in care un indicator ( impactDetectat) va fi activat , punand sistemul intr-o stare de alarma , repozitionand panoul astfel in cat orice impact cu bucatile de gheata sa fie la un unghi cat mai mic , micsorand astfel sansele avarierii panourilor.

void func\_10ms()

{

total\_X=0;

  total\_Y=0;

  total\_Z=0;

  for(int i=0;i<10;i++){

    total\_X=total\_X+achizitii\_X[i];

    total\_Y=total\_Y+achizitii\_Y[i];

    total\_Z=total\_Z+achizitii\_Z[i];

  }

  sensors\_event\_t event;

  accel.getEvent(&event);

  X\_out = event.acceleration.x;

  Y\_out = event.acceleration.y;

  Z\_out = event.acceleration.z;

  achizitii\_X[indexSample\_ADXL]=X\_out;

  achizitii\_Y[indexSample\_ADXL]=Y\_out;

  achizitii\_Z[indexSample\_ADXL]=Z\_out;

  if(pozFinala){

    if(X\_out-2.0>total\_X/10.0 || X\_out+2.0<total\_X/10.0 ||

     Y\_out-2.0>total\_Y/10.0 || Y\_out+2.0<total\_Y/10.0  ||

     Z\_out-2.0>total\_Z/10.0 || Z\_out+2.0<total\_Z/10.0)

     {      impactDetectat=true;     }

  }

  if(indexSample\_ADXL<9){    indexSample\_ADXL=indexSample\_ADXL+1;

  }else{    indexSample\_ADXL=0;  }

  if(impactDetectat){

      digitalWrite(motor1Ren,HIGH);

      analogWrite(motor1R\_PWM,0);

      digitalWrite(motor1Len,HIGH);

      analogWrite(motor1L\_PWM,225);

  }

}

Datele colectate sunt prelucrate tinand cont de abaterea de la medie a ultimelor 10 valori citite , in cazul unei erori mai mare de 2m/s^2 , se vor lua masurile necesare protejarii panourilor.

Datorita importantei protejarii panourilor, aceasta bucata de cod beneficiaza de cel mai mult timp de executie , ea fiind rulata de 100 de ori pe secunda pentru a actiona la primul impact .

Urmatoarea rutina ce se executa este cea de 50ms , aceasta are ca rol pozitionarea panoului mobil , tintind creerea unui unghi de 90 de grade cu razele soarelui , folosind valori provenite de la un ansamblu de fotorezistori in cadrul rutinei motoarele liniare sunt controlate cu ajutorul puntilor H existente in proiect , ce la randul lor sunt controlate folosind semnale digitale si PWM , pentru o miscare mai lina si controlata, factorul de umplere nu este setat la 100% (255) , am ales valoarea de 225 ( ~88%) pentru a alimenta motoarele cu aproximativ 10.5V , ele fiind supradimensionate pentru dimensiunile ansamblului).

void func\_50ms()

{

  //===============urmarire solara

  punct1=analogRead(adc1);  punct2=analogRead(adc2);  punct3=analogRead(adc3);

  punct4=analogRead(adc4);

  if(!pozFinala && !impactDetectat){

    //control M2

    if(punct1>900 ){

      digitalWrite(motor2Ren,HIGH);      analogWrite(motor2R\_PWM,0);

      digitalWrite(motor2Len,HIGH);      analogWrite(motor2L\_PWM,0);

      pozFinala=true;

    }else if(punct4<900){

      digitalWrite(motor2Ren,HIGH);      analogWrite(motor2R\_PWM,225);

      digitalWrite(motor2Len,HIGH);      analogWrite(motor2L\_PWM,0);

    }else if (punct3<900) {

      digitalWrite(motor2Ren,HIGH);      analogWrite(motor2R\_PWM,0);

      digitalWrite(motor2Len,HIGH);      analogWrite(motor2L\_PWM,225);

    }else{

      digitalWrite(motor2Ren,HIGH);      analogWrite(motor2R\_PWM,0);

      digitalWrite(motor2Len,HIGH);      analogWrite(motor2L\_PWM,0);

    }

    if(punct1>900 ){

      digitalWrite(motor1Ren,HIGH);      analogWrite(motor1R\_PWM,0);

      digitalWrite(motor1Len,HIGH);      analogWrite(motor1L\_PWM,0);

      pozFinala=true;

    }else if(punct3<900){

      digitalWrite(motor1Ren,HIGH);      analogWrite(motor1R\_PWM,225);

      digitalWrite(motor1Len,HIGH);      analogWrite(motor1L\_PWM,0);

    }else if(punct2<900){

      digitalWrite(motor1Ren,HIGH);      analogWrite(motor1R\_PWM,0);

      digitalWrite(motor1Len,HIGH);      analogWrite(motor1L\_PWM,225);

    }else{

      digitalWrite(motor1Ren,HIGH);      analogWrite(motor1R\_PWM,0);

      digitalWrite(motor1Len,HIGH);      analogWrite(motor1L\_PWM,0);

    }

  }

}

Rutina de 100ms, in cadrul acesteia se abordeaza masurarea productiei de energie de la cele 2 panouri solare , aici se face colectarea datelor neprelucrate , intensitatea si tensiunea curentrului.

In aceasta functie se mai creeaza si o animatie in coltul din dreapta al ecranului OLED, aceasta servind ca indicator pentru functionarea corecta a codului .

void func\_100ms()

{

  //=======================================================================================================INA219

  shuntvoltage\_0x40 = ina219\_0x40.getShuntVoltage\_mV();

  busvoltage\_0x40 = ina219\_0x40.getBusVoltage\_V();

  current\_mA\_0x40 = ina219\_0x40.getCurrent\_mA();

  power\_mW\_0x40 = current\_mA\_0x40\*busvoltage\_0x40;//ina219\_0x40.getPower\_mW();

  loadvoltage\_0x40 = busvoltage\_0x40 + (shuntvoltage\_0x40 / 1000);

  shuntvoltage\_0x44 = ina219\_0x44.getShuntVoltage\_mV();

  busvoltage\_0x44 = ina219\_0x44.getBusVoltage\_V();

  current\_mA\_0x44 = ina219\_0x44.getCurrent\_mA();

  power\_mW\_0x44 = busvoltage\_0x44\*current\_mA\_0x44;//ina219\_0x44.getPower\_mW();

  loadvoltage\_0x44 = busvoltage\_0x44 + (shuntvoltage\_0x44 / 1000);

  achizitii\_0x44\_mA[indexSample]=fabs(current\_mA\_0x44);

  achizitii\_0x44\_mW[indexSample]=power\_mW\_0x44;

  achizitii\_0x44\_V[indexSample] =busvoltage\_0x44;

  achizitii\_0x40\_mA[indexSample]=fabs(current\_mA\_0x40);

  achizitii\_0x40\_mW[indexSample]=power\_mW\_0x40;

  achizitii\_0x40\_V[indexSample] =busvoltage\_0x40;

  if(indexSample<9){

    indexSample=indexSample+1;

  }else{

    indexSample=0;

  }

  //======================================================================================================INA219

  //==========================================desenez un cerc pe OLED , feedback la rularea codului

  display.fillCircle(62+cursor,60,2,WHITE);

  display.display();

  if(cursor<64){

    cursor=cursor+8;

  }else{

    cursor=0;

  }

}

In cadrul rutinei de 1s au loc atat filtrarea datelor provenite de la senzorii INA219,preluarea datelor de la senzorii de temperatura AHT21 si BMP280, cat si afisarea parametrilor sistemului pe ecranul OLED.

Filtrarea datelor se face prin obtinerea unei medii a 10 valori colectate in rutina de 100ms , pe ecranul OLED afisandu-se valorile filtrate, afisarea fiecarei valori colectate facand urmarirea ecranului foarte dificila , astfel afisand mediile se pastreaza o rata de improspatare a datelor pe ecran de o secunda.

void func\_1s()

{

  pressure = bmp.readPressure();

  temperature = bmp.readTemperature()-1.0;

  aht.getEvent(&humidity, &temp);

  total\_0x40\_mA=0;

  total\_0x40\_mW=0;

  total\_0x40\_V =0;

  total\_0x44\_mA=0;

  total\_0x44\_mW=0;

  total\_0x44\_V =0;

  for(int i=0;i<10;i++){

    total\_0x40\_mA=total\_0x40\_mA+achizitii\_0x40\_mA[i];

    total\_0x40\_mW=total\_0x40\_mW+achizitii\_0x40\_mW[i];

    total\_0x40\_V =total\_0x40\_V+achizitii\_0x40\_V[i];

    total\_0x44\_mA=total\_0x44\_mA+achizitii\_0x44\_mA[i];

    total\_0x44\_mW=total\_0x44\_mW+achizitii\_0x44\_mW[i];

    total\_0x44\_V =total\_0x44\_V+achizitii\_0x44\_V[i];

  }

display.clearDisplay();

    display.setTextSize(1);

    display.setCursor(0  ,0  );display.print("MCU\_0x44:");

    display.setCursor(0  ,8  );display.print(round(total\_0x44\_mW/10));

    display.setCursor(40 ,8  );display.print("mW");

    display.setCursor(0  ,16 );display.print(total\_0x44\_mA/10.0);

    display.setCursor(40 ,16 );display.print("mA");

    display.setCursor(0  ,24 );display.print(total\_0x44\_V/10.0);

    display.setCursor(40 ,24 );display.print("V");

    display.setCursor(0  ,32 );display.print("X.Y\_0x40:");

    display.setCursor(0  ,40 );display.print(round(total\_0x40\_mW/10));

    display.setCursor(40 ,40 );display.print("mW");

    display.setCursor(0  ,48 );display.print(total\_0x40\_mA/10.0);

    display.setCursor(40 ,48 );display.print("mA");

    display.setCursor(0  ,56 );display.print(total\_0x40\_V/10.0);

    display.setCursor(40 ,56 );display.print("V");

    display.setCursor(60 ,0 );display.print("Presiune:");

    display.setCursor(60 ,8 );display.print(pressure);

    display.setCursor(115,8 );display.print("Pa");

    display.setCursor(60 ,16 );display.print("Temp:");

    display.setCursor(60 ,24 );display.print("T1:");

    display.setCursor(80 ,24 );display.print(temperature);

    display.setCursor(118,24 );display.print("C");

    display.setCursor(60 ,32 );display.print("T2:");

    display.setCursor(80 ,32);display.print(temp.temperature);

    display.setCursor(118,32);display.print("C");

    display.setCursor(60 ,40 );display.print("Umiditate:");

    display.setCursor(60 ,48 );display.print(humidity.relative\_humidity);

    display.setCursor(118,48 );display.print("%");

    display.display();

cursor=0;

}

In cadrul rutinei de 5 secunde se colecteaza datele sistemului pe cardul microSD , pentru a face mai usoara prelucrarea acestea sunt scrise fara unitati de masura, aranjate in grupuri cu linii goale intre ele , pot fi prelucrate ulterior pentru a obtine grafice ale productiei de energie corespunzatoare fiecarui panou, variatei de temperatura ,presiune si umiditate din cutia de control .

void func\_5s()

{

  if (!SD.begin(pinCS))

  {

    //Serial.println("SD card initialization failed");

  }

  myfile = SD.open("Date.txt", FILE\_WRITE);

   if (myfile)

  {

    myfile.println(round(total\_0x44\_mW/10));

    myfile.println(total\_0x44\_mA/10.0);

    myfile.println(total\_0x44\_V/10.0);

    myfile.println(round(total\_0x40\_mW/10));

    myfile.println(total\_0x40\_mA/10.0);

    myfile.println(total\_0x40\_V/10.0);

    myfile.println(pressure);

    myfile.println(temperature);

    myfile.println(temp.temperature);

    myfile.println(humidity.relative\_humidity);

    myfile.println();

    myfile.println();

    myfile.close();

  } else

  {

    //Serial.println("error opening the text file!");

  }

}

O data la un minut temperatura masurata este folosita pentru a porni sau opri ventilatoarele.

void func\_1m()

{

   if(temperature>40.00){

      digitalWrite(pinReleuVentIna219,HIGH);

      digitalWrite(pinReleuVentESP,HIGH);

   }else {

      digitalWrite(pinReleuVentIna219,LOW);

      digitalWrite(pinReleuVentESP,LOW);

   }

}

Ultima rutina folosita este cea de 10 minute, in cadrul acesteia se reactiveaza urmarirea solara pentru a repozitiona panoul , soarele avanseaza pe cer cu viteza de 2.5 grade o data la 10 minute , am considerat ca o corectie a pozitiei panoului in acest interval ar fi potrivita .

void func\_10m()

{

 pozFinala=false;// soarele isi schimba unghiul cu 15 grade pe ora, deci 2.5 grade o data la 10 minute

}

# **5. Rezultatele Obtinute**

In urma unui rulaj de o zi in data de 02.09.2023 am optat pentru a exporta datele colectate din Mega 2560 in cardul micro SD o data la 5 minute prin alterarea temporizatorului de la functia de 10 minute din proiect, exportul datelor la 5 secunde ar fi generat o cantitate de date ce depaseste capabilitatile laptopului folosit in prelucrarea datelor.

A graph of a graph showing the time and the time

Description automatically generated with medium confidenceAm reusit sa prelucrez grafice pe durate maxime de o ora cu rata de achizitie a datelor la 5 secunde ( 720 de achizitii de date/ora) , limitand achizitia de date la 5 minute ( 12 achizitii/ora) am generat graficul comparativ din figura 5-1.

Fig. 5-1 Rezultate Experiment 1

Experimentul 2 a avut loc ziua urmatoare , cand conditiile meteo nefavorabile (cer innorat si ploaie) au aratat punctele slabe ale sistemului, valorile din modulul software ce se ocupa cu achizitia datelor de pe fotorezistori au fost setate intr-o zi cu cer senin, valorile de la nivelul fotorezistorilor din data de 2.09.2023 au impiedicat panoul mobil sa plece din pozitia initiala in care asteapta rasaritul.Generand astfel rezultatele din graficul 5-2.

A graph with blue and orange lines

Description automatically generated

Fig. 5-2 Rezultate Experiment 2

# **6. Concluzii și direcții de dezvoltare**

## **6.1 Concluzii**

Sistemul realizat a trecut prin mai multe implementari decat am prevazut, pornindu-se de la ideea de baza ce presupunea urmarirea soarelui folosind un panou fotovoltaic s-a ajuns la forma actuala ce nu se limiteaza la o singura functionalitate , plecand de la o idee cu scopul implementarii acesteia pe instalatiile fotovoltaice de mici dimensiuni am ajuns la un spatiu de testare pentru functionalitati noi, cu directii de dezvoltare multiple.

Facand posibila testarea si compararea rezutatelor optinute intr-o multitudine de situatii plecand de la simpla compararare simultana a productiei energetice a unui panou fix si unul mobil , diferite tipuri de panouri , diferite tipuri de controlere pentru sistemele fotovoltaice, etc.

La realizarea proiectului am întampinat și unele probleme, care totuși au fost rezolvate. Câteva dintre problemele întâmpinate sunt:

* senzorii INA219 necesita inlocuirea cu componente mai robuste , sunt predispusi defectiunilor
* lipsa de experiență de lucru cu ESP32 ce a introdus delay-uri si intarzierea livrarii unui proiect functional,livrarea functionalitatilor bazate pe acesta fiind amanate

In cadrul proiectului exista si un numar considerabil de implementari reusite la nivel hardware cat si software printre care se afla:

* Sistemul functioneaza pe baza unui sistem de operare in timp real, in cadrul acestuia codul rulandu-se in mod continuu, fara intarzieri ,permitand astfel placii de dezvoltare Mega 2560 sa furnizeze un flux de date cu o precizie acceptabila timp de peste 30 de zile fara a necesita interventii
* Implementarea functionalitatii de detectie si protectie in cazul caderilor de grindina de dimensiuni mari .
* Modulul de urmarire solara, ce permite sistemului sa gaseasca pozitia soarelui relativ la acesta.
* Modulul de pozitionare a panoului mobil, un angrenaj destul de complex considerand faptul ca el realizeaza miscarea pe 2 axe.
* Realizarea unei magistrale de comunicatie aproape de limitele maxime sugerate in specificatiile protocolului I2C pentru distante de comunicatie intre module .

## **6.2 Direcții de dezvoltare**

Există diverse direcții de dezvoltare care pot fi aplicate acestui proiect. Cele mai consistente ar fi:

1)Inlocuirea modulelor de control al incarcarii PWM cu unele MPPT ce permit o analiza mai detalita a graficului de energie produsa si studiul productiei de energie chiar si pe timpul noptii ( luna poate servi ca un reflector) , modulele actuale facand imposibila extragerea de enrgie din panou cand aceasta se apropie de limita inferioara, facand astfel dificila chiar si atingerea unei concluzii cu privire la eficienta adaugata de urmarirea solara .

2)renuntarea la folosirea fotorezistorilor, implementarea unui algoritm de pozitionare bazat pe data,ora,orientare, latitudine si longitudine cu ajutorul ADXL345 si a unor senzori aditionali, aceasta imbunatatire ar putea facilita montarea panourilor pe platforme tractabile pentru rulote spre exemplu, facand operatiunea de pozitionare 100% automatizata dupa parcarea vehiculului.

3) Implementarea functionalitatii de protectie impotriva grindinei intr-un modul mai integrabil in sistemele existente pe piata , un accelerometru ce se lipeste pe panou, conectat la o cutie de control ce poate suprascrie comenzile venite de la cutiile de comanda a sistemelor mobile existente pe piata.

# **FIGURI și TABELE**

Fig. 1.1-1 Grafic comparatie instalatie fixa si mobila..................................................3

Fig. 1.1-2 Grafic de comparatie intre panouri fixe si mobile ......................................4

Fig. 2.1.1-1 Pinout-ul microcontroller-ului esp32........................................................6

Fig. 2.1.2-1 ATmega2560...........................................................................................7

Fig. 2.1.3-1 BMP280...................................................................................................8

Fig. 2.1.3-2. AHT21.....................................................................................................9

Fig. 2.1.4-1 Senzor de curent bidirectional INA219....................................................10

Fig. 2.1.5-1 Punte H de putere BTS7960....................................................................11

Fig. 2.1.6-1 Coborator de tesiune XL4015..................................................................12

Fig. 2.1.7-1 Modul microSD........................................................................................13

Fig. 2.1.8-1 Modul de control incarcare solara PWM................................................13

Fig. 2.1.9-1 Panou solar..............................................................................................14

Fig. 2.1.10-1 Actuatoare liniare....................................................................................15

Fig. 2.1.11-1 Senzor ADXL345....................................................................................16

Fig. 2.1.13-1 Translator nivel logic...............................................................................16

Fig. 2.2.1-1 Meniul pentru managementul placilor de dezvoltare................................19

Fig. 2.2.1-2 Meniul pentru managementul librariilor.....................................................20

Fig. 2.2.2-1 Management-ul branch-urilor in git...........................................................21

Fig. 2.2.2-2 Procesul de dezvoltare si integrare in Git.................................................21

Fig. 2.2.3-1 Structura programului C++.......................................................................23

Fig. 2.2.3-2 Structura programului în Arduino IDE.....................................................23

Fig. 2.2.4-1 Tranzitiile la inceputul si sfarsutul comunicatiei in I2C.............................25

Fig. 2.2.4-1 Transmisie de date prin I2C.....................................................................25

Fig. 3.1-1 Diagrama de stari........................................................................................26

Fig. 3.2-1 Diagrama module........................................................................................27

Fig. 3.3-1 Conexiuni sistem la nivel logic....................................................................28

Fig. 3.3-2 Pinout Mega 2560.......................................................................................29

Fig. 5-1 Rezultate Experiment 1..................................................................................37

Fig. 5-2 Rezultate Experiment 2..................................................................................37

## 

# **BIBLIOGRAFIE**

[1] <https://iopscience.iop.org/article/10.1088/1755-1315/709/1/012003/meta> Accesat în 20 August 2023

[2] <https://core.ac.uk/download/pdf/159179631.pdf> Accesat în 20 August 2023

[3] <https://www.sigmanortec.ro/placa-dezvoltare-esp32-cu-wifi-si-bluetooth> Accesat în 20 August 2023

[4] <https://www.sigmanortec.ro/Placa-dezvoltare-Mega-2560-R3-cablu-usb-compatibil-ARDUINO-p125149866> Accesat în 20 August 2023

[5] <https://www.sigmanortec.ro/Senzor-BMP280-presiune-atmosferica-temperatura-p125704124?fast_search=fs> Accesat în 22 August 2023

[6] <https://ardushop.ro/ro/home/2510-modul-senzor-umiditate-si-temperatura-aht21-groundstudio.html> Accesat în 22 August 2023

[7] <https://www.sigmanortec.ro/Senzor-monitor-curent-tensiune-bidirectional-INA219-p136254418> Accesat în 22 August 2023

[8] <https://www.sigmanortec.ro/punte-h-de-putere-driver-motor-bts7960-43a> Accesat în 22 August 2023

[9] <https://www.sigmanortec.ro/Modul-coborator-tensiune-XL4015-5-36VDC-5A-75W-cu-display-p158469512?fast_search=fs> Accesat în 23 August 2023

[10] <https://www.sigmanortec.ro/Modul-MicroSD-p126079625?fast_search=fs> Accesat în 23 August 2023

[11] <https://www.sigmanortec.ro/modul-control-incarcare-solara-pwm-rbl-30a-versiune-hq?fast_search=fs> Accesat în 23 August 2023

[12] <https://brecknergermany.ro/panouri-solare/panou-solar-20w-460x350x40mm-monocristalin-cu-regulator-de-incarcare-10a-12-24v-cablu-1900-mm-breckner-germany.html> Accesat în 23 August 2023

[13]<https://www.amazon.com/gp/product/B0B1P88CJB/ref=ppx_yo_dt_b_asin_title_o00_s00?ie=UTF8&psc=1> Accesat în 23 August 2023

[14] <https://www.sigmanortec.ro/modul-accelerometru-adxl345-i2cspi-33-5v> Accesat în 23 August 2023

[15] <https://www.sigmanortec.ro/Modul-Translator-nivel-logic-I2C-IIC-bidirectional-8-canale-3-3V-5V-TXS0108E-p136254032?fast_search=fs> Accesat în 23 August 2023

[16] <https://docs.arduino.cc/learn/starting-guide/the-arduino-software-ide> Accesat în 1 Septembrie 2023

[17] <https://docs.github.com/en/get-started/quickstart/hello-world> Accesat în 1 Septembrie 2023

[18]<https://www.ti.com/lit/an/slva704/slva704.pdf?ts=1693966525765&ref_url=https%253A%252F%252Fwww.google.com%252F> Accesat în 1 Septembrie 2023

[19] <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/about-state-diagrams/> Accesat în 26 August 2023

[20] <https://app.diagrams.net/> Accesat în 3 Septembrie 2023

**DECLARAŢIE DE AUTENTICITATE A**

**LUCRĂRII DE FINALIZARE A STUDIILOR\***

Subsemnatul Arnautu Dumitru-Florin\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,

legitimat cu CARTEA DE IDENTITATE\_seria \_\_\_GZ\_\_\_nr. \_\_\_\_\_\_\_\_\_836580\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,

CNP \_\_\_\_\_\_\_\_\_\_\_\_\_\_5000722181091\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,

autorul lucrării \_SISTEM PENTRU STUDIUL RENTABILITATII IMPLEMENTARII METODELOR DE EFICIENTIZAREA PRODUCTIEI DE ENERGIE IN SISTEMELE FOTOVOLTAICE\_\_\_\_\_\_

elaborată în vederea susţinerii examenului de finalizare a studiilor de \_\_\_\_\_\_\_\_\_\_\_LICENTA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_organizat de către Facultatea \_\_\_DE AUTOMATICA SI CALCULATOARE din cadrul Universităţii Politehnica Timişoara, sesiunea \_\_SEPTEMBRIE\_ a anului universitar \_\_\_2022-2023\_\_, coordonator \_ Conf.dr.ing. Adrian Korodi\_, luând în considerare art. 34 din *Regulamentul privind organizarea și desfășurarea examenelor de licență/diplomă și disertație*, aprobat prin HS nr. 109/14.05.2020 și cunoscând faptul că în cazul constatării ulterioare a unor declaraţii false, voi suporta sancțiunea administrativă prevăzută de art. 146 din Legea nr. 1/2011 – legea educației naționale și anume anularea diplomei de studii, declar pe proprie răspundere, că:

* această lucrare este rezultatul propriei activități intelectuale;
* lucrarea nu conține texte, date sau elemente de grafică din alte lucrări sau din alte surse fără ca acestea să nu fie citate, inclusiv situația în care sursa o reprezintă o altă lucrare/alte lucrări ale subsemnatului;
* sursele bibliografice au fost folosite cu respectarea legislaţiei române şi a convenţiilor internaţionale privind drepturile de autor;
* această lucrare nu a mai fost prezentată în fața unei alte comisii de examen/prezentată public/publicată de licență/diplomă/disertație;
* În elaborarea lucrării nu am utilizat instrumente specifice inteligenței artificiale (IA)[[1]](#footnote-1).

Declar că sunt de acord ca lucrarea să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului său într-o bază de date în acest scop.

Timişoara,

A black line drawn on a white surface

Description automatically generated Data Semnătura

\_\_\_03/09/2023\_\_\_\_\_\_\_\_\_\_\_

1. \*Declaraţia se completează de student, se semnează olograf de acesta și se inserează în lucrarea de finalizare a studiilor, la sfârșitul lucrării, ca parte integrantă.

   Se va păstra una dintre variante: 1 - s-a utilizat IA și se menționează sursa 2 – nu s-a utilizat IA [↑](#footnote-ref-1)