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](#_Toc144725507)

[REZUMAT 3](#_Toc144725508)

[**1 Produse existente în domeniu** 4](#_Toc144725509)

[**2. Fundamentele teoretice** 6](#_Toc144725510)

[**2.1 Limbajul de programare C++ în Arduino IDE** 6](#_Toc144725511)

[**2.2 Protocolul de comunicare I2C** 8](#_Toc144725512)

[**2.2.1 Introducere** 8](#_Toc144725513)

[**3. Componentele hardware** 9](#_Toc144725514)

[**3.1 Placa de dezvoltare ESP32** 9](#_Toc144725515)

[**3.2 Cloona a placii de dezvoltare Arduino Mega2560** 9](#_Toc144725516)

[**3.3 Senzorii de temperatura BMP280 si AHT21** 11](#_Toc144725517)

[**3.5 Puntile H** 12](#_Toc144725518)

[**3.6 Coborator de tesiune XL4015** 14](#_Toc144725519)

[**3.7 Modul microSD** 14](#_Toc144725520)

[**3.8 Module de control incarcare solara PWM , RBL-30A** 15](#_Toc144725521)

[**3.9 Panouri solare 20W** 16](#_Toc144725522)

[**3.10 Actuatoare liniare** 17](#_Toc144725523)

[**3.11 Modul SENZOR 3 AXE ADXL345** 18](#_Toc144725524)

[**3.13 TRANSLATOR NIVEL LOGIC I2C IIC BIDIRECTIONAL 8 CANALE 3.3V 5V TXS0108E** 18](#_Toc144725525)

[**4. Codul de la baza proiectului** 20](#_Toc144725526)

[**4.1 Mega2560:** 20](#_Toc144725527)

[**5. Implementarea proiectului** 27](#_Toc144725528)

[**6. Testarea proiectului** 29](#_Toc144725529)

[**7. Concluzii și direcții de dezvoltare** 30](#_Toc144725530)

[**7.1 Concluzii** 30](#_Toc144725531)

[**7.2 Direcții de dezvoltare** 30](#_Toc144725532)

[**FIGURI și TABELE** 31](#_Toc144725533)

[**BIBLIOGRAFIE** 32](#_Toc144725534)

# REZUMAT

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.

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.

# **1 Produse existente în domeniu**

În momentul de față există pe piața diverse sisteme ce satisfac o parte din funcțiile implementate in acest proiect , acestea sunt totuși costisitoare și încorporează componente greu de înlocuit, ce nu permit reparații si necesita inlocuirea unor ansamble complexe în cazul unor defecțiuni.

A picture containing design

Description automatically generated

Figura 1. Tracker solar pentru panouri solare fotovoltaice [1]

Close-up of a solar panel

Description automatically generated

Figura 2. Micro calculator tracker complet poziționer dual motor [2]

Produsele de mai sus asigura funcționalitatea de urmărire a soarelui, acestea nu dispun de alte funcționalitați.

# **2. Fundamentele teoretice**

## **2.1 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]

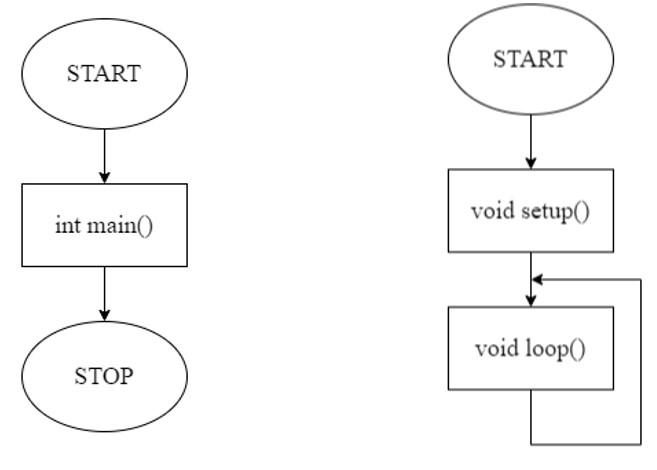


Figura 3. Structura programului C++ Figura 4. Structura programului în Arduino IDE

## **2.2 Protocolul de comunicare I2C**

### **2.2.1 Introducere**

Comunicarea prin interfața I2C 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 (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 .

# **3. Componentele hardware**

## **3.1 Placa de dezvoltare ESP32**

A circuit board with many different colored buttons

Description automatically generated with medium confidencePlaca de dezvoltate WiFi bazată pe ESP32, 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.

Figura 5. 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) [3]

## **3.2 Cloona a placii de dezvoltare Arduino Mega2560**

Placa de dezvoltare construită după schemele publice ale celebrei placi ArduinoMega , 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) .

A blue electronic board with black wires

Description automatically generated

Figura 6. 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 [4]

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

Cei 2 senzori 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.

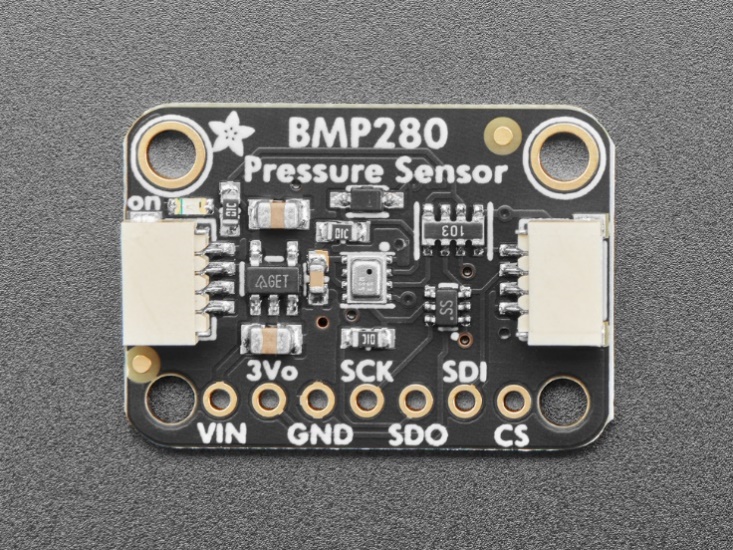
Optandu-se pentru o cutie din plastic cu capac transparent , aceasta fiind umpluta cu componente ce disipa caldură am constatat ca este absolut necesară o imbunatațire a racirii sistemului , temperatura maxima masurată cu senzorii fiind de 75°C , prag la care presupun că sistemul a suferit avarii ale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

Figura 7. 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 [5]

A black and white circuit board

Description automatically generated

Figura 8. 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 [6]

#### **3.4 Senzorii pentru masurarea curentului INA219** 4 senzori 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.

Figura 9. 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. [7]

## **3.5 Puntile H**

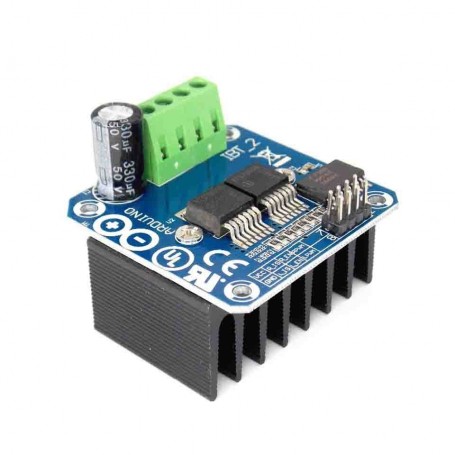
 2 punti H de mare putere, tip BTS7960 au fost folosite pentru a controla cele 2 motoare liniare ce directioneaza cadrul mobil prezent in proiect, acestea se controleaza in mod clasic folosind semnale digitale si PWM. Acestea nu au prezentat dificultati de integrare in sistem , acest fapt se atribuie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

Figura 10. 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 [8]

## **3.6 Coborator de tesiune XL4015**

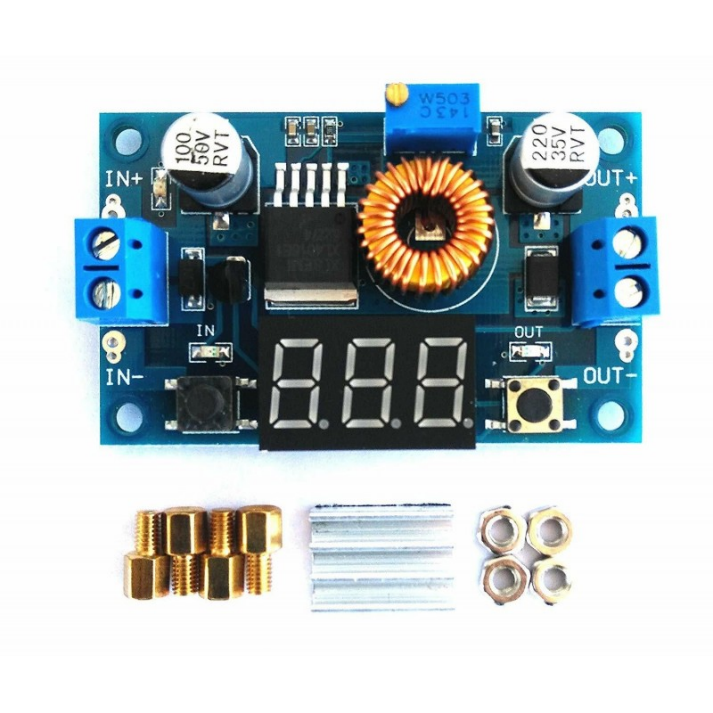
Un modul step down pentru coborarea tensiunii de la 12V( uneori chiar si 14V) provenite din circuitul realizat de panoul fotovoltaic la valori mai usor de gestionat , la bordul placilor de dezvoltare (Mega si ESP32) se regasesc regulatoare de tipul AMS1117 de 5V respectiv 3.3V , acestea disipand o cantitate 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 :

Figura 11. 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 11V din acumulator ( tensunea minima, sub aceasta tensiune acumulatorul poate fi avariat daca se continua descarcarea) in 6V.

Tensiune intrare: 5 - 36VDC

Tensiune iesire: 1.25 - 32VDC, reglabila

Curent: intre 0 si 5 Ah [9]

## **3.7 Modul microSD**

Modul folosit pentru colectarea datelor preluate de sistem :cantitatea de energie produsa , temperatura si umiditatea din cutia de control.



Figura 12. Modul microSD

Dimensiuni card microSD acceptate: 2GB ~ 32GB

Tensiune de alimentare: 3.3V sau 5V DC

Protocol de comunicare: SPI [10]

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

Module 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.

Figura 13. 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 [11]

## **3.9 Panouri solare 20W**

A solar panel with many black squares

Description automatically generated Panouri fotovoltaice monocristaline de 20W, rama de aluminiu , sticla de protectie securizata si cabluri de conectare atasate.

Figura 14. 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 [12]

## **3.10 Actuatoare liniare**

A metal frame with metal tubes and wires

Description automatically generated with medium confidence 2 actuatoare liniare sunt folosite pentru a facilita miscarea de rotatie in jurul axelor X si Y.

Figura 15. 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) [13]

## **3.11 Modul SENZOR 3 AXE ADXL345**

Modul accelerometru pe 3 axe 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

Figura 16. Senzor ADXL345

Tensiune alimentare: 3.3 - 5VDC

Integrat: ADXL345

Integrat putere: RT9161

Interfata: I2C, SPI

Dimensiuni mm: 28 x 14mm [14]

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

Modul 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

Figura 17. 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 [15]

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

## **4.1 Mega2560:**

**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. Implementarea proiectului**

A circuit board with many colored wires

Description automatically generated

Figura 18. Pinout Mega2560

O schema electrica completa va fi relizata in etapele viitoare de dezvoltare ale proiectului , acesta fiind un prototip o schema electrica finala este aproape imposibil de realizat si ar ajunge la o complexitate si un nivel de detaliu greu de cuprins intr-un document, necesitand aplicatii specializate pentru vizualizarea acesteia, in continuare voi merge pe ideea de mici scheme de conectare .

# **6. Testarea proiectului**

Folosind metode de testare manuala proiectului i-a fost testata functionalitatea atat in conditii normale, cat si in situatii mai speciale.

Testarea manuala este unul din procesele esentiale in dezvoltarea software, desi in momentul de fata tehnicile de testare automatizata au luat avant , testarea manuala ramane esentiala in descoperirea erorilor din cod, putem efectua orice tip de testare atat manuale cat si automatizate dar testarea 100% automata nu este pozibila sau nu este rentabila , timpul necesar dezvoltarii unor teste automate pentru particularitati depasind timpul necesar testarii manuale pentru testarea manuala a acelorasi particularitati ale sistemului. [99]

In cadrul proiectului am folosit mai multe teste pentru a usura integrarea modulelor individuale, fiecare componenta a proiectului a fost testata individual pentru a-i observa comportamentul si a crea scenariile potrivite pentru a rezolva problemele ce pot aparea prin nefunctionarea acesteia.

Ecranul Oled, in mod normal exemplele pentru folosirea acestuia trateaza cazul in care acesta este deconectat sau nu raspunde in cadrul initializarii prin introducerea microcontroller-ului intr-o bucla infinita in care se incearca conectarea la acesta, desigur, acest comportament duce la oprirea intregului sistem , prin simpla indisponibilitate a modulului OLED tot restul functiilor sistemului nefiind disponibile, rolul ecranului fiind pentru diagnoza rapida si de a indeplini rolul de interfata cu utilizatorul el nu indeplineste functii critice , astfel lipsa acestia este tratata prin executia in contnuare a codului.

Senzorii INA219, acestia indeplinesc un rol important in cadrul proiectului, scopul principal al proiectului fiind colectarea de date de la acesti senzori pentru stabilirea diferentei de cantitate de energie electrica generata de diverse sisteme fotovoltaice fie acestea mobile sau fixe, s-a optat pentru introducerea unui blocaj in rularea codului in cazul nefunctionarii acestora pentr a preveni colectarea de date irelevante dupa cum urmeaza:

if (! ina219\_0x40.begin()||!ina219\_0x44.begin()) {

    while (1) { delay(10); }

  }

Prin testarea senzorilor de temperatura am sesizat o deviatie de temperatura de 1 grad la nivelul senzorului BMP280 , prin compararea rezultatelor de pe acesta , a celor furnizate de AHT21 si a celor de la clima interioara.

Astfel o corectie de 1 grad a fost adaugata direct la citirea senzorului.

temperature = bmp.readTemperature()-1.0;

Tot in cadrul senzorului BMP s-au depistat aparitia sporadica a unor erori de masurare a presiunii atmosferice , datorita aparitiei sporadice si cu deviatii minore am renuntat la incercarea de corectare a acestora.Alti utilizatori au mai observat date eronate provenite de la acest senzor [100]

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

## **7.1 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 nelimitate, 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

## **7.2 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.

2)renuntarea la ADS1015 si 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.

# **FIGURI și TABELE**

Figura 1. Tracker solar pentru panouri solare fotovoltaice [1]......................................................4

Figura 2. Micro calculator tracker complet pozitioner dual motor [2]............................................4

Figura 3. Structura programului C++............................................................................................6

Figura 4. Structura programului în Arduino IDE..........................................................................6

Figura 5. Pinout-ul microcontroller-ului esp32..............................................................................8

Figura 6. ATmega2560.................................................................................................................9

Figura 7. BMP280.......................................................................................................................10

Figura 8. AHT21..........................................................................................................................11

Figura 9. Senzor de curent bidirectional INA219........................................................................12

Figura 10. Punte H de putere BTS7960......................................................................................13

Figura 11. Coborator de tesiune XL4015....................................................................................14

Figura 12. Modul microSD..........................................................................................................15

Figura 13. Modul de control incarcare solara PWM...................................................................15

Figura 14. Panou solar................................................................................................................16

Figura 15. Actuatoare liniare.......................................................................................................17

Figura 16. Senzor ADXL345.......................................................................................................18

Figura 17. Translator nivel logic..................................................................................................19

Figura 18. Pinout Mega2560.......................................................................................................27

## 

# **BIBLIOGRAFIE**

[1] <https://shop.ecosolaris.ro/tracker-orientare-solara/tracker_solar_cu_doua_axe> Accesat în 20 August 2023

[2] <https://shop.ecosolaris.ro/tracker-orientare-solara/micro-calculator-tracker-complet-pozitioner-dual-motor> 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>

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

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

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

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

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

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

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

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

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

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

[99] <https://www.guru99.com/manual-testing.html> Accesat în 1 Septembrie 2023

[100] <https://github.com/dotnet/iot/issues/1027> Accesat în 26 August 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)