**TITLUL LUCRĂRII DE LICENŢĂ**

LUCRARE DE LICENŢĂ

|  |  |  |
| --- | --- | --- |
|  | Absolvent: | **Tudor Ovidiu PAL** |
|  |  |  |
|  | Coordonator științific: | **titlul științific Prenume NUME** |

**2024**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | |  | |
| DECAN, |  | | DIRECTOR DEPARTAMENT, | |
| **Prof. dr. ing. Liviu MICLEA** |  | **Prof. dr. ing. Rodica POTOLEA** | |

Absolvent: **Prenumele NUMELE**

**TITLUL LUCRĂRII DE LICENŢĂ**

1. **Enunțul temei:** *Scurtă descriere a temei lucrării de licență și datele inițiale*
2. **Conținutul lucrării:** *(enumerarea pârților componente) Exemplu: Pagina de prezentare, aprecierile coordonatorului de lucrare, titlul capitolului 1, titlul capitolului 2,… titlul capitolului n, bibliografie, anexe.*
3. **Locul documentării**: *Exemplu*: Universitatea Tehnică din Cluj-Napoca, Departamentul Calculatoare
4. **Consultanți**:
5. **Data emiterii temei:** 1 noiembrie 2022
6. **Data predării:** 8 iulie 2023

|  |  |
| --- | --- |
| Absolvent: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  |
| Coordonator științific: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**Declarație pe propria răspundere privind**

**autenticitatea lucrării de licență**

Subsemnatul(a)**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**, legitimat(ă) cu \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ seria \_\_\_\_\_\_\_ nr. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   
CNP \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, autorul lucrării \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_elaborată în vederea susținerii examenului de finalizare a studiilor de licență la Facultatea de Automatică și Calculatoare, Specializarea \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ din cadrul Universității Tehnice din Cluj-Napoca, sesiunea \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ a anului universitar \_\_\_\_\_\_\_\_\_\_, declar pe propria răspundere că această lucrare este rezultatul propriei activități intelectuale, pe baza cercetărilor mele și pe baza informațiilor obținute din surse care au fost citate, în textul lucrării, și în bibliografie.

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

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

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

|  |  |  |
| --- | --- | --- |
| Data  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  | Nume, Prenume  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  |  |
|  |  | Semnătura |

**Instrucțiuni generale**

**De citit înainte** (paginile de aici până la cuprins se vor elimina din versiunea finală):

1. Cele trei pagini anterioare (foaie de capăt, foaie sumar, declarație) se vor lista pe foi separate (nu fată-verso), fiind incluse în lucrarea listată.
   1. Foaia de sumar (a doua) necesită semnătura absolventului, respectiv a coordonatorului.
   2. Pe declarație se trece data când se predă lucrarea la secretarii de comisie.
   3. Pe foaia de capăt, se va trece corect titulatura cadrului didactic îndrumător (consultați pagina de unde ați descărcat acest document pentru lista cadrelor didactice cu titulaturile lor).
2. **Cuprinsul** începe pe pagina nouă, impară (dacă se face listare față-verso). Pentru actualizarea cuprinsului, folosiți meniul *References: Table of Contents->Update table*. Cuprinsul este numerotat cu cifre romane, litere mici.
3. Toate capitolele încep pe o pagină nouă. Acest lucru este realizat folosind simbolul de formatare *Section Break* cu opțiunea *Next Page*, care a fost folosit la sfârșitul fiecărui capitol. Dacă ștergeți din greșeală simbolul, îl puteți pune înapoi folosind submeniul *Breaks* al meniului *Layout*.

Numerotarea paginilor începe de la 1 la primul capitol (**Introducere)**.

1. **Formatarea:**
   1. Mărimea paginii este A4, cu toate marginile de 25.4 mm
   2. Fontul folosit implicit în acest document este Times New Roman, 12pt, conform stilului *Normal*, cu spațiere la 1 rând (*Paragraph*->*Line spacing->* 1.0) și Alignment: *Justify***.**
   3. Capitolele încep pe o pagină nouă.
   4. Pentru prima linie din fiecare paragraf se folosește indentare (implicit in *Normal Style*), iar între paragrafe succesive nu se lasă distanță suplimentară.
   5. Folosiți stilurile predefinite în acest document (Headings, Figura, Tabel, Normal etc.)
   6. **Titlurile**:
      1. Pentru capitole: Heading 1, numerotat cu o cifra (Capitolul x. Numele capitolului), font Times New Roman 16pt, Bold, 30pt spațiu vertical înainte, 18pt spațiu după.
      2. Pentru secțiuni: Heading 2, font Times New Roman 14pt, Bold, spațiu vertical înainte de titlu 12pt, spațiu după 6pt.
      3. Pentru subsecțiuni: Heading 3, Times New Roman 12pt, Regular, spațiu vertical înainte de titlu 12pt, spacing după 6pt
2. Vizualizați (și în timpul editării, dacă e cazul) acest document după ce activați vizualizarea simbolurilor ascunse de formatare (click pe pictograma **** din meniul *Home->Paragraph*).
3. Respectați restul instrucțiunilor din fiecare capitol.
4. Înainte de salvarea ca fișier pdf, din meniul *Review* alegeți *No Markup*. Apoi alegeți din meniul *Options* opțiunile din figura următoare



Cuprins

[Capitolul 1. Introducere 1](#_Toc153376410)

[Capitolul 2. Obiectivele proiectului 2](#_Toc153376411)

[Capitolul 3. Studiu bibliografic 4](#_Toc153376412)

[3.1. Modele de rețele neuronale 5](#_Toc153376413)

[3.1.1. AlexNet sau ImageNet 5](#_Toc153376414)

[Capitolul 4. Analiză și fundamentare Teoretică 7](#_Toc153376415)

[4.1. Exemplu de titlu de secțiune 7](#_Toc153376416)

[4.1.1. Exemplu de titlu de subsecțiune 7](#_Toc153376417)

[Capitolul 5. Proiectare de detaliu și implementare 9](#_Toc153376418)

[5.1. Componente folosite 9](#_Toc153376419)

[5.1.1. Senzorul de presiune atmosferică BMP280 9](#_Toc153376420)

[5.1.2. Modul cu senzor de umiditate în sol 10](#_Toc153376421)

[5.1.3. Modul cu senzor de lumină 12](#_Toc153376422)

[5.1.4. Senzor de umiditate și temperatură DHT22 12](#_Toc153376423)

[5.1.5. Pompă și releu 14](#_Toc153376424)

[Capitolul 6. Testare și validare 15](#_Toc153376425)

[Capitolul 7. Manual de instalare si utilizare 17](#_Toc153376426)

[Capitolul 8. Concluzii 19](#_Toc153376427)

[Bibliografie 21](#_Toc153376428)

[Anexa 1 23](#_Toc153376429)

[Anexa 2 24](#_Toc153376430)

[Anexa 3 25](#_Toc153376431)

# Introducere

Ce se scrie aici:

* Contextul proiectului
* Conturarea domeniului exact al temei
* Grafice, utilitate, cum poti ajuta societatea, la ce e bun

Între 2 și 3 pagini. Nu puneți secțiuni

# Obiectivele proiectului

În acest capitol se prezintă tema propriu zisă (sub forma unei teme de proiectare/cercetare, formulată exact, cu obiective clare și eventuale figuri explicative).

Între 2 și 3 pagini

# Studiu bibliografic

Documentarea bibliografică are ca obiectiv prezentarea stadiului actual al domeniului/sub-domeniului în care se situează tema. În redactarea acestui capitol (în general a întregului document) se va ține cont de cunoștințele acumulate la disciplinele dedicate din semestrul 2, anul 4 (Metodologia Întocmirii Proiectelor, etc.), precum si la celelalte discipline relevante temei abordate.

Acest capitol va avea între 3 și 10 pagini.

Referințele se scriu în secțiunea *Bibliografie*. Formatul referințelor trebuie sa fie de tipul *IEEE* sau asemănător. Introducerea și formatarea referințelor în bibliografie, respectiv citarea în text, se poate face manual – nu recomandăm, întrucât sunt mai greu de gestionat corect – sau folosind instrumentele de lucru menționate în ultimele paragrafe din acest capitol.

In secțiunea *Bibliografie* sunt exemple de referințe pentru articol la conferințe [1] [2], articol în jurnal [3], sau cărți [4]. Referințele spre aplicații sau resurse online (pagini de internet) trebuie sa includă cel puțin o denumire sugestivă pe lângă URL ca în [5], plus alte informații dacă sunt disponibile (autori, an, etc.) și data ultimului acces. Referințele care prezintă doar o legătură spre resursa online se vor plasa în footer-ul paginii unde sunt referite.

Citarea referințelor în text este obligatorie, vezi exemplul de mai jos (în funcție de tema proiectului se poate varia modul de prezentare a metodei/aplicației).

În articolul [1] autorii prezintă un sistem pentru detecția obstacolelor în mișcare folosind stereo viziune și estimarea mișcării proprii (nu e adevărat despre articolul [1] ). Metoda se bazează pe …*trecerea în revistă a algoritmilor, structurilor de date, funcționalitate, aspecte specifice temei proiectului* *etc*….. Discuție: *avantaje – dezavantaje*.

În capitolul 4 al [4] *se prezintă* …..

Începând cu **MS Word 2007** se poate folosi sistemul integrat de gestiune bibliografiei: *References* submeniul *Citations & Bibliography.* Mai multe informații se găsesc în documentația online de la MS Office.

## Modele utilizate pentru a clasifica imagini

### AlexNet

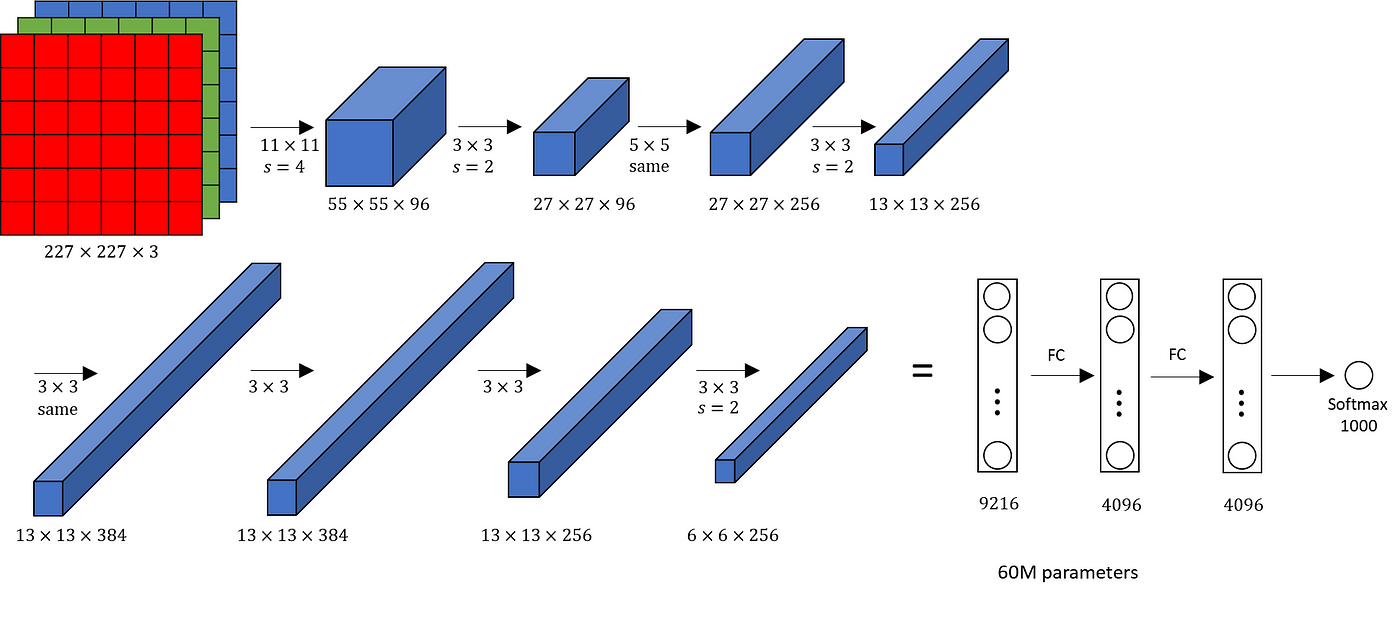
AlexNet este o arhitectură de rețea neuronala convoluționala (CNN) care a câștigat o competiție de recunoaștere a imaginilor in 2012. Modelul a fost propus intr-un articol de cercetare intitulat "ImageNet Classification with Deep Convolutional Neural Networks" de Alex Krizhevsky și colaboratorii săi.

<https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>

Acest model a pus bazele clasificării imaginilor folosind inteligență artificială prin două mari inovații:

* Utilizarea activării ReLU (Rectified Linear Unit) a contribuit la accelerarea antrenării modelului cu aproximativ șase ori, comparativ cu activarea tradițională Sigmoid sau Tanh
* Introducerea tehnicii de dropout, adaugând straturi specifice care au redus riscul de “overfitting”. Ideea principală a dropout-ului este de a elimina aleatoriu (a "abandona") o parte din neuroni în timpul antrenării.

Arhitectura AlexNet constă în opt straturi cu parametri ce pot fi învățați. Modelul are cinci straturi de convoluție urmate de trei straturi complet conectate, iar în toate aceste straturi, cu excepția stratului de ieșire, se utilizează activarea ReLU.



<https://medium.com/@siddheshb008/alexnet-architecture-explained-b6240c528bd5>

Fiind un model care utilizează o arhitectura destul de învechită, acest model vine cu câteva dezavantaje:

* Număr mare de parametrii (62.3 milioane), ceea ce duce la o complexitate computațională crescută in timpul antrenării și inferenței.
* Posibilitate de Overfitting. Chiar și cu introducerea tehnicii de dropout, AlexNet este predispus la overfitting, în special în situații în care setul de date de antrenare este relativ mic sau neechilibrat.
* Limitat la setul de date ImageNet. Modelul a fost antrenat pe setul de date ImageNet, care cuprinde 1000 de clase distincte. Transferul la alte domenii poate necesita ajustări

### K-Nearest-Neighbours (KNN)

K-Nearest Neighbors (KNN) este o tehnică de învățare supervizată, contrar cu modele de tip rețele neuronale, utilizată în domeniul prelucrării imaginilor pentru clasificare și găsirea de imagini similare. Această abordare se bazează pe ideea că imagini similare au caracteristici (atribute) similare în spațiul caracteristic.

La baza acestui algoritm stau următoarele principii:

* Fiecare imagine este reprezentată ca un vector de caracteristici (i.e. culoare, textură, margini). Aceste caracteristici pot fii extrase și prin deep-learning.
* Vectorii de caracteristici sunt adesea normalizați pentru a asigura o contribuție echitabilă a fiecărui atribut.
* Un aspect crucial in algoritmul KNN este alegerea valorii k, care reprezintă numărul de vecini luați în considerare în procesul de clasificare.

<https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/>

Decizia clasei din care o imagine face parte se ia în funcție de cei mai apropiați K vecini din setul de antrenare. Figura următoare ilustrează acest procedeu unde instanța de test este pătratul galben din centru înconjurat de instanțe etichetate din setul de antrenare. Interiorul cercului cuprinde 5 vecini pe baza cărora se va face clasificarea.

O imagine care conține cerc, Color, Grafică, clipart

Descriere generată automat

Figura 3.1. Exemplu de clasificator KNN

Pentru că algoritmul salvează toate caracteristicile tuturor imaginilor din setul de date, apar anumite limitări:

* Performanța k-NN poate scădea pe seturi de date mari din cauza complexității computaționale ridicate.
* Alegerea corectă a caracteristicilor și normalizarea adecvată pot influența semnificativ performanța modelului.
* Pentru imagini de dimensiuni mari, are un timp computațional cu mult mai mare decât un alt tip de model. <https://www.mdpi.com/2227-7080/11/4/99> acest articol demonstrează că pe o imagine de 64 pe 64 pixeli, timpul de prezicere este de 188 de secunde.

# Analiză și fundamentare Teoretică

Împreună cu următoarele 3 capitole trebuie să reprezinte aproximativ 70% din lucrare.

Scopul acestui capitol este să explice principiile funcționale ale aplicației implementate. Aici descrieți soluția propusă din punct de vedere teoretic - explicați și demonstrați proprietățile și valoarea teoretică:

* algoritmi utilizați și/sau propuși,
* protocoale utilizate,
* modele abstracte,
* explicații/argumentări logice ale soluției alese,
* structura logică și funcțională a aplicației.

NU SE FAC referiri la implementarea propriu-zisă.

NU SE PUN descrieri de tehnologii sau lucruri care nu țin strict de proiectul propriu-zis (materiale de umplutură).

## Exemplu de titlu de secțiune

### Exemplu de titlu de subsecțiune

Fiecare tabel introdus în lucrare este numerotat astfel: Tabelul *x.y*, unde *x* reprezintă numărul capitolului, iar *y* numărul tabelului din capitol. Se lasă un rând liber între tabel și paragraful anterior, respectiv următor.

Pentru a stabili titlul unui tabel, folosiți meniul *References,* submeniul *Insert caption* și din pop-up alegeți *Tabelul.*

Pentru a referi un tabel folosiți meniul *References,* submeniul *Captions, Cross-reference* și alegeti din pop-up *Tabelul*→*only label and number*.

Exemplu: în acest rând am inserat o referență la Tabelul 4.1.

Tabelul 4.1. Numele tabelului.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Times New Roman ( 12) | xxxx | xxxx | xxxx |  |
|  |  |  |  |  |

Fiecare figură introdusă în text trebuie numerotată și referită împreună cu o scurtă descriere a conținutului său. Numerotarea se face astfel Figura x.y unde x reprezintă numărul capitolului, iar y numărul figurii în acel capitol. De exemplu: în Figura *x.y* este prezentată o imagine sintetică a ... etc.

Folosiți meniul *References, Insert caption* și din pop-up alegeți *Figura*.

Pentru a referi o figură folosiți *References->Captions->Cross-reference* și alegeți din pop-up *Figura->only label and number*.

Exemplu: în acest rând am inserat o referință la Figura 4.1.



Figura 4.1. Numele figurii

# Proiectare de detaliu și implementare

Împreună cu capitolul precedent și cel următor reprezintă aproximativ 70% din total.

Scopul acestui capitol este să documenteze aplicația dezvoltată în așa fel încât dezvoltarea și întreținerea ulterioară să fie posibilă. Cititorul trebuie să poată identifica funcțiile principale ale aplicației din ceea ce este scris aici.

Capitolul ar trebui sa conțină (nu se rezumă neapărat la):

* schema generală aplicației,
* descriere a fiecărei componente implementate, la nivel de modul,
* diagrame de clase, clase importante și metode ale claselor importante.

## Componente folosite

### Senzorul de presiune atmosferică BMP280

BMP280 este un senzor larg utilizat care măsoară temperatura, presiunea barometrică și altitudinea.

Senzorul BMP280 este relativ simplu de utilizat. Este pre calibrat și nu necesită componente suplimentare. Acest senzor este folosit pentru a măsura temperatura, presiunea atmosferică și altitudinea.

Tabelul 5.1. prezinta precizia si intervalul de funcționare al temperaturii, presiunii și altitudinii pentru senzor.

Tabelul 5.1. Precizia și intervalul de funcționare al senzorului BMP280

|  |  |  |
| --- | --- | --- |
| **Senzor** | **Precizie** | **Interval de funcționare** |
| Temperatură | +/- 1,0 °C | -40 până la 85°C |
| Presiune | +/- 1 hPa | 300 până la 1100 hPa |
| Altitudine | +/- 1 m | 0 – 9144 m |

<https://www.bosch-sensortec.com/products/environmental-sensors/pressure-sensors/bmp280/>

Specificații ale senzorului BMP280:

* Cerințe de alimentare: 3.3V sau 5V (Modulul vine cu un regulator LM6206 3.3V încorporat)
* Ușor de interfațat cu orice microcontroller.
* BMP280 consumă mai puțin de 1mA în timpul măsurătorilor.
* Modulul are protocol de comunicare atât I2C, cât și SPI. Astfel, poate fi ușor interfațat cu orice microcontroller la alegere.
* Putem schimba adresa I2C implicită a modulului BMP280 cu ajutorul unui jumper de lipire lângă cip.

Tabelul 5.2. reprezintă pinout-ul senzorului BMP280.

Tabelul 5.2. Pinout-ul senzorului BMP280

|  |  |
| --- | --- |
| **Pin** | **Descriere** |
| VCC | Alimentare |
| GND | Împământare |
| SCL | Pinul SCL pentru comunicația I2C, pinul SCK pentru comunicarea SPI |
| SDA | Pinul SDA pentru comunicația I2C, pinul SDI (MOSI) pentru comunicarea SPI |
| SDO | Pinul SDO (MISO) pentru comunicarea SPI |
| CSB | Pinul de selectare a cipului pentru comunicarea SPI |

Pentru a conecta senzorul BMP280 la Arduino, conexiunile sunt destul de simple. Vom conecta pinul VIN la intrarea de 5V la Arduino si pinul GND la GND. Vom folosi pinii rămași SCL și SDA pentru comincarea I2C. Conectăm pinul SCL la A5 și pinul SDA la A4. Figura 5.1. și Tabelul 5.1.3. prezintă conexiunile necesare.

O imagine care conține text, captură de ecran, circuit, proiectare

Descriere generată automat

Figura 5.1. Interfața senzorului BMP280 cu Arduino Uno

Tabelul 5.3. Interfața senzorului BMP280 cu Arduino Uno

|  |  |
| --- | --- |
| **BMP280** | **Arduino Uno** |
| VIN | 5V |
| GND | GND |
| SCL | A5 |
| SDA | A4 |

### Modul cu senzor de umiditate în sol

Senzorul de umiditate în sol operează într-un mod simplu și eficient. Sonda cu două conductoare expuse, în formă de furcă, acționează ca un rezistor variabil (similar cu un potențiometru) a cărui rezistență variază în funcție de conținutul de umiditate al solului.

Această rezistență variază invers proporțional cu umiditatea solului:

* Cu cât este mai multă apă în sol, cu atât este mai bună conductivitatea și cu atât mai mică este rezistența.
* Cu cât este mai puțină apă în sol, cu atât este mai scăzută conductivitatea și, prin urmare, cu atât mai mare este rezistența.

Senzorul produce o tensiune de ieșire în funcție de rezistență, iar prin măsurare putem determina nivelul de umiditate al solului.

Un senzor tipic de umiditate în sol constă în două părți sonda și modulul. Sondă este în formă de furcă cu două conductoare expuse care este introdusă în sol, iar modulul genereaza o tensiune de ieșire pe baza rezistenței sondei. Această tensiune este disponibilă atât analog pe pinul A0, cât și digital pe pinul D0.

Tabelul 5.4. Pinout-ul senzorului de umiditate in sol

|  |  |
| --- | --- |
| **Pin** | **Descriere** |
| VCC | Alimentare |
| GND | Împământare |
| A0 | Pinul analog generează tensiune de ieșire analogă proporțională cu nivelul de umiditate al solului. Un nivel mai ridicat rezultă într-o tensiune mai mare, iar un nivel mai scăzut rezultă într-o tensiune mai mică. |
| D0 | Pinul digital indică dacă nivelul de umiditate al solului se află în limite. D0 devine LOW atunci când nivelul de umiditate depășește valoarea limită (setată de potențiometru) și HIGH în caz contrar. |

Pentru a conecta senzorul de umiditate in sol la Arduino, efectuăm următoarele conexiuni. Vom conecta pinul VIN la intrarea de 5V la Arduino si pinul GND la GND. Valoarea ne va fi trimisă analog conectând A0 la A0 pe Arduino

Figura 5.2. Interfața senzorului de umiditate in sol cu Arduino Uno

O imagine care conține Inginerie electronică, text, mașină, Componenta circuitului

Descriere generată automat

Tabelul 5.5. Interfața senzorului de umiditate în sol cu Arduino Uno

|  |  |
| --- | --- |
| **BMP280** | **Arduino Uno** |
| VIN | 5V |
| GND | GND |
| A0 | A0 |

### Modul cu senzor de lumină

Pentru acest proiect vom folosi un rezistor dependent de lumină. Un component cheie al acestor senzori este sulfura de cadmiu (CdS) care este sensibil la lumină. Datorită acestei proprietăți, acest modul iși poate schimba rezistența.

* Când lumina lovește CdS, rezistența sa devine mică astfel încât electricitatea trece ușor prin circuit.
* Când nu există lumină, rezistența devine mare, iar fluxul de electricitate devine restricționat.

Această cantitate de electricitate care trece prin modul poate fi apoi citită de placa Arduino, măsurând schimbările in fluxul de electricitate, de fapt se măsoară intensitatea luminii.

Structura acestui senzor este una simplă, doi electrozi conectați la o sursă de electricitate în timp ce senzorul este așezat între acestea. Modulul este capabil de a transmite un semnal digital bazat pe potențiometru, asta înseamnă ca senzorul va trimite către placa de dezvoltare 1 sau 0.

<https://arduinointro.com/articles/projects/light-sensor-how-to-make-your-robots-react-to-light>

Tabelul 5.6. Pinout-ul senzorului de umiditate in sol

|  |  |
| --- | --- |
| **Pin** | **Descriere** |
| VCC | Alimentare |
| GND | Împământare |
| D0 | Pinul digital indică dacă intensitatea luminii se află în limite. D0 devine LOW atunci când nivelul intensității depășește valoarea limită (setată de potențiometru) și HIGH în caz contrar. |

Pentru a conecta senzorul de lumină sol la Arduino, efectuăm următoarele conexiuni. Vom conecta pinul VIN la intrarea de 5V la Arduino si pinul GND la GND. Valoarea ne va fi trimisă digital conectând D0 la D5 pe Arduino

Tabelul 5.7. Interfața senzorului de lumină cu Arduino Uno

|  |  |
| --- | --- |
| **BMP280** | **Arduino Uno** |
| VIN | 5V |
| GND | GND |
| D0 | D5 |

### Senzor de umiditate și temperatură DHT22

Senzorul DHT22 este un senzor de temperatură și umiditate destul de low cost. Folosește un senzor de umiditate capacitiv și un termistor pentru a măsura temperatura din aer. Datele sunt transmise prin intermediul unui pin digital.

Alternativa acestui senzor este DHT11 care, pentru a avea un cost mai scazut, renunță la câteva specificații tehnice.

Tabelul 5.8. Precizia și intervalul de funcționare al senzorului BMP280

|  |  |  |
| --- | --- | --- |
| **Senzor** | **Precizie** | **Interval de funcționare** |
| Temperatură | +/- 0,5 °C | -40 până la 80°C |
| Umiditate | 2-5% | 0% până la 100% |

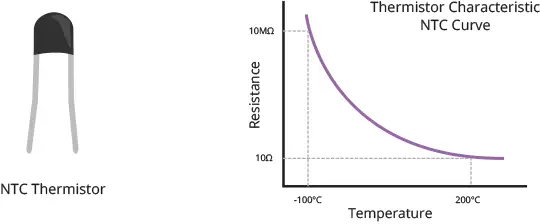
Acest senzor are două componente: senzorul de umiditate și termistorul NTC.

Senzorul de umiditate din aer are doi electrozi cu un substrat intre ei care absoarbe apa. Pe măsură ce umiditatea crește, substratul absoarbe vaporii de apă, rezultând eliberarea de ioni și o scădere a rezistenței între cei doi electrozi. Această schimbare în rezistență este proporțională cu umiditatea și poate fi măsurată pentru a estima umiditatea relativă.

Senzorul include, de asemenea, un termistor NTC pentru măsurarea temperaturii. Un termistor este un tip de rezistor a cărui rezistență variază în funcție de temperatură.

Termistorii sunt proiectați astfel încât rezistența lor să se schimbe dramatic în funcție de temperatură (cu 100 de ohmi sau mai mult pe grad). Termenul "NTC" înseamnă "Negative Temperature Coefficient," ceea ce înseamnă că rezistența scade pe măsură ce temperatura crește. (figura 5.3)

Figură 5.3 Curba caracteristică



Tabelul 5.9. Pinout-ul senzorului DHT22

|  |  |
| --- | --- |
| **Pin** | **Descriere** |
| VCC | Alimentare |
| GND | Împământare |
| NC | Not connected |
| Data | Pinul Data este folosit pentru a comunica valorile către plăcuță |

Pentru a conecta senzorul de lumină sol la Arduino, efectuăm următoarele conexiuni. Vom conecta pinul VIN la intrarea de 5V la Arduino si pinul GND la GND. Valoarea ne va fi trimisă digital conectând D0 la D5 pe Arduino

Tabelul 5.10. Interfața senzorului DHT22

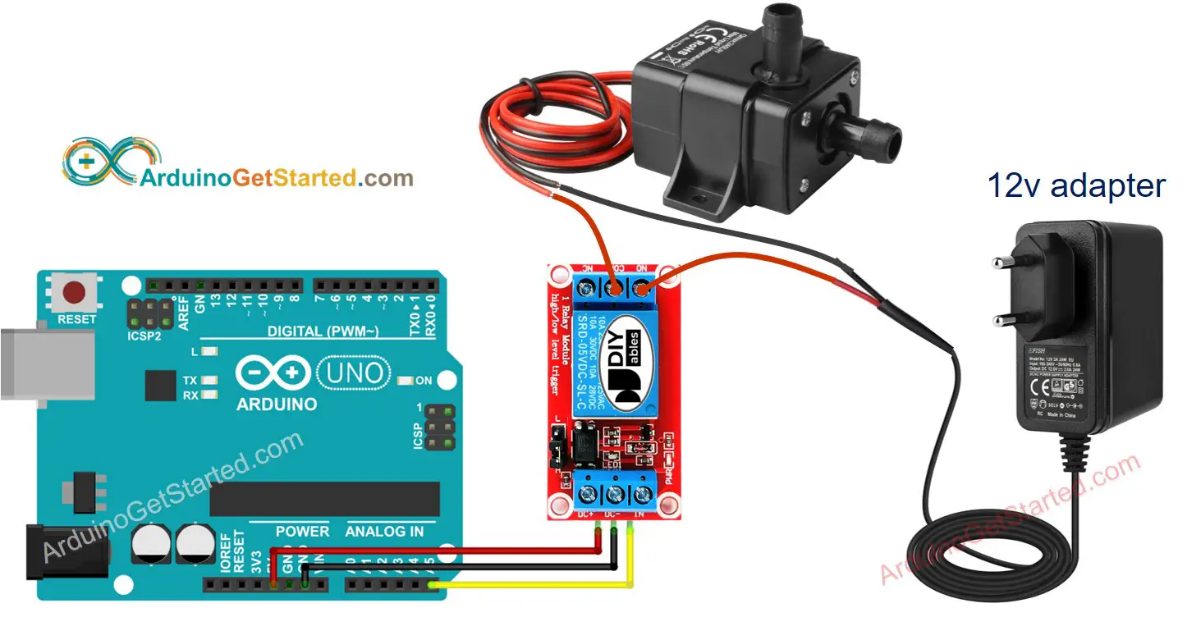
|  |  |
| --- | --- |
| **BMP280** | **Arduino Uno** |
| VIN | 5V |
| GND | GND |
| NC |  |
| Data | D2 |

Din cauza faptului că datele sunt transmise pe un pin digital, folosind un singur fir, către Arduino, avem nevoie de o librărie (DHTlib library) pentru a citi temperatura și umiditatea.

### Pompă și releu

Modulul cu 1 releu este util în cazul proiectului nostru pentru controlul unui aparat ce funcționează la tensiune înaltă.

Pentru a uda plantele când este necesar, vom folosi o pompa controlată cu ajutorul unui releu. Pompa are ground-ul întrerupt cu un alimentator de 12V pentru a avea destula putere pentru a funcționa. Alimentarea pompei este conectată la releu la unul dintre cei 2 pini (logică negativă sau logica pozitivă). Printr-un pin digital, vom transmite către releu un semnal pentru a porni pompa.



Tabelul 5.11. Interfața releului folosit pentru pompă

|  |  |
| --- | --- |
| **BMP280** | **Arduino Uno** |
| VIN | 5V |
| GND | GND |
| IN | D4 |

# Testare și validare

Acest capitol, împreună cu cele două care îl preced, va reprezenta aproximativ 70% din lucrare.

# Manual de instalare si utilizare

În secțiunea de Instalare trebuie să detaliați resursele software și hardware necesare pentru instalarea și rularea aplicației, precum și să descrieți pas cu pas procesul de instalare. Instalarea aplicației trebuie să se poată face folosind instrucțiunile date aici.

Utilizarea aplicației trebuie descrisă din punctul de vedere al utilizatorului, fără a menționa aspecte tehnice interne. Folosiți capturi ale ecranului și explicați pas cu pas interacțiunea cu persoana care execută instalarea. Folosind acest manual, o persoană ar trebui să poată instala și utiliza produsul vostru.

Minimum 1 pagină, până la 5 pagini

# Concluzii

Acest capitol va ocupa 1-2 pagini.

Capitolul ar trebui sa conțină (nu se rezumă neapărat la):

* un rezumat al contribuțiilor voastre
* o analiză critică a rezultatelor obținute
* o descriere a posibilelor dezvoltări și îmbunătățiri ulterioare

# Bibliografie

|  |  |
| --- | --- |
| [1] | G. Boella și L. van der Torre, „Contracts as Legal Institutions in Organizations of Autonomous Agents,” în *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS'04)*, New, 2004. |
| [2] | G. Boella, J. Hulstijn și L. van der Torre, „A Synthesis Between Mental Attitudes and Social Commitments in Agent Communication Languages,” în *Intelligent Agent Technology 05 (IAT 2005)*, Compiegne, 2005. |
| [3] | G. Cachon și M. Lariviere, „Supply chain coordination with revenue sharing contracts: strengths and limitations,” *Management Science,* vol. 51, pp. 30-44, 2005. |
| [4] | C. P. Pfleeger, S. L. Pfleeger și J. Margulies, Security in Computing, 5th Edition, Pearson, 2015. |
| [5] | Software Freedom Conservancy, „The Selenium Browser Automation Project,” [Interactiv]. Available: https://www.selenium.dev/. [Accesat 17 martie 2021]. |

După ce ați inserat/actualizat bibliografia selectați întregul tabel și aplicați stilul *Biblio.*  Stilul *Normal* are indentare la începutul paragrafelor și, de aceea nu veți obține formatul ca mai sus fără acest pas.

# Anexa 1

…

Secțiuni de cod relevante

# Anexa 2

Alte informații relevante (demonstrații etc.)

…

# Anexa 3

Lucrări publicate (dacă există)

etc.