

**Cerere nr. 2024-2-HU01-KA210-VET-000271388,
Aplicarea tehnologiilor Industry 4.0 în producția
și operarea caselor inteligente**



Conținut

1.	Introducerea proiectului.....	4
1.1.	Introducere – Proiectul și mediul educațional.....	4
1.2.	Scopul și rolul manualului.....	4
1.3.	Scurtă introducere și obiectivele proiectului	5
1.4.	Ideea de bază a proiectului „casă inteligentă / fermă inteligentă”.....	6
1.5.	Arhitectură educațională integrată Industrie 4.0 și mediu de dezvoltare internațională.....	6
2.	Structura proiectului și logica de implementare	7
2.1.	Fazele proiectului și interdependența lor.....	7
2.2.	Participanți și roluri.....	8
2.3.	Rezultate ale învățării și competențe care trebuie dezvoltate	9
3.	Prezentarea detaliată și prelucrarea Kitului Smart Farm.....	9
3.1.	Rolul Kitului Smart Farm în procesul de învățare.....	9
3.2.	Componentele principale ale kitului Smart Farm.....	9
3.2.1.	Controler și module de bază.....	10
3.2.2.	Elemente de intrare – Senzori.....	11
3.2.3.	Elemente de ieșire – Actuatore și afișaje.....	14
3.2.4.	Accesorii și unelte de montare.....	17
3.2.5.	Rolul componentelor în educație	17
3.3.	Rolul instrucțiunilor de asamblare din fabrică	17
3.3.1.	Asamblare mecanică și electronică și proiectare modulară	18
3.3.2.	Considerații ingineresti bazate pe experiența de asamblare.....	19
3.4.	Programarea kitului Smart Farm	20
3.4.1.	Mediu de programare bazat pe blocuri.....	20
3.4.2.	Conectarea la medii de programare bazate pe text.....	21
3.4.3.	Legătura cu etapele ulterioare ale proiectului.....	22
3.4.3.1.	Proiectul 1 – Controlul iluminatului (Sistem de iluminat).....	22
3.4.3.2.	Proiectul 2 – Sistem de control al luminii	26
3.4.3.3.	Proiectul 3 – Detecție a distanței și intervenție automată (Sistem inteligent de hrănire).....	29
3.4.3.4.	Proiectul 4 – Sistem de control al temperaturii	33
3.4.3.5.	Proiectul 5 – Sistem automat de irigații.....	38
3.4.3.6.	Proiectul 6 – Sistem Smart Farm controlat prin WiFi.....	42
3.4.3.7.	Interpretare pedagogică sumară.....	45
3.5.	Implementarea practică a capitolului.....	47
4.	Proiectați-vă digital propriul model de casă inteligentă / fermă inteligentă (Fusion 360).....	47
4.1.	Punct de plecare: analiza designului fizic al Kitului Fermă Inteligentă.....	48
4.2.	Limitări ale imprimării 3D și implicații de proiectare.....	49
4.3.	Conceptul fizic ca descriere a sistemului.....	50
4.4.	Mediul de proiectare Fusion 360 – noțiuni de bază și abordare.....	50
4.4.1.	Canvas – învățarea mișcării spațiale	52
4.4.2.	Concepte geometrice și de design de bază – într-un mod tangibil.....	54

4.5.	Pregătirea pentru imprimarea 3D – obiectiv practic final	56
4.6.	Rezultatele procesului de proiectare digitală – prezentarea și interpretarea modelelor studenților	57
4.7.	Rezumatul rezultatelor învățării bazat pe modele completate.....	60
4.8.	Implementarea practică a capitolului.....	60
5.	Experiențe în imprimarea 3D și fabricație.....	61
5.1.	Prezentarea imprimantei 3D și a Bambu Studio.....	61
5.1.1.	Scurtă descriere a procedurii aplicate.....	61
5.1.2.	Structura și funcționarea hardware-ului.....	61
5.1.3.	Sistemul de alimentare cu filament AMS.....	62
5.1.4.	Principii de feliere	63
5.1.5.	Materiale (PLA, PETG, ABS)	64
5.2.	Etapele procesului de imprimare.....	64
5.2.1.	Pregătirea și verificarea modelului.....	64
5.2.2.	Imprimare și postproducție	65
5.2.3.	Controlul calității și corectarea erorilor	66
5.3.	Probleme legate de dimensionare și potrivire.....	67
5.3.1.	Gestionarea inexactităților de fabricație	67
5.3.2.	Iterații de proiectare.....	67
5.3.3.	Semnificația educațională a proceselor, a experiențelor studenților și a lecțiilor învățate.....	67
6.	Sistem integrat de control – fundamente conceptuale.....	69
6.1.	Scopul proiectării sistemului.....	69
6.2.	Avantajele tehnice ale arhitecturii distribuite	71
6.2.1.	Model și abordare de comunicare	71
6.2.2.	Nod de control pe teren – unitate de eșantionare independentă	72
6.3.	Structura programului împărțită în unități funcționale	73
6.3.1.	Nod de control pe teren cu extindere de rețea – WiFi + Modbus TCP.....	76
6.3.2.	Nivel de supraveghere – HMI bazat pe Python / Ciclu de supraveghere	82
6.4.	Pregătirea și punerea în funcțiune a sistemului de monitorizare bazat pe Raspberry Pi.....	83
6.5.	Rolul clientului Python	86
6.6.	Înregistrarea datelor senzorilor în baza de date SQL.....	89
6.7.	Nivelul de gestionare web – rolul său în sistem.....	91
6.8.	Dezvoltarea de către studenți a interfeței web HMI în cadrul muncii în echipă	95
6.9.	Rezumat – integrarea sistemului și rezultatele învățării	97
7.	Rezumat	99

1. Prezentarea proiectului

1.1. Introducere – Proiectul și mediul educațional

Acest manual este documentația profesională și metodologică a unui proiect educațional internațional de colaborare bazat pe tehnologiile Industry 4.0 și IoT. Proiectul se concentrează pe un model de casă inteligentă/fermă inteligentă, a cărei dezvoltare va implica studenții de la proiectarea digitală și fabricația aditivă la integrarea electronicii și dezvoltarea de software, până la crearea unui sistem complex de management al clădirilor, bazat pe web.

Proiectul a început cu un kit de instruire disponibil comercial (Smart Farm Kit), care a oferit o bază solidă pentru învățarea despre senzori, actuatori și control bazat pe IoT. Cu toate acestea, procesul de dezvoltare nu s-a oprit la aplicarea soluțiilor din fabrică: participanții au efectuat analize critice, au identificat limitele sistemului și apoi au răspuns la acestea dezvoltând propria arhitectură de sistem scalabilă, urmând un model industrial.

Scopul manualului nu este doar de a documenta implementarea unui proiect specific, ci și de a prezenta un model educațional deschis, adaptabil, care poate fi urmat și dezvoltat în continuare de alte instituții și grupuri de studenți, chiar și fără utilizarea elementelor fabricate în fabrică.

1.2. Scopul și rolul manualului

Scopul acestui manual este de a oferi îndrumări profesionale și metodologice cuprinzătoare pentru un proiect educațional care sprijină dezvoltarea competențelor digitale, tehnice și ingineresti ale studenților prin aplicarea practică a tehnologiilor Industry 4.0 și IoT. Documentul nu numai că oferă soluții predefinite, ci prezintă și un proces de învățare și dezvoltare în care rezolvarea problemelor, analiza și iterația joacă un rol important.

Abordarea manualului este că educația nu se bazează pe transferul pasiv de cunoștințe, ci pe activitatea creativă activă. Pe parcursul proiectului, studenții nu sunt doar utilizatori, ci proiectanți, implementatori și dezvoltatori ai unui sistem tehnic complex. Documentul susține această abordare pe tot parcursul procesului, de la proiectare, producție și programare până la integrarea sistemului.

Funcția principală a manualului este de a fi utilizat ca material didactic. Conținutul său este potrivit pentru:

- servesc drept bază pentru proiectele din sala de clasă și extracurriculare,
- se integrează în programe de formare profesională, de management al talentelor sau bazate pe proiecte,
- oferirea de suport metodologic profesorilor în implementarea proiectelor tehnice și informatice complexe.

Abordarea metodologică se concentrează pe învățarea bazată pe proiecte și pe învățarea prin descoperire. Elevii progresează treptat de la sarcini mai simple la înțelegerea sistemelor complexe, în etape secvențiale. Prin urmare, manualul nu oferă o singură explicație „corectă”.

Nu numai că înregistrează o soluție, ci prezintă și alternative, puncte de decizie și oportunități de dezvoltare.

Documentul este relevant pentru mai multe grupuri țintă:

- Pentru studenți: manualul prezintă procesele de dezvoltare cu o logică prietenoasă pentru studenți, oferind oportunitatea experimentării independente, a greșelilor și a reproiectării. În timpul proiectului, studenții experimentează elementele de bază ale gândirii ingineresti: definesc o problemă, proiectează o soluție, o testează, o evaluează și o rafinează.
- Pentru educatori: manualul oferă sprijin în organizarea lecțiilor, atribuirea diferențiată a sarcinilor și gestionarea grupurilor eterogene. Oferă asistență în modul de implicare a elevilor cu medii diferite într-un proiect comun și în modul de procesare a conținutului tehnic într-un mod structurat pedagogic, dar flexibil.
- Pentru instituții: proiectul prezintă un model adaptabil care poate fi integrat în programele școlare locale, săptămânile de proiect sau colaborările internaționale. Natura deschisă permite instituțiilor să adapteze implementarea la propriile echipamente, infrastructură și profil de formare.

1.3. Scurtă introducere și obiectivele proiectului

Conectarea la învățământul tehnic și secundar IT: proiectul a demarat inițial într-un mediu tehnic, în care studenții de la IT industrială, mecatronică și învățământul tehnic conex aveau deja cunoștințe tehnologice de bază. Cu toate acestea, în cadrul cooperării internaționale, s-a alăturat programului și un liceu teoretic, ai cărui studenți nu aveau experiență tehnică anterioară. Această situație a reprezentat atât o provocare, cât și o oportunitate pedagogică.

Avantaje:

- Colaborarea studenților cu medii diferite a întărit gândirea explicativă și sistematică.
- Elevii școlii tehnice au preluat rolul de mentori, ceea ce le-a aprofundat propriile cunoștințe.
- Pentru studenții liceului teoretic, proiectul a format o punte între cunoștințele IT abstracte și sistemele fizice.

Posibile dificultăți:

- Lipsa cunoștințelor tehnice de bază ar fi putut încetini progresul inițial.
- A trebuit să se pună un accent sporit pe introducerea pas cu pas a conceptelor de bază (electronică, senzori, control).
- Rolul instructorului a devenit mai facilitator și de sprijin.

Pe baza experiențelor dobândite în cadrul proiectului, se poate afirma că modelul prezentat nu este funcțional doar într-un mediu școlar tehnic. Cu o structură metodologică și o diferențiere adecvate, acesta poate fi aplicat cu succes și în instituții de învățământ secundar non-tehnic.

1.4. Ideea de bază a proiectului „casă inteligentă / fermă inteligentă”

Ideea de bază a proiectului este de a crea un model fizic și digital care să ilustreze în mod tangibil funcționarea sistemelor inteligente, prezentând în același timp aplicarea tehnologiilor moderne prin probleme ușor de înțeles și din lumea reală pentru elevi. Modelul casei inteligente / fermei inteligente creează o oportunitate pentru ca aspectele tehnice, IT și de mediu să apară ca un sistem unificat în educație.

Modelul este potrivit pentru prezentarea simultană a:

- rolul senzorilor și actuatorilor în monitorizarea și controlul condițiilor de mediu,
- procesul de colectare a datelor, procesare a datelor și feedback,
- logica controlului și supravegherii în medii locale și la distanță,
- opțiuni de acces de la distanță bazate pe web ca instrument pentru funcționare eficientă din punct de vedere energetic și al resurselor.

În cadrul proiectului, casa inteligentă și ferma inteligentă apar nu doar ca o demonstrație tehnologică, ci și ca purtătoare a unei abordări ecologice. Prin măsurare bazată pe senzori și intervenție automatizată, studenții învață cum controlul inteligent poate contribui la optimizarea consumului de energie, la utilizarea conștientă a resurselor de mediu și la funcționarea durabilă (de exemplu, iluminat, ventilație, controlul irigațiilor).

Modelul nu este un produs final, ci un instrument de învățare conștient deschis, în continuă evoluție, care susține dezvoltarea gândirii sistemice prin cicluri de proiectare, experimentare, evaluare și reproiectare. Această abordare îi ajută pe studenți nu doar să stăpânească soluțiile tehnice, ci și să înțeleagă contextele lor de mediu, economice și sociale.

1.5. Arhitectură educațională integrată Industry 4.0 și mediu de dezvoltare internațională

Proiectul respectă principiile abordării Industriei 4.0:

- arhitectura sistemelor distribuite,
- operațiuni bazată pe date,
- utilizarea dispozitivelor conectate în rețea,
- utilizând protocoale standard de comunicație.

Utilizarea tehnologiilor IoT le permite studenților să nu lucreze cu dispozitive izolate, ci să interpreteze componentele hardware și software ca parte a unui sistem coerent.

Kitul Smart Farm a fost conceput în mod deliberat ca punct de plecare. Unul dintre elementele sale cheie a fost analiza sistemului din fabrică, identificarea limitelor acestuia și apoi răspunsul la acestea prin dezvoltarea propriei arhitecturi scalabile de sistem, care urmează un model industrial. Acest proces a evidențiat faptul că gândirea critică și dezvoltarea iterativă sunt fundamentele progresului tehnologic.

Abordarea este în întregime bazată pe proiecte. Elevii nu urmează pași predefiniți, ci mai degrabă rezolvă probleme, iau propriile decizii și analizează consecințele acestora. În timpul procesului de învățare, greșelile nu sunt văzute ca eșecuri, ci ca o parte naturală a dezvoltării.

Modelele fizice, feedback-ul imediat și rezultatele vizibile sporesc în mod natural motivația elevilor. Elementele de gamificare, cum ar fi provocările, etapele importante și obiectivele comune, apar pe tot parcursul proiectului pentru a structura și menține interesul.

Dezvoltă competențele digitale într-un mod complex: programarea, gestionarea bazelor de date, cunoștințele de rețea, designul digital și documentația sunt toate prezente. De asemenea, consolidează gândirea inginerescă, gândirea sistemică și recunoașterea relațiilor dintre unitățile funcționale.

Proiectul a fost implementat în cadrul cooperării internaționale, ceea ce a adus o valoare adăugată semnificativă în ceea ce privește dezvoltarea competențelor și conținutul pedagogic, pe lângă conținutul profesional. Instituțiile participante operează în țări, sisteme educaționale și profiluri instituționale diferite, astfel încât munca în comun a creat un mediu de colaborare real, care modelează bine situațiile viitoare ale pieței muncii.

Diferențele lingvistice au fost proeminente în timpul activităților comune. În multe cazuri, comunicarea profesională a avut loc în limba engleză, ceea ce a îmbunătățit competențele studenților în limbi străine și i-a făcut conștienți de importanța unei comunicări tehnice precise și clare.

Colaborarea multiculturală a devenit o parte naturală a procesului de lucru. Munca în comun a studenților cu medii educaționale diferite și strategii de rezolvare a problemelor a consolidat flexibilitatea, adaptabilitatea și dialogul profesional.

Funcționarea sa a reflectat așteptările companiilor multinaționale în mai multe aspecte: comunicarea în limbi străine, gestionarea comună a documentației, munca în echipă și responsabilitatea în sisteme complexe. Experiențele acumulate au contribuit la capacitatea studenților de a-și interpreta cunoștințele tehnice într-un context internațional mai larg.

2. Structura proiectului și logica de implementare

2.1. Fazele proiectului și interdependența lor

Proiectul a fost implementat în etape construite în mod conștient una peste cealaltă. Această structură nu a fost doar o necesitate tehnică, ci și o decizie pedagogică: scopul a fost ca elevii să ajungă treptat, bazându-se pe propriile experiențe, la sarcini de dezvoltare independentă. Etapizarea a permis implicarea elevilor cu medii diferite, precum și reflecția și evaluarea aferente fiecărei etape.

În prima fază, studenții au întâlnit un sistem funcțional, gata preparat. Scopul a fost de a învăța conceptele de bază, relațiile operaționale și logica sistemului. Accentul nu a fost pus pe „copierea” soluției, ci pe înțelegerea: ce face sistemul, de ce o face, din ce elemente este alcătuit și cum sunt conectate acestea între ele.

Această etapă s-a dovedit a fi deosebit de importantă pentru studenții cu cunoștințe mai puțin tehnice, deoarece a creat un punct de plecare comun și a redus bariera de acces pentru sarcini ulterioare, mai complexe.

A doua fază, faza de analiză conștientă, a urmat fazei de familiarizare. Studenții nu numai că au utilizat sistemul, dar au examinat și limitele sale operaționale, punctele forte și deficiențele. Această etapă a jucat un rol cheie în înțelegerea faptului că o soluție tehnică este întotdeauna rezultatul unor compromisuri și este optimizată pentru scopuri specifice.

Prin evaluare critică, au învățat să-și formuleze propriile nevoi de dezvoltare și să recunoască momentul potrivit pentru dezvoltarea sau regândirea în continuare a unui sistem existent. Acest mod de gândire este direct legat de principiile ingineriei și de rezolvarea problemelor IT.

În a treia fază a proiectului, studenții au început propria activitate de dezvoltare, bazându-se pe experiențele lor analitice. Aici, procesul nu a mai fost ghidat de pași predeterminați, ci de decizii luate în comun, planificare iterativă și testare continuă.

În această etapă, avantajul abordării bazate pe proiect a devenit cu adevărat vizibil: studenții au experimentat faptul că un sistem complex nu este „finalizat” dintr-o dată, ci prinde contur treptat, iar fiecare decizie are un impact asupra întregii operațiuni.

2.2. Participanți și roluri

O caracteristică definitorie a proiectului a fost împărțirea conștientă a rolurilor între participanți. Studenții, profesorii și instituțiile au lucrat împreună în roluri diferite, dar care se consolidau reciproc, ceea ce a contribuit la flexibilitatea și sustenabilitatea proiectului.

Pe parcursul participării lor, studenții nu au fost receptori pasivi, ci dezvoltatori activi. Au colaborat la planificare, interpretarea problemelor, testarea și evaluarea soluțiilor. În timpul distribuirii subsarcinilor, au experimentat importanța asumării responsabilității și modul în care munca lor se încadrează într-un sistem mai complex. Acest rol a consolidat independența, cooperarea și gândirea pe termen lung.

Rolul instructorilor a fost în primul rând de sprijin și îndrumare. Accentul a fost pus pe îndrumare, adresarea de întrebări și structurarea procesului de învățare, mai degrabă decât pe furnizarea de soluții predefinite.

Această abordare le-a permis studenților să găsească soluții prin propriile experiențe, oferindu-le în același timp o bază profesională și pedagogică solidă.

Cooperarea internațională a dus la crearea unor grupuri mixte de învățare, în care studenți cu medii educaționale și experiențe diferite au lucrat împreună. Această situație a consolidat abilitățile de comunicare, adaptabilitatea și rezolvarea comună a problemelor.

Munca în echipă mixtă a însoțit întregul proiect și a creat situații reale de colaborare care au depășit cadrul școlar tradițional.

2.3. Rezultatele învățării și competențele care trebuie dezvoltate

În timpul proiectării, a existat un obiectiv conștient de a nu limita rezultatele învățării la o singură arie de competențe. Procesul de dezvoltare a susținut dezvoltarea simultană a abilităților tehnice, cognitive și sociale.

Proiectul a contribuit la utilizarea conștientă a instrumentelor digitale, la munca structurată și la creație într-un mediu digital. Studenții au experimentat cum un sistem complex devine ușor de înțeles și de gestionat.

S-au confruntat cu probleme complexe care necesitau mai mulți pași, planificare și feedback pentru a fi rezolvate. Acest lucru a consolidat gândirea sistemică, recunoașterea relațiilor cauză-efect și capacitatea de a planifica pe termen lung.

Documentarea și prezentarea lucrărilor finalizate a fost o parte integrantă a proiectului. Studenții au învățat cum să înregistreze în mod transparent un proces de dezvoltare și să îl prezinte într-o formă ușor de înțeles. Portofoliile rezultate permit, de asemenea, utilizarea pe termen lung a rezultatelor învățării.

3. Prezentarea detaliată și prelucrarea Kitului Smart Farm

3.1. Rolul Kitului Smart Farm în procesul de învățare

Kitul Smart Farm este un kit modular IoT conceput în scopuri educaționale, demonstrând principiile de bază ale sistemelor inteligente printr-un model tangibil. Kitul își propune să expună elevii la un sistem real, funcțional, înainte de a începe să dezvolte propriile soluții.

În cadrul proiectului, kitul nu este prezentat ca produs final, ci ca instrument de învățare. În timpul procesării, accentul se pune pe înțelegerea funcționării elementelor, cartografierea logicii sistemului și punerea bazelor pentru dezvoltarea ulterioară.

3.2. Componentele principale ale kitului Smart Farm

Componentele electronice ale kitului Smart Farm sunt concepute pentru a demonstra un sistem IoT complex, incluzând mai mulți senzori de intrare, module de control și actuatore de ieșire. Lista se bazează pe sursa producătorului și pe descrieri fiabile ale produselor și acoperă dispozitivele utilizate în timpul programului de învățare al kitului.

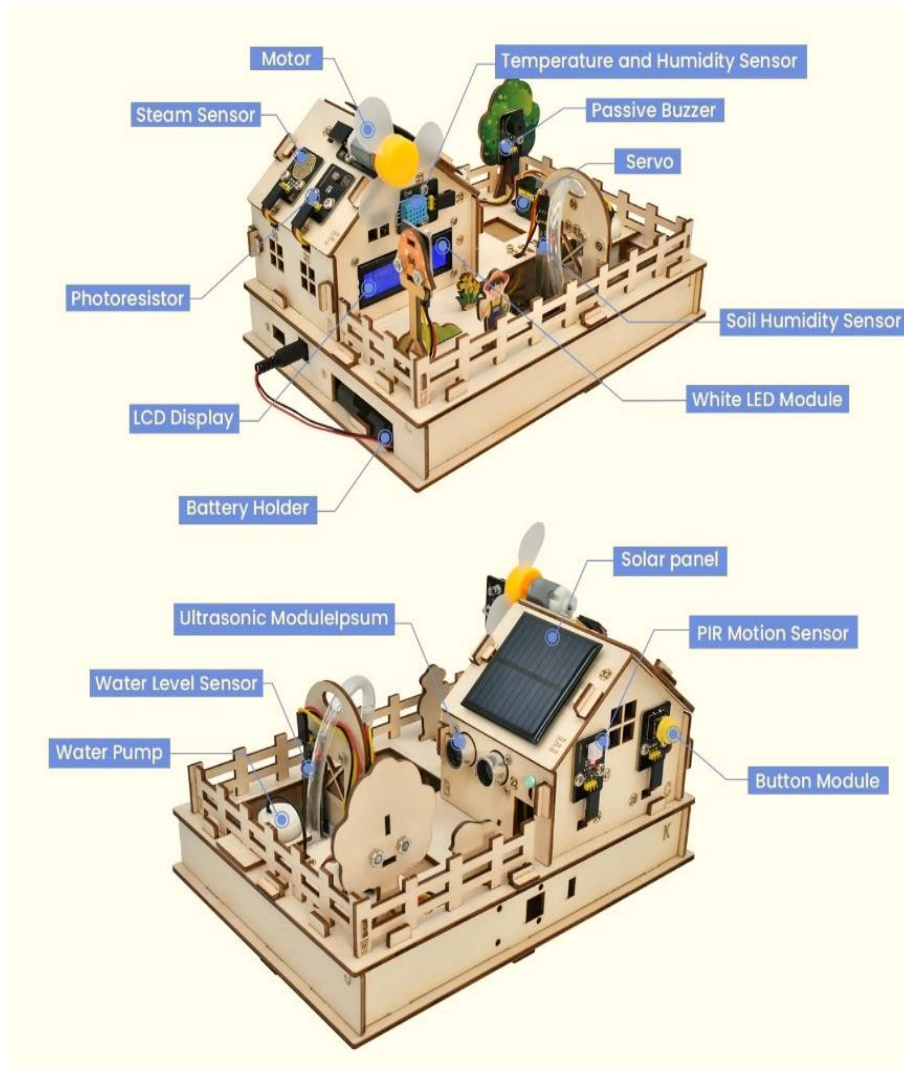


Figura 1: Părțile principale ale fermei inteligente Kif (Sursa:<https://www.keyestudio.com>)

3.3. Module de control și bază

- Microcontroler ESP32:
 - e| Structură: Microcontroler integrat cu modul Wi-Fi și Bluetooth
 - e| Funcție: „Creierul” kitului, care gestionează datele senzorilor, logica și comunicarea.
 - e| Semnificație educațională: Studiul comunicării în rețea, al gestionării evenimentelor paralele și al logicii I/O.

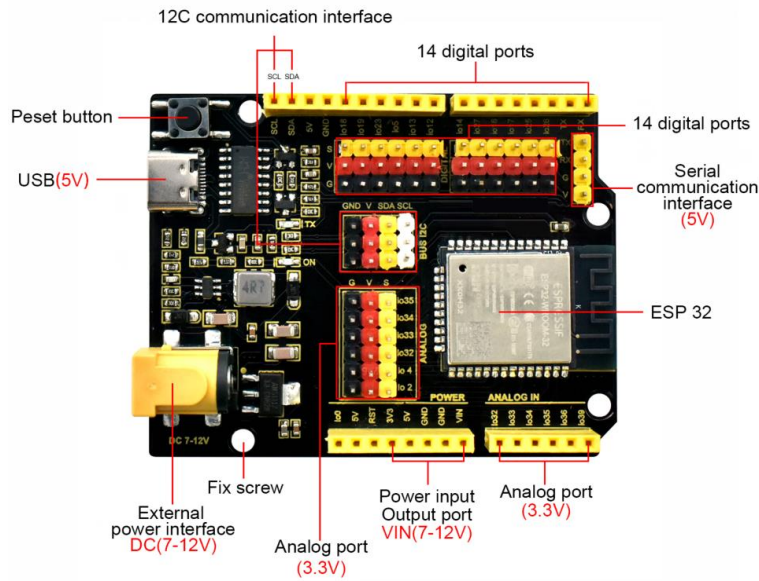


Figura 2: Microcontroler ESP32 (Sursa: <https://docs.keyestudio.com>)

- Modul releu:
 - e| Construcție: Releu optozolat cu control de 5 V
 - e| Funcție: Comutare dispozitive cu consum mai mare (de exemplu, pompă). Scop educațional: Izolarea și controlul sigur al actuatorilor.

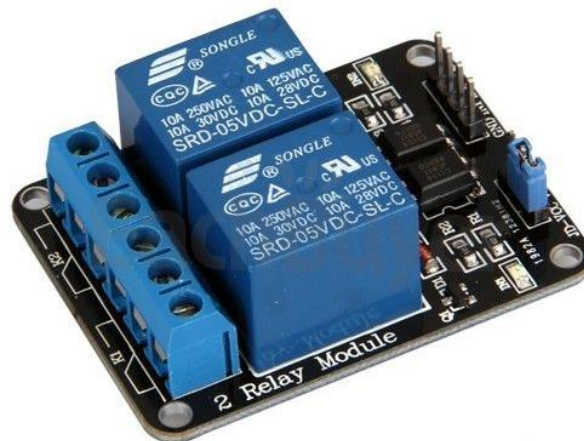


Figura 3: Modul releu (Sursa: <https://docs.keyestudio.com>)

3.4. Elemente de intrare – Senzori

- Fotorezistor: el
 - Construcție: rezistor semiconductor fotosensibil
 - e| Funcționare: rezistența se modifică în funcție de intensitatea luminii (semnal analogic). Scop
 - e| educațional: măsurarea condițiilor de lumină, gestionarea intrărilor analogice.



Figura 4: Fotorezistor (Sursa: <https://docs.keyestudio.com>)

- Senzor de umiditate a solului: el
Structură: doi electrozi și traductor
el Funcționare: apa din sol reduce rezistența → valoare a semnalului mai mare. Scop
el educațional: tehnici de măsurare analogică și interpretarea datelor de mediu.



Figura 5: Senzor de umiditate a solului (Sursa: <https://docs.keyestudio.com>)

- Senzor de nivel al apei:
el Structură: contact cu mai mulți electrozi
el Funcționare: semnal de închidere/deschidere bazat pe nivelul
el lichidului. Scop educațional: măsurare digitală bazată pe evenimente.

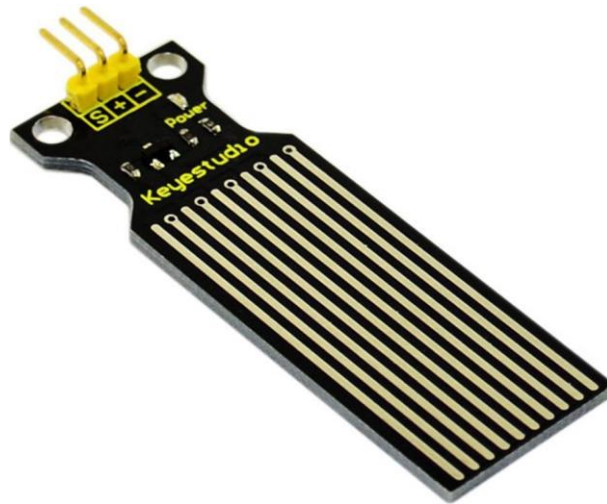


Figura 6: Senzor de nivel al apei (Sursa: <https://docs.keyestudio.com>)

- Senzor de temperatură și umiditate DHT11: el
Structură: senzor digital integrat în unitate
el Funcție: calculează temperatura și umiditatea cu un algoritm integrat. Scop
el educațional: comunicarea de date digitale și măsurarea mediului fizic.

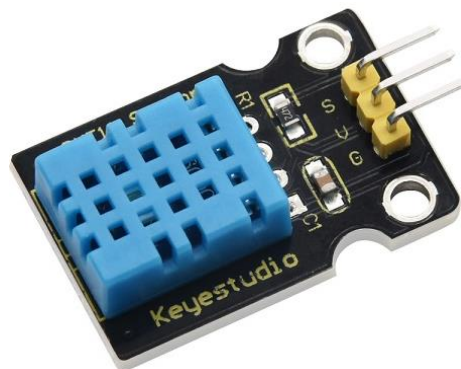


Figura 7: Senzor de temperatură și umiditate DHT11 (Sursa: <https://docs.keyestudio.com>)

- Senzor de distanță cu ultrasunete SR01 V3: el
Structură: elemente piezoelectrice emițător și receptor.
el Funcționare: impuls ultrasonic → reflexie → calcularea distanței din măsurarea
timpului
el Scop educațional: înțelegerea undelor fizice și a măsurării bazate pe timp.



Figura 8: Senzor de distanță cu ultrasunete SR01 V3 (Sursa: <https://docs.keyestudio.com>)

- Senzor de mișcare PIR:

- e| **Construcție:** senzor infraroșu pasiv
- e| **Funcție:** Detectează schimbările bruște de temperatură în câmpul vizual. Obiectiv
- e| **educațional:** Logică bazată pe evenimente și detectare în timp real.



Figura 9: Senzor de mișcare PIR (Sursa: <https://docs.keyestudio.com>)

- **Buzzer pasiv: e|**
 - Construcție:** difuzor piezo
 - e| **Funcționare:** răspunde la impulsuri de tensiune prin emiterea de sunet. Obiectiv
 - e| **educațional:** conectarea dintre modulele de ieșire și sistemul practic de semnalizare.

3.5. Elemente de ieșire – Actuatoare și afișaje

- **Modul LED alb:**
 - e| **Construcție:** LED premontat
 - e| **Funcționare:** ieșirea digitală controlează iluminatul Scop
 - e| **educațional:** feedback vizual pentru logică simplă.



Figura 10: Modul LED alb (Sursa: <https://docs.keyestudio.com>)

- **Servomotor SG90 9G: e|**
 - Structură:** microservomotor cu feedback de poziție
 - e| **Funcționare:** control al poziției bazat pe semnal PWM

elScop educațional: controlul precis al actuatorilor și al efectelor mecanice.



Figura 11: Servomotor SG90 9G (Sursa: <https://docs.keyestudio.com>)

- Motor/ventilator de curent continuu:
 - el Construcție: Motor de curent continuu cu ventilator mic
 - el Funcție: dispozitiv de ieșire care servește la mișcarea aerului înconjurător. Scop
 - el educațional: controlul motorului și monitorizarea performanței.

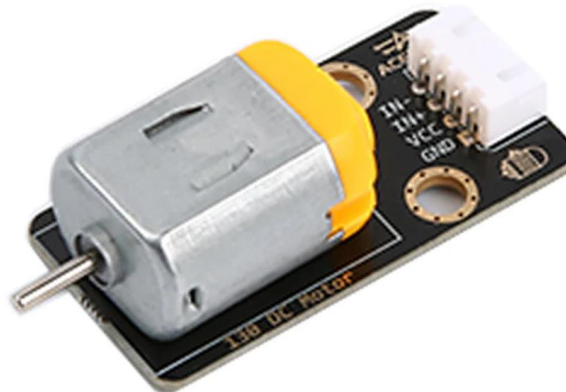


Figura 12: Motor/ventilator de curent continuu (Sursa: <https://docs.keyestudio.com>)

- Afișaj LCD I2C 1602:
 - el Structură: 2 linii × 16 caractere, comunicare I2C
 - el Funcție: afișarea datelor de caractere din mesajele microcontrolerului. Scop
 - el educațional: învățarea protocoalelor de afișare a datelor și de comunicare.

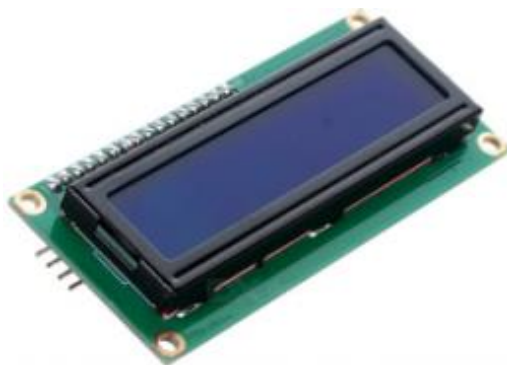


Figura 13: Afișaj LCD I2C 1602 (Sursa: <https://docs.keyestudio.com>)

- Panou solar:
 - eI Structură: panou fotovoltaic mic Funcționare: efect
 - eI fotovoltaic → electricitate Scop educațional: surse
 - eI regenerabile de energie în sistemele IoT.



Figura 14: Panou solar (Sursa: <https://docs.keyestudio.com>)

- Pompă de apă:
 - eI Construcție: pompă mică de curent continuu
 - eI Funcționare: transfer de lichid cu control prin releu
 - eI Scop educațional: înțelegerea controlului actuatorilor și a lucrului mecanic.



Figura 15: Pompă de apă (Sursa: <https://docs.keyestudio.com>)

3.6. Accesorii și unelte de montare

- elemente acrilice transparente și acoperire cu așchii de lemn
- cabluri de alimentare, fire DuPont, șuruburi, elemente de fixare
- suport pentru baterie sau suport pentru baterie
- Cablu USB pentru comunicare și alimentare

3.7. Rolul componentelor în educație

Cele 17 componente electronice incluse în kit acoperă bine programa introductivă în IoT și automatizare:

- senzorii asigură transmiterea semnalelor din mediul fizic,
- unitatea de control interpretează și decide,
- actorii afișează sau intervin în mediu.

Această structură oferă o imagine de ansamblu sistematică asupra modului în care elementele funcționează împreună într-un sistem IoT real și cum pot fi conectate prin programare, logica de măsurare și control.

Asamblarea Kitului Smart Farm este primul pas în cadrul proiectului, în care studenții lucrează cu un sistem fizic complex, cu mai multe componente. Documentația din fabrică ghidează utilizatorul pas cu pas prin procesul de asamblare a elementelor mecanice și electronice, ceea ce poate fi interpretat și ca un proces de învățare conștientă.

Asamblarea nu este doar o sarcină tehnică, ci mai degrabă o experiență care pune bazele pentru lucrări ulterioare independente de proiectare și fabricație.

3.8. Rolul instrucțiunilor de asamblare din fabrică

Manualul de fabricație al kitului Smart Farm servește ca referință principală în timpul asamblării. Manualul:

- determină secvența fiecărei etape de asamblare,
- arată poziția și fixarea corectă a pieselor,
- ajută la conectarea în siguranță a componentelor electronice,
- susține înțelegerea cu ilustrații vizuale.

În timpul utilizării ghidului, studenții vor experimenta faptul că realizarea unui sistem ingineresc nu este improvizație, ci un proces structurat, bazat pe documentație.



Figura 16: Instrucțiuni de asamblare din fabrică (Sursa: <https://www.keyestudio.com>)

3.9. Asamblare mecanică și electronică și proiectare modulară

În timpul asamblării, studenților li se va prezenta designul mecanic modular al kitului. Fiecare element are puncte de montare pre-proiectate pe baza distanțelor standard de conectare, permițând înlocuirea și relocarea componentelor.

Această abordare modulară transmite un principiu ingineresc important: extensibilitatea și mentenabilitatea sistemului sunt decise la nivel de proiectare fizică.

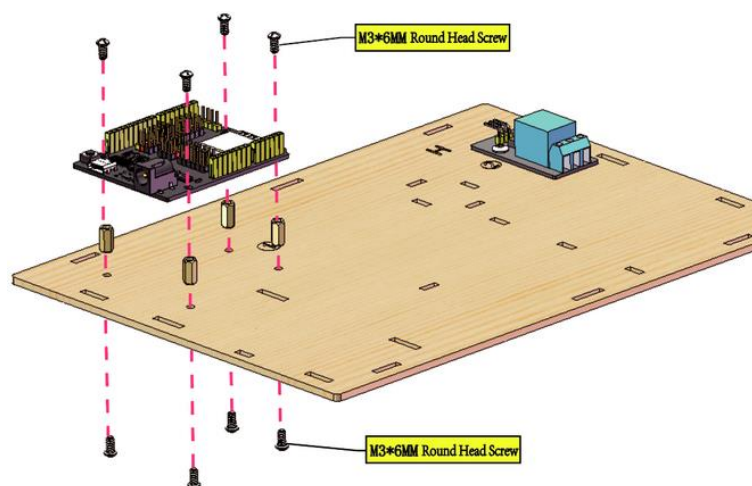


Figura 17: Placă de bază pentru casă inteligentă (Sursa: <https://docs.keyestudio.com>)

În timpul asamblării componentelor electronice, elevii identifică punctele de conectare ale senzorilor și actuatorilor, învață despre rolul sursei de alimentare și al firelor de semnal și urmează schemele de conectare din fabrică.

În timpul conexiunilor, se acordă o atenție deosebită gestionării corecte a polarității, distincției dintre intrările digitale și cele analogice, precum și protecției mecanice și clarității firelor.

Acest pas pune bazele pentru depanarea ulterioară a problemelor și pentru gândirea diagnostică a sistemului.

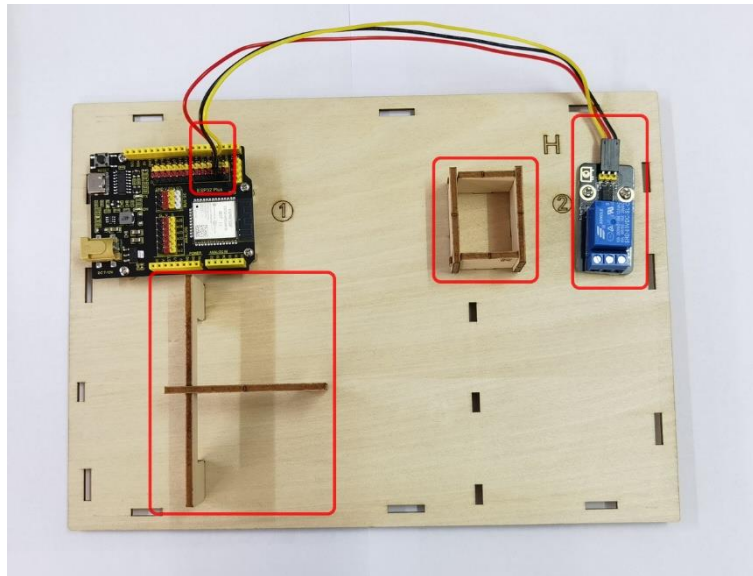


Figura 18: Puncte de conectare la placa de bază Smart Home (Sursa: <https://docs.keystudio.com>)

Asamblarea nu este un proces singular, închis, ci o activitate iterativă. După fiecare etapă de asamblare, elevii:

- verifică stabilitatea mecanică,
- ei testează conexiunile electrice,
- Funcționarea fiecărui element este validată cu programe de testare scurte.

Această abordare previne erorile complexe ulterioare și atrage atenția asupra importanței verificării treptate.

3.10. Considerații ingineresti bazate pe experiența de asamblare

În timpul procesului de asamblare, devin observabile mai multe aspecte ingineresti care pot fi utilizate direct în crearea ulterioară a elementelor proiectate de noi:

- Ajustarea geometrică, precizia dimensiunilor și punctele de montare ale componentelor afectează fundamental asamblarea.
- Proiectarea traseelor de cablare și aranjarea firelor afectează transparența, mentenanța și estetica.
- Accesibilitatea și ușurința în service, amplasarea senzorilor și a comenzilor afectează posibilitatea unor modificări și extinderi ulterioare.

- În ceea ce privește modularitatea și interschimbabilitatea, soluția din fabrică evidențiază modul în care un sistem poate fi construit astfel încât elementele individuale să poată fi operate independent.

Aceste experiențe îi fac pe studenți conștienți de faptul că designul fizic și funcționarea formează o unitate inseparabilă.

Pe măsură ce faza de asamblare se încheie, studenții nu mai sunt doar utilizatori, ci și analiști ai sistemului. Experiențele acumulate aici îi pregătesc în mod direct pentru:

- măsurătorile geometrice ulterioare,
- propriile sarcini de design 3D,
- proiectarea elementelor individuale de fixare și a carcasei,
- dezvoltarea ulterioară sau înlocuirea soluțiilor din fabrică.

Prin urmare, asamblarea Kitului Smart Farm nu este un punct final, ci un punct de plecare pentru un proces creativ de inginerie independent, de nivel superior.

3.11. Programarea Smart Farm Kit Mediu

3.12. de programare bazat pe blocuri

Programarea bazată pe blocuri joacă un rol important în înțelegerea sistemelor și în tranziția către alte modele de programare. Mediul grafic de programare al Smart Farm Kit (KidsBlock) urmează logica bazată pe Scratch. Programarea se face prin îmbinarea blocurilor vizuale, care sunt susținute de instrucțiuni de control reale, executabile. Acest lucru permite elevilor să evite să se ocupe de detaliile sintactice ale limbajelor de programare în timp ce controlează funcționarea propriu-zisă a sistemului.

Dintr-o perspectivă educațională, acest mediu îndeplinește mai multe funcții care se consolidează reciproc:

- Separă gândirea logică de dificultățile lingvistice, astfel încât elevii se pot concentra pe procesele de control, relațiile cauză-efect și schimbările de stare.
- Oferă feedback imediat, deoarece funcționarea sistemului fizic poate fi observată direct după modificarea programului.
- Susține experimentarea iterativă, ciclul natural de învățare încercare-eroare-îmbunătățire.
- Te pregătește pentru programarea textuală ulterioară, deoarece structuri de bază precum gestionarea evenimentelor, testarea condițiilor, buclele și gestionarea stărilor apar în spatele blocurilor.

Programarea bazată pe blocuri în acest proiect nu este un punct final, ci un nivel de intrare ales în mod conștient pe o cale de învățare a programării în mai mulți pași.

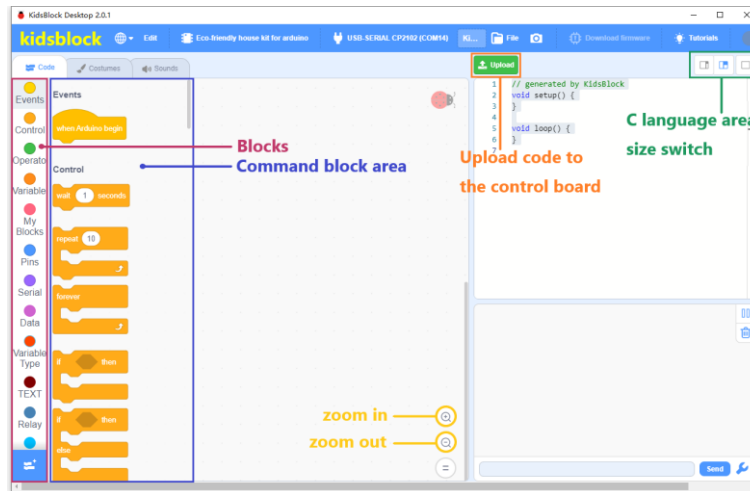


Figura 19: Interfață de programare bazată pe blocuri, prezentare generală a principalelor categorii de blocuri (Sursa: <https://docs.keyestudio.com>)

3.13. Conectarea la medii de programare bazate pe text

Unitatea de control care stă la baza kitului Smart Farm, microcontrolerul ESP32, nu este programabilă doar într-un mediu grafic. Proiectul oferă și oportunitatea de a implementa aceleași funcții într-un limbaj de programare bazat pe text, cum ar fi un mediu bazat pe Arduino.

Mediul Arduino:

- Folosește sintaxa bazată pe C/C++,
- permite gestionarea hardware-ului la nivel scăzut,
- Îți oferă informații directe despre execuția programului, variabile și utilizarea memoriei.

Din punct de vedere pedagogic, această schimbare:

- te ajută să înțelegi că în spatele elementelor bazate pe blocuri rulează cod specific,
- dezvoltă capacitatea de abstractizare (corespondență bloc ↔ linie de cod),
- te pregătește pentru utilizarea unor medii de dezvoltare mai profesionale.

În timpul implementării, programarea bazată pe blocuri și cea bazată pe text apar ca instrumente complementare, mai degrabă decât să se excludă reciproc.

Valoarea educațională a proiectului Smart Farm este sporită și mai mult de faptul că logica de control învățată de studenți este strâns legată de principiile de programare utilizate în automatizarea industrială.

Oportunitățile profesionale sunt extinse și mai mult prin platforma software open source OpenPLC, care permite programarea IEC 61131-3 nu numai pe PLC-uri clasice, ci și pe controlere bazate pe Arduino și ESP32. Mediul OpenPLC oferă suport complet pentru limbajul IEC 61131-3, permițând astfel practica logicii de control industrial pe hardware low-cost și acceptă protocoale de comunicație industrială, în special standardele Modbus și Modbus TCP.

Această abordare face legătura dintre dezvoltarea microcontrolerelor educaționale și sistemele de automatizare industrială din lumea reală. Studenții pot învăța despre

cu programare industrială, care este mai simplă din punct de vedere tehnologic, dar pe deplin compatibilă cu sistemele profesionale în abordarea sa.

3.14. Legătură către etapele ulterioare ale proiectului

În fazele ulterioare ale proiectului Smart Farm, comunicarea industrială va juca un rol cheie. Prin urmare:

- Utilizarea protocolului Modbus / Modbus TCP stă la baza integrării sistemului,
- Programarea în limbajul C, bazată pe Arduino, este încă utilizată în controlul microcontrolerelor,
- Abordarea bazată pe blocuri și cea conformă cu IEC 61131-3 servesc în principal ca un cadru de gândire și proiectare.

Această decizie garantează că studenții învață despre standardele și așteptările industriei în timp ce lucrează într-un mediu de dezvoltare flexibil și prietenos cu educația și se pregătesc pentru lucrul ulterior cu sisteme IoT industriale sau bazate pe PLC.

Proiectul Smart Farm folosește programarea bazată pe blocuri nu izolat, ci integrată într-un context industrial mai larg. Introducerea standardului IEC 61131-3, a platformei/proiectului OpenPLC și a comunicării bazate pe Modbus permite studenților să dobândească o abordare sistemică la un nivel intermediar care corespunde direct așteptărilor mediului industrial modern, multinațional, menținând în același timp natura sigură, graduală și motivantă a educației.

3.15. Proiectul 1 – Sistem de iluminat

Controlul iluminatului este unul dintre cele mai simple, dar importante din punct de vedere educativ, proiecte exemplu din cadrul Smart Farm Kit. În această sarcină, lanțul de control intrare digitală → logică de decizie → ieșire digitală apare pentru prima dată ca o unitate completă, închisă. Scopul proiectului nu este de a înțelege iluminatul ca o funcție, ci de a înțelege logica de bază a automatizării.

Butonul ca intrare digitală

Un buton este cel mai simplu dispozitiv digital de intrare care poate transmite două stări distincte către controler:

- stare neapăsată → logic LOW (0),
- stare apăsată → logic HIGH (1).

Pentru controler (ESP32), butonul nu este un „obiect fizic”, ci un semnal digital care apare pe un pin de intrare specific. Starea intrării este de obicei stabilizată de o rezistență internă sau externă de tracțiune (pull-up/pull-down), astfel încât controlerul detectează în permanență o valoare clară HIGH sau LOW.

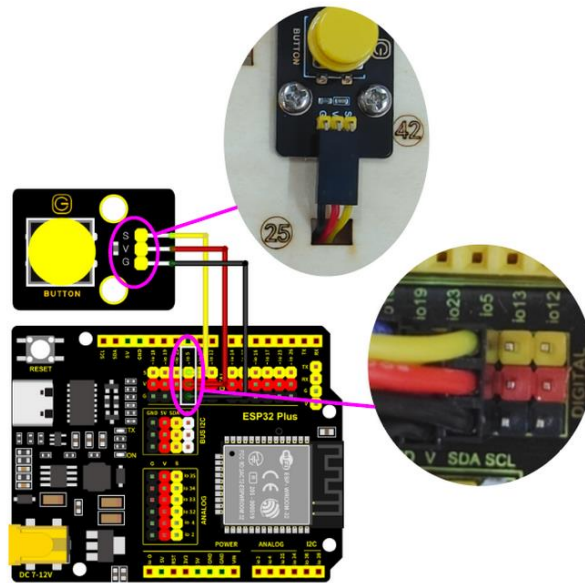


Figura 20: Schema de conectare a butonului de comandă ca intrare digitală (ESP32 GPIO) (Sursa: <https://docs.keyestudio.com>)

Butonul este excelent pentru introducerea conceptului de semnal digital, deoarece nu există o valoare „intermediară”, starea poate fi întotdeauna interpretată clar, iar informația nu provine din magnitudinea valorii, ci din schimbarea de stare.

Această abordare este în contrast puternic cu senzorii analogici, unde semnalul este continuu și necesită interpretare numerică. Înțelegerea diferenței dintre cei doi este esențială în sistemele de automatizare a învățării.

Butonul este doar una dintre numeroasele intrări digitale. În timpul proiectului – sau ca o dezvoltare ulterioară a acestuia – următoarele dispozitive funcționează, de asemenea, pe un principiu similar:

- Releu Reed (contact magnetic): închide sau deschide un circuit sub influența unui câmp magnetic; aplicația tipică este detectarea ușilor și ferestrelor.
- Comutator de limită: utilizat pentru a detecta poziția limită a mișcării mecanice.
- Ieșire digitală a senzorului de mișcare PIR: emite un semnal HIGH (nivel înalt) atunci când este detectată mișcare, altfel rămâne în starea LOW (nivel scăzut).
- Feedback de contact prin releu digital: o formă comună de feedback în sistemele industriale.

Trăsătura comună a acestor dispozitive este că oferă informații despre stare, nu o cantitate măsurată.

Semnificația educațională a utilizării intrărilor digitale – în special a butoanelor – constă în faptul că dezvoltă mai multe modele fundamentale de gândire:

- gândire bazată pe evenimente (se întâmplă ceva → urmează o reacție),
- conceptul de bază al interfeței om-mașină,
- abstractizarea, în timpul căreia acțiunea fizică este transformată în informație digitală,
- înțelegerea mașinii de stări ulterioare și a logicii de control.

În acest moment, studenții înțeleg mai întâi că un sistem automatizat nu „monitorizează constant”, ci răspunde la evenimente, ceea ce reprezintă un principiu fundamental al controalelor industriale.

Ieșire digitală – LED ca actuator controlat

LED-ul (diodă emițătoare de lumină) este unul dintre cele mai simple dispozitive digitale de ieșire din kitul Smart Farm, oferind o reprezentare vizuală a deciziilor controlerului. Funcționează în două stări:

- ieșire LOW → LED stins,
- ieșire HIGH → LED aprins.

Pinul de ieșire digitală al controlerului (ESP32) acționează LED-ul, de obicei, printr-o rezistență de limitare a curentului, care protejează componenta de supracurent. Aici, studenții se confruntă pentru prima dată cu faptul practic că controlerul nu poate acționa direct nicio sarcină, ci trebuie proiectat ținând cont de mediul electronic.

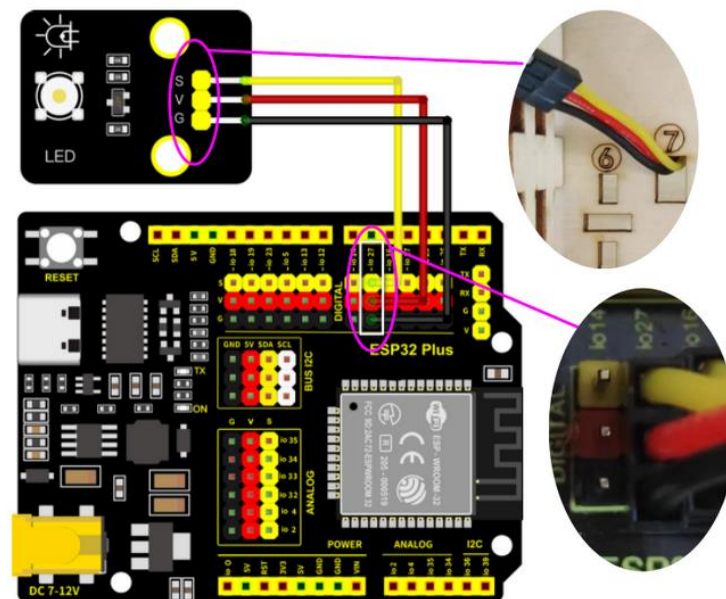


Figura 21: Schema de conectare a LED-ului pentru ieșirea digitală (ESP32 GPIO + rezistență de limitare a curentului) (Sursa: <https://docs.keyestudio.com>)

Conceptul de ieșire digitală este fundamental diferit de intrări. În timp ce intrările oferă informații despre sistem, ieșirile reprezintă o intervenție activă, deoarece sistemul nu numai că simte, ci și își influențează mediul.

În această etapă, LED-ul nu este folosit pentru iluminare, ci mai degrabă ca indicator de stare. Studenților le devine clar că starea LED-ului este o reprezentare a logicii interne a programului, deoarece ieșirea este întotdeauna rezultatul unui proces decizional. Feedback-ul fizic joacă un rol cheie în înțelegere și depanare.

Când controlează LED-ul, elevii experimentează pentru prima dată întregul lanț de control:

1. eveniment de introducere (de exemplu, apăsarea unui buton),
2. procesare / decizie (verificare condiție, verificare stare),
3. intervenție la ieșire (aprinderea sau stingerea LED-ului).

Acest lanț servește ca model abstract pentru sisteme ulterioare, mai complexe (controlul motoarelor, rele, automatizare).

Dintr-o perspectivă educațională, LED-ul, ca ieșire digitală, pregătește înțelegerea următoarelor dispozitive:

- module de relee (comutare dispozitive de putere mai mare),
- buzzer / sirene,
- semafoare (roșu-galben-verde),
- linii de control pentru afișaje digitale.

Studentii vor realiza astfel că LED-ul este doar un model simplificat, în spatele căruia se află același principiu de control ca în cazul actuatorilor industriale. Aplicația sa se dezvoltă:

- gândirea cauză-efect,
- abordarea sistemică (intrare → procesare → ieșire),
- interpretarea feedback-ului vizual,
- capacitatea de a abstractiza, ceea ce ne permite ulterior să abandonăm dispozitivele fizice și să înțelegem sistemele simulate.

În acest moment, studenții înțeleg că un sistem automatizat „nu gândește” ci reacționează conform unor reguli predefinite și că rezultatul în fiecare caz este o consecință a stării logice a programului.

Controlul iluminării – conectarea intrării și ieșirii

Următorul pas în cadrul proiectului este conectarea butonului și a LED-ului cu o logică de control simplă, dar completă. Această sarcină este prima care prezintă modelul de gândire intrare - procesare - ieșire ca un sistem unificat.

Modelul logic de control al iluminatului este alcătuit din următoarele elemente:

- eveniment: schimbarea stării butonului,
- decizie: verificarea stării actuale a LED-ului,
- intervenție: aprindeți sau stingeți LED-ul.

Această structură descrie deja un ciclu complet de automatizare, chiar dacă implementarea este simplă din punct de vedere tehnic.

Studenții ar trebui să fie conștienți de faptul că sarcina în sine nu necesită un microcontroler. Un buton și un LED pot fi conectate ca un circuit simplu cablat, unde butonul închide direct circuitul, iar LED-ul răspunde imediat la comutarea fizică. Totuși, aceasta nu este o automatizare, ci o operare hardware directă.

Scopul implicării microcontrolerului nu este de a „rezolva” sarcina, ci de a modela întregul lanț de efecte, adică digitalizarea unui eveniment fizic, luarea deciziilor logice în software și executarea intervențiilor programate. Această distincție este esențială în dezvoltarea înțelegerii sistemului de către studenți.

Elemente de programare (într-un mediu bazat pe blocuri)

În timpul programării bazate pe blocuri, apar următoarele elemente:

- blocuri declanșatoare de evenimente (apăsarea unui buton sau schimbarea stării),
- blocuri de control al ieșirilor digitale (LED pornit/oprit),
- variabilă de stare opțională care stochează starea curentă a LED-ului.

Introducerea variabilei nu este o constrângere tehnică, ci o decizie de proiectare conștientă care separă evenimentul (ce s-a întâmplat) și starea (care este starea curentă a sistemului).

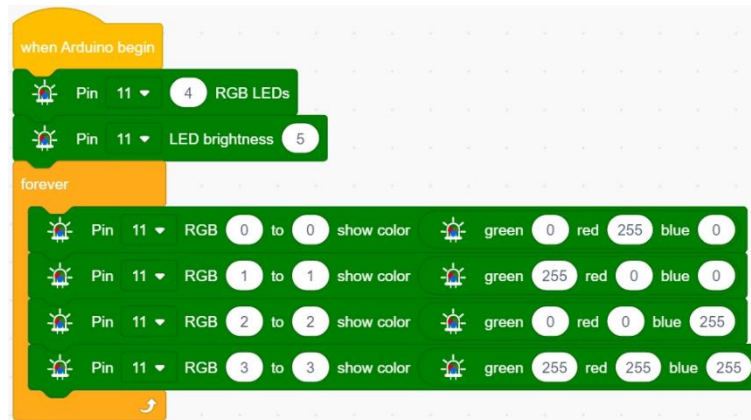


Figura 22: Structura completă a unui program de bloc de control al iluminatului, cu utilizare variabilă (Sursa: <https://docs.keyestudio.com>)

În timpul sarcinii, studenții își dau seama rapid că controlerul nu „își amintește” automat stările anterioare, ci că păstrarea stării este o responsabilitate a programării, ceea ce înseamnă că evenimentul și starea sunt concepte care trebuie tratate separat.

Această experiență este o pregătire directă pentru automatizarea cu stări multiple, controalele temporizate și sistemele asincrone în rețea.

Reflecție educațională

În ciuda simplității sale, proiectul de control al iluminatului are o valoare educațională remarcabilă. Studenții experimentează faptul că sistemele digitale nu gândesc, ci execută, ceea ce înseamnă că funcționarea sistemului depinde în întregime de logica proiectată, astfel încât chiar și o sarcină aparent banală se bazează pe decizii ingineresti conștiente.

Compararea unei soluții cablate cu o implementare bazată pe controler îi ajută pe studenți să înțeleagă că automatizarea nu se rezumă la instrumente, ci la mentalitate.

Această parte a proiectului creează o bază solidă pentru înțelegerea sistemelor de control ulterioare bazate pe senzori, pe condiții și în rețea și oferă o tranziție naturală către soluții IoT mai complexe.

3.16. Proiectul 2 – Sistem de control al luminii

Acest proiect introduce pentru prima dată măsurarea și procesarea mărimilor analogice, ceea ce reprezintă o diferență fundamentală față de sarcinile anterioare bazate exclusiv pe logica digitală. Aici, studenții nu mai lucrează cu semnale cu două stări, ci interpretează și transformă mărimi fizice în continuă schimbare în decizii de control.

Funcționare senzor de lumină (LDR) – intrare analogică

Un LDR (rezistor dependent de lumină) este un rezistor pe bază de semiconductori a cărui valoare a rezistenței variază în funcție de intensitatea luminii care cade pe el:

- iluminare puternică → rezistență mai mică,
- lumină slabă → rezistență mai mare.

Această schimbare este continuă, deci este de natură analogică. ESP32 nu „vede” luminozitatea direct, ci măsoară tensiunea care apare pe circuitul divizor de tensiune format cu LDR-ul pe intrarea sa analogică.

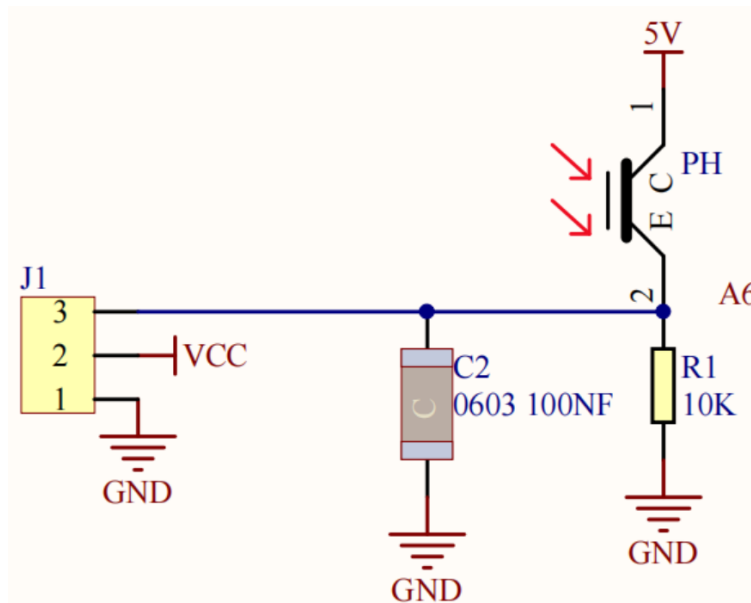


Figura 23: Schema caracteristică LDR și a conexiunii divizorului de tensiune (intrare analogică ESP32) (Sursa: <https://docs.keyestudio.com>)

Intrarea analogică:

- nu dă o valoare MARE sau MARE,
- ci un rezultat al măsurării într-un interval numeric (de exemplu, 0–4095),
- care este proporțională cu intensitatea luminii.

Studentii care participă la proiect se confruntă cu faptul că valoarea măsurată nu este absolută, ci dependentă de context, precizia senzorului și a componentelor electronice este limitată, iar interpretarea datelor măsurate necesită întotdeauna context.

Procesul decizional și intervenția – procesare analogică

Cea mai simplă structură logică a unui sistem controlat de lumină:

1. măsurarea intensității luminoase (intrare analogică),
2. comparație cu o valoare limită,
3. efectuarea unei intervenții (aprinderea sau stingerea iluminatului).

Acest model produce în continuare o decizie digitală bazată pe o intrare analogică.

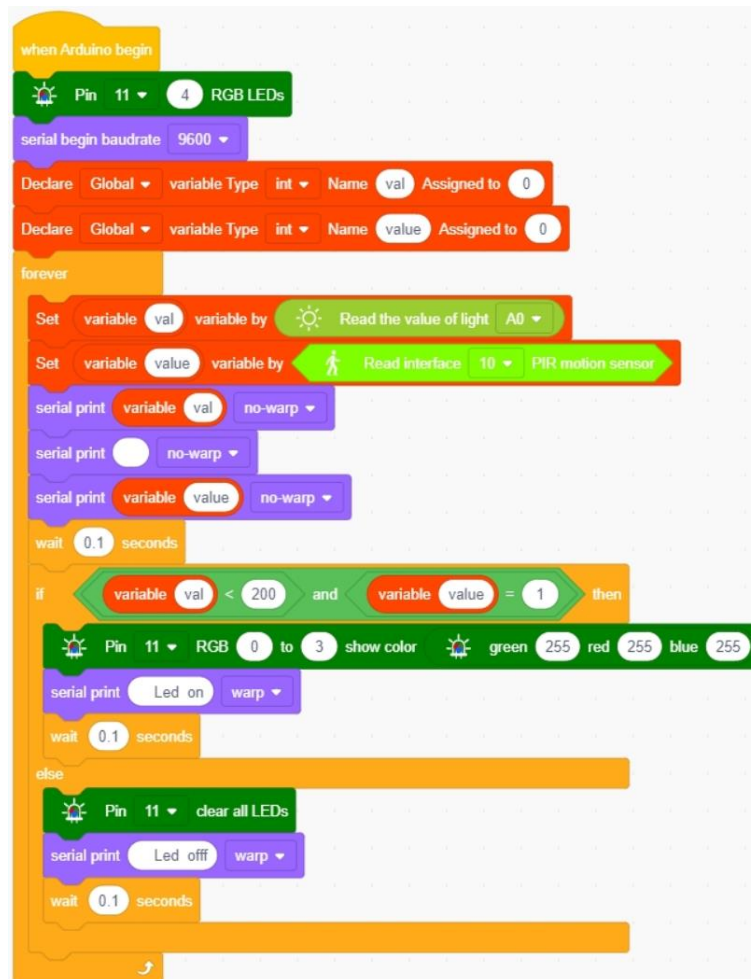


Figura 24: Structura blocului de control condiționat pentru intrarea analogică (Sursa: <https://docs.keyestudio.com>)

Semnificația educațională a tranziției analogice și digitale

Această etapă pune accent pe digitalizarea semnalului analogic, problema alegerii valorii limită și gestionarea datelor de măsurare zgomotoase și fluctuante.

Studentii vor experimenta faptul că o limită prea mică sau prea mare poate cauza o funcționare instabilă, așadar pot fi necesare histerezis sau filtrare, stabilind astfel că deciziile ingineresti afectează comportamentul sistemului.

Ieșire analogică – reglare a intensității luminii prin PWM

La nivelul următor al proiectului, intervenția nu mai este binară, ci continuă. Luminozitatea LED-ului nu este doar aprinsă și stinsă, ci este reglată proporțional.

PWM (Pulse Width Modulation - Modulația lății impulsurilor) este o tehnică ce ne permite să obținem un efect analogic cu o ieșire digitală:

- ieșirea se activează și se dezactivează rapid,
- proporția de timp petrecut activ (ciclul de funcționare) determină performanța medie,
- Luminozitatea LED-ului este proporțională cu factorul de umplere.

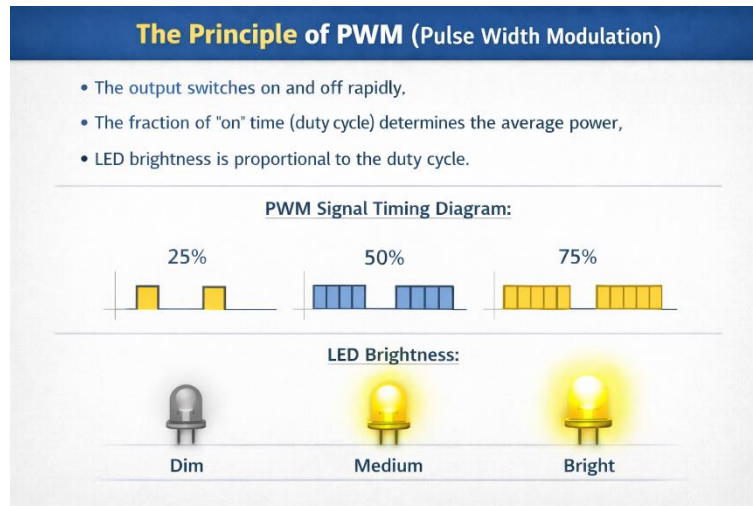


Figura 25: Diagrama de temporizare a semnalului PWM și modificarea luminozității LED-ului (Sursa: generată de inteligența artificială)

În acest proiect, elevii vor crea o relație proporțională:

- intrare: intensitatea luminii măsurată (valoare analogică),
- procesare: scalare, transformare,
- ieșire: luminozitatea LED-ului controlată prin semnal PWM.

Această abordare reflectă deja o perspectivă de control al proceselor și este un precursor direct al reglementărilor industriale.

Reflecție educațională

Proiectul sistemului controlat de lumină joacă un rol cheie în dezvoltarea gândirii elevilor:

- distinge între mărimi digitale și mărimi analogice,
- să înțeleagă că lumea reală este continuă, iar controlurile sunt discrete,
- recunosc principiul reglementării proporționale,
- Ei experimentează că o funcționare „mai bună” necesită mai multă planificare.

Această parte formează o punte naturală către sistemele ulterioare de control al temperaturii, control al motoarelor și optimizare a energiei și reprezintă un pas semnificativ în aprofundarea gândirii ingineresti.

3.17. Proiectul 3 - Detecție a distanței și intervenție automată (Sistem inteligent de hrănire)

Sistemul inteligent de alimentare este un proiect integrat în care studenții combină un senzor bazat pe timp și un actuator mecanic într-o automatizare funcțională. Acesta modelează o situație simplă, dar reală: prin detectarea proximității unui obiect, se efectuează o acțiune mecanică.

Aceasta acoperă complet lanțul operațional de bază al sistemelor automatizate:



Scopul funcțional al Sistemului Inteligent de Hrănire este acela că sistemul implementează operarea unui alimentator automat care detectează apropierea unui obiect (de exemplu, o mână), apoi ia o decizie pe baza distanței măsurate și efectuează o mișcare mecanică (de exemplu, deschiderea/închiderea unui capac) folosind un servomotor.

Funcția este simplă de înțeles, dar în același timp necesită o gândire tehnică și inginerescă complexă.

Senzor de distanță cu ultrasunete

Sarcina sa este de a determina distanța până la un obiect folosind măsurarea timpului. Funcționarea senzorului se bazează pe control digital (declanșator/ecou), iar valoarea distanței este calculată.

Semnificație educațională:

- învățarea unui nou principiu de măsurare,
- separarea datelor brute de măsurare de informațiile interpretate.

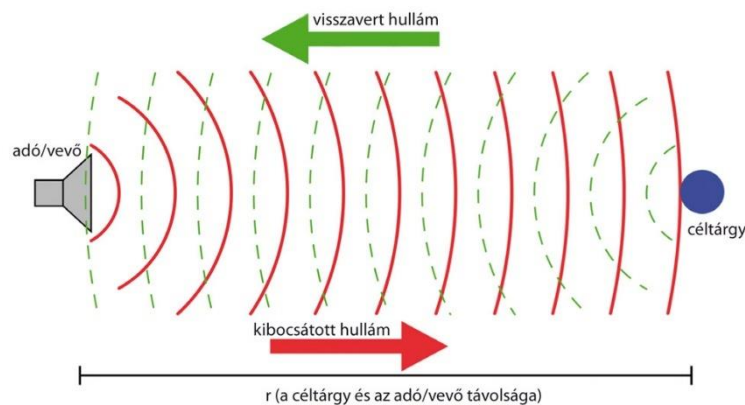


Figura 26: Diagramă schematică a măsurării distanței cu ultrasunete (Sursa: <https://docs.keyestudio.com>)

Servomotor ca actuator mecanic

Servomotorul este decizia de control fizic pentru implementarea sa răspuns.

Poate fi controlat cu un semnal PWM, ajustat la poziții unghiulare specifice și oferă o mișcare precisă și repetabilă.

Semnificație educațională:

- conectarea electronicii și mecanicii,
- experimentând efectul fizic direct al controlului.

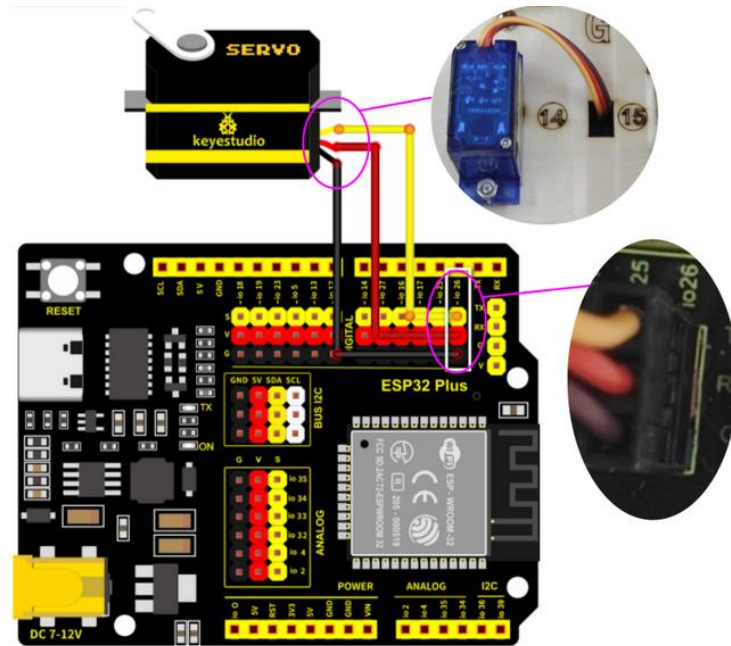


Figura 27: Structura și principiul de control al servomotorului (Sursa: <https://docs.keyestudio.com>)

Principiul măsurării distanței cu ultrasunete

Procesul de măsurare cu ultrasunete:

1. • senzorul emite un impuls sonor
2. • sunetul se reflectă de obiect
3. • controlerul măsoară timpul scurs
4. • distanța poate fi determinată prin calcul

Aceasta nu este o măsurare analogică a tensiunii, ci o măsurare bazată pe timp, bazată pe semnale digitale, care necesită un pas de calcul.

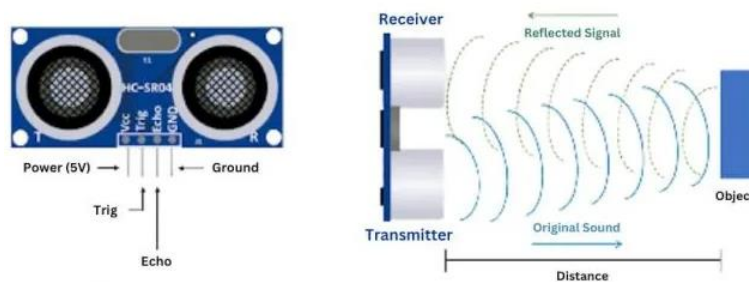


Figura 28: Procesul de măsurare cu ultrasunete (Sursa: <https://docs.keyestudio.com>)

Pentru a ajuta procesul de procesare a datelor senzorilor și logica decizională, distanța măsurată formează baza deciziei de control. Procesul logic:

1. măsurarea distanței
2. comparație cu valoarea limită
3. luarea unei decizii
4. inițierea unei intervenții

Această structură descrie deja un proces complet automatizat.

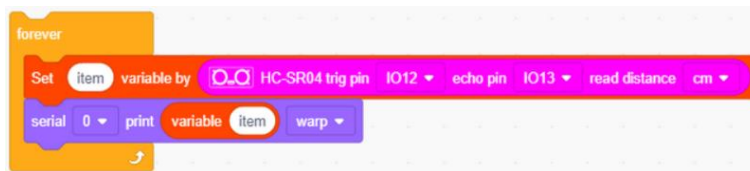


Figura 29: Prelucrarea datelor de distanță pe blocuri (Sursa: <https://docs.keyestudio.com>)

Controlul servomotorului

Servomotorul este controlat cu un semnal PWM, unde lățimea impulsului determină poziția unghiulară, iar controlul nu controlează puterea, ci poziția.

Aceasta este o diferență semnificativă în comparație cu reglarea luminozității PWM a LED-urilor.



Figura 30: Relația dintre semnalul PWM și poziția servomotorului (Sursa: <https://docs.keyestudio.com>)

Sistem inteligent de alimentare - funcționare integrată

Funcționarea Sistemului de Alimentare Inteligentă formează un întreg logic unificat: dacă distanța măsurată se încadrează într-o valoare prag, sistemul ia o decizie, servomotorul se mișcă și are loc o operație mecanică.

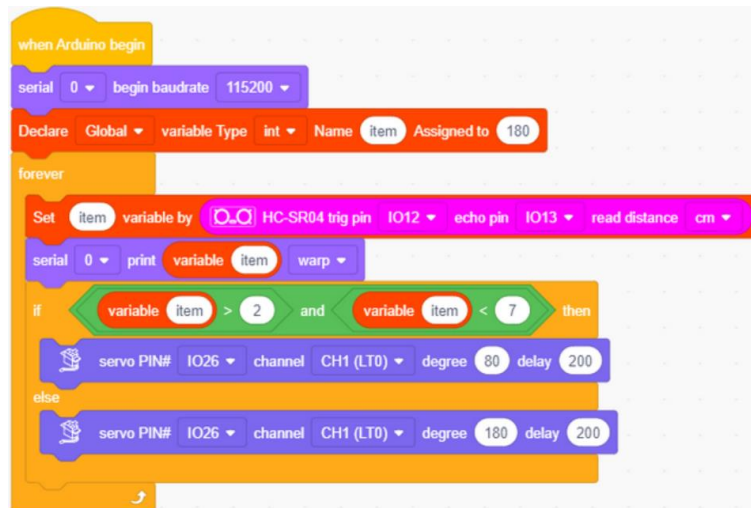


Figura 31: Program complet în blocuri al sistemului Smart Feeding (Sursa: <https://docs.keyestudio.com>)

Reflecție educațională

Proiectul Sistemul inteligent de hrănire este potrivit în special pentru elevi care:

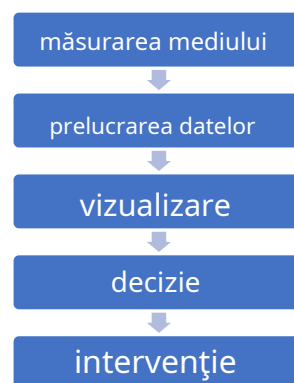
- a vedea funcționarea controlului la nivel de sistem,
- înțelegerea specificului măsurării bazate pe timp,
- recunoaște relația dintre logica decizională și efectul mecanic,
- dezvoltă o gândire inginerescă conștientă.

Proiectul arată clar că funcționarea sistemelor automatizate nu constă în elementele individuale, ci în interconectarea lor și în logica proiectată.

3.18. Proiectul 4 – Sistem de control al temperaturii

Sistemul de control al temperaturii modelează o sarcină clasică de control, fundamentală în automatizarea industrială și în sistemele de management al clădirilor. Proiectul se concentrează pe măsurarea continuă a unei mărimi de mediu, afișarea acesteia și apoi controlul automat al unui dispozitiv de intervenție.

Întregul lanț operațional al proiectului:



Această structură este ușor de înțeles pentru studenți, reflectând în același timp logica reală de reglementare industrială.

Senzor de temperatură și umiditate (senzor DHT)

Senzorul DHT este un senzor digital de mediu care măsoară temperatura și umiditatea relativă și transmite datele către controler sub forma unui pachet de date digitale.

Caracteristici electronice:

- comunică printr-o singură linie de date,
- rezultatul măsurării nu este o tensiune analogică, ci o valoare digitală procesată,
- Senzorul folosește un mecanism intern de temporizare și control.

Semnificație educațională:

- separă măsurarea analogică de comunicarea de date digitale,
- subliniază că unii dintre senzori furnizează deja date „procesate”,
- introduce conceptul de eșantionare și timp de actualizare.

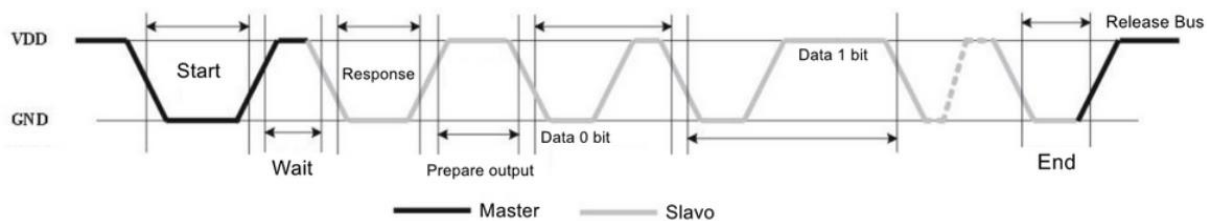


Figura 32: Principiul de funcționare al senzorului DHT și procesul de transmitere a datelor (Sursa: <https://docs.keyestudio.com>)

Afișaj LCD 1602 – afișarea datelor de măsurare

Un element important al sistemului de control al temperaturii este feedback-ul utilizatorului. Acest rol este îndeplinit de afișajul LCD 1602, care afișează datele măsurate sub formă numerică și permite monitorizarea continuă a stării actuale a sistemului.

Rol funcțional:

- afișarea temperaturii și umidității,
- feedback privind starea de funcționare (de exemplu, ventilator activ/inactiv),
- afișarea informațiilor de diagnosticare.

Semnificație educațională:

- accentuează separarea dintre măsurare și afișare,
- demonstrează că funcționarea sistemului nu este „invizibilă”,
- susține depanarea și înțelegerea sistemului.

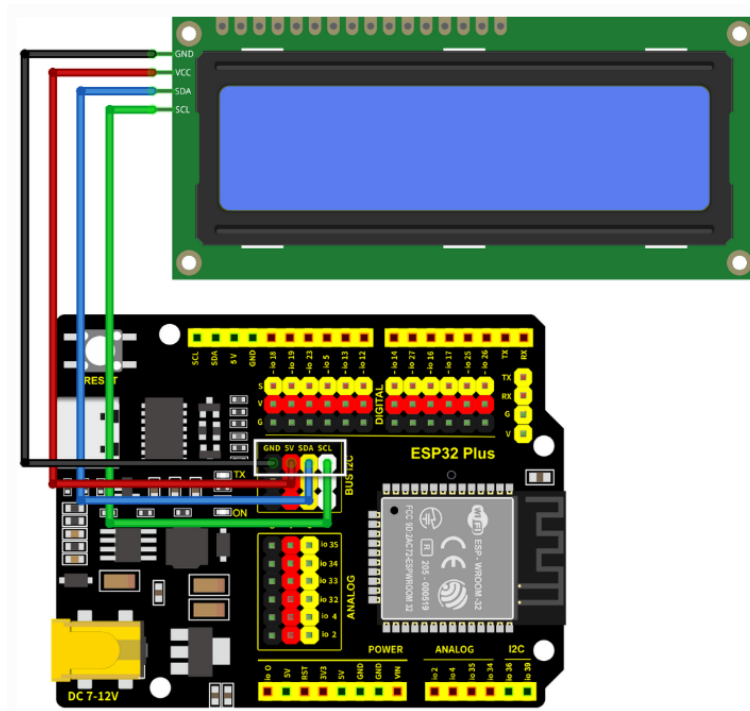


Figura 33: Schema de conectare LCD 1602 (Sursa: <https://docs.keyestudio.com>)

Ventilatorul ca element de intervenție

Ventilatorul este un actuator activ în sistem, care este utilizat pentru a modifica condițiile de mediu. Controlerul comută ventilatorul direct sau prin intermediul unui driver printr-o ieșire digitală.

Logică de control:

- dacă temperatura măsurată depășește o valoare limită setată → ventilatorul pornește,
- Dacă temperatura scade sub valoarea limită → ventilatorul se oprește.

Semnificație educațională:

- demonstrează un control simplu în două stări,
- înțelege avantajele și limitele controlului bazat pe limite,
- pregătește pentru înțelegerea ulterioară a histerezisului și a controlului PID.

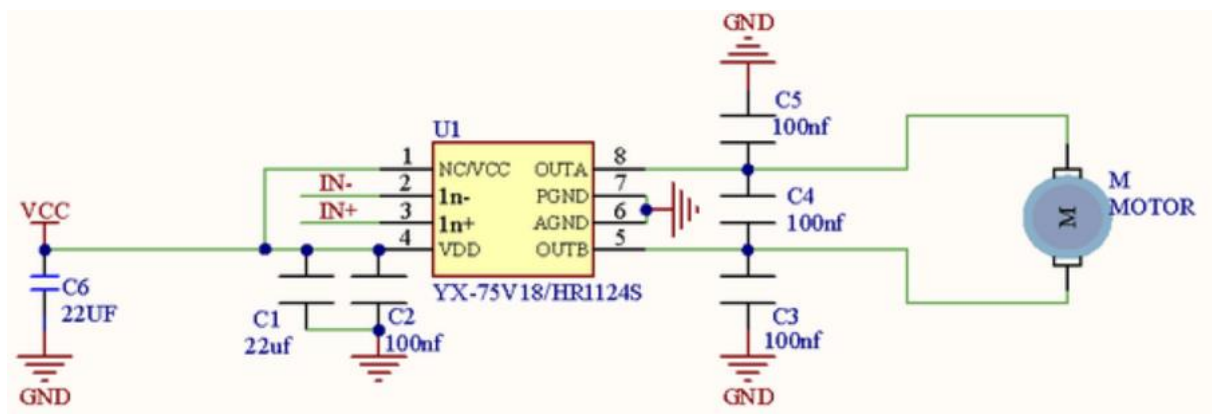


Figura 34: Schema de conectare a comenzii ventilatorului (Sursa: <https://docs.keyestudio.com>)



Figura 35: Diagrama blocului logic de control al ventilatorului (Sursa: <https://docs.keystudio.com>)

Funcționarea integrată a sistemului de control al temperaturii

Întregul proiect al Sistemului de Control al Temperaturii funcționează conform următoarei logici:

1. • senzorul DHT efectuează o măsurare
2. • datele sunt prelucrate
3. • valoarea măsurată este afișată pe ecranul LCD
4. • regulatorul compară valoarea cu valoarea limită
5. • starea ventilatorului se schimbă în mod corespunzător

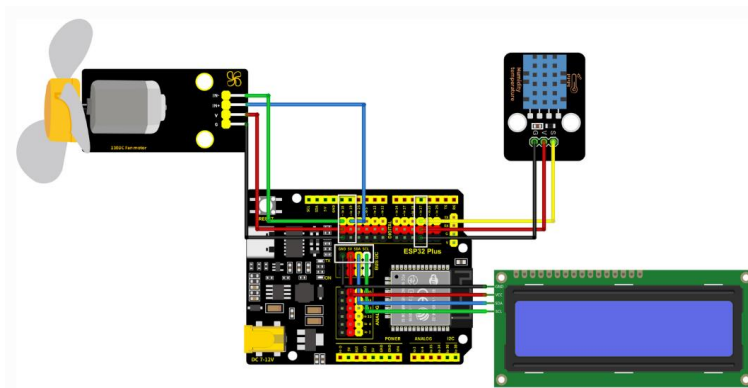


Figura 36: Schema de conectare a sistemului de control al temperaturii (Sursa: <https://docs.keystudio.com>)

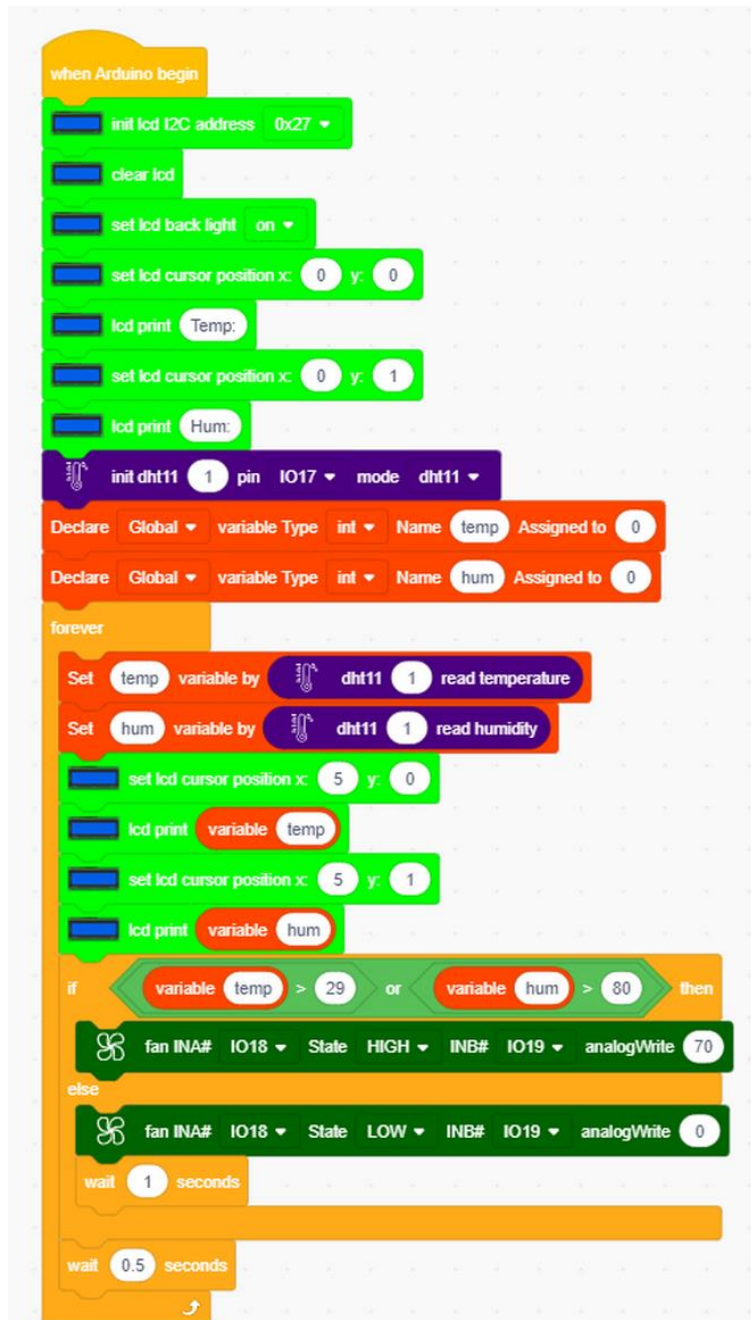


Figura 37: Program bloc complet al unui sistem de control al temperaturii (Sursa: <https://docs.keyestudio.com>)

Această structură este deja un model simplificat al unui sistem de reglementare real.

Reflecție educațională

Proiectul Sistem de Control al Temperaturii oferă studenților oportunitatea de a:

- înțelegerea rolului datelor de mediu în automatizare,
- distinge între măsurare, vizualizare și intervenție,
- recunoaște funcționarea și limitele reglementării bazate pe limite,
- gândește-te sistematic la o sarcină aparent simplă.

Acest lucru se potrivește bine cu conceptul de casă inteligentă și fermă inteligentă și creează o bază solidă pentru înțelegerea ulterioară a unor soluții de control și de rețea mai complexe.

3.19. Proiectul 5 – Sistem automat de irigații

Proiectul Sistemului de Auto-Irigații evidențiază mai întâi problema inginerescă legată de capacitatea electrică a unității de control, care este insuficientă pentru a acționa direct actuatorul. Scopul sarcinii este de a crea un model al unui sistem de irigații automatizat, în care se realizează o interfață sigură, de tip industrial, între luarea deciziilor și execuția fizică.

Lanțul operațional al proiectului:



Această structură este fundamentală pentru toate sistemele reale de automatizare

industrială. **Limitările ieșirilor microcontrolerului**

Controlerul Smart Farm Kit este un sistem bazat pe microcontroler ale cărui ieșiri digitale funcționează la niveluri de tensiune scăzută (de obicei 3,3 V sau 5 V CC) și pot furniza doar curenți mici (de ordinul câtorva zeci de mA).

Acest lucru este suficient pentru controlul LED-urilor, circuitelor electronice mai mici sau semnalelor logice.

Totuși, nu este suficient pentru acționarea directă a motoarelor, pompelor, valvelor sau dispozitivelor de alimentare cu tensiune de rețea (de exemplu, 230 V CA).

Dintr-o perspectivă educațională, acesta este un punct cheie: elevii recunosc că controlul și performanța nu sunt același concept.

Niveluri de tensiune și diferențe de putere

Irigarea automată este un bun exemplu de coexistență a mai multor niveluri de tensiune într-un sistem:

- partea de control: joasă tensiune (5 V CC),
- partea actuatorului: tensiune sau putere mai mare (de exemplu, 12 V CC, 24 V CC, până la 230 V CA).

În timpul proiectului, accentul se pune pe motivul pentru care un motor nu poate fi conectat direct la ieșirea microcontrolerului, de ce este necesar un element de adaptare a puterii și cum să se separe în siguranță cele două lumi.

Modul releu ca element de adaptare a puterii

Elementul central al sistemului de autoirigare este modulul de releu, care permite unui semnal de putere redusă de la microcontroler să controleze un circuit de putere mai mare.

Pe baza principiului de funcționare al releului, controlerul comută o bobină, câmpul magnetic al bobinei închide sau deschide mecanic contactele, iar contactele funcționează într-un circuit separat.

Aceasta implementează izolarea galvanică, care protejează electronica de control, crește fiabilitatea operațională a sistemului și respectă principiile automatizării industriale.

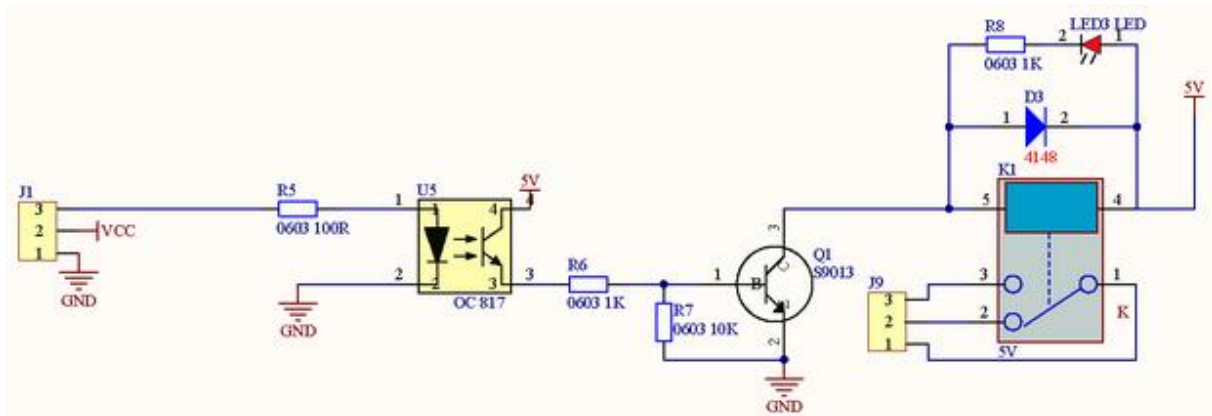


Figura 38: Contactele modulului releu și logica de funcționare (Sursa: <https://docs.keyestudio.com>)

Soluții industriale de ieșire PLC – Paralel

Utilizarea releelor nu este o „soluție de hobby”, ci un principiu industrial. Ieșirile controlerelor logice programabile (PLC-uri) sunt, de obicei, și:

- ieșiri de releu,
- ieșiri pe tranzistoare (PNP/NPN),
- mai rar ieșiri triac pentru curent alternativ.

Avantajele ieșirilor cu releu PLC:

- gamă largă de tensiune,
- Comutarea sarcinilor de curent alternativ și continuu,
- izolare electrică.

Proiectul Smart Farm modelează o mentalitate industrială în acest moment, chiar dacă instrumentul folosit este în scopuri educaționale.

Logica de control a sistemului de auto-irigare

Funcționarea sistemului este simplă, dar realizează automatizare completă:

1. controlerul evaluează condițiile de intrare
2. le compară cu o valoare limită,
3. decide asupra necesității irigațiilor
4. activează releul prin ieșirea digitală
5. releul comută unitatea de irigare


```
when Arduino begin
  init lcd I2C address 0x27
  clear lcd
  set lcd back light on
  Declare Global variable Type int Name waterlevel Assigned to 0
  Declare Global variable Type int Name soilhum Assigned to 0
  forever
    Set waterlevel variable by read the value of water level IO33
    Set soilhum variable by read the value of soil moisture IO32
    clear lcd
    set lcd cursor position x: 0 y: 0
    lcd print WaterLevel:
    set lcd cursor position x: 0 y: 1
    lcd print SoilHum:
    set lcd cursor position x: 11 y: 0
    lcd print variable waterlevel
    set lcd cursor position x: 11 y: 1
    lcd print variable soilhum
    if 700 > variable waterlevel then
      Tone PIN# IO16 frequency NOTE_E5 duration 131
      Tone PIN# IO16 frequency NOTE_A5 duration 131
      Tone PIN# IO16 frequency NOTE_B5 duration 131
      NoTone IO16
    if 1200 > variable soilhum then
      Tone PIN# IO16 frequency NOTE_D3 duration 131
      Tone PIN# IO16 frequency NOTE_F3 duration 131
      Tone PIN# IO16 frequency NOTE_A3 duration 131
      NoTone IO16
    if variable soilhum < 1200 and variable waterlevel > 700 then
      relay pin IO25 output HIGH
      wait 0.4 seconds
      relay pin IO25 output LOW
      wait 0.7 seconds
    wait 0.5 seconds
```

Figura 40: Control bazat pe blocuri al sistemului de irigații automate (Sursa: <https://docs.keystudio.com>)

Această structură poate fi mapată direct la ciclurile de control al proceselor industriale.

Lecții educaționale și de inginerie

În timpul proiectului de irigare automată, elevii vor:

- înțelegerea limitelor fizice ale microcontrolerelor,
- recunoaște nevoia de aliniere a performanței,
- întâlnește conceptul de izolare galvanică,
- Aceștia utilizează soluții analoage principiilor automatizării industriale.

Proiectul este deosebit de puternic în ceea ce privește creșterea gradului de conștientizare cu privire la consecințele fizice ale deciziilor de control, pregătirea pentru gândirea bazată pe PLC și punerea bazelor pentru înțelegerea sistemelor ulterioare care sunt conectate în rețea și utilizează comunicarea industrială.

Acest capitol formează o tranziție naturală între lumea automatizării IoT simple și sistemele profesionale de control industrial.

3.20. Proiectul 6 – Sistem Smart Farm controlat prin WiFi

Proiectul Smart Farm, controlat prin WiFi, reprezintă tranziția de la automatizarea operațională locală la sisteme conectate în rețea și monitorizate de la distanță. În această etapă, studenții nu mai controlează doar senzorii și actuatorii, ci implementează un model complet de monitorizare bazat pe IoT.

Ideea centrală a proiectului este ca sistemul:

- se conectează la rețea,
- furnizează date unui client la distanță,
- Poate fi controlat prin interfața web HMI.

Această funcționalitate depășește limitele simplelor sisteme integrate și vă aduce mai aproape de soluții industriale și pentru dispozitive inteligente reale.

Rolul conexiunii WiFi în sistemul IoT

Controlerul Smart Farm Kit ESP32 are un modul WiFi încorporat, care permite conectarea directă la o rețea wireless. Conexiunea WiFi în acest proiect nu este doar o adăugare tehnică, ci o schimbare sistemică de perspectivă: controlul și monitorizarea părăsesc mediul local, cablat, și intră într-un context de rețea.

Prin conectarea la rețea, sistemul nu funcționează ca un controler izolat, ci datele senzorilor pot fi accesate de la distanță prin intermediul unui browser, deoarece intervențiile nu sunt legate de input fizic direct, ci de solicitări din rețea.

Dintr-o perspectivă educațională, aici devine tangibil conceptul de Internet of Things (IoT): dispozitivul nu există în sine, ci este un nod activ într-un sistem de rețea.

În timpul stabilirii unei conexiuni WiFi, elevii vor fi introduși în conceptele de bază ale rețelelor, cum ar fi punctul de acces. ESP32 se conectează la o rețea WiFi existentă (de exemplu, un router școlar sau de acasă) care acționează ca punct de acces, astfel încât să înțeleagă că controlerul se conectează ca și client, iar rețeaua oferă mediul de comunicare.

Un ESP32 conectat la rețea primește o adresă IP unică care identifică dispozitivul din rețea, permite accesul dintr-un browser și reprezintă baza comunicării ulterioare cu sistemul de gestionare a clienților.

Aceasta este prima dată când studenții experimentează faptul că un dispozitiv fizic are o „adresă” în rețea, similar cu alte computere sau sisteme de supraveghere.

În timpul proiectului, devine clar că ESP32, computerul sau dispozitivul mobil al utilizatorului se află în aceeași rețea, ceea ce înseamnă că comunicarea este posibilă numai dacă aceste dispozitive „se văd reciproc” la nivel de rețea. Controlul și recuperarea datelor au loc sub formă de mesaje de rețea.

Semnificație educațională

În timpul introducerii conexiunii WiFi, elevii:

- stăpânesc gândirea de bază în rețea,
- înțelegerea rolului accesului bazat pe IP în sistemele moderne,
- recunoașteți diferența dintre controlul local și cel de rețea,
- experimentați una dintre condițiile de funcționare de bază ale sistemelor IoT.

Aceste cunoștințe te pregătesc în mod direct pentru:

- înțelegerea interfeței de administrare web,
- procesarea arhitecturilor ulterioare ale sistemului de supraveghere a clienților,
- și adoptarea unei abordări de rețea pentru sistemele industriale și distribuite.

Funcționarea sistemului de monitorizare web încorporat

În soluția Smart Farm din fabrică, interfața de gestionare web este implementată folosind un sistem de gestionare web care rulează direct pe ESP32. Aceasta înseamnă că controlerul însuși furnizează site-ul web, deci nu există un sistem de gestionare extern sau un computer în sistem, ci clientul (browserul) se conectează direct la ESP32.

Această soluție este extrem de intuitivă din punct de vedere educațional, deoarece apare într-un singur instrument:

- colectarea datelor,
- prelucrarea,
- afișajul,
- controlul.

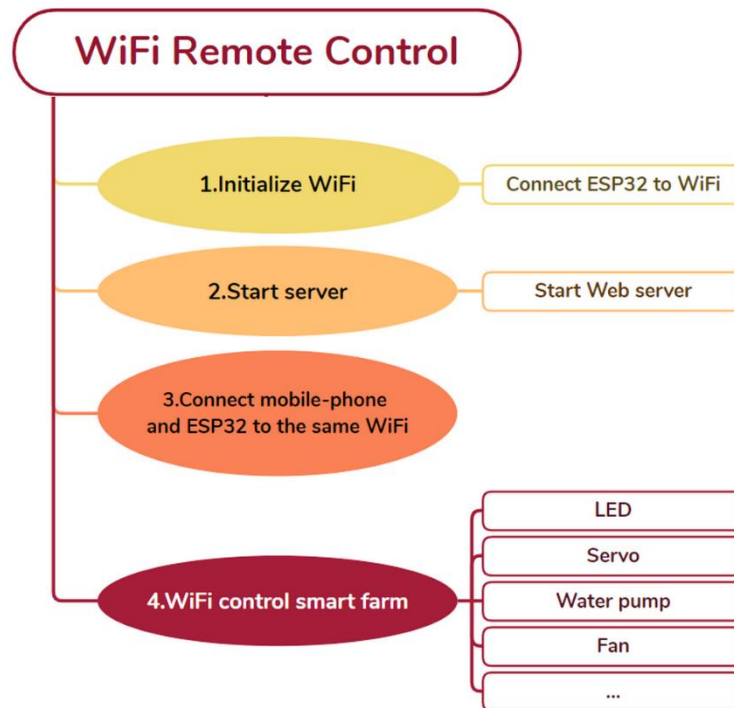


Figura 41: Model de comunicare al unui sistem de monitorizare web care rulează pe ESP32 (Sursa: <https://docs.keystudio.com>)

Pagina de administrare web este „fereastra” sistemului către utilizator. Conform tutorialului din fabrică, interfața implementează de obicei următoarele funcții:

- afișarea datelor senzorului de curent,
- feedback privind starea ieșirii,
- oferind comenzi simple (butoane, comutatoare).

Interfața nu este puternică prin complexitatea sa grafică, ci prin capacitatea sa de a reflecta starea sistemului fizic în timp real.

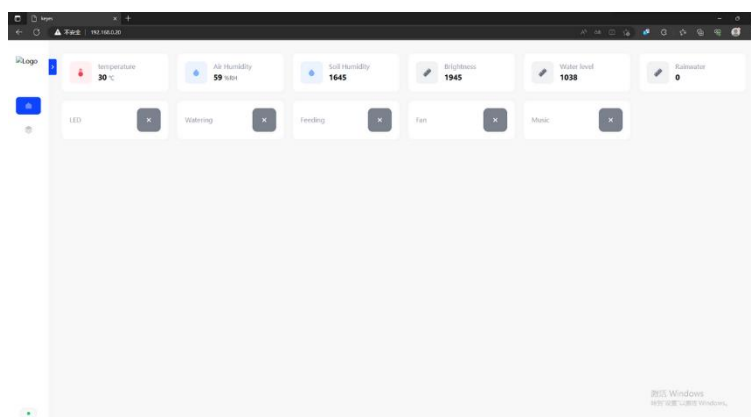


Figura 42: Aspectul paginii web de administrare Smart Farm (Sursa: <https://docs.keystudio.com>)

Un element educațional important este recunoașterea faptului că un sistem de monitorizare web nu funcționează cu resurse nelimitate. În cazul ESP32, studenții se confruntă cu următorii factori:

- memorie limitată,
- spațiu de stocare limitat,
- gestionarea unui număr limitat de conexiuni simultane,

- folosind structuri HTML simple.

Acest lucru duce la o funcționalitate a site-ului web deliberat simplă, astfel încât cantitatea de date afișate este limitată, ceea ce înseamnă că ESP32 nu este potrivit pentru rularea de aplicații web complexe.

Această realizare este o lecție inginerescă cheie: toate sistemele sunt construite în jurul unor compromisuri.

Elementele de control afișate pe interfața web HMI permit utilizatorului să trimită o comandă către sistem, controlerul interpretează aceasta ca o instrucțiune digitală, iar actuatoarele răspund la semnalul venit de la rețea.

Această operațiune demonstrează principiul de bază al sistemelor IoT moderne, unde utilizatorul nu este prezent fizic, dar controlul este imediat și bazat pe feedback.

În timpul proiectului Smart Farm controlat prin WiFi, elevii vor:

- înțelege esența controlului bazat pe rețea,
- recunoașterea limitelor de resurse ale sistemelor integrate,
- experimentați elementele de bază ale modelului de sistem de supraveghere client,
- acestea separă sarcinile de control și cele de afișare.

Punctul forte al proiectului constă în faptul că:

- oferă o experiență IoT reală cu cerințe minime pentru dispozitive,
- evidențiază clar limitele soluțiilor din fabrică,
- Oferă o pregătire naturală pentru înțelegerea arhitecturilor industriale ulterioare, externe, sistematice.

Acest capitol încheie cartografierea posibilităților sistemului Smart Farm al fabricii și pune bazele dezvoltării propriei arhitecturi de sistem, separate prin resurse și urmând un model industrial.

3.21. Interpretare pedagogică sumară

Partea de proiect Smart Farm, bazată pe programare pe blocuri, nu apare ca un obiectiv de învățare independent, ci ca o tranziție pedagogică planificată în mod conștient pe parcursul proiectului. În această fază, elevii dobândesc concepte de bază, relații și principii de funcționare care sunt ulterior esențiale pentru înțelegerea și dezvoltarea unor sisteme mai complexe, de tip industrial.

Pentru a promova gândirea sistemică, elevii lucrează în mod constant cu același model de bază pe tot parcursul proiectului:



Acest lanț de influență apare în fiecare parte a proiectului:

- intrări digitale (de exemplu, buton, comutator),
- intrări analogice (de exemplu, LDR, potențiometru),
- măsurători bazate pe timp (senzor cu ultrasunete),
- ieșiri digitale și bazate pe PWM (LED, ventilator, servomotor),
- comutare de putere cu releu (sistem de irigații),
- comunicare în rețea și control web (WiFi).

Elevii își dau seama treptat că elementele individuale nu pot fi interpretate izolat, ci sunt părți funcționale ale unui sistem mai amplu.

Conexiunea dintre lumea analogică și cea digitală

Un rezultat pedagogic proeminent este că elevii pot face o distincție clară între:

- semnale digitale (bistare, conduse de evenimente),
- mărimi analogice (valori continue),
- rolul conversiei analog-digitale,
- între valorile limită, controlul proporțional și modulația PWM.

Conceptul de zgomot de măsurare, importanța alegerii pragului de decizie și abordarea „măsurare ≠ informație” apar în timpul controlului luminii, controlului temperaturii și controlului ventilatorului.

Datorită înțelegerii intervenției și gestionării energiei, atunci când utilizează modulul de releu și actuatorii motorizate, studenții se confruntă cu faptul că ieșirile microcontrolerelor sunt capabile de o livrare limitată de curent și putere, prin urmare, controlul și transmisia puterii sunt separate, astfel încât semnalele logice de joasă tensiune (3,3–5 V CC) pot controla circuite de putere mai mare (de exemplu, 12 V CC, 230 V CA).

Aceasta creează o paralelă directă cu funcționarea controalelor industriale, unde PLC-urile controlează actuatorii externe prin ieșiri de releu, tranzistor sau semiconductori.

Gândirea în rețea și perspectiva IoT

În cadrul proiectului bazat pe WiFi, studenții vor experimenta:

- elementele de bază ale modelului sistemului de supraveghere client-client,
- funcționarea sistemului de monitorizare web încorporat,
- impactul constrângerilor de resurse fizice (memorie, stocare) asupra proiectării sistemului,
- posibilitățile și limitele monitorizării și controlului de la distanță.

Această experiență ne pregătește să recunoaștem că într-un mediu industrial, sarcinile sunt adesea separate (controlerul nodului de teren – monitorizarea centrală – serviciile backend).

Programarea bazată pe blocuri ajută la pregătirea pentru o schimbare în abordarea programării, care în acest context:

- decuplează logica de control de dificultățile sintactice,
- vizualizați gestionarea stării, evenimentele și condițiile,
- poate fi ușor adaptat la paradigmele de programare industrială (de exemplu, IEC 61131-3, blocuri funcționale, mașini de stare).

Studentii nu „învață Scratch”, ci mai degrabă învață gândirea bazată pe control, care poate fi aplicată ulterior în programarea microcontrolerelor bazate pe C, PLC-urile industriale, comunicarea bazată pe Modbus și arhitecturile sistemelor distribuite.

Rezumat

Această fază a proiectului creează o bază stabilă pentru dezvoltarea de elemente hardware proiectate de către dezvoltatori, tranziția la programarea bazată pe text, înțelegerea arhitecturilor distribuite, orientate spre industrie și dezvoltarea ulterioară a sistemelor cu o structură de monitorizare client-câmp nod, backend-frontend.

Prin urmare, mediul bazat pe blocuri nu este o stație terminală, ci o scară de învățare construită în mod conștient, care îi conduce treptat pe studenți către abordarea unor probleme reale de inginerie și industrie.

3.22. Implementarea practică a capitolului

Asamblarea și punerea în funcțiune a kitului Smart Farm a fost prima experiență cu adevărat „practică” pentru studenți. Observarea kitului din fabrică în funcțiune le-a dat un sentiment rapid de succes, dar a devenit curând clar că în spatele automatizărilor aparent simple se ascund conexiuni logice și electronice complexe.

Prelucrarea documentației fabricii, în special interpretarea descrierilor în limbi străine, a reprezentat o provocare serioasă. În timpul traducerii parțiale în limba maghiară și al adaptării educaționale, studenții s-au implicat activ în interpretarea conținutului, ceea ce le-a aprofundat semnificativ înțelegerea.

Programarea bazată pe blocuri părea ușoară la început, dar gestionarea stărilor și controlul bazat pe condiții au necesitat o gândire mai serioasă.

4. Proiectare digitală a propriului model de casă inteligentă / fermă inteligentă (Fusion 360)

Acest capitol reprezintă o piatră de hotar definitivă a proiectului: aici se află locul în care logica senzorilor, a controlului și a IoT învățată anterior devine o structură fizică concretă, tangibilă. În această etapă, studenții trec de la lumea electronicii și a gândirii software la domeniul proiectării inginerești, unde fiecare decizie are consecințe tehnologice geometrice, de asamblare și de fabricație.

Scopul acestui capitol nu este pur și simplu de a crea un model 3D, ci de a înțelege că un model fizic destinat imprimării 3D este o structură de susținere: trebuie să ofere spațiu pentru controler, senzori, actuatori și cablare, suportând în același timp asamblarea, activitățile de măsurare și modificările ulterioare. Modelul este, în acest sens, un instrument de învățare, nu un produs final.

Parcursul de învățare este structurat în mod deliberat în mai mulți pași. Designul digital este precedat de o fază de concept fizic, care analizează designul fabricii Kitului Smart Farm.

Se bazează pe identificarea elementelor funcționale și recunoașterea limitelor fabricației aditive. Aceasta este urmată de învățarea despre mediul de proiectare Fusion 360, apoi de dezvoltarea unui model digital parametric, bazat pe componente, până la o stare pregătită pentru producție.

Pe parcursul procesului, elevii experimentează:

- Un design bun nu începe cu software,
- Forma este întotdeauna o consecință a funcției,
- gândirea parametrică permite iterația și reproiectarea,
- Modelul digital acționează ca un „prototip virtual” pentru a ajuta la prevenirea problemelor de fabricație și asamblare.

Această parte formează astfel o punte între înțelegerea sistemelor electronice-informaticе și implementarea fizică și oferă o bază solidă pentru etapele ulterioare ale proiectului, care se ocupă de imprimarea 3D și integrarea sistemelor.

4.1. Punct de plecare: analiza designului fizic al Kitului Fermă Inteligentă

Primul pas în procesul de proiectare digitală nu este utilizarea software-ului CAD, ci analizarea conștientă a soluțiilor fizice existente. Kitul Smart Farm servește drept sistem de referință în acest proiect: un punct de plecare fără de care este imposibil să creezi un model propriu funcțional și optimizat din punct de vedere educațional.

Designul fizic al Kitului Smart Farm nu este prezentat ca un model de copiat, ci ca un exemplu de analizat. Scopul elevilor este de a explora:

- care elemente îndeplinesc o funcție reală,
- modul în care componentele electronice sunt conectate la spațiul fizic,
- care soluții servesc scopuri educaționale și care au fost dezvoltate din motive tehnologice de producție sau din considerente estetice.

Această abordare stabilește principiul de bază al designului bazat pe funcții în această etapă timpurie: forma este o consecință a funcției, nu punctul său de plecare. Studenții învață să examineze critic soluțiile predefinite și să distingă elementele necesare de detaliile contextuale sau opționale.

Baza conceptualizării este cunoașterea dimensiunilor reale. Elevii examinează elementele fabricii cu instrumente simple, accesibile tuturor. Folosind un șubler, determină dimensiunile din jur, apoi fac schițe proporționale pe hârtie milimetrică și înregistrează locațiile deschiderilor, punctele de montare și pozițiile senzorilor.

Din punct de vedere pedagogic, este important ca în această fază să nu se producă desene tehnice standard, deoarece obiectivul principal nu este acuratețea formală, ci mai degrabă înțelegerea relațiilor spațiale și a proporțiilor.



Figura 43: Măsurarea elementelor fizice ale Fermei Inteligente (Sursa: generat de inteligența artificială)

Acest pas pregătește direct pentru gândirea parametrică ulterioară, consolidând în același timp legătura dintre realitatea fizică și designul abstract.

4.2. Limitări ale imprimării 3D și implicații de design

În timp ce analizează componentele fabricii, studenții se vor confrunța cu limitele practice ale fabricației aditive. Vor recunoaște că unele detalii geometrice nu pot fi reproduse cu imprimarea 3D desktop, deoarece pereții subțiri, cavitățile închise și decupajele sunt problematice, necesitând o simplificare conștientă mai degrabă decât o fidelitate geometrică deplină.

Această realizare introduce abordarea de proiectare pentru fabricație și subliniază faptul că deciziile de proiectare sunt întotdeauna legate de tehnologia de fabricație aleasă.

Următorul pas în analiza fizică este luarea deciziilor conștiente: ce să păstrezi și ce să omiți atunci când dezvolți propriul model.

Elevii stabilesc împreună ce elemente sunt necesare pentru amplasarea senzorilor, actuatorilor și controlerelor, unde să se asigure punctele de atașare pentru aceștia și ce părți pot fi omise fără a compromite obiectivele de învățare.

Deciziile se bazează întotdeauna pe funcționalitate și instruire, nu pe similaritatea formală cu soluția din fabrică. Scopul conceptualizării este de a crea o structură fizică care:

- nu conține spații goale inutile,
- economie de materiale și imprimare rapidă,
- ușor de instalat și modificat,
- oferă un mediu educațional transparent.

Acest pas are un impact direct asupra timpului de imprimare, a utilizării materialelor și a vitezei ciclurilor de iterație.

4.3. Conceptul fizic ca descriere a sistemului

În această etapă, elevii nu mai interpretează modelul ca pe o „casă” sau o „cutie”, ci ca pe un sistem de puncte pentru plasarea senzorilor, o structură care oferă o gamă de mișcare pentru actuatori și un suport pentru control și cablare.

Această abordare ajută la asigurarea faptului că designul nu se înecă prea devreme în detalii formale, ci rămâne interpretabil la nivel sistemic.

Pregătire conceptuală pentru proiectarea parametrică

Deși acest capitol nu introduce încă utilizarea Fusion 360, este introdusă ideea de bază a designului parametric. Dimensiunile nu sunt numere izolate, ci parametri interconectați, unde o modificare poate afecta mai multe elemente și unde scopul designului este adaptabilitatea și reproiectarea.

Această mentalitate formează o tranziție directă către următoarea parte, unde conceptul fizic este transformat într-un model digital, parametric.

Rezumat pedagogic

Această fază pregătitoare și de conceptualizare asigură că proiectarea ulterioară bazată pe Fusion 360 nu este un simplu exercițiu de utilizare a software-ului, ci o reprezentare digitală a deciziilor ingineresti conștiente.

Studentii vor experimenta faptul că un design bun începe înainte de pornirea computerului, că simplificarea și abstractizarea sunt virtuți ingineresti și că lumea fizică și cea digitală formează o unitate strânsă.

4.4. Mediul de proiectare Fusion 360 – elemente de bază și abordare

În această etapă, studenții nu mai vorbesc despre design digital la nivel teoretic, ci experimentează prin operații concrete cum un concept fizic devine mapabil cu un model digital, modificabil.

Studentii vor învăța elementele de bază ale Fusion 360 și utilizarea acestuia folosind publicația oficială Autodesk, „Ghidul participanților pentru instruirea Autodesk Fusion 360: Ghidul viitorului producției de lucruri”.

Pe baza materialului educațional procesat, procesul de învățare nu se bazează pe liste de funcții de învățare, ci pe dezvoltarea unor rutine de proiectare de bază, de exemplu:

- cum lansăm un model nou,
- cum navigăm în spațiul 3D,
- cum să creezi o geometrie simplă, dar stabilă.

Scopul pedagogic este ca elevii să recunoască faptul că proiectarea CAD este o activitate, nu crearea unei imagini vizuale.

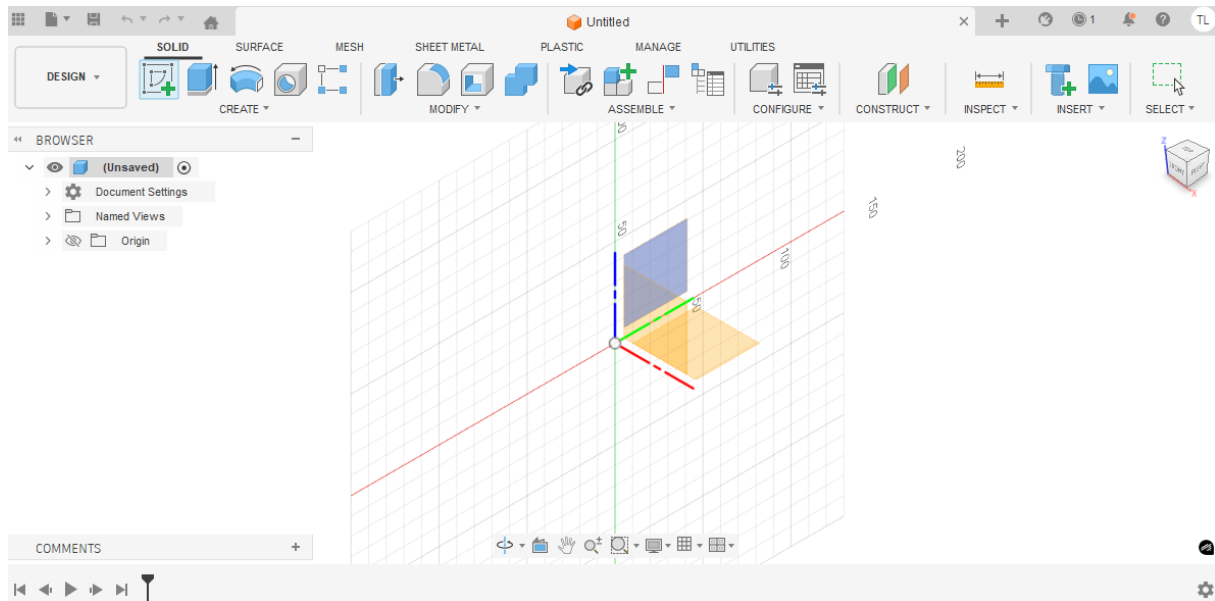


Figura 44: Mediul de proiectare Fusion 360 – elemente de bază și abordare (Sursa: Editare proprie)

Fusion 360 este prezentat în educație ca o „suprafață de gândire”. Documentul subliniază faptul că, chiar și la nivel de începător, se poate parcurge un ciclu complet de proiectare, creând modelul, modificându-l, revenind la stările anterioare și observând impactul modificărilor. Această abordare îi introduce în lumea modelării parametrice și a designului bazat pe povești, unde modelul nu este un obiect static, ci rezultatul unui proces.

Activități specifice studenților:

- Creați un design nou
- Salvare în proiect (înțelegerea funcționării bazate pe cloud)
- Revenirea la o stare anterioară în Cronologie

Termeni cheie din document:

- Modelare parametrică
- Design bazat pe istorie
- Editare nedistructivă

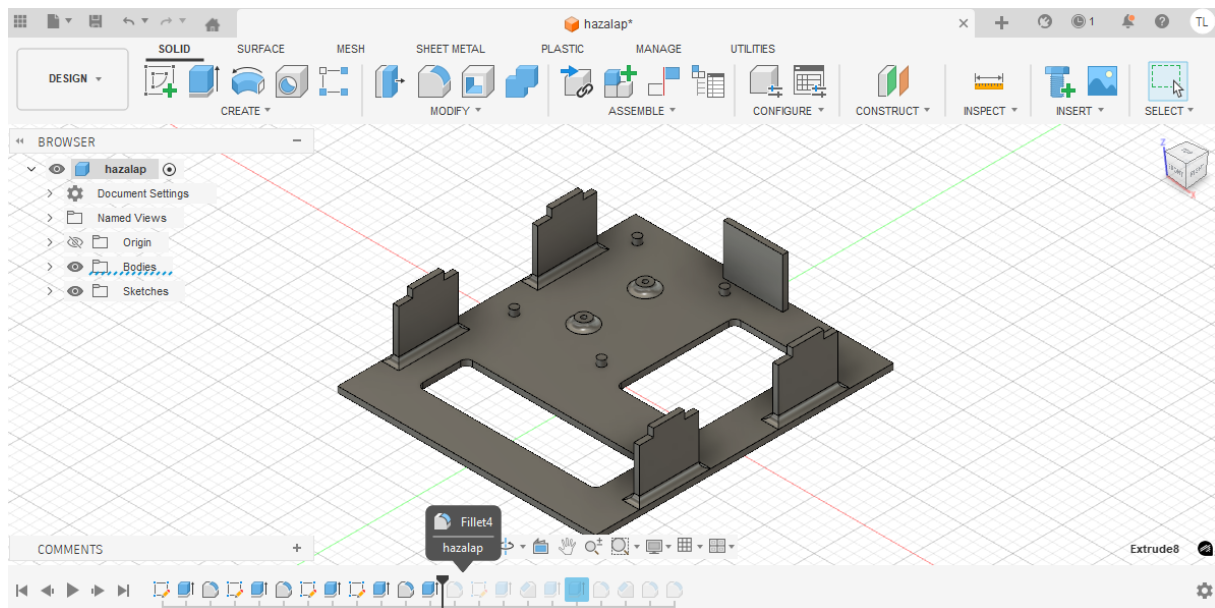


Figura 45: Crearea unui nou design – bază pentru casă inteligentă (Sursa: Editare proprie)

Focus pedagogic: elevii experimentează faptul că istoria modelului este la fel de importantă ca starea sa actuală.

Materialul de instruire Autodesk Fusion 360 prezintă interfața special concepută pentru începători, bazată pe logica de utilizare mai degrabă decât pe funcții.

4.5. Canvas – stăpânirea mișcării spațiale

Dezvoltării gândirii spațiale i se acordă un rol special. Elevii folosesc activ operațiuni de manipulare a vizualizării – rotirea, zoom-ul, mișcarea – precum și vizualizările de bază pentru a naviga cu încredere în spațiu. Această abilitate este o condiție prealabilă pentru crearea ulterioară a unor modele precise și bine gândite.

Exerciții pentru elevi:

- rotația modelului (Orbită),
- Mărire/micșorare,
- Pan,
- folosind vizualizări de bază (Sus, Față, Dreapta).

Scop pedagogic: dezvoltarea gândirii spațiale chiar înainte de desen.

Browser – „harta internă” a modelului

„Harta internă” a sistemului, Browser-ul, joacă un rol important în înțelegerea structurii modelului. Studenții învață să gestioneze schițe, corpuri și componente și își dau seama că un model bine structurat nu este doar mai transparent, ci și mai ușor de modificat ulterior.

Activități studentești:

- Ascunde/afișează schița,
- Separarea corpurilor,

- denumirea elementelor.

Termeni cheie:

- Dosar de schițe
- Dosarul Corpuri
- Componente

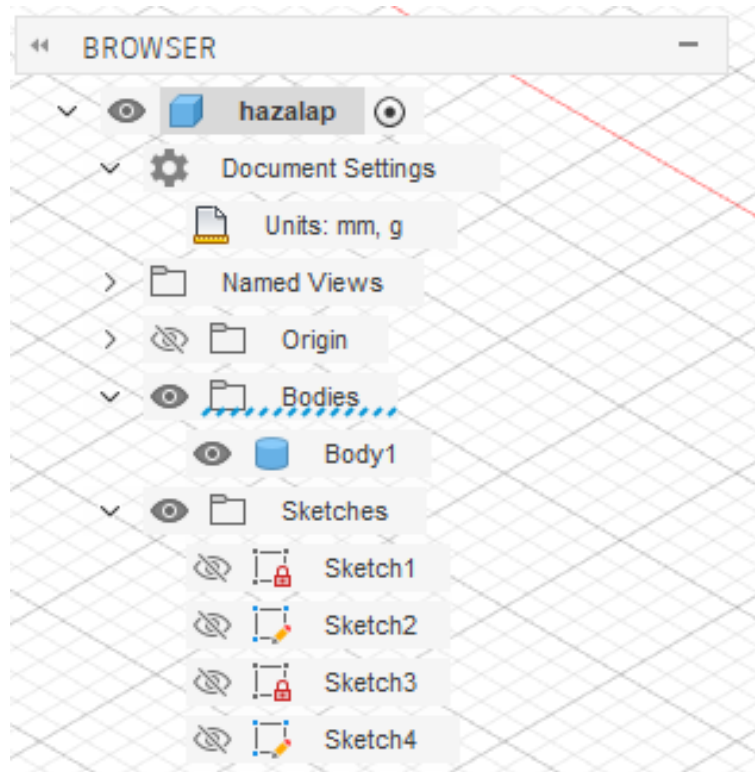


Figura 46: „Hartă internă” a modelului (Sursa: Editare proprie)

În paralel, utilizarea Cronologiei face vizibil sistemul cauzal al procesului de proiectare: modificarea unui pas anterior afectează întregul model, ceea ce ajută la dezvoltarea gândirii sistemice.

Documentul utilizează intens Cronologia ca instrument de învățare.

Practica studenților:

- modificați un pas de Extrudare,
- observând consecințele asupra întregului model.

Cuvinte cheie: Istoria designului



Figura 47: Utilizarea Cronologiei (Sursa: Editare proprie)

4.6. Concepte geometrice și de design de bază – într-un mod tangibil. Schiță – nu un desen, ci un set de reguli.

Este important să înțelegem conceptul de schiță, care nu este un desen, ci un set de reguli. Elevii experimentează faptul că o schiță nu este un desen simplu, ci un sistem definit matematic. Utilizarea dimensiunilor și a constrângerilor geometrice asigură stabilitatea și claritatea modelului. Problemele schițelor „incomplet definite” devin rapid vizibile, astfel încât elevii înțeleg importanța unui design precis din propria experiență.

Conform cursului Autodesk Fusion 360, cea mai frecventă greșeală făcută de designerii începători este o schiță nedefinită.

Practica studenților:

- desenarea unui dreptunghi,
- adăugarea dimensionării,
- aplicarea constrângerilor.

Termeni cheie:

- Schiță complet constrânsă
- Grade de libertate
- Constrângeri geometrice

Perspectivă pedagogică: ceea ce nu este definit va cauza probleme mai târziu.

Corp și componentă – când să folosiți care dintre ele?

Distincția dintre solide și componente apare, de asemenea, ca o decizie inginerescă importantă. Studenții recunosc că, deși un solid este potrivit pentru proiectarea rapidă, utilizarea componentelor necesită gândire structurală, în special pentru modele mai complexe. Această distincție îi pregătește și pentru sarcinile ulterioare de asamblare.

Documentul demonstrează prin exemple practice:

- Corp = modelare rapidă,
- Componentă = gândire structurală.

Situația decizională a elevului:

- când un singur corp este suficient,
- atunci când este justificată o componentă separată.

Concept cheie: Crearea componentei din corp

Parametrii ca instrumente de proiectare

În timpul introducerii în proiectarea parametrică, studenții experimentează faptul că modificarea unei singure dimensiuni poate afecta întregul model. Aceasta reprezintă nu doar o schimbare tehnică, ci și o schimbare de perspectivă: modelul nu mai este o formă fixă, ci un sistem schimbător care se adaptează nevoilor de proiectare.

Pe baza cursului de instruire Autodesk Fusion 360, studenții vor încerca:

- schimbarea unei dimensiuni,
- transformarea întregului model.

Termeni cheie:

- Parametri
- Dialog de modificare a parametrilor

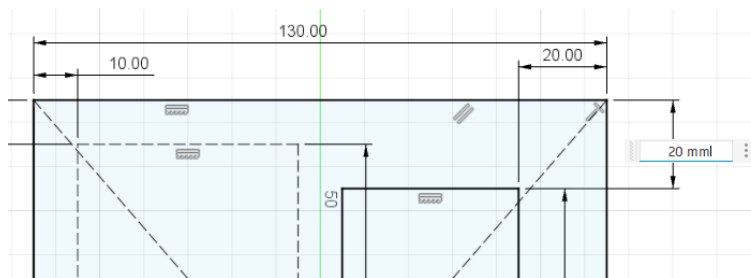


Figura 48: Utilizarea parametrilor (Sursa: Editare proprie)

Modelarea corpului – operațiuni de bază cu o mână sigură

Operațiile de bază de modelare solidă, cum ar fi extrudarea, rotunjirea sau repetarea, nu sunt prezentate izolat, ci sunt întotdeauna legate de situații specifice de proiectare. În acest fel, studenții nu învață comenzi, ci înțeleg rolul și consecințele acestora asupra modelului în ansamblu.

Operațiuni specifice, care pot fi învățate, bazate pe Autodesk Fusion 360 Training:

- Extrudare (Unire / Tăiere),
- Racordare și teșitură,
- Model (repetiție liniară).

Termeni cheie:

- Alăturare vs. Tăiere
- Racordare la margine
- Model dreptunghiular

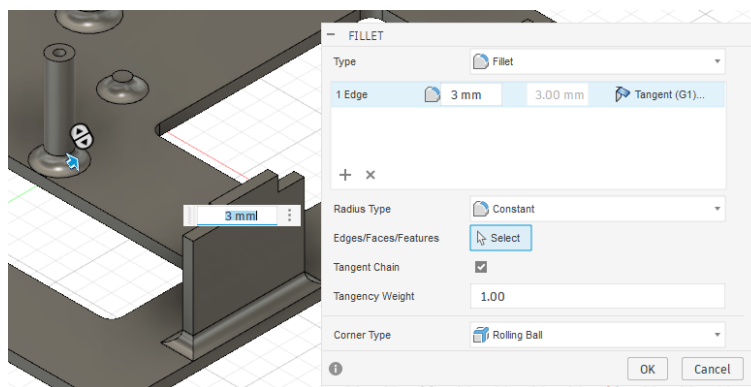


Figura 49: Modelarea corpului (Sursa: Editare proprie)

4.7. Pregătirea pentru imprimarea 3D – obiectiv practic final

Paginile 62–65 din Autodesk Fusion 360 Training se concentrează în mod specific pe starea „modelului finalizat”.

În etapa finală a procesului de proiectare, accentul se mută pe fezabilitatea fizică a modelului digital. Studenții învață că un model bine conceput nu este neapărat potrivit pentru fabricația automată, așadar trebuie acordată o atenție deosebită cerințelor imprimării 3D.

La verificarea modelului, principala considerație este asigurarea unei geometrii închise, deoarece imprimarea este posibilă doar cu volume clar definite. În plus, trebuie examinate grosimea peretelui, dimensiunea detaliilor și stabilitatea geometriei. Studenții vor experimenta modul în care deciziile de proiectare au un impact direct asupra calității și succesului fabricației.

Exportul modelului introduce formatele necesare pentru fabricație (cum ar fi STL sau 3MF) și procesul de creare a unui fișier de date imprimabil din modelul CAD. Acest pas încheie ciclul de proiectare și oferă și feedback: erorile și deficiențele de proiectare devin imediat vizibile în timpul creării modelului fizic.

Listă de verificare pentru elevi:

- corp închis (etanș),
- grosimea corespunzătoare a peretelui,
- export în format STL/3MF.

Termeni cheie:

- Previzualizare plasă
- Export pentru producție

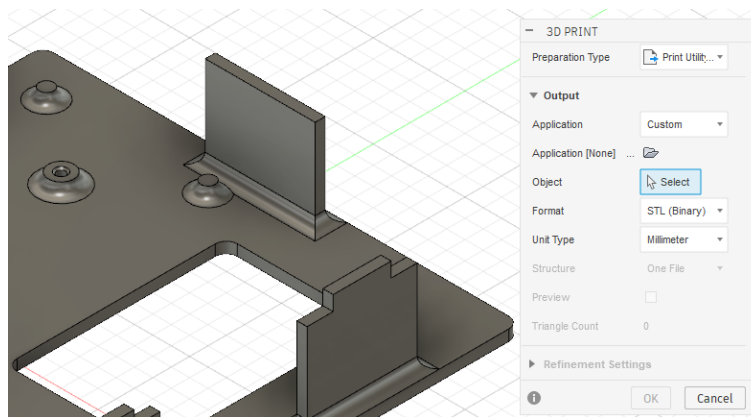


Figura 50: Pregătire pentru imprimare 3D (Sursa: Editare proprie)

Rezumat didactic

Această etapă este deosebit de importantă din punct de vedere didactic, deoarece devine clar pentru studenți că proiectarea digitală nu este o activitate în sine, ci pregătirea sistemelor care urmează să fie implementate în lumea fizică reală.

Până la sfârșitul întregului proces, studenții nu numai că vor crea un model, ci vor înțelege și întregul ciclu de proiectare: de la idee la modelul parametric și până la forma fabricabilă. Această experiență stabilește abordarea inginerescă în care proiectarea și implementarea formează o unitate inseparabilă.

În această parte a implementării proiectului, studenții nu învață să utilizeze un program software, ci mai degrabă încep să proiecteze, să construiască primul lor model digital pas cu pas și să realizeze că CAD este reprezentarea externă a gândirii.

Aceasta creează o bază solidă pentru următoarea fază, în care modelul digital devine un obiect fizic real prin imprimare 3D.

4.8. Rezultatele procesului de proiectare digitală – prezentarea și interpretarea modelelor studenților

Această subsecțiune prezintă modelele digitale create în timpul proiectului. Aceste elemente nu sunt mostre prefabricate, ci modele 3D pe care elevii le-au creat pe baza principiilor de proiectare învățate anterior și cu îndrumarea profesorului.

În timpul creării modelelor, studenții au aplicat în mod conștient principiul proiectării bazate pe funcții, gândirea parametrică, aspectele de modularitate și asamblare, precum și limitele geometrice ale fabricației aditive (imprimare 3D).

Dintr-o perspectivă pedagogică, aceste modele sunt amprente tangibile ale procesului de învățare: ele arată cum principiile abstracte de design se manifestă în forme fizice concrete.

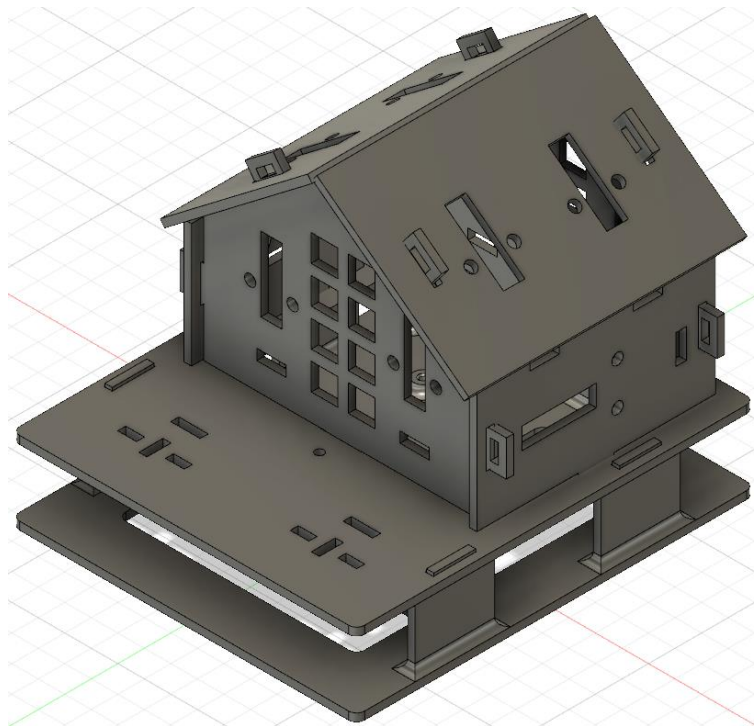


Figura 51: Model de asamblare a unei case inteligente (Sursa: Editare proprie)

Împreună, fișierele STL create de studenți formează baza fizică a unui model modular de casă inteligentă/fermă inteligentă. Scopul designului nu a fost de a crea un obiect final din punct de vedere estetic, ci de a crea o structură de susținere care:

- potrivit pentru găzduirea componentelor electronice,
- susține înregistrarea senzorilor și actuatorilor,
- permite asamblarea, măsurarea și modificarea,
- îndeplinește cerințele imprimării 3D desktop.

Astfel, studenților le-a devenit clar că modelul digital nu este un obiectiv independent, ci un cadru fizic pentru proiecte de electronică și control.

Placă de bază și elemente structurale inferioare – decizii de proiectare ale studenților

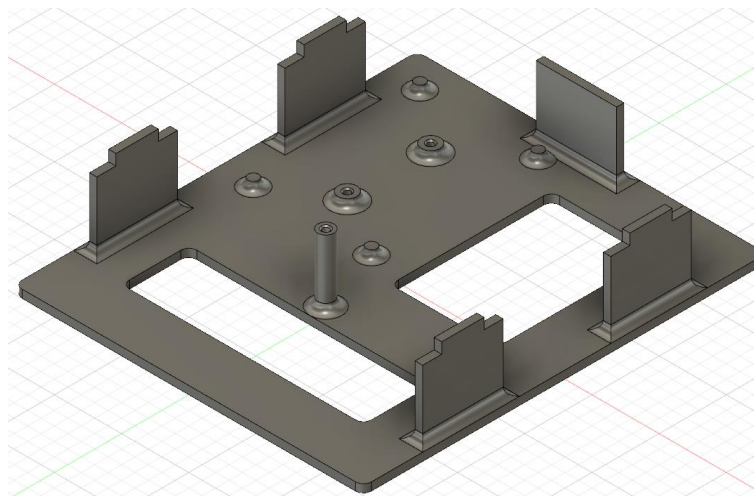


Figura 52: Placă de bază și elemente structurale inferioare (Sursa: Editare proprie)

La proiectarea plăcii de bază și a capacului inferior, studenții au aplicat aspectele ingineresti pe care le învățaseră anterior.

Decizii luate în procesul de proiectare:

- crearea unei suprafețe de sprijin stabile și plane,
- pregătirea pentru montarea modulelor electronice,
- asigurând o pozare ordonată a cablurilor.

Semnificație didactică: studenții au experimentat faptul că, în funcție de calitatea structurii de bază, este determinată utilizabilitatea întregului sistem și că, până și elementele aparent „simple” au în spate o gândire inginerească conștientă.

Panouri laterale – funcționalitate și accesibilitate

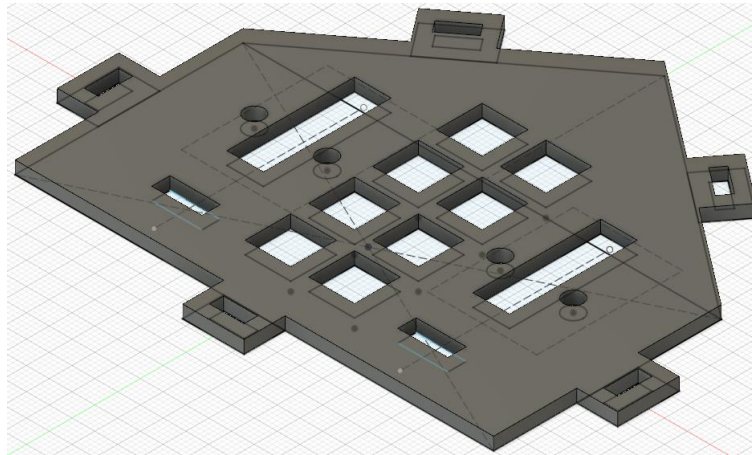


Figura 53: Panouri laterale (Sursa: Editare proprie)

La proiectarea panourilor laterale, studenții au urmărit în mod conștient să se asigure că modelul nu era o incintă închisă, ci o structură „lizibilă” care susținea obiectivele educaționale.

Considerațiile de proiectare au inclus asigurarea amplasării senzorilor și afișajelor, accesul pentru asamblare și măsurare și transparența vizuală a interiorului.

Lecție pedagogică: elevii au recunoscut că învelișul modelului educațional nu ascunde, ci face operațiunea inteligibilă.

Elemente de acoperiș – echilibru între protecție și demontabilitate

La proiectarea elementelor acoperișului, studenții au folosit soluții care protejează componentele electronice, fără a împiedica dezasamblarea și a facilita modificările ulterioare.

Această sarcină de proiectare a evidențiat clar faptul că „închiderea” nu înseamnă finalitate, ci mai degrabă un compromis conștient între protecție și accesibilitate.

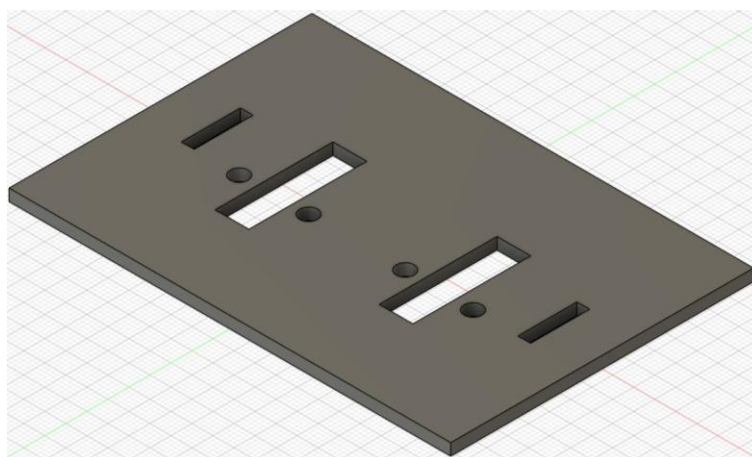


Figura 54: Element de acoperiș (Sursa: Editare proprie)

4.9. Rezumatul rezultatelor învățării pe baza modelelor completate

Modelele create în timpul proiectului oferă studenților oportunitatea de a analiza propriile decizii de proiectare și de a recunoaște consecințele acestora.

Concepte cheie surprinse prin intermediul modelelor:

- proiectare parametrică,
- modularitate,
- instalare,
- proiectare pentru producție,
- dezvoltare iterativă.

Această secțiune clarifică faptul că studenții nu au învățat doar utilizarea software-ului, ci și o abordare inginerească, care va fi completată în capitolele următoare cu imprimare 3D și asamblare fizică.

4.10. Implementarea practică a capitolului

În această etapă a devenit cu adevărat clar cât de dificil le-a fost studenților să accepte faptul că designul digital se bazează pe o înțelegere exactă a realității fizice. La început, analiza măsurătorilor, a schițelor proporționale și a relațiilor spațiale li s-a părut lentă și „mai puțin spectaculoasă”, dar mai târziu și-au dat seama că inexactitățile din model și din imprimare se reflectau imediat.

La introducerea Autodesk Fusion, a fost adesea necesară încetinirea deliberată a procesului. Studenții doreau să deseneze în timp ce trebuiau să învețe gândirea parametrică. Punctul de cotitură a fost atunci când au înțeles din propria experiență că o schiță prost structurată cauzează ulterior probleme serioase, în timp ce o structură bine gândită are ca rezultat un model stabil, ușor modificabil.

La proiectarea propriului nostru model, s-au luat decizii inginerești reale: ce să simplifice, unde să ofere acces, ce contează ca funcție reală. Supraaglomerarea era frecventă, ceea ce trebuia ținut sub control. Din perspectiva unui profesor, unul dintre cele mai mari rezultate a fost schimbarea gândirii elevilor: au început să vadă modelul nu ca pe un obiect, ci ca pe un sistem funcțional.

5. Experiență în imprimare 3D și producție

Un element cheie al proiectului a fost implementarea fizică a proceselor de proiectare digitală. Utilizarea imprimării 3D a oferit studenților oportunitatea de a dobândi experiență directă a conexiunilor dintre modelarea pe calculator și fabricația din lumea reală și de a înțelege importanța principiilor de proiectare pentru fabricație. Imprimarea 3D, care este populară astăzi și are o gamă de aplicații în continuă creștere, a fost în multe privințe o alegere ideală pentru implementarea fizică a acestei faze a proiectului.

În faza de producție, a devenit clar că soluțiile care păreau potrivite în spațiul digital necesitau adesea modificări în timpul implementării fizice. Această experiență a fost deosebit de valoroasă din punct de vedere educațional, deoarece studenții s-au confruntat cu probleme, analiza și rezolvarea acestora nefiind sarcini izolate, ci ca parte a unui proces de dezvoltare coerent.

5.1. Introducere în imprimanta 3D și Bambu Studio.

5.2. Scurtă introducere în procesul utilizat.

Procesul de fabricație aditivă utilizat în cadrul proiectului a fost tehnologia FDM (Fused Deposition Modeling), care este una dintre cele mai răspândite și ușor accesibile metode de imprimare 3D în prezent. Esența procesului constă în topirea de către imprimantă a unui fir continuu de plastic (filament) și apoi depunerea acestuia strat cu strat de-a lungul unor trasee predeterminate.

Tehnologia FDM este deosebit de potrivită pentru mediile educaționale, deoarece etapele procesului de fabricație pot fi ușor urmărite, iar efectele modificării parametrilor de imprimare pot fi observate imediat. Acest lucru a creat o oportunitate pentru studenți nu doar de a observa rezultatul final, ci și de a înțelege și interpreta întregul proces de fabricație.

5.3. Structura și funcționarea hardware-ului

Lucrările de imprimare sunt trimise către un **Bambu Lab X1 Carbon** Am implementat proiectul cu o imprimantă 3D desktop de tip închis. Dispozitivul are un sistem avansat de senzori, nivelare automată a mesei și un sistem de mișcare de înaltă precizie, care împreună asigură o funcționare stabilă și fiabilă. Am ales această imprimantă în timpul implementării proiectului deoarece căutam un dispozitiv printre alternativele disponibile pe piață care să nu numai că poată produce obiecte de înaltă calitate, cu numeroase opțiuni de calibrare și personalizare, ci și să o facă într-un mod eficient din punct de vedere al timpului. Utilizarea imprimantei a necesitat semnificativ mai puține procese de calibrare în fundal decât alte dispozitive, ceea ce ne-a permis să petrecem timp suplimentar pe zone mai importante ale proiectului. De asemenea, am selectat o imprimantă Prusa i3 MK3S+ ca alternativă, dar în cele din urmă am decis asupra X1 Carbon datorită manevrabilității sale mai ușoare și designului profesional.



Figura 55: Imprimantă 3D Bambu Lab X1 Carbon (Sursa:
<https://www.3djake.hu/bambu-lab/x1c>)

Spațiul de imprimare închis a jucat un rol deosebit de important în cadrul proiectului, deoarece a redus apariția erorilor de imprimare prin minimizarea influențelor mediului (temperatură, mișcarea aerului). Acest lucru le-a permis studenților să se concentreze asupra contextului tehnologic și să nu lase problemele generate de instabilitatea mediului să le domine experiența.

5.4. Sistemul de alimentare cu filament AMS



Figura 56: Dispenser de filamente AMS (Sursa: <https://www.3djake.hu/bambu-lab/x1c>)

Sistemul automat de alimentare cu filament AMS (Automatic Material System) conectat la imprimantă este capabil să gestioneze mai multe role de material simultan. Deși proiectul a produs în principal piese fabricate dintr-o singură culoare și un singur tip de material, utilizarea sistemului AMS a oferit cunoștințe tehnologice importante. De asemenea, a facilitat procesul de producție, deoarece nu a fost nevoie să se scoată și să se introducă filamentul în imprimantă la înlocuirea rolelor goale sau la schimbarea culorilor.

Studenții au învățat despre importanța manipulării materialelor și starea materiilor prime, acordând o atenție deosebită efectului umidității și importanței depozitării, evidențiind AMS-ul pasiv și activ.

dispozitive de uscare a filamentelor. În plus, funcționarea AMS a oferit un bun punct de plecare pentru introducerea ulterioară a capacităților de fabricație multi-material și multi-color.

5.5. Principii de feliere

Conversia modelelor digitale în forme fabricabile este **Bambu Studio** Am făcut asta folosind un software de slicing. Bambu Studio este software-ul open source propriu al producătorului de imprimante (Bambu Lab), care se bazează pe programul de slicing PrusaSlicer și este disponibil pentru toate sistemele de operare desktop majore.

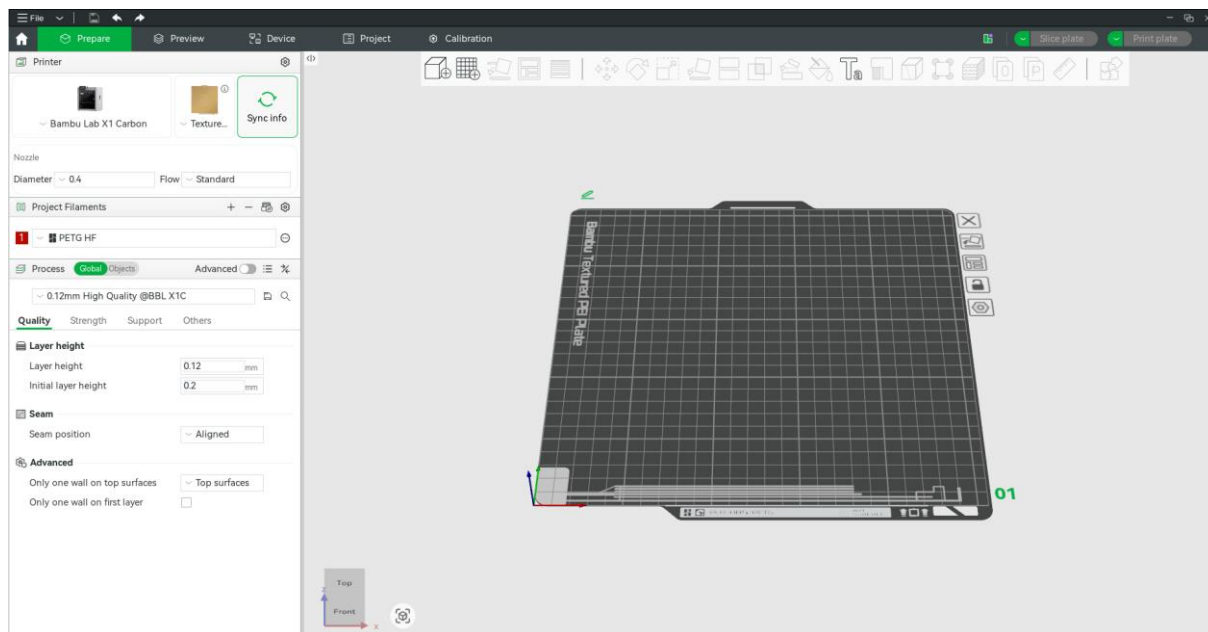


Figura 57: Software-ul de slicing Bambu Studio (Sursa: Editare proprie)

În timpul felierii, studenții s-au confruntat cu faptul că geometria singură nu este suficientă pentru o imprimare reușită; parametrizarea corectă este cel puțin la fel de importantă.

Prin modificarea setărilor de feliere, a devenit evidentă relația dintre timpul de imprimare, consumul de material și stabilitatea structurală. Această experiență i-a ajutat pe studenți să înțeleagă procesul de fabricație ca pe un sistem complex, mai degrabă decât ca pe o serie de etape izolate. În timpul proceselor de feliere, profilurile de imprimare implicate integrate în software au oferit un bun punct de plecare, care nu numai că a abordat calitatea imprimării, ci și configurația optimă rezultată din materialul utilizat. Cu ajutorul lor, diferența dintre setările individuale a devenit și mai clară.

Pe lângă specificarea parametrilor de imprimare, orientarea modelelor imprimate, setările de culoare și pregătirea tăvii au fost la fel de importante. O atenție deosebită a trebuit acordată aderenței optime a modelelor și necesității suporturilor.

5.6. Materiale (PLA, PETG, ABS)

În timpul proiectului, am examinat proprietățile mai multor materiale de imprimare 3D utilizate în mod obișnuit, însă, pentru producția reală **PETG** am ales materialul. Decizia s-a bazat atât pe considerații profesionale, cât și pedagogice.

PETG oferă un echilibru bun între ușurința imprimării și rezistența mecanică, fiind în același timp mai puțin predispus la deformare decât ABS. Acest lucru a fost benefic în special pentru piesele funcționale cu timpi de imprimare mai lungi. Acest lucru le-a permis studenților să experimenteze importanța selecției materialelor într-o situație reală de luare a deciziilor ingineresti.

Dorim să menționăm că în timpul implementării proiectului am avut la dispoziție și filament PLA, considerat materialul implicit pentru majoritatea proiectelor de imprimare 3D, dar datorită durabilității și rezistenței sale reduse la impacturi fizice, l-am abandonat și l-am folosit doar în scopuri de prototipare și testare. Pe lângă cele trei tipuri de materiale menționate, nu am investigat utilizarea altor tipuri de filament (de exemplu, TPU).

5.7. Pașii din procesul de imprimare

Implementarea practică a imprimării 3D s-a realizat pe parcursul unor etape structurate și interconectate. Structura conștientă a procesului le-a permis studenților să interpreteze fazele individuale nu ca operațiuni izolate, ci ca parte a unui sistem de producție coerent. În timpul etapelor de imprimare, s-a pus un accent deosebit pe pregătirea, monitorizarea procesului și evaluarea pieselor finalizate.

5.8. Pregătirea și verificarea modelului

În fiecare caz, imprimarea a fost precedată de o pregătire detaliată a modelului, care a avut ca scop prevenirea erorilor de fabricație și utilizarea eficientă a resurselor. Studenții au verificat corectitudinea geometrică a modelelor digitale, acordând o atenție deosebită corpurilor închise, grosimilor pereților și suprafețelor critice de îmbinare.

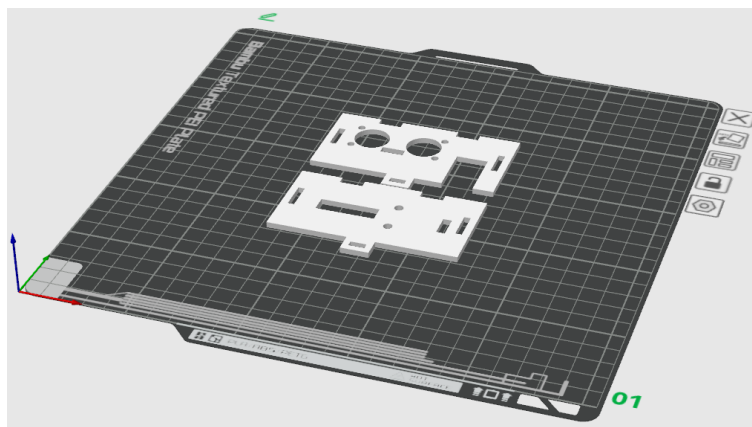


Figura 58: Pregătirea și verificarea modelului
(Sursa: Editare proprie)

În timpul pregătirii, alegerea orientării imprimării a jucat, de asemenea, un rol important. Studenții au observat că amplasarea piesei în spațiul de imprimare are un impact direct asupra

Există o nevoie de calitate a suprafeței, rezistență mecanică și cantitatea de material suport și suport necesară. Această situație decizională a facilitat dezvoltarea percepției spațiale și aplicarea conștientă a aspectelor legate de tehnologia de producție. Este de la sine înțeles că, odată cu aranjamentul optim al elementelor, studenții au avut oportunitatea de a începe procese de imprimare mai optimizate, deoarece au putut plasa mai multe modele pe o singură tavă.

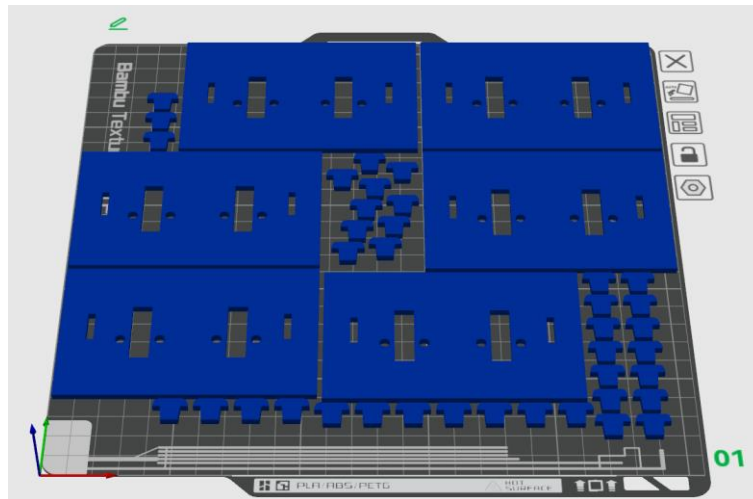


Figura 59: Optimizarea imprimării
(Sursa: Editare proprie)

La sfârșitul procesului de verificare, modelele au fost validate înainte de secționare, ceea ce a confirmat opinia că detectarea timpurie a erorilor este semnificativ mai eficientă decât corectarea lor ulterioară.

5.9. Imprimare și postproducție

La începutul procesului de imprimare, studenții au acordat o atenție deosebită aderenței corecte a primului strat, aceasta fiind unul dintre cei mai importanți factori pentru succesul întregii producții. Imprimanta X1 Carbon are un asistent AI încorporat care oprește imprimarea pentru scurt timp după plasarea primului strat și apoi îl inspectează prin intermediul camerei încorporate. Dacă detectează o problemă, o indică utilizatorului prin intermediul programului de slicing (sau al aplicației mobile Bambu Handy, instalabilă separat). După începerea imprimării, am considerat că este mai informativ ca studenții participanți să analizeze calitatea imprimării cu propriii ochi. Observarea primului strat le-a oferit studenților oportunitatea de a primi feedback imediat cu privire la corectitudinea pregătirii și setărilor.



Figura 60. Imprimanta la lucru

Monitorizarea continuă a procesului în timpul lucrărilor de imprimare mai lungi a fost, de asemenea, un element important de învățare. Studenții au experimentat faptul că imprimarea nu este o operațiune complet autonomă, ci un proces de fabricație care poate necesita intervenție sau reproiectare. Acesta este cazul, de exemplu, dacă un obiect se mișcă din loc sau filamentul iese din dozatorul AMS. Studenții au avut, de asemenea, ocazia să vadă cum sunt gestionate aceste situații.

După imprimare, piesele finite au fost post-procesate. Aceasta a inclus tratarea neregularităților suprafeței și rafinarea punctelor de îmbinare. În timpul post-procesărilor, a devenit clar că imprimarea 3D are uneori ca rezultat un produs semifinisat care necesită muncă manuală suplimentară pentru utilizare funcțională. Aceasta crește odată cu complexitatea modelelor și numărul de suporturi.



Figura 61: Rezultatele imprimării incorecte (Sursa: Editare proprie)

5.10. Controlul calității și remediarea erorilor

Piesele finite au fost supuse unui control detaliat al calității în toate cazurile. Inspecția nu a acoperit doar aspectele estetice, ci a evaluat în primul rând adecvarea funcțională și asamblarea.

Studentii au comparat dimensiunile pieselor imprimate cu datele modelului digital și au analizat cauzele oricăror discrepanțe. La identificarea erorilor, au luat în considerare și efectele setărilor de feliere, caracteristicile materialelor și mediul de imprimare. Astfel, corectarea erorilor nu a fost o intervenție izolată, ci mai degrabă o buclă de feedback pe parcursul întregului proces de fabricație.

Această abordare a favorizat dezvoltarea gândirii analitice și a pus bazele execuției justificate profesional a iterațiilor ulterioare de proiectare.

5.11. Probleme de dimensionare și potrivire

Una dintre cele mai importante lecții învățate în urma proiectului a fost importanța practică a aspectelor legate de dimensionare și montare. Studentii au experimentat direct faptul că dimensiunile nominale utilizate în proiectarea digitală nu duc întotdeauna la o potrivire corectă în timpul fabricației, prin urmare, gestionarea conștientă a toleranțelor este esențială.

5.12. Gestionarea inexactităților de fabricație

Inexactitățile din timpul imprimării 3D au fost rezultatul unei combinații de factori, cum ar fi contracția materialului, caracteristicile structurii straturilor și orientarea imprimării. Aceste fenomene au făcut ca limitele tehnologiei de fabricație să fie tangibile pentru studenți și au necesitat ajustări minime ale dimensionării modelelor.

Pentru a aborda variațiile de fabricație, studenții au aflat despre utilizarea spațiilor de montare și despre faptul că piesele mobile sau interconectate necesită o abordare de proiectare diferită față de piesele solide. Această abordare le-a ajutat să aprofundeze gândirea tehnică.

5.13. Iterații de proiectare

Rezolvarea problemelor de dimensionare necesita adesea mai multe cicluri de proiectare și fabricație. De asemenea, a fost necesar să se rezolve cazurile în care piesele nu se potriveau între ele din cauza erorilor de proiectare, mai degrabă decât din cauza procesului de imprimare. Pe baza experiențelor lor, studenții au modificat modelele și au retipărit piesele afectate, primind astfel feedback cu privire la corectitudinea deciziilor lor într-un timp scurt.

Avantajul designului parametric a fost evident în special în timpul procesului iterativ, deoarece capacitatea de a modifica și reface rapid dimensiunile a permis o experimentare eficientă. Această metodă i-a ajutat pe studenți să gândească în termeni de proces de dezvoltare flexibil, mai degrabă decât în termeni de soluții statice.

5.14. Semnificația educațională a proceselor, a experiențelor studenților și a lecțiilor învățate

Gestionarea problemelor de dimensionare și montare a avut o valoare educațională remarcabilă, deoarece studenții s-au confruntat cu situații ingineresti reale. Gândirea lor independentă și luarea deciziilor responsabile s-au dezvoltat prin analiza problemelor, recunoașterea erorilor și dezvoltarea de soluții.

Studentilor le-a devenit clar că designul și fabricația formează o unitate strânsă și niciuna nu poate fi înțeleasă fără cealaltă. Experiențele au contribuit la proiect nu doar prin transmiterea de cunoștințe tehnologice, ci și prin dezvoltarea unei abordări care poate fi utilizată pe termen lung.

În plus, experiența acumulată în imprimarea 3D oferă oportunități suplimentare de transfer de cunoștințe, care pot fi folosite atât la învățarea altor materii (matematică, fizică, materii tehnice), cât și chiar în mediul casnic, atunci când se operează propria imprimantă 3D.



Figura 62. Carcasă imprimată și asamblată

6. Sistem integrat de control – fundamente conceptuale

6.1. Scopul proiectării sistemului

În faza de electronică-software a proiectului a fost luată o decizie de proiectare conștientă: sistemul nu este un dispozitiv monolitic bazat pe un singur microcontroler, ci este implementat sub forma unei arhitecturi de control distribuit.

Este important de subliniat că acest lucru nu se datorează „slăbiciunii” ESP32. ESP32 este un controler de teren și o unitate de achiziție de date excelentă: oferă o gestionare stabilă a I/O-urilor digitale, are interfețe multiple de comunicație (I²C, SPI, UART, WiFi) și este, de asemenea, potrivit pentru rularea unei logici de control independente. Dacă numărul de GPIO-uri nu este suficient, în conformitate cu practicile din industrie, se pot utiliza expandoare I/O (de exemplu, expandoare de porturi I²C) sau noduri suplimentare - acest lucru în sine indică spre sisteme distribuite.

Adevărata limitare în timpul dezvoltării nu a fost în domeniul I/O sau al comunicării, ci în implementarea funcțiilor de monitorizare și afișare (HMI). ESP32:

- are spațiu limitat de RAM și stocare flash,
- optimizat pentru rularea interfețelor utilizator non-web, gestionarea bazelor de date și înregistrarea în jurnal persistentă,
- Când se afișează grafică complexă și se deservesc mai mulți clienți, se ating rapid limitele resurselor.

În practică, aceasta însemna că, deși ESP32 era capabil să îndeplinească stabil sarcini de teren (senzori, actuatoare, control local), stratul de monitorizare al sistemului justifică includerea unui dispozitiv cu o capacitate de calcul mai mare.

Arhitectura rezultată împarte sistemul în straturi clar distincte:

Nivel	Funcție	Dispozitiv
Nivel de teren	Colectarea datelor senzorilor, controlul actualelor, logică locală	ESP32
Control / nivelul de gestionare a datelor	Procesare ciclică, decizii logice, rol de master Modbus	Raspberry Pi / PC (Modbus)
Nivel de supraveghere (HMI)	Afișaj web, control utilizator, înregistrare date	Raspberry Pi / PC (server bazat pe Flask)

În acest model, ESP 32 acționează ca un nod de câmp și, urmând un model industrial, furnizează date sistemului de nivel superior prin intermediul unui protocol standard (Modbus TCP). Este important de menționat că într-un sistem pur monolitic, cu un singur dispozitiv, acest standard de comunicație nu ar fi necesar - dar arhitectura distribuită îl justifică.

Prin urmare, scopul proiectului nu a fost doar de a crea un model funcțional de fermă inteligentă, ci de a demonstra că:

- cum un microcontroler devine un nod de câmp,
- cum este organizată deasupra acesteia o logică de sistem de supraveghere și un strat HMI,
- și cum poate fi modelată arhitectura stratificată, în rețea, a automatizării industriale într-un mediu educațional.

Această abordare îi apropie pe studenți de funcționarea sistemelor industriale reale, în timp ce sistemul poate fi extins cu ușurință cu noi senzori, noduri și servicii.

Conceptul de sistem distribuit

Modelul de control dezvoltat în cadrul proiectului urmează în mod conștient o arhitectură stratificată, pe două niveluri. Această structură nu a reprezentat doar o decizie tehnică, ci și un pas în ceea ce privește abordarea de la abordarea monolitică, „totul pe un singur microcontroler”, către structura tipică sistemelor industriale.

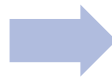
Nivelul inferior este nivelul de câmp, unde senzorii și actuatorii sunt gestionați direct. Acest rol este jucat de ESP32 în sistem. Sarcinile sale includ gestionarea semnalelor digitale și analogice, colectarea datelor senzorilor, preprocesarea de bază și operarea ieșirilor fizice (de exemplu, LED, releu, servo). Acest nivel este locul unde are loc conexiunea directă cu procesul fizic real.

Stratul superior este nivelul de supraveghere, care în proiect este implementat de un sistem bazat pe Raspberry Pi (sau PC). O diferență importantă este că la acest nivel, logica centrală de control nu rulează, ci este afișată o imagine digitală de stare a sistemului. Pi colectează date de la dispozitivele de nivel de teren (ESP32), le înregistrează și le afișează pe o interfață web HMI și oferă posibilitatea intervenției de la distanță.

Controlul propriu-zis se face la nivel de teren, pe ESP32. Gestionarea I/O, procesarea datelor senzorilor și logica de operare de bază rulează local, astfel încât sistemul rămâne operațional chiar și în modul offline. Defectarea nivelului de monitorizare nu oprește procesul fizic, fiind afectate doar afișajul și accesul de la distanță.

Prin urmare, această structură poate fi comparată mai precis cu următorul model industrial:

I/O la distanță + control încorporat



Sistem de supraveghere (SCADA/HMI)

Echivalentele din proiect sunt:

- ESP32 ca unitate inteligentă de câmp care
 - el Efectuează gestionarea I/O
 - el colectează date de la senzori
 - el rulează logica de control local
 - el Comunică ca un sistem de supraveghere Modbus
- Raspberry Pi + Python + Flask ca strat de management, care
 - el colectează și afișează date care
 - el reflectă starea sistemului

- el permite intervenția operatorului, dar nu este
- el esențială pentru funcționarea procesului

În viitor, la nivel de Raspberry Pi pot fi implementate funcții suplimentare, cum ar fi:

- înregistrarea datelor și analiza tendințelor pe termen lung
- afișaj grafic
- gestionarea autorizațiilor utilizatorilor
- notificări, jurnale de alarme
- interfață de configurare la distanță

Aceste funcții ar trebui să fie doar de natură de supraveghere și informare. Este un principiu de proiectare ca adăugirile care rulează pe Raspberry Pi să nu afecteze funcționalitatea de bază sau siguranța sistemului. Logica de control critic și gestionarea stării de bază rămân la nivel de teren.

6.2. Avantajele tehnice ale arhitecturii distribuite

Una dintre cheile sistemului este separarea funcțională, care crește în mod direct fiabilitatea și scalabilitatea. ESP32 rămâne la nivel de câmp: senzorii, intrările, ieșirile și logica de control de bază rulează local, astfel încât funcționarea procesului fizic nu depinde de o conexiune continuă la rețea. Acest lucru are ca rezultat un comportament I/O mai determinist și o funcționare mai stabilă.

Stratul superior este responsabil pentru monitorizare, gestionarea și vizualizarea datelor, nu pentru controlul critic.

Această arhitectură stratificată urmează un model industrial, în care dispozitivul de teren, stratul de control și sistemul de monitorizare operează în roluri separate.

Un avantaj semnificativ al arhitecturii este scalabilitatea: noi noduri bazate pe ESP pot fi adăugate la rețea, se pot integra senzori suplimentari, iar sistemul poate fi extins funcțional fără a suprasolicita resursele unui singur dispozitiv. Această extindere orizontală este o caracteristică fundamentală a sistemelor distribuite.

Din perspectiva fiabilității, principiul este că nivelul de teren rămâne autonom: o defecțiune a sistemului sau a rețelei de supraveghere nu oprește operațiunile locale, fiind afectate doar monitorizarea și intervenția de la distanță. Acest lucru corespunde direct practicii automatizării industriale.

6.3. Model și abordare de comunicare

Conexiunea dintre straturi se bazează pe Modbus TCP. Aceasta implementează un schimb de date bazat pe stări, cu o structură standard de registre. Datele senzorilor și semnalele de control nu se deplasează sub forma unor „cereri și răspunsuri” specifice, ci ca stări accesibile la adrese bine definite. Aceasta diferă de abordarea tranzacțională tipică bazată pe HTTP și este mai apropiată de modelele de comunicație industrială, unde controlerul citește ciclic stările la nivel de câmp și apoi scrie registrele de ieșire în consecință.

Cadrul sistemului – structură logică

La nivel general, calea către datele sistemului poate fi descrisă după cum urmează:

Senzorii (buton, DHT etc.) și actuatorii din lumea fizică sunt conectați la ESP32. ESP32 acționează ca un sistem de supraveghere Modbus și pune la dispoziție starea intrărilor și ieșirilor prin intermediul registrelor. Prin intermediul rețelei TCP bazate pe WiFi, sistemul care rulează la nivel de supraveghere – Raspberry Pi sau PC – citește și scrie aceste date ciclic ca un client Modbus. Stările procesate și valorile măsurate sunt apoi afișate pe interfața web HMI, unde sunt implementate și vizualizarea și controlul de la distanță.

În acest model, ESP-ul nu este un controler central, ci o unitate inteligentă de intrare/ieșire pe teren, în timp ce Raspberry Pi servește ca centru logic și de monitorizare.

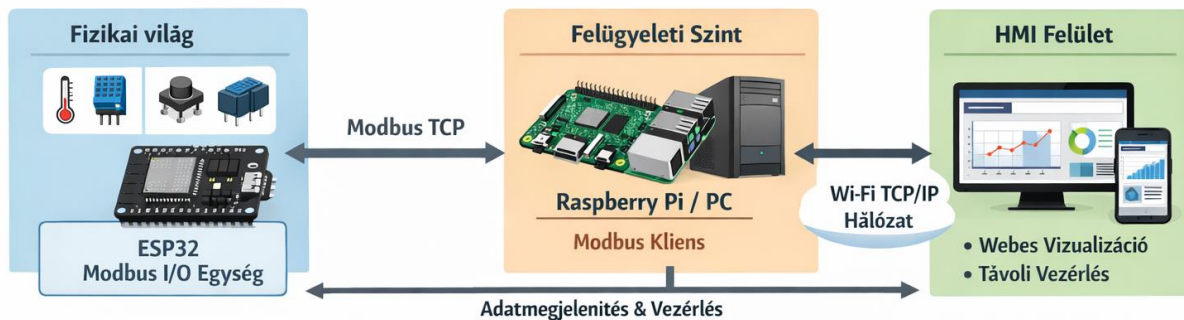


Figura 63: Cadrul sistemului (Sursa: generat de inteligența artificială)

Semnificație educațională

Arhitectura aplicată nu este doar o soluție tehnică, ci și un instrument pentru modelarea atitudinilor. Studenții înțeleg motivul separării nivelurilor de teren și de supraveghere, rolul alocării resurselor și modul în care este construită structura stratificată a unui sistem de automatizare industrială. Aceasta oferă o bază directă pentru înțelegerea programării PLC, a proiectării HMI și a sistemelor SCADA, deoarece modelul utilizat în proiect este o reprezentare simplificată, dar realistă a acestora.

În secțiunea următoare, prezentăm implementarea software specifică a arhitecturii: programul de câmp ESP, procesarea ciclică pe partea de monitorizare și funcționarea interfeței web.

6.4. Nod de control pe teren - unitate de probă independentă

Nivelul inferior al sistemului electronic al proiectului este reprezentat de un nod de câmp autonom, bazat pe un microcontroler ESP32. Această unitate este conectată direct la lumea fizică: citește senzorii, controlează actuatorii și oferă feedback local. Configurația prezentată este un exemplu deliberat simplu și educativ care ilustrează logica de bază a sistemului și nu reprezintă o versiune finală.

Nodul gestionează următoarele elemente:

- un buton care funcționează ca o intrare digitală,
- un LED de ieșire digitală,
- un senzor de temperatură și umiditate DHT11,
- un afișaj LCD de 16x2 caractere conectat la o magistrală I²C.

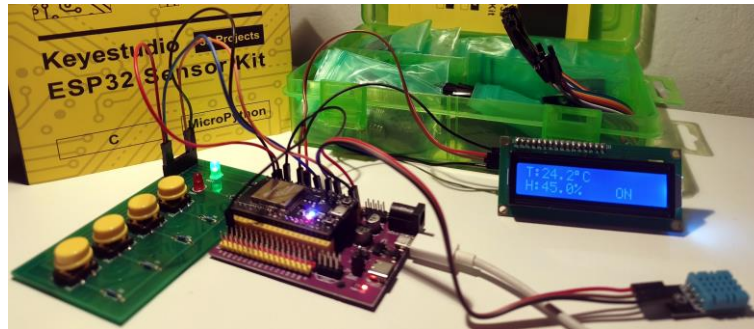


Figura 64: Mediu de testare (Sursa: Editare proprie)

Esența funcționării constă în faptul că dispozitivul își îndeplinește sarcinile de bază complet independent, fără o conexiune la rețea. Utilizatorul poate interveni în starea sistemului (LED pornit/oprit) cu ajutorul butonului fizic, în timp ce datele de mediu sunt măsurate continuu și afișate pe ecran. Această abordare modelează principiul industrial conform căruia nivelul de teren ar trebui să poată funcționa autonom, independent de sistemul de monitorizare.

Programul prezentat demonstrează astfel comportamentul unei unități inteligente de intrare/ieșire (I/O). Structura este modulară: senzori noi, dispozitive de ieșire sau afișaje pot fi adăugate cu ușurință folosind GPIO-uri suplimentare sau magistrale de comunicație. Ulterior, acest nod poate fi integrat și cu sisteme de nivel superior (PLC, HMI, SCADA), dar în acest capitol ne vom concentra exclusiv pe gestionarea resurselor locale.

6.5. Structura programului defalcată în unități funcționale

Biblioteci și definiții hardware

```
1 // ===== KÖNYVTÁRAK BETÖLTÉSE =====
2 // Ebben a szakaszban történik a felhasznált perifériákhoz szükséges könyvtárak betöltése.
3 // DHT könyvtár a szenzor kommunikációját valósítja meg.
4 // A Wire és hd44780_I2Cexp az I2C LCD vezérléséhez szükséges.
5 #include <DHT.h>
6 #include <Wire.h>
7 #include <hd44780.h>
8 #include <hd44780ioClass/hd44780_I2Cexp.h>
9
10
11 // ===== HARDVER KIOSZTÁS =====
12 //A #define direktívák rögzítik a fizikai bekötés és a program közötti kapcsolatot, így a
13 //hardverkonfiguráció egyértelműen dokumentált és könnyen módosítható.
14 #define BTN 17 // Fizikai gomb bemenet
15 #define LED 2 // LED kimenet
16 #define DHTPIN 4 // DHT adatvonal
17 #define DHTTYPE DHT11 //DHT típus megadása
```

În această etapă, se încarcă bibliotecile necesare pentru perifericele utilizate.

- Biblioteca DHT implementează comunicarea între senzori.
- Cablul și hd44780_I2Cexp sunt necesare pentru a controla ecranul LCD I²C.

Directivile #define înregistrează relația dintre conexiunea fizică și program, astfel încât configurația hardware este documentată clar și poate fi modificată cu ușurință.

Obiecte și variabile globale

```
20 //Itt jönnek létre a szenzor- és kijelzőobjektumok. Ezek a program teljes futása alatt elérhetők, így
    bármelyik ciklusban használhatók.
21 // Szenzor objektum
22 DHT dht(DHTPIN, DHTTYPE);
23 // I2C LCD objektum
24 hd44780_I2Cexp lcd;
```

Aici sunt create senzorul și obiectele de afișare. Acestea sunt disponibile pe toată durata rulării programului, astfel încât pot fi utilizate în orice ciclu.

Gestionarea butonului anti-ridicare

```
27 // ===== GOMB PRELLEGÉS SZŰRÉS =====
28 // A mechanikus gombok érintkezési zárásakor rövid ideig „rezegnek” (prellegnek), ami több hamis jelváltást
    okozhat.
29 // Ez a logika időalapú szűréssel biztosítja, hogy csak a stabil állapotváltás legyen érvényes.
30
31 bool lastReading = false; // Legutóbbi nyers bemenet
32 bool stableState = false; // Szűrt, stabil állapot
33 unsigned long lastChangeTime = 0;
34 const unsigned long debounceTime = 50; // 50 ms stabilitási küszöb
```

Contactele butoanelor mecanice „vibrează” (sari) pentru o perioadă scurtă de timp atunci când sunt închise, ceea ce poate cauza multiple tranziții false de semnal. Această logică folosește filtrarea bazată pe timp pentru a se asigura că doar tranzițiile de stare stabile sunt valide.

Starea LED-urilor și variabilele de măsurare

```
37 // A LED állapota külön változóban tárolódik. A DHT mérések időzítéséhez és az aktuális értékek
    tárolásához is globális változók szükségesek.
38 bool ledState = false;
39 unsigned long lastDHTread = 0; // Utolsó olvasás időpontja
40 float temp = 0; // Hőmérséklet cache
41 float hum = 0; // Páratartalom cache
```

Starea LED-ului este stocată într-o variabilă separată. Variabilele globale sunt necesare și pentru cronometrarea măsurătorilor DHT și stocarea valorilor curente.

Inițializare – setup()

```
50 void setup() {
51
52     // Hardver inicializálás
53     pinMode(BTN, INPUT);
54     pinMode(LED, OUTPUT);
55
56     // DHT inicializálás
57     dht.begin();
58
59     // I2C inicializálás ESP32 alapértelmezett lábakkal
60     Wire.begin(21,22);
61
62     // LCD inicializálás
63     lcd.begin(16,2);
64     lcd.backlight();
65
66     lcd.setCursor(0,0);
67     lcd.print("Smart Farm Node");
68     delay(2000);
69     lcd.clear();
}
```

setup() rulează o singură dată la pornire. Iată ce se întâmplă:

- setarea direcțiilor GPIO,
- inițializarea senzorului și a magistralei I²C,
- pornirea ecranului LCD.

Gestionarea butoanelor și comutarea LED-urilor

Acest bloc implementează comutarea LED-ului pe muchia de apăsare. LED-ul se schimbă doar atunci când butonul este apăsat constant.

```
74 //Ez a blokk valósítja meg a lenyomási élre történő LED-váltást.  
75 //A LED csak akkor változik, amikor a gomb stabilan lenyomott állapotba kerül.  
76 // Gomb olvasása  
77 bool reading = digitalRead(BTN) ;  
78  
79 // Prellégés detektálása  
80 if (reading != lastReading) {  
81     lastChangeTime = millis() ;  
82 }  
83  
84 // Stabil állapot vizsgálata  
85 if ((millis() - lastChangeTime) > debounceTime) {  
86  
87     // Állapotváltozás kezelése  
88     if (reading != stableState) {  
89         stableState = reading;  
90  
91         // Lenyomás detektálás  
92         if (stableState == true) {  
93             ledState = !ledState; // LED kapcsolás  
94         }  
95     }  
96 }  
97  
98 lastReading = reading;
```

Măsurare periodică DHT11

```
100 // DHT szenzor olvasás időzítettten  
101 //A szenzor csak 2 másodpercenként kerül kiolvasásra, ami megfelel a DHT11 időzítési követelményeinek.  
102 if(millis() - lastDHTread > 2000){  
103     lastDHTread = millis();  
104  
105     float h = dht.readHumidity();  
106     float t = dht.readTemperature();  
107  
108     // Hibás olvasás kizárása  
109     if(!isnan(h) && !isnan(t)){  
110         temp = t;  
111         hum = h;  
112     }  
113 }
```

Senzorul este citit doar la fiecare 2 secunde, ceea ce îndeplinește cerințele de temporizare ale DHT11.

Afișaj LCD

```
115 // LCD kijelzés  
116 //A kijelző a mért adatokat és a LED állapotát jeleníti meg, így a rendszer állapota hálózat nélkül is ellenőrizhető.  
117 lcd.setCursor(0,0);  
118 lcd.print("T:");  
119 lcd.print(temp,1);  
120 lcd.print((char)223);  
121 lcd.print("C ");  
122  
123 lcd.setCursor(0,1);  
124 lcd.print("H:");  
125 lcd.print(hum,1);  
126 lcd.print("% ");  
127  
128 lcd.setCursor(11,1);  
129 lcd.print(ledState ? "ON " : "OFF");
```

Afișajul afișează datele măsurate și starea LED-urilor, astfel încât starea sistemului poate fi verificată chiar și fără rețea.

Rezumat

Acest program demonstrează funcționarea unei unități autonome de control pe teren:

- procesarea fizică a intrărilor
- controlul actuatorului
- colectarea datelor de la senzori
- vizualizare locală

Structura codului separă bine funcțiile individuale, ceea ce facilitează extinderea ulterioară și conectarea la sisteme de nivel superior.

6.6. Nod de control pe teren cu extensie de rețea – WiFi + Modbus TCP

În această versiune, controlerul de câmp independent introdus anterior este suplimentat cu capacități de rețea.

Operarea locală (buton → LED, măsurare DHT, afișaj LCD) rămâne neschimbată, dar dispozitivul este acum capabil să:

- Conectați-vă la o rețea WiFi
- Acționează ca un sistem de monitorizare Modbus TCP
- face disponibilă starea intrărilor și ieșirilor fizice prin intermediul registrelor
- pentru citire la distanță și intervenție de către un sistem extern (PLC / software de monitorizare)

Este important ca logica să nu se mute în afara dispozitivului de teren: controlul local rămâne autonom, rețeaua fiind utilizată doar pentru partajarea stării și intervenția de la distanță.

Structura programului conform unităților

funcționale Biblioteci și definiții hardware

```
1 #include <WiFi.h> // ESP32 WiFi stack
2 #include <ModbusTCP.h> // Modbus TCP szerver kezelése
3 #include <DHT.h> // DHT szenzor magas szintű kezelése (protokollt elrejt)
4 #include <Wire.h> // I2C kommunikáció (LCD miatt)
5 #include <hd44780.h> // LCD vezérlő
6 #include <hd44780ioClass/hd44780_I2Cexp.h> // I2C LCD expander kezelés
7
8 ModbusTCP mb; // Az ESP Modbus TCP szerverének létrehozása.
9
10
11 // ===== HARDVER KIOSZTÁS =====
12 // GPIO-k definiálása hardver absztrakcióként
13
14 #define BTN 17 // Fizikai gomb bemenet
15 #define LED 2 // Fő LED kimenet (vezérelt fogyasztó)
16 #define WIFI_LED 5 // WiFi státusz visszajelző LED (terepi hibajelzés)
17 #define DHTPIN 4 // DHT adatvonal
18 #define DHTTYPE DHT11 // Szenzor típusa
19
20 // Szenzor objektum létrehozása
21 DHT dht(DHTPIN, DHTTYPE);
22
23 // I2C LCD objektum létrehozása (automatikus I2C címfelismerés)
24 hd44780_I2Cexp lcd;
```

Parametrizarea rețelei

```
27 // ===== WIFI HÁLÓZATI PARAMÉTEREK =====
28 // Ezek konfigurálják a hálózati kapcsolatot, valamint biztosítják,
29 // hogy az ESP mindig ugyanazon IP címen legyen elérhető.
30
31 const char* ssid = "plc";
32 const char* pass = "12345678";
33
34 IPAddress local_IP(192, 168, 137, 200);
35 IPAddress gateway(192, 168, 137, 1);
36 IPAddress subnet(255, 255, 255, 0);
37 IPAddress primaryDNS(8,8,8,8);
38 IPAddress secondaryDNS(8,8,4,4);
39
40
41 // ===== WIFI ÚJRACSATLAKOZÁSI LOGIKA =====
42 // A rendszer nem blokkoló reconnect stratégiát használ.
43 // Ha a kapcsolat megszakad, a program nem áll meg,
44 // hanem meghatározott időközönként újrapróbál csatlakozni.
45
46 unsigned long lastReconnectAttempt = 0; // Utolsó reconnect próbálkozás időpontja
47 const unsigned long reconnectInterval = 300000; // 5 perc újrapróbálkozási ciklus
48 bool wifiConnected = false; // Hálózati állapot logikai jelző
49
50 // WiFi csatlakozási rutin
51 // Ez a függvény bontja az előző kapcsolatot,
52 // majd új kapcsolatot kezdeményez a megadott SSID-re.
53 void connectToWiFi() {
54     WiFi.disconnect(true);
55     delay(100); // Rövid stabilizációs várakozás
56     WiFi.begin(ssid, pass);
57 }
58
59
60 // ===== WIFI HIBA LED IDŐZÍTÉS =====
61 // Amennyiben nincs WiFi kapcsolat,
62 // a WIFI_LED másodperces periódussal villog.
63 // A megvalósítás millis() alapú, tehát nem blokkolja a fő ciklust.
64
65 unsigned long lastBlinkTime = 0; // Utolsó LED állapotváltás
66 const unsigned long blinkInterval = 1000; // 1 másodperces villogási periódus
67 bool wifiLedState = false; // WiFi LED aktuális állapota
```

Gestionarea anti-reculului butonului, starea LED-urilor și variabilele de măsurare

```
70 // ===== BELSŐ ÁLLAPOT RÉTEG =====
71 // A rendszer egyetlen "igazságforrást" használ a LED állapotára.
72 // A fizikai gomb, a Modbus és a fizikai LED is ezt az állapotot követi.
73
74 bool ledState = false; // A LED központi logikai állapota
75
76
77 // ===== GOMB PRELL SZŰRÉS =====
78 // Mechanikus gomb érintkezése záráskor pattog (prell).
79 // A stabil állapot detektálás millis alapú időszűréssel történik.
80
81 bool lastReading = false; // Utolsó nyers bemeneti érték
82 bool stableState = false; // Szűrt, stabil állapot
83 unsigned long lastChangeTime = 0;
84 const unsigned long debounceTime = 50; // 50 ms stabilitási küszöb
85
86
87 // ===== DHT MINTAVÉTELI IDŐZÍTÉS =====
88 // A DHT szenzor olvasása nem történhet túl gyakran,
89 // ezért 2 másodperces mintavételezési ciklust alkalmazunk.
90
91 unsigned long lastDHTread = 0;
92 float temp = 0;
93 float hum = 0;
```

Inițializare

```
97 // ===== SETUP =====
98 // =====
99 void setup() {
100
101     Serial.begin(115200);
102
103     // Hardver inicializálás
104     pinMode(BTN, INPUT);
105     pinMode(LED, OUTPUT);
106     pinMode(WIFI_LED, OUTPUT);
107
108     // ===== Stabil statikus IP konfiguráció =====
109     // Az ESP32 WiFi stack sajátossága miatt:
110     // - Station mód explicit beállítása szükséges
111     // - DHCP cache törlése
112     // - Statikus IP konfigurálása csatlakozás előtt
113
114     WiFi.mode(WIFI_STA);
115     WiFi.persistent(false);
116     WiFi.disconnect(true);
117     delay(100);
118
119     WiFi.config(local_IP, gateway, subnet, primaryDNS, secondaryDNS);
120     WiFi.begin(ssid, pass);
121
122     // Induláskori maximum 10 másodperces várakozás.
123     // Ha ezalatt nem jön létre kapcsolat,
124     // a program offline módban folytatódik.
125     unsigned long startAttemptTime = millis();
126     while (WiFi.status() != WL_CONNECTED &&
127           millis() - startAttemptTime < 10000) {
128         delay(500);
129     }
130
131     wifiConnected = (WiFi.status() == WL_CONNECTED);
132
133     // Szenzor és perifériák inicializálása
134     dht.begin();
135     Wire.begin(21,22);
136
137     lcd.begin(16,2);
138     lcd.backlight();
139     lcd.setCursor(0,0);
140     lcd.print("Smart Farm Node");
141     delay(2000);
142     lcd.clear();
```

Alocare registre Modbus

```
144 // ===== MODBUS REGISZTERKIOSZTÁS =====
145 // Coil 0 → LED állapot
146 // Ists 0 → Gomb állapot
147 // Hreg 0 → Hőmérséklet (x10 skálázva)
148 // Hreg 1 → Páratartalom (x10 skálázva)
149
150 mb.server();
151 mb.addCoil(0);
152 mb.addIsts(0);
153 mb.addHreg(0);
154 mb.addHreg(1);
155
156 // Coil inicializálása a belső logikai állapotból
157 mb.Coil(0, ledState);
158 }
```

Monitorizare WIFI, serviciu Modbus cu conexiune activă

```
164 void loop() {
165
166     // ===== WIFI FELÜGYELET =====
167     // Nem blokkoló hálózati állapotellenőrzés.
168     // Ha nincs kapcsolat, 5 percenként újrapróbál.
169
170     if (WiFi.status() != WL_CONNECTED) {
171
172         wifiConnected = false;
173
174         if (millis() - lastReconnectAttempt >= reconnectInterval) {
175             lastReconnectAttempt = millis();
176             connectToWiFi();
177         }
178     }
179     else {
180         wifiConnected = true;
181     }
182
183     // ===== WIFI HIBA LED KEZELÉS =====
184     // Offline állapotban 1 Hz villogás.
185     // Online állapotban LED kikapcsolva.
186
187     if (!wifiConnected) {
188
189         if (millis() - lastBlinkTime >= blinkInterval) {
190             lastBlinkTime = millis();
191             wifiLedState = !wifiLedState;
192             digitalWrite(WIFI_LED, wifiLedState);
193         }
194     }
195     else {
196         digitalWrite(WIFI_LED, LOW);
197         wifiLedState = false;
198     }
199
200     // ===== MODBUS KISZOLGÁLÁS =====
201     // Csak aktív hálózat esetén kezel TCP kéréseket.
202     if (wifiConnected) {
203         mb.task();
204     }
205
206     // =====
207     // ===== MODBUS - BELSŐ ÁLLAPOT SZINKRON =====
208     // =====
209     // Ha külső rendszer írja a Coil 0-t,
210     // a belső állapot frissül.
211
212     bool coilState = mb.Coil(0);
213
214     if (coilState != ledState) {
215         ledState = coilState;
216     }

```

Citire butoane, control stare LED și monitorizare de la distanță

```
218 // =====  
219 // ===== GOMB - BELSŐ ÁLLAPOT =====  
220 // =====  
221 // Fizikai gombnyomás esetén a belső állapot toggolódik,  
222 // majd a Coil regiszter tükrözi azt.  
223  
224 bool reading = digitalRead(BTN);  
225  
226 if (reading != lastReading) {  
227     lastChangeTime = millis();  
228 }  
229  
230 if ((millis() - lastChangeTime) > debounceTime) {  
231  
232     if (reading != stableState) {  
233         stableState = reading;  
234  
235         if (stableState == true) {  
236             ledState = !ledState;  
237             mb.Coil(0, ledState);  
238         }  
239     }  
240 }  
241  
242 lastReading = reading;  
243  
244 // =====  
245 // ===== FIZIKAI LED VEZÉRLÉS =====  
246 // =====  
247 // A fizikai kimenet mindig a belső állapotot követi.  
248  
249 digitalWrite(LED, ledState);
```

Mérés periódikus, monitorizare de la distanță a DHT11, raportare stare buton prin Modbus

```
251 // =====  
252 // ===== DHT IDŐZÍTETT OLVASÁS =====  
253 // =====  
254  
255 if (millis() - lastDHTread > 2000) {  
256  
257     lastDHTread = millis();  
258  
259     float h = dht.readHumidity();  
260     float t = dht.readTemperature();  
261  
262     if (!isnan(h) && !isnan(t)) {  
263         mb.Hreg(0, (int)(t * 10));  
264         mb.Hreg(1, (int)(h * 10));  
265         temp = t;  
266         hum = h;  
267     }  
268 }  
269  
270 // Gomb státusz jelentése Modbuson  
271 mb.Ists(0, stableState);
```

Afișaj LCD

```
273 // =====
274 // ===== LCD KIJELZÉS =====
275 // =====
276 // Első sor: Hőmérséklet + Hálózati állapot
277 // Második sor: Páratartalom + LED állapot
278
279 lcd.setCursor(0,0);
280 lcd.print("T:");
281 lcd.print(temp,1);
282 lcd.print((char)223);
283 lcd.print("C ");
284 lcd.print(wifiConnected ? "N-ON " : "N-OFF");
285
286 lcd.setCursor(0,1);
287 lcd.print("H:");
288 lcd.print(hum,1);
289 lcd.print("% ");
290 lcd.print(ledState ? "L-ON " : "L-OFF");
291
292 delay(10); // CPU terhelés csökkentése
293 }
```

Diagrama bloc a sistemului

Diagrama bloc ilustrează arhitectura stratificată, asemănătoare PLC-ului, a firmware-ului bazat pe ESP32. La nivelul superior se află interfața WiFi și comunicarea ModbusTCP, care sunt singurele responsabile pentru schimbul de date în rețea și gestionarea spațiului de registre. Centrul sistemului este Stratul de Stare, care stochează toate variabilele de stare interne - cum ar fi starea LED-urilor, temperatura măsurată, umiditatea și starea WiFi - ca o singură sursă de adevăr.

Stratul de intrare gestionează intrările fizice, în timp ce stratul de ieșire gestionează ieșirile, ambele bazându-se direct pe stratul de stare internă. Controlul rămâne în întregime pe ESP32, adică la nivel de câmp, astfel încât sistemul funcționează autonom, indiferent de rețea. Straturile Modbus și de supraveghere reflectă doar starea internă ca o umbră digitală, dar nu transportă logica de control. Această structură asigură funcționare deterministă, offline și arhitectura robustă tipică sistemelor industriale.

Firmware Architecture

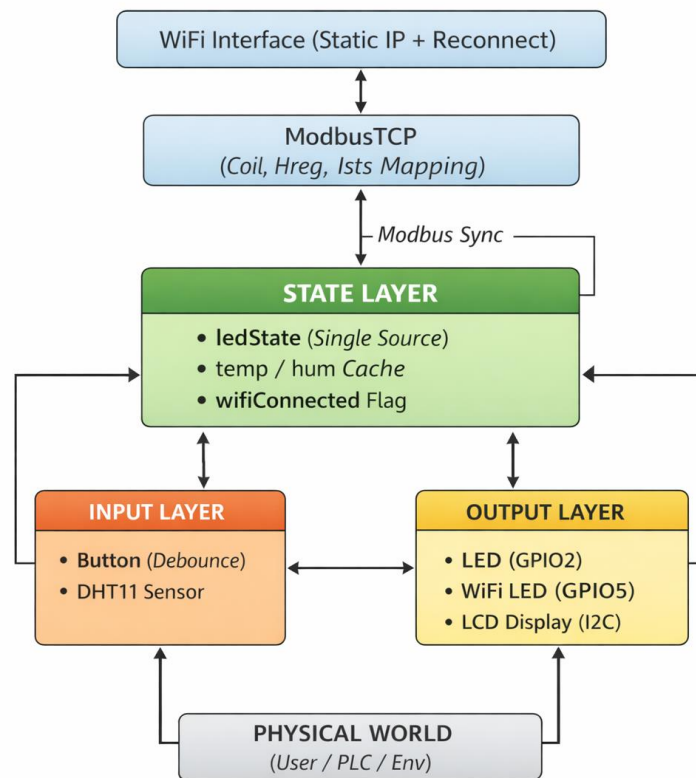


Figura 65: Diagrama bloc a sistemului (Sursa: generată de inteligența artificială)

6.7. Nivel de supraveghere - HMI bazat pe Python / ciclul de supraveghere

Nodul ESP32 de câmp pune la dispoziție starea senzorilor și actuatorilor prin intermediul registrelor Modbus. Nivelul superior nu este responsabil pentru preluarea controlului fizic, ci pentru cartografierea digitală și monitorizarea sistemului.

Acest strat din proiect este o aplicație ciclică, bazată pe Python, care, din punct de vedere al funcției sale, este un sistem de monitorizare de tip HMI/SCADA.

Sarcini:

- interogare ciclică a datelor de câmp
- stările de stocare și procesare
- oferirea de opțiuni de intervenție la distanță
- serviciu de afișare web

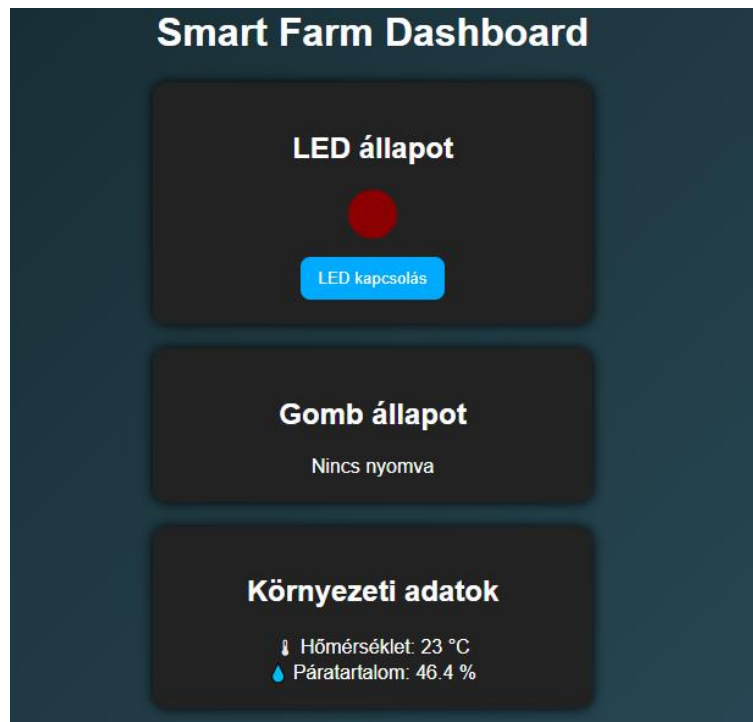


Figura 66: HMI bazat pe Python (Sursa: Editare proprie)

6.8. Pregătirea și implementarea unui sistem de monitorizare bazat pe Raspberry Pi

Următorul pas a fost rularea logicii de monitorizare software pe un hardware independent, cu consum redus de energie, care poate fi utilizat și în medii industriale. În acest scop **Zmeură Pi** Am folosit un computer bazat pe Raspberry Pi, care oferă o platformă ideală pentru aplicații bazate pe Python, ce includ atât comunicare în rețea, cât și o interfață web. Raspberry Pi este un computer relativ performant, cu un raport bun preț-performanță, care poate fi adaptat flexibil la o varietate de aplicații. În mod implicit, rulează o distribuție bazată pe Debian, Raspberry OS, care a îndeplinit pe deplin cerințele proiectului.



Figura 67: Raspberry Pi (Sursa: <https://malnapc.hu/raspberry-pi-5-4gb>)

Raspberry Pi în această arhitectură **ca strat de monitorizare și vizualizare** Funcționează ca un controler care comunică cu subsistemul bazat pe ESP32 prin protocolul Modbus TCP, oferind în același timp o interfață HMI bazată pe web pentru utilizatori.

Instalarea sistemului de operare și a sistemului de bază

Pregătirea dispozitivului a început cu instalarea unui sistem de operare stabil, bazat pe Linux, cu suport pe termen lung. Deși este posibilă instalarea unor sisteme de operare personalizate, sistemul de operare Raspberry Pi a oferit suportul hardware adecvat, stabilitatea sistemului și accesul facil la mediul de dezvoltare Python. Sistemul a fost instalat folosind programul oficial Raspberry Pi Imager, care, după specificarea setărilor de configurare de bază, le-a oferit studenților un mediu desktop gata preparat pentru lucrări ulterioare.



Figura 68: Instalarea sistemului de operare Raspberry Pi (Sursa: Editare proprie)

În timpul instalării sistemului, s-au întâmplat următoarele:

- configurați contul de utilizator implicit,
- actualizare de sistem,
- și verificarea conexiunii la rețea.

Funcționarea fiabilă a rețelei a fost de o importanță capitală, deoarece întreaga logică de monitorizare se bazează pe comunicarea IP.

Pregătirea runtime-ului și a dependențelor Python

Întrucât aplicația de monitorizare este scrisă în Python, pe Raspberry Pi a fost creat un mediu Python unificat și ușor de întreținut. Sistemul include Python 3.

```
tanulo@raspberrypi: ~  
Fájl Szerkesztés Lapok Súgó  
tanulo@raspberrypi:~$ python3  
Python 3.9.2 (default, Feb 28 2021, 17:03:44)  
[GCC 10.2.1 20210110] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> █
```

Versiunea 1.

În timpul pregătirii mediului, s-a pus accent pe:

- folosind managerul de pachete Python (PIP),
- instalarea bibliotecilor externe necesare (server web, client Modbus),
- și evitarea conflictelor de versiuni.

```
tanulo@raspberrypi:~$ pip install flask
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Requirement already satisfied: flask in /usr/lib/python3/dist-packages (1.1.2)
tanulo@raspberrypi:~$ pip install pymodbus
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting pymodbus
  Downloading https://www.piwheels.org/simple/pymodbus/pymodbus-3.8.6-py3-none-any.whl (164 kB)
    |#####| 164 kB 938 kB/s
Installing collected packages: pymodbus
  WARNING: The script pymodbus.simulator is installed in '/home/tanulo/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed pymodbus-3.8.6
tanulo@raspberrypi:~$
```

Acest pas a atras atenția studenților asupra reproductibilității și mentenabilității sistemelor software.

Configurarea rețelei și conexiunea Modbus TCP

Cerința de bază pentru ca sistemul să funcționeze este o conexiune Modbus TCP stabilă între Raspberry Pi și ESP32.

Pentru a realiza acest lucru, Raspberry Pi a funcționat cu o adresă IP fixă într-un mediu de rețea predictibil, astfel încât aplicația de monitorizare să poată ajunge întotdeauna la subsistem.

Studenții au înțeles că fiabilitatea comunicațiilor în rețea este crucială pentru un sistem de supraveghere.

Conexiunea Modbus în acest context nu a fost utilizată pentru control de nivel scăzut, ci pentru interogarea stării și transmiterea comenzilor, ceea ce se potrivește bine cu conceptul ciclului software PLC.

Lansarea aplicațiilor și gestionarea operațiunilor paralele

Când programul de monitorizare este pornit, logica pentru efectuarea actualizărilor ciclice de stare rulează pe un fir de execuție separat, în timp ce serverul web bazat pe Flask gestionează cererile utilizatorilor. Această soluție asigură că deservirea interfeței web nu blochează funcțiile de monitorizare ale sistemului.

Resursele Raspberry Pi s-au dovedit a fi suficiente:

- pentru comunicare Modbus continuă,
- pentru procesarea stărilor ciclice,
- și pentru a deservi HMI-ul web.

Acest model de operare a demonstrat bine studenților diferența dintre procesarea paralelă și operarea deterministă, dar care nu este în timp real.

Experiență în educație și integrare de sisteme

Dezvoltarea mediului de execuție bazat pe Raspberry Pi a fost parte integrantă a etapelor anterioare ale proiectului. Studenții au văzut un sistem complet în acțiune, în care senzorii și actuatorii apar la nivel fizic, comunicarea se face prin protocoale industriale, iar monitorizarea și controlul sunt implementate folosind instrumente software.

În plus, sistemul de operare bazat pe Debian le-a oferit studenților oportunitatea de a-și aprofunda cunoștințele despre Linux. Întrucât configurarea acestor dispozitive într-un mediu industrial nu se face neapărat printr-o interfață grafică, ne-am propus ca studenții să lucreze cât mai mult posibil prin intermediul unei interfețe de linie de comandă.

Totuși, în timpul implementării practice, eu și studenții am atins limitele hardware ale Raspberry Pi. Când întreaga echipă de studenți s-a conectat simultan la serverele web, a existat o încetinire a procesării datelor. În plus, sarcina suplimentară ar fi necesitat instalarea unui sistem de răcire activ, deoarece PC-ul cu cip s-a repornit de mai multe ori din cauza supraîncălzirii. Deși a îndeplinit nevoile proiectului, a servit ca o lecție importantă în timpul implementării, și anume că într-un sistem de nivel industrial, planificarea în avans a resurselor dispozitivului server și evaluarea încărcării preconizate ar trebui să fie o prioritate.

6.9. Rolul clientului Python

Sistemul de monitorizare ESP Modbus, programul Python, își propune să implementeze logica de monitorizare și „umbra” digitală.

Conectarea la ESP

```
1 # ===== FLASK + MODBUS TCP PLC RÉTEG =====
2 # Ez a program egy "szoftveres PLC ciklust" valósít meg.
3 # Feladata:
4 #   • ESP32 Modbus eszköz lekérdezése
5 #   • LED vezérlési logika kezelése
6 #   • Szenzoradatok továbbítása a webes HMI felé
7
8
9 # ===== MODULOK IMPORTÁLÁSA =====
10 from flask import Flask, render_template, jsonify
11 from pymodbus.client import ModbusTcpClient
12 import threading, time
13
14
15 # ===== WEB SZERVER OBJEKTUM =====
16 # Flask kezeli a HTTP kéréseket (HMI felület)
17 app = Flask(__name__)
18
19
20 # ===== MODBUS KLIENS (ESP32 FELÉ) =====
21 # Ez a sor létrehozza a TCP kapcsolatot az ESP Modbus felügyeleti rendszerével.
22 esp = ModbusTcpClient("192.168.0.200", port=502)
23
24
25 # ===== ÁLLAPOTVÁLTOZÓK LÉTREHOZÁSA =====
26 # Ezek a változók a rendszer állapotát tükrözik a szerver oldalon
27
28 led_state = False      # LED logikai állapot (kimenet)
29 btn_state = False      # Fizikai gomb állapota
30 prev_btn_state = False # Előző ciklus gomb állapota (élelérzékeléshez)
31
32 temperature = 0        # Hőmérséklet cache
33 humidity = 0           # Páratartalom cache
```

Structura ciclului de supraveghere

```
37 # ===== PLC CIKLUS (IPARI LOGIKA MINTAJÁRA) =====
38 # Ez a függvény a felügyeleti alkalmazás ciklikusan futó központi része.
39 # Bár a neve „felügyeleti ciklus”, funkcionálisan nem valós idejű vezérlési ciklust, hanem egy
    állapotfrissítő, felügyeleti lekérdezési ciklust valósít meg.
40 # Ez egy végtelen ciklus, amely fix periódusban (200 ms) végzi az I/O beolvasást és a vezérlési logikát.
41
42 def plc_cycle():
43     global led_state, btn_state, prev_btn_state, temperature, humidity
44
45     while True:
46
47         # ===== DIGITÁLIS BEMENET OLVASÁSA (FIZIKAI GOMB) =====
48         # Modbus Input Status regiszter 0
49         # Az ESP az aktuális gombállapotot a Discrete Input 0 címre írja.
50         rr_btn = esp.read_discrete_inputs(address=0, count=1)
51
52         if not rr_btn.isError():
53             btn_state = rr_btn.bits[0]
54
55
56         # ===== ÉLÉRZÉKELÉS (EDGE TRIGGER) =====
57         # Ez a PLC logika:
58         #   # ha a gomb éppen most lett lenyomva – LED állapot vált,
59         #   # a parancs visszakerül az ESP-re
60         # Ipari PLC logikában ez "RISING EDGE DETECT".
61         # Csak akkor reagálunk, amikor a gomb 0-1 állapotba vált
62
63         if btn_state and not prev_btn_state:
64             led_state = not led_state # LED állapot váltás
65             esp.write_coil(0, led_state) # Fizikai LED frissítés Modbuson
66
67         prev_btn_state = btn_state
68
69
70         # ===== ANALÓG ADATOK OLVASÁSA =====
71         # Holding Register 0 – hőmérséklet
72         # Holding Register 1 – páratartalom
73         # Az ESP tizedes pontossággal küldi az adatot.
74
75         rr_temp = esp.read_holding_registers(address=0, count=2)
76
77         if not rr_temp.isError():
78             temperature = rr_temp.registers[0] / 10.0
79             humidity = rr_temp.registers[1] / 10.0
80
81
82         # ===== PLC CIKLUS IDŐZÍTÉS =====
83         # 200 ms ciklusidő (ipari PLC mintára)
84         time.sleep(0.2)
```

Afișaj web (HMI)

```
88 # ===== WEB OLDAL =====
89 # A HMI felület betöltése
90
91 @app.route("/")
92 def index():
93     return render_template("index.html")
94
95
96 # ===== ÁLLAPOT API =====
97 # A webes felület innen kapja az aktuális rendszerállapotot
98
99 @app.route("/status")
100 def status():
101     return jsonify({
102         "led": led_state,           # LED állapot
103         "btn": btn_state,          # Gomb állapot
104         "temp": temperature,      # Hőmérséklet
105         "hum": humidity           # Páratartalom
106     })
107
108
109 # ===== VIRTUÁLIS GOMB (WEB HMI) =====
110 # Ez a fizikai gomb funkcióját utánozza a weboldalon keresztül
111
112 @app.route("/toggle_led", methods=["POST"])
113 def toggle_led():
114     global led_state
115
116     led_state = not led_state     # LED állapot váltás
117     esp.write_coil(0, led_state)  # Parancs küldése az ESP-nek
118
119     return "OK"
120
121
122
123 # ===== PROGRAM INDÍTÁS =====
124 if __name__ == "__main__":
125
126     # PLC ciklus külön szálon fut, hogy a web szerver ne blokkolódjon
127     threading.Thread(target=plc_cycle, daemon=True).start()
128
129     # Flask webservert indítása
130     app.run(host="0.0.0.0", port=5000)
131
```

6.10. Înregistrarea datelor senzorilor în baza de date SQL

Ca următor pas în dezvoltarea sistemului de monitorizare, ne-am stabilit obiectivul nu doar de a afișa datele senzorilor în timp real pe interfața web HMI, ci și de a le înregistra într-un format structurat, cu funcție de căutare. Pentru a realiza acest lucru, a fost creată o integrare de baze de date bazată pe SQL pentru a completa aplicația Python existentă.

Această dezvoltare a adăugat o nouă dimensiune proiectului, deoarece sistemul îndeplinește acum nu doar funcții de monitorizare, ci și funcții de colectare a datelor, permițând analiza ulterioară a acestora.

Rolul înregistrării datelor în sistemul de supraveghere

Salvarea datelor senzorilor într-o bază de date permite urmărirea modificărilor în timp, analiza pe termen lung a funcționării și procesarea ulterioară a datelor. Acest lucru le-a demonstrat clar studenților că, în sistemele reale, afișarea stării actuale nu este suficientă; stocarea istorică a datelor este o cerință fundamentală.

În timpul dezvoltării, accentul s-a pus pe înțelegerea faptului că colectarea datelor nu servește scopurilor de control continuu, ci mai degrabă creează o bază de suport pentru decizii și analiză.

Selectarea bazei de date și structura logică

A fost creată o structură simplă de baze de date, bazată pe SQL, pentru a se potrivi mediului educațional, care este ușor de utilizat, dar ilustrează clar principiile managementului datelor relaționale. Baza de date înregistrează datele măsurate într-un singur tabel, bine definit. Am folosit un motor de baze de date MariaDB (bazat pe MySQL) pentru baza de date.

Datele stocate sunt de obicei (**date senzoriale**masă):

- identificator de înregistrare (**ID date**, int, auto-incrementare, cheie primară)
- marcaj temporal (**timpul măsurării**, dată și oră, implicit marcaj_temporal_actual)
- valorile temperaturii (**ritm**, dublu),
- date despre umiditate (**zumzet**, dublu),

Această structură permite recuperarea cronologică a datelor și o prelucrare statistică simplă.

Completarea aplicației Python cu gestionarea bazelor de date

În timpul extinderii aplicației existente bazate pe Flask, gestionarea bazei de date a fost integrată ca o unitate logică separată. Datele senzorilor sunt încă citite ciclic prin comunicarea Modbus TCP, dar valorile actualizate sunt introduse și în baza de date.

Deși în implementarea actuală, datele sunt trimise continuu către baza de date, am atras atenția studenților asupra faptului că într-un sistem complex, diferite funcții operează cu criticități de timp diferite. Pe baza acestui fapt, următorul pas în dezvoltarea aplicației ar putea fi reducerea frecvenței de înregistrare a datelor la doar o dată la câteva secunde.

Pentru a extinde programul Python, a fost mai întâi necesar să importăm modulul mysql.connector.

```
9 # ===== MODULOK IMPORTÁLÁSA =====
10 from flask import Flask, render_template, jsonify
11 from pymodbus.client import ModbusTcpClient
12 import threading, time
13 import mysql.connector #Az SQL kapcsolatot kezelő modul importálása.
```

În pasul următor, am creat obiectele necesare pentru a gestiona conexiunea la baza de date și a trimite date.

```
26 # ===== ADATBÁZIS KAPCSOLAT KONFIGURÁLÁSA =====
27 # Az adatbázis kapcsolatot kezelő objektum létrehozása.
28 db = mysql.connector.connect(
29     host="localhost",
30     port="3306",
31     user="root",
32     password="",
33     database="sensordatabase"
34 )
35
36 #Az adstbázist kezelő cursor objektum létrehozása.
37 dbcursor = db.cursor()
```

În cele din urmă, la interogarea datelor senzorilor analogici, am extins codul prin trimiterea datelor către serverul bazei de date:

```
85 # ===== ANALÓG ADATOK OLVASÁSA =====
86 # Holding Register 0 – hőmérséklet
87 # Holding Register 1 – páratartalom
88 # Az ESP tizedes pontossággal küldi az adatot.
89
90 rr_temp = esp.read_holding_registers(address=0, count=2)
91
92 if not rr_temp.isError():
93     temperature = rr_temp.registers[0] / 10.0
94     humidity = rr_temp.registers[1] / 10.0
95     dbcursor.execute("INSERT INTO sensordata (temp,hum) VALUES (" + temperature + ","+humidity+")")
96
97
98 # ===== PLC CIKLUS IDŐZÍTÉS =====
99 # 200 ms ciklusidő (ipari PLC mintára)
100 time.sleep(0.2)
```

Semnificația educațională și experiențele studenților

Introducerea integrării bazelor de date SQL a extins semnificativ valoarea de învățare a proiectului. Studenții au fost introduși nu doar în gestionarea datelor în timp real, ci și în stocarea structurată a datelor și în utilizarea ulterioară.

În timpul dezvoltării, au evoluat următoarele:

- cunoștințe de bază despre gestionarea datelor și modelarea datelor,
- gândirea sistemică,
- și sensibilitate față de securitatea datelor și consecvența datelor.

Înregistrarea datelor senzoriale într-o bază de date s-a integrat astfel organic în ansamblul proiectului și a consolidat și mai mult ideea că baza sistemelor IT și industriale moderne este gestionarea conștientă și structurată a datelor.

6.11. Nivelul de gestionare web – rolul său în sistem

Aplicția Python oferă acces la date prin intermediul punctelor finale HTTP, în timp ce interfața HTML + JavaScript care rulează în browser se ocupă de afișare și interacțiunea utilizatorului.

Acest strat:

- nu controlează direct I/O-urile fizice
- afișează o umbră digitală a stării ESP-ului
- transmite comenzile utilizatorului către aplicația Python

Calea de date:



Bloc de stare LED

Acest buton nu controlează direct ESP-ul, ci sistemul de management Python.

```
11 <!-- ===== LED KÁRTYA ===== -->
12 <!-- Ez a blokk a LED állapotát mutatja és innen lehet vezérelni -->
13 <div class="card">
14   <h2>LED állapot</h2>
15
16   <!-- A LED vizuális visszajelzője
17       A színét a JavaScript módosítja:
18       zöld = világít
19       piros = nem világít -->
20   <div id="led" class="led"></div>
21
22   <!-- Virtuális kapcsoló gomb
23       onclick esemény → toggleLED() JS függvény fut le
24       Ez HTTP POST kérést küld a Flask szervernek (/toggle_led)
25       Ez a gomb nem közvetlenül az ESP-t vezérli, hanem a Python felügyeleti rendszert.-->
26   <button onclick="toggleLED()">LED kapcsolás</button>
27 </div>
```

Blocul de stare al butonului

```
30 <!-- ===== FIZIKAI GOMB ÁLLAPOT ===== -->
31 <!-- Ez CSAK kijelzés, nem vezérlő szerv -->
32 <div class="card">
33   <h2>Gomb állapot</h2>
34
35   <!-- A fizikai gomb állapotának szöveges kijelzése.
36       A Flask szerver PLC ciklusa frissíti.
37       "Nyomva" / "Nincs nyomva" szöveg jelenik meg -->
38   <div id="btn">Nincs nyomva</div>
39 </div>
40
```

Afișaj text al stării butonului fizic.

Bloc senzor

```
42 <!-- ===== KÖRNYEZETI ADATOK ===== -->
43 <!-- DHT11 szenzor által mért adatok -->
44 <div class="card">
45   <h2>Környezeti adatok</h2>
46
47   <!-- Hőmérséklet kijelzés
48       Ezek a DHT szenzor értékei számára fenntartott mezők.
49       A program ide írja be a /status API-ből érkező értéket. -->
50   <div>🌡 Hőmérséklet: <span id="temp">--</span> °C</div>
51
52   <!-- Páratartalom kijelzés -->
53   <div>💧 Páratartalom: <span id="hum">--</span> %</div>
54 </div>
```

Acestea sunt câmpuri rezervate valorilor senzorului DHT.

JavaScript - actualizare live

```
1 // ===== ÁLLAPOTFRISSÍTŐ FÜGGVÉNY =====
2 // Ez a függvény 200 ms-onként lefut.
3 // Feladata: lekérdezni a Flask szervertől az aktuális rendszerállapotot és frissíteni a weboldalon
  látható adatokat.
4
5 function update(){
6
7 // HTTP GET kérés a szerver /status végpontjára
8 // A szerver JSON formátumban küldi vissza az adatokat:
9 // { led: bool, btn: bool, temp: float, hum: float }
10 fetch("/status")
11
12 // A válasz átalakítása JSON objektummá
13 .then(r => r.json())
14
15 // A kapott adatok feldolgozása
16 .then(data => {
17
18 // ===== LED VIZUÁLIS ÁLLAPOT =====
19 // A LED kör színének beállítása:
20 // zöld ha világít (true), sötétpiros ha nem (false)
21 document.getElementById("led").style.background =
22 data.led ? "limegreen" : "darkred";
23
24
25 // ===== FIZIKAI GOMB ÁLLAPOT KIÍRÁS =====
26 // A Flask PLC ciklus által olvasott bemenet jelenik meg
27 document.getElementById("btn").innerText =
28 data.btn ? "Nyomva" : "Nincs nyomva";
29
30
31 // ===== HŐMÉRSÉKLET KIÍRÁS =====
32 // DHT11 szenzor adata az ESP - Modbus - Flask útvonalon
33 document.getElementById("temp").innerText = data.temp;
34
35
36 // ===== PÁRATARTALOM KIÍRÁS =====
37 document.getElementById("hum").innerText = data.hum;
38 });
39 }
```

JavaScript preia și actualizează aceste variabile.

Funcționarea butonului comutator LED

```
43 // ===== VIRTUÁLIS LED KAPCSOLÓ =====
44 // Ez a függvény akkor fut le, amikor a felhasználó megnyomja a weboldalon a "LED kapcsolás" gombot.
45
46 function toggleLED(){
47
48 // HTTP POST kérés küldése a szervernek
49 // A Flask oldalon ez a /toggle_led útvonalat hívja meg, ami megfordítja a LED állapotát és kiírja
  Modbus-on az ESP-re.
50 fetch("/toggle_led", { method: "POST" });
51
52 // Fontos: itt nincs válaszfeldolgozás, a következő update() ciklus már lekéri az új állapotot.
53 }
```

Proces:



Ciclul de actualizare

```
57 // ===== IDŐZÍTETT FRISSÍTÉS =====
58 // A böngésző 200 ms-onként automatikusan meghívja az update() függvényt.
59 // Ez kvázi PLC ciklus a weboldalon (HMI polling).
60 // Ez megfelel a Python programban lévő ciklus ütemének.
61
62 setInterval(update, 200);
```

Aceasta se potrivește cu rata de ciclu Python, astfel încât sistemul rămâne sincron.

Rolul stilului în sistem

CSS asigură că interfața web HMI:

- să fie lizibil chiar și în condiții de lumină slabă
- oferiți feedback vizual clar despre stări
- blocuri de informații separate (logica panoului HMI)

Această structură urmează abordarea minimalistă a SCADA / HMI (Interfața Om-Mașină) industrială.

CSS complet – comentat în detaliu

```
1  /* ===== OLDAL ALAPSTÍLUS ===== */
2  body {
3      font-family: Arial;
4      background: linear-gradient(135deg,#0f2027,#203a43,#2c5364);
5      color: white;
6      text-align: center;
7      margin: 0;
8  }
9
10 /* ===== KÁRTYA (PANEL) STÍLUS =====
11 Minden információs blokk ezt használja:
12 LED, Gomb, Szenzor adatok*/
13 .card {
14     background: #222;
15     margin: 20px auto;
16     padding: 20px;
17     width: 320px;
18     border-radius: 12px;
19     box-shadow: 0 0 10px #000;
20 }
21
22 /* ===== LED VISSZAJELZŐ KÖR =====
23 Ez a vizuális LED indikátor*/
24 .led {
25     width: 40px;
26     height: 40px;
27     margin: 15px auto;
28     border-radius: 50%;
29     background: darkred; /* Alapértelmezett állapot: LED kikapcsolva. A program dinamikusan felülírja:
30 limegreen = bekapcsolva */
31 }
32
33 /* ===== GOMB STÍLUS =====
34 Virtuális LED kapcsoló*/
35 button {
36     padding: 10px 15px;
37     border-radius: 8px;
38     border: none;
39     background: #00aaff;
40     color: white;
41     cursor: pointer;
42 }
```

Aceasta corespunde principiilor de afișare ale unui PLC real, ale unui HMI, ale unui terminal SCADA și ale unui panou de control industrial.

De ce este important acest lucru în educație?

Studentii nu numai că afișează date, dar învață și cum o interfață om-mașină (HMI) poate:

- sistem vizual funcțional
- unde culoarea, dimensiunea și aranjamentul au semnificație

Stratul web este acum complet:



6.12. Dezvoltarea interfeței HMI web de către studenți în lucrul în echipă

După implementarea funcțiilor de bază ale sistemului de monitorizare, următorul element al proiectului a fost dezvoltarea ulterioară a interfeței HMI (om-mașină) web. În această fază, studenții nu au mai implementat soluții predefinite, ci au dezvoltat interfața existentă bazată pe HTML și CSS pe baza propriilor idei și în cadrul muncii în echipă.

Scopul sarcinii a fost ca elevii să experimenteze cum o interfață funcțională, dar cu aspect de bază, poate fi făcută mai ușor de utilizat, mai clară și mai informativă din punct de vedere vizual, menținând în același timp stabilitatea funcțională a sistemului.

Munca în echipă și partajarea sarcinilor

Studenții au lucrat în grupuri mici, în care împărțirea sarcinilor a fost deliberată. Unii studenți s-au concentrat în principal pe transformarea structurii (aranjarea elementelor HTML, gruparea logică), în timp ce alții s-au concentrat pe rafinarea aspectului (culori, aspect, feedback vizual).

Această formă de organizare a muncii le-a permis studenților să experimenteze beneficiile și provocările dezvoltării comune, precum și să practice coordonarea și dezvoltarea pe baza muncii fiecăruia.

Îmbunătățirea funcțională și vizuală a interfeței

Interfața web, care a servit drept punct de plecare, includea deja afișarea stărilor de bază ale sistemului și a opțiunilor de control de bază. Studenții au dezvoltat în continuare această interfață:

- afișarea informațiilor a fost rearanjată,
- au extins gama de date afișate de la senzori, pregătindu-se pentru o dezvoltare ulterioară,
- a fost dezvoltat un feedback vizual mai clar,
- iar stilul de aspect a fost modificat în funcție de nevoile echipei lor.

În timpul modificărilor, s-a acordat o atenție deosebită clarității și ușurinței în utilizare, în special pentru a se asigura că interfața este ușor de interpretat într-un mediu de monitorizare din lumea reală.

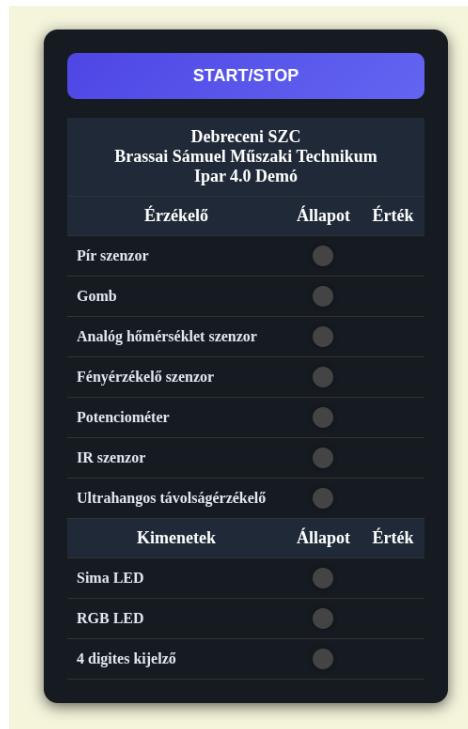


Figura 69: Lucrări îmbunătățite ale studenților 1. (Sursa: Editare proprie)

Apariția nevoilor individuale și a soluțiilor creative

În timpul procesului de dezvoltare, studenții au putut modela aspectul și funcționalitatea interfeței în funcție de propriile idei. Drept urmare, au fost create diverse soluții, de exemplu:

- codificare prin culori diferite pentru indicatorii de stare,
- machete alternative pentru afișarea datelor senzorilor,
- un feedback mai pronunțat în timpul intervențiilor.

Această libertate a oferit o oportunitate de dezvoltare a creativității, în timp ce studenții au învățat să echilibreze ideile individuale cu limitările tehnice și funcționale ale sistemului.

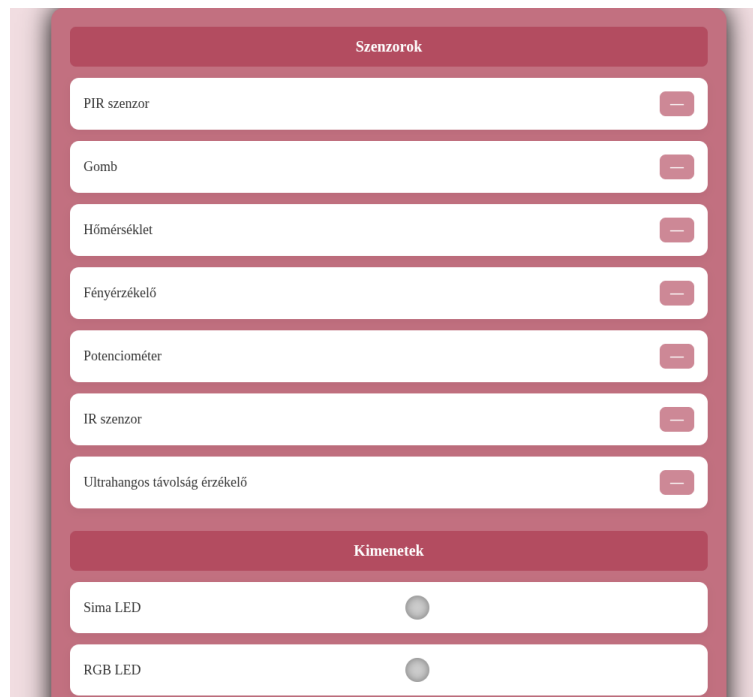


Figura 70: Lucrări îmbunătățite ale studenților 2. (Sursa: Editare proprie)

Semnificația educațională și experiențele de dezvoltare

În timpul dezvoltării ulterioare a interfeței web HMI, studenții au aplicat cunoștințele dobândite anterior într-un mod complex. Sarcina a dezvoltat simultan:

- competențe digitale și IT,
- abilități de cooperare și comunicare,
- și gândire centrată pe utilizator.

Procesul a fost bine aliniat cu obiectivele proiectului, deoarece studenții nu s-au limitat la crearea de soluții tehnice, ci au gândit interfața ca pe un sistem funcțional, având în vedere rolul acesteia în arhitectura generală de supraveghere.

6.13. Rezumat - integrarea sistemului și rezultatele învățării

Activitățile prezentate în acest capitol au constituit una dintre cele mai complexe și valoroase din punct de vedere pedagogic faze ale proiectului. În această fază, studenții au implementat un sistem de supraveghere complet și funcțional, în care elementele hardware, comunicarea în rețea, logica de control software și afișajul web au funcționat ca un sistem unificat.

Pregătirea mediului de execuție bazat pe Raspberry Pi și implementarea aplicației bazate pe Flask le-au oferit studenților oportunitatea de a-și aplica abilitățile de programare și operare a sistemului în medii IT și industriale reale. Utilizarea comunicării Modbus TCP a fost o experiență deosebit de importantă, deoarece studenții au putut

Au putut lucra cu un protocol la nivel de industrie și să înțeleagă rolul acestuia într-o arhitectură de supraveghere.

În timpul dezvoltării ulterioare a interfeței web HMI, studenții nu numai că au creat soluții tehnice, dar au stăpânit și o abordare centrată pe utilizator. În timpul dezvoltărilor efectuate în cadrul muncii în echipă, au apărut creativitatea individuală, luarea independentă a deciziilor și consultarea profesională comună. Personalizarea interfețelor a contribuit la sentimentul că sistemul este al lor și la asumarea responsabilității pentru funcționarea acestuia de către studenți.

Dintr-o perspectivă pedagogică, merită subliniat faptul că studenții au primit feedback continuu cu privire la consecințele deciziilor lor pe parcursul întregului proces de dezvoltare. Testarea funcționării sistemului, identificarea și corectarea erorilor și evaluarea în comun a soluțiilor au consolidat autoreflexia și gândirea orientată spre rezolvarea problemelor. Abordarea erorilor nu a fost văzută ca un eșec, ci ca o parte naturală a procesului de învățare.

Activitățile implementate în Capitolul 6 au contribuit la dezvoltarea mai multor competențe cheie, în special:

- competență digitală,
- abilități de colaborare și comunicare,
- gândirea algoritmică și sistemică,
- și în domeniile învățării independente și responsabilității.

Per total, acest capitol este un bun exemplu al modului în care o abordare bazată pe proiecte și orientată spre practică poate conecta cunoștințele teoretice cu un mediu de aplicare din lumea reală. Studenții nu numai că au creat un sistem tehnic funcțional, dar au dobândit și experiențe care vor pune bazele dezvoltării lor profesionale și ale orientării în carieră pe termen lung.

7. Rezumat

Documentul prezintă un proiect educațional complex, bazat pe Industria 4.0 și IoT, care se concentrează pe dezvoltarea unui model de casă inteligentă/fermă inteligentă. Conform filosofiei proiectului, învățarea nu este o receptare pasivă, ci un proces creativ activ, în care elevii ajung la soluții prin probleme reale. Accentul nu se pune pe răspunsuri predefinite, ci pe procesul de gândire, decizii și iterație. Elevii nu folosesc pur și simplu tehnologia, ci o proiectează, o analizează și o dezvoltă, experimentând în același timp că funcționarea unui sistem este întotdeauna rezultatul unor compromisuri și alegeri conștiente.

Prin urmare, obiectivele educaționale depășesc cunoștințele tehnice clasice. Proiectul dezvoltă simultan cunoștințe de programare și electronică, gândirea sistemică și abilități de colaborare. Studenții învață să interpreteze fenomenele lumii fizice în sistemele digitale, să recunoască relațiile cauză-efect și să fie capabili să gestioneze probleme complexe în mai multe etape. Este deosebit de important ca proiectul să ofere o oportunitate de a implica studenți cu medii diferite, ceea ce consolidează gândirea explicativă și rezolvarea comună a problemelor.

Dintr-o perspectivă tehnică, proiectul își propune să creeze un sistem bazat pe IoT în care senzorii, controlerile și actuatorii să funcționeze împreună ca un sistem logic unificat. Punctul de plecare este un kit Smart Farm fabricat în fabrică, care, însă, servește doar drept bază: sarcina studenților nu este de a-l reproduce, ci de a înțelege sistemul și apoi de a-l dezvolta în continuare către o arhitectură proprie, orientată spre industrie. Acest proces modelează bine funcționarea inginerescă din lumea reală, unde sisteme noi, optimizate, sunt create după analizarea soluțiilor existente.

Implementarea se realizează printr-un proces clar structurat, dar flexibil, cu etape cheie clar identificabile. Parcursul de învățare nu este liniar, ci urmează de obicei următorii pași:

- învățarea și asamblarea sistemului din fabrică (Smart Farm Kit),
- analiza funcționării componentelor individuale (senzori, actuatori, control),
- dezvoltarea limitelor și deficiențelor sistemului,
- definirea propriilor obiective și funcții de dezvoltare,
- proiectarea elementelor mecanice individuale și a pieselor de carcasă (de exemplu, modelare 3D),
- producția de elemente fizice (de exemplu, imprimare 3D),
- reproiectarea și extinderea sistemului electronic,
- programare (bazată pe blocuri → bazată pe text),
- integrarea subsistemelor într-un singur sistem,
- testare, depanare și dezvoltare iterativă.

Acest proces nu numai că are ca rezultat progres tehnic, dar îi învață și pe elevi că un sistem complex nu este niciodată „terminat”, ci este în continuă evoluție și rafinare.



Figura 71: Casă inteligentă (Sursa: Imagine proprie, fundal generat de inteligența artificială)

Dificultățile întâmpinate pe parcursul proiectului – lipsa conceptelor de bază, înțelegerea diferențelor dintre semnalele analogice și cele digitale și depanarea – s-au transformat în experiențe de învățare valoroase până la sfârșitul procesului. Problemele tehnice, cum ar fi datele de măsurare zgomotoase sau funcționarea instabilă din cauza unor valori limită necorespunzătoare, i-au ajutat pe studenți să recunoască faptul că comportamentul sistemului este o consecință directă a deciziilor de proiectare.

Până la sfârșitul proiectului, studenții au fost capabili să gândească în sistem, să interpreteze logica de procesare a datelor de intrare și de ieșire și să înțeleagă relația dintre lumea fizică și cea digitală. Soluțiile create erau deja sisteme funcționale, construite în mod conștient.

Cea mai importantă lecție este că proiectul nu numai că a oferit cunoștințe tehnice, ci a dezvoltat și o abordare care poate fi aplicată direct în mediul industrial și tehnologic modern.

Finanțat de Uniunea Europeană. Informațiile și declarațiile conținute aici reprezintă opiniile autorului/autorilor și nu reflectă neapărat opinia oficială a Uniunii Europene sau a Fundației Publice Tempus. Nici Uniunea Europeană, nici autoritatea finanțatoare nu pot fi trase la răspundere pentru acestea.