

Ipar 4.0 technológiák alkalmazása okosházak gyártásában és üzemeltetésében

Feladatgyűjtemény Smart Farm, ESP32, KidsBlock, Python, Raspberry Pi és Modbus TCP alapú projektoktatáshoz

2024-2-HU01-KA210-VET-000271388

Bevezető

Ez a feladatgyűjtemény a Smart Farm és okosház projekt oktatási feldolgozásához készült. A projekt középpontjában egy Ipar 4.0 szemléletű, hálózatba kapcsolt, adatvezérelt rendszer megismerése és fokozatos felépítése áll. A tanulók a feladatokon keresztül nem csupán az egyes eszközök működését sajátítják el, hanem azt is megértik, hogyan kapcsolódnak egymáshoz a szenzorok, aktuátorok, vezérlőegységek, hálózati kommunikációs megoldások és felügyeleti rendszerek egy összetett technológiai környezetben.

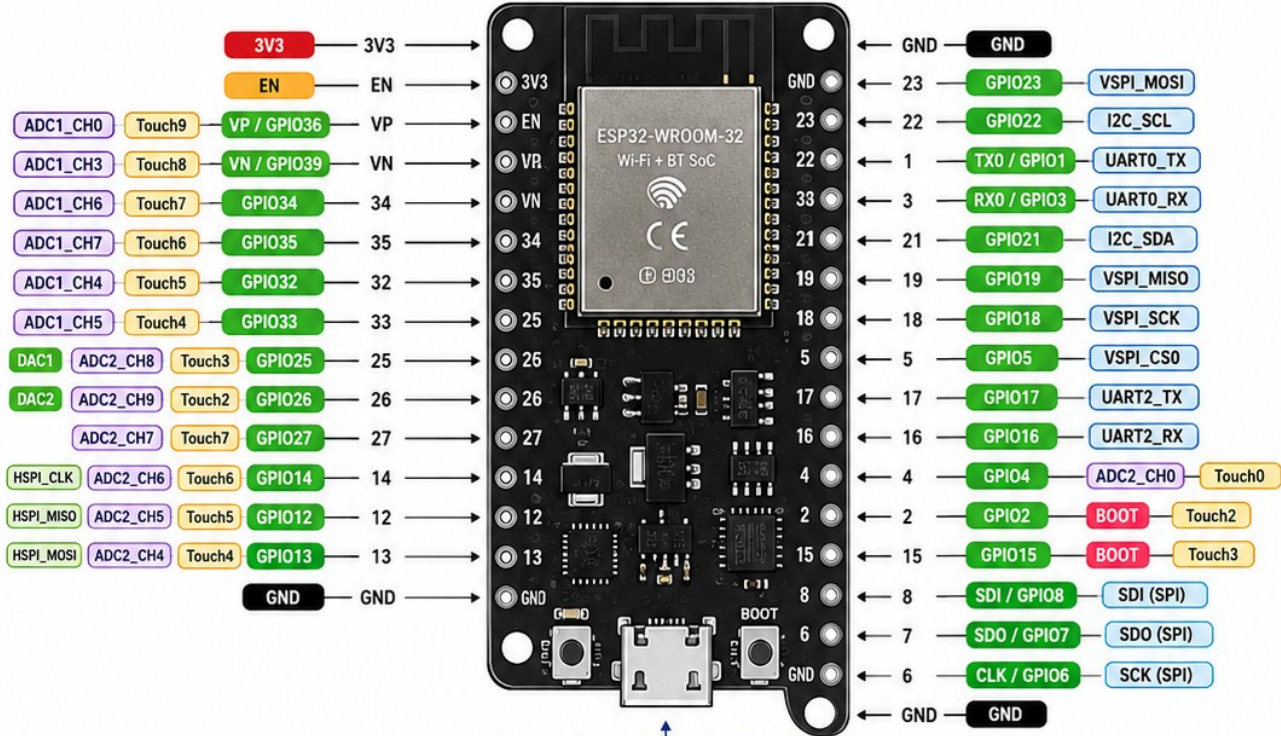
A feladatsor a Keystudio KidsBlock blokkprogramozási környezetéből indul ki, majd lépésről lépésre vezeti a tanulókat az ESP32 hardveres működésének, a WiFi és HTTP alapú IoT kommunikációnak, a Modbus TCP ipari adatcserének, valamint a Raspberry Pi / Python / Flask alapú webes HMI felügyeleti rendszernek a megértése felé. A cél, hogy a résztvevők ne elszigetelt gyakorlatokat oldjanak meg, hanem egy fokozatosan bővülő, valódi rendszer logikáját építsék fel, amelyben a fizikai és digitális világ működése egységes egészként jelenik meg.

A feladatok didaktikai íve

- Alapfogalmak és rendszerszemlélet: IoT, Ipar 4.0, szenzor, aktuátor, digitális és analóg jel.
- Hardverismeret: ESP32, GPIO, tápellátás, I2C, PWM, szenzorok és beavatkozók.
- KidsBlock programozás: egyszerű vezérlések, eseménykezelés, delay nélküli időzítés.
- Haladó automatizálás: hiszterézis, állapotgép, kézi/automata mód, prioritáskezelés.
- IoT kommunikáció: WiFi, webszerver, HTTP, webes megjelenítés.
- Ipari kommunikáció: Modbus TCP, coil, register, terepi és felügyeleti szint.
- Raspberry Pi és Python HMI: polling ciklus, JSON API, Flask, dashboard.
- Hibakeresés: mérési zaj, hálózati hibák, timeout, túlterhelés, instabil működés.
- Komplex projektfeladatok: mini okosház, smart greenhouse, digitális árnyék, többcsomópontos rendszer.

ESP32 lábkiosztás – áttekintő ábra

ESP32 DevKit (30 tűs) – leggyakoribb funkciók



Csatlakozások
Micro USB – tápellátás és programozás

JELMAGYARÁZAT

- Digitális GPIO Általános célú be-/kimenet (PWM képes)
- ADC bemenet 12 bites analóg-digítál átalakító (ADC1/ADC2)
- DAC kimenet Analóg kimenet (8 bites)
- Érintésérzékelő Kapacitív érintés érzékelés (Touch)
- UART / soros Soros kommunikáció (UART0, UART2)
- SPI / I2C funkciók SPI és I2C perifériák
- Tápellátás / föld Tápfeszültség (3,3 V) és föld (GND)
- Különleges / boot Vezérlő és boot (strap) lábak

i Az ESP32 minden jelének logikai szintje 3,3 V!

FONTOS TUDNIVALÓK

- ✓ Az ESP32 logikai szintje 3,3 V.
- ✓ A legtöbb GPIO bemenetként és kimenetként is használható.
- ✓ Egyes lábak csak bemenetként működnek (GPIO34, GPIO35, GPIO36, GPIO39).
- ✓ Az I2C (SDA, SCL) és SPI (MOSI, MISO, SCK, CS) sok esetben szabadon hozzárendelhető.
- ✓ Boot/strap lábak (GPIO0, GPIO2, GPIO12, GPIO15) állapota befolyásolja az indulást. Használatuknál körültekintés szükséges.
- ✓ PWM (LEDC) több GPIO lábón elérhető.
- ✓ Ne köss 5 V-ot a GPIO lábakra. Csak 3,3 V-ot tolerálnak!

TIPIKUS ALKALMAZÁSOK

- LED vezérlés**
Pl. állapotjelző, fényeffektek
- Nyomógomb / kapcsoló**
Felhasználói bemenet
- DHT szenzor**
Hőmérséklet és páratartalom
- I2C LCD kijelző**
Szöveg és adatok megjelenítése
- Relé modul**
Nagyfeszültségű eszközök vezérlése
- Szervó motor**
Pozicionálás és vezérlés
- Analóg érzékelők**
Fény, hőmérséklet, feszültség mérés

CSAK BEMENET LÁBAK

Láb	Megjegyzés
GPIO34	Csak bemenet (ADC1_CH6 / Touch7)
GPIO35	Csak bemenet (ADC1_CH7 / Touch6)
GPIO36 (VP)	Csak bemenet (ADC1_CH0 / Touch9)
GPIO39 (VN)	Csak bemenet (ADC1_CH3 / Touch8)

BOOT (STRAP) LÁBAK – INDULÁSI MÓD

Láb	Normál indítás	Megjegyzés
GPIO0	Magasan (1)	Alacsonyan: bootloader mód
GPIO2	Magasan (1)	Alacsonyan: indulási hiba
GPIO12	Alacsonyan (0)	Magasan: indulási hiba
GPIO15	Alacsonyan (0)	Magasan: indulási hiba

További részletekért lásd az ESP32 műszaki dokumentációját.

1. fejezet – IoT és Ipar 4.0 alapfogalmak

A fejezet célja a projekt fogalmi megalapozása. A tanulók itt értik meg, hogy az okosház vagy okosfarm nem önálló eszközök gyűjteménye, hanem adatokon, beavatkozásokon és hálózati kapcsolatokon alapuló rendszer.

1. feladat – IoT fogalma és példái

Nehézségi szint: alap | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Az IoT fogalmának értelmezése saját példákon keresztül.

Szükséges eszközök / szoftverek

- füzet
- internetelérés opcionálisan
- Smart Farm Kit bemutató elemei

Előismeret

A tanulók ismerjék a hálózatba kapcsolt eszközök alap gondolatát.

Feladatleírás

Fogalmazd meg saját szavaiddal, mit jelent az IoT. Hozz három példát okosházból, okosfarmból vagy ipari környezetből.

Munkalépések

1. Írd le, milyen adatot gyűjt az eszköz.
2. Írd le, mire használható ez az adat.
3. Döntsd el, történik-e automatikus beavatkozás.

Elvárt eredmény

A tanuló képes megkülönböztetni a hagyományos elektronikai eszközt az IoT eszköztől.

Továbbgondolási lehetőség

Keress olyan hétköznapi eszközt, amely IoT képessé tehető lenne.

Tanári megjegyzés / megoldási irány

Jó válaszban megjelenik: adatgyűjtés, hálózati kapcsolat, távoli elérés, automatizálás.

2. feladat – Szenzor vagy aktuátor?

Nehézségi szint: alap | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

A bemeneti és kimeneti eszközök szerepének felismerése.

Szükséges eszközök / szoftverek

- Smart Farm Kit
- DHT11
- LED
- PIR
- szervó
- pumpa

Előismeret

A tanulók ismerjék a bemenet és kimenet fogalmát.

Feladatléírás

Sorold be az eszközöket szenzor vagy aktuátor kategóriába, majd indokold meg a döntést.

Munkalépések

4. Készíts kétoszlopos táblázatot.
5. Minden eszközhöz írd egy mondatos magyarázatot.
6. Jelöld, hogy digitális vagy analóg jel kapcsolódhat-e hozzá.

Elvárt eredmény

A tanuló felismeri, hogy a szenzor mér, az aktuátor beavatkozik.

Továbbgondolási lehetőség

Adj hozzá saját példákat ipari környezetből.

Tanári megjegyzés / megoldási irány

DHT11: szenzor; LED: aktuátor; ultrahangos érzékelő: szenzor; vízpumpa: aktuátor; PIR: szenzor; szervó: aktuátor.

3. feladat – Ipar 4.0 rendszerlánc

Nehézségi szint: alap-közép | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

A teljes rendszerlánc megértése a fizikai világtól a HMI-ig.

Szükséges eszközök / szoftverek

- tábla
- füzet
- projektábra

Előismeret

A tanulók ismerjék a szenzor, vezérlő és beavatkozó fogalmát.

Feladatleírás

Rajzolj blokkdiagramot: szenzor → terepi vezérlő → kommunikáció → felügyeleti rendszer → HMI.

Munkalépések

7. Nevezd meg az egyes szinteket.
8. Írj minden szinthez egy konkrét projektbeli példát.
9. Jelöld nyilakkal az adat útját.

Elvárt eredmény

A tanuló képes a Smart Farm rendszert rétegzett architektúraként értelmezni.

Továbbgondolási lehetőség

Egészítsd ki a rajzot egy adatbázis vagy naplózó réteggel.

Tanári megjegyzés / megoldási irány

A jó megoldásban elkülönül a terepi szint és a felügyeleti szint.

4. feladat – Digitális iker és digitális árnyék

Nehézségi szint: közép | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

A digitális leképezés alapfogalmának megértése.

Szükséges eszközök / szoftverek

- füzet
- webes HMI példa
- ESP32 állapotadatok

Előismeret

A tanulók ismerjék az állapotadat fogalmát.

Feladatleírás

Magyarázd el, mi a különbség a fizikai rendszer és annak digitális állapotképe között.

Munkalépések

10. Válassz egy eszközt, például LED-et.
11. Írd le a fizikai állapotát.
12. Írd le, hogyan jelenhet meg ugyanez a webes felületen.

Elvárt eredmény

A tanuló megérti, hogy a HMI nem maga a folyamat, hanem annak digitális megjelenítése.

Továbbgondolási lehetőség

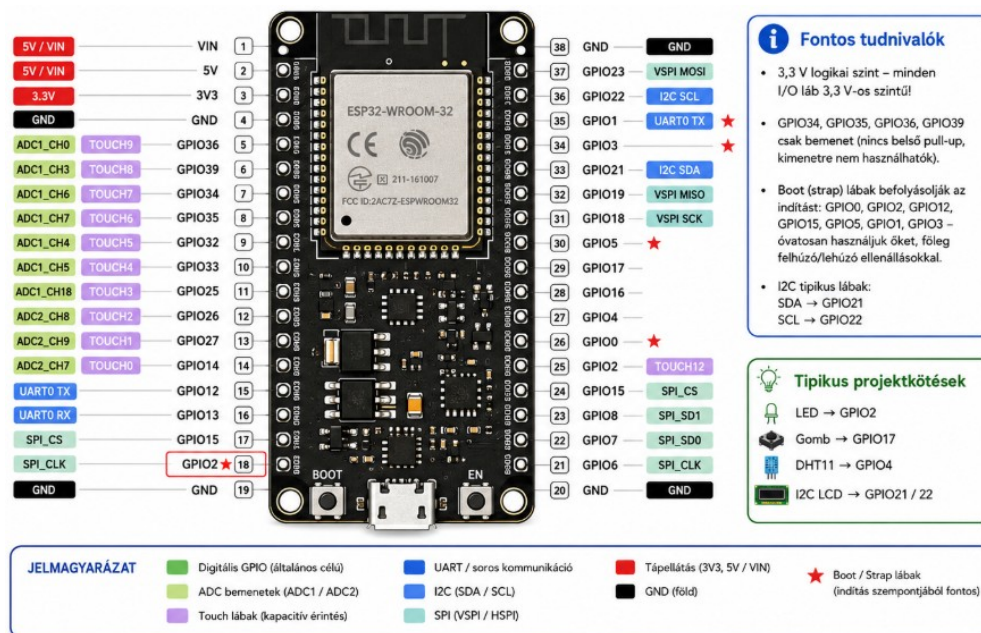
Beszélgétek meg, mikor lehet veszélyes, ha a digitális állapot eltér a valós állapottól.

Tanári megjegyzés / megoldási irány

A digitális árnyék az aktuális állapot leképezése; a digitális iker ennél több, modell- és szimulációs elemeket is tartalmazhat.

2. fejezet – ESP32 hardver és elektronikai alapok

Ebben a fejezetben a tanulók az ESP32-t terepi vezérlőcsomópontként értelmezik. A hangsúly nem azon van, hogy 'melyik lábra mit kötünk', hanem azon, hogyan lesz a fizikai jelből feldolgozható állapot.



5. feladat – ESP32 szerepe a Smart Farm rendszerben

Nehézségi szint: alap | Munkaforma: egyéni / páros / kiscsoportos

Cél

Az ESP32 terepi vezérlőegységként való értelmezése.

Szükséges eszközök / szoftverek

- ESP32 fejlesztőpanel
- Smart Farm Kit

Előismeret

Alapvető mikrokontroller fogalmak.

Feladatleírás

Írd le, milyen feladatokat lát el az ESP32 a Smart Farm rendszerben.

Munkalépések

13. Sorolj fel legalább három bemenetet.
14. Sorolj fel legalább három kimenetet.
15. Írd le, miért fontos a helyi működés.

Elvárt eredmény

A tanuló nem központi szerverként, hanem terepi I/O és lokális vezérlő egységként értelmezi az ESP32-t.

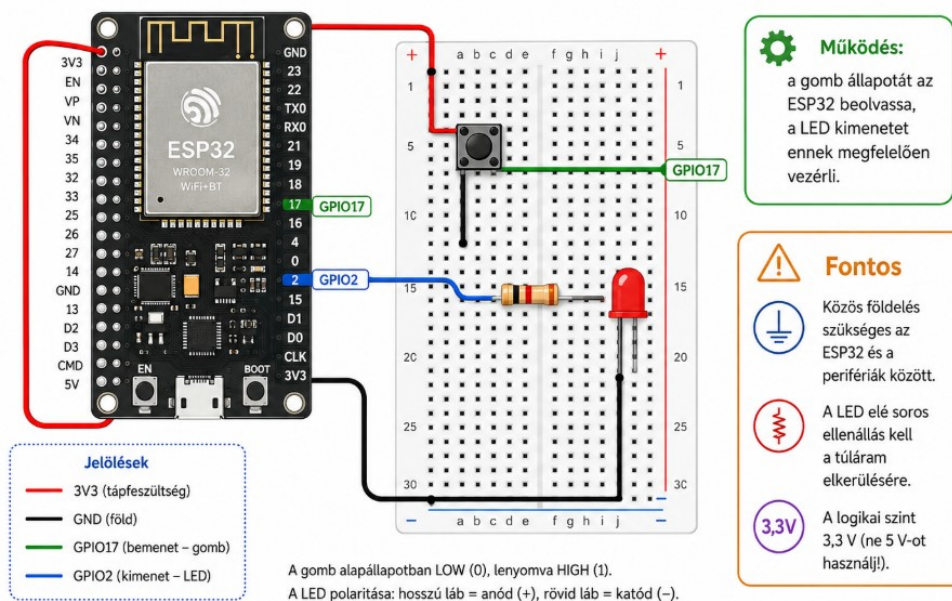
Továbbgondolási lehetőség

Hasonlítsd össze egy PLC távoli I/O moduljával.

Tanári megjegyzés / megoldási irány

Az ESP32 szenzoradatokat olvas, aktuátorokat vezérel, lokális logikát futtat és kommunikációt biztosít.

6. feladat - Digitális és analóg bemenetek



Nehézségi szint: alap | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

A digitális és analóg jelek közötti különbség megértése.

Szükséges eszközök / szoftverek

- nyomógomb
- PIR
- fotoellenállás

- talajnedvesség-szenzor

Előismeret

A tanulók ismerjék a HIGH/LOW fogalmát.

Feladatleírás

Adj példát digitális és analóg bemenetre, majd magyarázd el a különbséget.

Munkalépések

16. Mérd meg vagy olvasd ki a két jel értékét.
17. Figyeld meg, hogyan változik a jel.
18. Írd le, milyen döntés hozható belőle.

Elvárt eredmény

A tanuló megérti, hogy a digitális jel állapot, az analóg jel értéktartomány.

Továbbgondolási lehetőség

Készíts küszöbértékes döntést analóg jelből.

Tanári megjegyzés / megoldási irány

Digitális: nyomógomb, PIR. Analóg: fotoellenállás, talajnedvesség.

7. feladat – Kimenetek és beavatkozók

Nehézségi szint: alap | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

A kimeneti eszközök biztonságos és logikus vezérlésének megértése.

Szükséges eszközök / szoftverek

- LED
- buzzer
- relé
- ventilátor
- pumpa

Előismeret

Bemenet-kimenet logika.

Feladatleírás

Sorold fel, mely eszközök vezérelhetők közvetlenül és melyek igényelnek illesztést.

Munkalépések

19. Csoportosítsd az eszközöket teljesítmény szerint.
20. Jelöld, hol kell relé vagy meghajtó.
21. Indokold meg a választást.

Elvárt eredmény

A tanuló megérti, hogy a mikrokontroller nem minden terhelést hajthat meg közvetlenül.

Továbbgondolási lehetőség

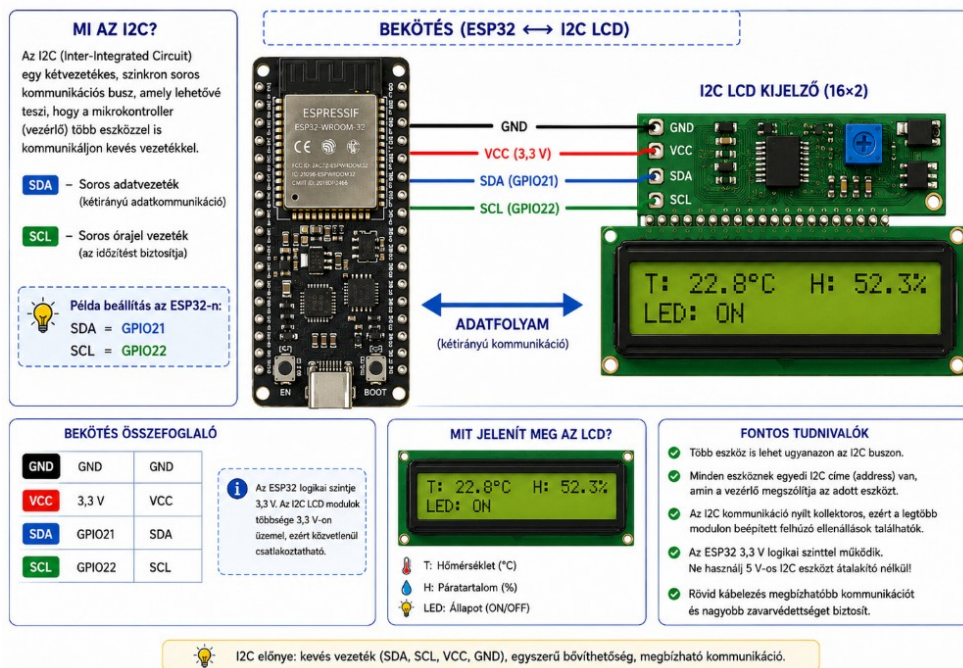
Keress példát motorvezérlő modulra.

Tanári megjegyzés / megoldási irány

LED kis áramú kimenet, pompa/ventilátor általában relét vagy tranzistoros illesztést igényel.

8. feladat – I2C LCD bekötése

I2C kommunikáció – ESP32 és 16×2 LCD kijelző



Nehézségi szint: közép | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Az I2C busz és a kijelző szerepének megértése.

Szükséges eszközök / szoftverek

- ESP32
- I2C 16x2 LCD
- vezetékek

Előismeret

Alapvető tápellátási és GPIO ismeretek.

Feladtleírás

Kösd be az LCD kijelzőt, majd jeleníts meg rajta két sor információt.

Munkalépések

22. Azonosítsd az SDA és SCL lábakat.
23. Ellenőrizd a tápellátást.
24. Írd ki az első sorba a rendszer nevét, a második sorba egy mért értéket.

Elvárt eredmény

Az LCD helyi visszajelzést ad, így a rendszer hálózat nélkül is értelmezhető.

Továbbgondolási lehetőség

Jelenítsd meg egyszerre a hőmérsékletet és a LED állapotát.

Tanári megjegyzés / megoldási irány

ESP32-n gyakori I2C lábkiosztás: SDA=21, SCL=22, de paneltől függően ellenőrizni kell.

3. fejezet – KidsBlock alapgyakorlatok

A KidsBlock feladatok célja, hogy a tanulók vizuális programozási környezetben értsék meg a vezérlési logikát. A blokkprogramozás nem a szakmai mélység csökkentése, hanem a logikai gondolkodás fokozatos felépítése.

9. feladat – LED villogtatás

Nehézségi szint: alap | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Első kimeneti vezérlés megvalósítása blokkprogrammal.

Szükséges eszközök / szoftverek

- KidsBlock
- ESP32
- LED

Előismeret

A tanulók ismerjék a digitális kimenet fogalmát.

Feladtleírás

Készíts blokkprogramot, amely 500 ms időközzel villogtatja a LED-et.

Munkalépések

- Állítsd be a LED kimeneti lábát.
- Kapcsold be a LED-et.
- Várj 500 ms-ot.
- Kapcsold ki a LED-et.
- Ismételd végtelen ciklusban.

Elvárt eredmény

A LED szabályos ütemben villog.

Továbbgondolási lehetőség

Próbáld ki 100 ms, 500 ms és 1000 ms értékekkel.

Tanári megjegyzés / megoldási irány

Ez a feladat a digitális kimenet vezérlésének alapja, de a wait/delay blokkolójellegét később tudatosítani kell.

10. feladat – Gombvezérelt LED

KidsBlock programlogika – gomb vezéri a LED-et

Egyszerű logika blokkokkal – érthetően gyerekeknek

A PROGRAM LÉPÉSEI

- Induláskor inicializálás**
Beállítjuk a LED kimenetet, és ki kapcsoljuk.
- Fő ciklus**
A program folyamatosan ismétlődik.
- Gomb állapot olvasása**
Megnézzük, lenyomva van-e a gomb.
- Ha gomb NYOMVA**
Akkor a LED **bekapcsol**.
- Különben**
Akkor a LED **kikapcsol**.

MIT TANULUNK EBBŐL?

Feltétel
Eldöntjük, mi történjen attól függően, hogy igaz vagy hamis.

Ciklus
A program újra és újra fut, amíg be van kapcsolva.

Digitális bemenet
A gomb két állapotú jel: NEM NYOMVA vagy NYOMVA.

Digitális kimenet
A LED két állapotú: BEKAPCSOLVA vagy KIKAPCSOLVA.

BLOKKPROGRAM – LOGIKA

Program indul

Inicializálás

Állítsd a LED kimenetet: LED → KIMENET

LED kikapcsolása

Ismételd örökké

Gomb állapot olvasása

gomb_nyomva = GOMB állapot

Ha gomb_nyomva = IGAZ akkor

LED bekapcsolása

különben

LED kikapcsolása

KAPCSOLÁSI VÁZLAT

GOMB (BEMENET)

LED (KIMENET)

ESP32

GPIO 19

GPIO 2

220 Ω

GND

BEMENET
Nyomógomb a GPIO 19-re kötve
A gomb lenyomva = IGAZ (1)
A gomb felengedve = HAMIS (0)

KIMENET
LED a GPIO 2-re kötve
LED BE = bekapcsolva (1)
LED KI = kikapcsolva (0)

ÖSSZEFOGLALÁS: A program folyamatosan figyel a gombot. Ha megnyomod, a LED világít, ha elengeded, a LED kialszik.

Nehézségi szint: alap | Munkaforma: egyéni / páros / kiscsoportos

Cél

Bemenet alapján történő kimenetvezérlés kialakítása.

Szükséges eszközök / szoftverek

- KidsBlock
- nyomógomb
- LED

Előismeret

Digitális bemenet és kimenet ismerete.

Feladatleírás

Készíts programot, amely gombnyomásra bekapcsolja a LED-et, elengedéskor kikapcsolja.

Munkalépések

30. Olvasd be a gomb állapotát.
31. Ha a gomb aktív, LED ON.
32. Különben LED OFF.
33. Teszteld több gombnyomással.

Elvárt eredmény

A LED a gomb állapotát követi.

Továbbgondolási lehetőség

Alakítsd át úgy, hogy gombnyomásra váltson állapotot.

Tanári megjegyzés / megoldási irány

A feladat alkalmas az állapot és esemény különbségének bevezetésére.

11. feladat – Toggle működés gombbal

Nehézségi szint: közép | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Állapotváltozó használata gombnyomásra.

Szükséges eszközök / szoftverek

- KidsBlock
- nyomógomb
- LED

Előismeret

A tanulók ismerjék a változó fogalmát.

Feladatleírás

Készíts programot, amely minden gombnyomásra átváltja a LED állapotát.

Munkalépések

34. Hozz létre ledAllapot változót.
35. Gombnyomáskor fordítsd meg az értékét.
36. A LED állapota kövesse a változót.
37. Figyeld meg a prellégés hatását.

Elvárt eredmény

A LED nem a gomb nyomva tartását, hanem a gombnyomási eseményeket követi.

Továbbgondolási lehetőség

Adj hozzá egyszerű prellégéskezelést időzítéssel.

Tanári megjegyzés / megoldási irány

Fontos tanári pont: a fizikai gomb nem ideális jel, a prellégés valós mérnöki probléma.

12. feladat - Delay nélküli villogás

Nehézségi szint: közép | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Nem blokkoló időzítési logika megértése.

Szükséges eszközök / szoftverek

- KidsBlock
- LED
- idő / millis blokk

Előismeret

Változók és feltételek használata.

Feladatleírás

Készíts villogó LED programot wait/delay nélkül, időbélyegek összehasonlításával.

Munkalépések

38. Hozz létre elozoldo változót.
39. Olvasd az aktuális időt.
40. Ha eltelt 500 ms, válts LED állapotot.
41. Frissítsd az elozoldo változót.

Elvárt eredmény

A LED villog, miközben a program nem áll meg.

Továbbgondolási lehetőség

Kombináld gombnyomás érzékeléssel.

Tanári megjegyzés / megoldási irány

Ez ipari szemléletű időzítés: a ciklus folyamatosan fut, a feladatok nem blokkolják egymást.

13. feladat - Kétsebességű villogás

Nehézségi szint: közép | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Állapotváltozó és időzítési paraméter együttes használata.

Szükséges eszközök / szoftverek

- KidsBlock
- LED
- gomb

Előismeret

Delay nélküli időzítés alapjai.

Feladatléírás

Készíts programot: első gombnyomás lassú villogás, második gyors villogás, harmadik kikapcsolás.

Munkalépések

42. Hozz létre mod változót 0–2 értékekkel.
43. Gombnyomáskor növeld a módot.
44. Mód alapján állíts intervallumot.
45. Időzítéssel villogtasd a LED-et.

Elvárt eredmény

Három működési mód stabilan váltakozik.

Továbbgondolási lehetőség

Jelezd a módot LCD-n is.

Tanári megjegyzés / megoldási irány

A feladat bevezeti az állapotgép gondolkodást egyszerű formában.

4. fejezet – Szenzoros automatizálási feladatok

A fejezetben a tanulók már nem csak egy bemenetre reagálnak, hanem mért értékek alapján hoznak vezérlési döntést. Itt jelenik meg a küszöbérték, hiszterézis és kombinált logika.

14. feladat – Fényvezérelt világítás

Nehézségi szint: alap-közép | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Analóg jel küszöbértékes feldolgozása.

Szükséges eszközök / szoftverek

- fotoellenállás
- LED
- KidsBlock

Előismeret

Analóg bemenet olvasása.

Feladatleírás

Készíts programot, amely sötétben bekapcsolja a LED-et, világosban kikapcsolja.

Munkalépések

46. Olvasd be a fotoellenállás értékét.
47. Válassz küszöbértéket.
48. Ha az érték sötétet jelez, LED ON.
49. Világosban LED OFF.

Elvárt eredmény

A világítás a fényviszonyok alapján automatikusan működik.

Továbbgondolási lehetőség

Készíts három zónát: világos, félhomály, sötét.

Tanári megjegyzés / megoldási irány

A küszöbérték környezetfüggő, kalibrálni kell.

15. feladat – Háromzónás fényjelzés

Nehézségi szint: közép | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Több küszöbérték alapján működő állapotlogika.

Szükséges eszközök / szoftverek

- fotoellenállás
- LED

Előismeret

Fényvezérelt világítás ismerete.

Feladatleírás

Készíts programot: világosban LED kikapcsol, félhomályban lassan villog, sötétben gyorsan villog.

Munkalépések

50. Mérd legalább három fényértéket.
51. Állíts be két határértéket.
52. Rendelj módot minden tartományhoz.
53. Használj delay nélküli villogást.

Elvárt eredmény

A LED jelzése a fényerő tartományától függ.

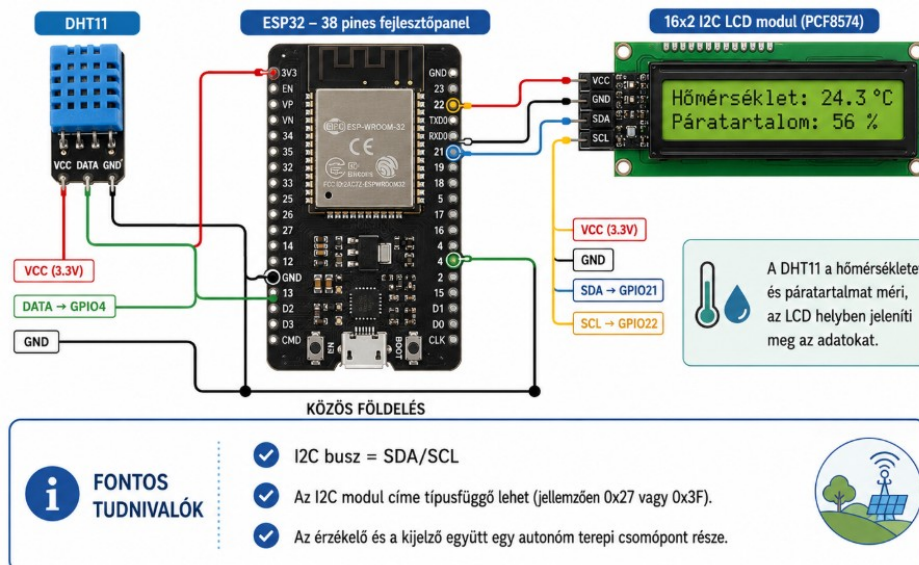
Továbbgondolási lehetőség

Jelenítsd meg LCD-n az aktuális zónát.

Tanári megjegyzés / megoldási irány

A feladat jól mutatja az analóg jel digitalizált döntési logikává alakítását.

16. feladat - DHT11 hőmérséklet- és páratartalom-mérés



Nehézségi szint: alap-közép | Munkaforma: egyéni / páros / kiscsoportos

Cél

Környezeti adatok mérésének és értelmezésének gyakorlása.

Szükséges eszközök / szoftverek

- DHT11
- ESP32
- KidsBlock

Előismeret

Szenzor fogalma.

Feladtleírás

Olvasd ki a DHT11 hőmérséklet- és páratartalom-értékeit.

Munkalépések

54. Illeszd a DHT11 modult.
55. Olvasd ki a hőmérsékletet.
56. Olvasd ki a páratartalmat.
57. Írd ki soros monitorra vagy LCD-re.

Elvárt eredmény

A tanuló két környezeti adatot tud megjeleníteni.

Továbbgondolási lehetőség

Adj hozzá riasztást 30 °C felett.

Tanári megjegyzés / megoldási irány

A DHT11 lassú szenzor, túl gyakori olvasása hibás értéket adhat.

17. feladat - Hiszterézises ventilátorvezérlés

Nehézségi szint: közép | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Instabil ki-be kapcsolás elkerülése hiszterézissel.

Szükséges eszközök / szoftverek

- DHT11
- ventilátor vagy LED szimuláció
- KidsBlock

Előismeret

Hőmérsékletmérés és feltételkezelés.

Feladatleírás

Készíts programot: ventilátor 28 °C felett indul, 25 °C alatt áll le.

Munkalépések

58. Olvasd be a hőmérsékletet.
59. Ha $T > 28$, ventilátor ON.
60. Ha $T < 25$, ventilátor OFF.
61. Köztes tartományban tartsd meg az előző állapotot.

Elvárt eredmény

A ventilátor nem kapcsolgat bizonytalanul a határérték körül.

Továbbgondolási lehetőség

Tedd be kézi felülbírási módba gombbal.

Tanári megjegyzés / megoldási irány

A hiszterézis ipari automatizálásban alapvető stabilizáló megoldás.

18. feladat – Intelligens öntözés

Nehézségi szint: közép | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Talajnedvesség alapján történő beavatkozás kialakítása.

Szükséges eszközök / szoftverek

- talajnedvesség-szenzor
- pumpa vagy LED szimuláció
- relé modul

Előismeret

Analóg jel, küszöbérték.

Feladatleírás

Készíts programot: száraz talajnál pumpa indul, nedves talajnál leáll.

Munkalépések

62. Olvasd a talajnedvesség értékét.
63. Kalibrálj száraz és nedves állapotot.
64. Állíts be küszöböt.

65. Kapcsold a pumpát a döntés alapján.

Elvárt eredmény

A rendszer a mért talajállapot alapján avatkozik be.

Továbbgondolási lehetőség

Adj hozzá maximális öntözési időt.

Tanári megjegyzés / megoldási irány

Valós pumpánál figyelni kell a tápellátásra és a relés illesztésre.

19. feladat – PIR alapú riasztás

Nehézségi szint: alap-közép | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Mozgásérzékelés és riasztási kimenet kialakítása.

Szükséges eszközök / szoftverek

- PIR
- LED
- buzzer

Előismeret

Digitális bemenet ismerete.

Feladatléírás

Mozgás esetén villogjon a LED és szóljon a buzzer, mozgás hiányában álljon le.

Munkalépések

66. Olvasd a PIR állapotát.
67. Mozgás esetén riasztás=IGAZ.
68. Riasztáskor LED villog és buzzer ON.
69. Mozgás hiányában minden OFF.

Elvárt eredmény

A rendszer mozgásra reagál és jelzést ad.

Továbbgondolási lehetőség

Adj hozzá 5 másodperces utójelzést delay nélkül.

Tanári megjegyzés / megoldási irány

A PIR érzékelők indítás után bemelegedési időt igényelhetnek.

20. feladat - Ultrahangos távolságriasztó

Nehézségi szint: közép | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Távolságmérés és határértékes riasztás.

Szükséges eszközök / szoftverek

- ultrahangos érzékelő
- LED
- buzzer

Előismeret

Digitális mérési eredmény értelmezése.

Feladtleírás

Készíts programot: 10 cm alatt LED villogjon, 10 cm felett kapcsoljon ki.

Munkalépések

70. Olvasd ki a távolságot.
71. Állíts határértéket.
72. Határ alatt indíts riasztást.
73. Határ felett állítsd le.

Elvárt eredmény

A tanuló megérti a távolságmérés és riasztási logika kapcsolatát.

Továbbgondolási lehetőség

Készíts három zónás jelzést távolság alapján.

Tanári megjegyzés / megoldási irány

Az ultrahangos mérés pontosságát befolyásolja a felület és a mérési szög.

21. feladat - LCD adatmegjelenítés

Nehézségi szint: közép | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Helyi HMI jellegű visszajelzés kialakítása.

Szükséges eszközök / szoftverek

- I2C LCD
- DHT11
- ESP32

Előismeret

I2C LCD alapok.

Feladatleírás

Jelenítsd meg LCD-n a hőmérsékletet, páratartalmat és egy rendszerállapotot.

Munkalépések

74. Első sor: T és H érték.
75. Második sor: LED vagy üzemmód állapot.
76. Frissítsd időzítve.
77. Kerüld a felesleges villogó újraírást.

Elvárt eredmény

A kijelző valós idejű helyi állapotképet ad.

Továbbgondolási lehetőség

Jelenítsd meg a WiFi kapcsolat állapotát is.

Tanári megjegyzés / megoldási irány

A helyi kijelzés fontos, mert a terepi egység hálózat nélkül is értelmezhető marad.

5. fejezet – Haladó vezérlési logikák

22. feladat – Kézi és automata mód

Nehézségi szint: közép-haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Üzemmódváltás kialakítása egy vezérlőrendszerben.

Szükséges eszközök / szoftverek

- gomb
- DHT11
- ventilátor
- LED

Előismeret

Alap szenzoros vezérlés.

Feladtleírás

Készíts rendszert, amely automata módban hőmérséklet alapján vezérel, kézi módban gombbal kapcsolható.

Munkalépések

78. Hozz létre uzemmod változót.
79. Gombbal válts kézi és automata mód között.
80. Automata módban DHT alapján dönts.
81. Kézi módban a felhasználó parancsa érvényesüljön.

Elvárt eredmény

A rendszer kétféle üzemben stabilan működik.

Továbbgondolási lehetőség

Adj LCD kijelzést az aktuális módról.

Tanári megjegyzés / megoldási irány

A kézi/automata mód ipari rendszerekben alapvető kezelési logika.

23. feladat – Prioritáskezelés riasztás esetén

Nehézségi szint: haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Több feltétel közötti prioritási sorrend megértése.

Szükséges eszközök / szoftverek

- PIR
- DHT11
- LED
- buzzer

Előismeret

Feltételes logika.

Feladtleírás

Készíts programot, ahol riasztás esetén a biztonsági jelzés felülírja a normál világításvezérlést.

Munkalépések

82. Hozz létre riasztás változót.
83. Mozgás esetén riasztás aktív.
84. Riasztáskor LED gyorsan villog és buzzer szól.
85. Ha nincs riasztás, normál fényvezérlés működik.

Elvárt eredmény

A biztonsági funkció elsőbbséget kap.

Továbbgondolási lehetőség

Adj hozzá riasztás törlését gombbal.

Tanári megjegyzés / megoldási irány

A prioritáskezelés megakadályozza, hogy két logika ellentmondó parancsot adjon.

24. feladat – Állapotgép négy üzemmóddal

Nehézségi szint: haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Állapotgép szemléletének kialakítása.

Szükséges eszközök / szoftverek

- gomb
- LED
- ventilátor/pumpa szimuláció

Előismeret

Változók és események.

Feladatléírás

Készíts négyállapotú rendszert: kikapcsolt, világítás, ventilátor, öntözés.

Munkalépések

86. Hozz létre állapot változót.
87. Gombnyomáskor lépj a következő állapotba.
88. Minden állapothoz rendelj kimeneti viselkedést.
89. Jelenítsd meg az állapotot LCD-n.

Elvárt eredmény

A rendszer egyértelmű állapotok mentén működik.

Továbbgondolási lehetőség

Készíts állapotdiagramot.

Tanári megjegyzés / megoldási irány

Az állapotgép később PLC és ipari vezérlési logikák megértéséhez is alap.

25. feladat – Energiatakarékos mód

Nehézségi szint: közép-haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Időalapú automatikus kikapcsolás megvalósítása.

Szükséges eszközök / szoftverek

- PIR
- LED
- ventilátor/pumpa szimuláció

Előismeret

Millis alapú időzítés.

Feladatléírás

Ha 30 másodpercig nincs mozgás, minden kimenet kapcsoljon ki.

Munkalépések

90. Mozgás esetén mentsd az utolsó mozgás időpontját.
91. Folyamatosan ellenőrizd az eltelt időt.
92. Ha túl sok idő telt el, kimenetek OFF.
93. Mozgáskor állítsd vissza az aktív működést.

Elvárt eredmény

A rendszer energiatakarékos módba lép inaktivitás esetén.

Továbbgondolási lehetőség

Állíts be éjszakai módot is.

Tanári megjegyzés / megoldási irány

A feladat környezettudatos és ipari energiamedzsment szemléletet is fejleszt.

26. feladat – Mini okosház kombinált logika

Nehézségi szint: haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Több alrendszer együttes kezelése.

Szükséges eszközök / szoftverek

- PIR

- DHT11
- fotoellenállás
- talajnedvesség
- LCD
- LED/pumpa/ventilátor

Előismeret

Korábbi szenzoros feladatok.

Feladatleírás

Készíts teljes mini okosház logikát: PIR világítás, DHT ventilátor, talajnedvesség öntözés, LCD kijelzés.

Munkalépések

94. Bontsd fel a programot.
95. Készíts külön blokkot a világításnak.
96. Készíts külön blokkot a hőmérsékletnek.
97. Készíts külön blokkot az öntözésnek.
98. Frissítsd az LCD-t.

Elvárt eredmény

A rendszer több funkcionális egységből felépített, működő modell.

Továbbgondolási lehetőség

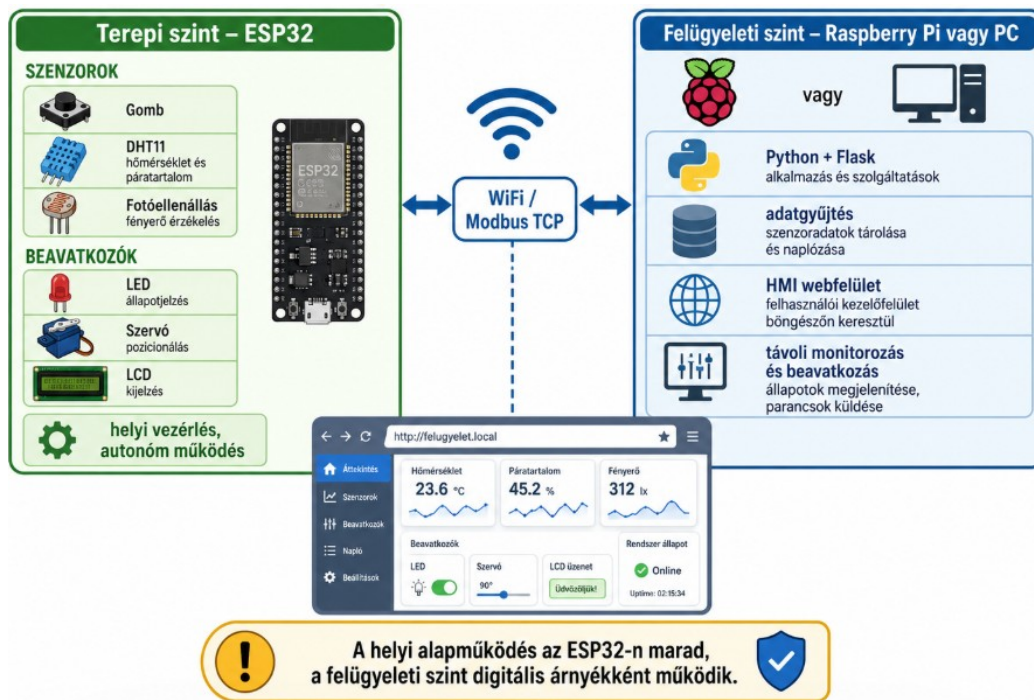
Adj hozzá kézi/automata módot.

Tanári megjegyzés / megoldási irány

A komplex programnál a strukturálás fontosabb, mint az egyes blokkok gyors összerakása.

6. fejezet – WiFi, HTTP és webes IoT feladatok

27. feladat – WiFi csatlakozás



Nehézségi szint: közép | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

ESP32 hálózati kapcsolatának létrehozása.

Szükséges eszközök / szoftverek

- ESP32
- WiFi hálózat
- KidsBlock vagy Arduino IDE

Előismeret

Alap programfeltöltés.

Feladatleírás

Készíts programot, amely csatlakozik a helyi WiFi hálózathoz, majd kiírja az IP-címet.

Munkalépések

99. Add meg az SSID-t.
100. Add meg a jelszót.
101. Indíts csatlakozást.
102. Írd ki vagy jelenítsd meg az IP-címet.

Elvart eredmény

Az ESP32 elérhető a helyi hálózaton.

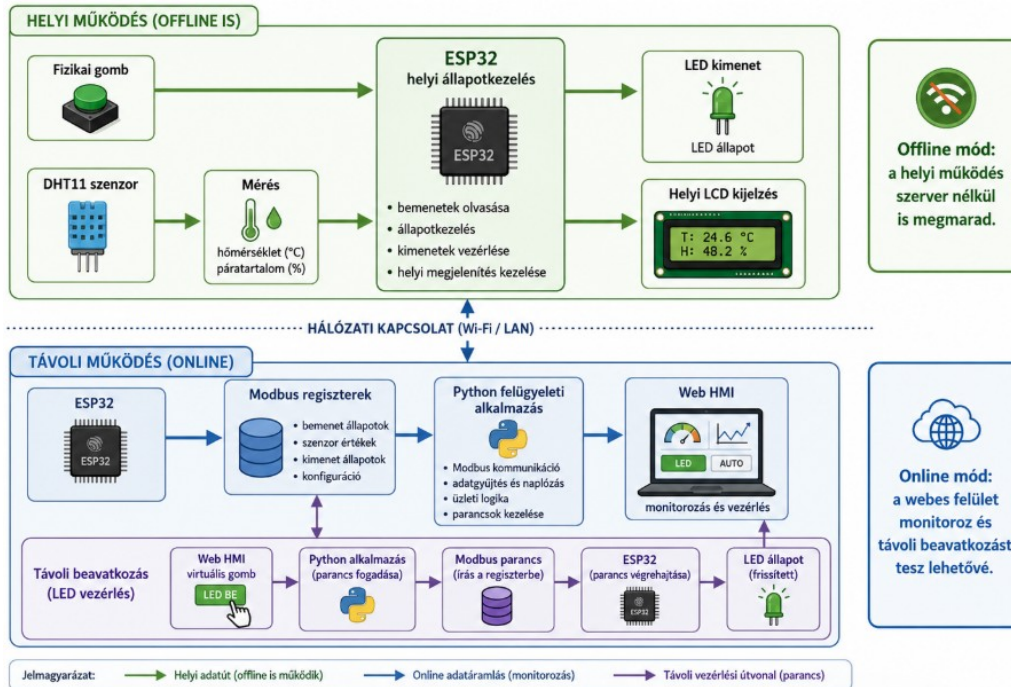
Továbbgondolási lehetőség

Jelenítsd meg az IP-t LCD-n.

Tanári megjegyzés / megoldási irány

A stabil IP-cím később a felügyeleti rendszer alapfeltétele.

28. feladat - Egyszerű ESP webszerver



Nehézségi szint: közép-haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Webes állapotmegjelenítés alapjainak megértése.

Szükséges eszközök / szoftverek

- ESP32
- WiFi
- böngésző

Előismeret

WiFi csatlakozás.

Feladatleírás

Készíts egyszerű weboldalt, amely megjeleníti a LED állapotát.

Munkalépések

103. Indíts webszervert.

104. Hozz létre főoldalt.
105. Olvasd a LED állapotát.
106. Jelenítsd meg szövegesen.

Elvárt eredmény

A böngészőben látható a rendszer állapota.

Továbbgondolási lehetőség

Adj hozzá frissítést JavaScript segítségével.

Tanári megjegyzés / megoldási irány

Ez még nem ipari HMI, de jó belépő a webes felügyelet logikájába.

29. feladat – Webes LED kapcsolás

Nehézségi szint: haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Távoli beavatkozás kialakítása webes felületről.

Szükséges eszközök / szoftverek

- ESP32
- WiFi
- LED
- böngésző

Előismeret

Webszerver alapok.

Feladatleírás

Készíts webes gombot, amely kapcsolja a LED állapotát.

Munkalépések

107. Hozz létre /on és /off végpontot.
108. A böngészőből hívd meg a végpontokat.
109. Kapcsold a LED-et.
110. Jelenítsd meg az aktuális állapotot.

Elvárt eredmény

A LED távolról kapcsolható.

Továbbgondolási lehetőség

Biztosíts helyi gombos vezérlést is.

Tanári megjegyzés / megoldási irány

Fontos: a távoli vezérlés ne vegye el a helyi működés lehetőségét.

30. feladat – Szenzoradatok webes megjelenítése

Nehézségi szint: haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Mért értékek webes publikálása.

Szükséges eszközök / szoftverek

- ESP32
- DHT11
- WiFi

Előismeret

DHT11 és webszerver alapok.

Feladatleírás

Jelenítsd meg a hőmérsékletet és páratartalmat böngészőben.

Munkalépések

111. Olvasd a DHT11 értékeit.
112. Készíts HTML oldalt.
113. Írd ki az értékeket.
114. Frissítsd az adatokat.

Elvárt eredmény

A böngészőben látható a környezeti adat.

Továbbgondolási lehetőség

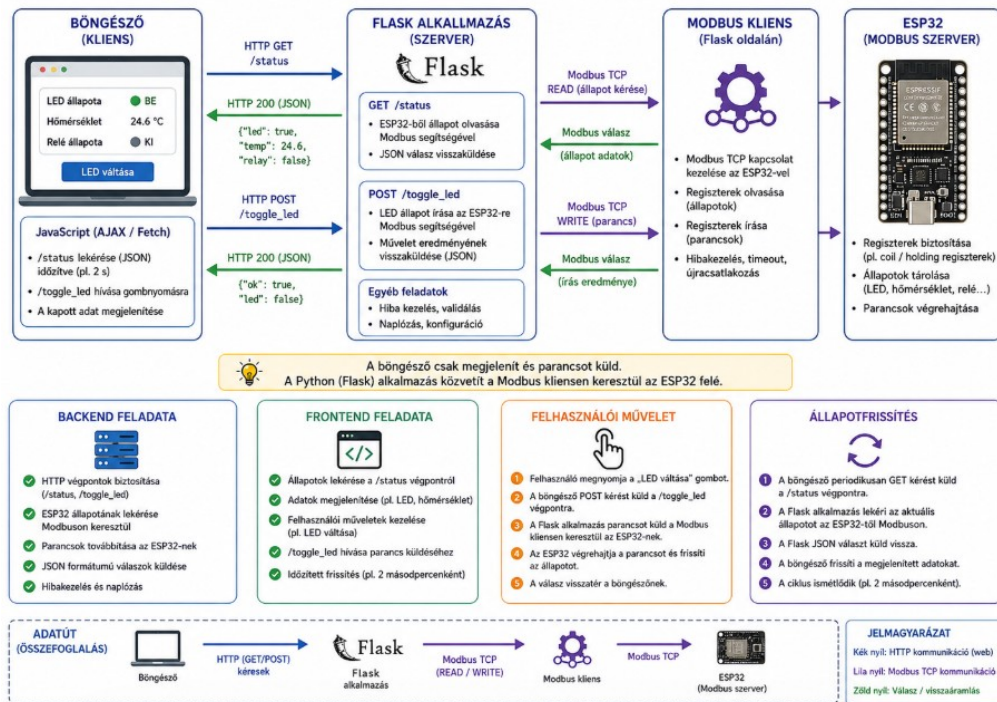
Készíts JSON formátumú /status végpontot.

Tanári megjegyzés / megoldási irány

A JSON végpont jó átmenet a Python/Flask HMI felé.

31. feladat – Webes dashboard alap

Python + Flask webes HMI működése



Nehézségi szint: haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Egyszerű HMI jellegű webes felület kialakítása.

Szükséges eszközök / szoftverek

- ESP32 vagy Python backend
- HTML
- CSS
- JavaScript

Előismeret

Szenzoradat webes megjelenítése.

Feladatleírás

Készíts dashboardot LED állapot, gombállapot, hőmérséklet és páratartalom kijelzésére.

Munkalépések

115. Hozz létre státusz API-t.
116. HTML-ben készíts kártyákat.
117. JavaScript fetch segítségével olvasd az adatokat.
118. CSS-sel különítsd el az állapotokat.

Elvárt eredmény

A felület átláthatóan jeleníti meg a rendszer állapotát.

Továbbgondolási lehetőség





Adj hozzá grafikonos trendmegjelenítést.


Tanári megjegyzés / megoldási irány

A HMI feladata megjelenítés és beavatkozás, nem a terepi logika átvétele.

7. fejezet – Modbus TCP és ipari kommunikáció


A Modbus TCP feladatok célja, hogy a tanulók megértsék: ipari környezetben az adatok gyakran szabványos címeken, regisztereken keresztül érhetőek el. A kommunikáció nem öncélú, hanem az elosztott architektúra része.


Típus	Cím (0-alapú)	Modbus funkciókód	Leírás	Adatformatum / Értékpélda
 Coil	0	01 (Olvasás) 05 (Írás)	LED állapot/vezérlés 0 = LED kikapcsolva 1 = LED bekapcsolva	1 bit 0 vagy 1
 Input Status	0	02 (Olvasás)	Gomb állapot 0 = gomb nincs lenyomva 1 = gomb lenyomva	1 bit 0 vagy 1
 16 bit Holding Register	0	03 (Olvasás) 06 (Írás) 16 (Több írás)	Hőmérséklet (tizedfok) Érték = hőmérséklet × 10 Példa: 228 = 22,8 °C	16 bit (UINT) 0...65535 Példa: 228
 16 bit Holding Register	1	03 (Olvasás) 06 (Írás) 16 (Több írás)	Páratartalom (tizedszázalék) Érték = páratartalom × 10 Példa: 523 = 52,3 %	16 bit (UINT) 0...65535 Példa: 523





A felügyeleti rendszer ciklikusan olvassa az állapotokat, és szükség esetén írja a kimeneti coil értékeket.

PÉLDA MŰVELETEK

 **Olvasás:**
btn, temp, hum

 **Írás:**
LED coil

 **Megjegyzés:** A címek 0-alapúak. A Modbus címzés a funkciókód és a regiszter típusa szerint történik. A példák UINT (előjel nélküli 16 bites) formátumot használnak.



32. feladat – Modbus adattípusok

Nehézségi szint: közép | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Coil, discrete input és holding register fogalmának megértése.

Szükséges eszközök / szoftverek

- füzet
- Modbus táblázat

Előismeret

Bináris és numerikus adatok ismerete.

Feladatleírás

Dönts el, melyik Modbus adattípus illik LED, gomb, hőmérséklet és páratartalom tárolására.

Munkalépések

119. Sorold fel az eszközöket.
120. Rendelj hozzá adattípust.
121. Indokold meg a választást.

Elvárt eredmény

A tanuló képes adatokat Modbus regisztertérben gondolkodva elhelyezni.

Továbbgondolási lehetőség

Tervezd meg talajnedvesség regiszterét is.

Tanári megjegyzés / megoldási irány

LED: coil; gomb: discrete input; hőmérséklet/páratartalom: holding register.

33. feladat - Regisztertérkép készítése

Nehézségi szint: közép-haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Átlátható Modbus címkiosztás megtervezése.

Szükséges eszközök / szoftverek

- táblázatkezelő vagy füzet

Előismeret

Modbus adattípusok.

Feladatleírás

Készíts regisztertérképet egy Smart Farm node számára.

Munkalépések

122. Határozd meg a coil címeket.
123. Határozd meg a bemeneteket.
124. Határozd meg a holding regisztereket.
125. Írj mértékegységet és skálázást.

Elvárt eredmény

A rendszer adatai dokumentált címeken érhetőek el.

Továbbgondolási lehetőség

Adj hozzá több ESP node-ot külön címtartománnyal.

Tanári megjegyzés / megoldási irány

A regisztertérkép a későbbi integráció alapidokumentuma.

34. feladat – ESP32 mint Modbus szerver

Nehézségi szint: haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Terepi csomópont adatpublikálásának megértése.

Szükséges eszközök / szoftverek

- ESP32
- Modbus TCP könyvtár
- WiFi

Előismeret

WiFi kapcsolat és ESP programozás.

Feladatleírás

Készíts ESP programot, amely Modbus szerverként elérhetővé teszi a LED és DHT adatait.

Munkalépések

126. Indíts WiFi kapcsolatot.
127. Indíts Modbus szerveret.
128. Adj hozzá coil-t a LED-hez.
129. Adj hozzá holding regisztereket DHT értékekhez.

Elvárt eredmény

Külső klienssel lekérdezhetők az állapotok.

Továbbgondolási lehetőség

Adj hozzá talajnedvesség regisztert.

Tanári megjegyzés / megoldási irány

A vezérlési logika továbbra is az ESP-n marad; a Modbus csak állapotmegosztás.

35. feladat – Távoli beavatkozás Modbus-on

Nehézségi szint: haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Felügyeleti szintből érkező parancs fogadása.

Szükséges eszközök / szoftverek

- ESP32
- Modbus kliens
- LED

Előismeret

ESP32 Modbus szerver feladat.

Feladatléírás

Tedd lehetővé, hogy egy külső kliens coil írással kapcsolja a LED-et.

Munkalépések

130. Hozz létre LED coil-t.
131. A coil állapotát írd ki LED-re.
132. Kliensből írd coil értéket.
133. Ellenőrizd a fizikai kimenetet.

Elvárt eredmény

A felügyeleti rendszer képes távoli beavatkozásra.

Továbbgondolási lehetőség

Tartsd meg a helyi gombos vezérlést is.

Tanári megjegyzés / megoldási irány

Fontos ipari elv: a távoli beavatkozás ne szüntesse meg a helyi kontrollt.

36. feladat - Modbus hibakeresés

Nehézségi szint: haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Kommunikációs hibák felismerése és elemzése.

Szükséges eszközök / szoftverek

- ESP32
- PC/Raspberry Pi
- ping
- Modbus kliens

Előismeret

Alap hálózati ismeretek.

Feladatleírás

Vizsgáld meg, mi történik hibás IP, nem futó szerver vagy hibás cím esetén.

Munkalépések

134. Pingeld az ESP-t.
135. Próbáld hibás címet olvasni.
136. Állítsd le az ESP-t és figyeld a timeoutot.
137. Dokumentáld a hibajelenségeket.

Elvárt eredmény

A tanuló képes elkülöníteni hálózati és regiszterhibákat.

Továbbgondolási lehetőség

Készíts hibakeresési döntési fát.

Tanári megjegyzés / megoldási irány

A Modbus timeout nem feltétlen programhiba; lehet hálózati, IP vagy eszközoldali probléma is.

8. fejezet – Raspberry Pi, Python és HMI felügyeleti szint

37. feladat – Raspberry Pi fix IP

Nehézségi szint: közép | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Stabil hálózati környezet kialakítása.

Szükséges eszközök / szoftverek

- Raspberry Pi
- Debian/Raspberry Pi OS
- terminál

Előismeret

Alap Linux parancsok.

Feladatleírás

Állíts be vagy ellenőrizd fix IP-címet a felügyeleti rendszer számára.

Munkalépések

138. Ellenőrizd az aktuális IP-t.
139. Pingeld az ESP32-t.
140. Dokumentáld az IP címeket.
141. Teszteld újraindítás után.

Elvárt eredmény

A Pi stabil címen működik és eléri az ESP-t.

Továbbgondolási lehetőség

Készíts hálózati topológia rajzot.

Tanári megjegyzés / megoldási irány

A fix IP vagy megbízható DHCP-foglalás alapfeltétel ipari jellegű felügyeletnél.

38. feladat – Python Modbus kliens

Nehézségi szint: haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Terepi adatok ciklikus lekérdezése Pythonból.

Szükséges eszközök / szoftverek

- Python
- pymodbus
- ESP32 Modbus szerver

Előismeret

Modbus regisztertérkép.

Feladatleírás

Készíts Python programot, amely ciklikusan olvassa a LED, gomb és DHT adatokat.

Munkalépések

142. Hozz létre ModbusTcpClient objektumot.
143. Olvasd a bemeneteket.
144. Olvasd a holding regisztereket.
145. Írd ki az értékeket.

Elvárt eredmény

A Python alkalmazás rendszeresen frissíti a digitális állapotképet.

Továbbgondolási lehetőség

Adj hibakezelést isError esetére.

Tanári megjegyzés / megoldási irány

A Python itt felügyeleti ciklust valósít meg, nem veszi át a terepi vezérlés kritikus szerepét.

39. feladat – Digitális árnyék változók

Nehézségi szint: haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Terepi állapot szoftveres leképezésének kialakítása.

Szükséges eszközök / szoftverek

- Python
- ESP32
- Modbus TCP

Előismeret

Python változók, Modbus olvasás.

Feladatléírás

Hozz létre Python változókat a terepi állapotok tárolására.

Munkalépések

146. Hozz létre led_state, btn_state, temperature, humidity változókat.
147. Frissítsd őket ciklikusan.
148. Kezeld a kommunikációs hibát.
149. Írd ki az aktuális állapotképet.

Elvárt eredmény

A Python program digitális árnyékot tart fenn.

Továbbgondolási lehetőség

Egészítsd ki soil_moisture változóval.

Tanári megjegyzés / megoldási irány

A digitális árnyék hasznos HMI és naplózás számára, de nem azonos a fizikai folyamattal.

40. feladat – Flask státusz API

Nehézségi szint: haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

HTTP alapú adatkiadás Pythonból.

Szükséges eszközök / szoftverek

- Python
- Flask
- böngésző

Előismeret

Python változók.

Feladatléírás

Készíts /status végpontot, amely JSON formátumban adja vissza az állapotokat.

Munkalépések

150. Indíts Flask alkalmazást.
151. Hozz létre /status útvonalat.
152. Térj vissza jsonify objektummal.
153. Teszteld böngészőből.

Elvárt eredmény

A böngésző vagy JavaScript kliens lekérdezheti a rendszer állapotát.

Továbbgondolási lehetőség

Adj külön /history végpontot trendadatokhoz.

Tanári megjegyzés / megoldási irány

A JSON API választja el a backend logikát a webes megjelenítéstől.

41. feladat - Webes HMI dashboard

Nehézségi szint: haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Felhasználói felület kialakítása rendszerállapotokhoz.

Szükséges eszközök / szoftverek

- HTML
- CSS
- JavaScript
- Flask

Előismeret

Flask státusz API.

Feladatleírás

Készíts webes felületet LED állapot, gombállapot, hőmérséklet és páratartalom megjelenítésére.

Munkalépések

154. Készíts HTML kártyákat.
155. JavaScript fetch-csel hívd a /status végpontot.
156. Frissítsd a DOM elemeket.
157. CSS-sel tegyél különbséget állapotok között.

Elvárt eredmény

A HMI áttekinthetően mutatja a rendszer állapotát.

Továbbgondolási lehetőség

Adj hozzá grafikonos megjelenítést Chart.js-sel.

Tanári megjegyzés / megoldási irány

A HMI felügyeleti eszköz, nem helyettesítheti a biztonságkritikus terepi logikát.

42. feladat – Távoli beavatkozás HMI-ből

Nehézségi szint: haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Felhasználói parancs továbbítása a terepi rendszer felé.

Szükséges eszközök / szoftverek

- Flask
- JavaScript
- Modbus TCP

Előismeret

Webes HMI dashboard.

Feladatleírás

Készíts gombot, amely HTTP POST kéréssel kapcsolja a LED-et a Python alkalmazáson keresztül.

Munkalépések

158. HTML-ben hozz létre gombot.
159. JS-ben küldj POST kérést.

160. Pythonban kezeld a végpontot.
161. Modbus write_coil segítségével továbbítsd a parancsot.

Elvárt eredmény

A webes felületről parancs adható a terepi rendszernek.

Továbbgondolási lehetőség

Jelezd vissza, ha a parancs nem sikerült.

Tanári megjegyzés / megoldási irány

A távoli beavatkozás csak kiegészítő funkció; offline módban a helyi vezérlés működjön tovább.

43. feladat – Párhuzamos működés

Nehézségi szint: haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Felügyeleti ciklus és webkiszolgáló egyidejű futtatása.

Szükséges eszközök / szoftverek

- Python
- threading
- Flask

Előismeret

Python alapok.

Feladatleírás

Indíts külön szálon Modbus polling ciklust, miközben a Flask szerver is fut.

Munkalépések

162. Készíts plc_cycle vagy supervisory_cycle függvényt.
163. Indítsd daemon threadként.
164. Futtasd a Flask appot.
165. Teszteld a folyamatos frissítést.

Elvárt eredmény

A webkiszolgálás nem blokkolja az állapotfrissítést.

Továbbgondolási lehetőség

Mérd meg, mi történik túl gyakori lekérdezésnél.

Tanári megjegyzés / megoldási irány

A több szál kezelése bevezetést ad a párhuzamos feldolgozás problémáiba.

9. fejezet – Hibakeresés és diagnosztikai feladatok

44. feladat – LED nem világít

Nehézségi szint: alap | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Egyszerű hardveres hiba diagnosztikája.

Szükséges eszközök / szoftverek

- ESP32
- LED
- ellenállás
- multiméter opcionálisan

Előismeret

LED bekötés.

Feladtleírás

Keresd meg, miért nem világít a LED.

Munkalépések

166. Ellenőrizd a GPIO számot.
167. Ellenőrizd a polaritást.
168. Ellenőrizd az ellenállást.
169. Teszteld egyszerű ON programmal.

Elvárt eredmény

A tanuló képes alapvető hardveres hibát elkülöníteni szoftverhibától.

Továbbgondolási lehetőség

Írj hibakeresési jegyzőkönyvet.

Tanári megjegyzés / megoldási irány

Egyszerű kimeneti hibánál mindig a minimál teszttel kell kezdeni.

45. feladat – DHT11 hibás értéket ad

Nehézségi szint: közép | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Szenzorolvasási hiba okainak feltárása.

Szükséges eszközök / szoftverek

- DHT11
- ESP32
- soros monitor

Előismeret

DHT mérés.

Feladatléírás

Vizsgáld meg, mi okozhat NaN vagy irreális DHT értékeket.

Munkalépések

170. Ellenőrizd a bekötést.
171. Ellenőrizd a szenzortípust.
172. Növeld az olvasási időközt.
173. Teszteld külön programmal.

Elvárt eredmény

A tanuló felismeri, hogy szenzorhibánál időzítés és könyvtárbeállítás is számít.

Továbbgondolási lehetőség

Hasonlítsd össze DHT11 és DHT22 pontosságát.

Tanári megjegyzés / megoldási irány

DHT11-et nem célszerű túl gyakran olvasni.

46. feladat - WiFi nem csatlakozik

Nehézségi szint: közép | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Hálózati hibák alapdiagnosztikája.

Szükséges eszközök / szoftverek

- ESP32
- WiFi router
- soros monitor

Előismeret

WiFi programozás.

Feladatleírás

Keresd meg, miért nem csatlakozik az ESP32 a hálózathoz.

Munkalépések

174. Ellenőrizd SSID-t.
175. Ellenőrizd jelszót.
176. Figyeld a soros monitort.
177. Próbáld másik hálózatot.

Elvárt eredmény

A tanuló képes elkülöníteni konfigurációs és hálózati hibát.

Továbbgondolási lehetőség

Jelenítsd meg LCD-n a WiFi státuszt.

Tanári megjegyzés / megoldási irány

Az ékezetes vagy rejtett karakteres jelszavak is okozhatnak problémát.

47. feladat - HMI nem frissül

Nehézségi szint: haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Webes és backend hibák elkülönítése.

Szükséges eszközök / szoftverek

- Flask
- JavaScript
- böngésző fejlesztői eszközök

Előismeret

Flask és JS alapok.

Feladatleírás

Vizsgáld meg, miért nem változik a webes HMI állapota.

Munkalépések

178. Nyisd meg a /status végpontot.
179. Ellenőrizd a böngésző konzolt.
180. Ellenőrizd a JavaScript ID-ket.

181. Figyeld a Flask logot.

Elvárt eredmény

A tanuló képes frontend és backend hibát megkülönböztetni.

Továbbgondolási lehetőség

Adj hibaüzenetet a felhasználói felületre.

Tanári megjegyzés / megoldási irány

A HMI hibakeresése több réteget érint: adatforrás, API, JS, HTML.

48. feladat - Modbus timeout

Nehézségi szint: haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Ipari kommunikációs hiba elemzése.

Szükséges eszközök / szoftverek

- ESP32
- Python kliens
- hálózat

Előismeret

Modbus TCP.

Feladatléírás

Elemezd, mi okozhat Modbus timeout hibát.

Munkalépések

182. Pingeld az ESP-t.
183. Ellenőrizd az IP-t.
184. Ellenőrizd, fut-e a Modbus szerver.
185. Ellenőrizd a portot.

Elvárt eredmény

A tanuló strukturáltan tud Modbus kapcsolatot diagnosztizálni.

Továbbgondolási lehetőség

Készíts diagnosztikai táblázatot hibajelenség-ok-megoldás formában.

Tanári megjegyzés / megoldási irány

A timeout gyakran hálózati vagy szolgáltatáselérési hiba, nem feltétlen kódhiba.

49. feladat – Instabil vezérlés határérték körül

Nehézségi szint: közép-haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Hiszterézis szükségességének felismerése.

Szükséges eszközök / szoftverek

- DHT11 vagy analóg szenzor
- aktuátor

Előismeret

Küszöbértékes vezérlés.

Feladatléírás

Figyeld meg, hogyan kapcsolgat a rendszer, ha a mért érték a határérték körül ingadozik.

Munkalépések

186. Állíts be egyetlen küszöbértéket.
187. Figyeld a kimenetet.
188. Adj hozzá hiszterézist.
189. Hasonlítsd össze a működést.

Elvárt eredmény

A tanuló megérti, miért kell külön be- és kikapcsolási határ.

Továbbgondolási lehetőség

Készíts grafikus magyarázatot.

Tanári megjegyzés / megoldási irány

A zajos mérés és a rossz határérték instabil működést okozhat.

10. fejezet – Komplex projektfeladatok

50. feladat – Smart Farm mini rendszer

Nehézségi szint: projekt | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Komplex okosfarm funkciók integrálása.

Szükséges eszközök / szoftverek

- ESP32
- DHT11
- talajnedvesség
- fotoellenállás
- LCD
- pumpa/LED/ventilátor

Előismeret

Korábbi fejezetek feladatai.

Feladatléírás

Építs mini Smart Farm rendszert, amely világítást, szellőztetést, öntözést és helyi kijelzést tartalmaz.

Munkalépések

190. Tervezd meg az I/O listát.
191. Készíts regisztertérképet.
192. Építsd meg a hardvert.
193. Írd meg a helyi vezérlést.
194. Teszteld funkcionként.

Elvárt eredmény

Működő, több alrendszerből álló demonstrációs modell.

Továbbgondolási lehetőség

Adj hozzá webes HMI-t.

Tanári megjegyzés / megoldási irány

A komplex feladatban a dokumentáció és moduláris programfelépítés kulcsfontosságú.

51. feladat - Elosztott két node-os rendszer

Nehézségi szint: projekt-haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Több terepi csomópont koncepciójának megértése.

Szükséges eszközök / szoftverek

- 2 ESP32 vagy szimulált node
- Raspberry Pi/PC
- Modbus TCP

Előismeret

Modbus és hálózat.

Feladatleírás

Tervezd meg, hogyan működne két külön Smart Farm node egy közös felügyeleti rendszer alatt.

Munkalépések

195. Adj IP címet a node-oknak.
196. Készíts külön regisztertérképet.
197. Tervezd meg a Python lekérdezést.
198. Jelenítsd meg egy dashboardon.

Elvárt eredmény

A tanuló képes több eszközből álló rendszert tervezni.

Továbbgondolási lehetőség

Adj hozzá prioritáskezelést, ha az egyik node hibás.

Tanári megjegyzés / megoldási irány

A több node-os architektúra már valódi ipari skálázhatósági szemléletet ad.

52. feladat – Digitális árnyék dashboard

Nehézségi szint: projekt | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Fizikai rendszer webes állapotképének létrehozása.

Szükséges eszközök / szoftverek

- ESP32
- Python
- Flask
- HTML/JS

Előismeret

HMI alapok.

Feladatleírás

Készíts dashboardot, amely a terepi rendszer állapotát digitális árnyékként jeleníti meg.

Munkalépések

199. Olvasd az ESP állapotait.

200. Tárold Python változóiban.
201. Add ki JSON API-n.
202. Jelenítsd meg webes felületen.
203. Adj manuális beavatkozási lehetőséget.

Elvárt eredmény

A webes HMI követi a fizikai rendszer állapotát.

Továbbgondolási lehetőség

Adj hozzá trendgrafikont.

Tanári megjegyzés / megoldási irány

Fontos megérteni: a digitális árnyék csak akkor hiteles, ha az adatfrissítés megbízható.

53. feladat - Mini SCADA szemléletű feladat

Nehézségi szint: projekt-haladó | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Felügyeleti rendszer szemléletének bevezetése.

Szükséges eszközök / szoftverek

- Raspberry Pi/PC
- ESP32
- Flask
- Modbus TCP

Előismeret

Digitális árnyék dashboard.

Feladatleírás

Készíts mini SCADA jellegű rendszert: állapotok, riasztások, naplózás, trendek.

Munkalépések

204. Határozd meg a riasztási feltételeket.
205. Készíts naplózást.
206. Jeleníts meg trendet.
207. Adj riasztási állapotot a HMI-n.

Elvárt eredmény

A rendszer nemcsak vezérel, hanem felügyel és dokumentál.

Továbbgondolási lehetőség

Exportáld az adatokat CSV-be.

Tanári megjegyzés / megoldási irány

Ez már ipari felügyeleti gondolkodás: nem elég működtetni, látni és naplózni is kell.

54. feladat – Saját okosház konfigurátor

Nehézségi szint: projekt | **Munkaforma:** egyéni / páros / kiscsoportos

Cél

Adatbázis alapú rendelési logika tervezése.

Szükséges eszközök / szoftverek

- webes környezet
- adatbázis
- HTML/Python opcionális

Előismeret

Projektarchitektúra ismerete.

Feladatleírás

Tervezd meg, hogyan rendelhető egyedi okosház konfiguráció egy webes felületen.

Munkalépések

208. Határozd meg a választható funkciókat.
209. Tervezd meg az adatbázis mezőit.
210. Írd le a rendelés útját.
211. Kapcsold a gyártási feladathoz.

Elvárt eredmény

A tanuló megérti a gyártástámogató informatikai rendszer alaplogikáját.

Továbbgondolási lehetőség

Készíts prototípus űrlapot.

Tanári megjegyzés / megoldási irány

Ez kapcsolja össze az okosház projektet az Ipar 4.0 vállalati folyamatmodellel.

Melléklet – Javasolt értékelési szempontok

Szempont	Alap	Jó	Kiemelkedő
----------	------	----	------------

Működés	A program részben működik	A fő funkció stabil	A rendszer több helyzetben is megbízható
Dokumentáció	Alap leírás készült	Átlátható lépések	Mérnöki igényű rajzok, táblázatok, indoklások
Rendszerszemlélet	Egyes elemeket ért	Kapcsolatokat felismer	Teljes architektúrát értelmez
Hibakeresés	Segítséggel javít	Önállóan azonosít hibát	Diagnosztikai módszert alkalmaz
Csapatmunka	Részt vesz	Feladatot vállal	Szervezi és támogatja a közös munkát

Záró megjegyzés

A feladatgyűjtemény célja, hogy a tanulók ne csupán programokat készítsenek, hanem megértsék, hogyan épül fel egy valós, mérnöki szemléletű, Ipar 4.0 jellegű rendszer. A Smart Farm projekt ebben a formában nem késztermék, hanem tanulási környezet: lehetőséget ad a hibákból történő tanulásra, az iteratív fejlesztésre, a hálózati gondolkodásra és a fizikai-digitális rendszerek összekapcsolására.