

JEGYZŐKÖNYV

Robotika I.

Féléves beadandó

Készítette: **Orosz Kristóf**
Neptunkód: **EYZWG9**
Dátum: 2025. december 02.

Sárospatak, 2025

Tartalomjegyzék

Bevezetés	3
Projektterv	3
Készletlista	4
Arduinos megvalósítás	4
Sematikus nézet	7
Projektépítés	8
Forrásfájl (kód)	11

Bevezetés

A féléves beadandóm során egy Arduino-alapú, kétirányú gépjárműforgalmat és gyalogos átkelést kezelő közlekedési csomópont vezérlésének megvalósítását hajtottam végre . A projekt során egy valós forgalomirányító rendszer működési logikáját modellezem A rendszer feladata, hogy biztonságos és szabályozott jelzésváltást biztosítson az autós és gyalogos forgalom között, valamint kezelje az üzemen kívüli állapotot és a felhasználói visszajelzéseket.

A vezérlés alapját egy Arduino mikrokontroller fogja képezi, amely koordinálja a különböző bemeneti és kimeneti modulok működését. A projekt során várhatóan felhasznált alkatrészek közé tartoznak többek között a LED-ek (piros, sárga, zöld jelzőlámpák), nyomógombok (gyalogos és üzemmód kapcsoló), piezo hangjelző (buzzer), valamint egy 16×2 karakteres LCD I2C kijelző a vizuális visszajelzések biztosítására.

Projektterv

1. Alapállapot:

Ebben az állapotban a rendszer alapértelmezetten a járműforgalmat engedi haladni, vagyis:

- A járműforgalmi lámpa zöld.
- A gyalogos lámpa piros.
- Az LCD kijelzőn az üzenet:
„Kerem varjon!”

2. Gyalogos-átkelés kérése:

Amikor a gyalogos nyomógombbal átkelést kér:

1. Az Arduino megerősíti a bemenetet.

Ekkor a járműforgalom számára előkészítési fázis indul: A zöld LED elalszik, a sárga LED világít kb. 3 másodpercig, jelezve, hogy a forgalom meg fog állni, majd pirosra vált, és a gyalogosoknak a zöld jelzés elindul.

3. Átkelési ciklus:

Ebben a fázisban a gyalogos zöld jelzés aktívvá válik.

a) Átkelés megkezdése

- A jármű lámpa piros marad,
- A gyalogoslámpa zöldre vált,
- A hangjelző folyamatosan sípol.
- Az LCD kijelzőn megjelenik:
„Atkelhet!”

b) Biztonsági lezáró szakasz (villogó zöld)

- A zöld lámpa háromszor villog, majd pirosra vált.
- A buzzer közben 1 ms -ként pittyeg.

c) Rendszer visszaállása alapállapotba

- A gyalogoslámpa pirosra vált,
- A járműforgalmi lámpa zöldre kapcsol,
- Az LCD-n:
„Kerem varjon!”

d, Rendszer kikapcsolása

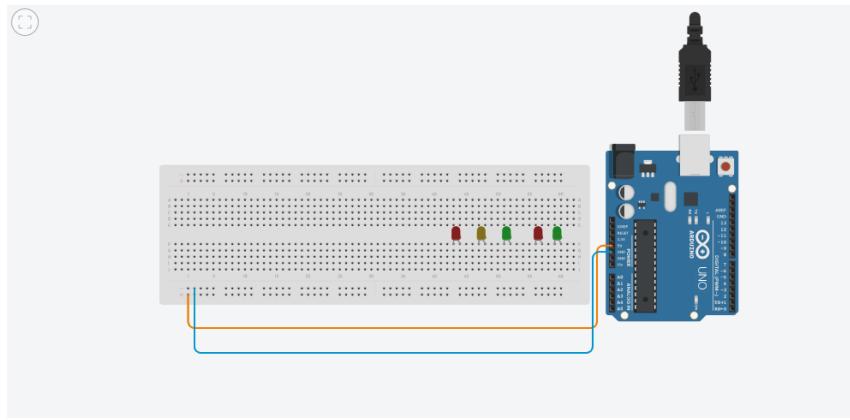
- Az egyik gomb 5 ms -ig történő megnyomása.
- Az LCD-n:
„Uzemen kivul!”

Készletlista

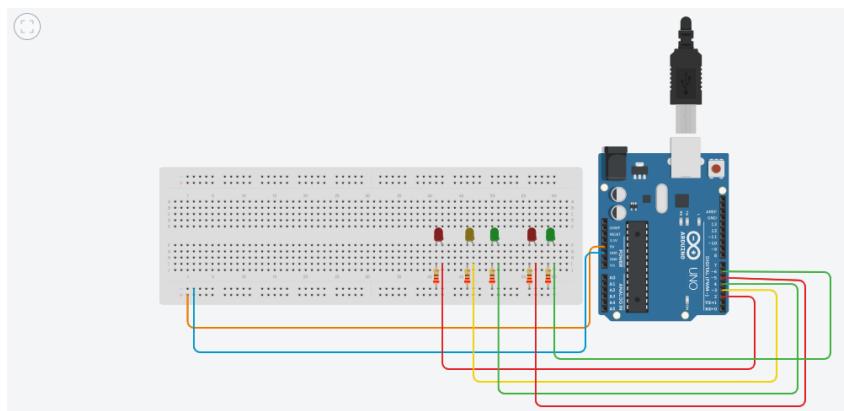
Sorszám	Megnevezés	Mennyiség	Funkció
1.	Arduino Uno mikrokontroller	1 db	A vezérlőegység szerepét tölti be
2.	Breadboard, 400 pontos	1 db	Kapcsolások építéséhez
3.	16×2 karakteres LCD kijelző	1 db	Állapotüzenetek megjelenítése
4.	10 kΩ ellenállás	2 db	Ellenállás nyomógombhoz.
5.	Nyomógombok	2 db	Gyalogos jelzés aktív kérése
6.	Aktív buzzer	1 db	Hangjelzés gyalogosok számára
7.	LED – piros (jármű jelzőlámpa)	1 db	Jármű STOP jelzés
8.	LED – sárga (jármű jelzőlámpa)	1 db	Jármű figyelmeztető jelzés
9.	LED – zöld (jármű jelzőlámpa)	1 db	Jármű lehet jelzés
10.	LED – piros (gyalogos jelzőlámpa)	1 db	Gyalogos STOP jelzés
11.	LED – zöld (gyalogos jelzőlámpa)	1 db	Gyalogos lehet jelzés
12.	LED-ekhez tartozó ellenállások (220Ω)	5 db	LED-hez tartozó ellenállás.

Arduinos megvalósítás

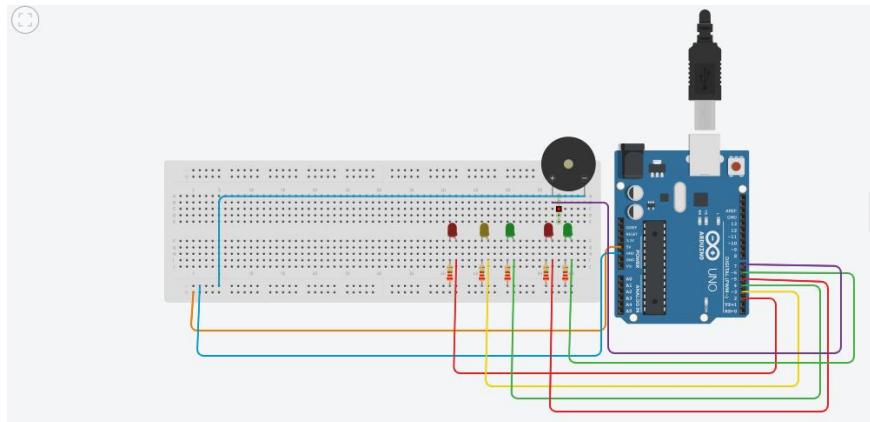
1. lépés: Ebben a lépésben összekötöttem az Arduino Uno alapvető tápvezetékeit a breadboarddal. Az Arduino 5V kimenetét rákötöttem a breadboard pozitív táp sínjére, a GND pontját pedig a negatív sínre, így a teljes breadboard egységes tápellátást kapott. Ezt követően beépítettem a rendszerbe a jelzőlámpákhoz szükséges LED-eket. Először elhelyeztem a jármű jelzőlámpa három LED-jét – a pirosat, a sárgát és a zöldet –, majd melléjük a gyalogos jelzőlámpa két LED-jét, a pirosat és a zöldet.



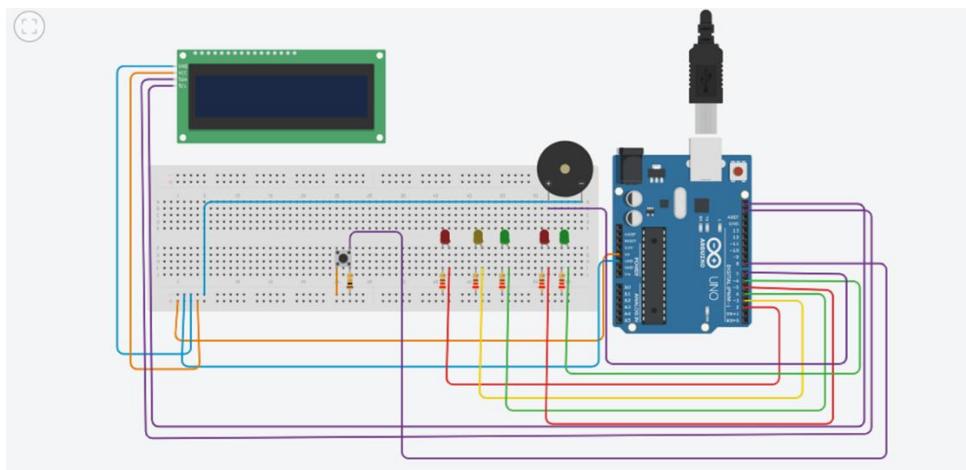
2. lépés: Ebben a lépésben minden LED alá beépítettem a szükséges soros ellenállásokat, majd az egyes LED-ek kivezetéseit összekötöttem az Arduino megfelelő digitális lábaival. minden LED katódját a breadboard közös negatív sínjére kötöttem, az anód elé pedig külön ellenállást helyeztem el, hogy biztosítsam a megfelelő áramkorlátozást. Ezután az ellenállások másik végéről vezetékekkel csatlakoztattam a LED-eket az Arduino digitális kimeneteihez. A jármű jelzőlámpa piros, sárga és zöld LED-jeit külön vezetékeken keresztül kapcsoltam be, majd ugyanígy kötöttem rá a gyalogos jelzőlámpa piros és zöld LED-jeit is.



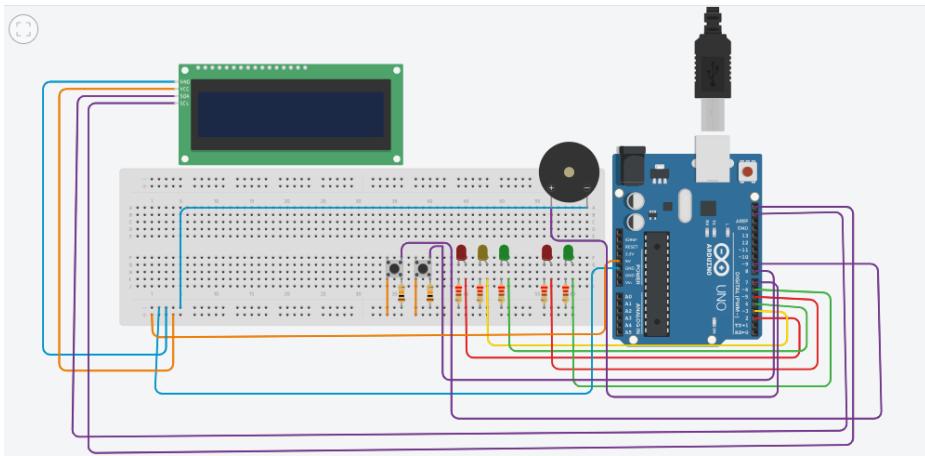
3. lépés: Ebben a lépésben tovább bővítettem a rendszert, mégpedig azzal, hogy beépítettem az aktív buzzert, amely a gyalogosok számára ad hangjelzést. A buzzert a breadboard jobb oldalán helyeztem el, majd az egyik lábát a breadboard pozitív sínjére kötöttem, a másik lábat pedig egy külön vezetékkel összekötöttem az Arduino egyik digitális kimenetével. Ezzel lehetővé vált, hogy programból vezéreljem a hangjelzést, például akkor, amikor a gyalogosok számára zöld jelzés jelenik meg.



4. lépés: Ebben a lépésben tovább fejlesztettem a rendszert azzal, hogy beépítettem a gyalogos kérőgombot és a 16×2-es LCD kijelzőt. Először a nyomógombot helyeztem el a breadboardon, majd az egyik lábát ellenálláson keresztül a földre kötöttem, a másik lábat pedig egy vezetékkel összekötöttem az Arduino egyik digitális bemenetével. Így a gomb alapállapotban alacsony jelet ad, megnyomáskor pedig magas szintet, amit az Arduino könnyen érzékelni tud. Ezután elkezdtem bekötni a 16×2-es LCD kijelzőt. A kijelző VCC és GND lábait rákötöttem a breadboard táp- és földsínjére, majd az adat- és vezérlővonalakat egyenként bekötöttem az Arduino megfelelő digitális lábaira.



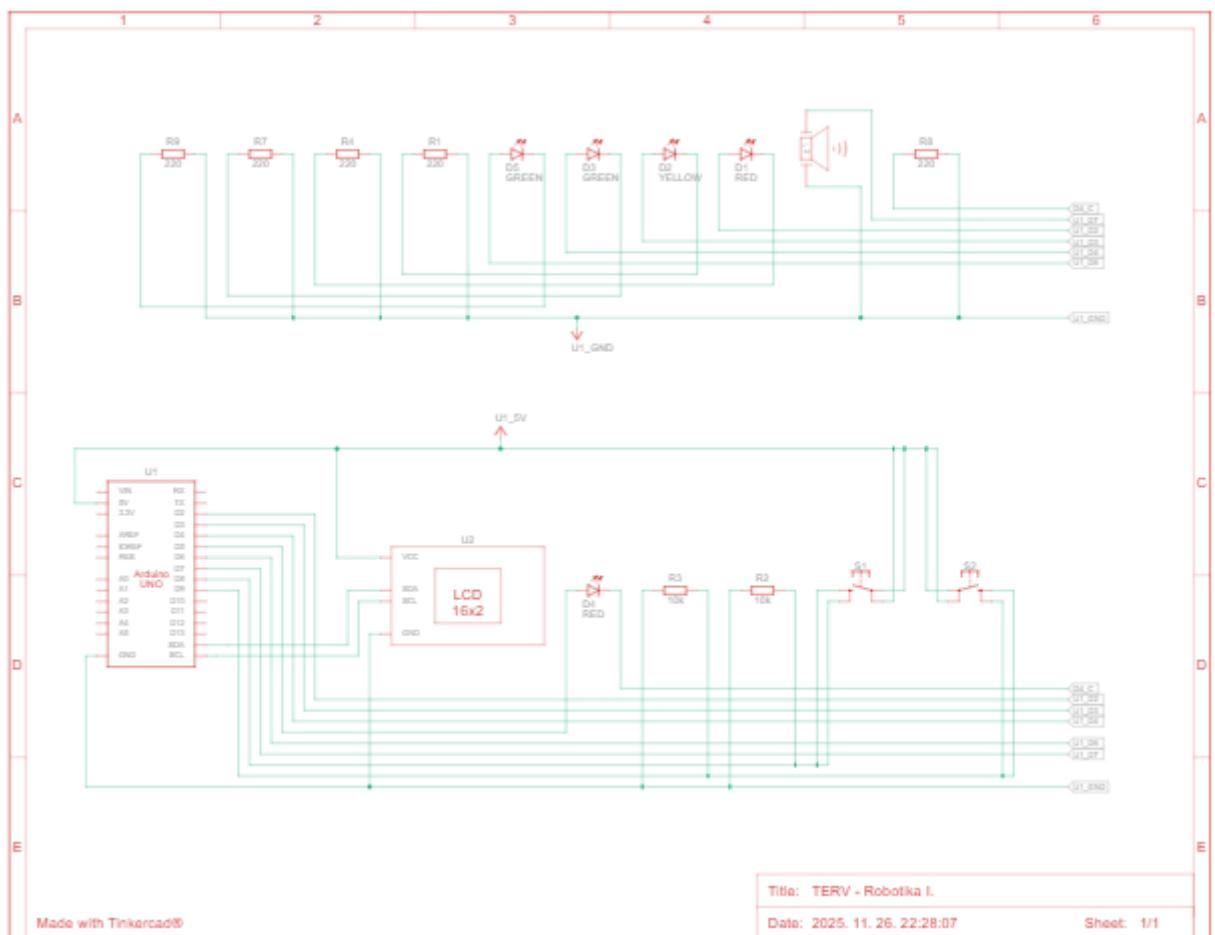
5. lépés: Ebben a lépésben tovább bővítettem a gyalogosátkelő rendszer kezelőfelületét azzal, hogy beépítettem egy második nyomógombot a breadboardra. A második gombot ugyanúgy kötöttem be, mint az elsőt: az egyik lábát ellenálláson keresztül a földre csatlakoztattam, a másik lábat pedig egy külön vezetékkel összekötöttem az Arduino egy szabad digitális bemenetével. Ezzel a gombbal a rendszer kikapcsolását tudom végrehajtani.



Sematikus nézet

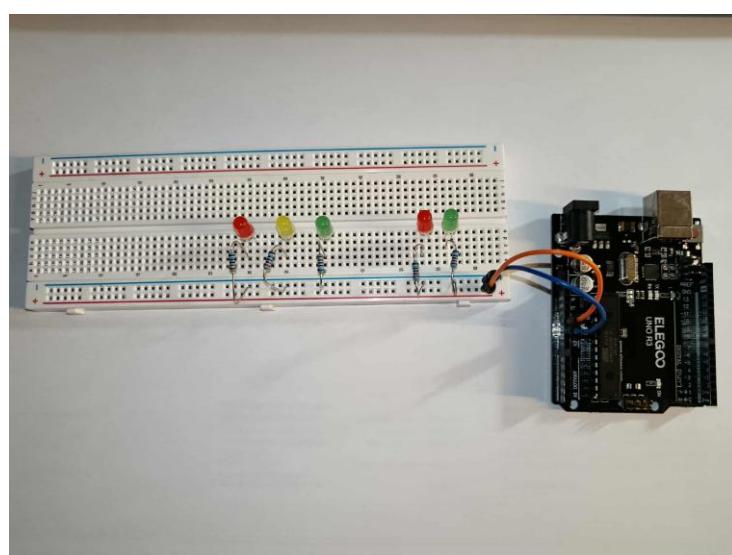
A sematikus nézeten a megépített arduinos projektem kapcsolási rajza látható, amely a gyalogosátkelő-forgalomirányító rendszer teljes bekötését mutatja. A rajz felső részén helyezkednek el a jelzőlámpák LED-jei: külön látható a jármű jelzőlámpa három LED-je (piros, sárga, zöld), valamint a gyalogos jelzőlámpa két LED-je (piros és zöld). Mindegyik LED elő soros ellenállás került, amely biztosítja a megfelelő áramkorlátozást. Ugyanezen a részen található az aktív buzzer is, amely hangjelzést ad a gyalogosok számára.

A rajz alsó részén kapott helyet az Arduino Uno mikrokontroller, amely a rendszer központi vezérlője. A LED-ek, a buzzer, valamint a nyomógombok minden digitális kivezetéseire csatlakoznak. A kapcsolási rajz egy 16×2-es LCD kijelző bekötését is tartalmazza: láthatóak a tápellátási pontok (VCC, GND), valamint a vezérlő- és adatvonalak, amelyek az Arduino megfelelő lábaira futnak be. A kijelző működéséhez szükséges kontrasztállító potméter lábai és csatlakozásai is jelölve vannak. A jobb oldalon két nyomógomb kapott helyet, külön ellenállásokkal és vezetékekkel, amelyek biztosítják a stabil bemeneti jelet az Arduino felé. A rajz jól strukturált, logikusan tagolt: külön szintekre rendezve mutatja be a vezérlőt, a kijelzőt, a jelzőfényeket és a kezelőgombokat.

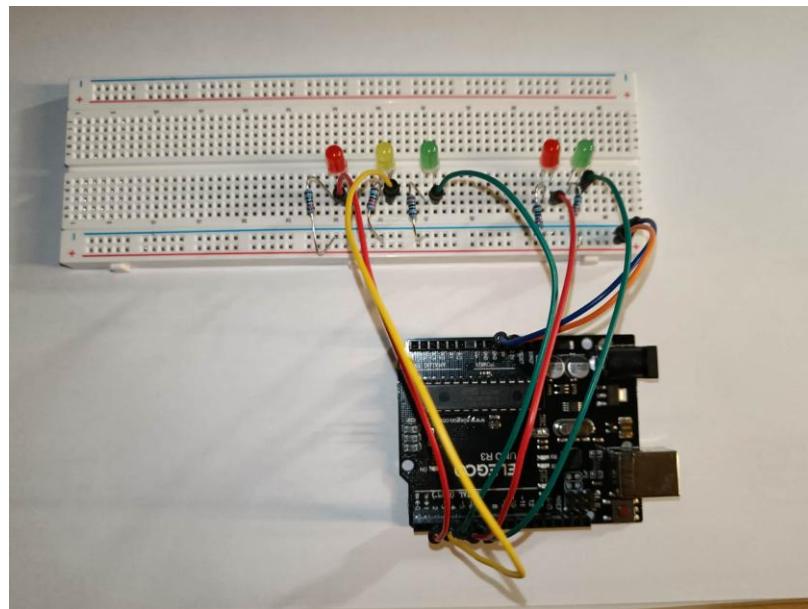


Projektépítés

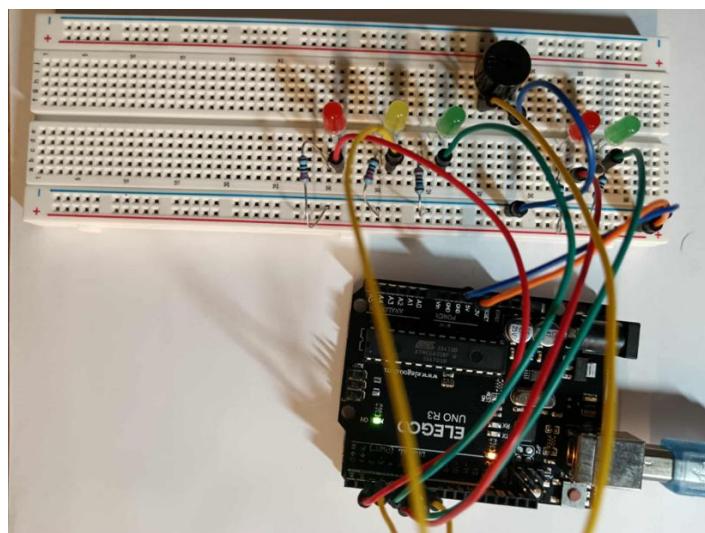
1 . lépés: Ebben a lépésben elkezdtem kiépíteni a jelzőlámpák alapját, ezért a breadboardra elhelyeztem az öt darab LED-et: háromat a jármű jelzőlámpához (pirosat, sárgát és zöldet), valamint két darab LED-et a gyalogosok jelzéséhez (pirosat és zöldet). minden LED előtt bekötöttem egy-egy soros ellenállást, hogy szabályozzam az átfolyó áramot, és ezzel megvédjem az alkatrészeket a túlterheléstől.



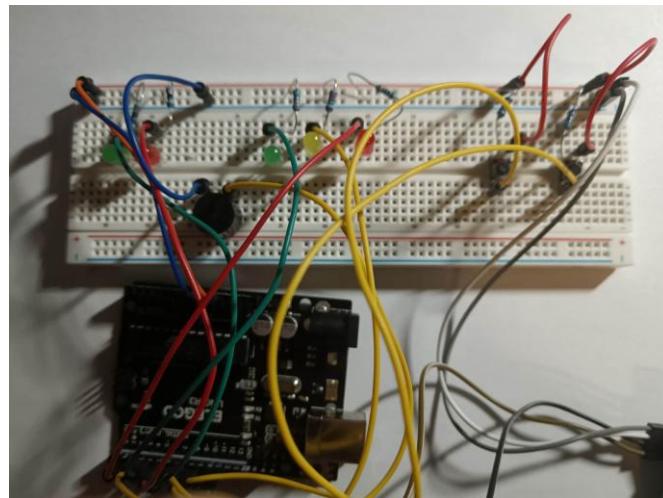
2.lépés: Ebben a lépésben tovább építettem a fényjelző rendszert azáltal, hogy az összes LED-et véglegesen összekötöttem az Arduino megfelelő digitális lábaival. A korábban elhelyezett ellenállások és LED-ek után most minden egyes LED anódját egy külön vezetékkel csatlakoztattam az Arduino digitális kimeneteihez, így mindegyik fényjelzés függetlenül vezérelhetővé vált.



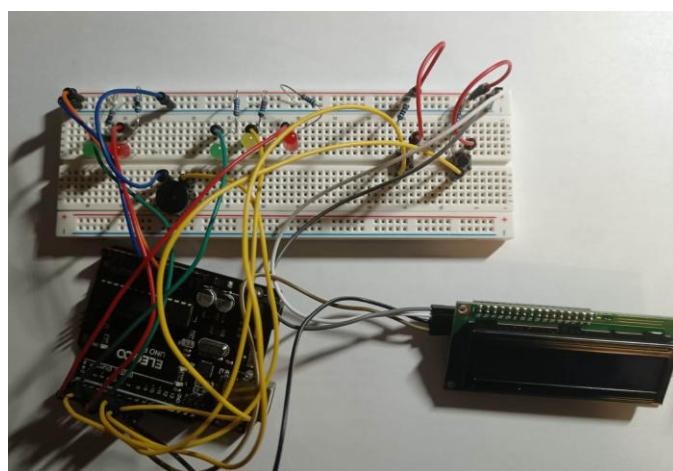
3.lépés: Ebben a lépésben tovább bővítem a gyalogosátkelő rendszer funkcióit azzal, hogy beépítettem az aktív buzzert a breadboardra. A buzzert a LED-ek mellé helyeztem el, és az egyik lábat a breadboard pozitív tápvonalára kötöttem, a másik lábat pedig külön vezetékkel az Arduino megfelelő digitális kimenetéhez csatlakoztattam. Ezzel az eszköz immár képes hangjelzést adni, amit később a programban a gyalogos zöld jelzés fázisához tudok majd kötni.



4.lépés: A breadboardra ebben a lépésben a két nyomógombot helyeztem fel. Mindkét gombot úgy rögzítettem. Ezután minden gomb egyik lábát a breadboard pozitív sínjére kötöttem, amelyet 5V tápfeszültségre csatlakoztattam. A gombok másik lábát külön sorokba vezettem, majd ezeket a sorokat sárga vezetékekkel összekötöttem az Arduino megfelelő digitális bemeneti lábaival. A stabil működés érdekében minden gomb mellé sorosan egy-egy ellenállást is beépítettem, amelyeket a GND sínre kötöttem.



6.lépés: Ebben a lépésben a kijelzőt helyeztem el a kapcsolási áramkörben. A modul fizikai rögzítéséhez a breadboard mellé igazítottam, hogy a csatlakozótüskék könnyen hozzáférhetők legyenek, majd a kijelző I2C interfészének négy vezetékét egyenként bekötöttem az Arduino megfelelő lábaira. A GND vezetéket az Arduino földpontjához, a VCC vezetéket az 5V tápfeszültséghez csatlakoztattam. Ezt követően az SDA vezetéket az A4-es, az SCL vezetéket pedig az A5-ös analóg lábra kötöttem, ezzel biztosítva a kijelző I2C kommunikációját a mikrokontrollerrel.



Forrásfájl (kód)

```
#include <Adafruit_LiquidCrystal.h> // LCD kijelző könyvtára

Adafruit_LiquidCrystal lcd_1(0); // LCD objektum (0 → alap I2C emuláció)

// ---- KIVEZETÉSEK DEFINIÁLÁSA ----
// Autó jelzőlámpa LED-ek
const int autoPiros = 2;
const int autoSarga = 3;
const int autoZold = 4;

// Gyalogos jelzőlámpa LED-ek
const int gyalogPiros = 5;
const int gyalogZold = 6;

// Hangjelző (buzzer) és kérőgomb
const int buzzer = 7;
const int gomb = 8;

// Üzemmódváltó gomb ("üzemen kívül" mód)
const int uzemGomb = 9;

// ---- ÁLLAPOTVÁLTOZÓK ----
bool gyalogMod = false; // Gyalogos mód be/ki
bool elozoAllapot = HIGH; // Kérőgomb előző állapota

bool uzemenKivul = false; // Üzemen kívül mód állapota
bool elozoUzemAllapot = HIGH; // Üzemgomb előző állapota

void setup() {

    // LED-ek beállítása kimenetként
    pinMode(autoPiros, OUTPUT);
    pinMode(autoSarga, OUTPUT);
    pinMode(autoZold, OUTPUT);

    pinMode(gyalogPiros, OUTPUT);
    pinMode(gyalogZold, OUTPUT);

    pinMode(buzzer, OUTPUT);

    // Gombok bemenetként (felhúzó ellenállással)
    pinMode(gomb, INPUT_PULLUP);
```

```

pinMode(uzemGomb, INPUT_PULLUP);

// LCD inicializálás
lcd_1.begin(16, 2);
lcd_1.setBacklight(1);

// Kezdő állapot: autó lehet, gyalogos tilos
digitalWrite(autoZold, HIGH);
digitalWrite(autoPiros, LOW);
digitalWrite(autoSarga, LOW);

digitalWrite(gyalogPiros, HIGH);
digitalWrite(gyalogZold, LOW);

// LCD kezdő szöveg
lcd_1.clear();
lcd_1.setCursor(0, 0);
lcd_1.print("Kerem varjon...");
}

void loop() {

// ---- ÜZEMMÓD GOMB FIGYELÉSE ----
bool uzemJelenlegi = digitalRead(uzemGomb);

// Élérzékelés
if (elozoUzemAllapot == HIGH && uzemJelenlegi == LOW) {

uzemenKivul = !uzemenKivul; // módváltás

if (uzemenKivul) {
// ---- UZEMEN KÍVÜL MÓD ----

// minden LED kikapcsolása
digitalWrite(autoZold, LOW);
digitalWrite(autoSarga, LOW);
digitalWrite(autoPiros, LOW);

digitalWrite(gyalogZold, LOW);
digitalWrite(gyalogPiros, LOW);

lcd_1.clear();
lcd_1.setCursor(0, 0);
lcd_1.print("Uzemen kivul");

} else {
// ---- VISSZA NORMAL MÓDRA ----
}
}
}

```

```

lcd_1.clear();
lcd_1.setCursor(0, 0);
lcd_1.print("Kerem varjon...");

digitalWrite(autoPiros, LOW);
digitalWrite(autoZold, HIGH);
digitalWrite(autoSarga, LOW);

digitalWrite(gyalogPiros, HIGH);
digitalWrite(gyalogZold, LOW);
}

delay(200); // pergésgátlás
}

elozoUzemAllapot = uzemJelenlegi;

// ---- UZEMEN KÍVÜL: SÁRGA VILLÓZIK ----
if (uzemenKivul) {
    digitalWrite(autoSarga, HIGH);
    delay(400);
    digitalWrite(autoSarga, LOW);
    delay(400);
    return; // kilépés, nem működik a normál logika
}

// ---- GYALOGOS GOMB FIGYELÉSE ----
bool jelenlegi = digitalRead(gomb);

if (elozoAllapot == HIGH && jelenlegi == LOW) {

    gyalogMod = !gyalogMod;

    // ===== GYALOGOS MÓD BEKAPCSOLÁSA ======
    if (gyalogMod) {

        // Autó: zöld → sárga → piros
        digitalWrite(autoZold, LOW);

        digitalWrite(autoSarga, HIGH);
        delay(3000);
        digitalWrite(autoSarga, LOW);

        digitalWrite(autoPiros, HIGH);
    }
}

```

```

// Gyalogos zöld
digitalWrite(gyalogPiros, LOW);
digitalWrite(gyalogZold, HIGH);

lcd_1.clear();
lcd_1.setCursor(0, 0);
lcd_1.print("Atkelhet!");

// Buzzer alap hang
tone(buzzer, 1500);

// 15 MP GYALOGOS ZÖLD

delay(12000); // 12 mp folyamatos zöld

// Utolsó 3 másodperc villogás
for (int i = 0; i < 6; i++) { // 6x 0.5 mp = 3 mp
  digitalWrite(gyalogZold, LOW);
  noTone(buzzer);
  delay(250);

  digitalWrite(gyalogZold, HIGH);
  tone(buzzer, 2000);
  delay(250);
}

// ----- VISSZA PIROSRA -----
digitalWrite(gyalogZold, LOW);
digitalWrite(gyalogPiros, HIGH);
noTone(buzzer);

lcd_1.clear();
lcd_1.setCursor(0, 0);
lcd_1.print("Kerem varjon...");

// Autók visszarendezése: piros → sárga → zöld
digitalWrite(autoPiros, LOW);

digitalWrite(autoSarga, HIGH);
delay(3000);
digitalWrite(autoSarga, LOW);

digitalWrite(autoZold, HIGH);

```

```

gyalogMod = false; // gyalog mód lezárása

// ===== GYALOGOS MÓD KIKAPCSOLÁSA ======
} else {

    // Biztonsági: buzzer leállítás
    noTone(buzzer);

    // Rövid figyelmeztető villogás
    for (int i = 0; i < 5; i++) {
        digitalWrite(gyalogZold, HIGH);
        tone(buzzer, 2000);
        delay(100);

        digitalWrite(gyalogZold, LOW);
        noTone(buzzer);
        delay(100);
    }

    // Gyalogos piros
    digitalWrite(gyalogZold, LOW);
    digitalWrite(gyalogPiros, HIGH);

    lcd_1.clear();
    lcd_1.setCursor(0, 0);
    lcd_1.print("Kerem varjon...");

    // Autók visszarendezése
    digitalWrite(autoPiros, LOW);

    digitalWrite(autoSarga, HIGH);
    delay(3000);
    digitalWrite(autoSarga, LOW);

    digitalWrite(autoZold, HIGH);
}

delay(200); // pergésgátolás
}

elozoAllapot = jelenlegi; // gomb előző állapota
}

```