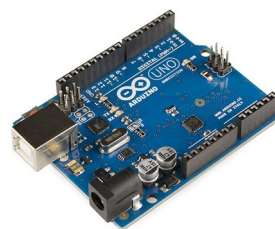


Milyen csodabogár az a mikrovezérlő?

A mikrovezérlő egy tenyérnyi, programozható, elektronikus eszköz, amely képes más elektronikus eszközöket vezérelni. Egyfajta miniszámítógép, amely a rá telepített programnak megfelelően képes érzékelni a külvilágból érkező jeleket (nyomógombok, fények, hangok, hőmérséklet...) és szabályozni a rá kapcsolt eszközök működését. (Például lámpákat, hangszórókat, motorokat.)

A mikrovezérlőre építve különféle elektronikus készülékeket állíthatunk össze, az egyszerű villogó, csipogó áramköröktől a bonyolultabb hobbrobotokig. Eközben szinte játszva ismerhetjük meg az elektronika és a számítógépes programozás alapjait.



Mire lesz szükségünk kezdetben?

Az első áramkörök összeállításához a következő alkatrészekre lesz szükség:

Egy **Arduino mikrovezérlő**. Többféle típus létezik, amelyek különféle tudásúak. Az egyik legelterjedtebb az Arduino Uno.

Egy **USB adatkábel**. (Olyan, mint a nyomtatóhoz való.) Ezen át juttatjuk a programot a számítógépről a mikrovezérlőre, illetve ezen át kapja az áramot a mikrovezérlő. (9V-os elemről is kaphatja.)

Egy **Számítógép**. Windows, Mac OS vagy Linux operációs rendszerű egyaránt jó lesz.

Az **Arduino fejlesztő-környezet**. (Arduino IDE) Letölthető a www.arduino.cc oldalról. (Létezik letöltés nélkül használható, online változata is: Arduino Web Editor)

Egy **próbapanel** (ún. breadboard), amely az alkatrészeket összeköti. A próbapanel arra jó, hogy különféle áramköröink összeállításakor ne kelljen forrasztópákával veszélyeztetnünk az alkatrészek (és az asztal) épségét. A különféle alkatrészeket és az összekötő kábeleket a lyukakba dugdosva állíthatjuk össze a kísérleteinket.

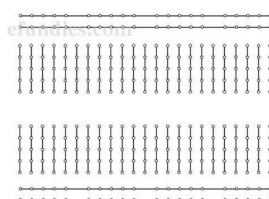
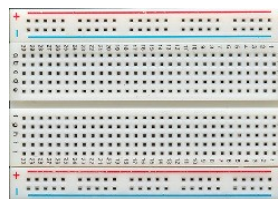
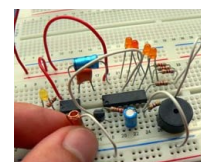
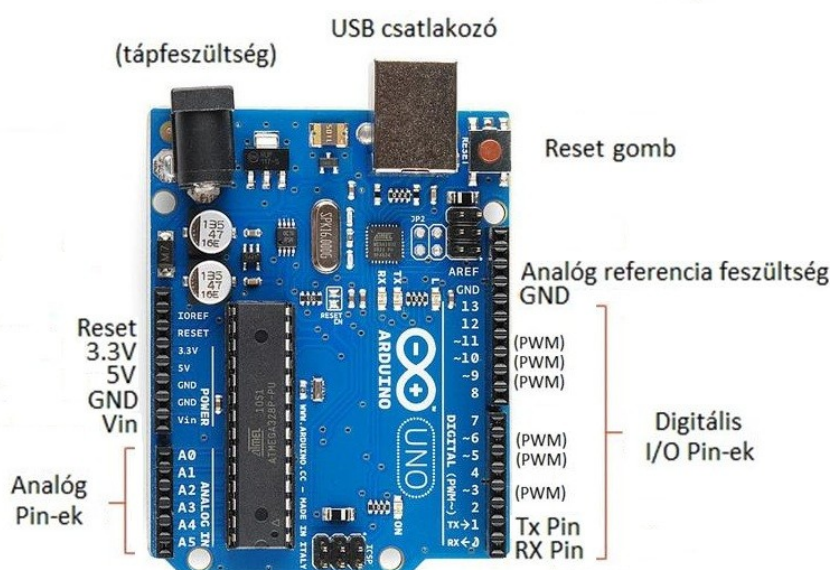
A próbapanel temérdek lyuka közül némelyek gyárilag össze vannak kötve elektromos vezetékkel. (Lásd a jobb szélső ábrát.)

Átkötőkábelek. Rövid, vékony, tömör végű (nem sodrott) vezetékek, amiket a próbapanel lyukaiba és a mikrovezérlő kivezetéseibe dughatunk.

Elektronikai ellenállások. Ezek az alkatrészek arra szolgálnak, hogy a rajtuk átfolyó elektromos áram erősségét csökkentsék. Az ellenállás nagyságának mértékegysége az Ohm. (Jele a görög ábécé omega betűje: Ω) Ha nem ismerjük a készletünkben lévő ellenállások nagyságát, akkor határozzuk meg a rajtuk lévő színes csíkok segítségével, vagy mérjük meg multiméterrel és jegyezzük fel!

Az áramköri rajzokon az ellenállások jele a következő:  vagy: 

LED-ek, avagy *fénykibocsájtó diódák*. A LED első ránézésre egy mini izzólámpának tűnik. Valójában csak annyiban hasonlít egy hagyományos izzólámpához, hogy fényt képes kibocsájtani. Az izzólámpával ellentétben csak egy irányba engedi át az elektromos áramot! Csak ha az áramforrás pozitív pólusát kötjük a LED hosszabbik (anód) kivezetésére, a negatív pólust a rövidebbikre (katód), akkor világít. Ellenkező irányú bekötés esetén nem engedi át az áramot.



Fontos tudni, hogy a LED (akárcsak a hagyományos izzólámpa) nem képes elviselni akármekkora elektromos áramot! Mivel a LED saját ellenállása kicsi, ezért egy megfelelő méretű ellenállást kell sorba kötni vele. A LED által kibocsájtott fény erőssége függ a rajta átfolyó áram erősségétől. Nagyobb áram esetén fényesebben világít, túl kis áram esetén nem kapcsol be, túl nagy áram esetén kiég. Az áramerősséget a diódával sorba kötött ellenállás nagyságával, vagy a rákötött elektromos feszültséggel lehet szabályozni.

Az áramköri rajzokon a LED-ek jele a következő:



Elektronikai **mérőműszer** (ún. multiméter). Elektromos feszültséget, áramerősséget és ellenállást lehet mérni vele. (Multimétert beszerezniük nem feltétlenül szükséges.)

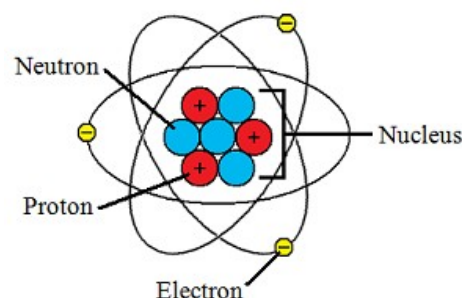


Az elektromos áram

Kísérleteink láthatatlan, ámde mégis legfontosabb "alkatrésze" maga az elektromos áram. De mi is valójában ez a titokzatos "anyag", ami áramköreinkben "csordogálva" működteti készülékeinket?

A természetben található különféle anyagok mind ugyanabból a háromféle elemi részecskéből állnak:

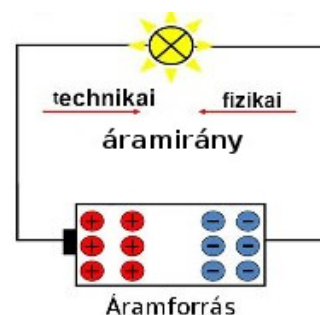
- pozitív töltésű *protonokból*,
- negatív töltésű *elektronokból*,
- és elektromosan semleges *neutronokból*.



A protonok és a neutronok egy helyben csücsülnek (ez az ún. *atommag*, latinul *nucleus*), az elektronok pedig körülöttük keringenek, szaladgálnak. Bizonyos anyagokban - ezeket nevezzük szigetelőknél - az elektronok nem hajlandók messze eltávolodni a protonoktól, viszont másféle anyagokban - ezeket nevezzük elektromos vezetőknél - az elektronok képesek elhagyni a protonok és neutronok szomszédságát és vándorútra kelni. Persze ezt csak akkor teszik, ha az anyagot elektromos erőterbe helyezzük. Ilyenkor az anyagban lévő sok-sok kicsiny elektron mind ugyanabba az irányba kezd el áramlani, ezt nevezzük elektromos áramnak.

Ezt az áramlást sokféle dologra használhatjuk: világíthatunk és fűthetünk vele, vagy információt továbbíthatunk. A különféle elektronikus építőelemeket vezetékekkel összekötve ún. *áramköröket* építhetünk. Ilyen áramkörök alkotják a körülöttünk lévő technikai világ különféle kűtyűit. Egyszerűbb áramköröket akár gyerekek is tudnak építeni, a bonyolultakat az erre szakosodott szakemberek, a villamosmérnökök tervezik.

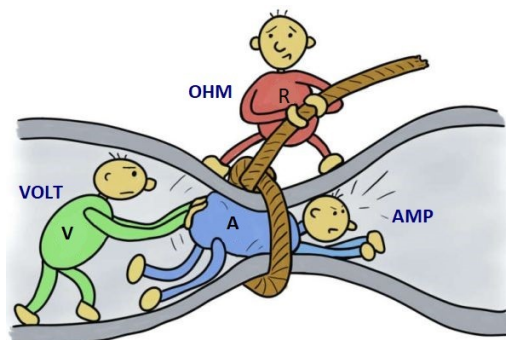
Az áramkörök építésénél az egyik legfontosabb kérdés **az áram iránya**. Fizikaórán azt tanultuk (vagy fogjuk majd tanulni), hogy az elektronok az áramforrás negatív "pólusától" áramlanak a pozitív pólus felé. Az áramforrás (például egy ceruzaelem) ugyanis úgy működik, hogy a negatív pólusán elektronokat bocsájt ki, a pozitív pólusán pedig elnyeli őket. Ezt az áramlási irányt nevezzük *fizikai* áramiránynak.



Viszont a mérnöki gyakorlatban az ún. *technikai* áramirányt használják, amely az áramforrás pozitív pólusától a negatív felé mutat. Ennek a gyakorlatnak az eredete a régmúlt időkbe nyúlik vissza, amikor még nem ismerték az elemi részecskék természetét, és az volt az elterjedt (tév)hit, hogy a vezetékben a pozitív töltésű részecskék áramlanak. Mi is ezt az irányt fogjuk használni a magyarázatokban.

Az elektromos feszültség, áramerősség és ellenállás

Amint arról már szó volt, az elektronok áramlását az elektromos erőter idézi elő. Ennek az erőternek az "erejét" jellemzi az elektromos *feszültség*. Minél nagyobb feszültségű az elektronokra ható erőter, annál inkább igyekeznek áramlani. Az elektromos feszültség jele az U betű, mértékegysége a Volt. (Ezt a mértékegységet a V betűvel szoktuk rövidíteni.)



Az elektromos áram folyásának intenzitását jellemzi az elektromos *áramerősség*. Minél erősebb az áram, annál nagyobb hatást vált ki azokon az eszközökön, amiken átfolyik. Az izzólámpa fényesebben világít, a fűtőszál jobban melegszik, az elektromágnes erősebben vonz és az élő szervezet jobban károsodik. Valóban, ha erős áram folyik át valakinek a testén, súlyos, akár halálos sérüléseket is szenvedhet. A fali konnektor nem játék! Az elektromos áramerősség jele az I betű, mértékegysége az Amper. (Ezt a mértékegységet az A betűvel szoktuk rövidíteni.)

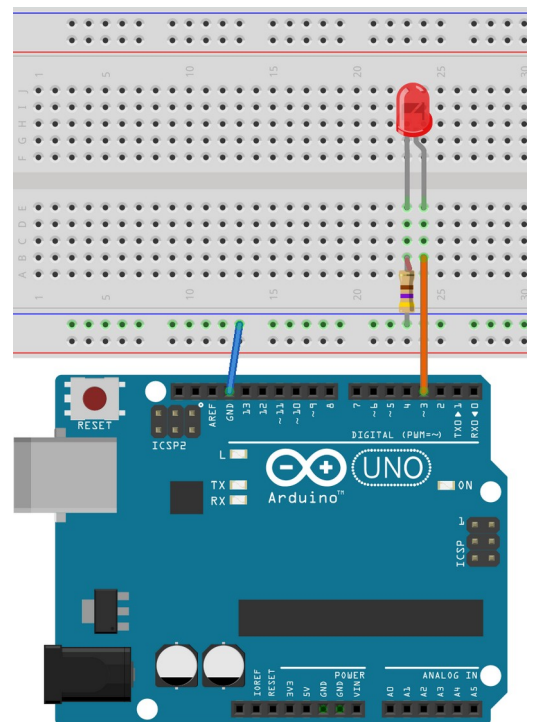
Az elektromos *ellenállás* az áramköri elemek jellemző tulajdonsága, amely azt jelzi, hogy mennyire akadályozzák az áram folyását. Minden áramköri elemnek van ellenállása, de olykor ez olyan kicsi, hogy elhanyagolhatjuk. A vezetékek ellenállását például nullának tekintjük. Az elektromos ellenállás jele az R betű, mértékegysége az Ohm. (Ezt a mértékegységet a görög Omega betűvel szoktuk rövidíteni: Ω)

Az áramkörben kialakuló elektromos áram erőssége alapvetően két dologtól függ: Az áramkörre kapcsolt áramforrás feszültségétől és az áramkört alkotó építőelemek *ellenállásától*. Minél nagyobb a feszültség, illetve minél kisebb az áramkör ellenállása, annál erősebb áram fog folyni az áramkörben. Erről szól az elektromosságtan egyik alapvető törvénye, az ún. Ohm-törvény: $I = U / R$

Pislog a Kütyü

Az első áramkörünk egy egyszerű villogó lesz. A kapcsolást a jobb oldali ábra mutatja. A LED-del sorba kell kötni egy 220-600 Ohm közé eső áramkorlátozó (védő) ellenállást. A LED polaritására ügyelni kell: a hosszabbik (pozitív, anód) lába legyen a mikrovezérlő 3-as digitális kimenetére kötve!

Kövessük nyomon az áram útját a „technikai” áramirány szerint: A mikrovezérlő 3-as kimenete lesz az áramforrás pozitív pólusa. Innen indulva az áram a piros vezetéken át a próbapanelbe jut. A próbapanel lyukai függőleges irányban össze vannak kötve, ezért az áram továbbhaladva eléri a LED pozitív (hosszabbik, anód) lábát. A LED-en keresztülhaladva az áram a próbapanel másik lyuksorába ér. Ezek a lyukak is függőlegesen vannak összekötve, így az áram eléri az ellenállást. Az ellenálláson áthaladva az áram a próbapanel kék csíkkal jelölt, vízszintes irányban összekötött lyuksorába érkezik és a kék vezetéken át végül eléri az áramforrás negatív pólusát, azaz a mikrovezérlő GND (ground = föld) kimenetét.



Az összeszerelt áramkör természetesen még nem működik, hiszen egyelőre nem kap áramot és a mikrovezérlő sem tudja, hogy mit kell tennie. (Azaz nem töltöttük fel rá a vezérlő programot.)

Az Arduino fejlesztőkörnyezet telepítése

Az Arduino fejlesztőkörnyezet (Integrated Development Environment, IDE) letölthető a következő Webhelyről: <https://www.arduino.cc> A telepítés végén, az első indításkor engedélyezzük az Internet-elérést a szoftvernek! (Erre azért van szükség, hogy automatikusan telepíteni tudja a frissítéseket és programkönyvtárakat.)

Létezik telepítést nem igénylő hordozható változat is, amit a letöltés után csak ki kell csomagolni.

Létezik továbbá felhő-alapú, online használható verzió is, ennek használatához azonban regisztrálnunk kell. (Arduino Web Editor) Az online verzió előnye az, hogy projektjeinket később bárhol elérhetjük.

A fejlesztői környezetet elindítva – jó esetben – magyar nyelvű felület fogad. (Amennyiben angol nyelven indul a program, úgy a *File/Preferences/Language* menüpontban állíthatjuk át magyar nyelvűre.)

Az IDE telepítése után érdemes kipróbálni a rendszer működését. Ehhez az USB kábel segítségével kapcsoljuk rá a mikrovezérlőt a számítógépre! A fejlesztőrendszer ablakában felül látható „Alaplap választás” (Select Board) rubrikában tudjuk kiválasztani azt, hogy milyen típusú mikrovezérlőt használunk és melyik portra csatlakoztattuk.

Ezután a *Fájl/Példák/01.Basics/Blink* menüpontot kiválasztva egy új ablak nyílik, amelyben a „Blink” nevű, egyszerű példaprogram látható. Töltsük fel azt a programot a mikrovezérlőre a jobb oldalon látható gombra kattintva!



Ilyenkor az IDE lefordítja az ablakban látható programot gépi kódra (a mikrovezérlő saját „nyelvére”) és feltölti a vezérlő memóriájába. Ha minden jól működik, akkor a fejlesztőrendszerben először a „Fordítás kész”, majd a „Feltöltés kész” üzeneteket láthatjuk, a mikrovezérlőn pedig elkezd villogni egy piciny LED, egy másodperces időközönként.

Az első program

A próbapanelen megépített saját LED-es villogónk életre keltéséhez az alábbi programot kell az Arduino mikrovezérlőbe programozni. A programozás különös figyelmet kíván! Minden egyes betű (karakter) számít, sőt a programozási nyelv a kis- és nagybetűkre is érzékeny! A // jelektől jobbra lévő szövegeket nem kell begépelni, azok csak magyarázatul szolgálnak.

```
void setup() {  
    pinMode(3, OUTPUT);    // a 3-as csatlakozó kimeneti módba állítása  
}  
  
void loop() {  
    digitalWrite(3, HIGH);    // a LED bekapcsolása  
    delay(1000);              // várakozás 1 másodpercig  
    digitalWrite(3, LOW);     // a LED kikapcsolása  
    delay(1000);              // várakozás 1 másodpercig  
}
```

A program begépelése után kattintsunk a Feltöltés gombra! Ha mindent jól csináltunk, a próbapanelbe dugott LED pislogni kezd.



A program működésének magyarázata

A program elején a **setup** részben a **pinMode(3, OUTPUT)** paranccsal kimenetnek állítjuk be a 3-as csatlakozót, amelyikre a LED-et kötöttük. (A csatlakozók eredetileg bemeneti módba vannak állítva.)

A **loop** részben a LED csatlakozójára a **digitalWrite(3, HIGH)** utasítással +5V feszültséget kapcsolunk, ettől a LED áramot kap és világítani kezd. Ezután a **delay(1000)** utasítással várakoztatjuk a vezérlőt egy másodpercig (1000ms), majd a csatlakozóra a **digitalWrite(3, LOW)** utasítással 0V-ot kapcsolunk (azaz kikapcsoljuk a LED-et). Végül ismét egy másodperc várakozás következik. Erre az utóbbi várakozásra azért van szükség, mert a **loop** programrész végrehajtása újra kezdődik és a végtelenségig ismétlődik.

Az Arduino programok felépítése

A mikrovezérlőt a C++ nevű, széles körben elterjedt programozási nyelven lehet programozni. (A mikrovezérlővel való „játék” során megszerzett programozási ismeretek ezért később az informatika sok más területén is hasznosíthatók.)

Az Arduino programokat vázlatnak/skicc-nek (sketch) hívják. A program nem más mint a mikrovezérlő által végrehajtandó utasítások sorozata. Minden Arduino program három fő részből áll:

1. **Változók megadása.** A programban használt változók *típusát* és *nevét* kell itt megadni.
Változókat használhatunk különféle adatok ideiglenes tárolására. (Az előző programban ilyen nem volt.)
2. **Kezdeti beállítások: `setup(){ }`** A kapcsos zárójelekben belül lévő programkód az Arduino bekapcsolása után csak egyetlen egyszer fut le, mielőtt a programunk ismétlődő része elindulna. Itt adjuk meg például, hogy mely pin-ek legyenek ki- vagy bemenetek, a soros porti kommunikáció sebességét, stb. (Alapértelmezetten minden pin bemenet, ezért általában csak a kimeneteket kell beállítanunk.)
3. **Ismétlődő programrész: `loop(){ }`** Itt adjuk meg a mikrovezérlőnek, hogy mit csináljon folyamatosan. Ez a rész a *setup* rész lefutása után indul, és folyamatosan ismétlődik mindaddig, amíg az Arduino be van kapcsolva. Minden egyes alkalomkor, amikor a program végrehajtása a *loop* rész végére ér, átugrik a *loop* rész elejére és kezdi újra előlről.

Eddig a következő utasításokat ismertük meg:

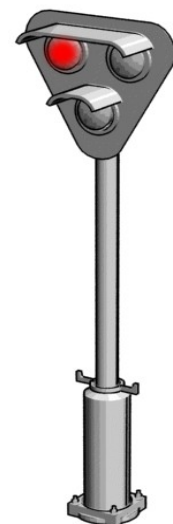
`pinMode(csatlakozó, mód);` - A csatlakozók adatirányát állíthatjuk be ezzel az utasítással.
(OUTPUT: kimenet, INPUT: bemenet)

`digitalWrite(csatlakozó, érték);` - Ha egy csatlakozót kimenetnek állítottunk be, akkor ezzel az utasítással az állapotát magasra (HIGH: +5V feszültség), vagy alacsonyra (LOW: 0V feszültség) állíthatjuk.

`delay(idő);` - A zárójelek között megadott ideig várakoztatjuk a programot. Az időt ezredmásodpercben (milliszekundum) kell megadni.

Gyakorló feladatok

1. Alakítsuk át úgy a programot, hogy az SOS jelzést adja le a készülék! (Morze-jelekkel)
2. Egészítsük ki még egy LED-del az áramkört! Villogtassuk a LED-eket felváltva!
3. Készítsünk közlekedési jelzőlámpát autóknek! (Piros, sárga és zöld LED-ek.)
4. Egészítsük ki az autós jelzőlámpát gyalogos jelzőlámpával is! (Egy piros és egy zöld LED.)
5. Készítsünk fénysorompót! (A vasúti átjáróban lévő jelzőlámpát.)
6. Készítsünk futófényt! (Sok LED egymás mellett, a LED-soron végigfut a fény.)



Megoldási útmutató

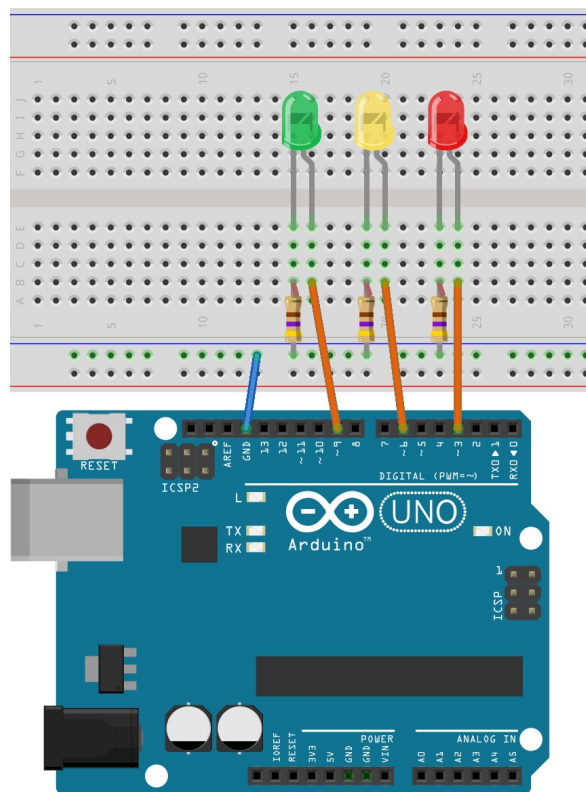
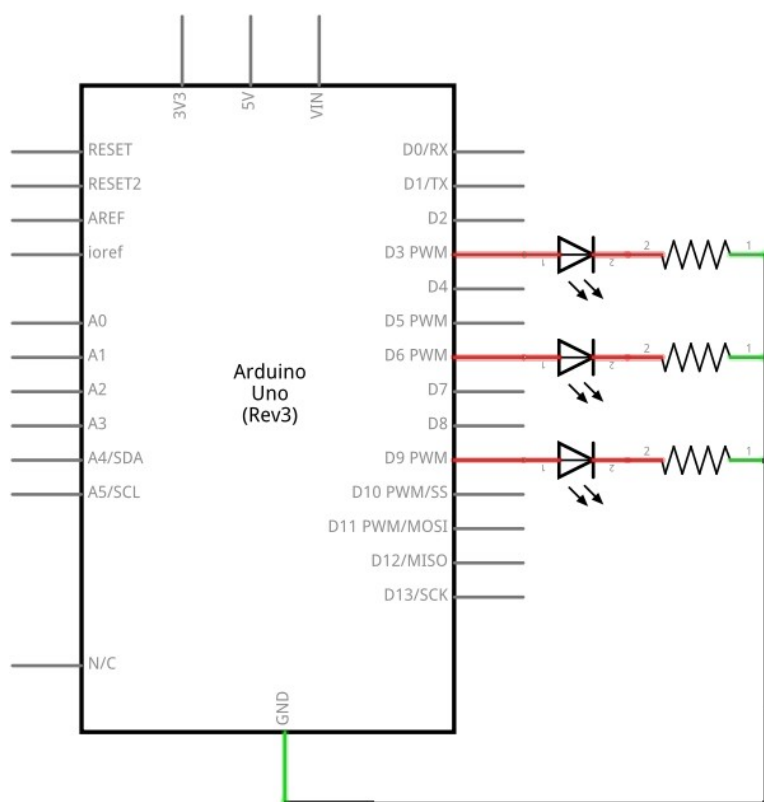
Az SOS jelzés leadásához másoljuk le a program **loop** részében lévő 4 utasítást 9 példányban és írjuk át a késleltetési időket. (Az S betű Morze jele 3 rövid villanás, az O betű jele pedig 3 hosszú villanás.)

Íme a program:

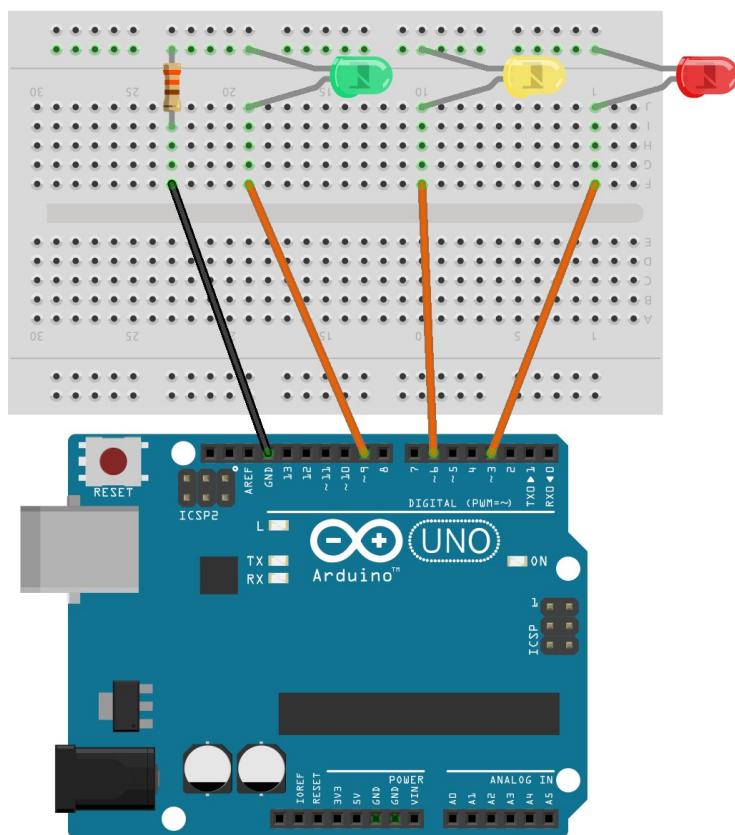
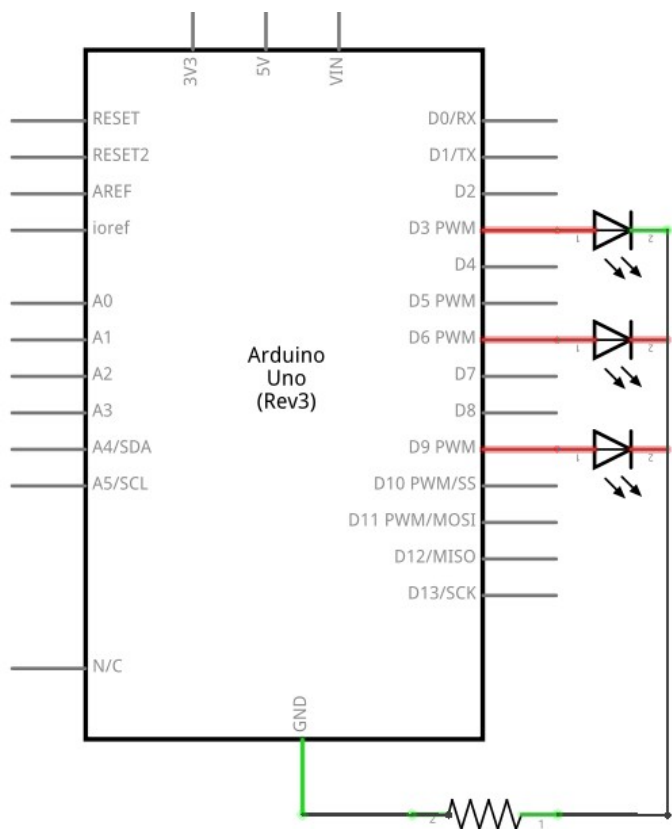
```
void setup() {  
    pinMode(3, OUTPUT);    // a 3-as csatlakozó kimeneti módba állítása  
}  
  
void loop() {  
    digitalWrite(3, HIGH);    // a LED bekapcsolása  
    delay(500);               // várakozás fél másodpercig  
    digitalWrite(3, LOW);     // a LED kikapcsolása  
    delay(500);               // várakozás fél másodpercig  
  
    digitalWrite(3, HIGH);  
    delay(500);  
    digitalWrite(3, LOW);  
    delay(500);  
  
    digitalWrite(3, HIGH);  
    delay(500);  
    digitalWrite(3, LOW);  
    delay(500);  
  
    digitalWrite(3, HIGH);    // a LED bekapcsolása  
    delay(1000);              // várakozás egy másodpercig  
    digitalWrite(3, LOW);     // a LED kikapcsolása  
    delay(1000);              // várakozás egy másodpercig  
  
    digitalWrite(3, HIGH);  
    delay(1000);  
    digitalWrite(3, LOW);  
    delay(1000);  
  
    digitalWrite(3, HIGH);  
    delay(1000);  
    digitalWrite(3, LOW);  
    delay(1000);  
  
    digitalWrite(3, HIGH);    // a LED bekapcsolása  
    delay(500);               // várakozás fél másodpercig  
    digitalWrite(3, LOW);     // a LED kikapcsolása  
    delay(500);               // várakozás fél másodpercig  
  
    digitalWrite(3, HIGH);  
    delay(500);  
    digitalWrite(3, LOW);  
    delay(500);  
  
    digitalWrite(3, HIGH);  
    delay(500);  
    digitalWrite(3, LOW);  
    delay(500);  
}
```

A közlekedési jelzőlámpa többféleképpen is megépíthető.

Íme az egyik megoldás kapcsolási rajza és egy lehetséges megvalósítása a próbapanelen:



Egy másik megoldáshoz elegendő egy ellenállást használni:



Ennél a megoldásnál viszont nagyon kell vigyáznunk a LED-ek helyes bekötésére! Ha az egyik LED-et fordítva kötjük be, akkor rövidzárlat alakulhat ki, azaz olyan áramkör, amelynek túl kicsi az ellenállása. A rövidzárlat tönkretelheti a LED-et, vagy akár a mikrovezérlőt is! Biztosabb, ha a későbbiekben az első módszerrel építjük meg az áramköreinket, azaz minden LED-del sorba kötünk egy védőellenállást.

LED-ek bekötése közös védőellenállással

Amikor sok LED-et használunk Arduino-alapú áramkörünkben, előfordulhat, hogy kifogyunk az ellenállásokból. Ilyenkor lehetőség van arra, hogy közös védőellenállást alkalmazzunk a LED-ekhez, de ez a megoldás fokozott óvatosságot igényel, továbbá hatással van a LED-ek fényességére is.

Tekintsünk például egy három LED-et tartalmazó áramkört, amelyben minden LED külön védőellenállással van bekötve!

Amennyiben az egyik LED-et (például a sárgát) fordítva kötjük be, akkor a piros LED bekapcsolásakor azt tapasztaljuk, hogy a sárga LED is felkapcsolódik. Ha ugyanis a mikrovezérlő 3-as kimenete HIGH állapotban van, a 6-os kimenet pedig LOW-ban, akkor áram fog folyni a 3-as kimenettől a 6-os felé. Ez az áram 2 ellenálláson is áthalad, ezért a piros és sárga LED kissé halványabban fog világítani, mint normál esetben.

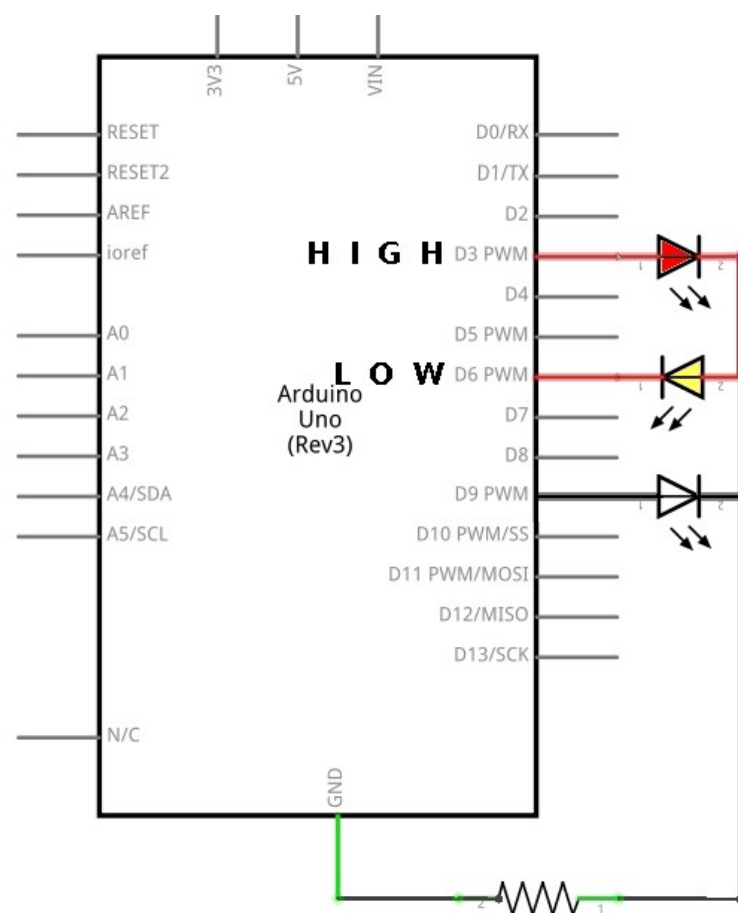
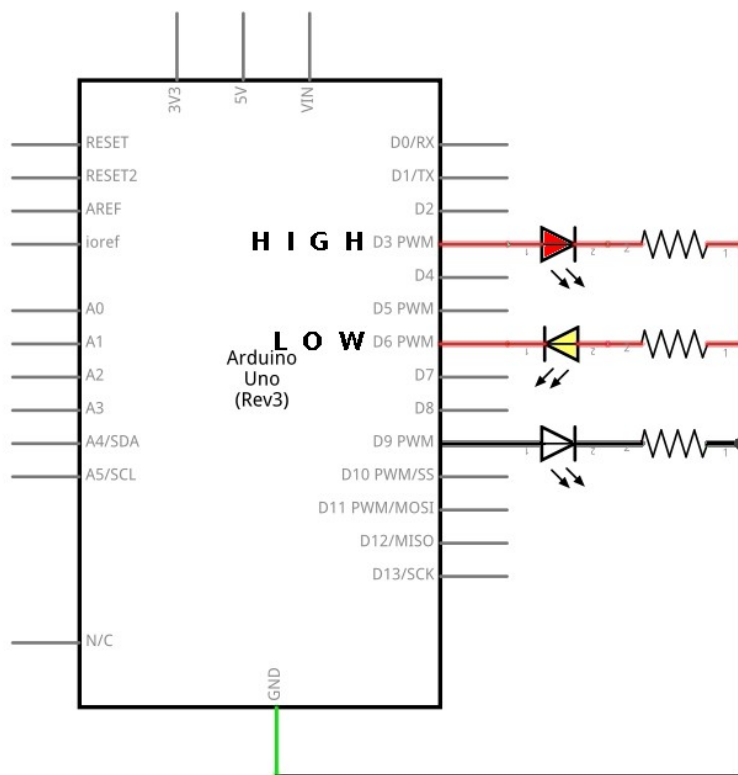
A sárga LED helyes bekötése esetén azért nem tud kialakulni ez az áramkör, mert a LED-ek csak egy irányba engedik át az áramot. (Az anódtól, azaz a hosszabbik lábuktól a katód, azaz a rövidebbik láb felé. Az áramköri jelük is ezért „nyílhegy” alakú.)

Közös védőellenállás használata esetén az egyik led fordított irányú bekötése azonban rövidzárlatot eredményezhet, amitől a LED-ek és a mikrovezérlő is tönkremehet!

Ha a mikrovezérlő 3-as kimenete HIGH állapotban van, a 6-os kimenet pedig LOW-ban, akkor nagyon erős áram fog folyni a 3-as kimenettől a 6-os felé, ugyanis a kialakuló áramkörben nincs védőellenállás!

A LED-ek helyes bekötése esetén persze ez az áramkör sem tud kialakulni, mert a sárga LED nem engedi át az áramot.

Aki már kipróbálta a LED-ek közös védőellenállással való bekötését észrevehette, hogy minél több LED-et kapcsolunk be, annál halványabban világítanak. Ennek a jelenségnek az az oka, hogy az újabb LED-ek bekapcsolásával az áramkör eredő ellenállása nem változik, ezért a GND kimeneten (és a közös védőellenálláson) ugyanakkora áram folyik át, viszont ez az áram az egyes LED-ekhez menő vezetéseken szétoszlik.



Külön védőellenállások használata esetén ugyan szintén szétoszlik az áram a LED-ek vezetékein, azonban minél több LED-et kapcsolunk be, annál kisebb lesz az áramkör eredő ellenállása.

Ha egy LED helyett kettőt kapcsolunk be, akkor az eredő ellenállás a felére csökken, ezért a GND kimeneten dupla akkora áram folyik át, ami kettéoszlik a két LED vezetékein. Az egyes LED-ekre tehát ugyanannyi áram jut, akármennyi LED van bekapcsolva.

Természetesen a mikrovezérlőnek is megvannak a maga korlátai: egy-egy digitális kimeneten legfeljebb 20 mA áramot képes kiadni (és elviselni), a GND kimenetén pedig legfeljebb 200 mA-t. Az egyes áramköri ágakban folyó áram erősségét a védőellenállások nagyságával tudjuk szabályozni. Minél nagyobb a védőellenállás, annál kisebb áram folyik át rajta.

A 20 mA maximális áram elérésével inkább ne kísérletezzünk, ugyanis a hozzá szükséges védőellenállás értékét nehéz pontosan kiszámolni! (A különféle színű és típusú LED-ek ugyanis más-más feszültségesést eredményeznek az áramkörben.) Azért sem érdemes túlságosan megközelíteni a 20 mA-t, mert akkor már 10 LED esetén elérjük a mikrovezérlőből összesen kivehető 200 mA-es határt.