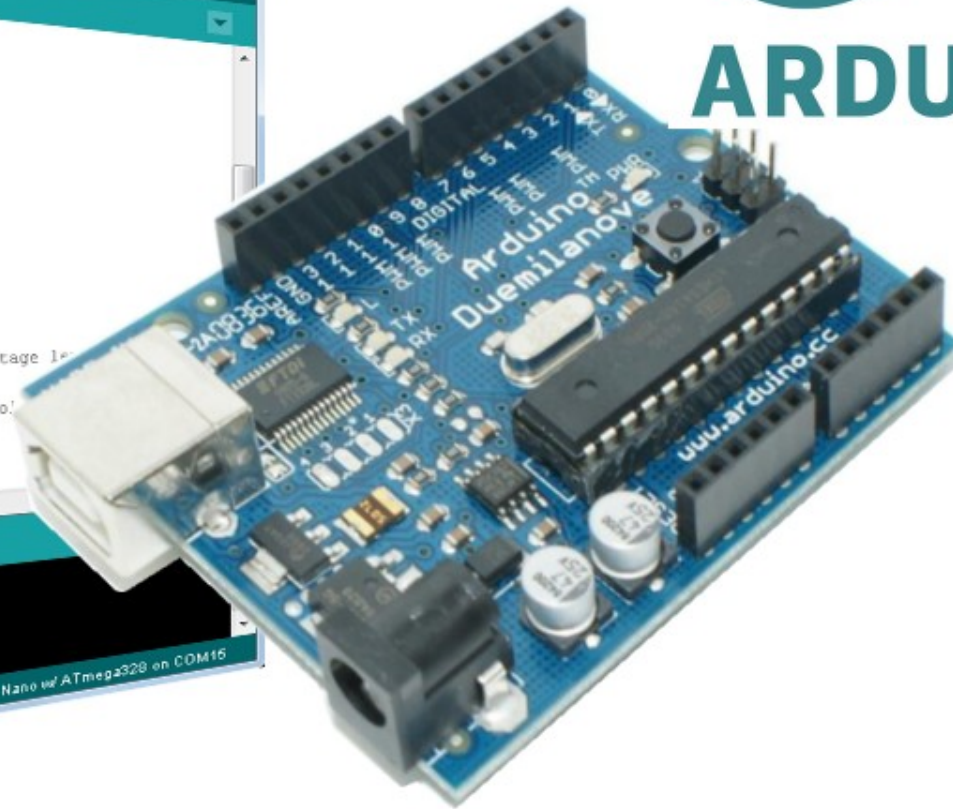
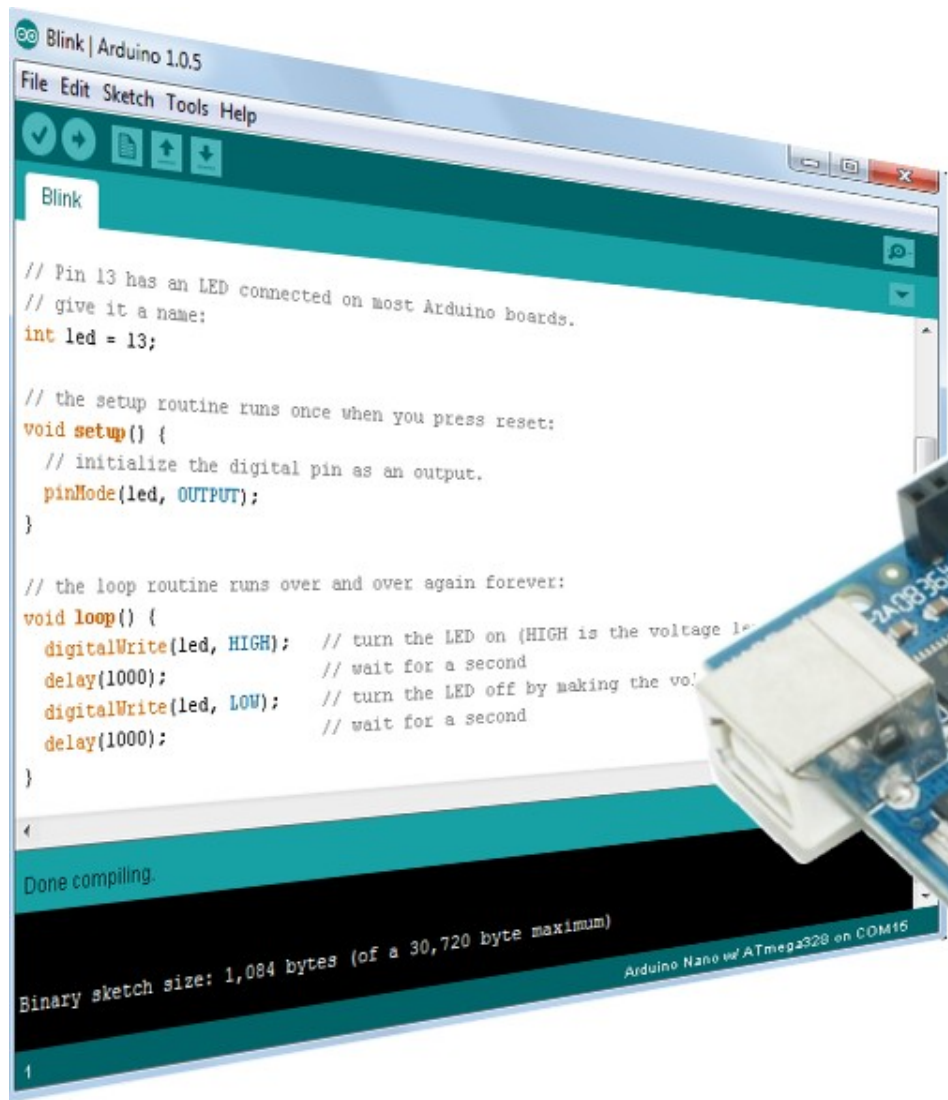


Arduino tanfolyam kezdőknek és haladóknak



13. Fényérzékelés, motorvezérlés, a robotika alapjai

Ajánlott irodalom

- ❑ Aduino LLC.: [Arduino Language Reference](#)
- ❑ ATMEL: [ATmega328p mikrovezérlő adatlapja](#)
- ❑ Brian W. Kernighan, Dennis Ritchie: [A C programozási nyelv](#)
- ❑ Cseh Róbert: Arduino programozási kézikönyv
- ❑ Harsányi Réka – Juhász Márton András:
[Fizikai számítástechnika: elektronikai alapok és Arduino programozás](#)
- ❑ Ruzsinszki Gábor: Mikrovezérlős rendszerfejlesztés C/C++ nyelven I. – PIC mikrovezérlők
- ❑ Ruzsinszki Gábor: Mikrovezérlős rendszerfejlesztés C/C++ nyelven II. – Arduino

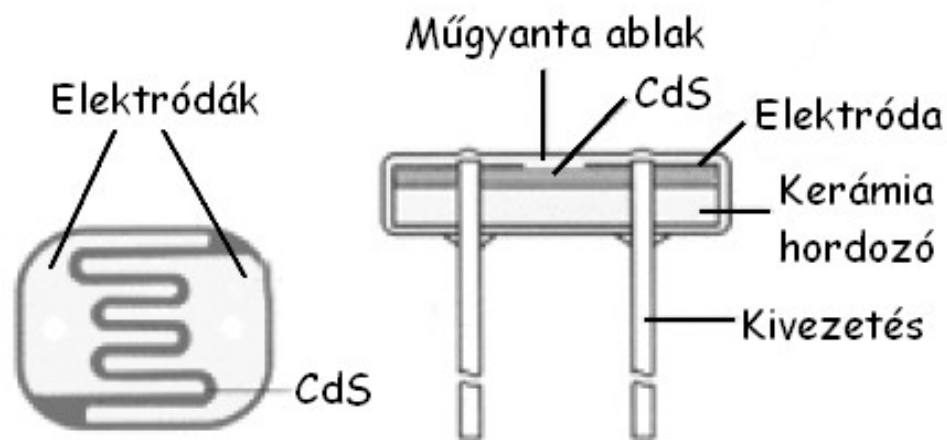
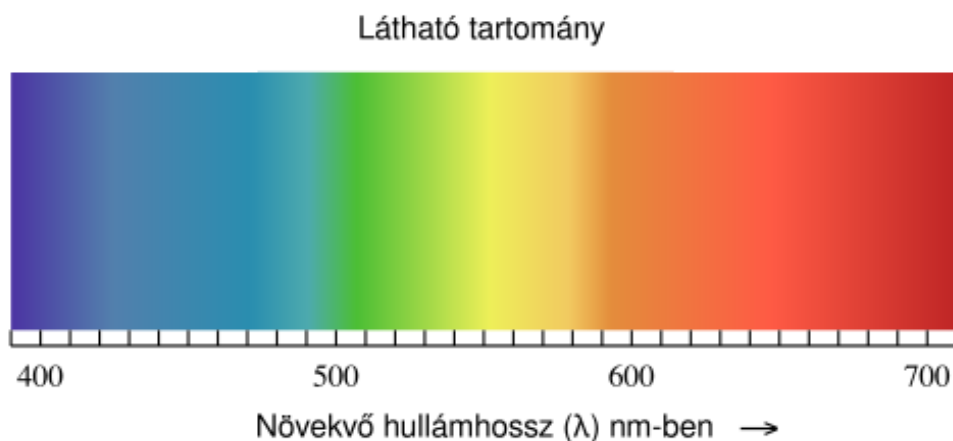
Témák és mintaprogramok

- **LDR_test.ino** – CdS fényérzékeny ellenállásokkal már többször is találkozhattunk előadásainkon, ezzel az egyszerű mintaprogrammal fényt mérünk, s a fénymérőnket megpróbáljuk kalibrálni egy luxmérő segítségével
- **tcrt5000_test1.ino** – visszavert IR fény mérése **TCRT5000** szenzorral, környezeti hatások korrekciója nélkül
- **tcrt5000_test2.ino** – visszavert IR fény mérése **TCRT5000** szenzorral, környezeti hatások korrekciójával
- **L293D_test_1M.ino** – egy DC motor vezérlése **L293D** IC-vel
- **L293D_test_2M.ino** – két DC motor vezérlése **L293D** IC-vel
- **motor_test.ino** – a kétkerék meghajtású robot motorjainak ellenőrzése (**L298N** motorvezérlő kártyával)
- **robot_with_tcrt5000.ino** – **TCRT5000** szenzor alkalmazása ütközésgátlóként (akadály észlelésekor leállítja a motort)

CdS fényérzékeny ellenállás

A GL55 típusú kadmiumsulfid (CdS) anyagú ellenállások vezetőképessége a fény hatására növekszik.

A spektrális érzékenység maximuma 540 nm (sárgászöld).



Típus	Sötét-ellenállás	Ellenállás @ 10 lx	γ	Válaszidő
GL5528	1 M Ω	10 – 20 k Ω	0.6	20 – 30 ms

$$\gamma = \lg \left(\frac{R_{10}}{R_{100}} \right)$$

R10 és R100 a 10 és 100 lux-nál mért ellenállás értékek.

Fotometriai alapfogalmak

Fényforrás fényerőssége: az egységnyi térszögbe kisugárzott fényteltjesítmény (ref 555 nm). Mértékegysége: 1 cd (= 1W/683 sr), kb. 1 gyertya fényerőssége. Egy 100 W izzó fényerőssége kb. 120 cd.

Fényáram: a fényerősség és a kisugárzási térszög szorzata. Mértékegysége: 1 lm (= 1 cd · 1 sr) Egy 100 W izzólámpa fényárama kb. 1380 lumen.

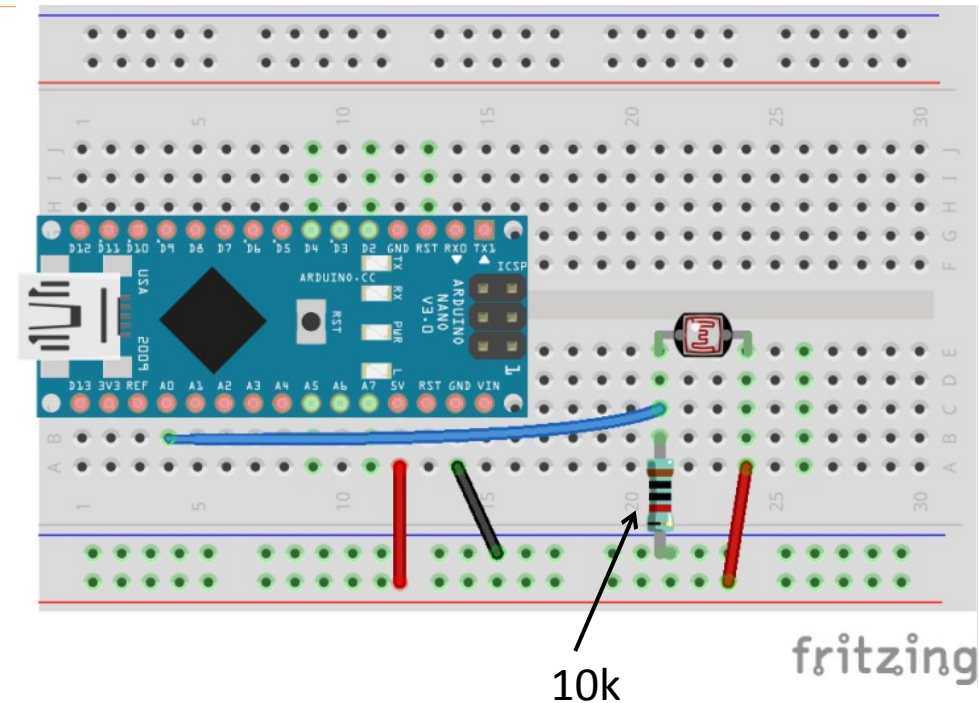
Megvilágítás erőssége: az adott felületre eső fényáram és a felület nagyságának hánydosa. Mértékegysége: 1 lx (1 lux a megvilágítása annak a felületnek, amelynek 1 négyzetméterére merőlegesen és egyenletesen 1 lumen fényáram esik). Egy 100 W izzó 1,5 m távolságból kb. 100 lx megvilágítást eredményez.

Környezet	Megvilágítás (lux)
Iroda	500
Folyosó	50
Napfény nyáron	100 000
Napfény télen	10 000
Telihold	0,2
Színérzékelés határa	3

LDR_test.ino

Az LDR és a 10 kΩ-os ellenállás feszültségosztót alkot. Az osztó feszültségét az ADC segítségével megmérve, az LDR ellenállása meghatározható

```
void setup() {  
  Serial.begin(9600);  
  analogReference(DEFAULT);  
}  
  
void loop() {  
  int sensorValue = analogRead(A0);  
  Serial.print("A0 = ");  
  Serial.print(sensorValue);  
  Serial.print(" V0: ");  
  float voltage = sensorValue*4.66/1023;  
  Serial.print(voltage,3);  
  float Rx = 10000.0;  
  if(sensorValue > 0) {  
    Rx = 46.6/voltage - 10;  
  }  
  Serial.print(" V Resistance = ");  
  Serial.print(Rx,1);  
  Serial.println(" kOhm");  
  delay(2000);  
}
```



Itt 4,66 V a mért tényleges referencia feszültség (5 V helyett).

Rx kiszámítása:

$$voltage = 4,66 V * 10k / (Rx + 10k)$$

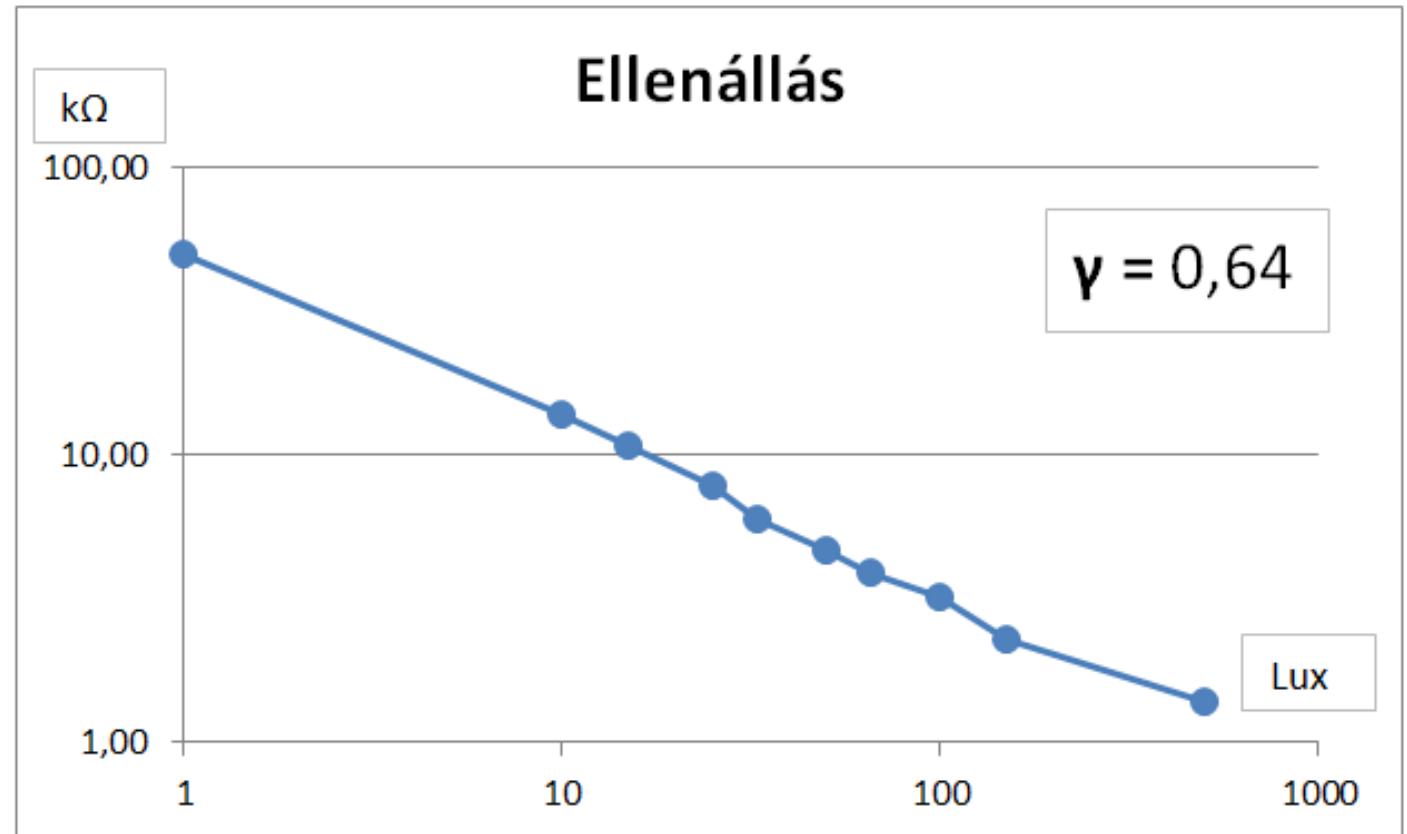
Ha Rx-et kifejezzük, akkor:

$$Rx = 4,66 V * 10k / voltage - 10k$$

A fotoellenállás kalibrálása

Feladat: változtassuk a megvilágítást, és mérjük meg az LDR ellenállását a megvilágítás függvényében! (Luxmérő alkalmazást találunk az okostelefonokhoz is)

Megvilágítás	Ellenállás
1	50,00
10	14,00
15	10,90
25	7,80
33	6,00
50	4,70
65	3,90
100	3,20
150	2,30
500	1,40



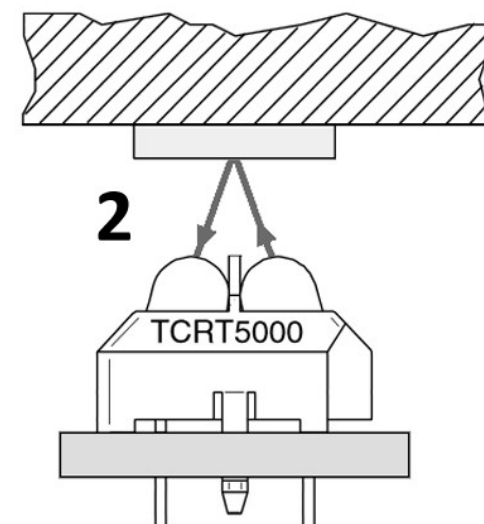
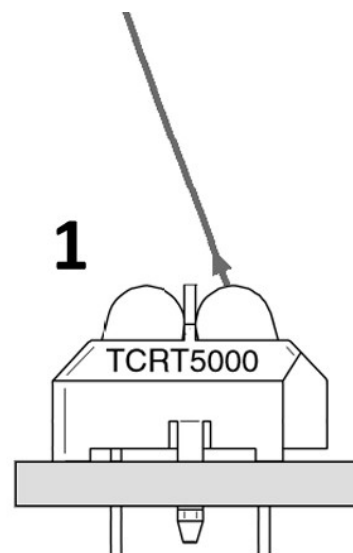
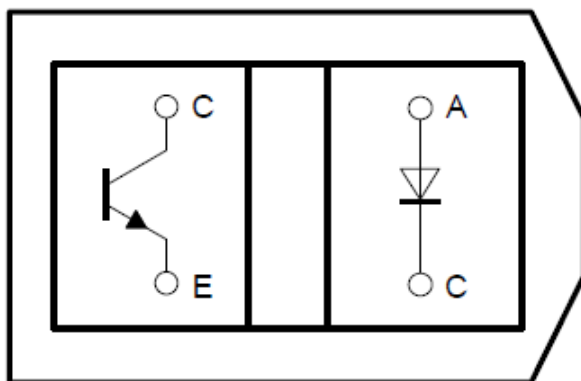
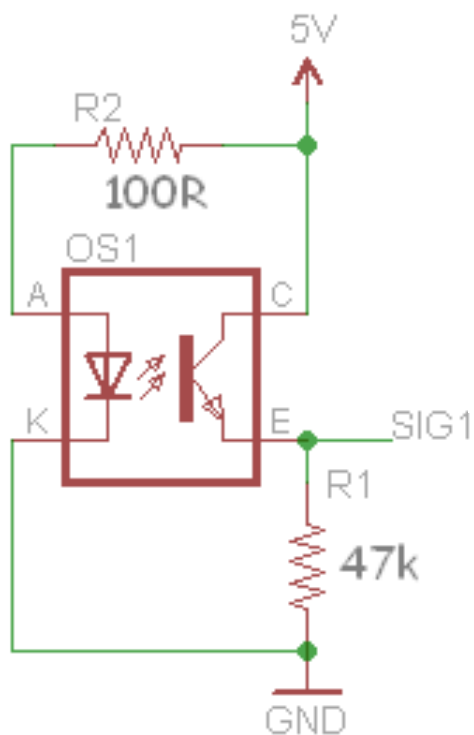
Mindkét skála logaritmikus!

TCRT5000 reflektív optikai érzékelő

Egy IR LED-et és egy fototranzisztort tartalmaz.

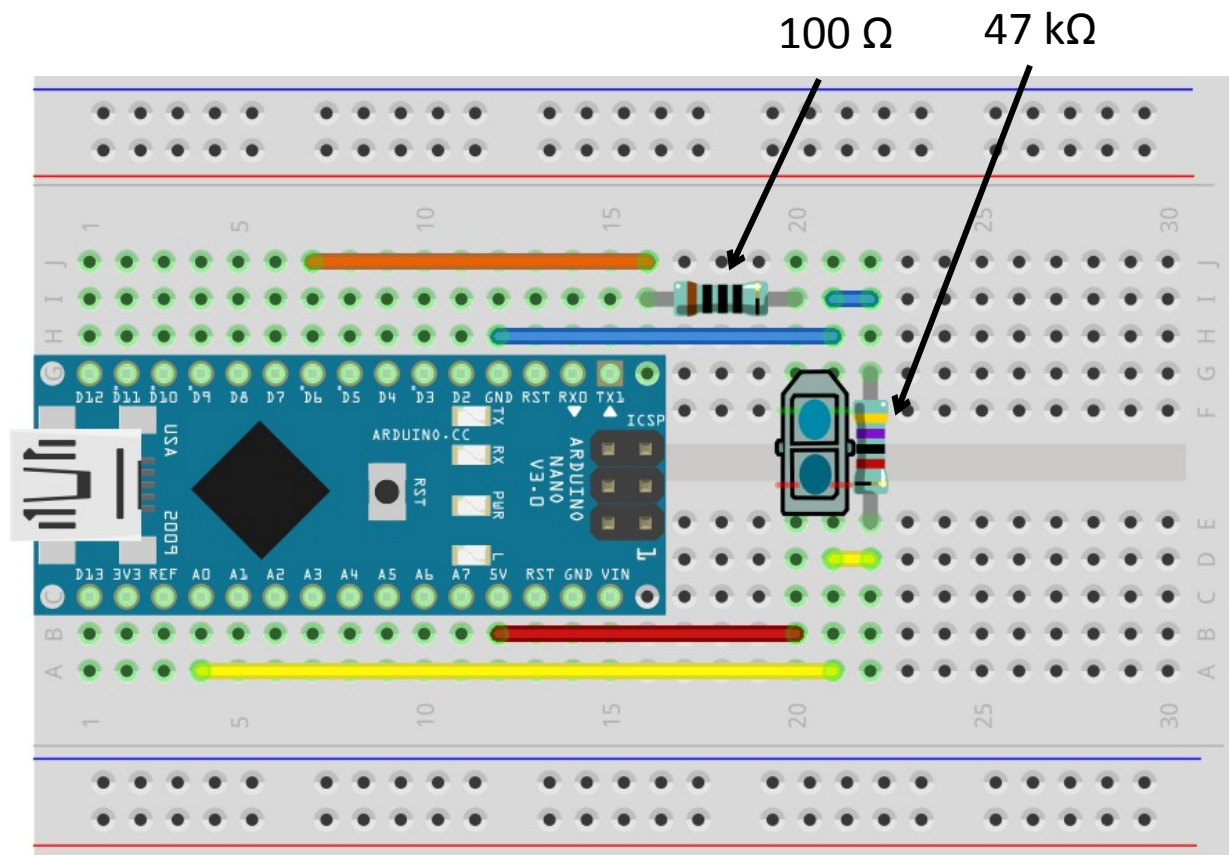
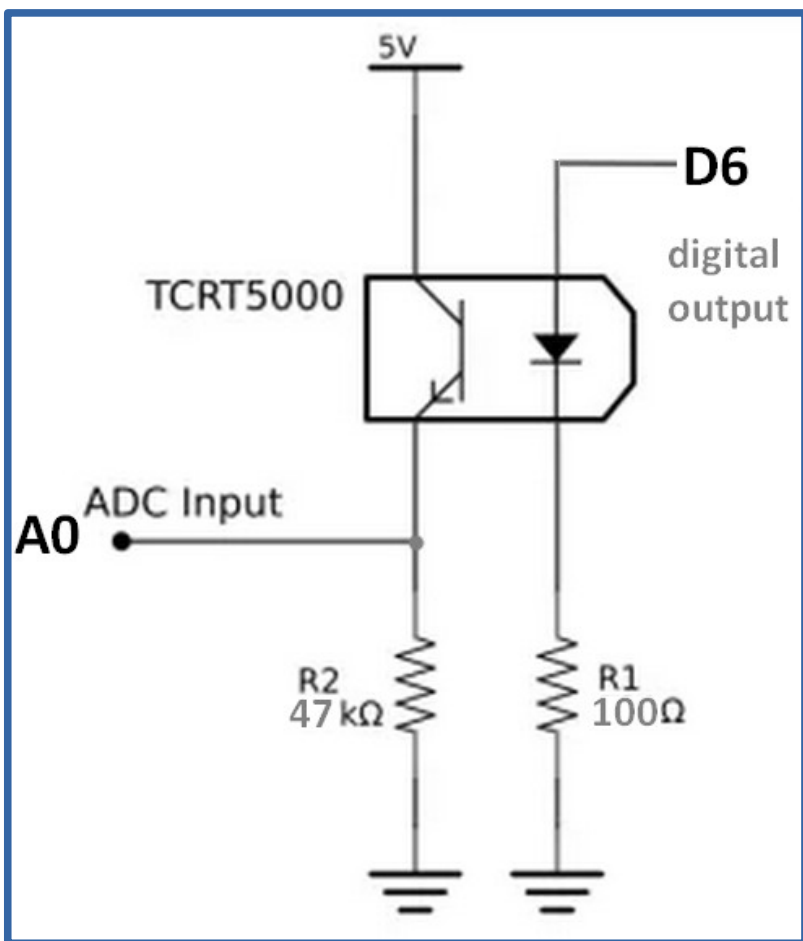
Működési elv: ha nincs akadály, a fény nem verődik vissza, a tranzisztor nem érzékel fényt.

Visszaverődés esetén a távolságtól és a reflexiós tényezőtől függ a visszaverődési arány.



Bekötési vázlat

Fototranzisztornál mindegy, hogy kollektor- vagy emitter oldalra kerül a munkaellenállás. Mi most emitter oldalra tettük, így a nagyobb feszültség nagyobb megvilágítottságot jelent.



fritzing

TCRT5000_test1.ino

- Több mérést végzünk, átlagolással

```
void setup() {
  Serial.begin(9600);
  analogReference(DEFAULT);    //VCC referencia
  pinMode(6,OUTPUT);
  digitalWrite(6,HIGH);
}

void loop() {
  int a = meres(A0, 500);
  Serial.print("A = ");
  Serial.println(a);
  delay(500);
}

int meres(int pin, int n) {
  long sum=0;
  for(int i=0; i<n; i++) {
    sum += analogRead(pin);
  }
  return sum/n;
}
```

Kísérletezzünk!

- Távolítsunk/közelítsünk egy tárgyat, és figyeljük a mért értékek változását!



tcrt5000_test2.ino

- A beszűrődő fény hatását úgy minimalizálhatjuk, hogy a LED ki- és bekapcsolt állapotában is végzünk egy-egy mérést, s csak a változást vesszük figyelembe
- A LED-et a **D6** kimenet vezérli, az analóg jelet az **A0** bemenet fogadja

```
void setup() {
  Serial.begin(9600);
  analogReference(DEFAULT);
  pinMode(6, OUTPUT);
  digitalWrite(6, LOW);
}

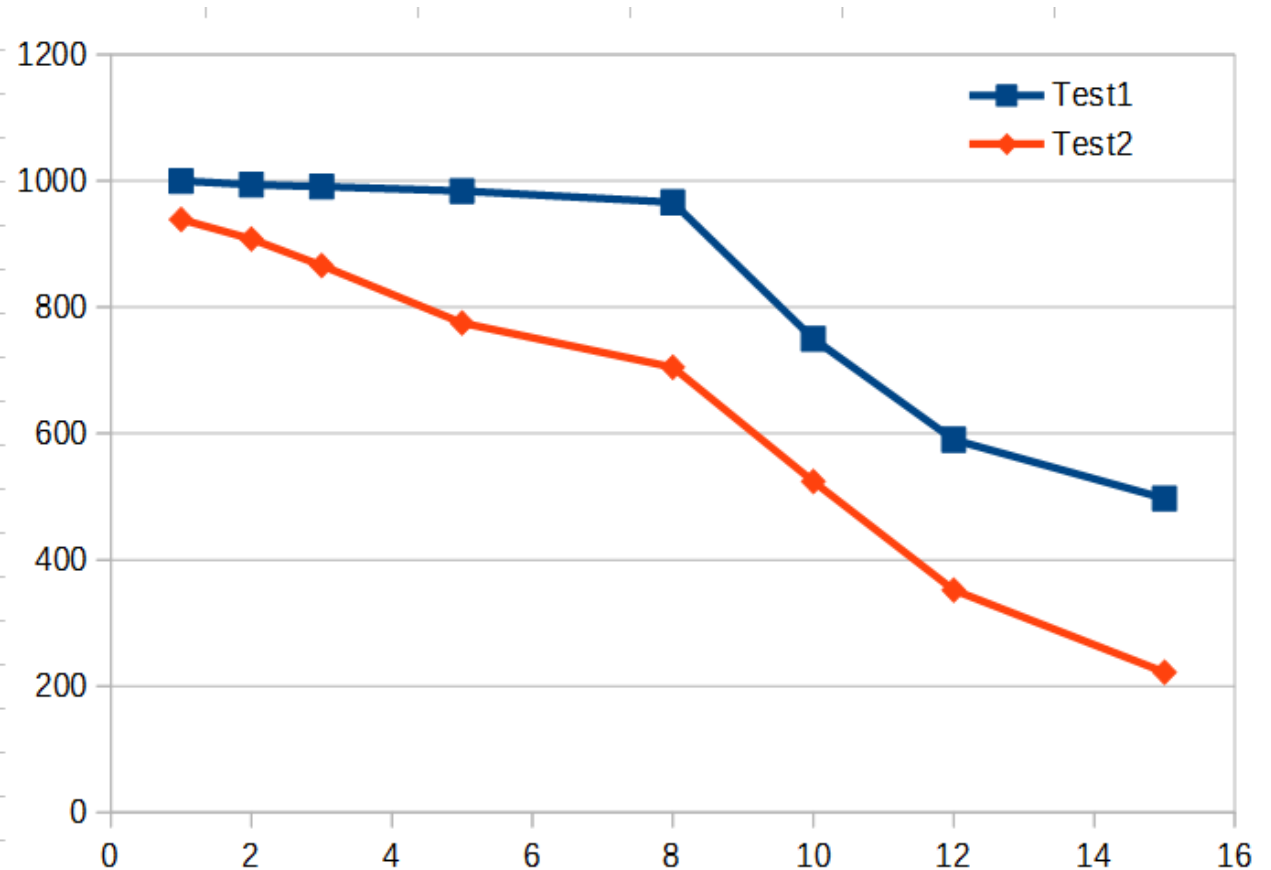
void loop() {
  int a = meres(6, A0, 500);
  Serial.print("A = ");
  Serial.println(a);
  delay(500);
}
```

```
int meres(int pin1, int pin2, int n) {
  long sum1=0, sum2=0, difi;
  digitalWrite(pin1, HIGH);
  for(int i=0; i<n; i++) {
    sum1 += analogRead(pin2);
  }
  digitalWrite(pin1, LOW);
  for(int i=0; i<n; i++) {
    sum2 += analogRead(pin2);
  }
  difi = (sum1 - sum2)/n;
  return difi;
}
```

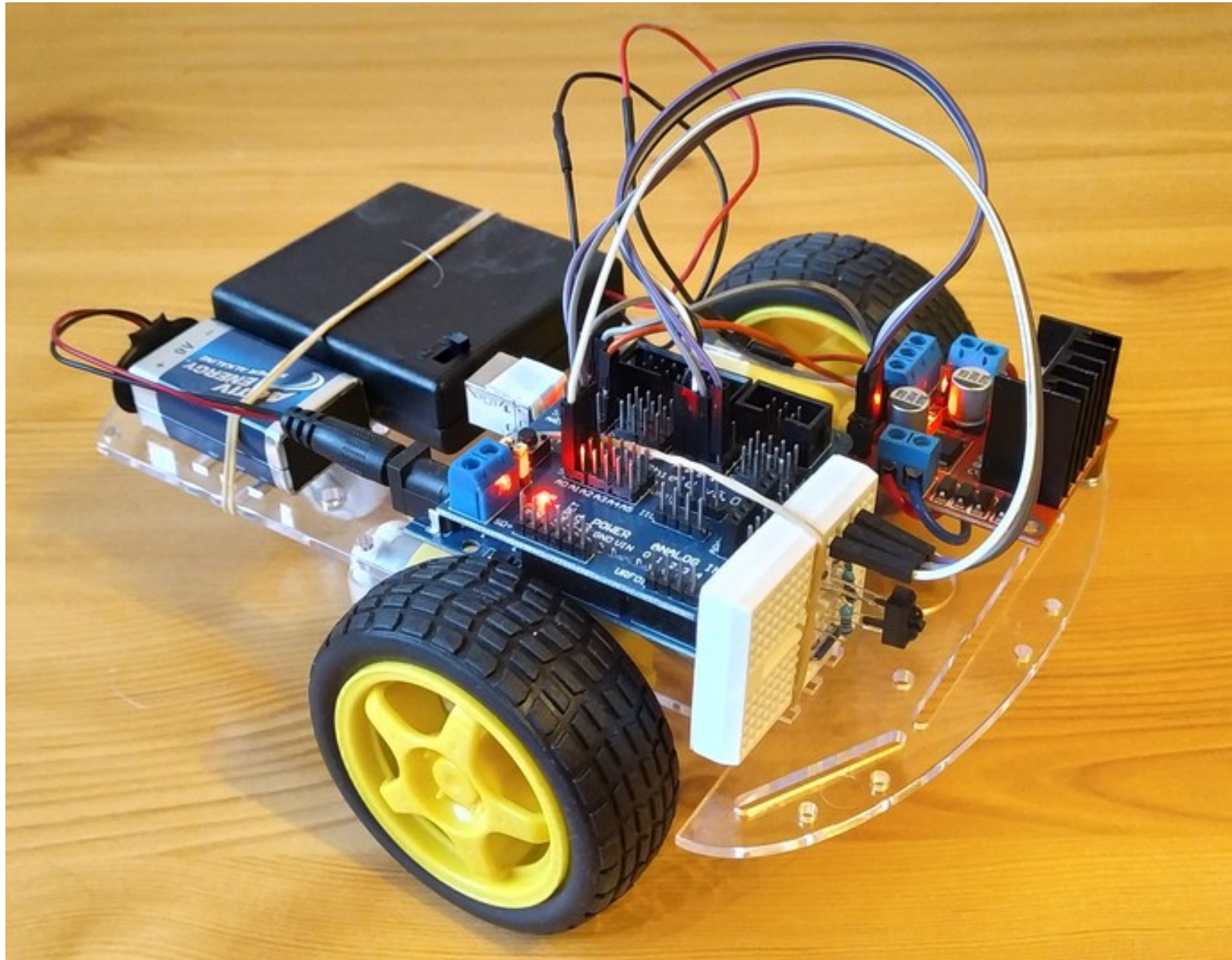
A beszűrődő fény hatása

- Az alábbi táblázatban és ábrán összehasonlítottuk a két program eredményét. A különbséget a beszűrődő fény okozza

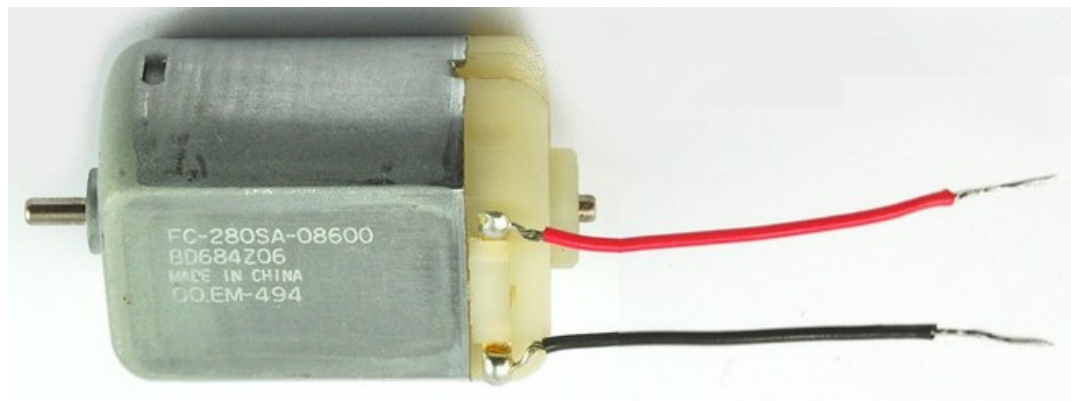
Távolság [cm]	Test1	Test2
1	1000	939
2	994	908
3	991	866
5	984	775
8	966	705
10	750	524
12	590	352
15	497	222



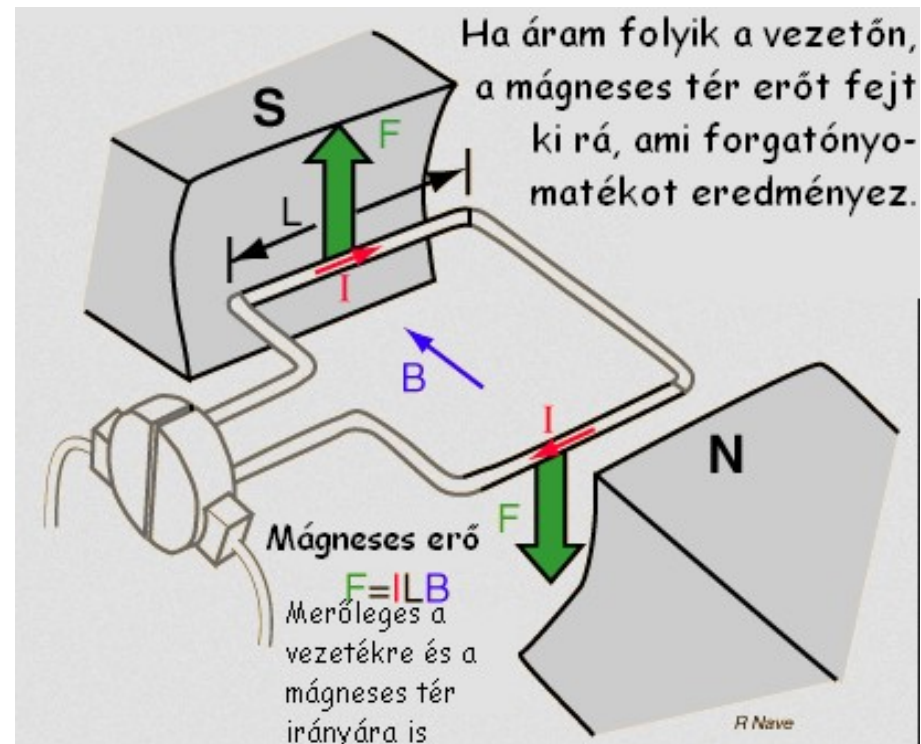
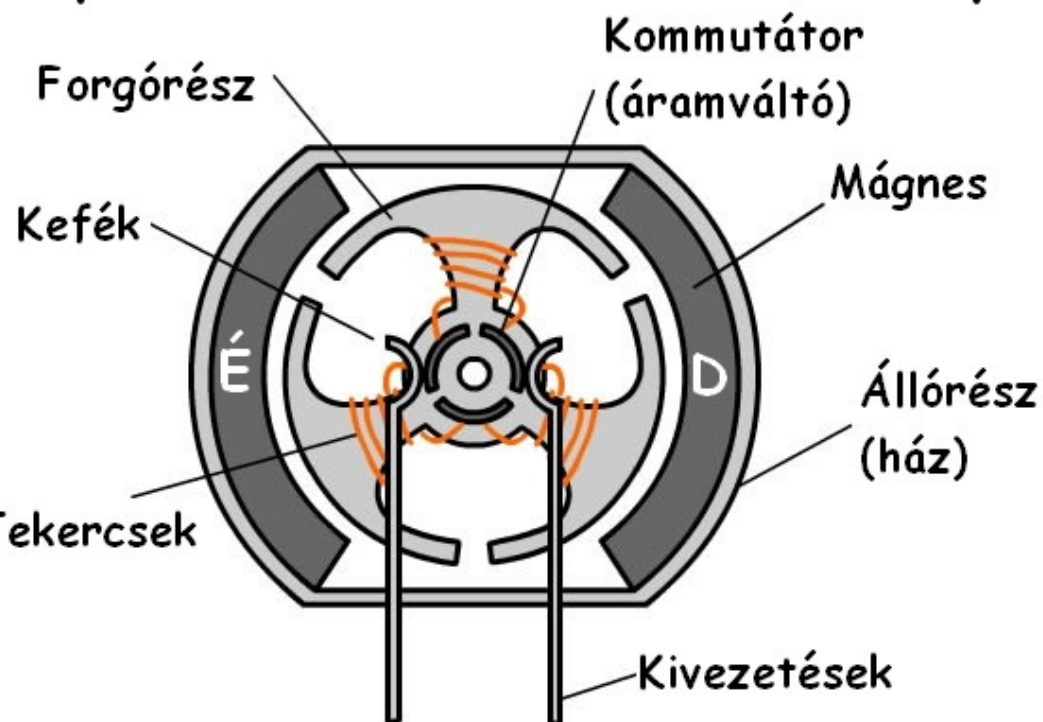
Bevezetés a robotikába



Kisteljesítményű DC motorok



Tipikus kefések motor metszeti képe



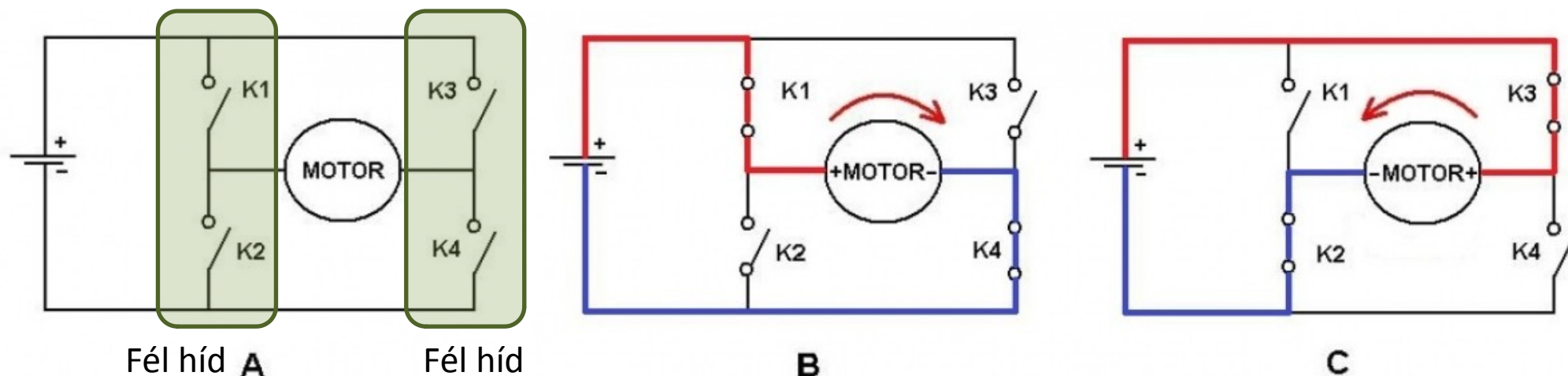
Feszültség: 3 V – 6V

Áram: 0.4 A – 0.6 A

Forgásirány váltás: polaritásváltással

Motorvezérlés: H-híd

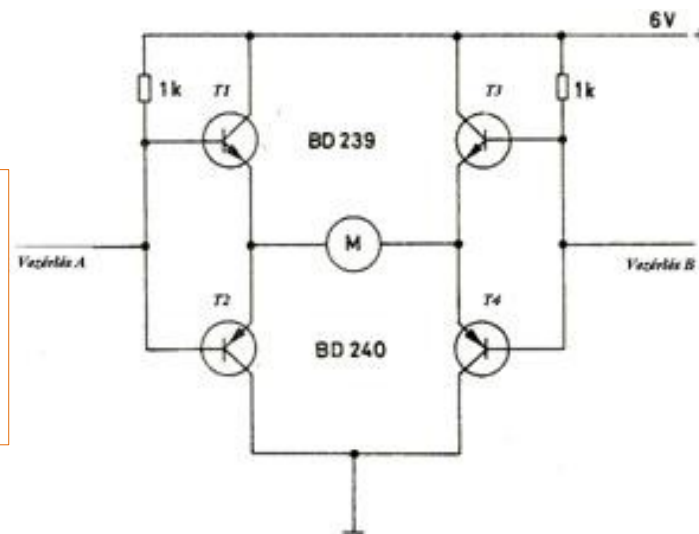
Ha a motor forgásirányát meg akarjuk változtatni, fel kell cserélni a polaritást. Ezt négy kapcsolóval tudjuk megoldani. Az elrendezést H-hídnak nevezzük.



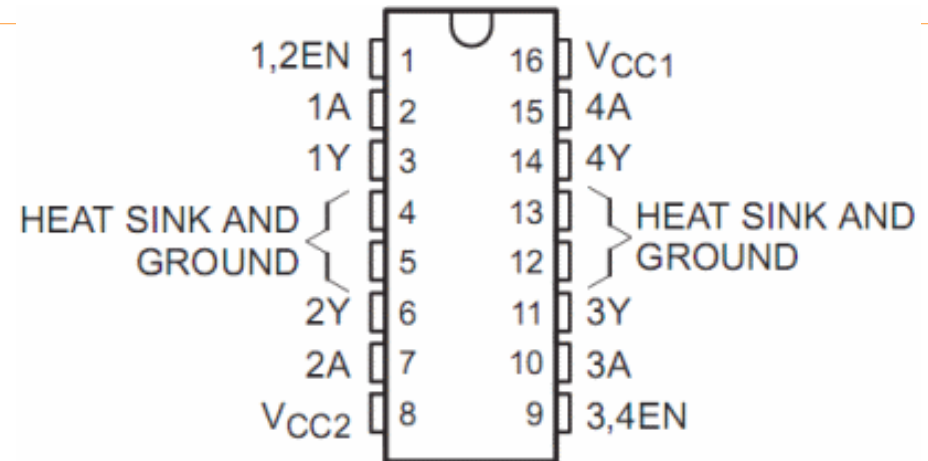
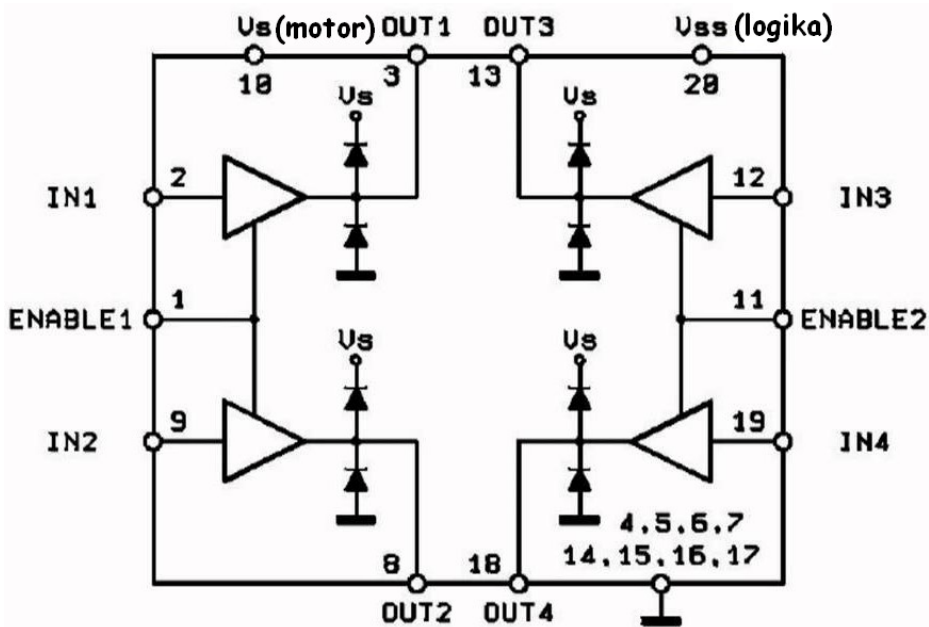
Forrás és leírás: hobbyrobot.hu/content/vonalkoveto-i-robotika-kezdoknek

K1 és **K4** zárásakor a motor az egyik irányba forog.
K2 és **K3** zárásakor az ellenkező irányba forog.

Elektronikus vezérlés: tranzisztorokkal
Mivel **K1** és **K2**, s hasonlóan **K3** és **K4** sohasem lehet egyidejűleg zárva, elegendő félhidanként egy-egy vezérlőjel, mely a kapcsolókat ellenütemben zárja.



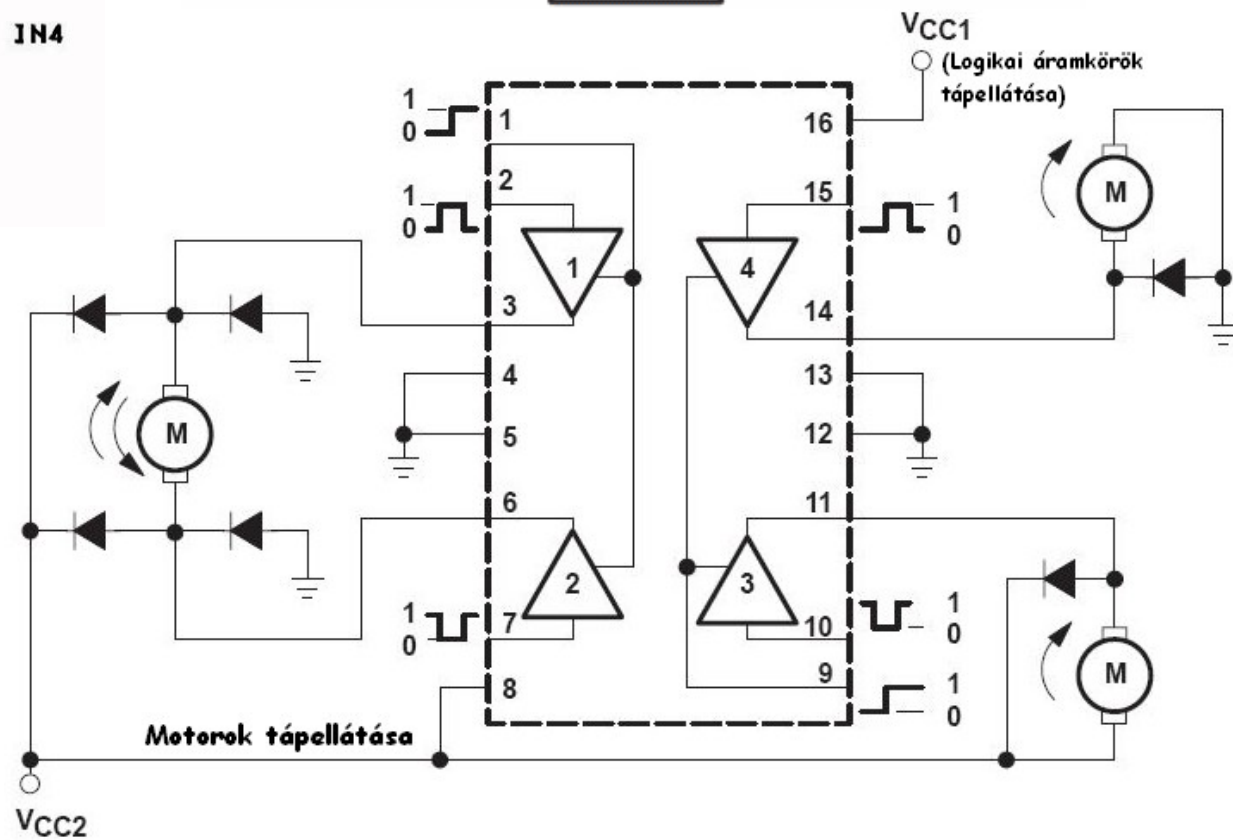
L293D 4db félhíd, vagy 2db H-híd



Félhíd: egyirányú forgás vezérlésére
(pl. ventilátorok)

H-híd: polaritásváltást igénylő
vezérlésekhez

$V_{CC} = 4,5 - 36 \text{ V}$
 $V_{A}, V_{EN} = 2.3 - 7 \text{ V}$
 $I_{max} = 0,6 \text{ A}$
 $I_{peak} = 1,2 \text{ A (100 } \mu\text{s)}$



Egy motor vezérlése

Enable1,2	IN1	IN2	Motor
H	0	0	Stop
H	1	0	Előremenet
H	0	1	Hátramenet
H	1	1	Stop
L	X	X	Stop

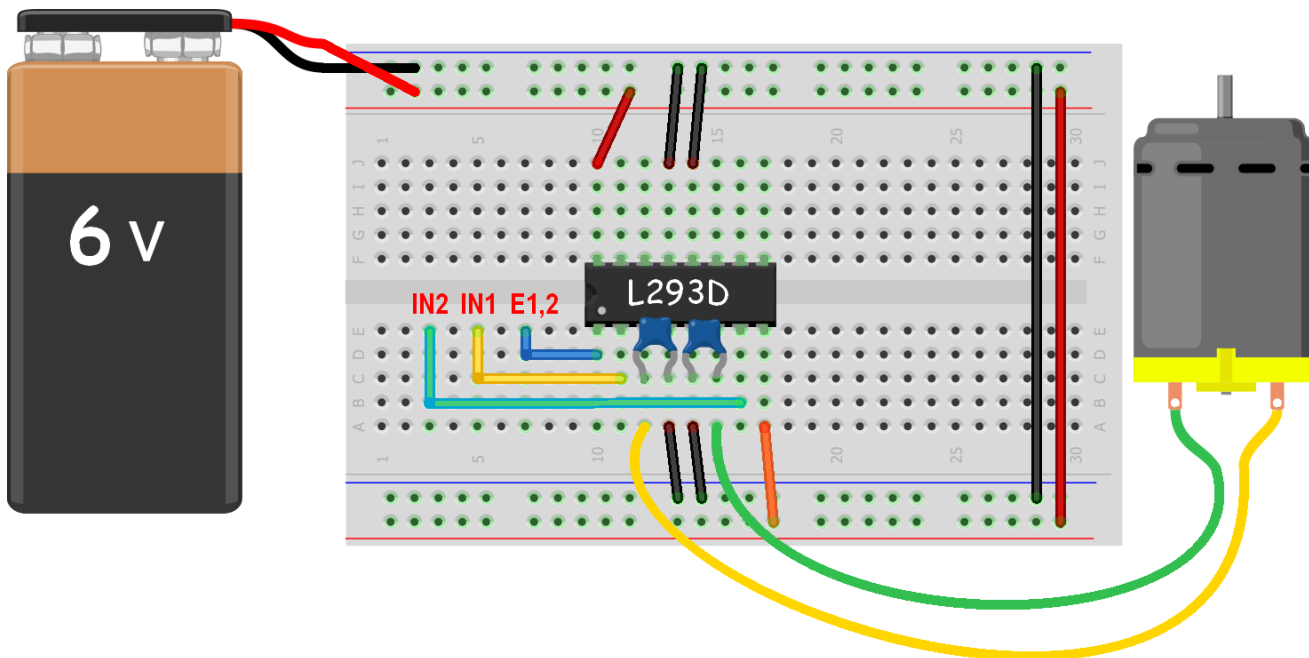
E1,2: Letiltja vagy engedélyezi az 1. H-hidat

In1 és **In2** a forgás irányát adja meg.

Csak az **1,0** illetve **0,1** kombináció esetén van forgás.

Egyszerűsítési lehetőségek:
(ha sokalljuk a kivezetéseket)

1. Ha **E1,2** állandóan magas szintet kap, **In1** és **In2** önmagában elég a vezérléshez.
2. Ha **In2 = In1** (inverter, IC-vel vagy tranzisztorral) akkor **E1,2** és **In1** elegendő a vezérléshez.



Made with  Fritzing.org

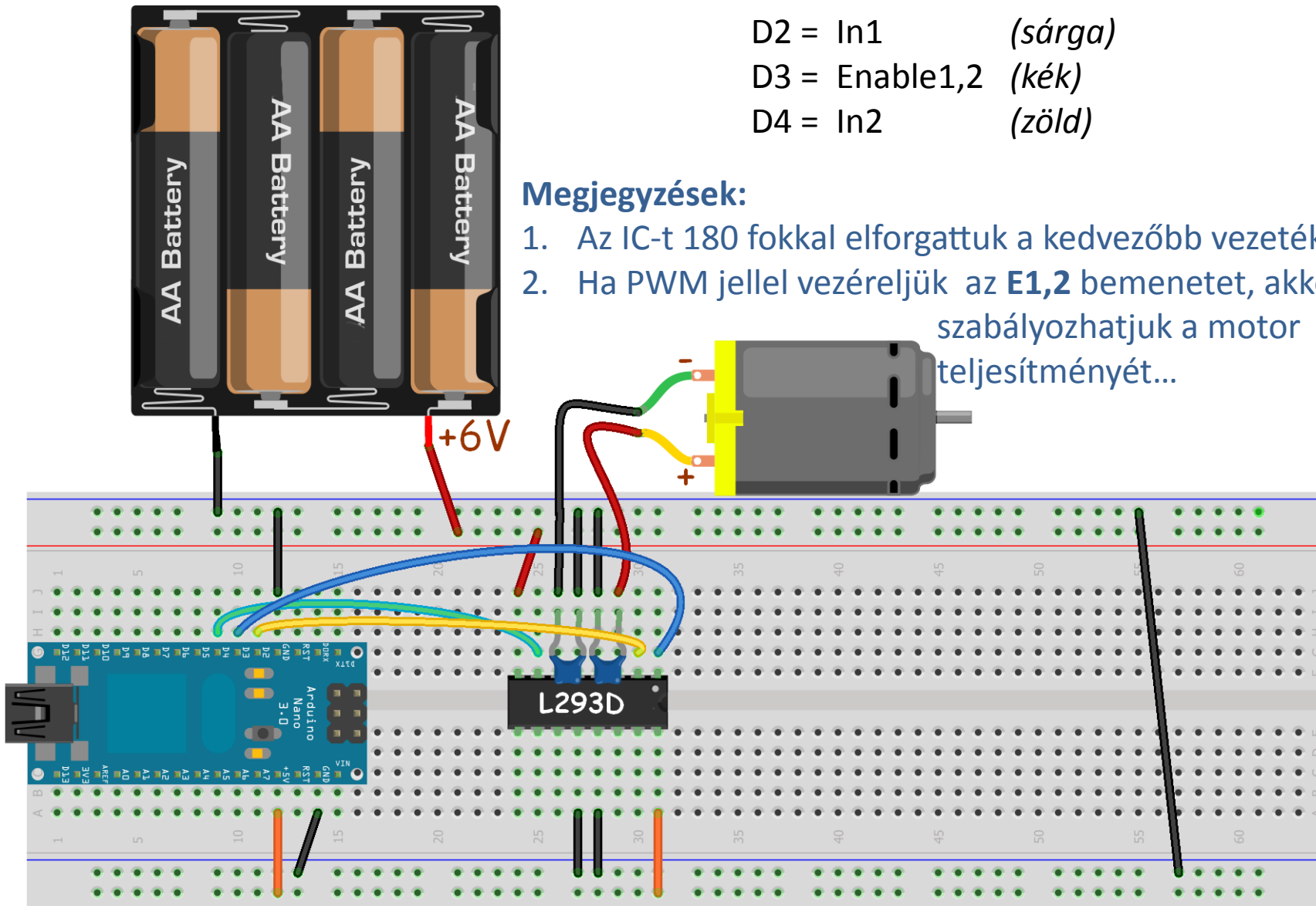
Egy motor vezérlése Arduinoval

Bekötési vázlat:

D2 = In1 (sárga)
D3 = Enable1,2 (kék)
D4 = In2 (zöld)

Megjegyzések:

1. Az IC-t 180 fokkal elforgattuk a kedvezőbb vezetékezéshez!
2. Ha PWM jellel vezéreljük az **E1,2** bemenetet, akkor szabályozhatjuk a motor teljesítményét...



Az **Arduino** itt az USB csatlakozón keresztül kap tápfeszültséget!

Made with  Fritzing.org

L293D_test_1M.ino

- Egy motor vezérlése. A végtelen ciklusban 5s várakozás után 2s előremenet, 1s várakozás, majd 2s hátramenet ismétlődik. Az előre- és hátramenet 75 %-os teljesítménnyel történik.

```
#define PWMA 3 //L293D pin1
#define D1A 2 //L293D pin2
#define D2A 4 //L293D pin7

void setMotor(int speed, boolean reverse)
{
    analogWrite(PWMA, speed);
    digitalWrite(D1A, !reverse);
    digitalWrite(D2A, reverse);
}

void setup()
{
    Serial.begin(9600);
    pinMode(PWMA, OUTPUT);
    pinMode(D1A, OUTPUT);
    pinMode(D2A, OUTPUT);
    Serial.println("L293D test program");
}
```

```
void loop()
{
    delay(5000);
    Serial.println("move forward 75%");
    setMotor(196, 0); // Forward 75%
    delay(2000);
    setMotor(0, 0); // Stop motor
    delay(1000);
    Serial.println("move reverse 75%");
    setMotor(196, 1); // Reverse 75%
    delay(2000);
    setMotor(0, 0); // Stop motor
}
```

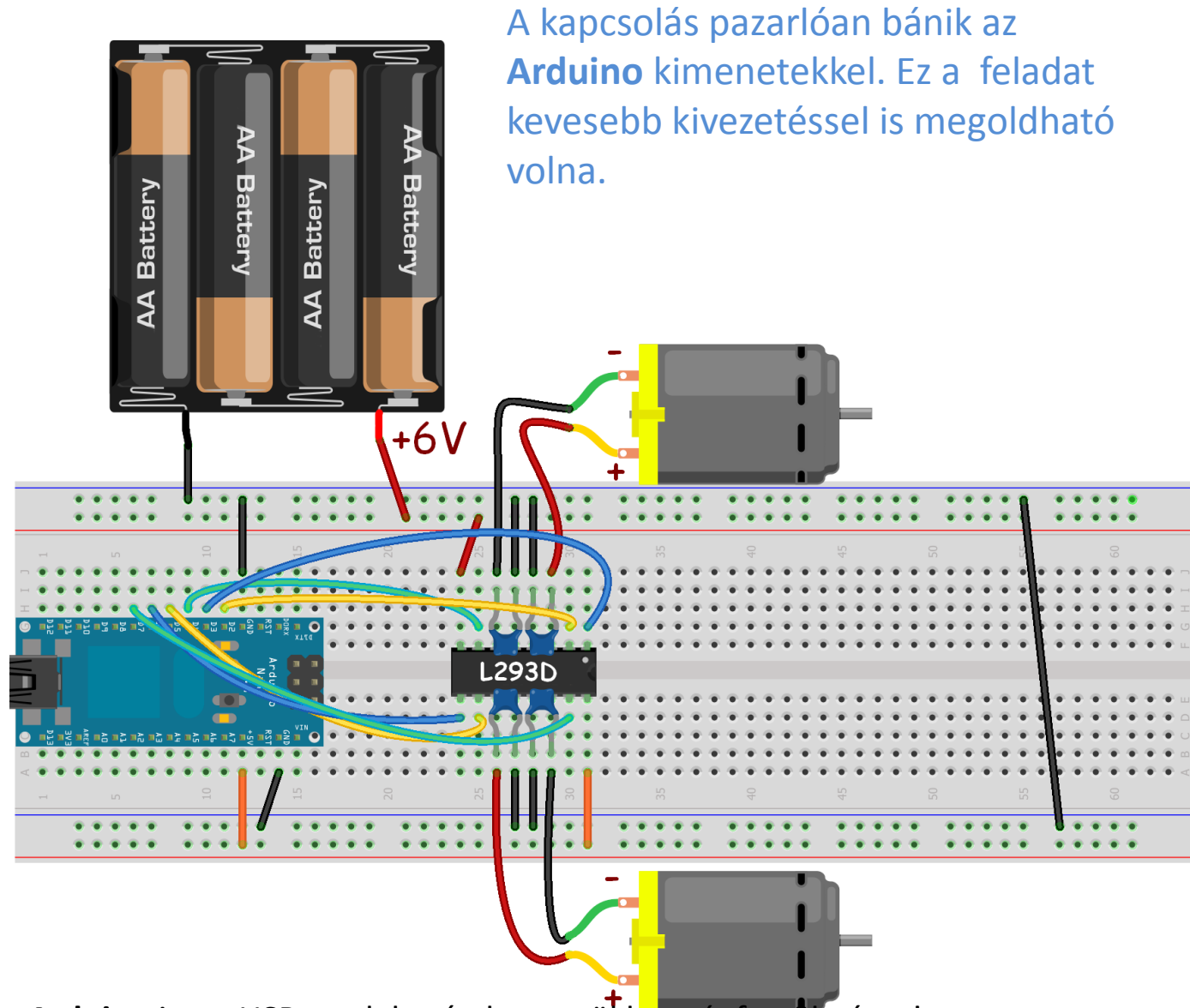
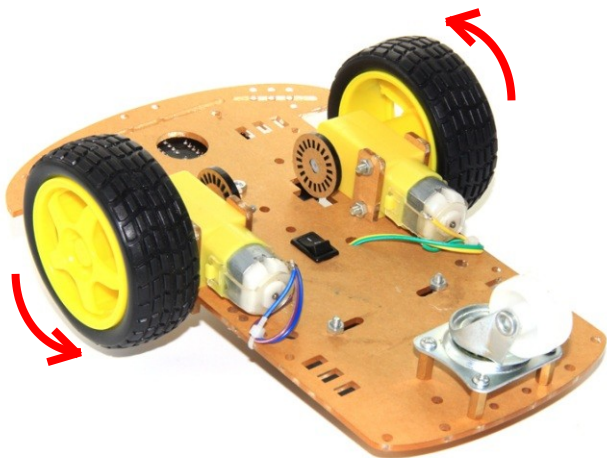
Két motor vezérlése Arduinoval

Megjegyzés:

A rajzon látható kapcsolásban azonos vezérlés mellett a két motor azonos irányba forog.

A fényképen látható független jobb és bal kerék meghajtású kocsinhoz azonban az egyik motor polaritását fel kell cserélni!

Egyenesvonalú mozgásnál a két motor forgása ellentétes irányú!



A kapcsolás pazarlóan bánik az **Arduino** kimenetekkel. Ez a feladat kevesebb kivezetéssel is megoldható volna.

Az **Arduino** itt az USB csatlakozón keresztül kap tápfeszültséget!

Made with Fritzing.org

L293D_test_2M.ino

- Két motor vezérlése. A végtelen ciklusban 5s várakozás után 2s előremenet, 1s várakozás, majd 2s hátramenet ismétlődik.
- Az előre- és hátramenet 75 %-os teljesítménnyel történik.

```
#define PWMA 3 //L293D pin1
#define D1A 2 //L293D pin2
#define D2A 4 //L293D pin7
#define PWMB 6 //L293D pin9
#define D3B 5 //L293D pin10
#define D4B 7 //L293D pin15

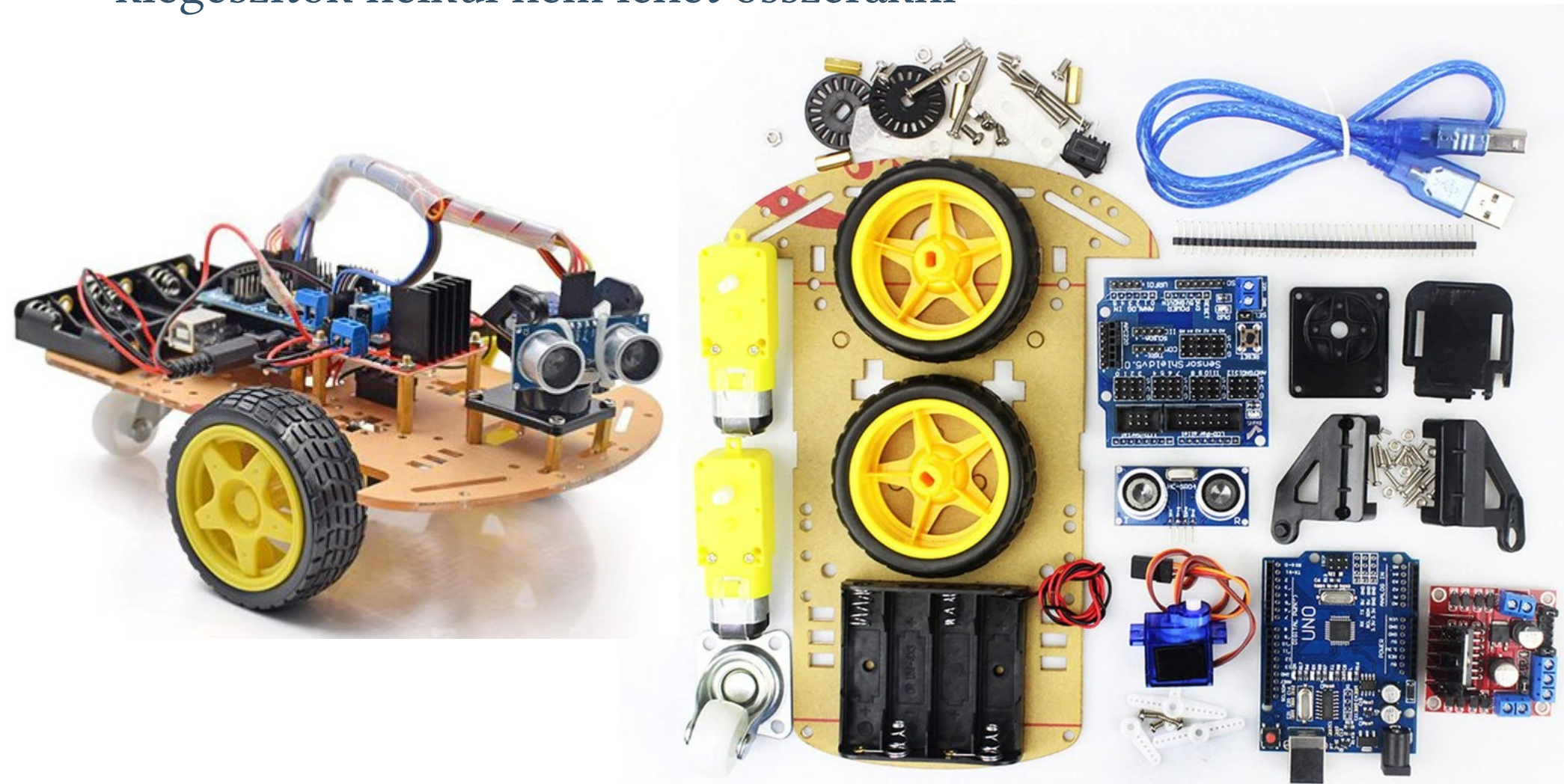
void setup() {
  Serial.begin(9600);
  pinMode(PWMA, OUTPUT);
  pinMode(D1A, OUTPUT);
  pinMode(D2A, OUTPUT);
  pinMode(PWMB, OUTPUT);
  pinMode(D3B, OUTPUT);
  pinMode(D4B, OUTPUT);
  Serial.println("L293D test program");
}
```

```
void setMotor(int speed, boolean reverse)
{
  analogWrite(PWMA, speed);
  analogWrite(PWMB, speed);
  digitalWrite(D1A, !reverse);
  digitalWrite(D2A, reverse);
  digitalWrite(D3B, !reverse);
  digitalWrite(D4B, reverse);
}

void loop() {
  delay(5000);
  Serial.println("move forward 75%");
  setMotor(196, 0); // Forward 75%
  delay(2000);
  setMotor(0, 0); // Stop motor
  delay(1000);
  Serial.println("move reverse 75%");
  setMotor(196, 1); // Reverse 75%
  delay(2000);
  setMotor(0, 0); // Stop motor
}
```

Építsünk robotot!

- A képen látható 2WD (kétkerék meghajtású) robot az olcsóbbak közé tartozik, de nem volt szerencsés választás mert barkácsolás és kiegészítők nélkül nem lehet összerakni



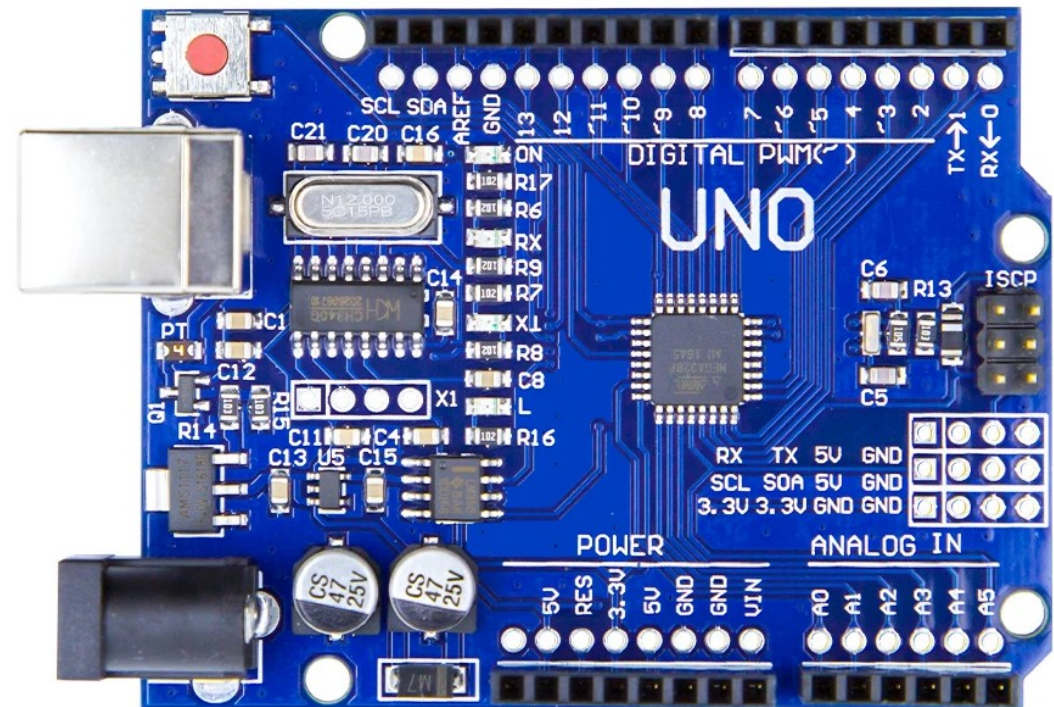
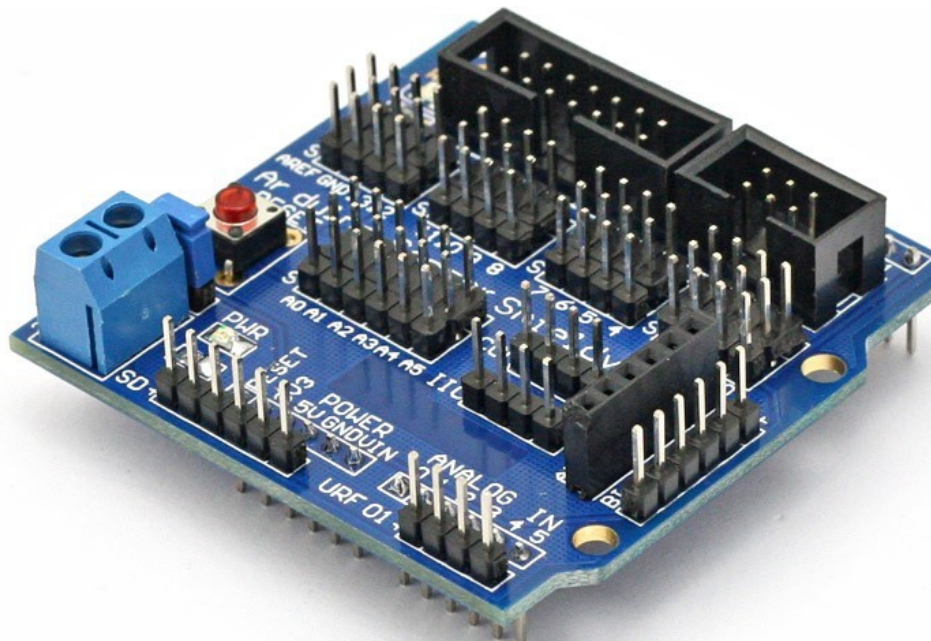
A robot alváz és a motorok

- A 2WD robotalváz (szerelőlap, bolygókerék, elemtartó, 2 db egyenáramú motor a műanyag kerekekkel) külön készletként is beszerezhető
- Könnyen összeszerelhető, de a szerelőlapon található furatok egyetlen általam ismert mikrovezérlő vagy motorvezérlő kártyához sem illeszkednek – jó, ha van otthon fúrógép, távtartók és csavarok



A mikrovezérlő kártya és kiegészítői

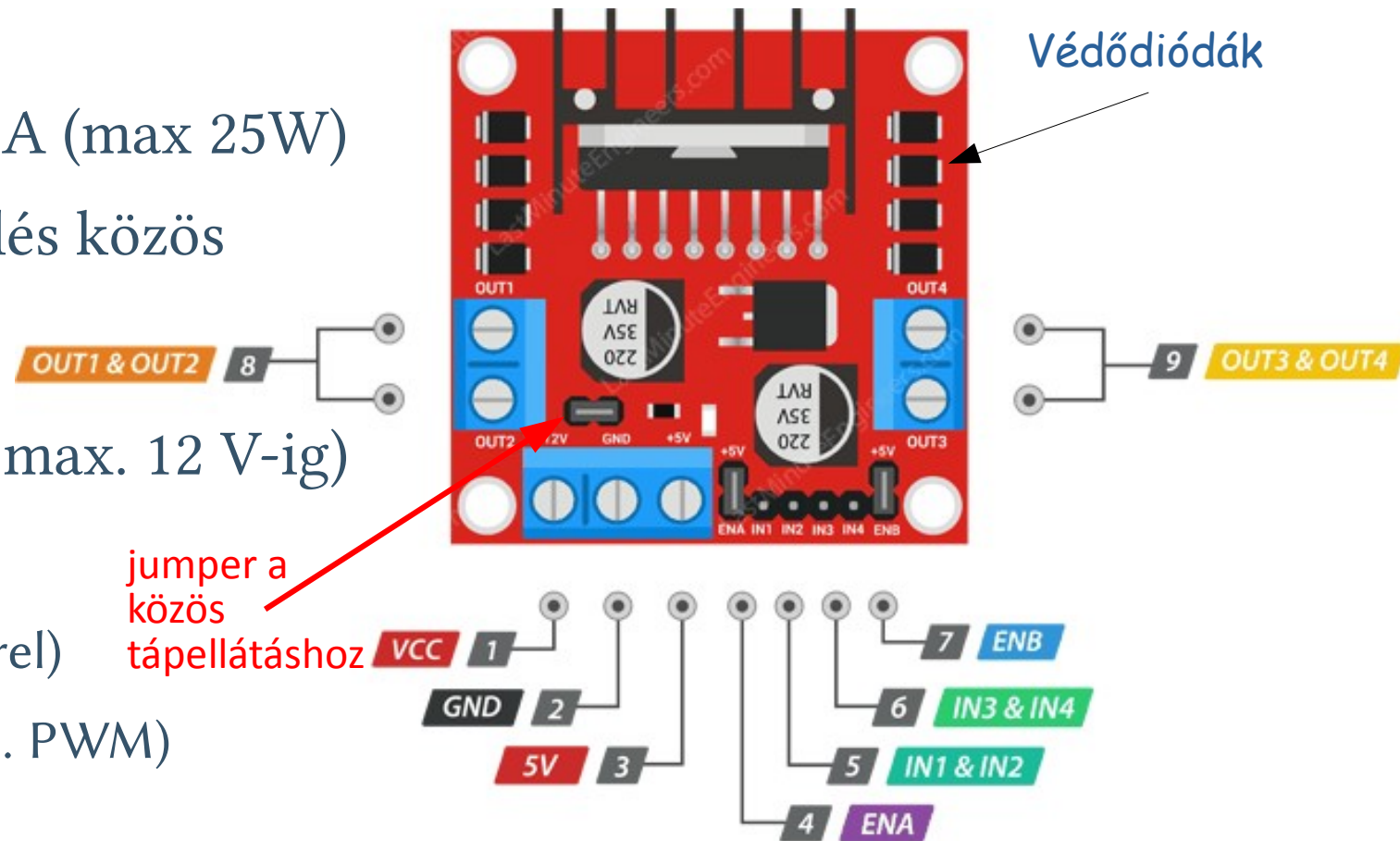
- Az Arduino kártya egy szokványos UNO klón, CH340 USB-soros átalakítóval
- Az Arduino IDE Tools menüjében az **Arduino/Genuino Uno** kártyát kell kiválasztani



- A Sensor shield kártya csak az Arduino kártya kivezetéseinek kényelmesebb elérésére való, egyébként nélkülözhető

Az L298N motorvezérlő kártya

- A motorvezérlő kártya egy L298N IC-t tartalmaz (Stmicroelectronics)
- Két teljes híd
- Max 46 V, ill. 2 A (max 25W)
- Motor és vezérlés közös vagy külön táplálással (közös táplálás max. 12 V-ig)

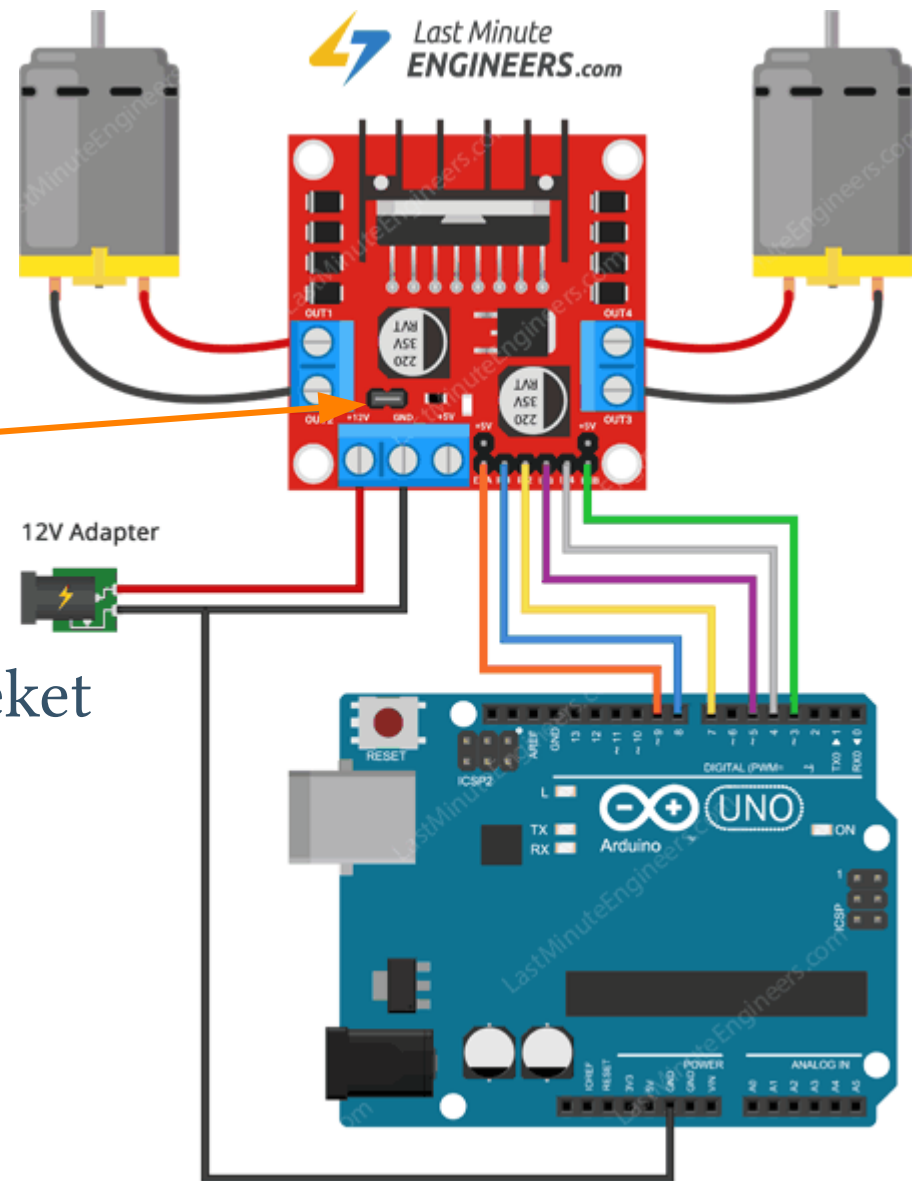


- Engedélyezés:
 - ❖ fixen (jumperrel)
 - ❖ Külső jellel (pl. PWM)

- Forrás: lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/

Egy lehetséges kapcsolás

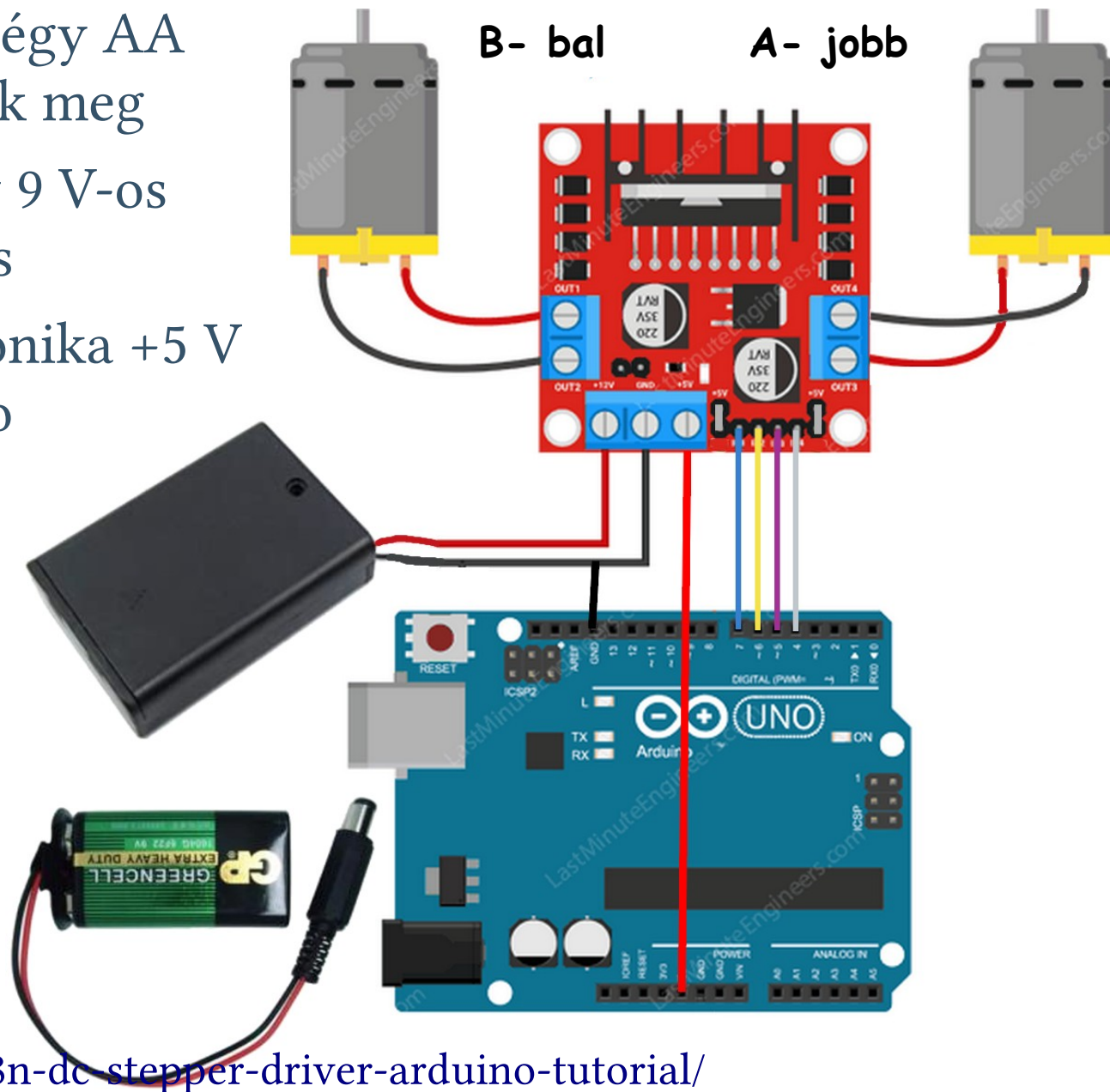
- Az Arduino kártya itt külön táplálást kap, csak a közös pont (GND) van közösítve, a motorvezérlő kártya pedig saját maga állítja elő az 5V tápfeszültséget (jumper van)
- Ha a motorok teljesítményét (a robot sebességét) is szabályozni akarjuk, akkor az Enable bemeneteket is az Arduino-val kell meghajtani



- Forrás: lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/

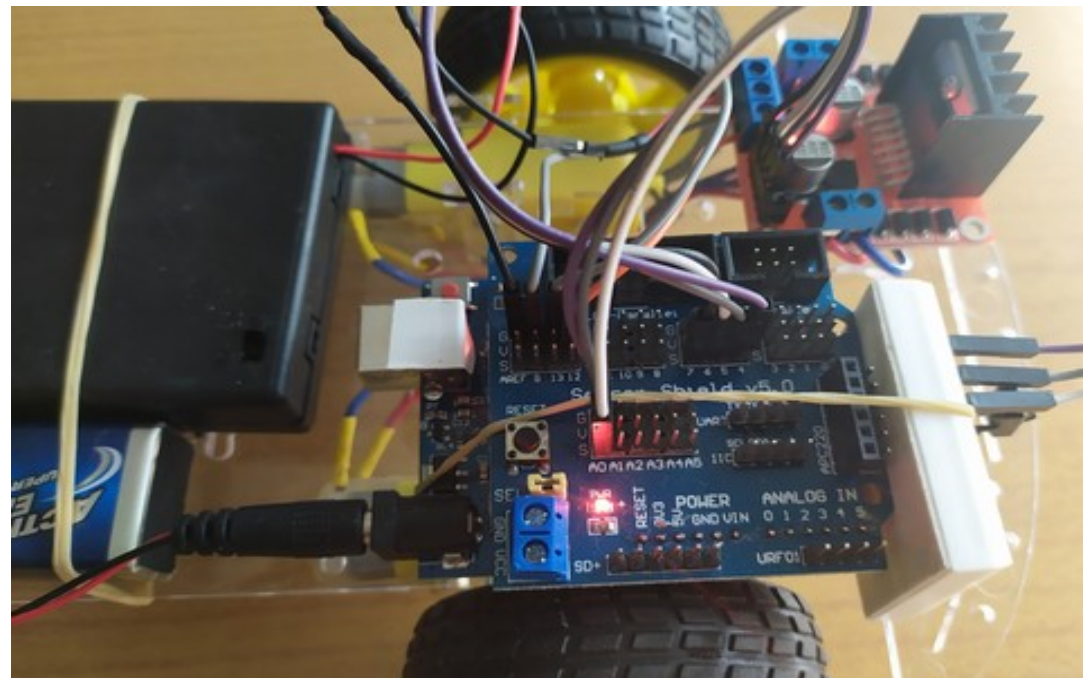
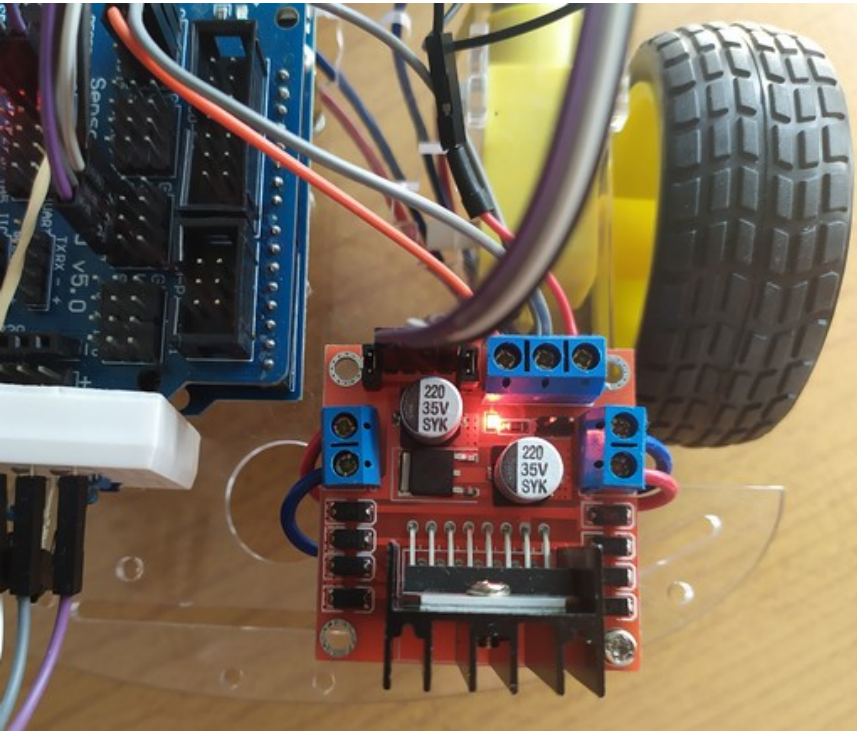
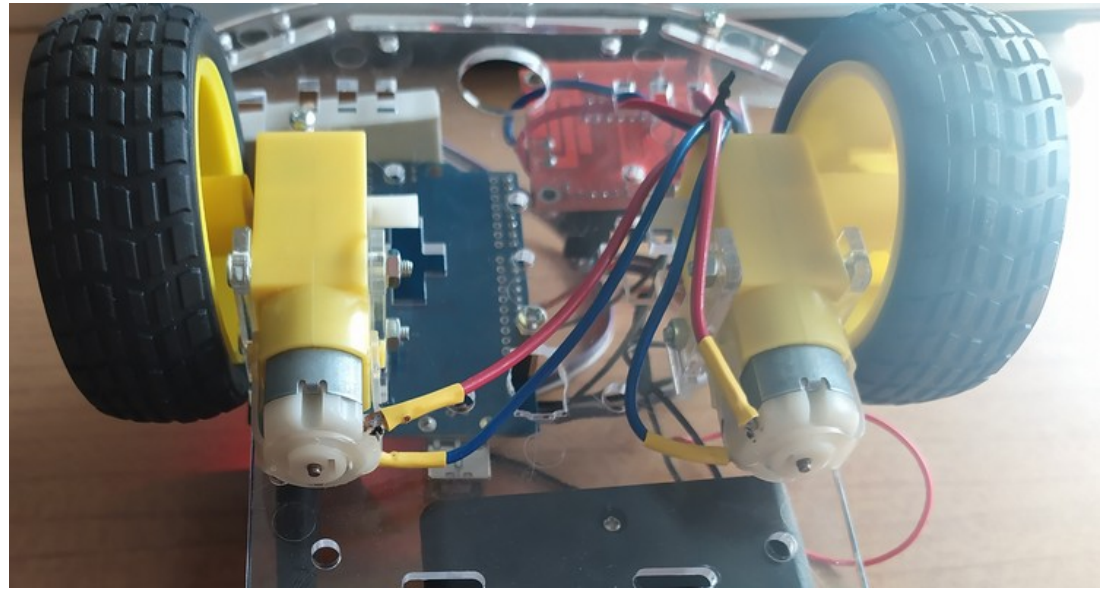
Az általunk megépített kapcsolás

- A motorok táplálását négy AA ceruzaelemmel oldottuk meg
- Az Arduino kártya egy 9 V-os elemről kap táplálást és
- A motorvezérlő elektronika +5 V feszültségét az Arduino kártya biztosítja
- A motorok tükör-szimmetrikus bekötése hardveresen biztosítja az ellentétes irányú forgást a bal és jobb motornál
- Felhasznált forrás:
lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/



A megépített kapcsolás részletei

- A részletfotókon látható a motorok bekötése és a tápellátás megoldása



motor_test.ino

- Ez az egyszerű kis program csak arra szolgál, hogy a motorok működését és forgásirányát ellenőrizni tudjuk

- **Vezérlés:**

B1A	B1B	A1A	A1B	
0	0	0	0	Stop
1	0	1	0	Forward (Előre)
0	1	0	1	Reverse (Hátra)
1	0	0	1	Left (Balra)
0	1	1	0	Right (Jobbra)

- A program a soros porton kiírja, hogy éppen mit csinál (9600 bps)
- A program: Előre 3 mp, Hátra 3mp, Jobbra 3 mp, Balra 3 mp, közben 1-1 mp Stop

```
#define B1A 7 // Bal motor előre
#define B1B 6 // Bal motor hátra
#define A1A 5 // Jobb motor előre
#define A1B 4 // Jobb motor hátra
void setup() {
  Serial.begin(9600);
  Serial.println("Motor test");
  pinMode(B1A,OUTPUT);
  pinMode(B1B,OUTPUT);
  pinMode(A1A,OUTPUT);
  pinMode(A1B,OUTPUT);
  digitalWrite(B1A,LOW);
  digitalWrite(B1B,LOW);
  digitalWrite(A1A,LOW);
  digitalWrite(A1B,LOW);
}
```

Folytatás a következő oldalon...

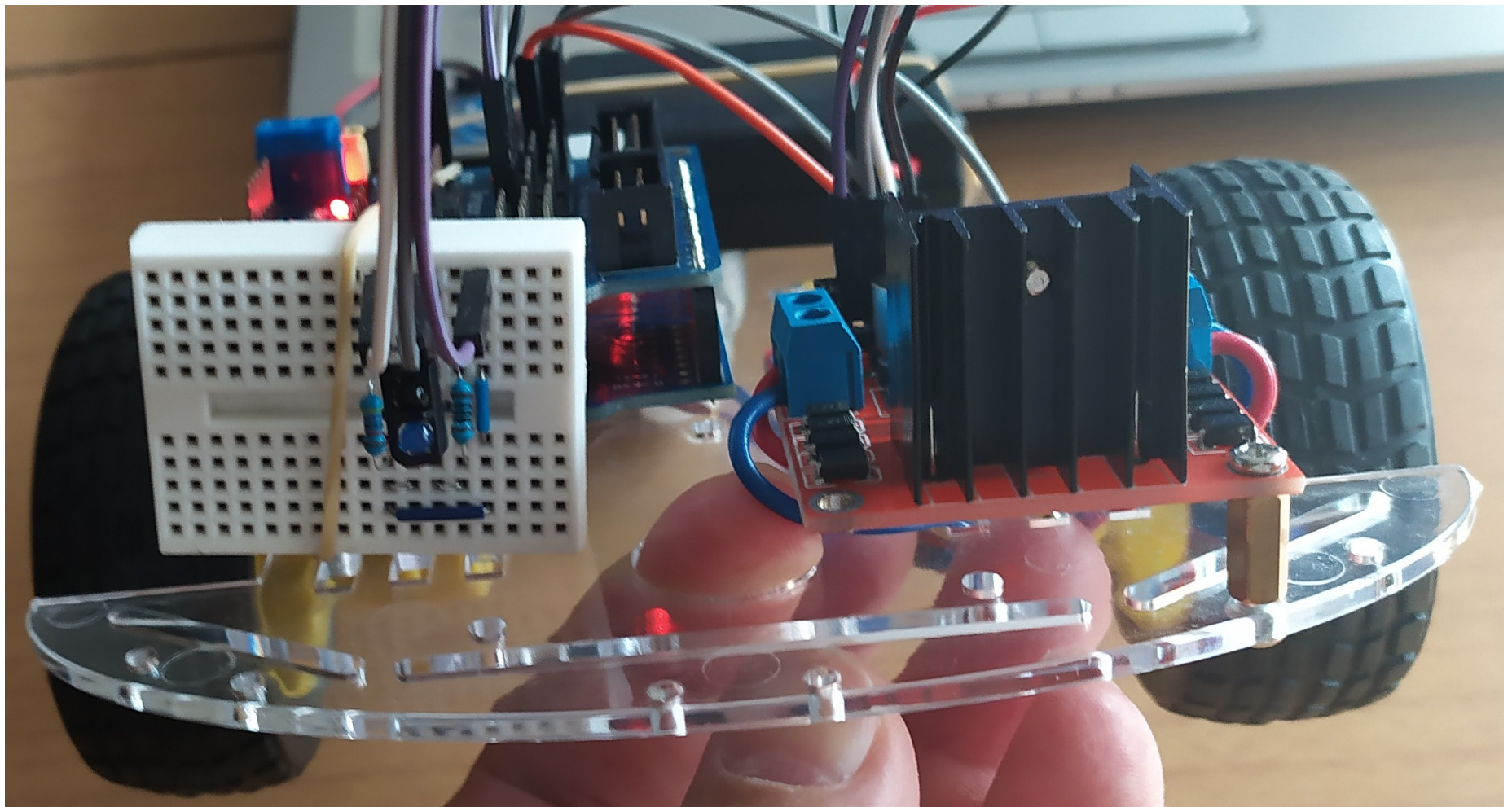
motor_test.ino

```
void loop() {  
  Serial.println("Forward ALL");  
  digitalWrite(B1A,HIGH);  
  digitalWrite(B1B,LOW);  
  digitalWrite(A1A,HIGH);  
  digitalWrite(A1B,LOW);  
  delay(3000);  
  digitalWrite(B1A,LOW);  
  digitalWrite(B1B,LOW);  
  digitalWrite(A1A,LOW);  
  digitalWrite(A1B,LOW);  
  delay(1000);  
  Serial.println("Reverse ALL");  
  digitalWrite(B1A,LOW);  
  digitalWrite(B1B,HIGH);  
  digitalWrite(A1A,LOW);  
  digitalWrite(A1B,HIGH);  
  delay(3000);  
  digitalWrite(B1A,LOW);  
  digitalWrite(B1B,LOW);  
  digitalWrite(A1A,LOW);  
  digitalWrite(A1B,LOW);  
  delay(1000);  
}
```

```
Serial.println("Turn RIGHT");  
  digitalWrite(B1A,HIGH);  
  digitalWrite(B1B,LOW);  
  digitalWrite(A1A,LOW);  
  digitalWrite(A1B,LOW);  
  delay(3000);  
  digitalWrite(B1A,LOW);  
  digitalWrite(B1B,LOW);  
  digitalWrite(A1A,LOW);  
  digitalWrite(A1B,LOW);  
  delay(1000);  
  Serial.println("Turn LEFT");  
  digitalWrite(B1A,LOW);  
  digitalWrite(B1B,LOW);  
  digitalWrite(A1A,HIGH);  
  digitalWrite(A1B,LOW);  
  delay(3000);  
  digitalWrite(B1A,LOW);  
  digitalWrite(B1B,LOW);  
  digitalWrite(A1A,LOW);  
  digitalWrite(A1B,LOW);  
  delay(2000);  
}
```

Robot akadályérzékelővel

- Szereljük fel egy TCRT5000 érzékelőt a robot elejére és kössük a kimenetét az Arduino A0 analóg bemenetére!
- A következő program leállítja a motorokat, ha akadályt észlel



robot_with_tcrt5000.ino

- Ebben a programban nem kapcsolgatjuk a LED-et
- A program indításakor végzünk egy „háttér” mérést, s ha ennél 10 %-kal több fényt érzékelünk, az akadálnak tekintjük

```
#define B1A 7 // Bal motor előre
#define B1B 6 // Bal motor hátra
#define A1A 5 // Jobb motor előre
#define A1B 4 // Jobb motor hátra

int state = 0; // 0:stop, 1:forward
int hatter, adat;

int meres(int pin, int n) {
    long sum = 0;
    for (int i = 0; i < n; i++) {
        sum += analogRead(pin);
    }
    return sum / n;
}

void setup() {
    Serial.begin(9600);
    Serial.println("Motor with TCRT5000");
    analogReference(DEFAULT);
    pinMode(B1A, OUTPUT);
    pinMode(B1B, OUTPUT);
    pinMode(A1A, OUTPUT);
    pinMode(A1B, OUTPUT);
    digitalWrite(B1A, LOW);
    digitalWrite(B1B, LOW);
    digitalWrite(A1A, LOW);
    digitalWrite(A1B, LOW);
    hatter = meres(A0, 100);
    Serial.print("Háttér = ");
    Serial.print(hatter);
    Serial.print(" Limit = ");
    hatter = hatter + hatter / 10;
    Serial.println(hatter);
}
```

robot_with_tcrt5000.ino

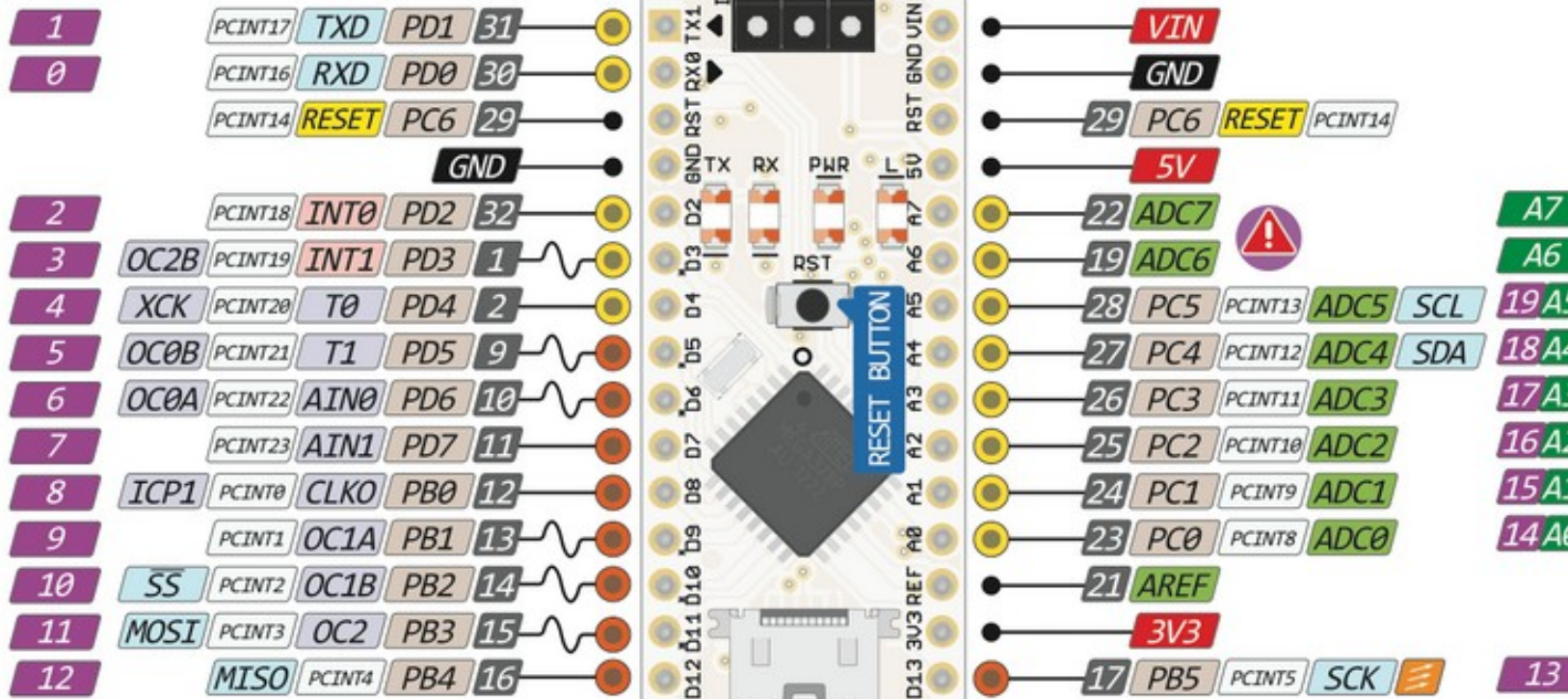
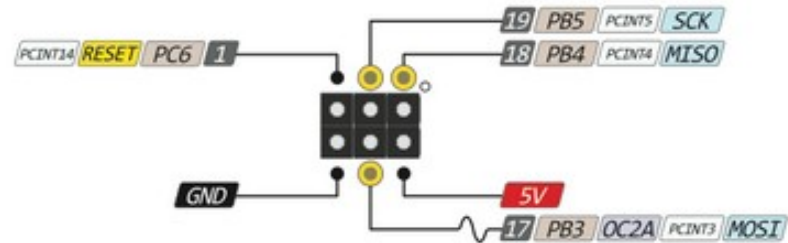
```
void loop() {
  adat = meres(A0, 100);
  switch (state) {
    case 0: { digitalWrite(B1A, LOW);
              digitalWrite(B1B, LOW);
              digitalWrite(A1A, LOW);
              digitalWrite(A1B, LOW);
              if ( adat < hatter) {
                Serial.println("Forward ALL"); state = 1;
              }
              break;
            }
    case 1: { digitalWrite(B1A, HIGH);
              digitalWrite(B1B, LOW);
              digitalWrite(A1A, HIGH);
              digitalWrite(A1B, LOW);
              if ( adat > hatter) {
                Serial.println("Stop ALL"); state = 0;
              }
              break;
            }
  }
  delay(50);
}
```

Az Arduino nano kártya kivezetései



NANO PINOUT

The power sum for each pin's group should not exceed 100mA



- Power
- GND
- Serial Pin
- Analog Pin
- Control
- INT
- Physical Pin
- Port Pin
- Pin function
- Interrupt Pin
- PWM Pin
- Port Power

Absolute MAX per pin 40mA recommended 20mA

Absolute MAX 200mA for entire package

Analog exclusively Pins

Ellenállás színkódok

