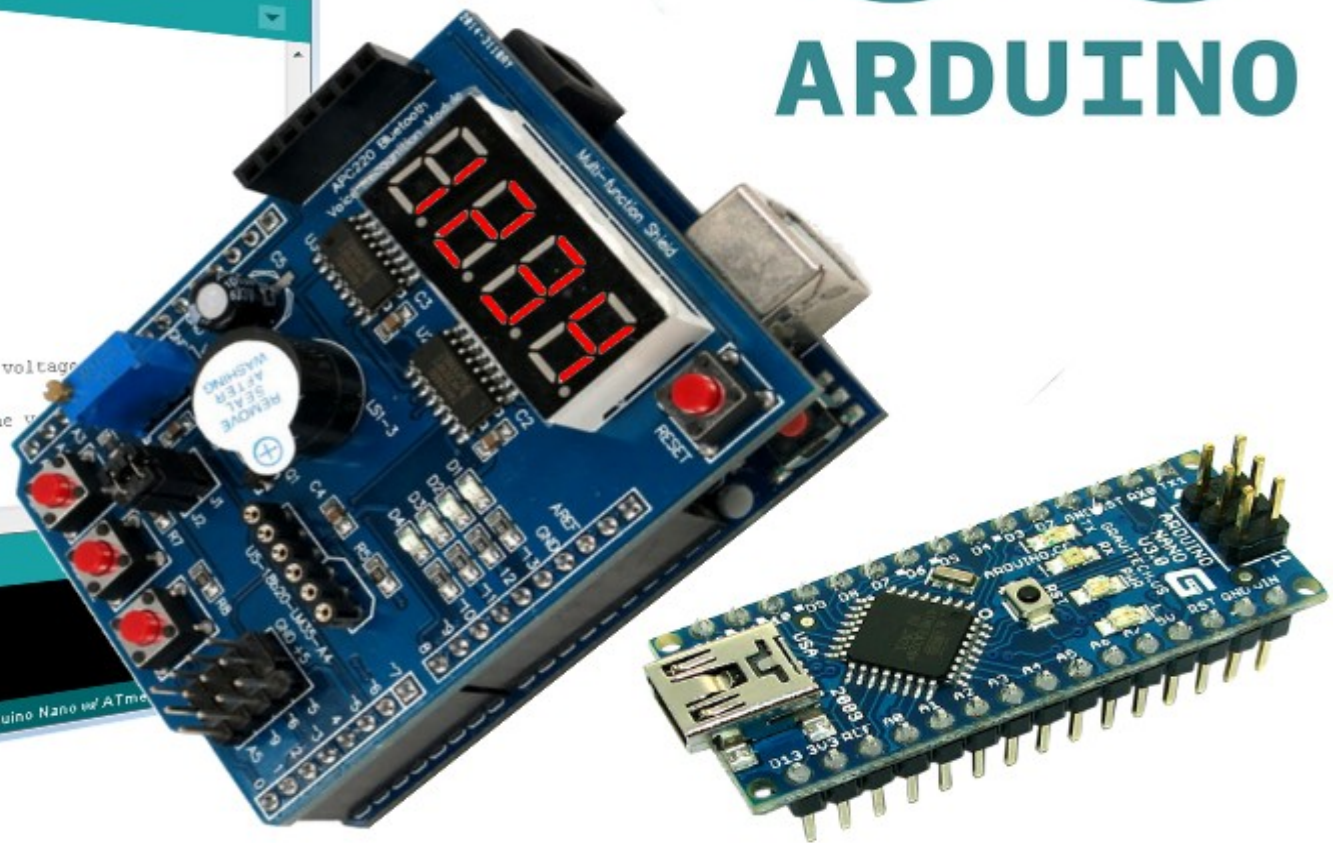
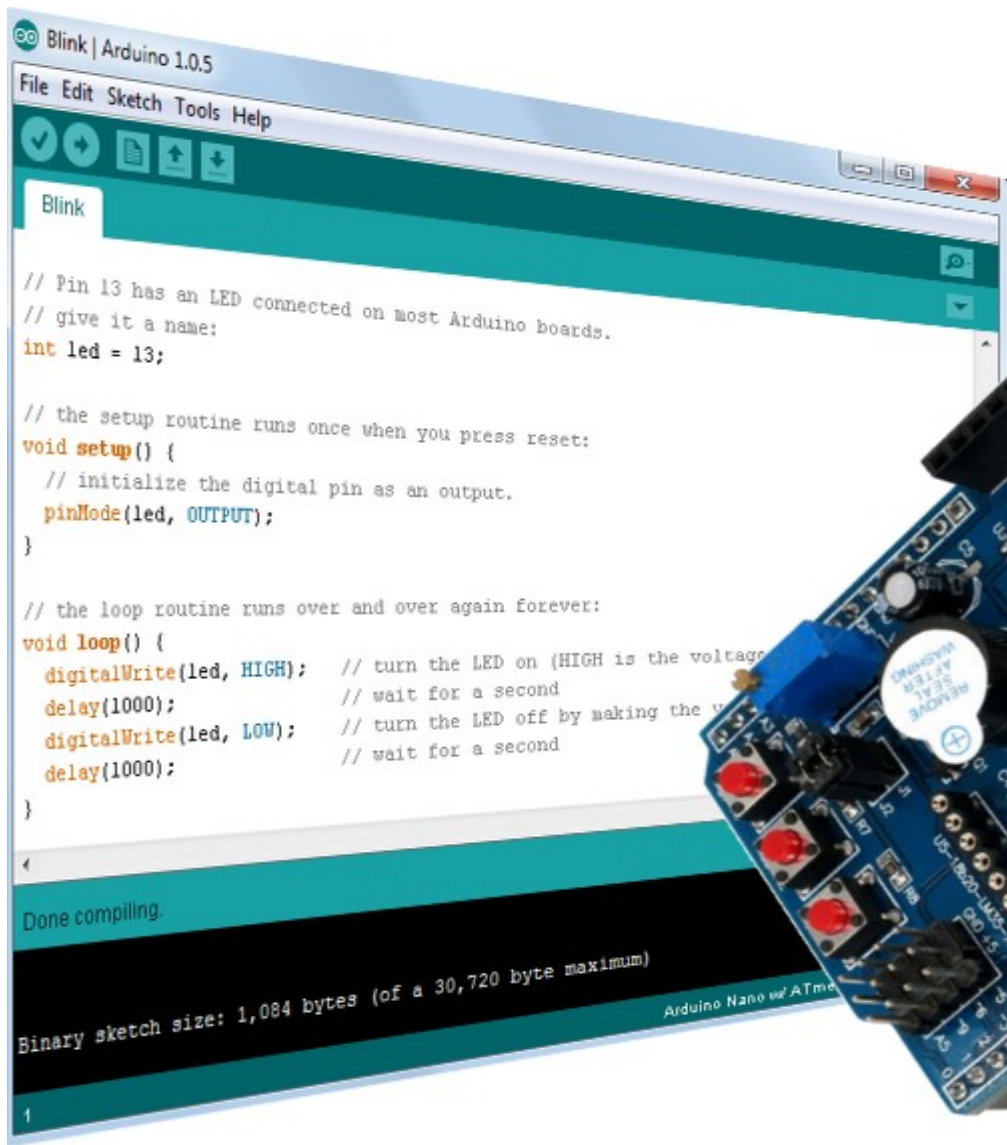


Arduino tanfolyam kezdőknek és haladóknak



10. Reflexiós fényérzékelők, akadályelkerülő robot

Felhasznált és ajánlott irodalom

Néveri Gábor (Fizikus) cikkeiből:

- FizikusRobotBlog: [Arduino alapok](#)
- FizikusRobotBlog: [Akadálykikerülő Robot v1.0](#)
- FizikusRobotBlog: [Akadálykikerülő Robot v2.0](#)
- FizikusRobotBlog: [Akadálykikerülő Robot v3.0](#)

Külföldi cikkek:

- Makerblog: [How to use a SHARP GP2Y0A21YK0F IR Distance Sensor ...](#)
- Instructables: [Using TCRT500 IR sensor with Arduino \(Remove noises\)](#)
- Arduino learning: [TCRT5000 Reflective Optical Sensor module example](#)

Korábbi előadások a témakörben

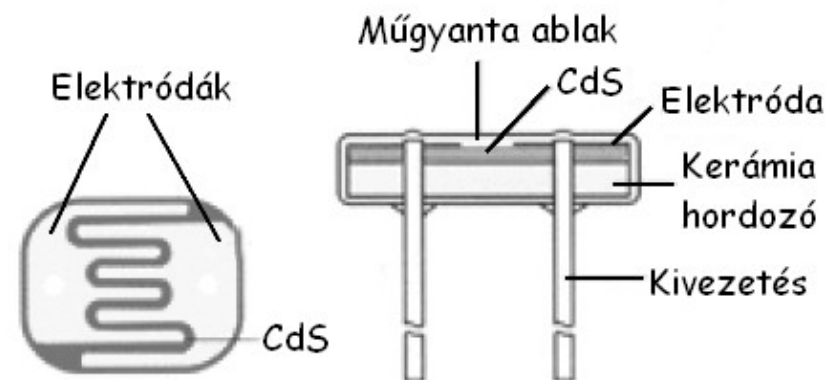
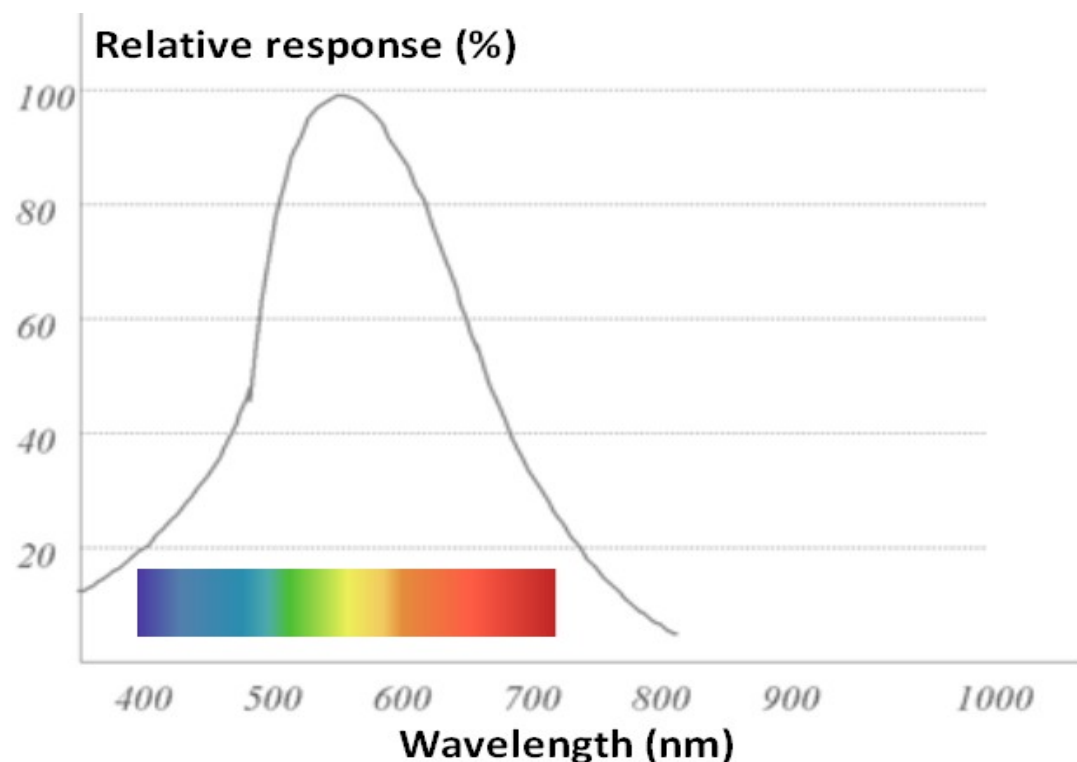
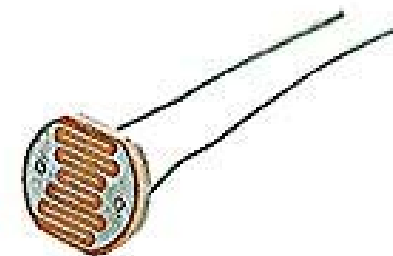
- Kisteljesítményű egyenáramú motorok vezérlése (2015. december 17.)
[📄 előadásvázlat](#) [💾 mintaprogramok](#)
- Robotvezérlés sebességmérő szenzorral (2019. január 31.)
[📄 előadásvázlat](#) [💾 mintaprogramok](#)
- Robotvezérlés WiFi kapcsolaton keresztül 1. rész (2019. február 14.)
[📄 előadásvázlat](#) [💾 mintaprogramok](#)
- Robotvezérlés WiFi kapcsolaton keresztül 2. rész (2019. február 28.)
[📄 előadásvázlat](#) [💾 mintaprogramok](#)
- Servo vezérlés, Processing keretrendszer, ultrahangos "radar"
(2020. március 5.) [📄 előadásvázlat](#) [💾 mintaprogramok](#)
- Optoérzékelők, motorvezérlés, bevezetés a robotikába (2020. március 19.)
[📄 előadásvázlat](#) [💾 mintaprogramok](#) [🎥 video](#)
- Motorvezérlés, bevezetés a robotikába (2021. január 28.)
[📄 előadásvázlat](#) [💾 mintaprogramok](#) [🎥 video](#)

E
S
P
8
2
6
6

CdS fényérzékeny ellenállás

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

A spektrális érzékenység maximuma 540 nm (sárgászöld).



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

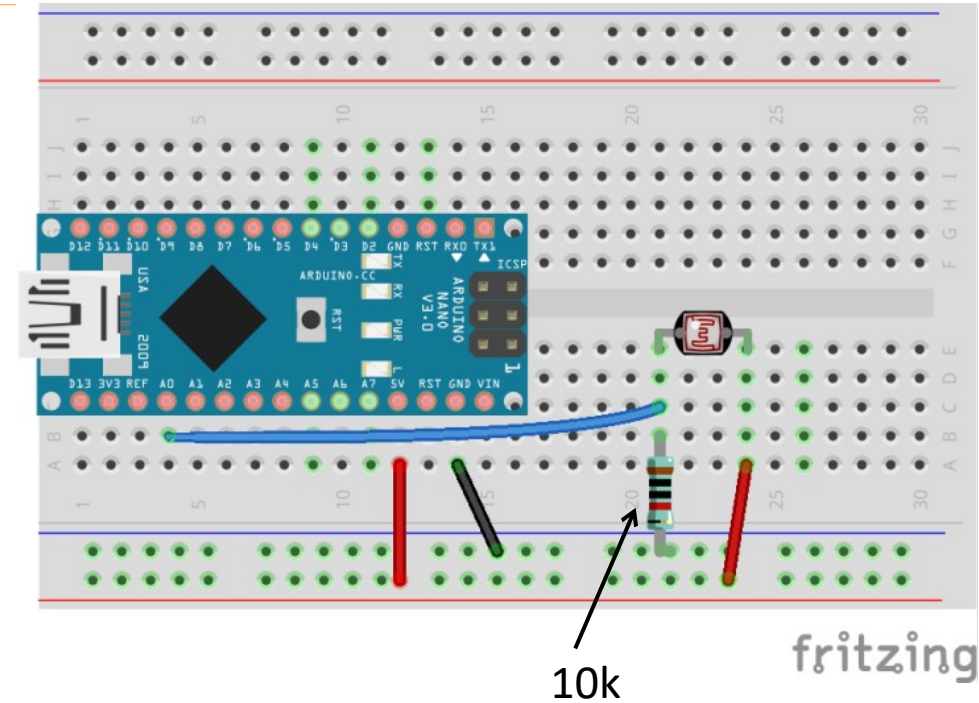
R10 és R100 a 10 és 100 lux-nál mért ellenállás értékek.

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

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 volt (5 V helyett).

Rx kiszámítása:

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

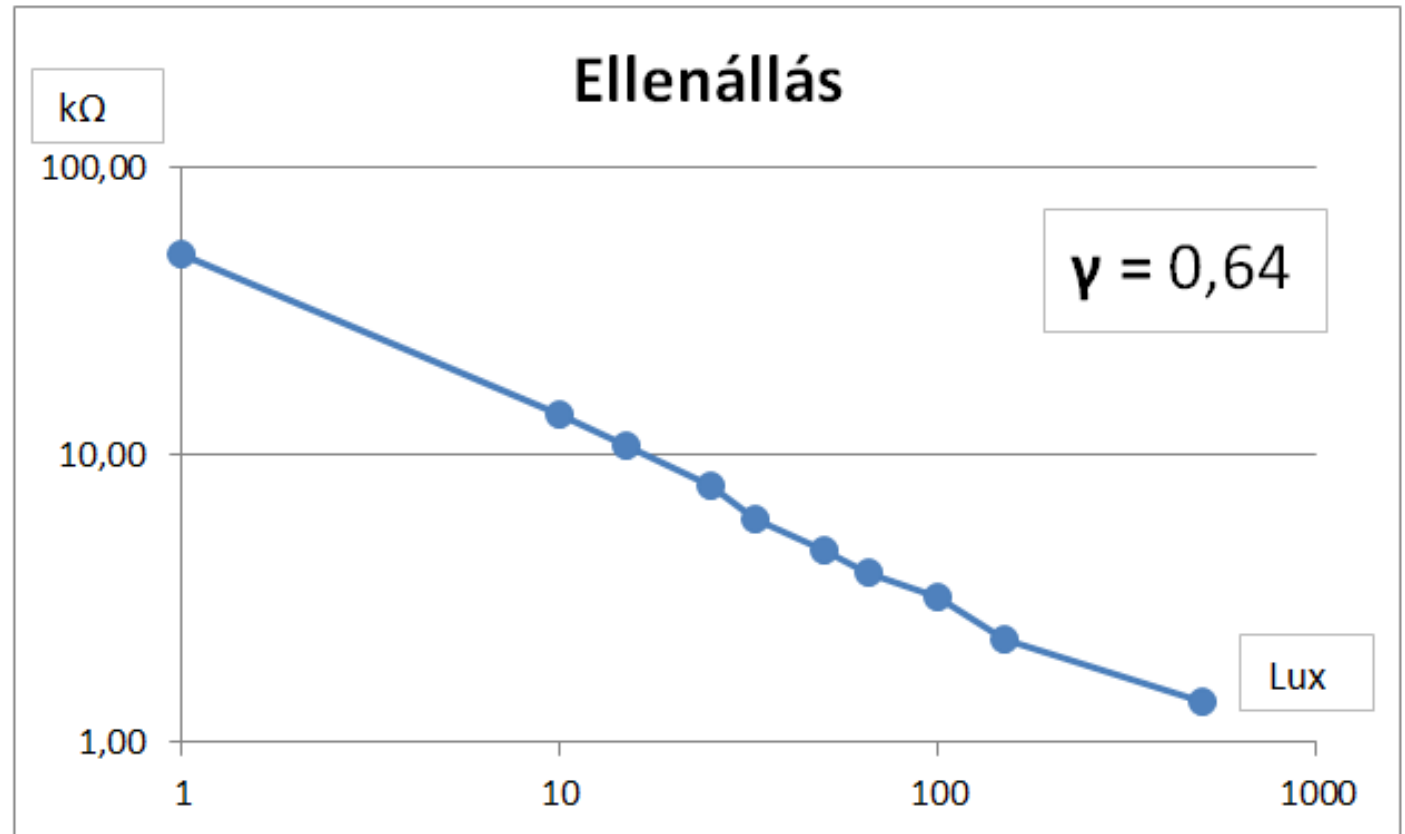
Ha Rx-et kifejezzük, akkor:

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

A fotoellenállás kalibrálása

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

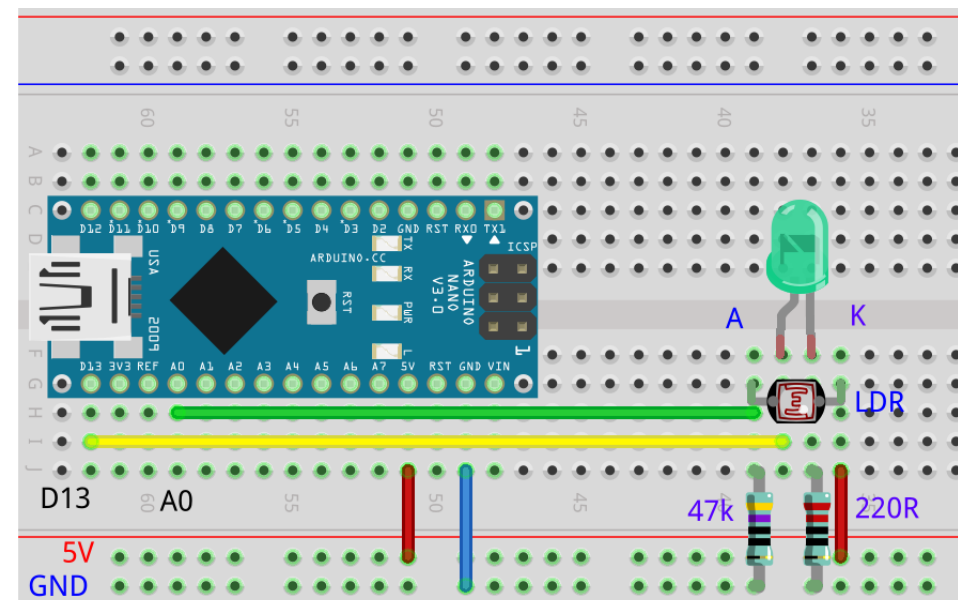
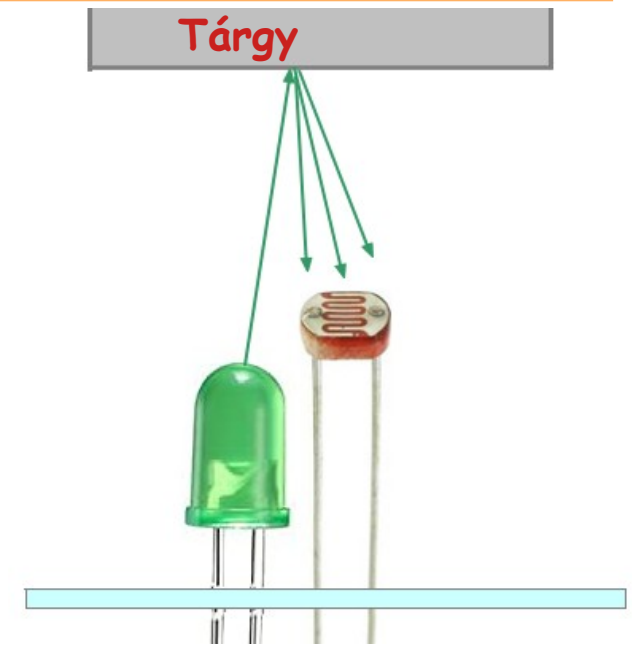
Megvilágítás [Lux]	Ellenállás [k Ω]
1	50,00
10	14,00
15	10,90
25	7,80
33	6,00
50	4,70
65	3,90
100	3,20
150	2,30
500	1,40



Mindkét skála logaritmikus!

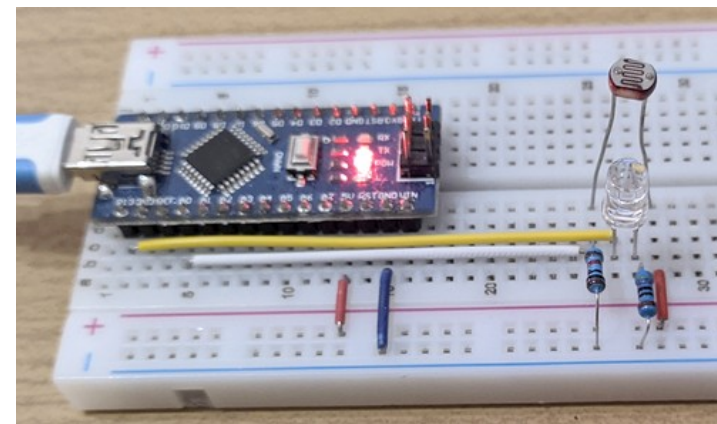
Távolság és fényelnyelés érzékelése

- A CdS fotoellenállás és egy LED felhasználásával reflexiós fényérzékelőt is készíthetünk (a LED fénye közvetlenül ne jusson rá az érzékelő felületre!)
- Ha a szenzor előtt nincs tárgy, akkor nincs visszaverődés, nincs érzékelt fény
- Ha valami visszaveri a fényt, a távolságtól és a reflexiós tényezőtől függ a visszavert fény erőssége
- A szórt háttérsugárzás (a környezetből beszűrődő fény) kiküszöbölésére két mérést végzünk: bekapcsolt és kikapcsolt LED-del, s a két mérés különbségét vesszük



cds_sensor.ino

- Be- illetve kikapcsolt LED-del végzett, 2 x 500 mérés átlagának különbségét vesszük
- A LED-et a **D13** kimenet vezérli, az analóg jelet az **A0** bemenet fogadja



```
void setup() {  
  Serial.begin(9600);  
  analogReference(DEFAULT);  
  pinMode(13, OUTPUT);  
  digitalWrite(13, LOW);  
}  
  
void loop() {  
  long a = meres(13, A0, 500);  
  Serial.print("A = ");  
  Serial.println(a);  
  delay(500);  
}
```

```
long 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;  
}
```

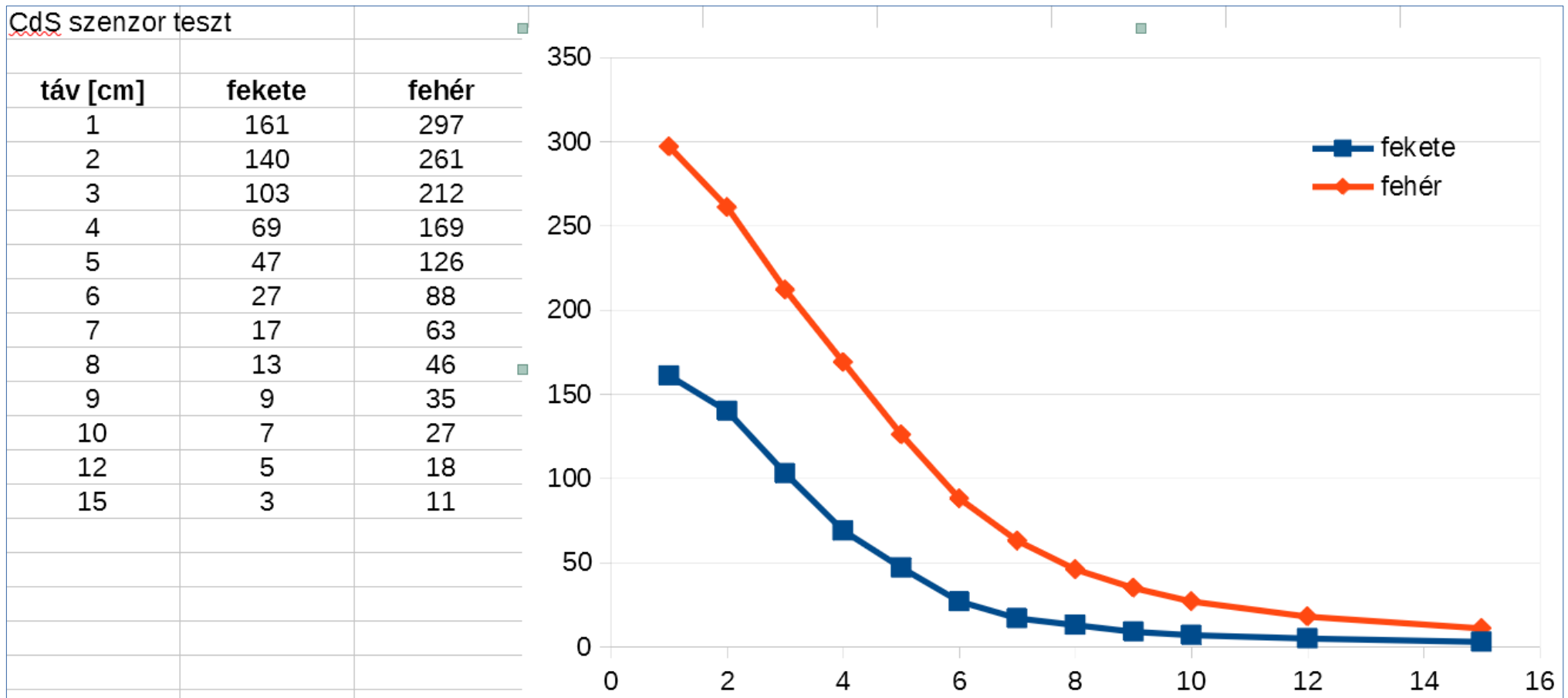

A reflexió tényező vizsgálata

- Különböző színű anyagok fényvisszaverő képességét vizsgáltuk a `cds_sensor.ino` program segítségével
- A távolságot rögzítettük (kb. 3,5 cm)
- A mért értékek az **ADC**-ből kiolvasott számok (pontosabban azok különbsége be- és kikapcsolt LED-nél), tehát önkényes egységben adottak
- A táblázatból kiolvasható eltérések teszik lehetővé, hogy a szenzorunkat pl. egy vonalkövető robotban hasznosítsuk (fehér papíron a fekete vonal jól megkülönböztethető)

A vizsgált anyag	Mért érték
Fehér papír	183
Világosbarna asztallap	158
Fekete műanyag	92

Távolságmérés CdS szenzorral

- Fehér és fekete papír bevonatú dobozok fényvisszaverő képességét vizsgáltuk a távolság függvényében (1 – 15 cm tartományban)
- Az eredményeket táblázatos formában és grafikusan is mutatjuk (az adatok most is az ADC-ből kinyert számok különbségei)

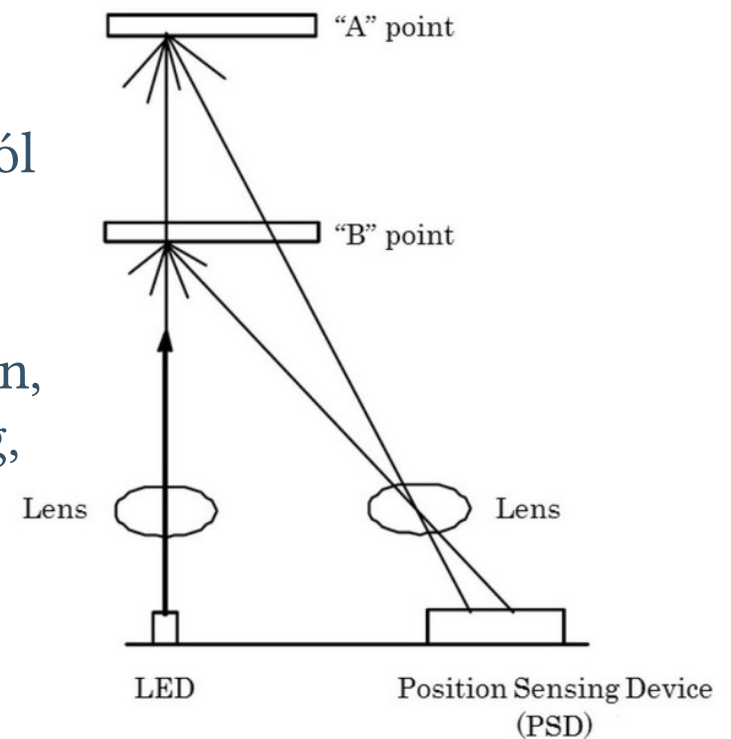


Távolságmérés CdS szenzorral



SHARP infravörös távolságmérők

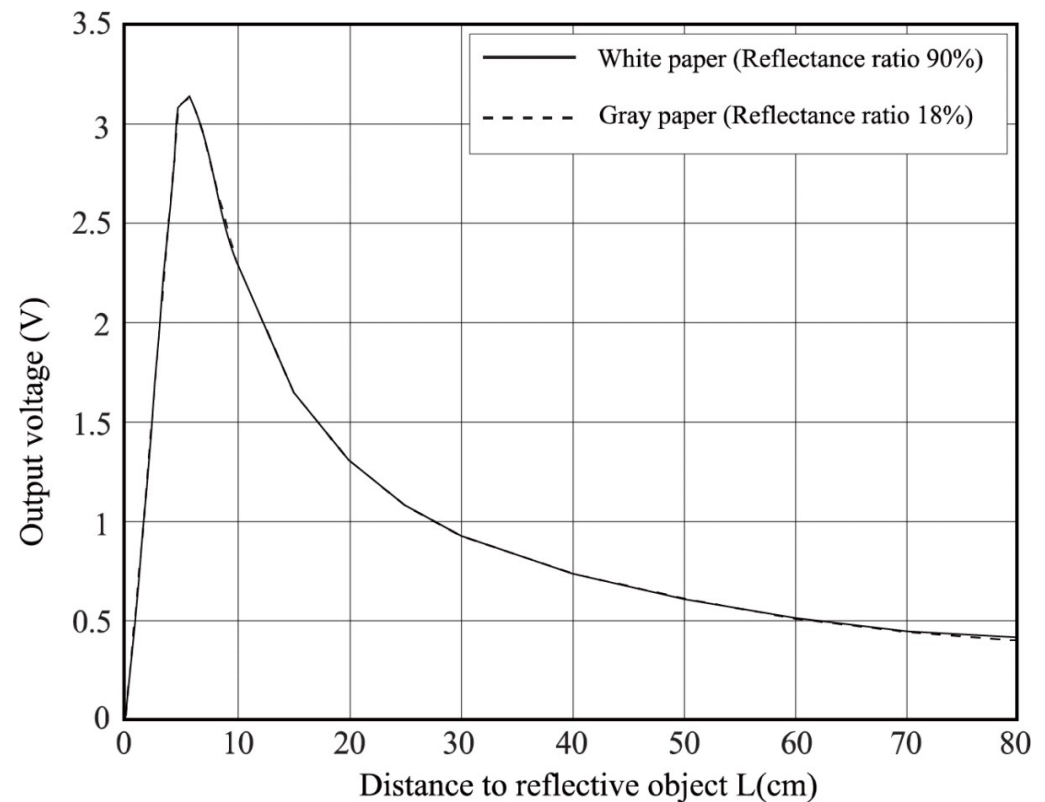
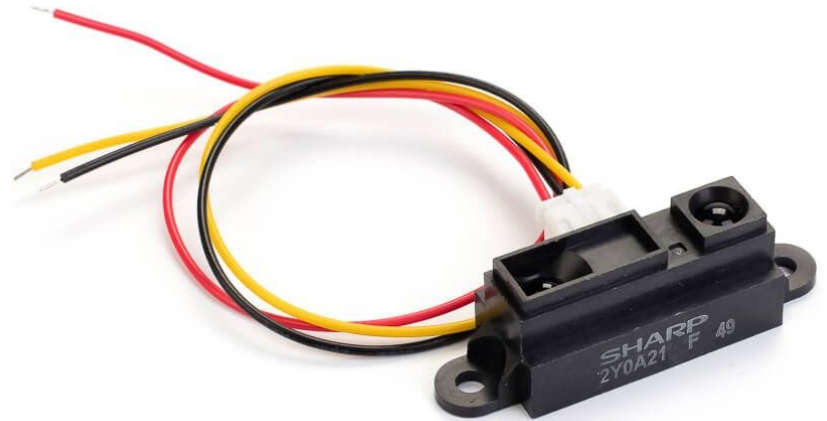
- A Sharp infravörös távolságérzékelő szenzorok háromszögletes módszerrel működnek: a kibocsátott IR fény (850 nm) az akadály távolságától függően más-más szögben visszaverődve jut a helyzetérzékelő detektorba
- Többféle típus létezik, melyek mérési tartományban, válaszidőben (16.5 – 39 ms), és kimenetben (analóg, vagy digitális jel) különböznek.
- Néhány ismertebb típust az alábbi táblázatban soroltunk fel



Típus	Mérési tartomány	Alkalmazási terület
GP2Y0A41SK0F	4 – 30 cm	Robotporszívó, szaniter alkalmazások
GP2Y0A21YK0F	10 – 80 cm	Érintésmentes kapcsolók
GP2Y0A02YK0F	20 – 150 cm	Érintésmentes kapcsolók
GP2Y0A710K0F	100 – 550 cm	Projektor autofókusz, emberi jelenlét detektor

A SHARP GP2Y0A21YK0F IR szenzor

- A jelleggörbéből látszik a SHARP szenzorok egyik negatív tulajdonsága: nagyon kis távolságoknál ugyanakkor jelet ad, mint nagy távolságoknál
- Ez a típus pl. csak 10 cm-től képes mérni, ha egyértelmű adatot akarunk kapni
- Adatlap: [GP2Y0A21YK.pdf](#)

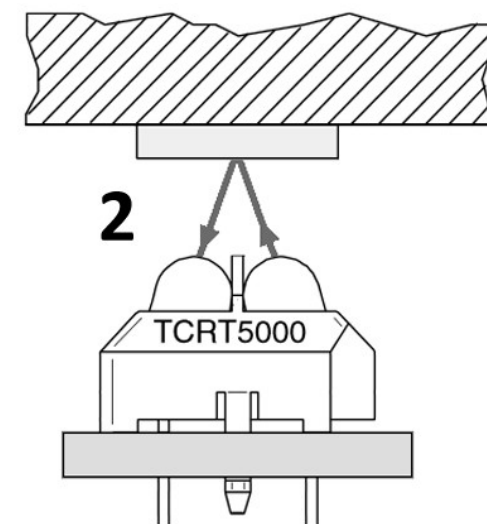
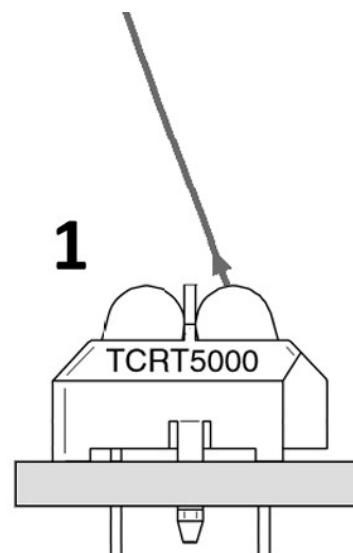
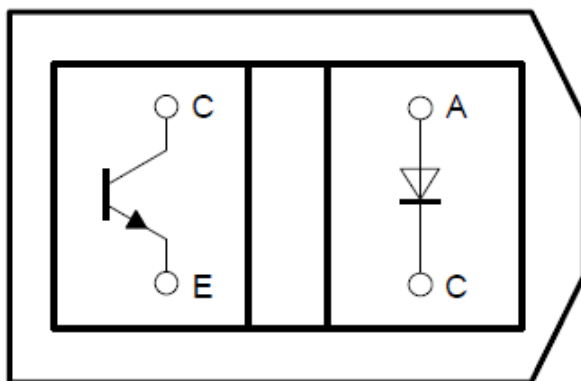
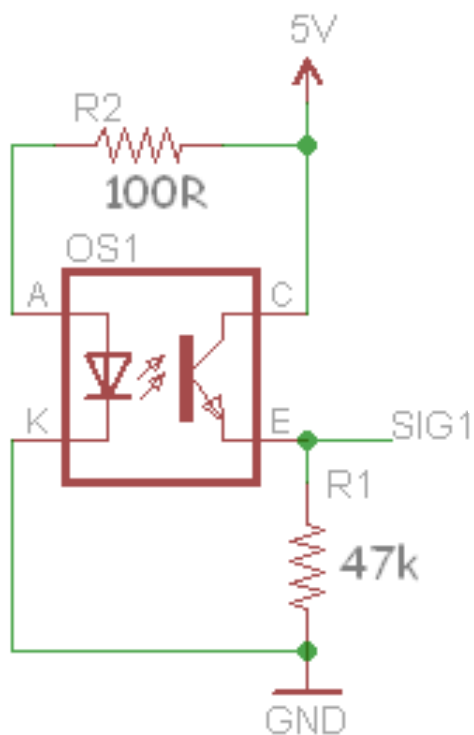


TCRT5000 reflektív optikai érzékelő

Egy IR LED-et és egy fototranzisztort tartalmaz.

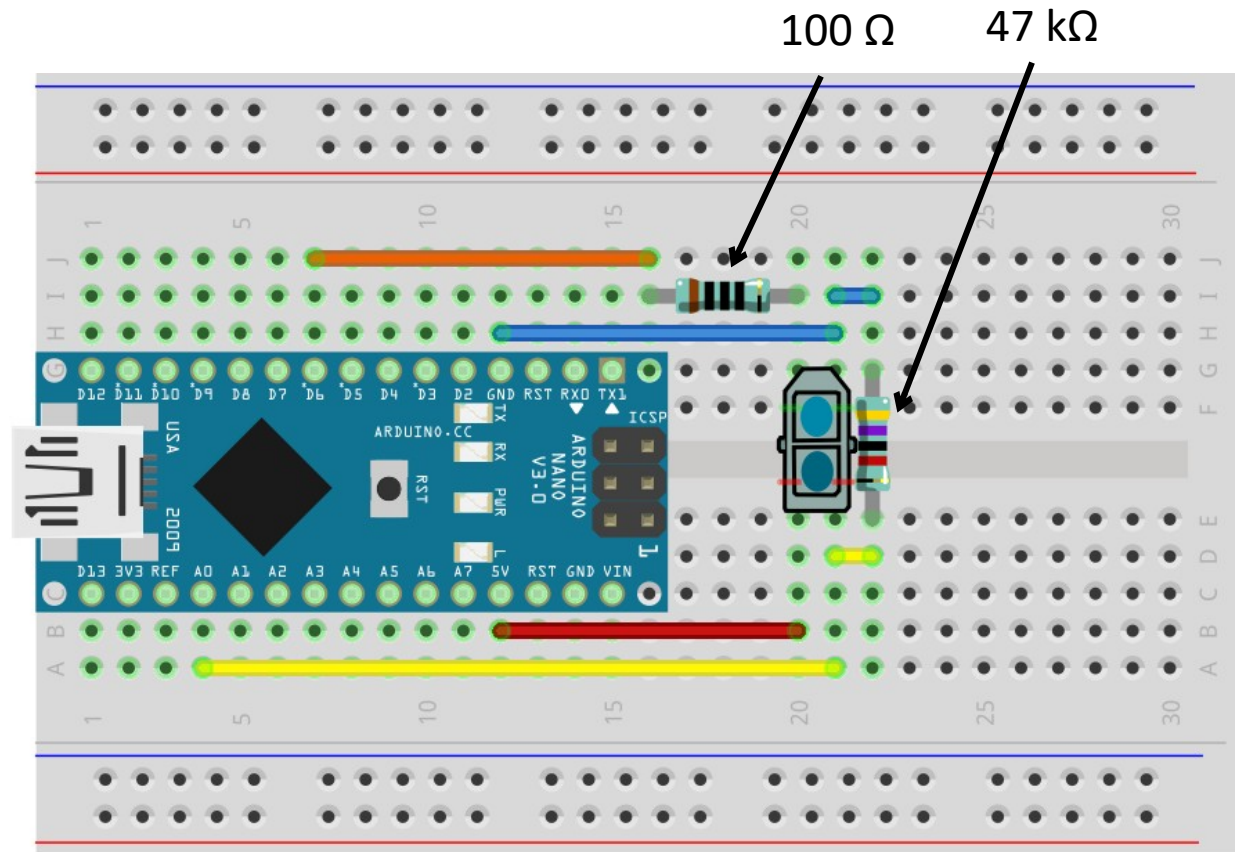
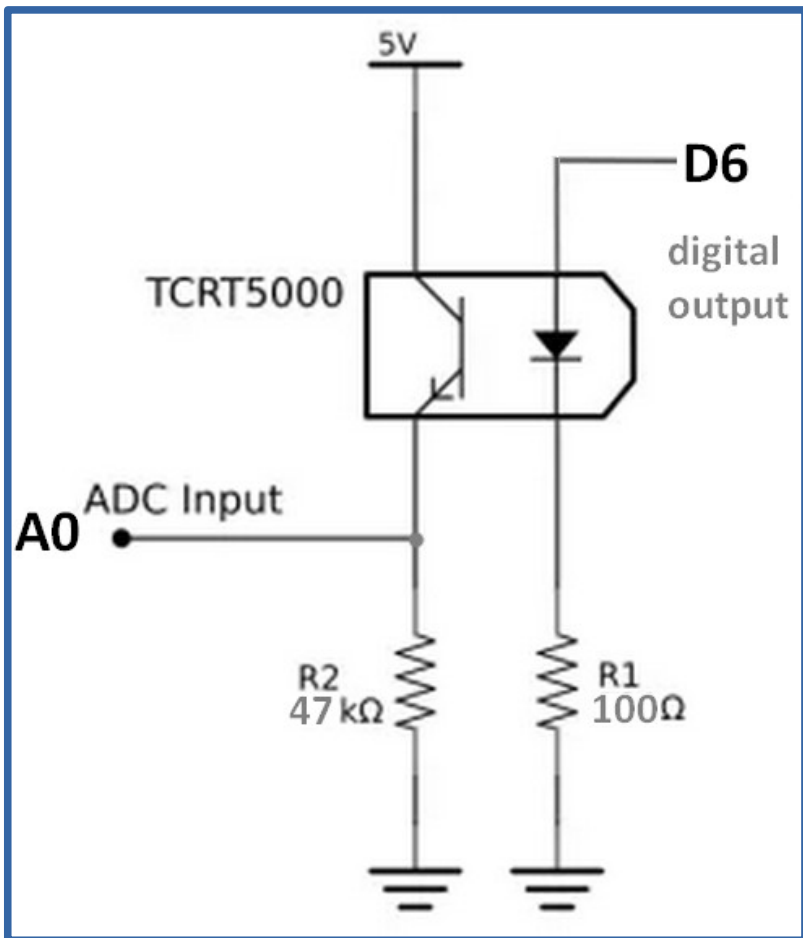
Működési elv: ha nincs akadály, a fény nem verődik vissza, a tranzisztor nem érzékel fényt.

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 besűrődő 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

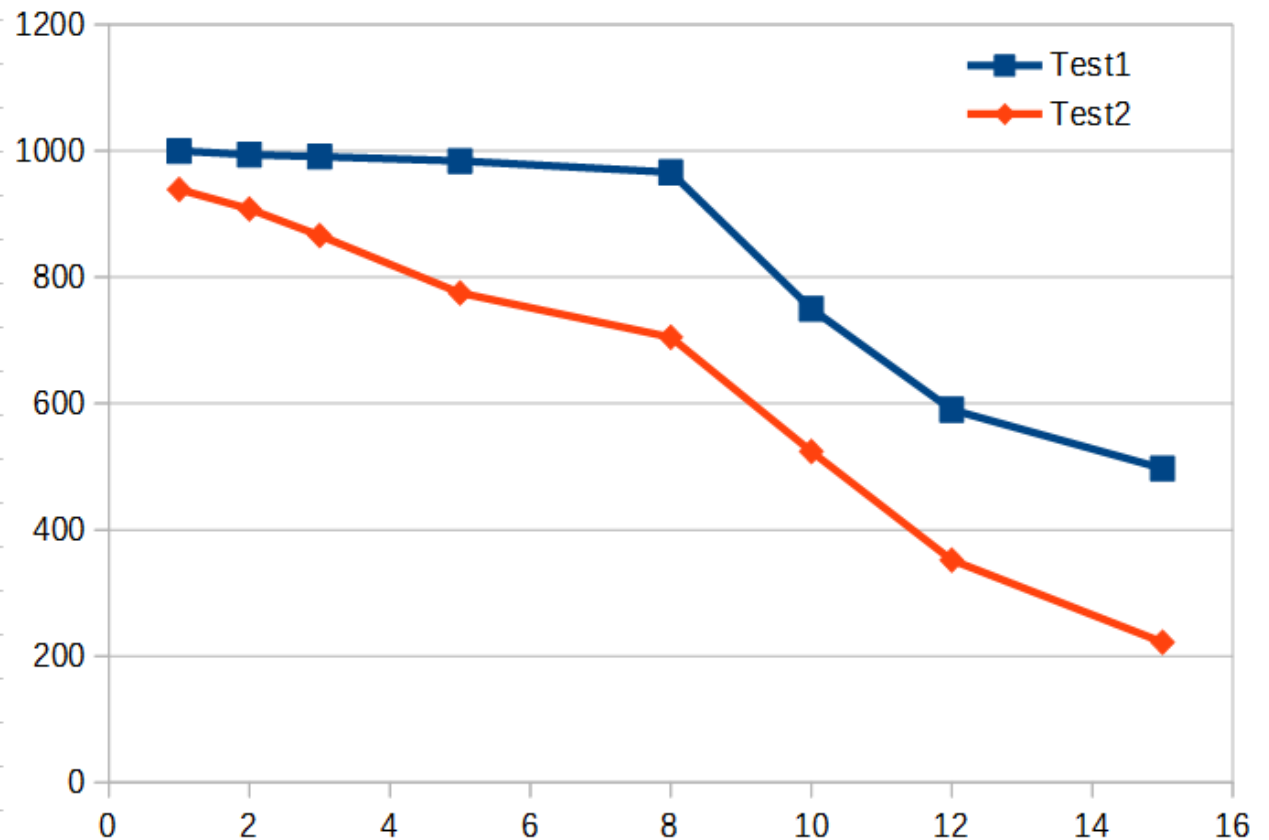
```
void setup() {  
  Serial.begin(9600);  
  analogReference(DEFAULT);  
  pinMode(6, OUTPUT);  
  digitalWrite(6, LOW);  
}  
  
void loop() {  
  int a = meres(6, A0, 500);  
  Serial.print("A = ");  
  Serial.println(a);  
  delay(500);  
}
```

```
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

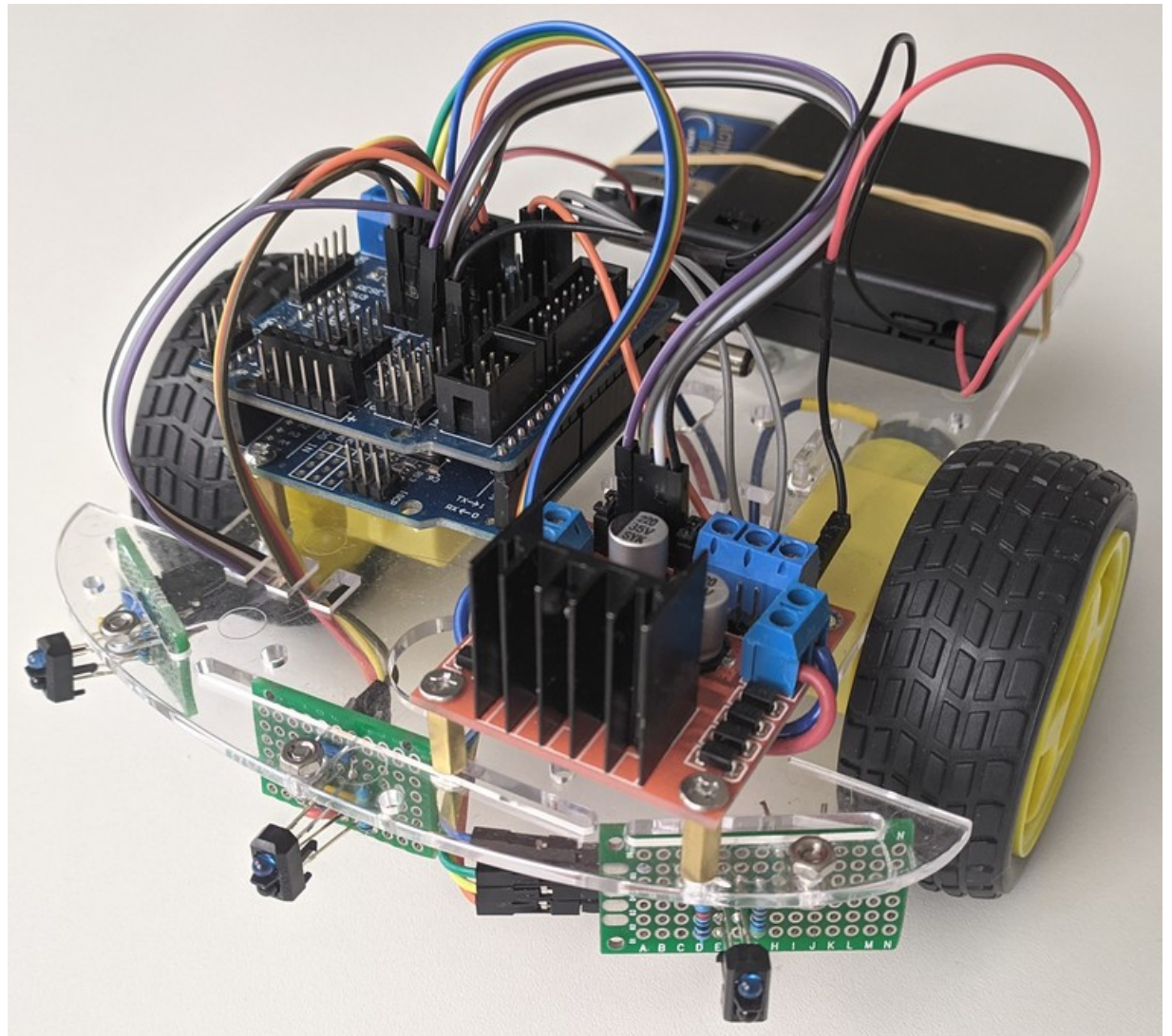
- Az alábbi táblázatban és ábrán összehasonlítottuk a két program eredményét. A különbséget a beszűrődő fény okozza

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



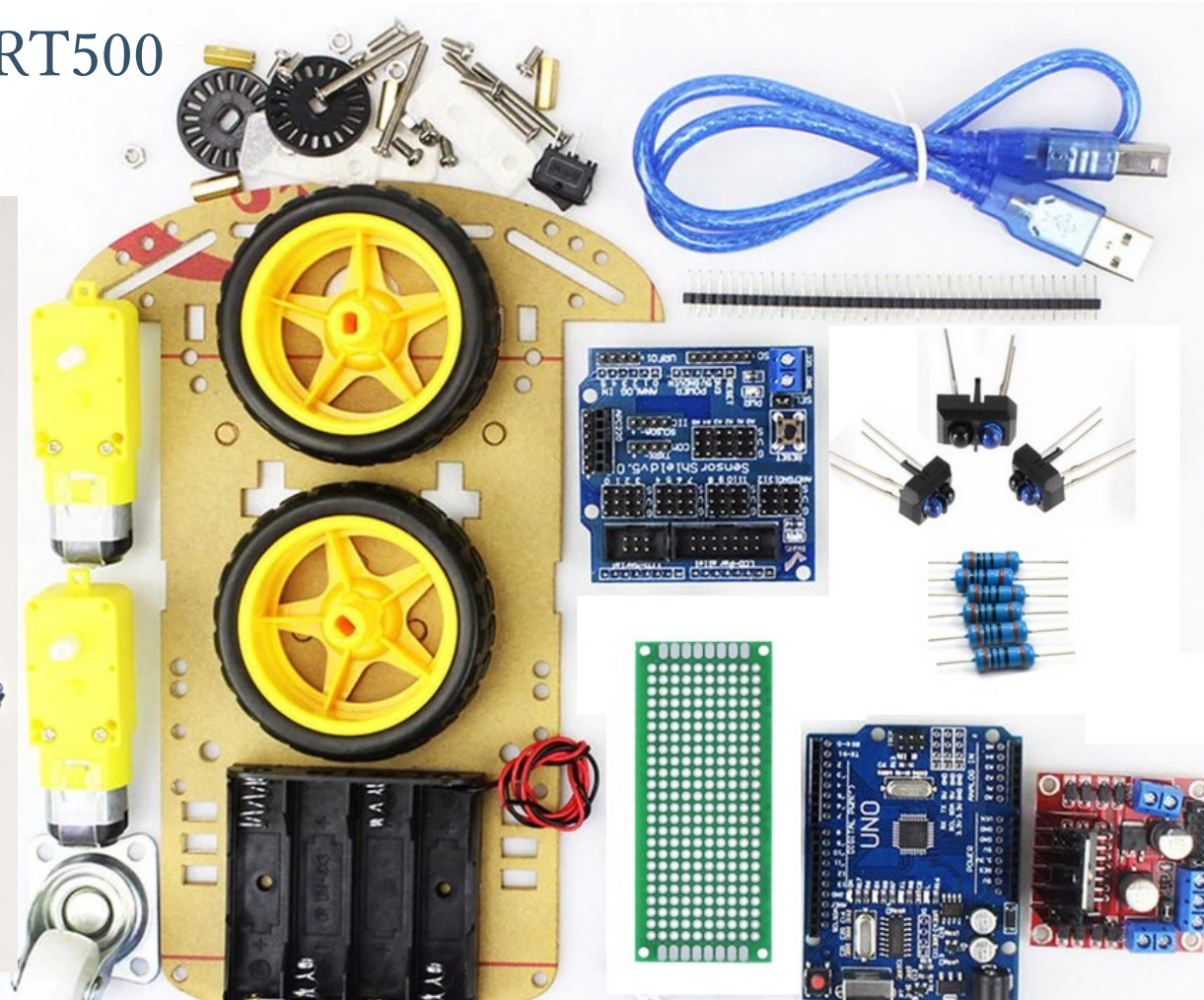
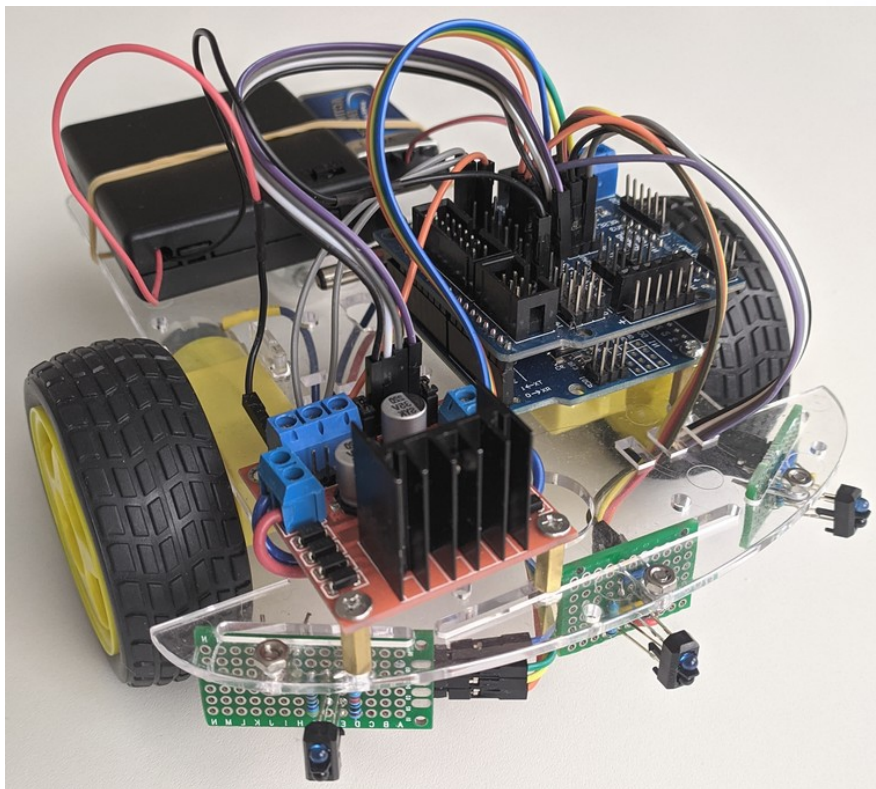
Akadálykikerülő robot

- Két kerék meghajtású (2WD) robotalváz
- Három, rögzített irányú,
- TCRT5000 szenzor
- A reflexiós optikai szenzorok jó fényvisszaverő képességű anyagok esetén 1-30 cm távolságból észlelik az akadályokat
- A különböző irányból észlelt akadályok kikerülésére alkalmas stratégiát kell választani



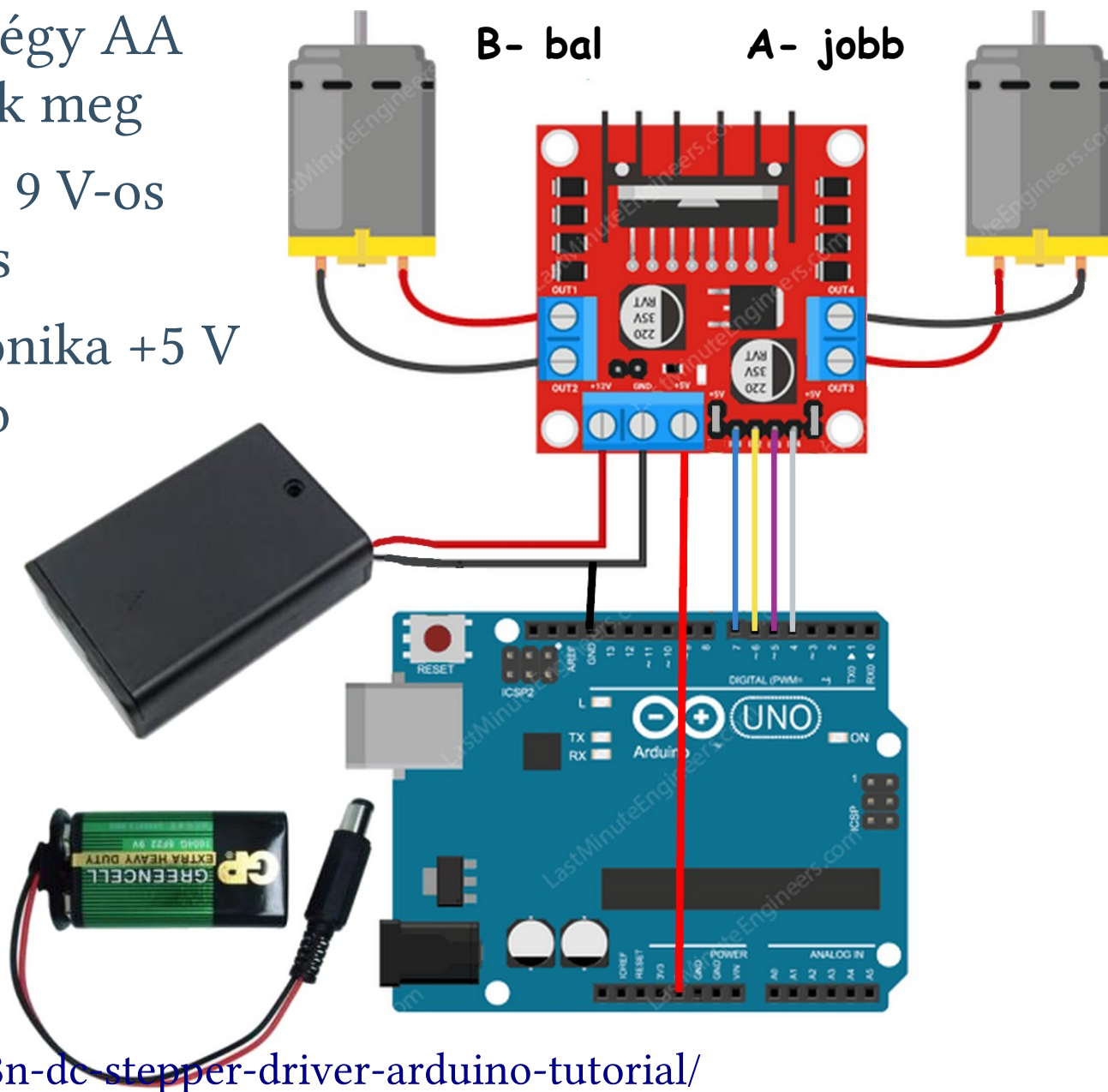
A hozzávalók

- Az előző előadásban már bemutatott 2WD robot alvázat most is egy *Arduino UNO* kártya, egy **L298N** motorvezérlő panel és egy szenzor fedlap segítségével hajtjuk meg
- Az érzékelő most 3 db TCRT500 szenzor lesz



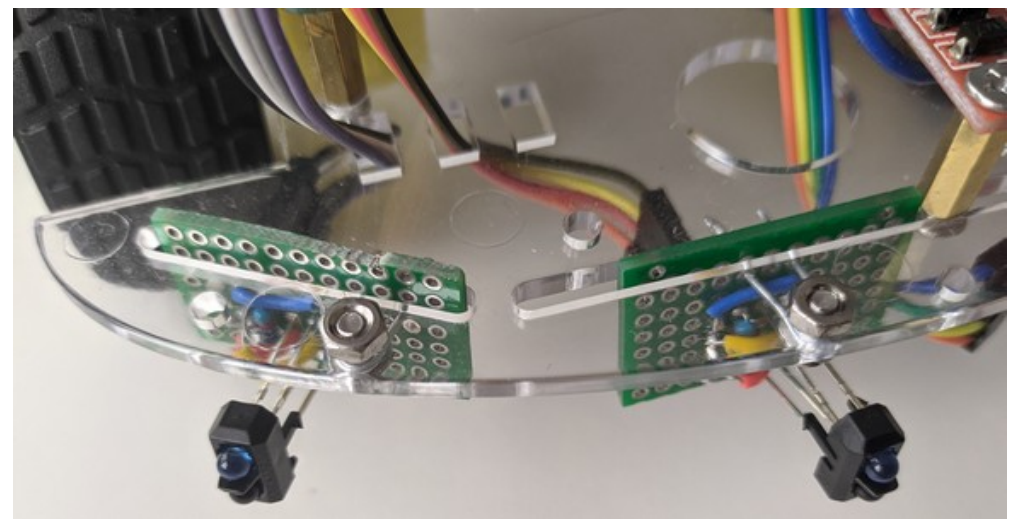
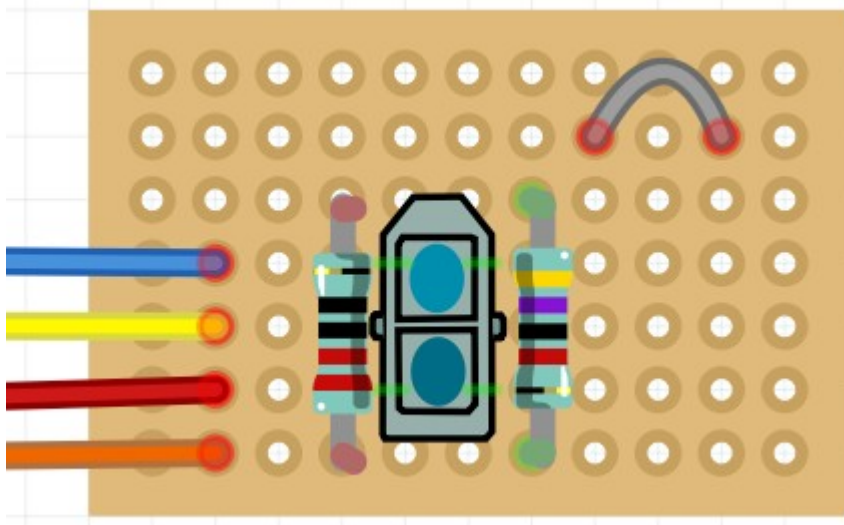
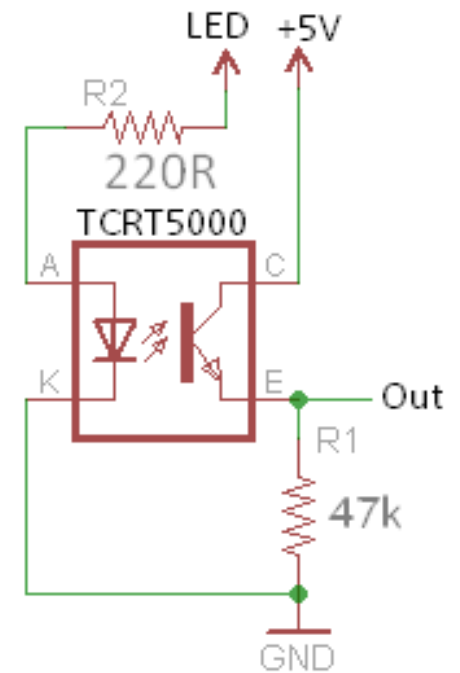
A motorvezérlés kapcsolási vázlata

- 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 tükörszimmetrikus bekötése hardveresen biztosítja az ellentétes irányú forgást a bal és jobb motornál
- Felhasznált forrás:
lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/



Több szem többet lát!

- Barkácsoljunk három szenzort TCRT5000 IR érzékelők felhasználásával és rögzítsük azokat a képen látható módon!
- A rögzítést egy, a panelbe forrasztott drót "fül" (gémkapocs) és egy M3-as csavar segítségével oldottuk meg
- A kimeneteket az Arduino A0, A1, A2 bemeneteire, a LED vezérlést a D8, D9 D10 kimenetekre kötjük



Akadálykikerülési stratégia

Induljunk ki a FizikusRobotBlog Akadálykikerülő Robot v1.0 projektjéből, s módosítsuk azt!

Eltérések:

- Az általunk használt szenzorok érzékenysége kisebb → más határértékeket kell választani
- Mi a háttérfény hatását a LED-ek ki-be kapcsolásával kiszűrjük → több mérés, több kivezetés
- A fenti projektben túl sok a balra fordulás → az első elágazás után újabb feltételvizsgálatot vezetünk be
- A fenti projekt folyamatábrája és a programkód nem egyezik → próbájuk ki, melyik jobb!



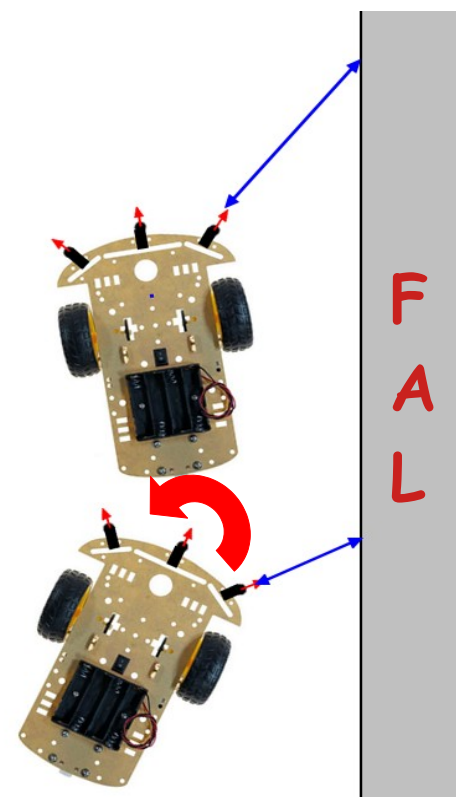
Érdemes lenne megvizsgálni, hogy itt balra vagy jobbra célszerű fordulni!

Ha akadály van előttünk...

- Induljunk el előre!
- Mérjük a szenzorokkal balra, jobbra előre (nagy szám=közeli akadály!)
- Ha akadály van előttünk (középső szenzor értéke a felső határon felüli), akkor nézzük meg, balra, vagy jobbra van több hely, arra fordulunk!
(az eredeti folyamatára szerint itt mindig jobbrafordulás volt)
 - ❖ Addig fordulunk a választott irányba, amíg az ellenkező oldali és a középső szenzor jele az alsó küszöb alá nem csökken (közben adott időközönként újra mérjük a vizsgált két szenzor jelét)
 - ❖ Előre menetbe kapcsolunk
- **Megjegyzés1:** Az eredeti folyamatára szerint innen a **loop()** függvény végére kell ugrani (mi is ezt tesszük), a leközölt program azonban nem így volt megírva. „Házi feladatként” próbáljuk ki, hogy mi változást okoz a robot viselkedésében ez a különbség!
- **Megjegyzés2:** az alsó és felső határérték megkülönböztetésre azért van szükség, hogy legyen egy kis hiszterézis a rendszerben

Ha akadály van jobbról vagy balról...

- Ha a jobb, vagy a baloldali szenzor akadályt észlel (a szenzor által adott érték nagyobb, mint az előírt felső határérték) akkor az ellenkező irányba fordulunk
- A fordulás addig tart, amíg az akadályt észlelő szenzor jele az alsó határérték alá nem csökken (közben adott időközönként újra mérünk a szenzorral)
- Ha sikerült kikerülni az akadályt, akkor előremenetbe kapcsolva megyünk tovább
- **Megjegyzés:** Az eredeti folyamatábra szerint innen a **loop()** függvény végére kell ugrani (mi is ezt tesszük), a leközölt program azonban nem így volt megírva. „Házi feladatként” próbáljuk ki, hogy mi változást okoz a robot viselkedésében ez a különbség!



opto_robot.ino 3/1.

- A program motorvezérléssel kapcsolatos része megegyezik az előző előadásban bemutatott robotvezérlő programmal
- Bekötés: **A0/D10** – bal szenzor, **A1/D9** – középső szenzor, **A2/D8** – jobb szenzor, **D7/D6** – bal motor, **D5/D4** – jobb motor

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

#define L_sense A0 // Bal szenzor
#define L_LED 10
#define M_sense A1 // Középső szenzor
#define M_LED 9
#define R_sense A2 // Jobb szenzor
#define R_LED 8
#define NMEAS 100 // Mérések száma

#define L_LIMIT_LOW 50 // Bal limit
#define L_LIMIT 100
#define M_LIMIT_LOW 30 // Középső limit
#define M_LIMIT 40
#define R_LIMIT_LOW 50 // Jobb limit
#define R_LIMIT 100
```

```
void setup() {
  analogReference(DEFAULT);

  //-- A TCRT5000 LED vezérlő kivezetések
  pinMode(L_LED, OUTPUT); // bal
  pinMode(M_LED, OUTPUT); // középső
  pinMode(R_LED, OUTPUT); // jobb

  //-- Motorvezérlő lábak konfigurálása
  pinMode(B1A, OUTPUT);
  pinMode(B1B, OUTPUT);
  pinMode(A1A, OUTPUT);
  pinMode(A1B, OUTPUT);
  digitalWrite(B1A, LOW);
  digitalWrite(B1B, LOW);
  digitalWrite(A1A, LOW);
  digitalWrite(A1B, LOW);
}
```

opto_robot.ino 3/2.

```
uint32_t meres(int p_sense, int p_LED) {
    uint32_t sum0, sum1;
    digitalWrite(p_LED, LOW);
    sum0 = 0;
    for (int i = 0; i < NMEAS; i++) {
        sum0 += analogRead(p_sense);
    }
    digitalWrite(p_LED, HIGH);
    sum1 = 0;
    for (int i = 0; i < NMEAS; i++) {
        sum1 += analogRead(p_sense);
    }
    if (sum0 > sum1) sum0 = sum1;
    return ((sum1 - sum0) / NMEAS);
}

void motor_forward() {
    digitalWrite(B1A, HIGH);
    digitalWrite(B1B, LOW);
    digitalWrite(A1A, HIGH);
    digitalWrite(A1B, LOW);
}
```

```
void motor_reverse() {
    digitalWrite(B1A, LOW);
    digitalWrite(B1B, HIGH);
    digitalWrite(A1A, LOW);
    digitalWrite(A1B, HIGH);
}

void motor_left() {
    digitalWrite(B1A, LOW);
    digitalWrite(B1B, HIGH);
    digitalWrite(A1A, HIGH);
    digitalWrite(A1B, LOW);
}

void motor_right() {
    digitalWrite(B1A, HIGH);
    digitalWrite(B1B, LOW);
    digitalWrite(A1A, LOW);
    digitalWrite(A1B, HIGH);
}

void motor_stop() {
    digitalWrite(B1A, LOW);
    digitalWrite(B1B, LOW);
    digitalWrite(A1A, LOW);
    digitalWrite(A1B, LOW);
}
```

opto_robot.ino 3/3.

- A `loop()` függvény valósítja meg az akadálykikerülő stratégiát

```
void loop() {
  motor_forward();
  uint32_t bal = meres(L_sense, L_LED);
  uint32_t kozep = meres(M_sense, M_LED);
  uint32_t jobb = meres(R_sense, R_LED);
  if (kozep > M_LIMIT) {
    if ( bal < jobb ) {
      motor_left();
      while ((jobb > R_LIMIT_LOW) ||
             (kozep > M_LIMIT_LOW)) {
        kozep = meres(M_sense, M_LED);
        jobb = meres(R_sense, R_LED);
        delay(50);
      }
    } else {
      motor_right();
      while ((bal > L_LIMIT_LOW) ||
             (kozep > M_LIMIT_LOW)) {
        kozep = meres(M_sense, M_LED);
        bal = meres(L_sense, L_LED);
        delay(50);
      }
    }
  }
}
```

```
// Ha jobbról akadályt észlelünk...
else if (jobb > R_LIMIT) {
  motor_left(); // balra fordulunk
  while (jobb > R_LIMIT_LOW) {
    jobb = meres(R_sense, R_LED);
    delay(50);
  }
}

// Ha balról akadályt észlelünk...
else if (bal > L_LIMIT) {
  motor_right(); // jobbra fordulunk
  while (bal > L_LIMIT_LOW) {
    bal = meres(L_sense, L_LED);
    delay(50);
  }
}
}
```

opto_robot.ino: tapasztalatok

- *„Válámi ván, dé ném áz igázi...” / Arkagyij Rajkin/*

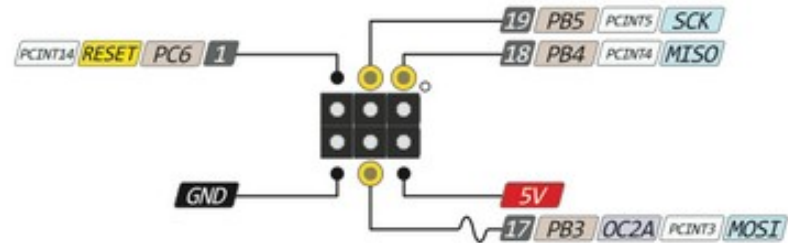


Az Arduino nano kártya kivezetései

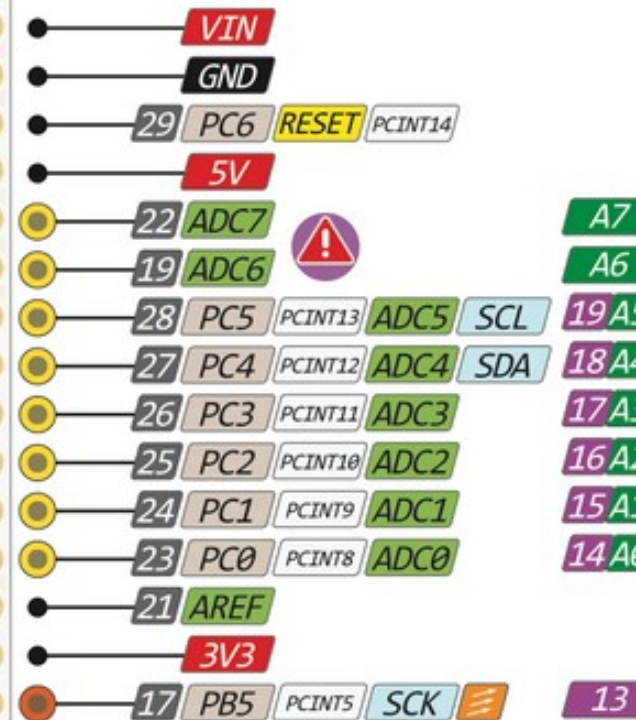
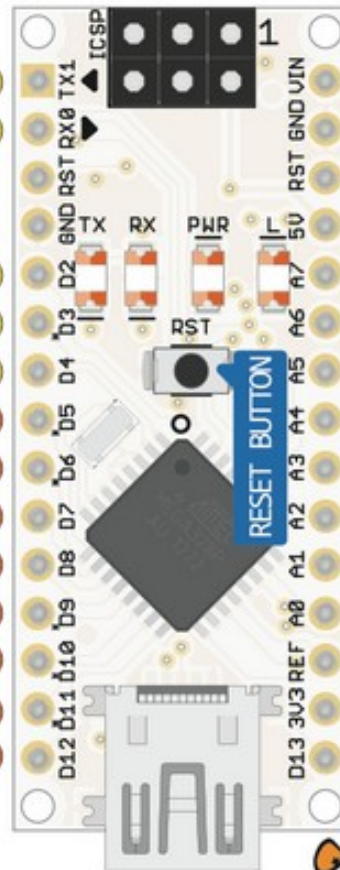
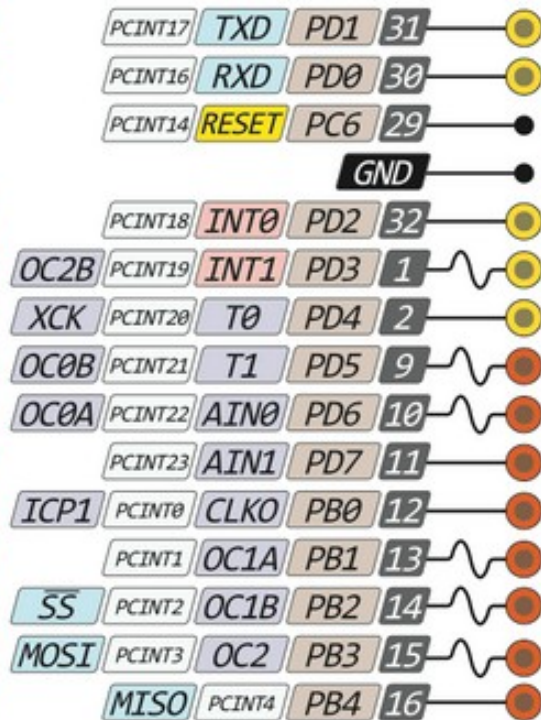


NANO PINOUT

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



- 1
- 0
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12



- 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