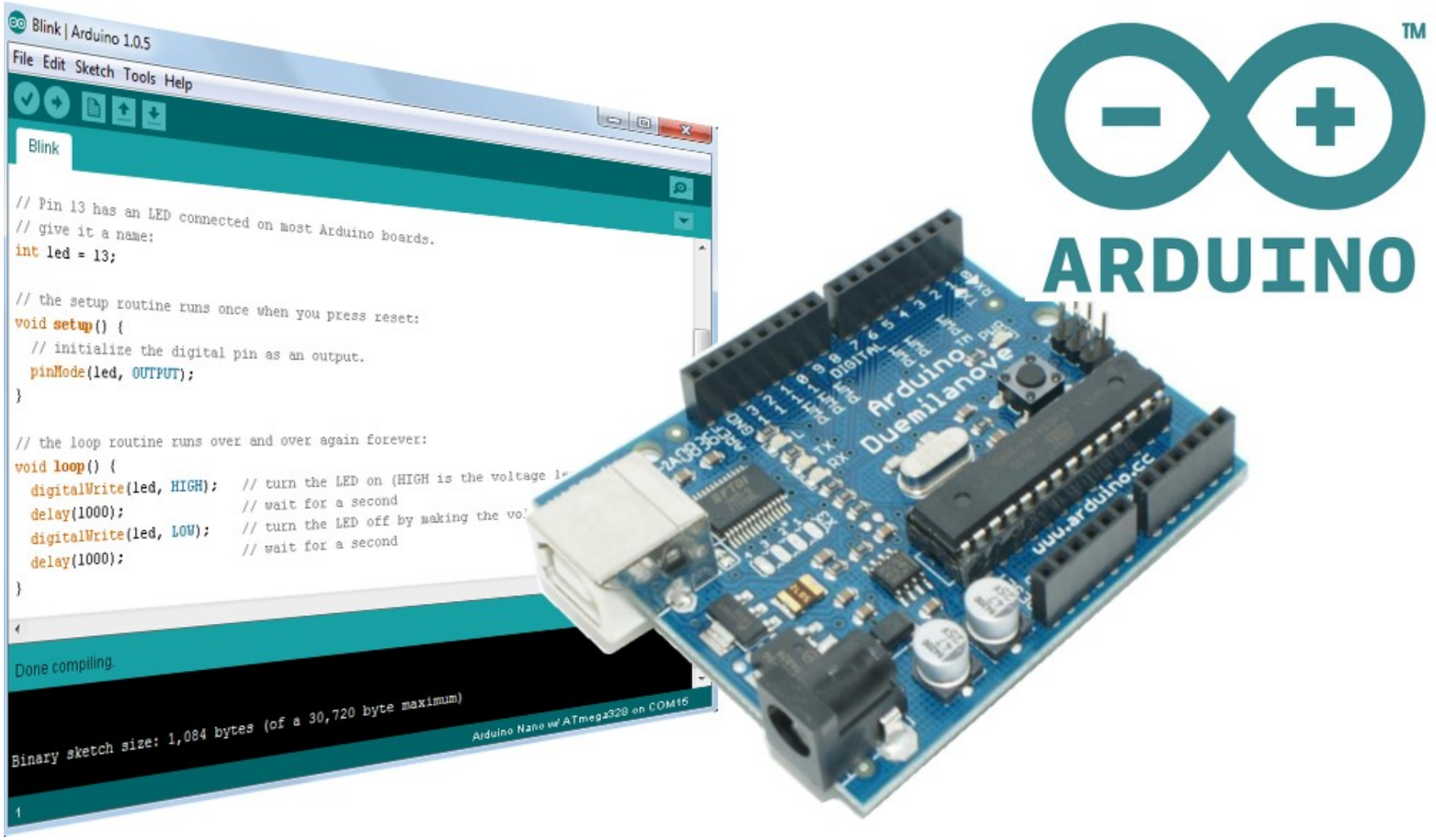
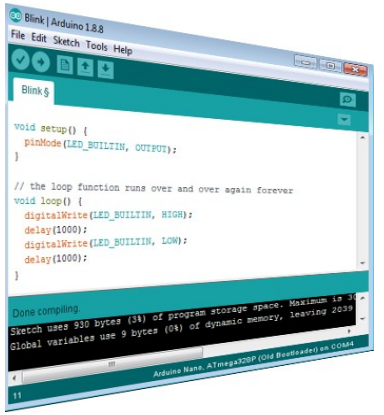
Bevezetés az elektronikába



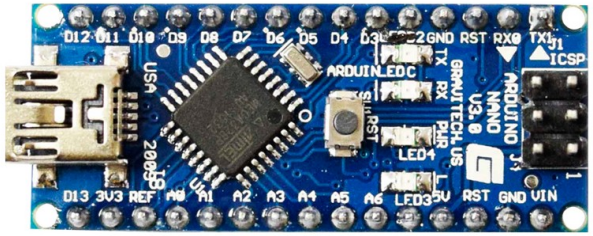
8. Ismerkedés az Arduino kártyával (MiniPirate) – 2. rész

Mi az Arduino?

- Az **Arduino** egy szabad szoftveres, nyílt forráskódú elektronikai fejlesztőplatform, vagy ökoszisztéma az elektronikus eszközök könnyen megtanulható kezeléséhez



  - **Arduino IDE** (integrált fejlesztői környezet): Java alapú, keresztplatformos fejlesztői környezet (szerkesztő, fordító, programletöltő stb.)
  - **Arduino kártya: ATmega328P** vagy más mikrovezérlőn alapuló hardver, amely önállóan vagy a számítógéppel összekapcsolva is működhet



  - **Arduino programnyelv és programkönyvtár-gyűjtemény**: amely lehetővé teszi, hogy a mikrovezérlő részleteinek pontos ismerete nélkül, egyszerűen írhassunk programot

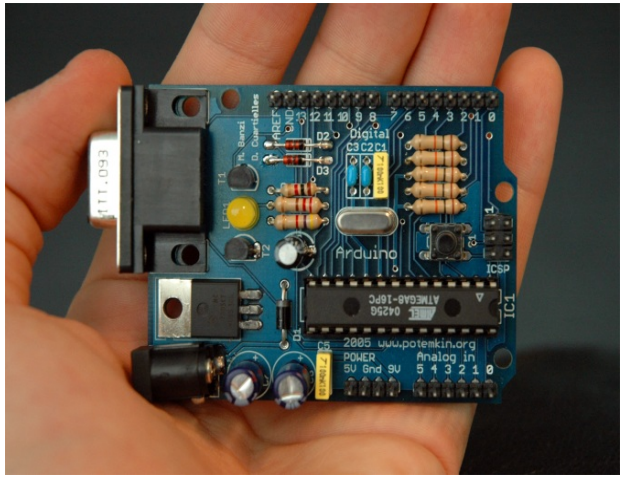
```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
}
```

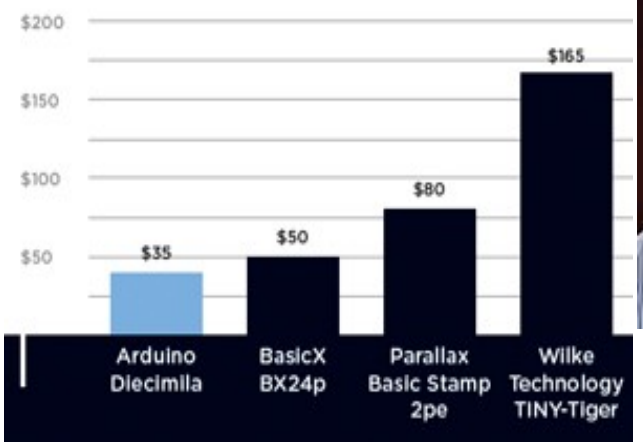
Az Arduino születése

- **2005**-ben az Ivreában az **Interaction Design Institute** tanárai és diákjai fejlesztették ki.
- Cél: olcsó és egyszerűen használható mikrovezérlős fejlesztőeszköz (hardver és szoftver) létrehozása, amellyel a diákok vagy hobbisták rövid idő alatt (~ 1 hó) interaktív eszközöket tudnak alkotni
- **Előzmények:**

Processing – nyíltforrású programnyelv és IDE (Casey Reas, Benjamin Fry)

Wiring – Nyílforrású mikrovezérlős fejlesztőkártya és programnyelv (Hernando Barragán)





Gianluca Martino | Massimo Banzi | David Cuertiellas

A Wiring koncepció

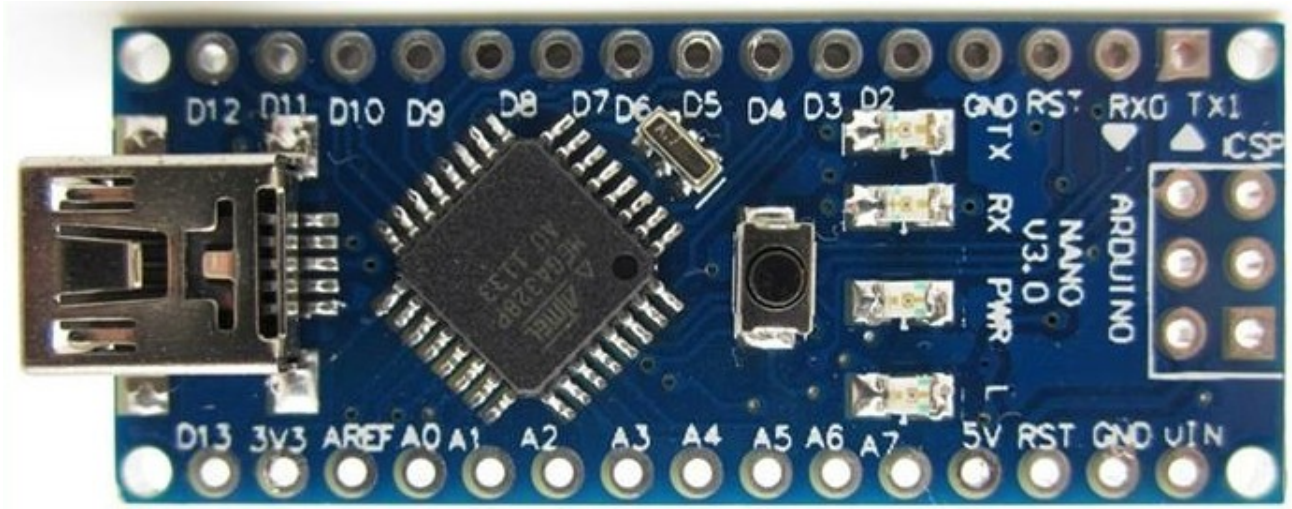
- A Wiring koncepció, amely kártyát, programnyelvet és fejlesztői környezetet is jelent számos követőre talált.
- Leszármazottjai közé sorolható az Arduino, az Energia, a Leaflabs Maple, a ChipKit MPIDE és még sokan mások...

Miért az Arduino?

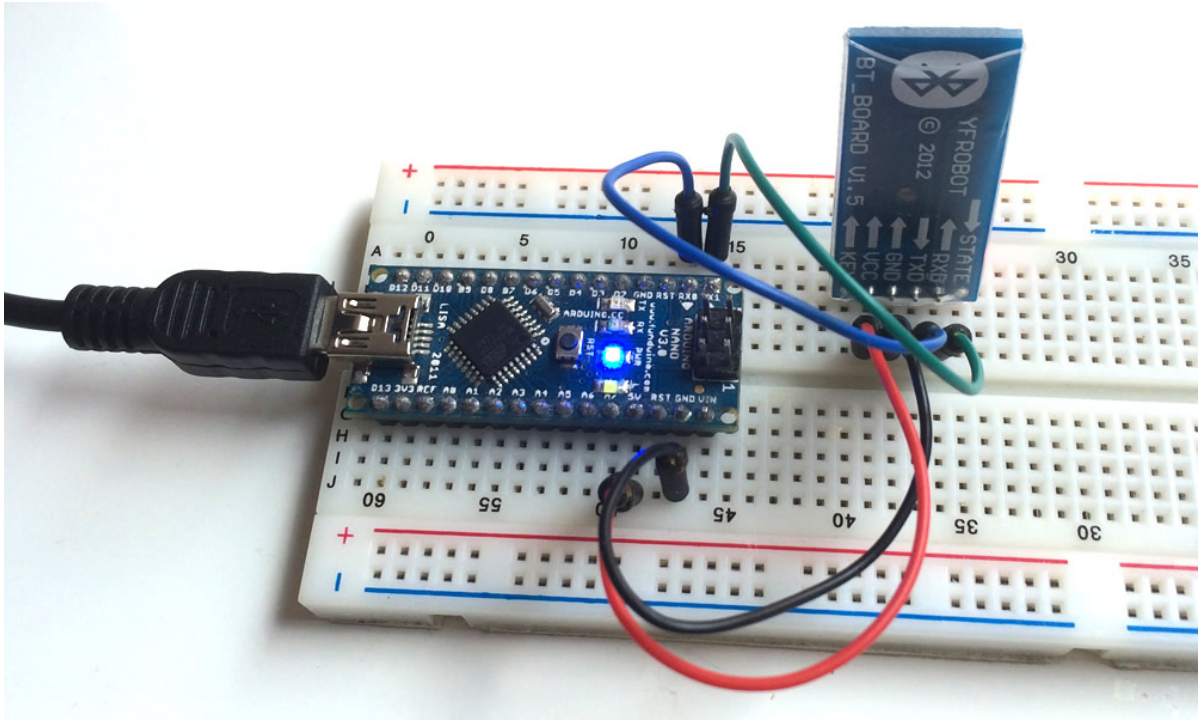
- Jelenleg ez a legolcsóbban beszerezhető fejlesztőeszköz
- Könnyen használható, ingyenes programfejlesztői környezet
- Világszerte elterjedt, rengeteg mintapélda, programkönyvtár, leírás, tankönyv található hozzá
- Van hozzá többféle szimulátor, közöttük ingyenesek is
- Nagy választékban találunk hozzá olcsó kiegészítőket
  - Szenzorok
  - Kommunikációs modulok
  - Kijelzők
  - Motorvezérlők
  - Relé modulok
  - Robot KIT-ek


IR Control

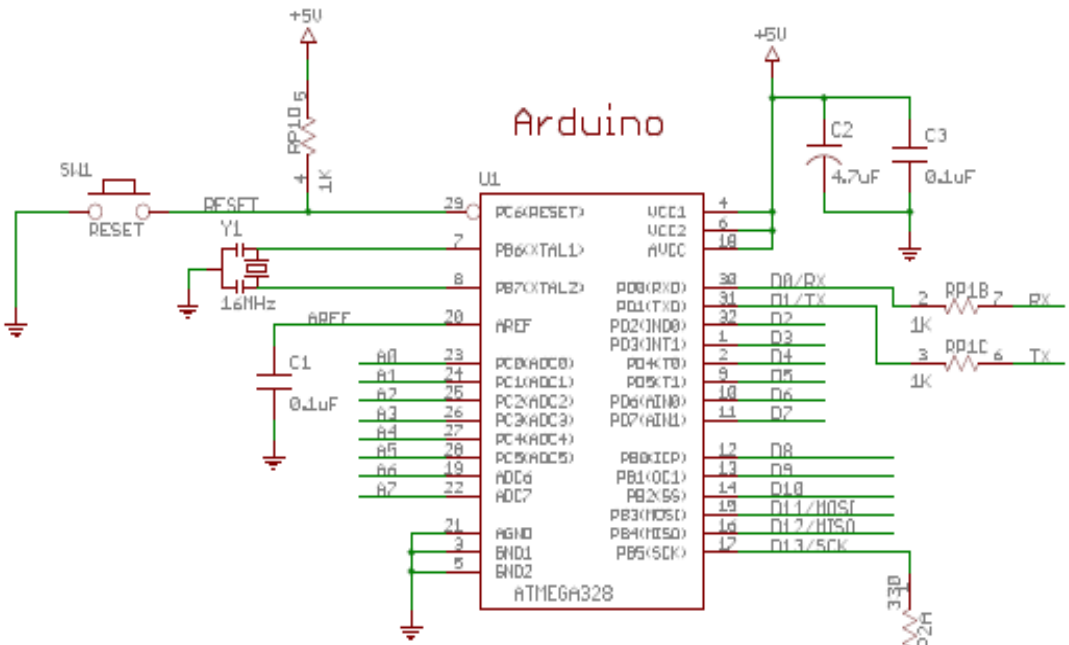
Arduino nano v3.0

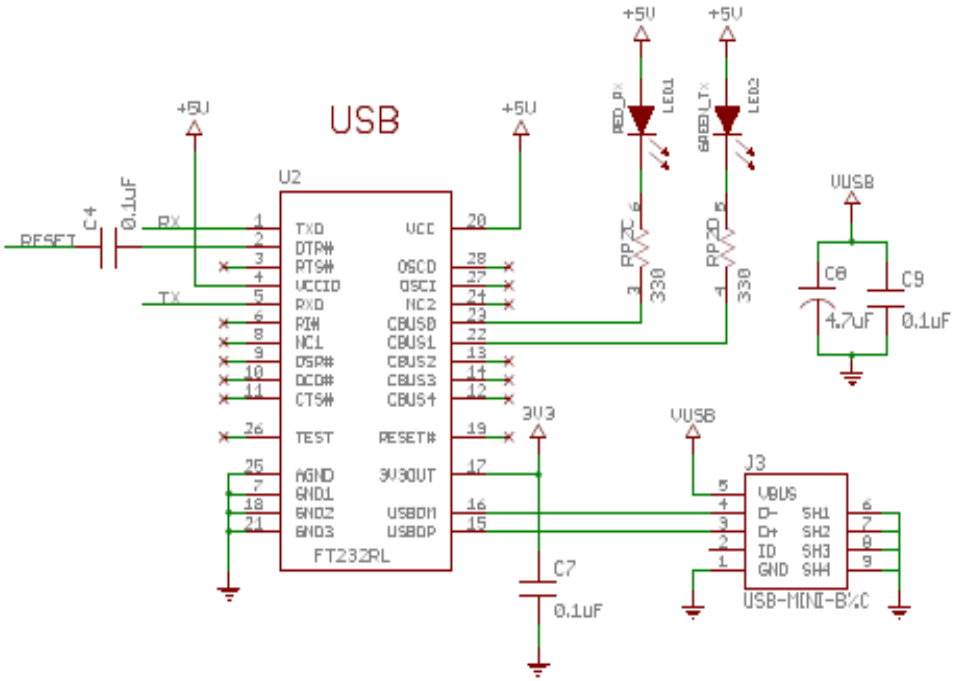


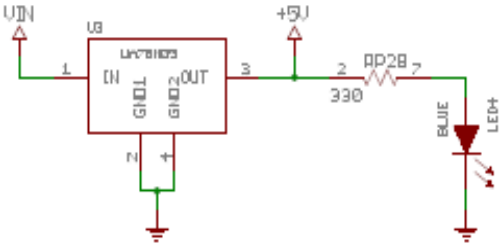
- Dugaszolós próbapanelhoz optimális
- Ára kedvező
- A gyári fedlapokhoz csak kiegészítő kártyával használható!

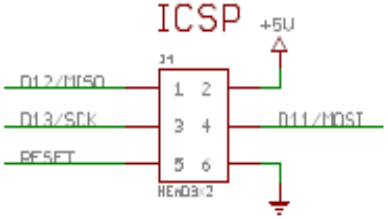


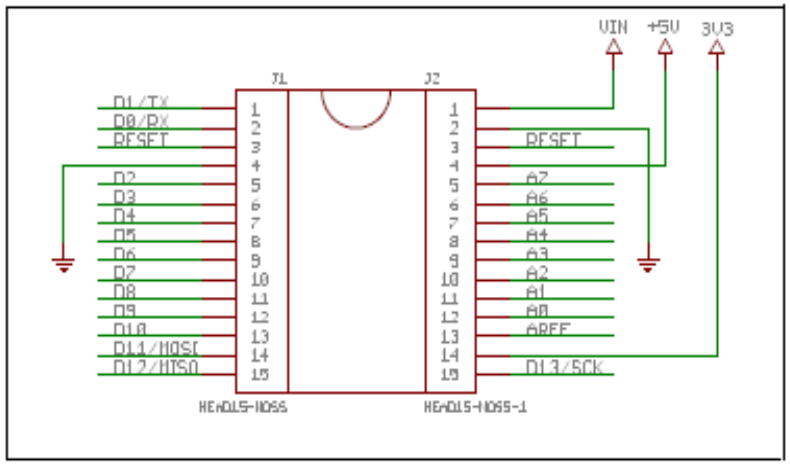
Arduino nano v3.0



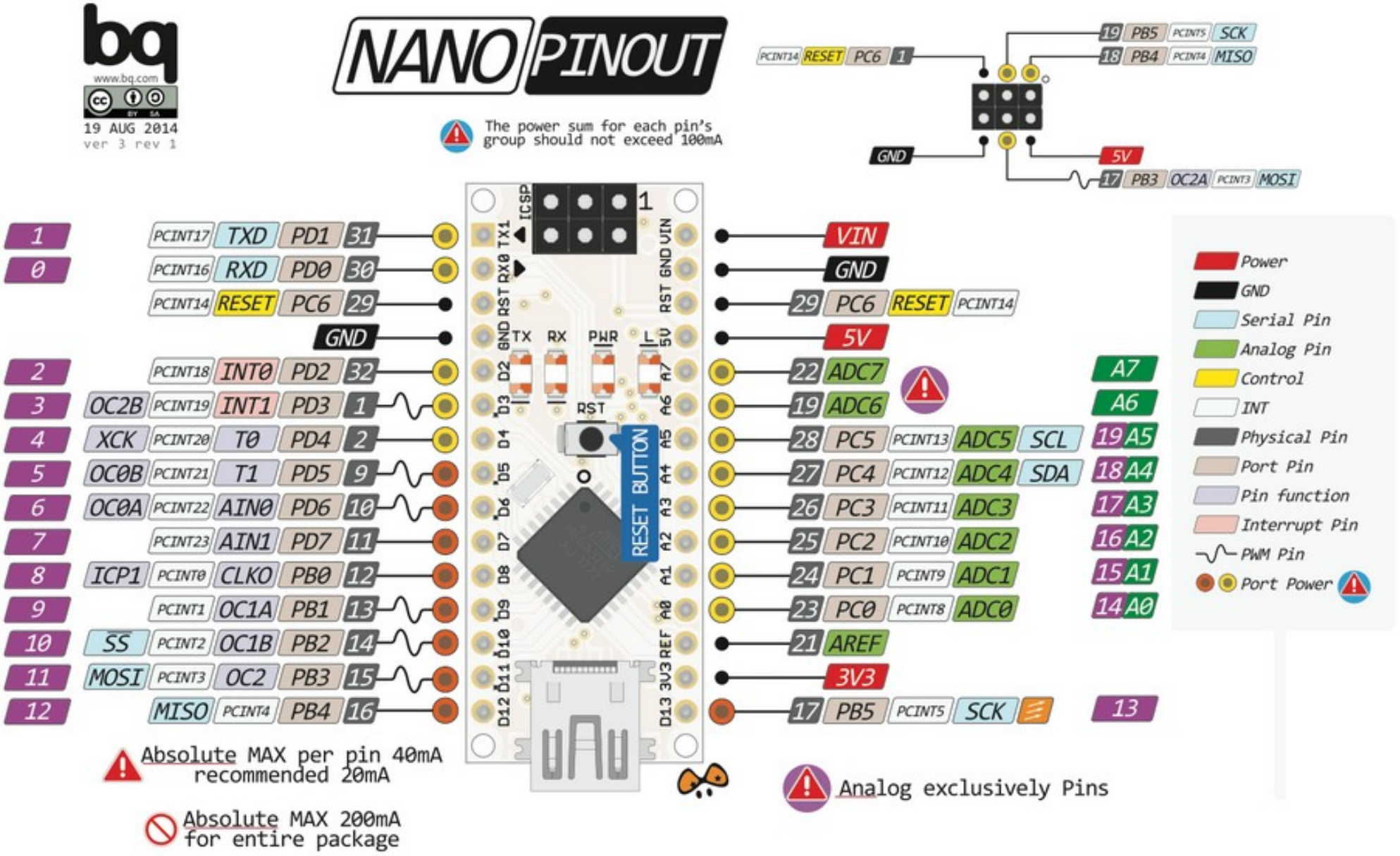








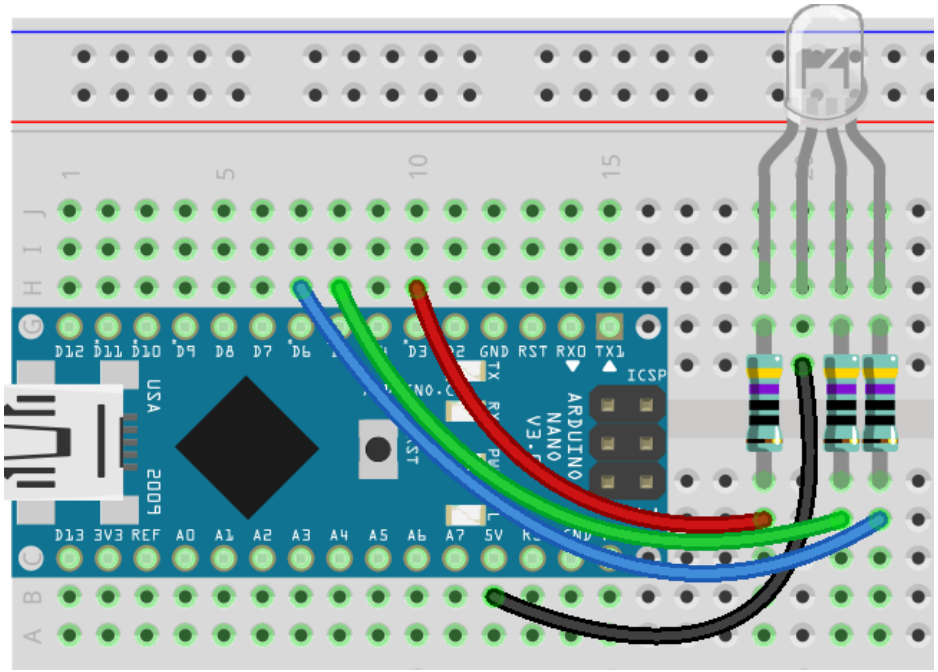
Az Arduino nano kártya kivezetései



Arduino, MiniPirate, RGB LED

- Folyatassuk az ismerkedést a MiniPirate segítségével!

```
LIST OF SUPPORTED COMMANDS
==========================
h/? - Show this help
p - Show all port values & directions
q - Show all port values & directions (quick)
< - Set a port as INPUT
> - Set a port as OUTPUT
/ - Set a port to HIGH (clock up)
\ - Set a port to LOW (clock down)
^ - Set a port LOW-HIGH-LOW (one clock)
$ - Do a pin sweep
c - Set port to clock high and low with given delay
g - Set analog (pwm) value
s - Set servo value
i - Scan i2c device addresses
# - Set i2c device active x
r # - Read i2c n bytes from active device
w # # # - Write i2c bytes to active device
x - save current config to eeprom
y - load last config from eeprom
z - set all ports to input and low
v - Show AVR VCC reading
t - Show AVR internal temperature reading
f - Show free memory
u - Show system uptime (or clock)
e - Erase EEPROM
* - Reboot
```

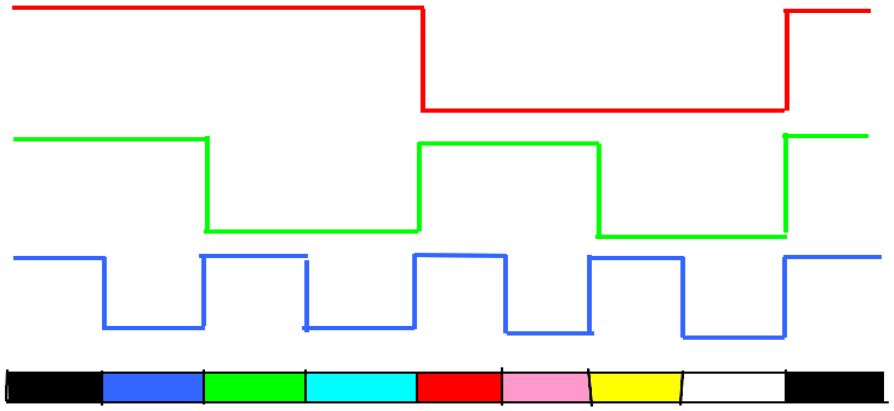


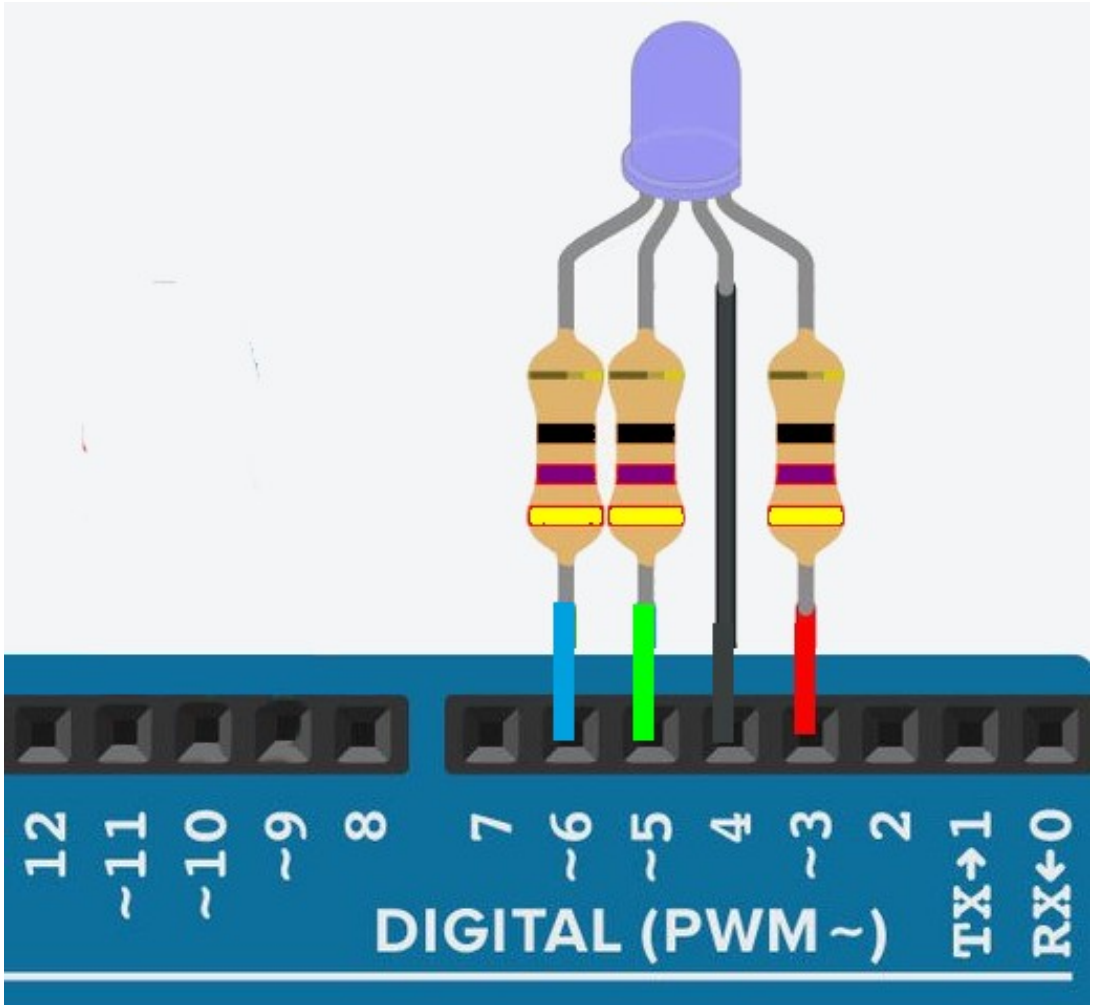
RGB LED D3 = piros, D5 = zöld,
D6 = kék katód, D4 = közös anód
Inicializálás: **>3>4>5>6**
Piros: **\3 /4 /5 /6**
Zöld: ***/3 /4 \5 /6***
Kék: ***/3 /4 /5 \6***

RGB LED villogtatás

- A bekötés ugyanaz, mint az előző oldalon
- Inicializálás: `>3>4>5>6/3/4/5/6`
- Kék villogtatás: `c6 500`
  (500 ms késleltetés)
- Leállítás: `c6 0`
- Színek léptetése:

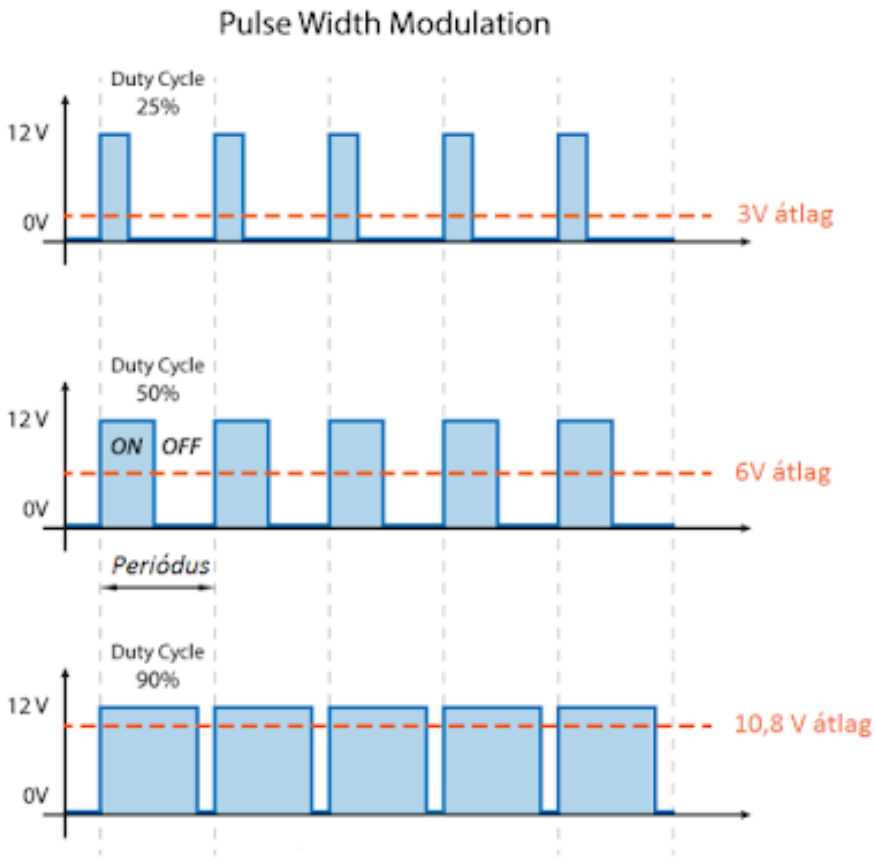
```
c6 1000
c5 2000
c3 4000
```





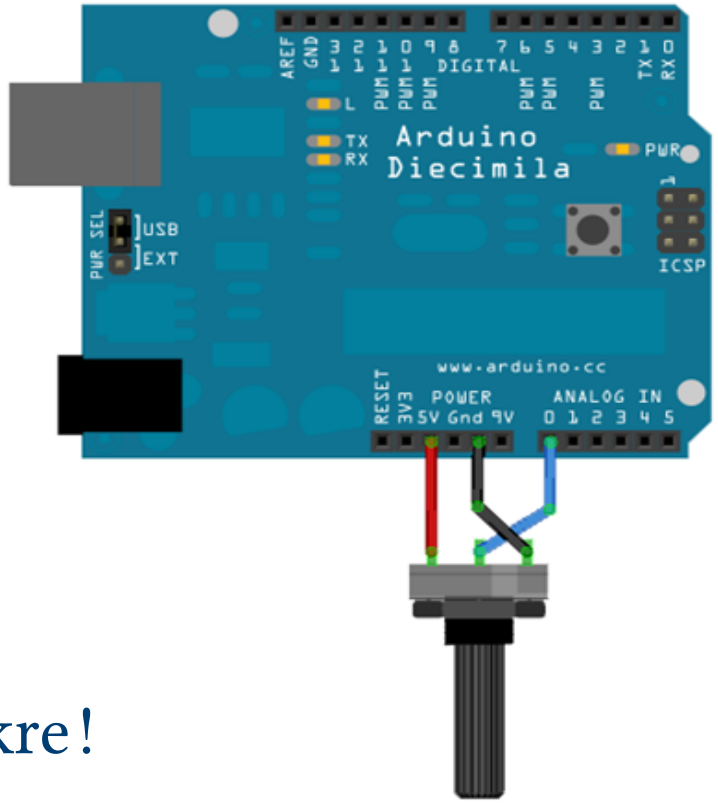
„Folytonos” színvezérlés PWM-mel

- PWM = Pulse Width Modulation (impulzus-szélesség moduláció)
- A nagyobb kitöltés nagyobb átlagteljesítményt jelent
- Esetünkben a kitöltés 0 – 255 között változtatható, s a közös anódú RGB LED-nél 0-nál maximális a fényerő (a fordított logika miatt a fehér terület számít)
- Inicializálás: `>3>4>5>6/3/4/5/6`
- Kék halványan: `g6 240`
- Zöld hozzáadása: `g5 220`
  Az eredmény: türkiz
- Piros hozzáadása: `g3 100`
  Az eredmény: rózsaszín
- Házi feladat: keverjünk színeket!



Analóg jelek vizsgálata

- Analóg (folytonos) jeleket az A0 – A7 bemeneteken vizsgálhatunk (a 28 lábú, DIP tokozású ATmega328 esetén csak A0 – A5 létezik)
- Az analóg jeleket a mikrovezérlő ADC egysége digitalizálja és 0 – 1023 közötti számmal fejezi ki (ahol a mértékegység a referencia-feszültség 1/1024-ed része)
- **Referencia lehet:**
  - A tápfeszültség (4,7 – 5.0 V)
  - Belső referencia (1,13 V)
  - Külső referencia (REF bemenetre kötve)
- A kísérlethez kössük egy 10 kΩ-os potméter csúszkáját az **A0** bemenetre!
- A potméter két szélső kivezetését pedig kössük a **GND** és a **VCC** (5V) kivezetésekre!



Analóg jelek vizsgálata

- Először határozzuk meg a tápfeszültséget a **v** parancs segítségével!

```
> v
4.71 Volts, based on a nominal internal reference
of 1.13 Volts, +/-10% per chip
```

A kacsacsőrt nem mi írjuk be!!!

- Állítsuk be a potmétert valahová, és adjuk ki a **p** parancsot!

```
Value on pin A0   INPUT: HIGH / 0623 / 2.87V
```

Számolt érték

ADC konverzió eredménye

- A kiszámolás menete:

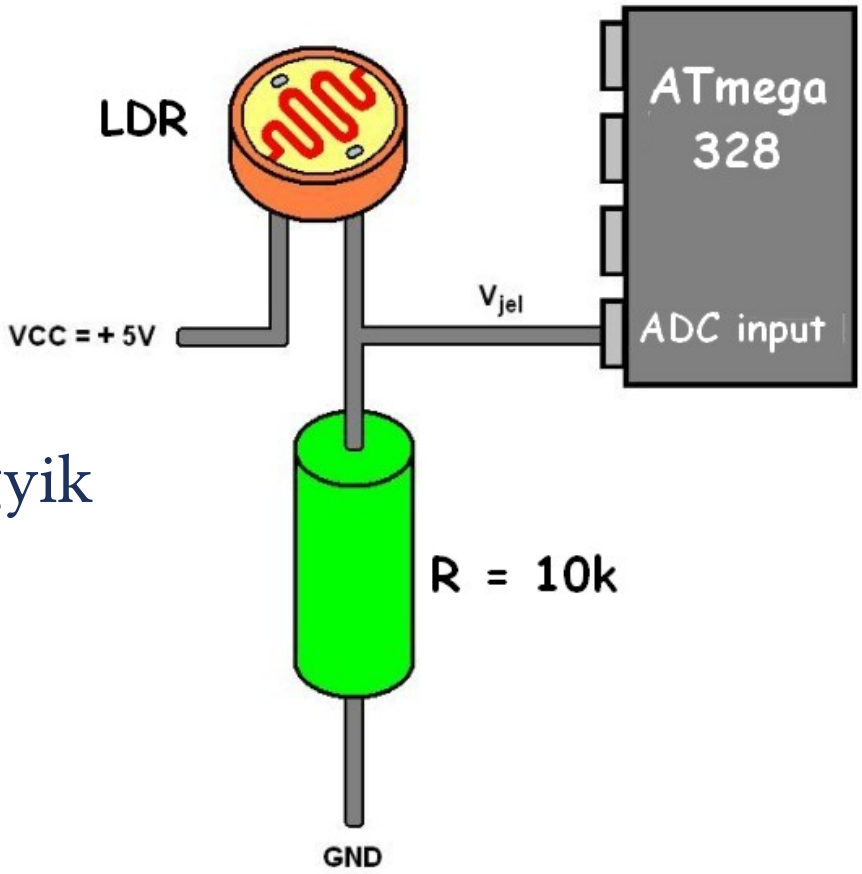
*feszültség = ADC érték * Vref / 1024*

*feszültség = 623 * 4,71 V / 1024 = 2,87 V*

- Ha több mérést végzünk, az eredmény kissé ingadozhat a zaj, vagy instabilitások (pl. melegedés, kontaktus bizonytalansága) hatására. **Több mérés átlaga** általában megbízhatóbb eredményt ad!

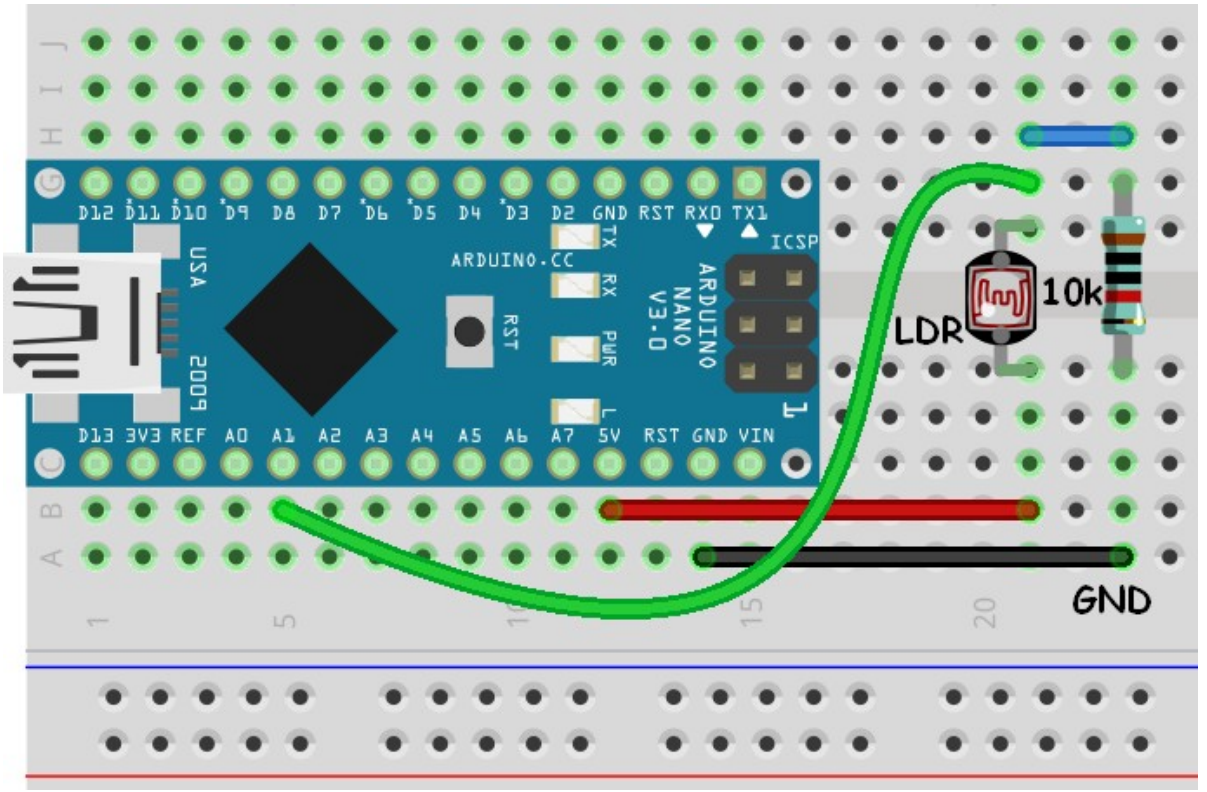
Fénymérés LDR ellenállással

- A fényérzékeny ellenállás (LDR = Light Dependent Resistor) általában kadmiumszulfid (CdS) anyagú
- Erős fényben az LDR ellenállása lecsökken (tipikusn 1-2 kΩ), a feszültségosztón mérhető feszültség magas (4,5 V közeli)
- Sötétben az LDR ellenállása megnő (tipikusan 100 kΩ nagyságrendű), a feszültségosztón mérhető feszültség alacsony (0-hoz közelít)
- A leosztott feszültséget az Arduino egyik analóg bemenetán mérhetjük meg, ugyanúgy, mint az előző kísérletnél a potméterrel leosztott feszültséget



Fénymérés LDR ellenállással

- A kapcsolást az ábra szerint állítottuk össze
- A **p** parancs kiadása után az **A1** bemenetre kapott értékre figyeljünk!

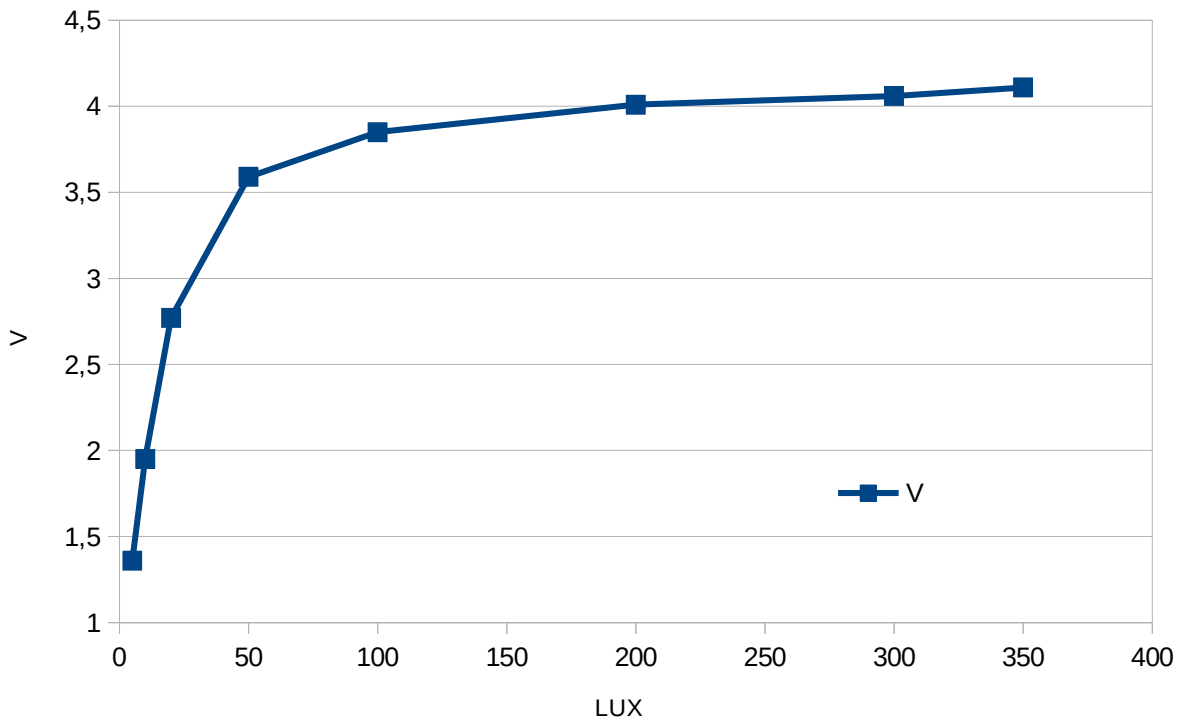



```
Value on pin D11  INPUT: LOW  PWM
Value on pin D12  INPUT: HIGH
Value on pin D13 OUTPUT: LOW
Value on pin A0   INPUT: HIGH / 0623 / 2.87V
Value on pin A1   INPUT: HIGH / 0849 / 3.91V
Value on pin A2   INPUT: HIGH / 0646 / 2.97V
Value on pin A3   INPUT: HIGH / 0542 / 2.49V
Value on pin A4   INPUT: HIGH / 1014 / 4.67V
Value on pin A5   INPUT: HIGH / 1008 / 4.64V
Value on pin A6   INPUT: LOW  / 0731 / 3.36V
Value on pin A7   INPUT: LOW  / 0600 / 2.76V
```

Mérési eredmények

- Mobiltelefonnal mérjük a megvilágítást (**Light Meter** alkalmazás)
- Az Arduino **A1** bemenetén mérjük a feszültséget
- A fényerősséget a redőny le- fel húzogatásával szabályozzuk
- Az eredményt táblázatosan és grafikon formájában ábrázoljuk

Megvilágítás [LUX]	Feszültség [V]
5	1,36
10	1,95
20	2,77
50	3,59
100	3,85
200	4,01
300	4,06
350	4,11



Ellenállás színkódok