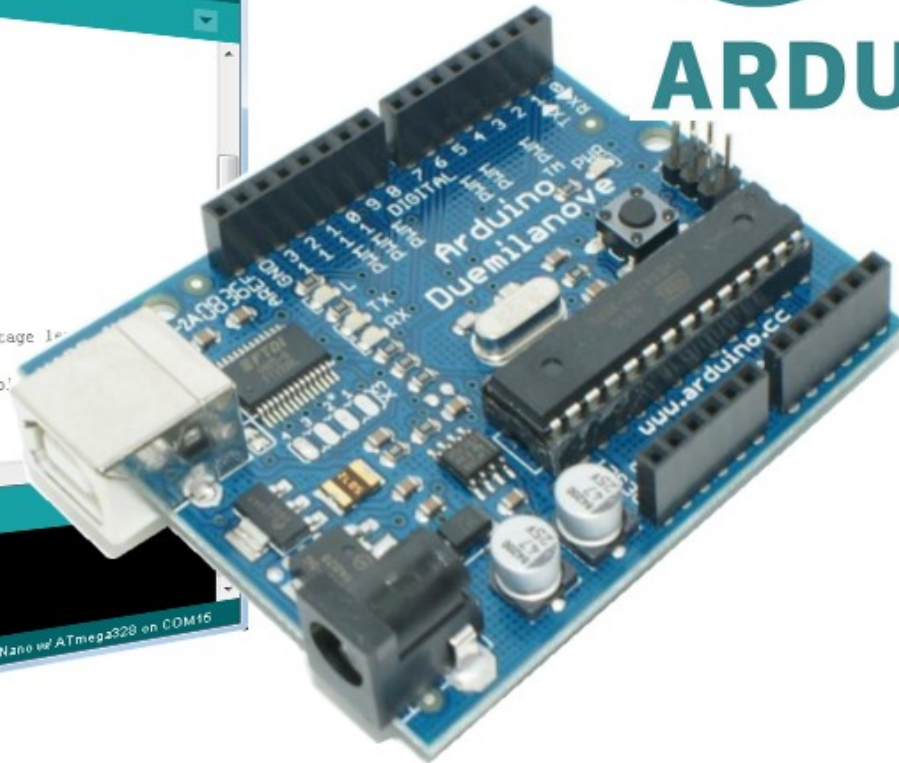
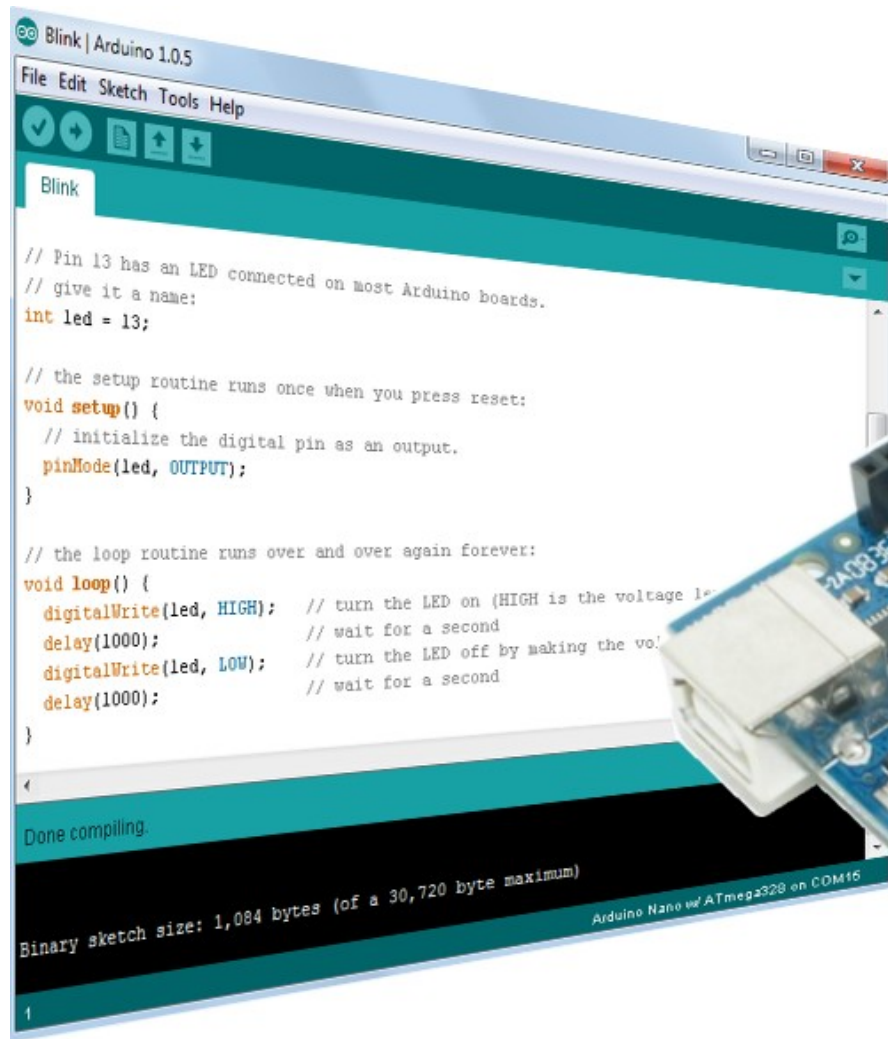


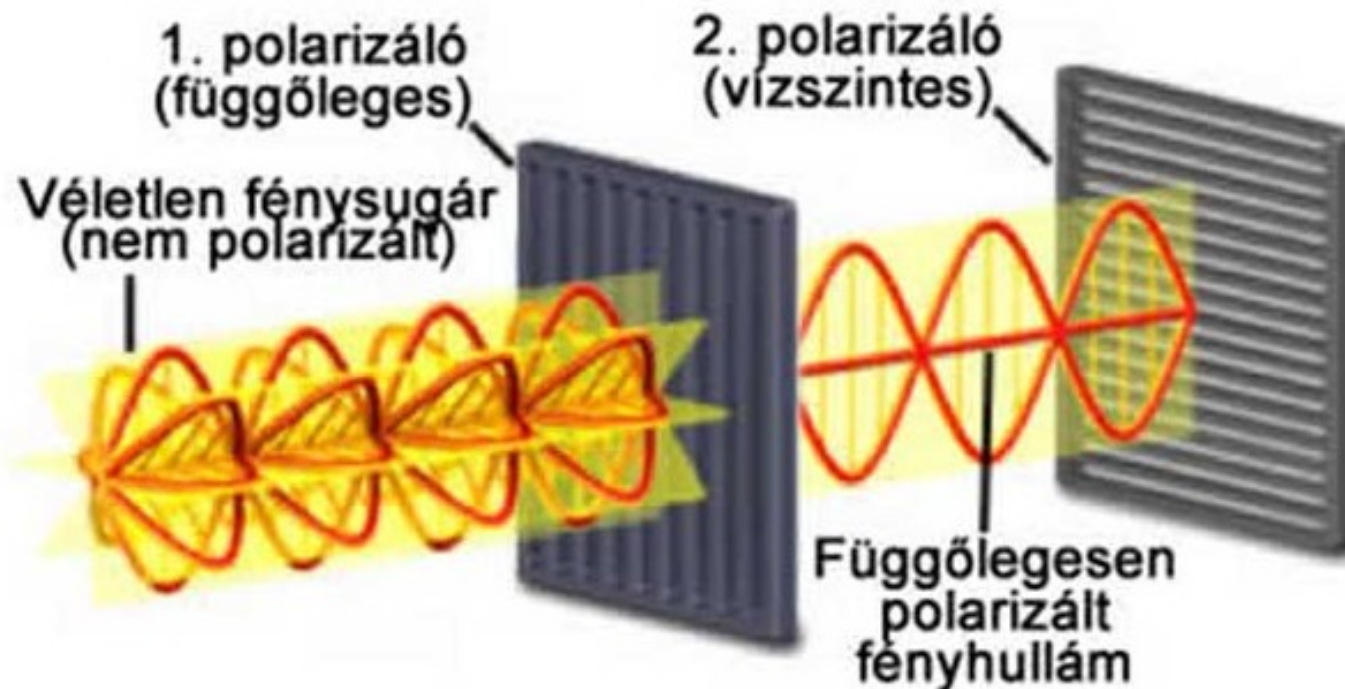
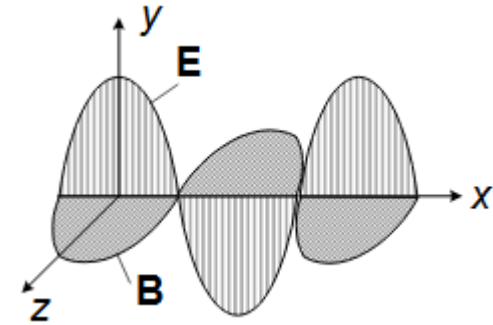
Bevezetés az elektronikába



17. Arduino programozás LCD kijelzők alkalmazása

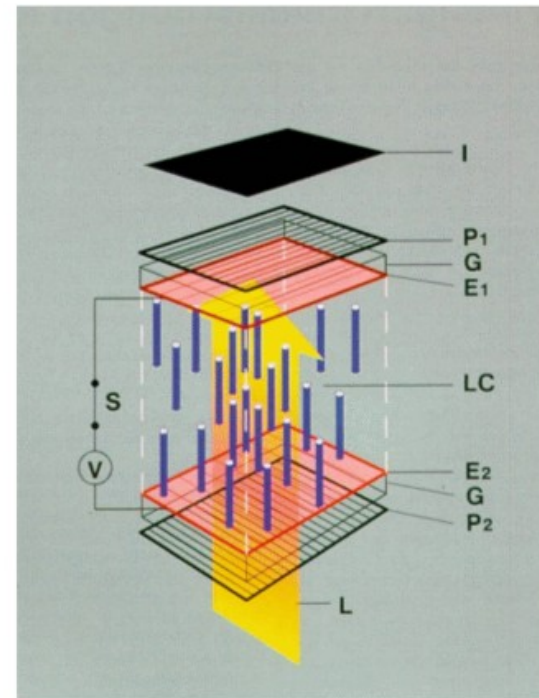
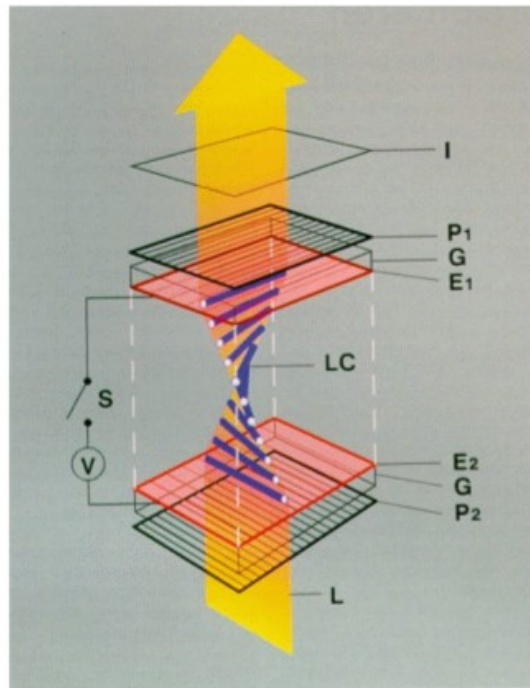
A fény polarizációja

- A fény elektromágneses sugárzás, benne az elektromos és a mágneses tér folytonosan egymásba alakul
- A fény transzverzális hullám, az elektromos és mágneses térerősség a haladási irányra merőleges
- A polarizált fény csak „egy síkban rezeg”



LCD = Liquid Crystal Display

- **Folyadékkristály (Liquid Crystal):** olyan (szerves) anyag, mely sűrű folyadéknak tekinthető, ugyanakkor molekulái – a kristályokhoz hasonlóan – képesek struktúrákba rendeződni.

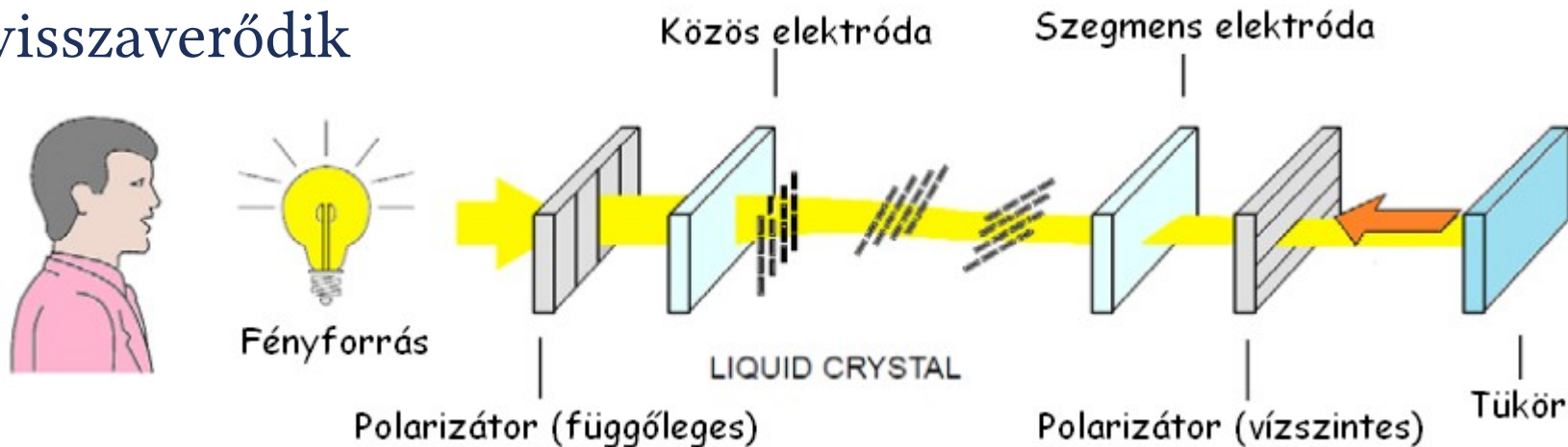


- Az alapesetben csavart struktúra segít átjuttatni a polarizált fényt az elforgatott polárszűrőkön. Az elektromos tér átrendezi a molekulákat, már nem segítik a fény átjutását.

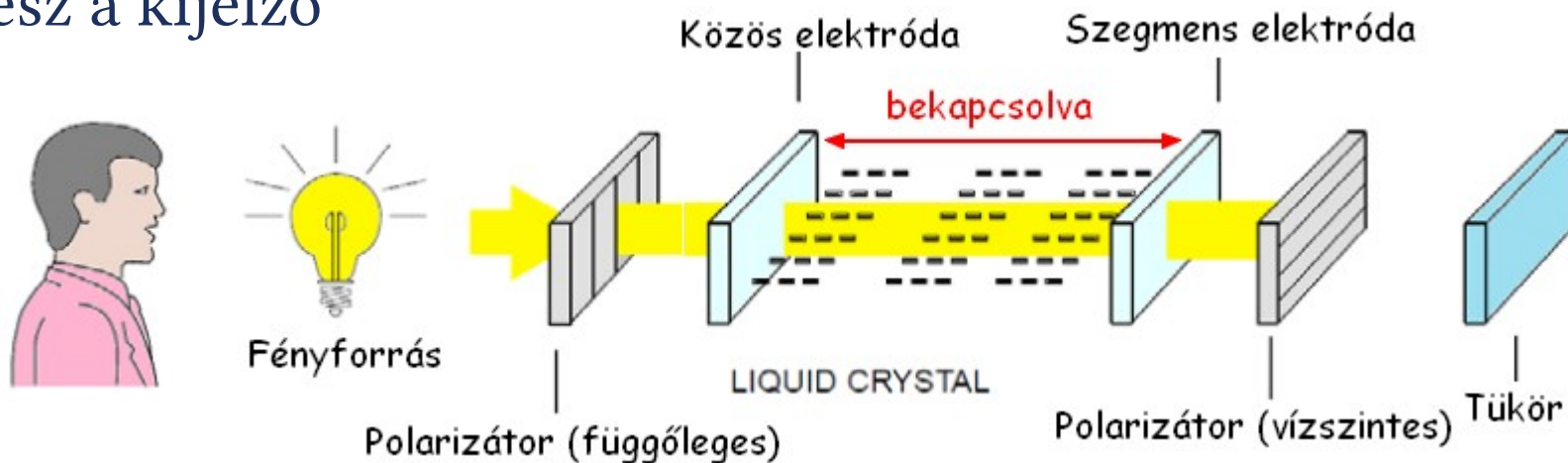
Forrás: https://en.wikipedia.org/wiki/Twisted_nematic_field_effect

Reflektív (fényvisszaverő) kijelző

- A belépő polarizált fénynyaláb a molekulák fokozatos elfordulását követve elfordul, majd a hátul elhelyezett tükörről visszaverődik

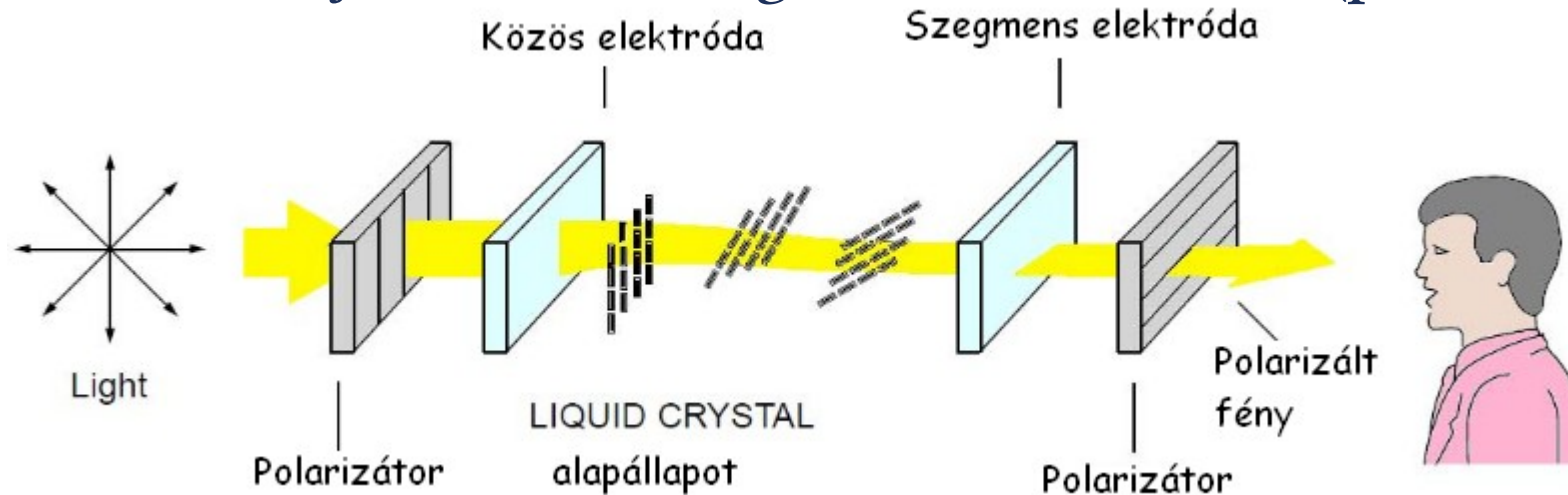


- A „bekapcsolt” helyeken a visszavert fény hiánya miatt sötétebb lesz a kijelző

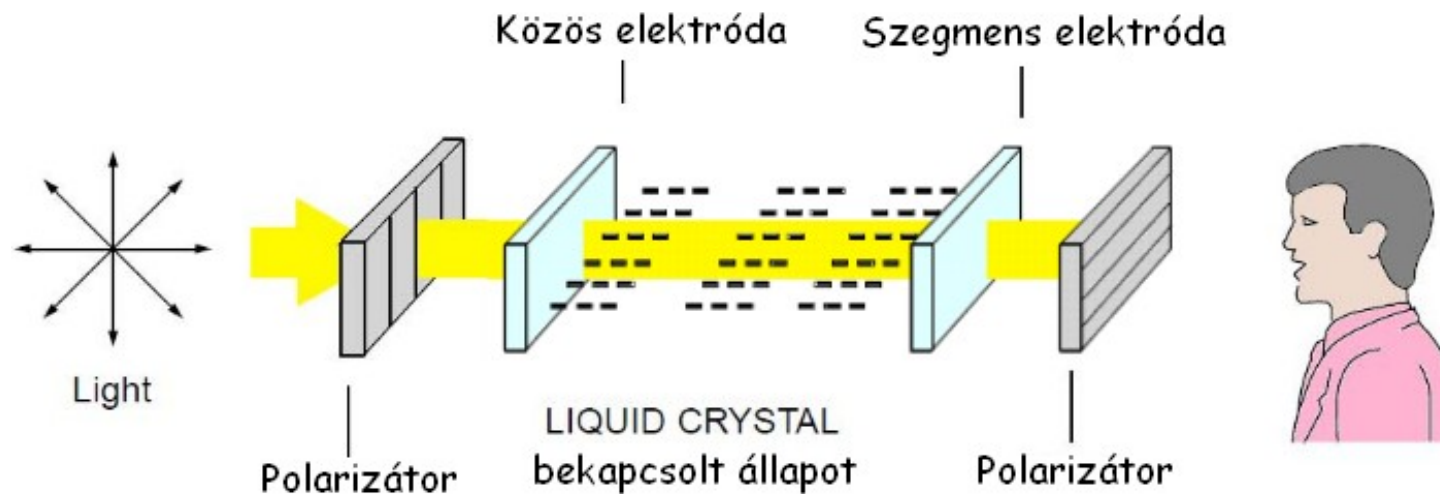


Transmissive (áteresztő) kijelző

- Az áteresztő kijelző háttérvilágítással rendelkezik (pl. LED panel)

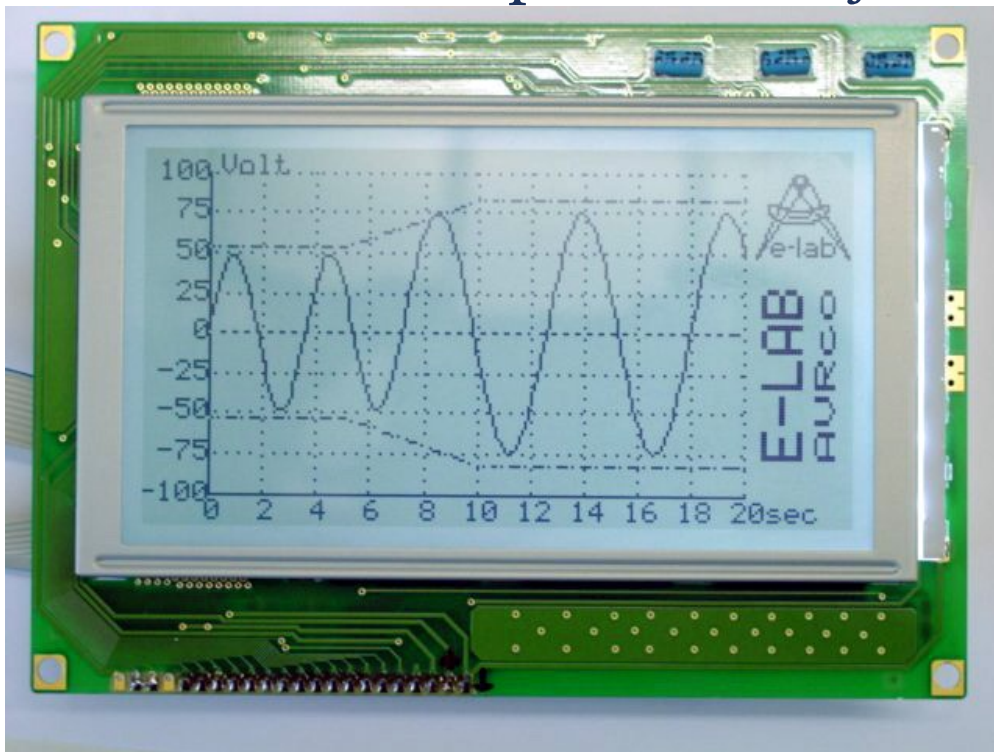


- Az elektromos tér hatására a folyadékkristály molekulái átrendeződnek, a fényáteresztés lecsökken.

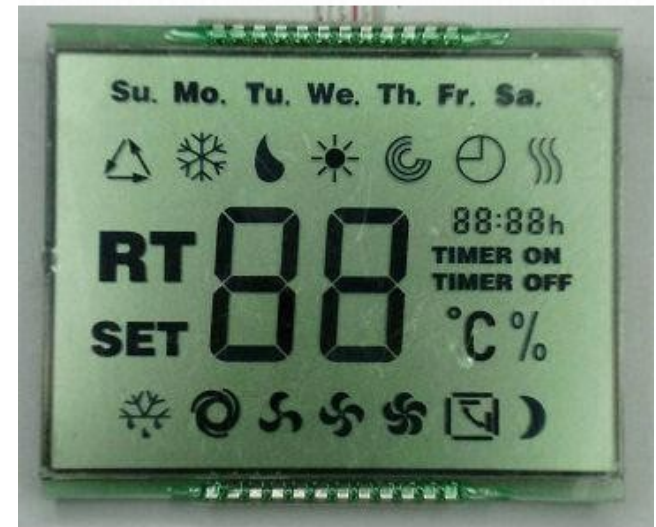


LCD kijelző típusok

- Az LCD kijelzők több lehetőséget biztosítanak, mint a 7-segmens kijelzők, ezért a technika fejlődésével jobban elterjedtek
- Lehetőségeik, kivitelük, fogyasztásuk szerint több csoportba soroljuk ezeket



Grafikus pontmátrix kijelző



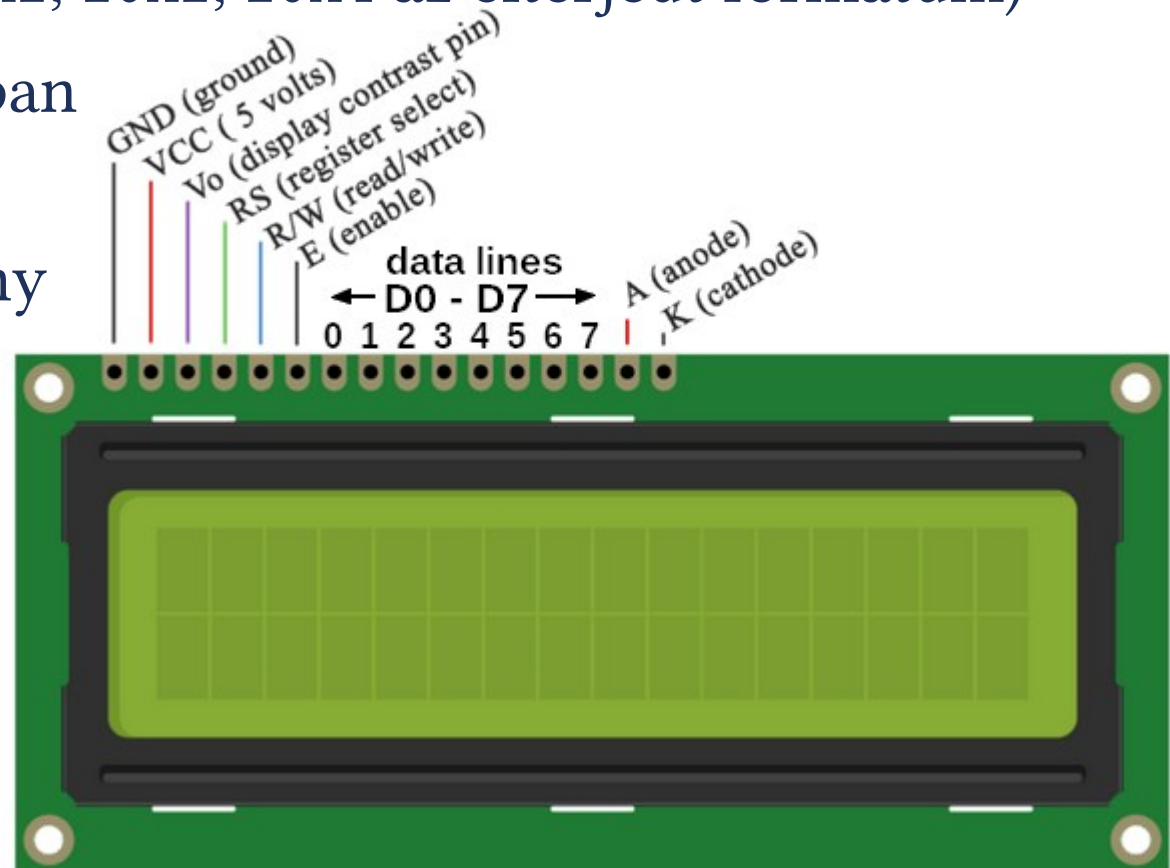
Szegmens kijelző



Alfanumerikus kijelző (4x20)

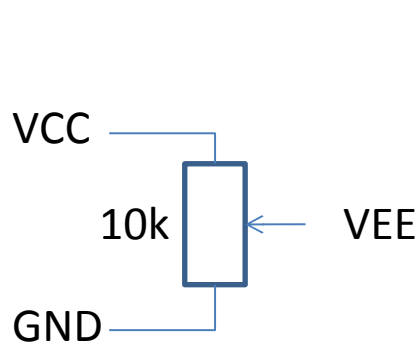
HD44780 kompatibilis kijelzők

- A HD44780 vagy kompatibilis vezérlővel ellátott alfanumerikus (csak rögzített karakterképet jeleníthetünk meg, pl. betű, szám, írásjel) kijelzők elterjedtségük miatt ipari szabványnak tekinthetők
- Tipikus a 16 kivezetéses, kétsoros, 2x16 karakteres modul, LED háttérvilágítással (8x2, 16x2, 20x2, 20x4 az elterjedt formátum)
- 8 és 4 adatvezetékes módban is használható
- R/W lehet mindig alacsony
- R/S adat/parancs váltó
0: adat 1: parancs küldés
- E – beíró jel (felfutó él) előtte a bemeneteket be kell állítani

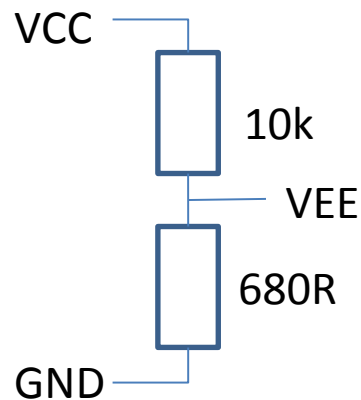


Kontraszt beállítása (variációk)

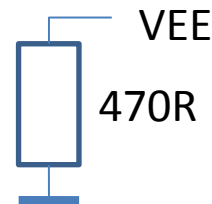
- Adatlap: 25 °C-on VEE számára VCC-4.5 V körüli érték az optimális
- A beállítást többféle módon is megoldhatjuk, legáltalánosabb módon egy 10 kΩ-os potméter segítségével. Sok esetben a GND-re kötés is működőképes megoldás



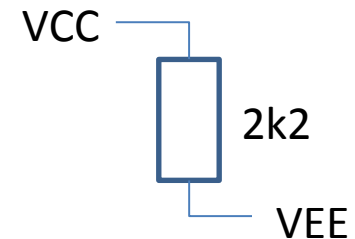
1. Az általánosan javasolt megoldás



2. Fixen beállított kontraszt



3. Egyszerűsített kontraszt beállítás (néhány 5 V-os kijelzőnél bevált Ez, vagy a direkt földre kötés is...)



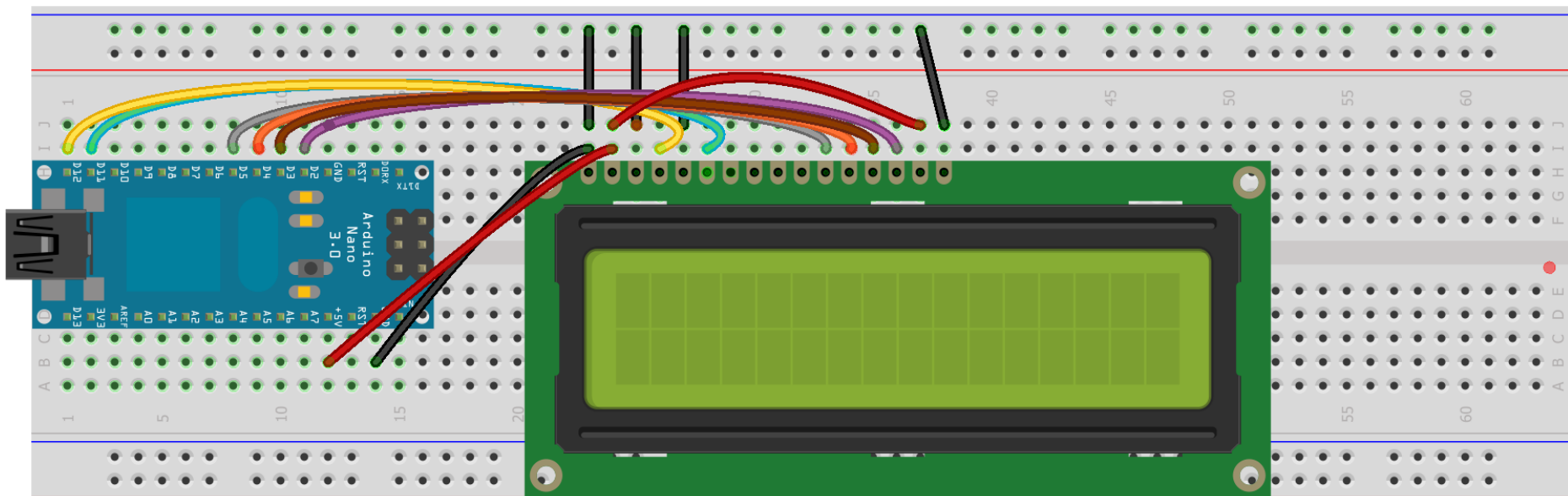
4. Egyszerűsített kontraszt beállítás 3,3V-os kijelzőhöz

Bekötési vázlat

- Az LCD kijelzőt az Arduino beépített függvényei 4-bites módban használják (minden bájtot két részletben küldünk ki), ezért a **data0-data3** vonalakat nem kell bekötni!

1. GND – GND
2. VCC – +5V
3. VEE – GND
4. RS – D12
5. RW – GND
6. EN – D11

11. data4 – D5
12. data5 – D4
13. data6 – D3
14. data7 – D2
15. LED+ – +5V
16. LED- – GND



Made with  Fritzing.org

LiquidCrystal programkönyvtár

- Az LCD kijelzőt Arduino környezetben a **LiquidCrystal** beépített programkönyvtár segítségével kezelhetjük legegyszerűbben
- **A legfontosabb metódusok:**
 - ❖ **lcd.begin**(oszlop,sor) – a képernyő inicializálása és konfigurálása
 - ❖ **lcd.setCursor**(oszlop, sor) – pozíció beállítása a megadott helyre
 - ❖ **lcd.write**(karakterkód) – egy karakter kiírása
 - ❖ **lcd.print**("szöveg") – szöveg kiírása
 - ❖ **lcd.print**(kifejezés) – számérték kiírása
- Bővebb információ az Arduino IDE Help/Referencia menüpontjában található, illetve az alábbi weblapon:
arduino.cc/en/Reference/LiquidCrystal

HelloWorld.ino

- Írjuk ki a *hello, world!* (helló világ) szöveget az első sorba, a második sorba pedig a bekapcsolás óta eltelt időt (másodpercben)!

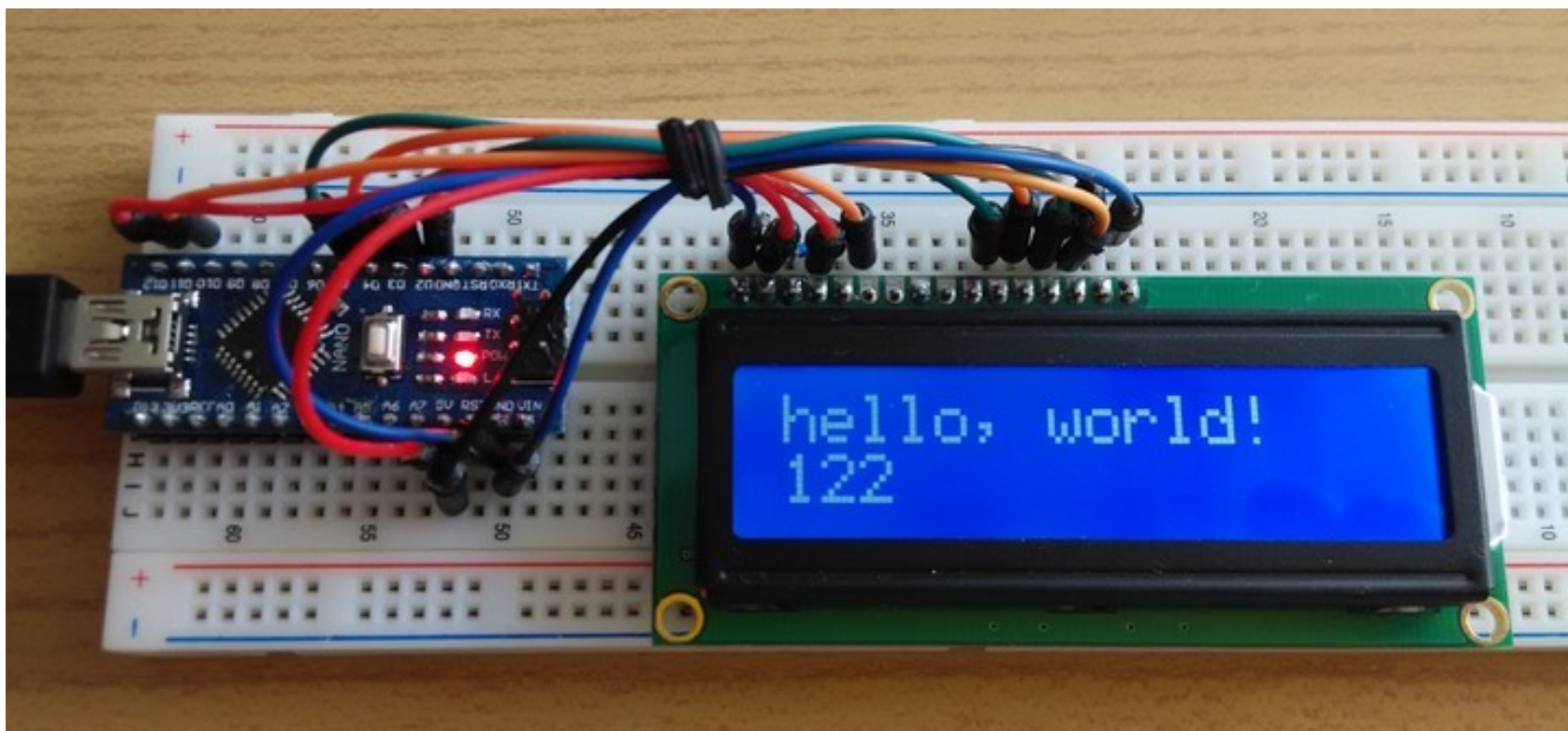
```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // Oszlopok és sorok számának megadása
  lcd.begin(16, 2);
  // Üzenet kiírása az első sorba
  lcd.print("hello, world!");
}

void loop() {
  // Állítsuk be a kurzort: 0. oszlop, 1. sor
  // (az 1. valójában a második, mert 0-nál kezdődik a számozás)
  lcd.setCursor(0, 1);
  // Az (újra)indítás óta eltelt idő kiírása másodpercekben
  lcd.print(millis()/1000);
}
```

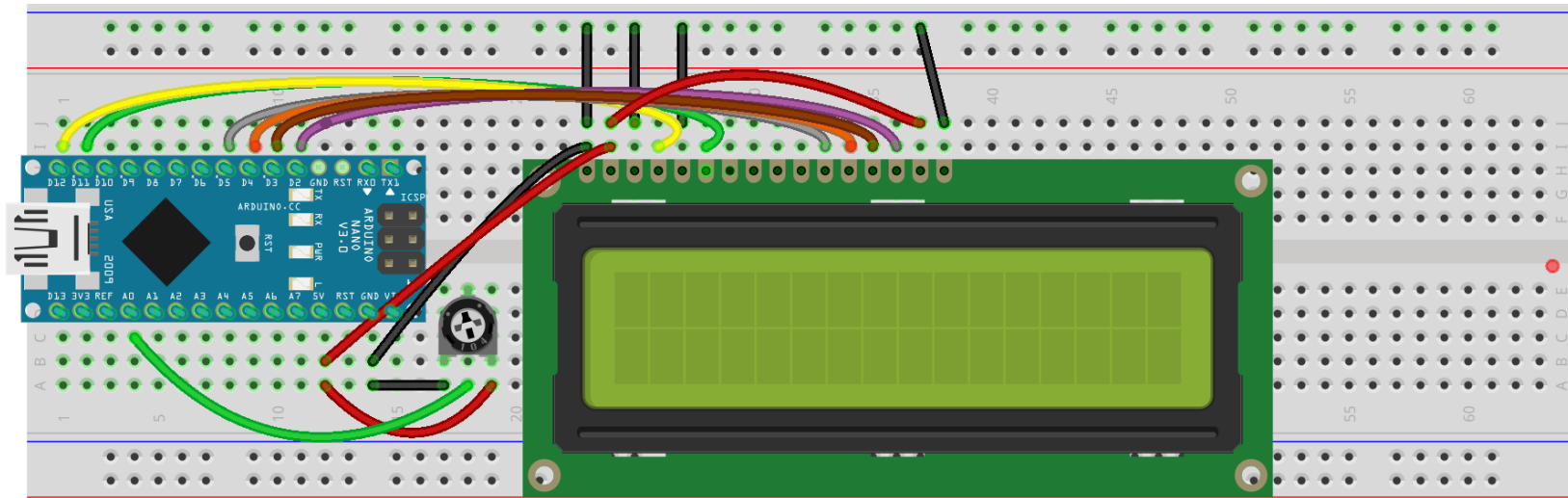
HelloWorld.ino

- Helyes bekötés és a program letöltése után a programfutás eredményeként ezt látjuk



Feszültségmérő

- Építsünk egy egyszerű feszültségmérőt! Az A0 analóg bemenetre kapcsolt 0 – 5 V feszültséget fogjuk megmérni, és a kijelzőn megjeleníteni
- A márciusi Lab04 mintaprogramoknál találkoztunk már ilyen feladattal, csak most a soros port helyett az LCD kijelzőre írunk
- Az alábbi ábrán egy potméterrel leosztott feszültséget mérünk



fritzing

LCD_16x2_voltmeter.ino

```
#include <LiquidCrystal.h>           //Becsoljuk az LCD programkönyvtárt
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
                RS  E   D4 D5 D6 D7

void setup() {
  lcd.begin(16, 2);                  // Oszlopok és sorok száma
  lcd.print("Analog Voltmeter");    //Kiírunk egy szöveget az első sorba
}

void loop() {
  int sensorValue = analogRead(A0); //Mérés az A0 bemeneten
  // Átszámítjuk a 0 - 1023 közötti számot voltra (0 - 5V):
  // 4.81 itt a tényleges USB feszültség
  float voltage = sensorValue * (4.81 / 1024.0);
  lcd.setCursor(0, 1);              //Kurzor a második sor elejére
  lcd.print(voltage,3);              //Kiíratjuk az eredményt 3 tizedesre
  lcd.print(" V");
  delay(100);
}
```

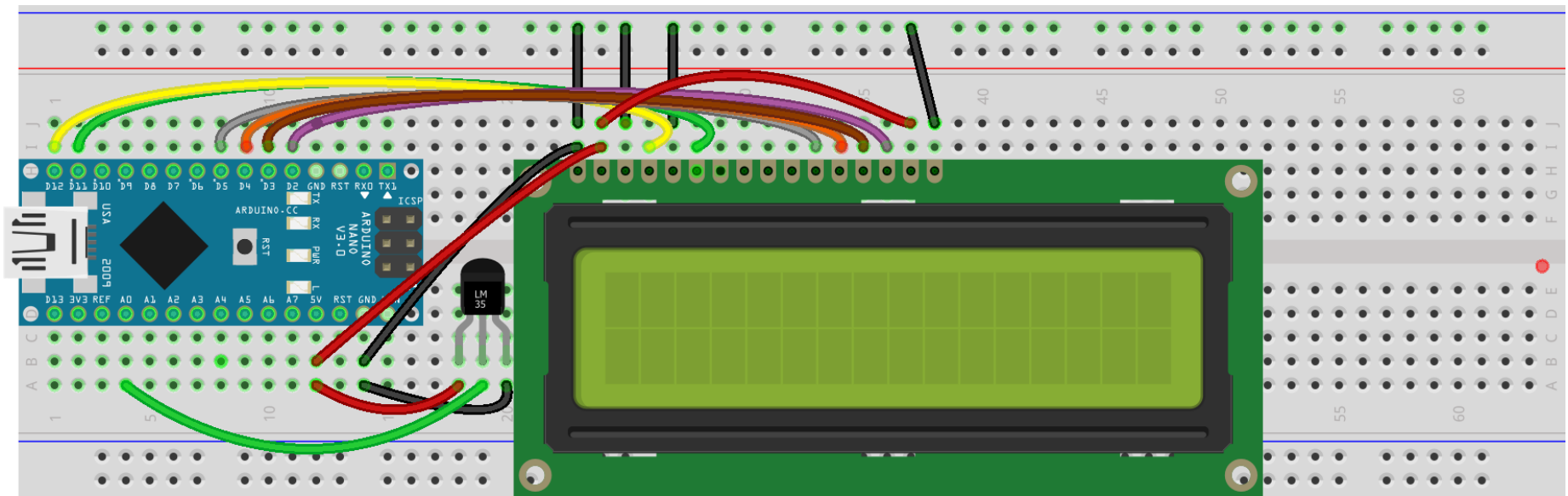
LCD_16x2_voltmeter.ino

- Helyes bekötés és a program letöltése után a programfutás eredményeként ezt látjuk
- Amint látjuk, a kontrasztbeállítás itt nem volt optimális, GND helyett kb. 0,2 V-ra kellett volna emelni a VEE feszültséget



Hőmérés analóg szenzorral

- Az A0 bemenetre egy analóg hőmérő jelét (pl. MCP9700) is vezethetjük, ekkor hőmérsékletet tudunk mérni
- Kiírathatjuk a mért feszültséget, de a feszültségből kiszámolt hőmérsékletet is
- **MCP9700** esetén az érzékenység $10\text{ mV}/^{\circ}\text{C}$, a nullapont pedig 500 mV ($0\text{ }^{\circ}\text{C}$ -nál)



LCD_16x2_thermometer.ino

```
#include <LiquidCrystal.h>           //Becsoljuk a programkönyvtárat

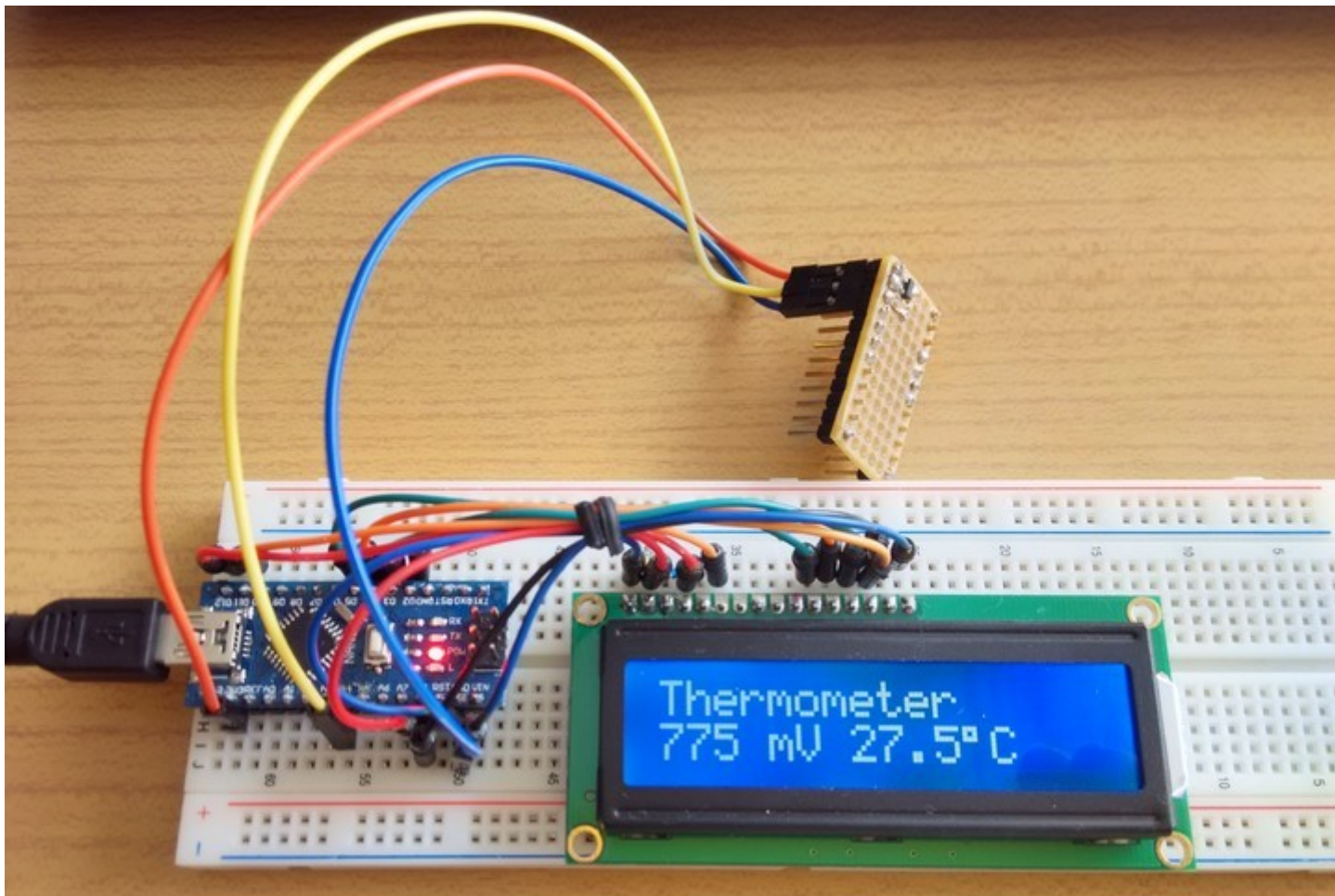
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
                        RS   E   D4  D5  D6  D7

void setup() {
    analogReference(INTERNAL);        //Az 1,1 V-os belső referencia
    lcd.begin(16, 2);                 //Beállítjuk a sorok és oszlopok számát)
    lcd.print("Thermometer");        //Kiírunk egy szöveget az első sorba
}

void loop() {
    float voltage = analogRead(A0)*1100/1024.0;
    float tempC = (voltage-500)/10; //Átszámítás fokokra
    lcd.setCursor(0, 1);              //Kurzor a második sor elejére
    lcd.print(voltage,0);             //Kiíratjuk az eredményt
    lcd.print(" mV ");
    lcd.print(tempC,1);
    lcd.write(0xDF);                  //Fok jel - bizonyos karaktergenerátornál
    lcd.print("C ");
    delay(1000);
}
```

LCD_16x2_thermometer.ino

- Helyes bekötés és a program letöltése után a programfutás eredményeként ezt látjuk
- Itt egy próbanyákra forrasztott SMD kivitelű MCP9700 szerepel

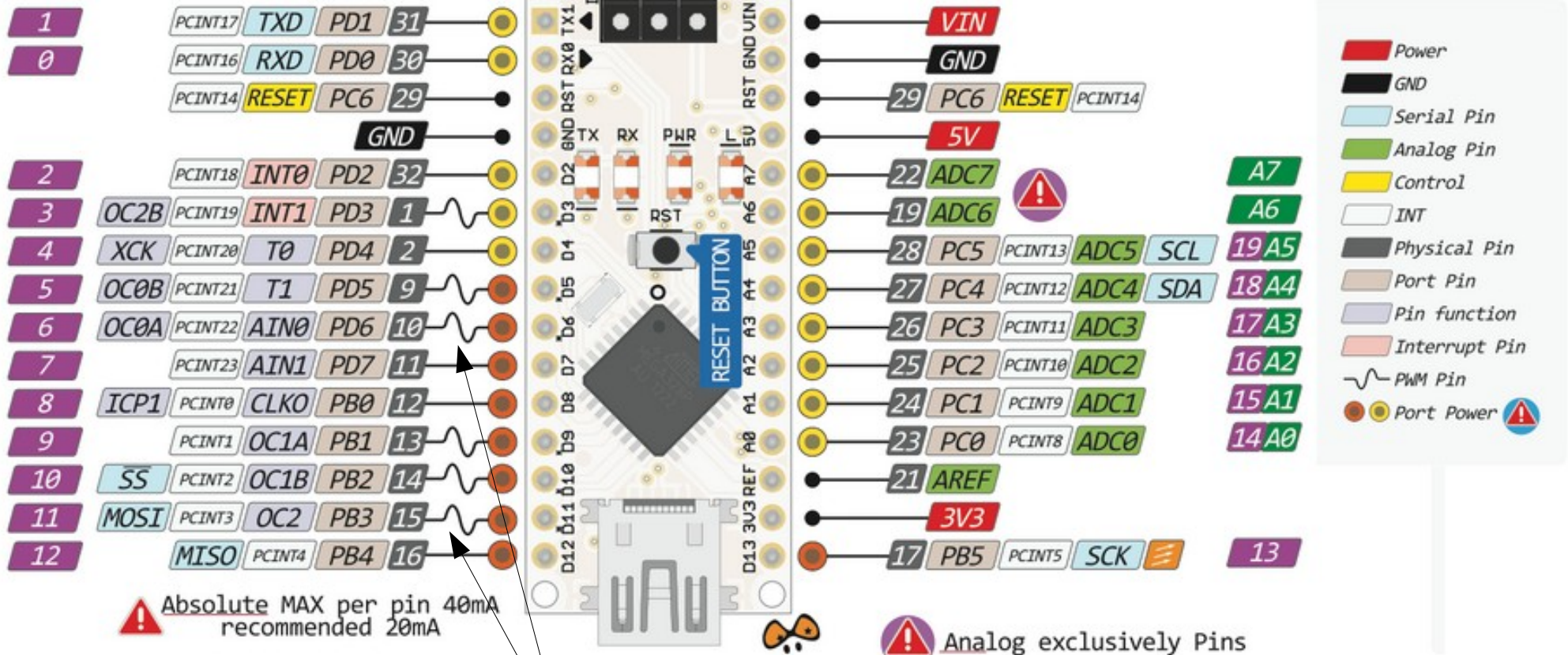
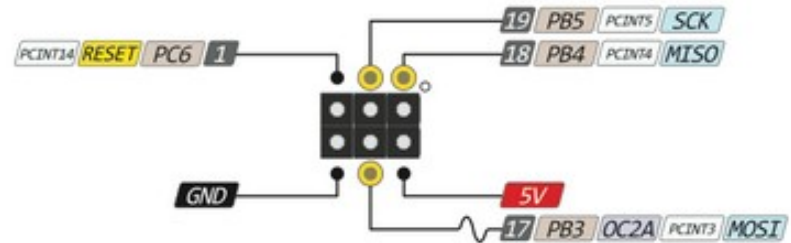


Az Arduino nano kártya kivezetései



NANO PINOUT

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



PWM kimenetek