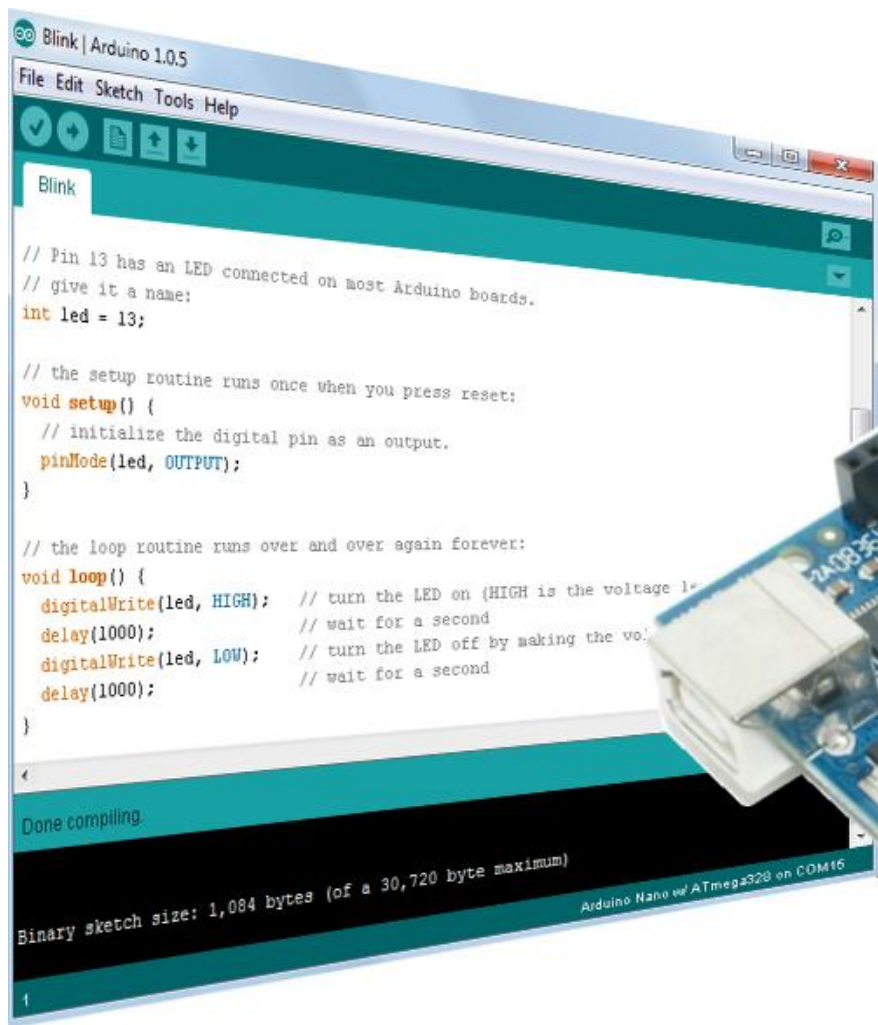




Bevezetés a mikrovezérlők programozásába: LCD kijelzők

Megjegyzés: A mai előadásban ismertetett kijelzők **MSP430 Launchpad** kártyával történő vezérléséhez a 2013/14-es évad 11. és 15. foglalkozásához tartozó tananyagban ([talk11](#) és [talk15](#) előadásvázlatok, valamint a [Lab11.zip](#) és [Lab15.zip](#) mintaprogramok) található útmutatás!

Lab 16 projektek



HelloWorld – egyszerű kiírások alfanumerikus LCD 16x2 kijelzőn



LCD_16x2_thermometer – Analóg hőmérő alfanumerikus LCD kijelzővel



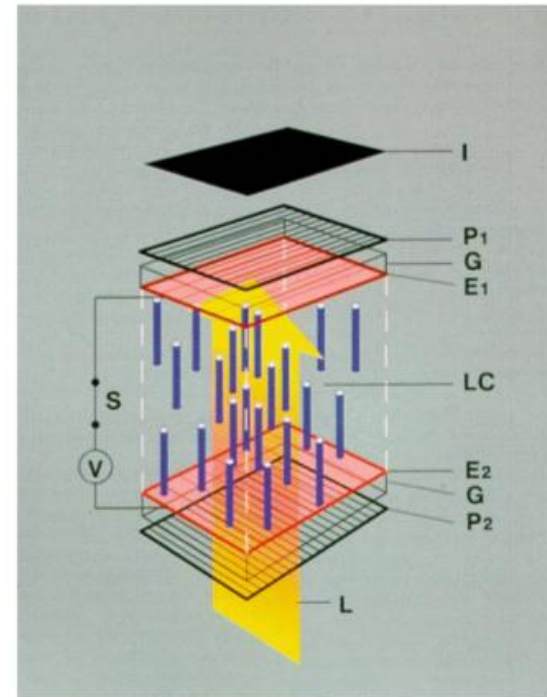
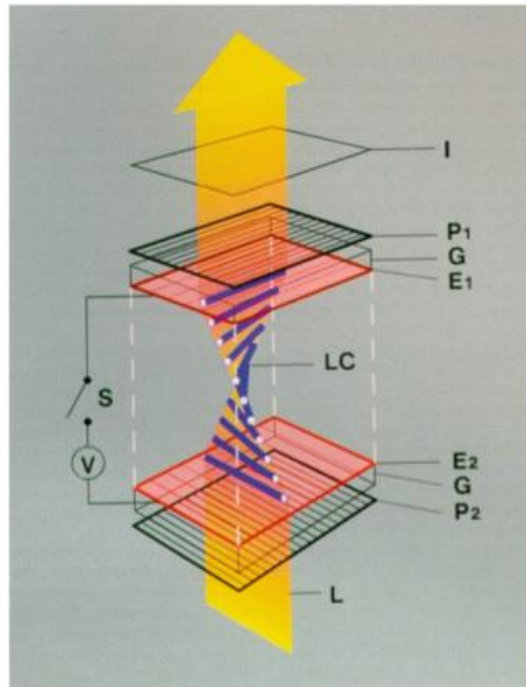
pcdtest – egyszerű mintaprogram a Nokia 5110 grafikus kijelző kipróbálásához



LCD_5110_thermometer – Analóg hőmérő Nokia 5110 kijelzővel

LCD = Liquid Crystal Display (folyadékkristály kijelző)

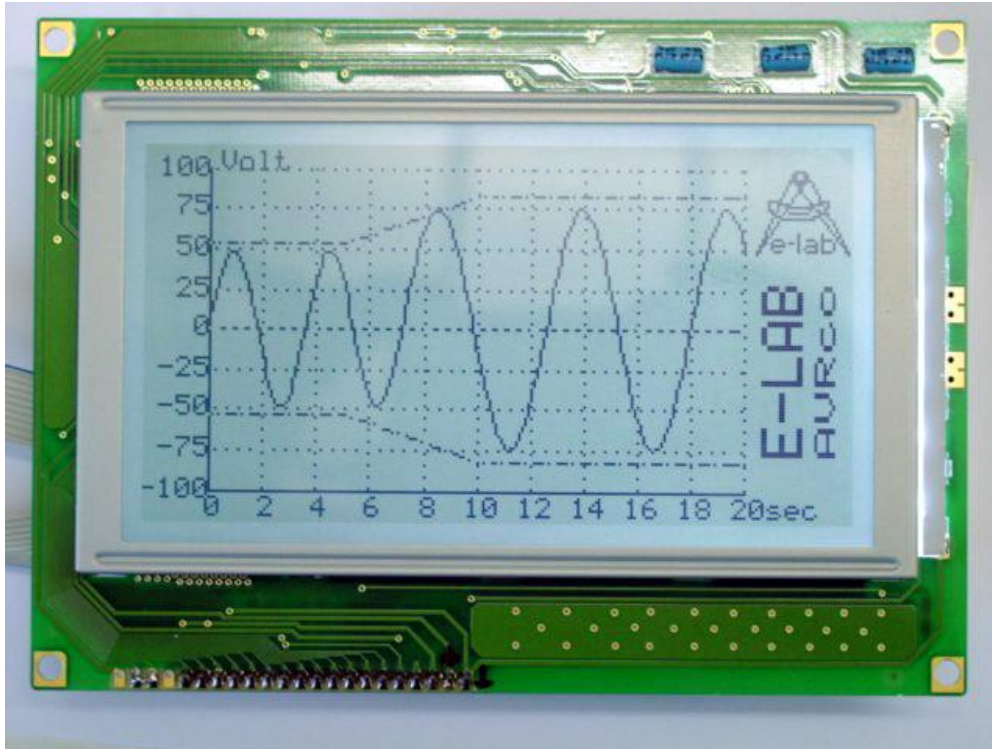
Folyadékkristály: olyan (szerves) anyag, mely sűrű folyadéknak tekinthető, ugyanakkor molekulái – a kristályokhoz hasonlóan – képesek struktúrákba rendeződni.



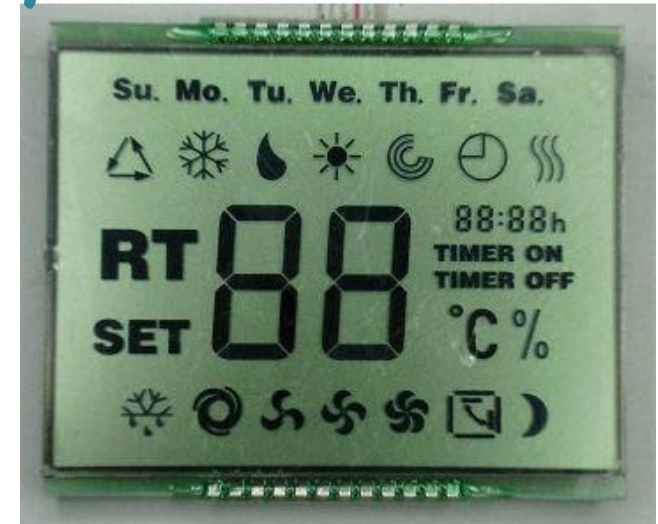
Az alapesetben csavart struktúra segít átjuttatni a polarizált fényt az elforgatott polárszűrőkön. Az elektromos tér átrendezi a molekulákat, már nem segítik a fény átjutását.

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

LCD kijelző típusok



Grafikus pontmátrix kijelző



Szegmens kijelző



Alfanumerikus pontmátrix kijelző (4x20 karakter)

Alfanumerikus LCD

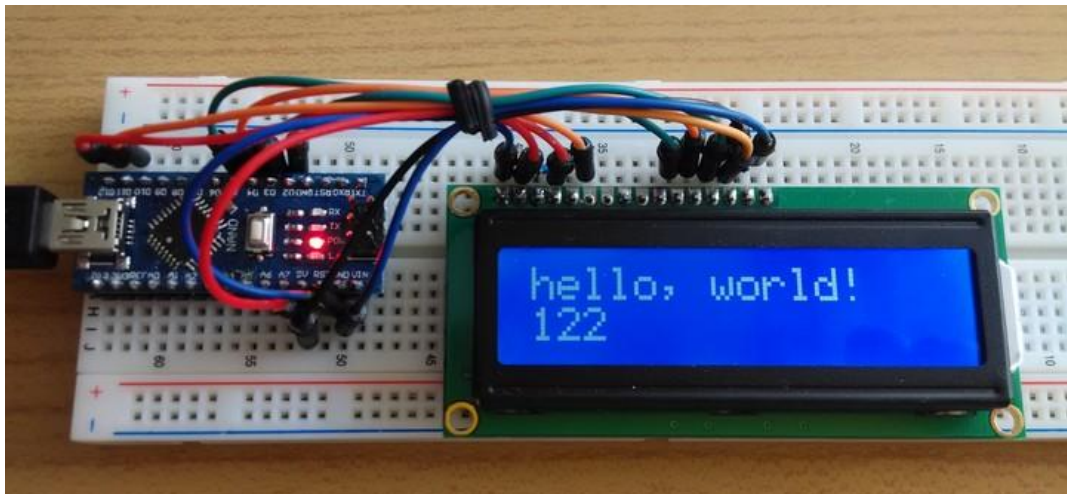
- ❑ HD44780 vezérlő („ipari szabvány”)
- ❑ Háttérvilágítással, vagy reflexiós típus
- ❑ Alfa numerikus: Csak rögzített karakterképet (betű, szám, írásjel) jeleníthetünk meg.
- ❑ 8x2, 16x2, 20x2, 20x4 az elterjedt forma

További információ:

- [Alfanumerikus kijelzők programozása](#)
- [Az intelligens LCD modulok használata - I. rész](#)
- [Az intelligens LCD modulok használata - II. rész](#)

A kivezetések szerepe

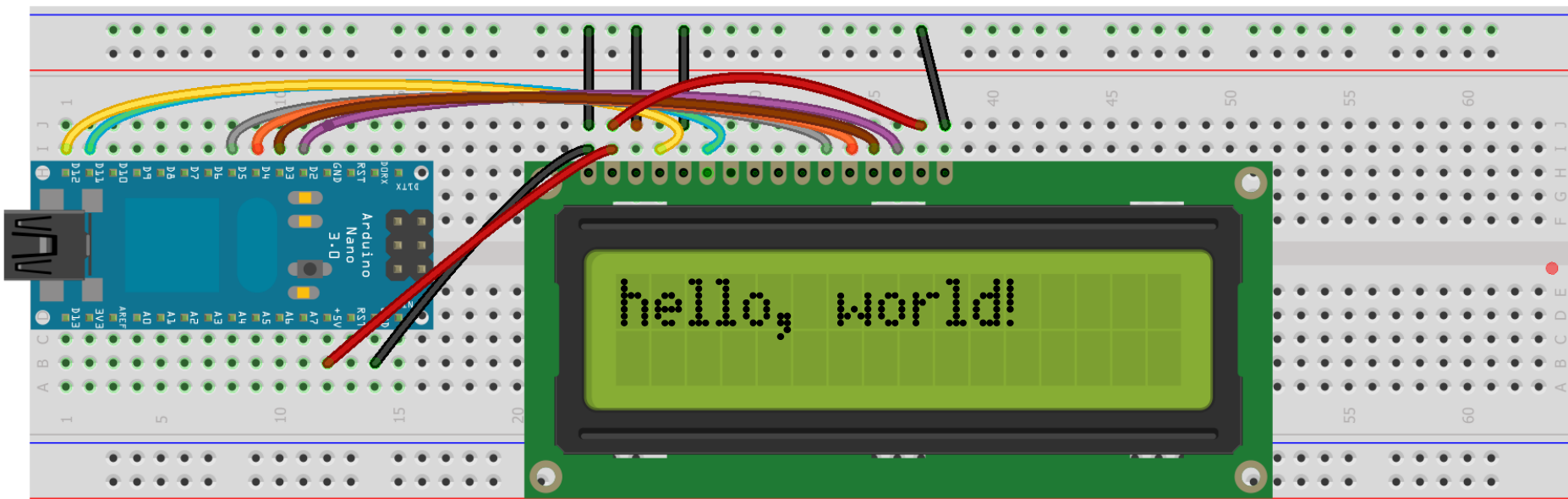
Sorszám	Jel	Funkció
1.	VSS	A tápegység negatív sarka, GND
2.	VCC	Tápfeszültség (többnyire +5 V)
3.	VEE	Kontraszt szabályozó jel (többnyire 0,5 V körül)
4.	RS	Parancs/adat választó (Register Select)
5.	R/W	Olvasás/írás (adatáramlási irány választása)
6.	E	Engedélyező jel (Enable)
7.	D0	Adatbusz 0.bit
8.	D1	Adatbusz 1.bit
9.	D2	Adatbusz 2.bit
10.	D3	Adatbusz 3.bit
11.	D4	Adatbusz 4.bit
12.	D5	Adatbusz 5.bit
13.	D6	Adatbusz 6.bit
14.	D7	Adatbusz 7.bit
15.	A	A háttérvilágítás pozitív sarka (LED anód)
16.	K	A háttérvilágítás negatív sarka (LED katód)



Bekötési vázlat

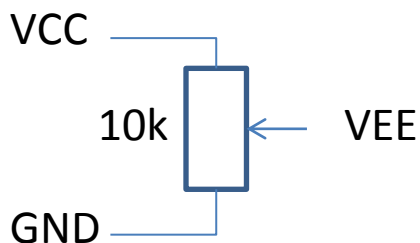
- | | |
|--------------|----------------|
| 1. GND – GND | 11. data4 – D5 |
| 2. VCC – +5V | 12. data5 – D4 |
| 3. VEE – GND | 13. data6 – D3 |
| 4. RS – D12 | 14. data7 – D2 |
| 5. RW – GND | 15. LED+ – +5V |
| 6. EN – D11 | 16. LED- – GND |

Az LCD kijelzőt az Arduino beépített függvényei **4-bites módban** használják (minden bájt két részletben küldünk ki), ezért a **data7-data10** vonalakat nem kell bekötni!

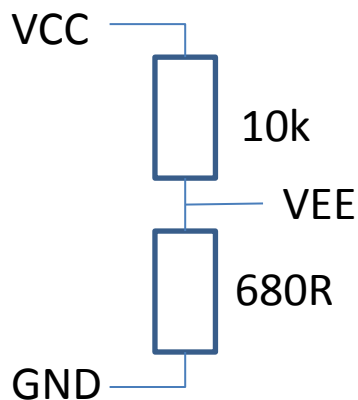


Made with  Fritzing.org

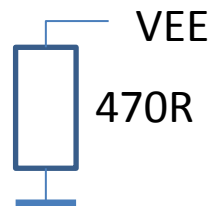
Optimális kontrasztbeállítás



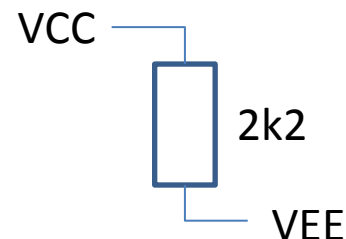
1. Az általánosan javasolt megoldás



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

Adatlap szerint 25 °C-on VEE számára VCC – 4.5 V körüli érték az optimális.

HelloWorld.ino (Helló világ)

Az Arduino IDE beépített mintaprogramja egy szöveget ír ki, majd az indítástól eltelt időt.

```
#include "LiquidCrystal.h"           // LCD "gyári" programkönyvtár becsatolása
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //Példányosítás, lábkiosztás

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

void loop() {
    lcd.setCursor(0, 1);             //Kurzor a második sor elejére
    lcd.print(millis()/1000);        //Kiíratjuk a Reset óta eltelt időt
}
```

LiquidCrystal alkalmazói függvények:

lcd.begin(oszlop,sor) – képernyő inicializálása

lcd.setCursor(oszlop, sor) – kurzor beállítása a megadott helyre

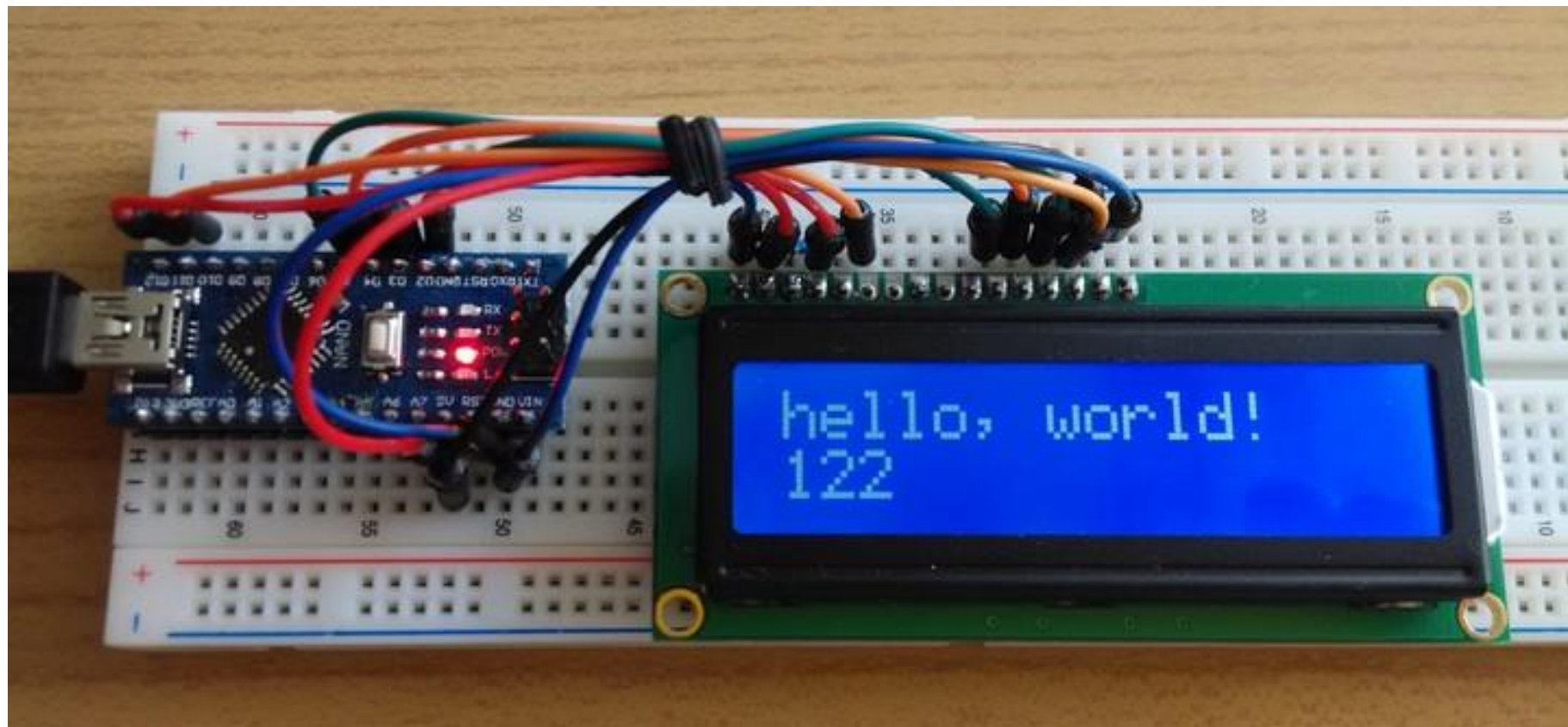
lcd.write(karakterkód) – egy karakter kiírása

lcd.print("szöveg") – szöveg kiírása

lcd.print(kifejezés) – számérték kiírása

HelloWorld.ino (Helló világ)

Az 5V-os kijelző és az **Arduino nano v3.0** kártya közvetlenül összeköthető, nincs inkompatibilitás.
Az **MSP430 Launchpad** kártya esete más, lásd a [2013/14-es Talk11 és Lab11](#) tananyagban!

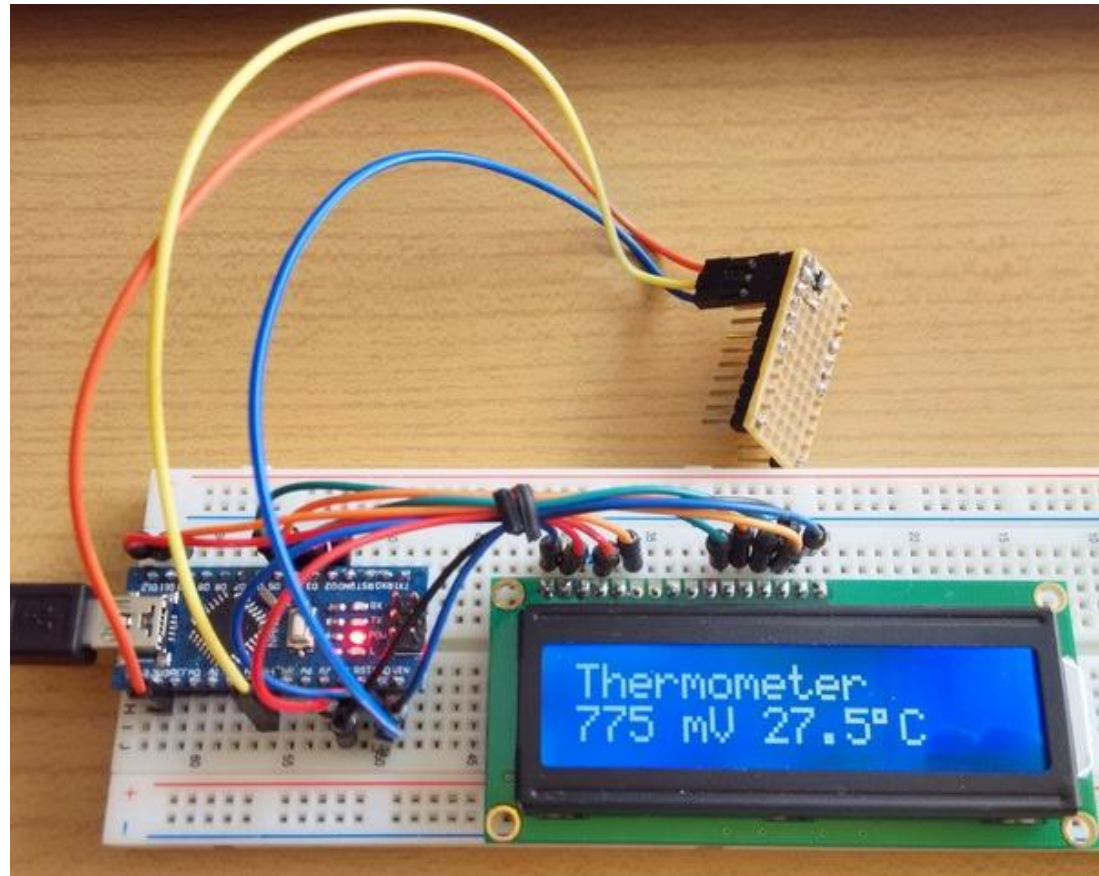
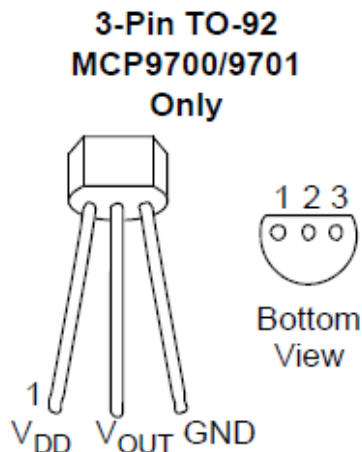


Analóg hőmérő

Csináljunk valami hasznosat! Mérjük meg egy **MCP9700** analóg hőmérő jelét a beépített ADC felhasználásával! A fokokra átszámított hőmérsékletet kijelezhetjük az LCD modul segítségével! A hőmérés elve ugyanaz, mint a **Lab13 AnalogThermometer2** mintapéldában. A kiíráshoz viszont most nem kell számítógép, csak egy LCD kijelző.

Kapcsolás

- ❖ Az LCD bekötése ugyanaz, mint az előző példában.
- ❖ A hőmérő kimenetét az **A4** bemenetre kössük!

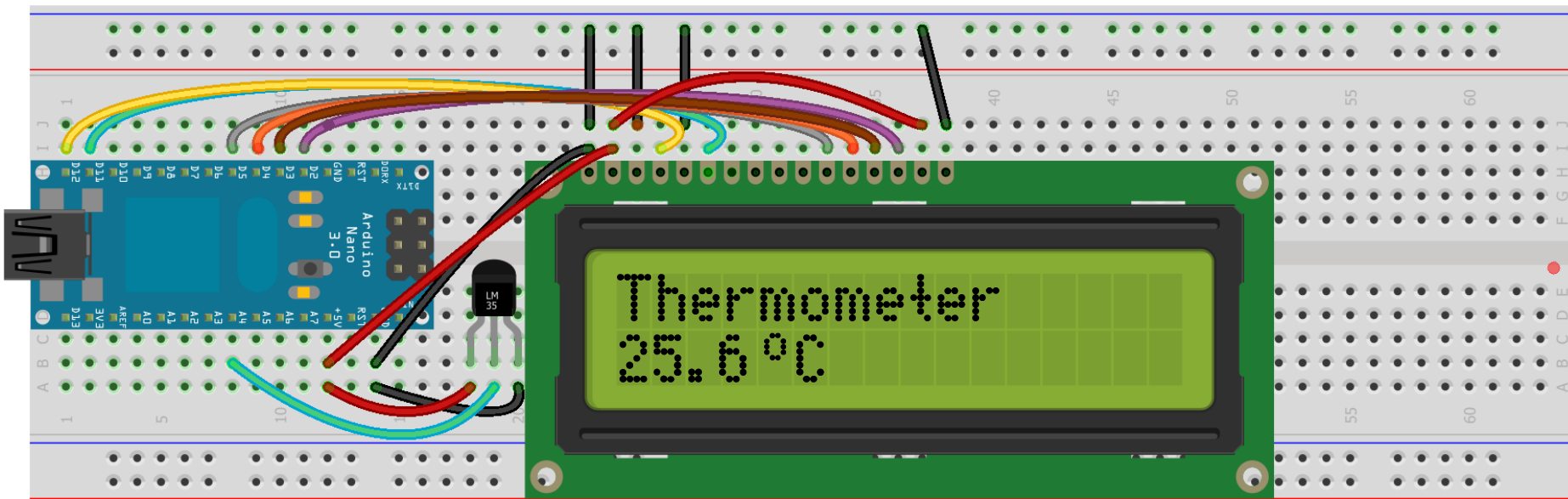


Bekötési vázlat

- | | |
|--------------|----------------|
| 1. GND – GND | 11. data4 – D5 |
| 2. VCC – +5V | 12. data5 – D4 |
| 3. VEE – GND | 13. data6 – D3 |
| 4. RS – D12 | 14. data7 – D2 |
| 5. RW – GND | 15. LED+ – +5V |
| 6. EN – D11 | 16. LED- – GND |

Az LCD kijelzőt az Arduino beépített függvényei **4-bites módban** használják (minden bájt két részletben küldünk ki), ezért a **data7-data10** vonalakat nem kell bekötni!

A hőmérő bekötése: 1: VCC – 5 V, 2: kimenet – A4, 3: GND – GND



LCD_16x2_thermometer.ino

```
#include "LiquidCrystal.h"           //LCD "gyári" programkönyvtár becsatolása
LiquidCrystal lcd(12,11,5,4,3,2);    //Példányosítás, lábkiosztás

void setup() {
    analogReference(INTERNAL);        //Az 1,1 V-os belső referenciát választjuk
    lcd.begin(16, 2);                //Oszlopok és sorok száma
    lcd.print("Thermometer");        //Kiírunk egy szöveget az első sorba
}

void loop() {
    long mysum = 0;                  //ebben összegezzük az eredményt
    for(int i=0; i<1100; i++) {      //Annyi ciklus, ahány mV a referencia...
        mysum += analogRead(A4);     //Itt mérjük meg a feszültséget
    }
    float voltage = mysum>>10;       //Osztás 1024-gyel!!!
    float tempC = (voltage-500)/10;  //Átszámítás Celsius fokokra
    lcd.setCursor(0, 1);             //Kurzor a második sor elejére
    lcd.print(voltage,0);            //Kiíratjuk a mért feszültséget
    lcd.print(" mV ");
    lcd.print(tempC,1);              //Kiíratjuk a hőmérsékletet
    lcd.write(0xDF);                 //fok jelének kiíratása
    lcd.print("C ");                //C betű és térköz kiíratása
    delay(1000);
}
```


Nokia 5110 grafikus kijelző

Ezeket a kijelzőket másfél évtizede még a Nokia 5110/3310 mobilokban használták.

Felbontása: 84x48 pixel, reflexiós típusú monokróm LCD kijelző. Kisméretű, képátlója 1.5".

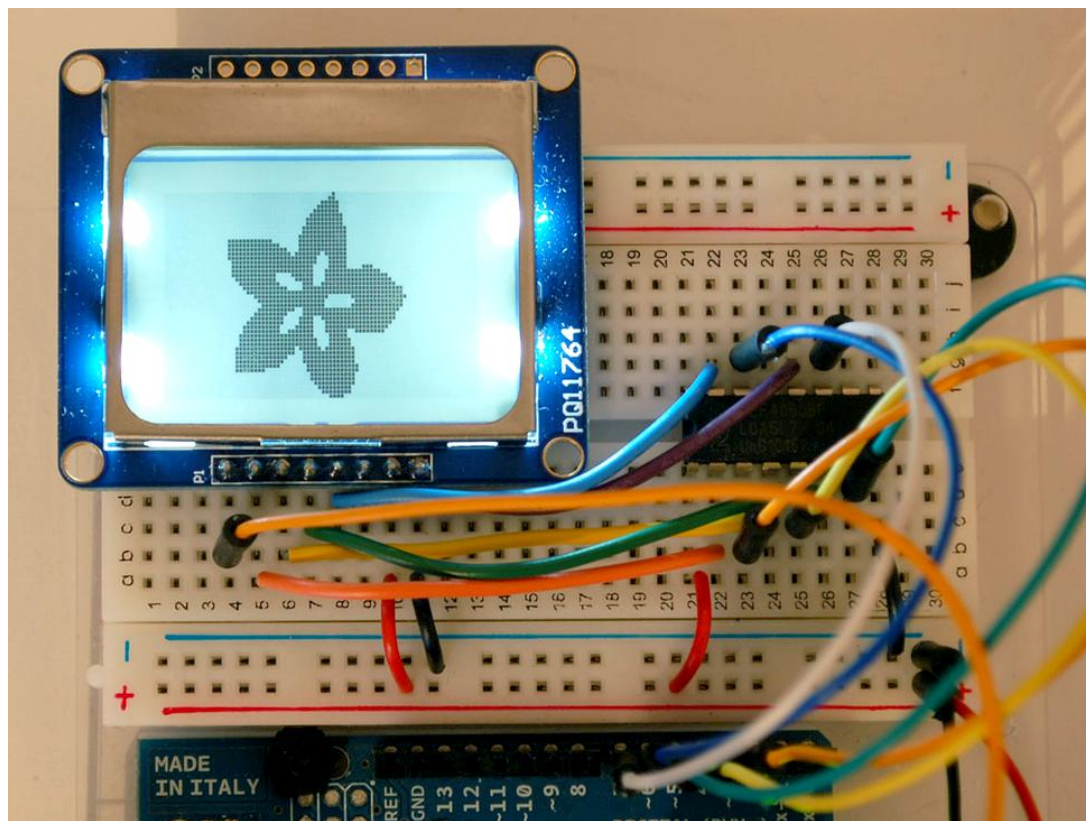
Minden képpont címezhető, így grafikus, szöveges vagy bittérképes kijelzésre használható.

Kis fogyasztású (néhány mA). A „háttérvilágítás”, ami valójában oldalról ad fényt, 4 db SMD LED-et tartalmaz. A vezérlő PCD8744 típusú, a képfelbontás 84*48 képpont.

Megjegyzés: A kijelző 3,3V-os tápfeszültséggel működik, ezért az **Arduino** kártyához történő illesztése odafigyelést igényel!

Jelszint illesztési lehetőségek

- ❖ 3,3 V-os tápfeszültségű mikrovezérlő kártya (MSP430 Launchpad vagy Meduino nano)
- ❖ Szintillesztő áramkör (a képen például CD4050) használata.
- ❖ Feszültségosztó vagy áramkorlátozó ellenállás (5-10 k Ω) az adatvonalakon.



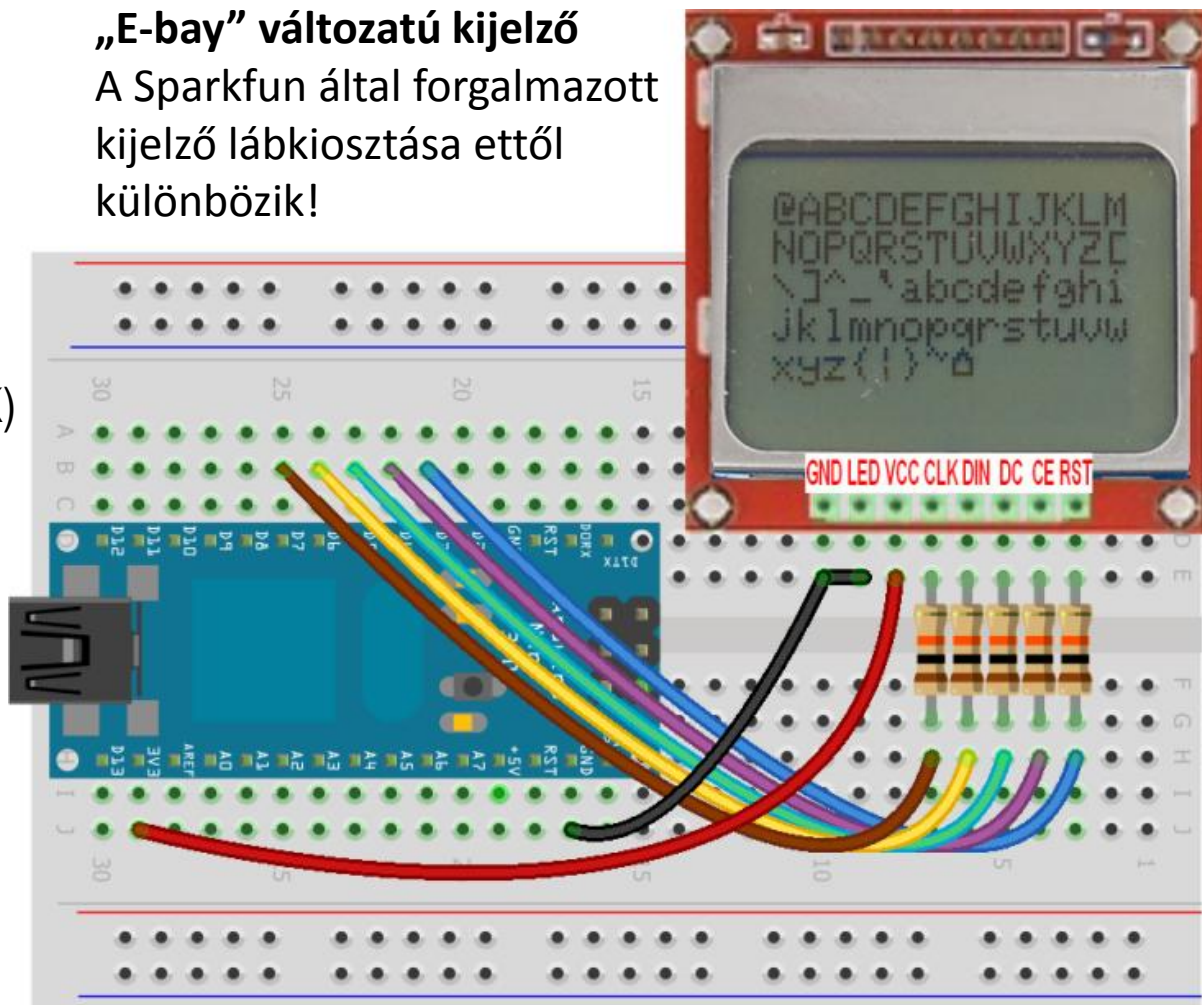
Bekötési vázlat

Nokia 5110 (E-bay)		Arduino
1. RST	RESET	D3
2. CE	kiválasztó jel	D4
3. DC	adat/parancs	D5
4. DIN	MOSI adatvonal	D6
5. CLK	SPI órajel	D7
6. VCC	tápfeszültség	3.3V
7. LIGHT	háttérvilágítás	GND (K)
8. GND	közös pont	GND

Nokia 5110(Sparkfun)		Arduino
1. VCC	tápfeszültség	3.3V
2. GND	közös pont	GND
3. SCE	kiválasztó jel	D4
4. RST	RESET	D3
5. D/C	adat/parancs	D5
6. DIN	MOSI adatvonal	D6
7. SCLK	SPI órajel	D7
8. LED	háttérvilágítás	3.3V (A)

„E-bay” változatú kijelző

A Sparkfun által forgalmazott kijelző lábkiosztása ettől különbözik!



Made with  Fritzing.org

Az Adafruit_PCD8544 programkönyvtár

Lelőhelye: github.com/adafruit/Adafruit-PCD8544-Nokia-5110-LCD-library

Telepíteni kell mellé az [Adafruit-GFX-Library](#) is! A kibontott ZIP állományokat a felhasználói **Arduino** mappa **libraries** nevű almappjába kell elhelyezni.

Az objektumosztály példányosítása és az Arduino kártya kivezetéseinek hozzárendelése):

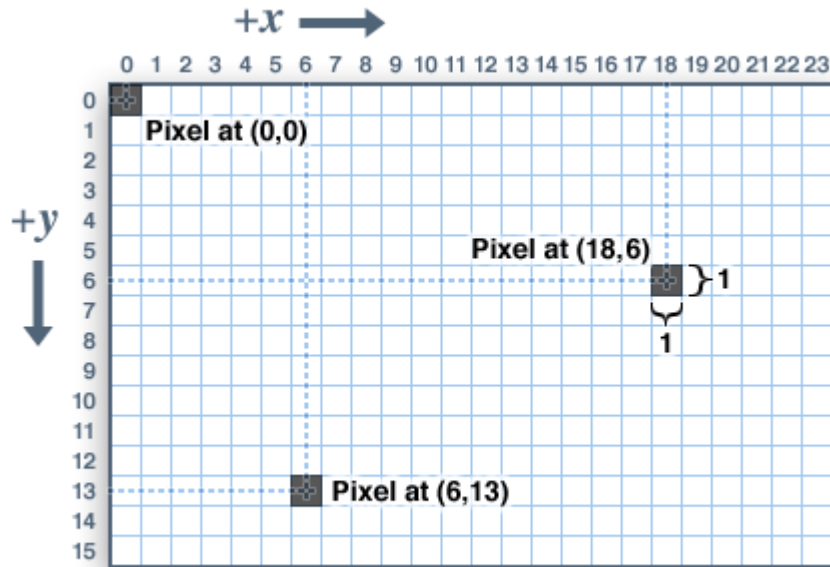
```
Adafruit_PCD8544 display = Adafruit_PCD8544(7, 6, 5, 4, 3);  
                                     SCLK, DIN, DC, CE, RST
```

<code>display.begin();</code>	Inicializálás
<code>display.setContrast(50);</code>	Kontraszt beállítása (általában 50 – 60)
<code>display.clearDisplay();</code>	Képernyő és rajzoló buffer törlése
<code>display.display();</code>	Képernyő újrarajzolása (buffer kiírása)
<code>display.setCursor(x,y);</code>	Kurzor pozíció beállítása (innen rajzolunk)
<code>display.setTextSize(1);</code>	Szövegméret megadása (1: kicsi, 2: nagy)
<code>display.setTextColor(BLACK);</code>	Tintasín megadása (BLACK/WHITE)
<code>display.write(c);</code>	Egy karakter kiírása
<code>display.print("szöveg");</code>	Szöveg kiírása
<code>display.print(kifejezés);</code>	Számérték kiírása
<code>display.drawPixel(x,y,BLACK);</code>	Egy képpont kirajzolása
<code>drawLine(x0,y0,x1,y1,BLACK);</code>	Egy szakasz kirajzolása

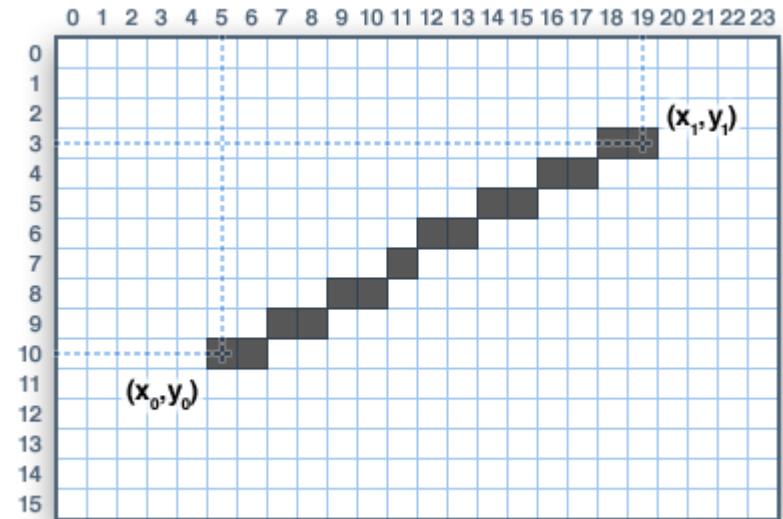
És még sokan mások...

Az Adafruit_GFX programkönyvtár

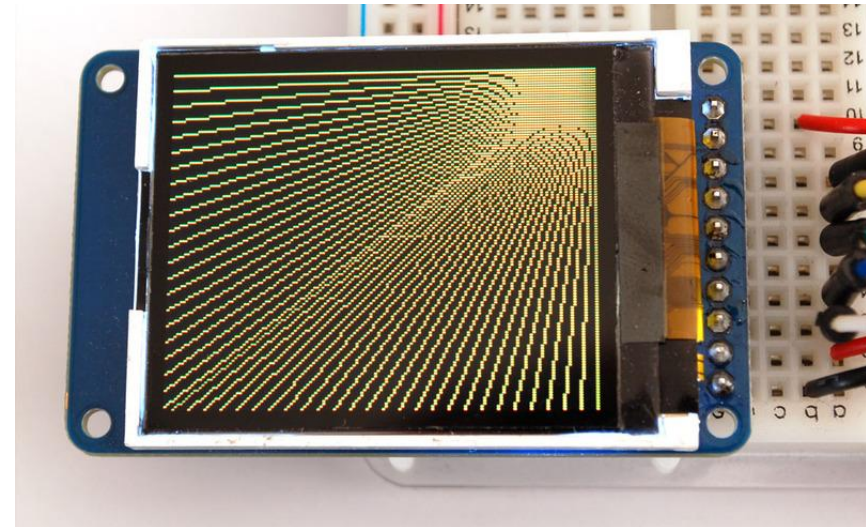
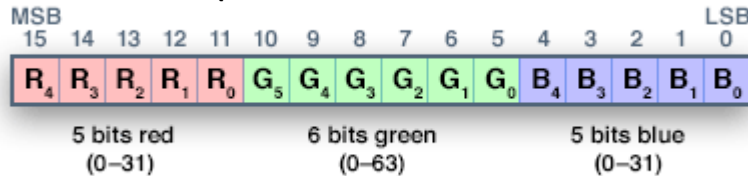
Koordináta-rendszer



Vonal rajzolása



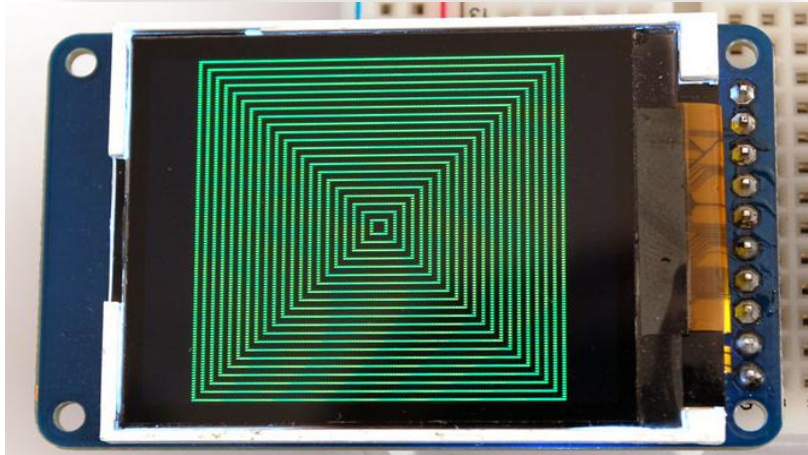
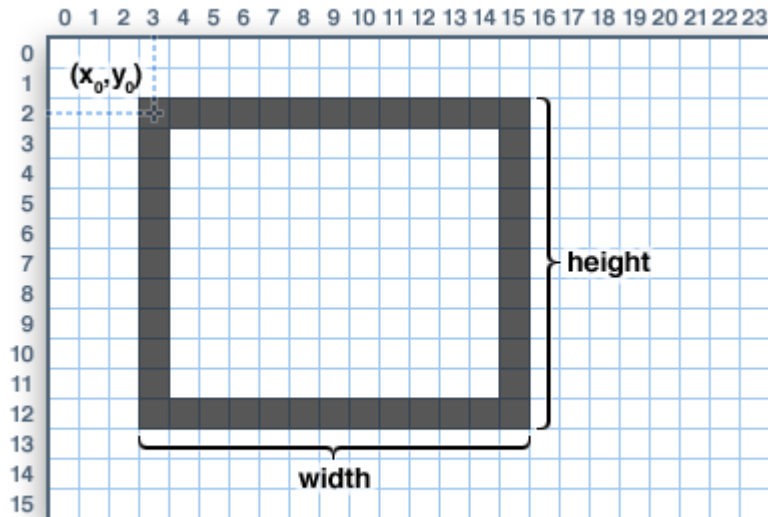
Színkódolás (esetünkben csak 0 és 1 lehet!)



Az Adafruit_GFX programkönyvtár

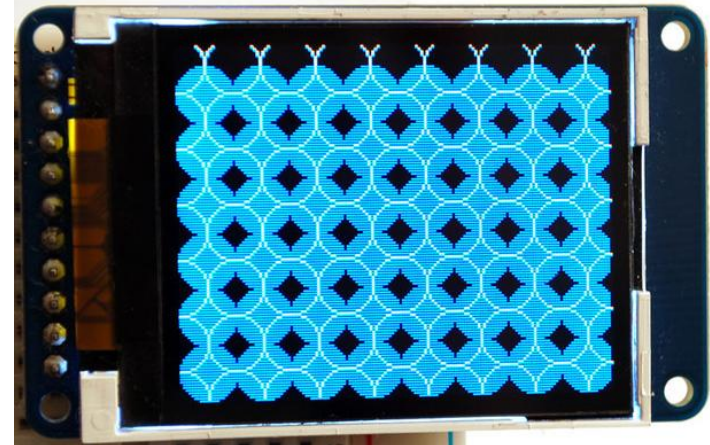
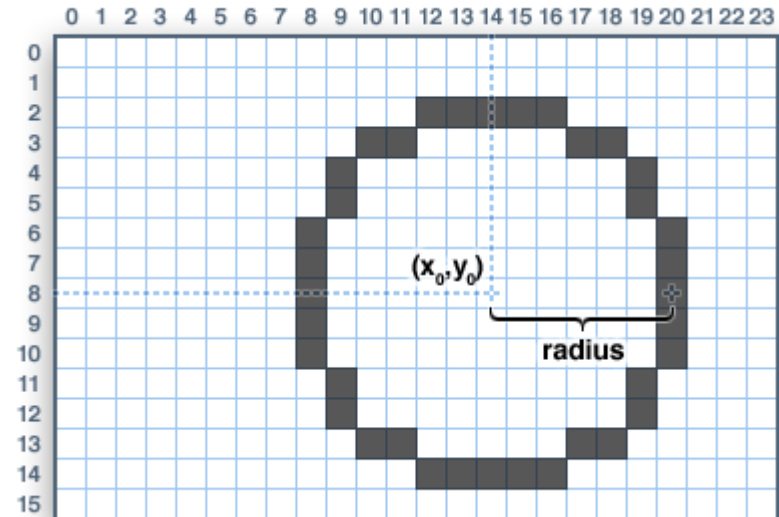
Négyszög rajzolása

```
void drawRect(x0, y0, w, h, color);  
void fillRect(x0, y0, w, h, color);
```



Vonal rajzolása

```
void drawCircle(x0, y0, r, color);  
void fillCircle(x0, y0, r, color);
```



pcdtest.ino (programrészlet)

```
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Adafruit_PCD8544.h>
```

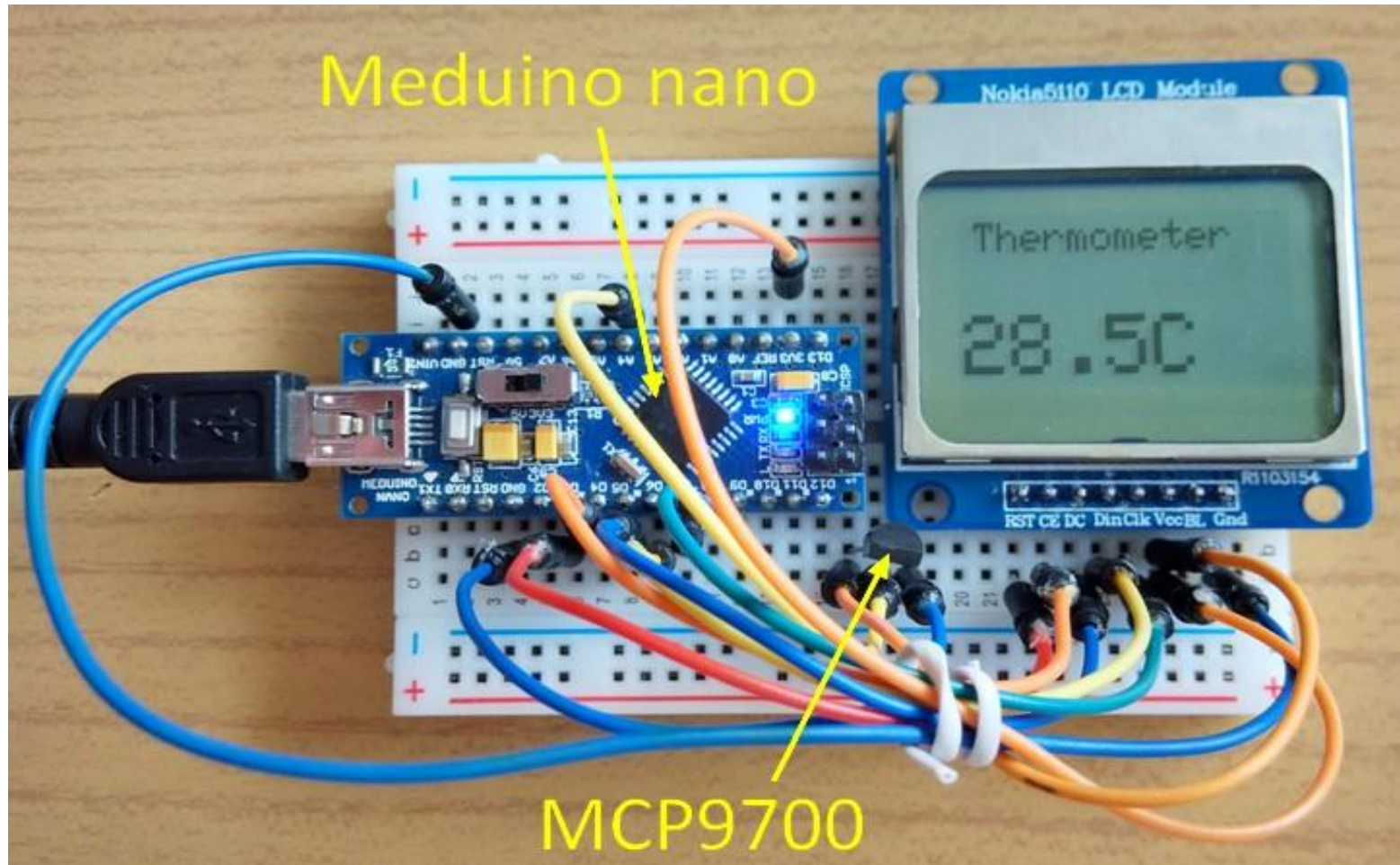
```
Adafruit_PCD8544 display = Adafruit_PCD8544(7, 6, 5, 4, 3);
```

```
void setup() {
  display.begin();           // init done
  display.setContrast(50);
  display.display();         // show splashscreen
  delay(2000);
  display.clearDisplay();    // clears the screen and buffer
  display.drawPixel(10,10,BLACK); // draw a single pixel
  display.display(); delay(2000);
  display.clearDisplay();
  testdrawline();           // draw many lines
  display.display(); delay(2000);
  display.clearDisplay();   // draw rectangles
  testdrawrect();
  display.display(); delay(2000);
  display.clearDisplay();   // draw multiple rectangles
  testfillrect();
  display.display(); delay(2000);
  display.clearDisplay();
  . . .
}
```

Ez a program bemutatja a GFX grafikus programkönyvtár lehetőségeit.

Hőmérő Nokia 5110 kijelzővel

Készítsünk hőmérőt a **Nokia 5110** kijelző felhasználásával! A hőmérést ugyanúgy egy **MCP9700** analóg hőmérővel és a beépített ADC felhasználásával végezzük, mint a korábbi példaprogramnál, csak a kijelzést kell átdolgozni.



LCD_5110_thermometer.ino

Bekötés a szoftveres SPI támogatás szerint:

D7 – SPI clock	(SCLK)
D6 – SPI data	(DIN)
D5 - Data/Command	(DC)
D4 - LCD chip select	(CS)
D3 - LCD reset	(RST)

A hőmérő kimenete az
A4 analóg bemenetre
legyen kötve!

```
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Adafruit_PCD8544.h>
```

```
Adafruit_PCD8544 display = Adafruit_PCD8544(7, 6, 5, 4, 3);
```

```
void setup() {
    display.begin();           //inicializálás
    display.setContrast(50);   //kontraszt beállítása (50-60)
    display.display();         //Adafruit logo megjelenítése
    delay(2000);
    analogReference(INTERNAL); //A belső 1.1V referencia kell!
}
```

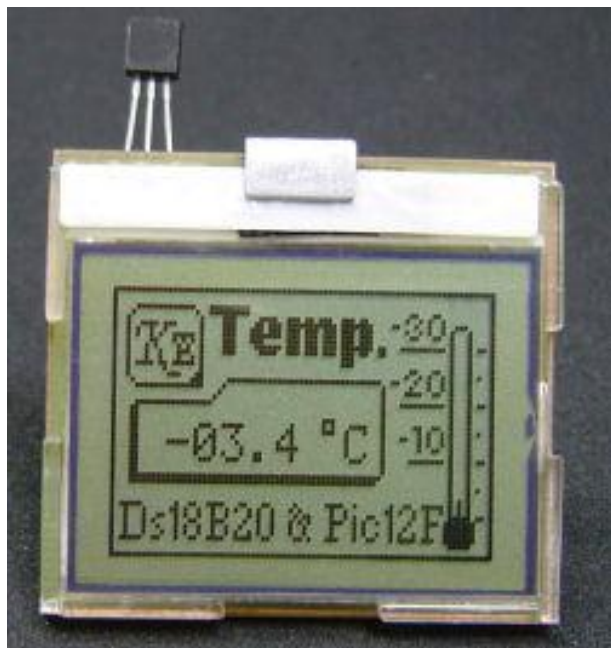
Folytatás a következő oldalon. . .

LCD_5110_thermometer.ino

```
void loop() {  
    long mysum = 0;  
    for(int i=0; i<1100; i++) {  
        mysum += analogRead(A4);    //Mérés és átlagolás  
    }  
    delay(2000);  
    // Átszámítjuk millivoltokra  
    float voltage = mysum>>10;    //Osztás 1024-gyel  
    float tempC = (voltage-500)/10; //Átszámítás fokokra  
    display.clearDisplay();        //Képernyő törlés  
    display.setTextSize(1);        //Szövegméret: kicsi  
    display.setTextColor(BLACK);   //Fekete tintaszín  
    display.setCursor(4,2);        //kiindulási hely  
    display.print("Thermometer");  //Szöveg kiírás  
    display.setTextSize(2);        //Szövegméret: nagy  
    display.setCursor(1,24);       //Kiindulási hely  
    display.print(tempC,1);        //Hőmérséklet kiírása  
    display.print("C");            //Celsius jelzése  
    display.display();             //Megjelenítés  
}
```


További felhasználási ötletek

Hőmérő, alfanumerikus és grafikus kijelzéssel



Óra, ébresztő, öröknaptár, névnap kijelzéssel és hőmérő

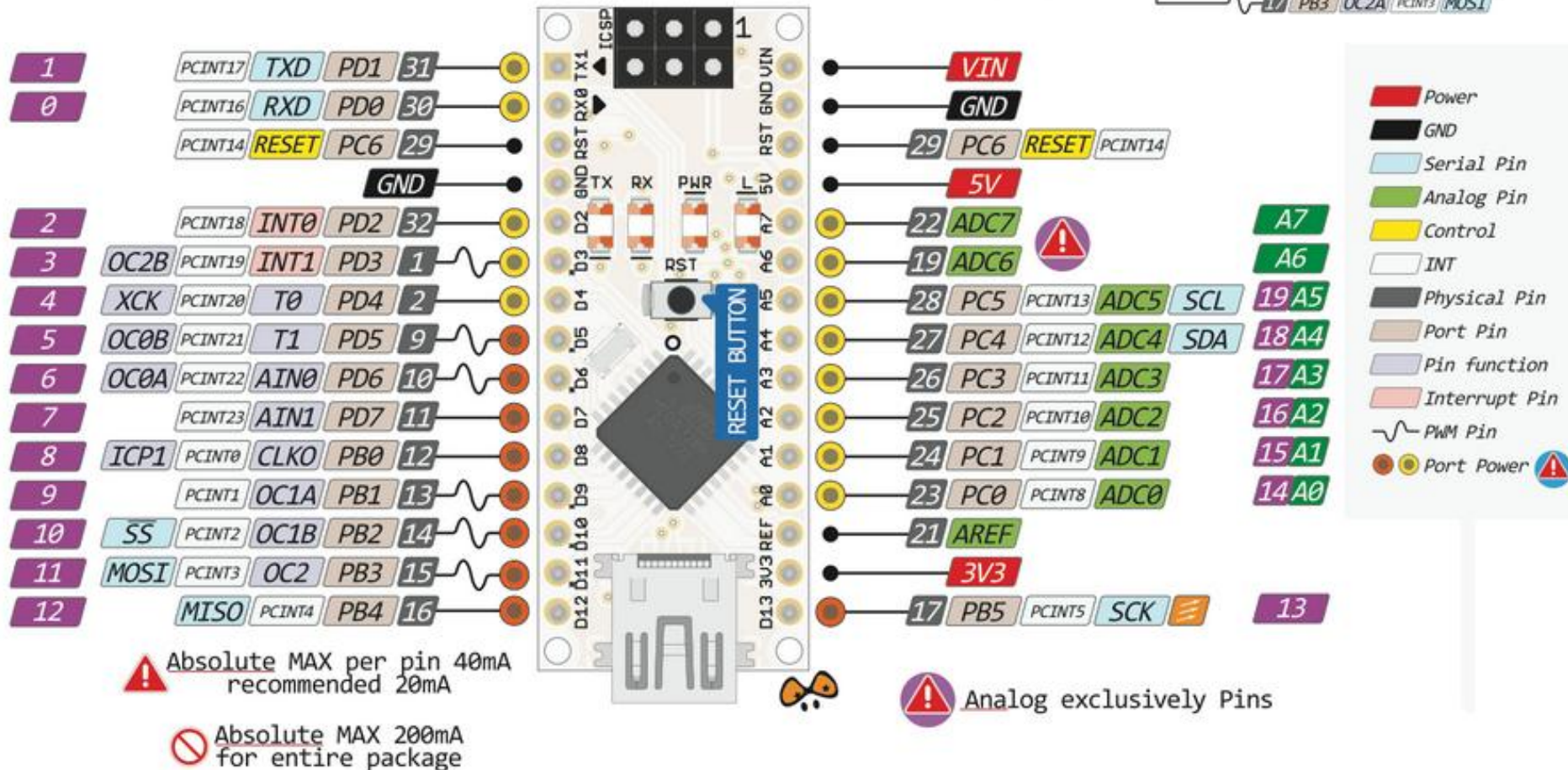
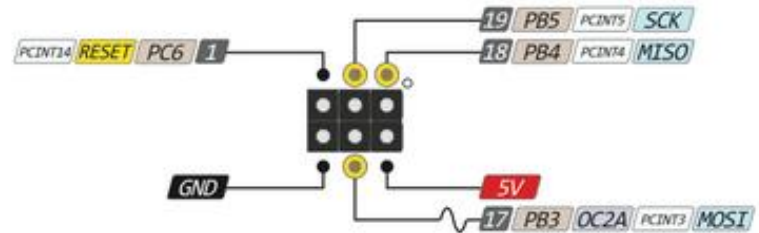


Emlékeztető: Arduino nano v3.0



NANO PINOUT

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



MSP430 Launchpad : Energia Pinout

<http://github.com/energia/Energia/wiki/Hardware>



Energia

LaunchPad with MSP430G2553

Revision 1.5

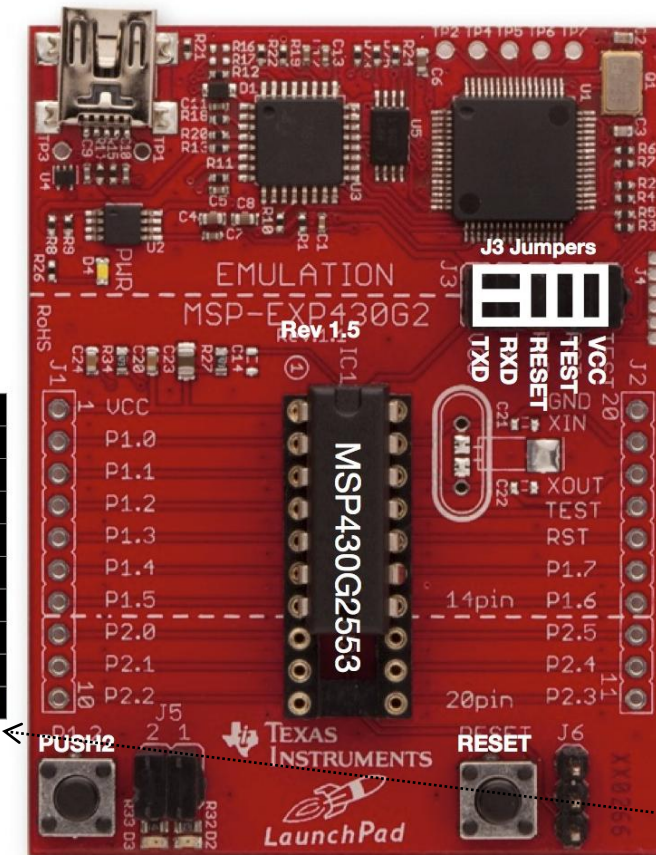
Flash 16 KB
Serial Hardware

Hardware
Pin number

I²C
Serial UART
SPI

analogRead()
digitalRead() and digitalWrite()
digitalRead(), digitalWrite()
and analogWrite()

+3.3V				1
RED_LED		A0	P1_0	2
	RXD	A1	P1_1	3
	TXD	A2	P1_2	4
PUSH2		A3	P1_3	5
		A4	P1_4	6
	SCK (B0)	A5	P1_5	7
	CS (B0)		P2_0	8
			P2_1	9
			P2_2	10



20				GROUND
19	P2_6			XIN
18	P2_7			XOUT
17				TEST
16				RESET
15	P1_7	A7	SDA	MOSI (B0)
14	P1_6	A6	SCL	MISO (B0)
13	P2_5			GREEN_LED
12	P2_4			
11	P2_3			

Rei Vilo, 2012

embeddedcomputing.weebly.com

version 1.3 2102-09-09

Arduino/Energia logical pin #'s