

```
Blink | Energia 0101E0010
File Edit Sketch Tools Help
Blink $
/*
Blink
Egy másodpercre bekapcsoljuk a piros LED-et, azután egy
másodpercre lekapcsoljuk, s ezt ismételtetjük.

Ez a mintaprogram szabadon felhasználható (public domain).
*/

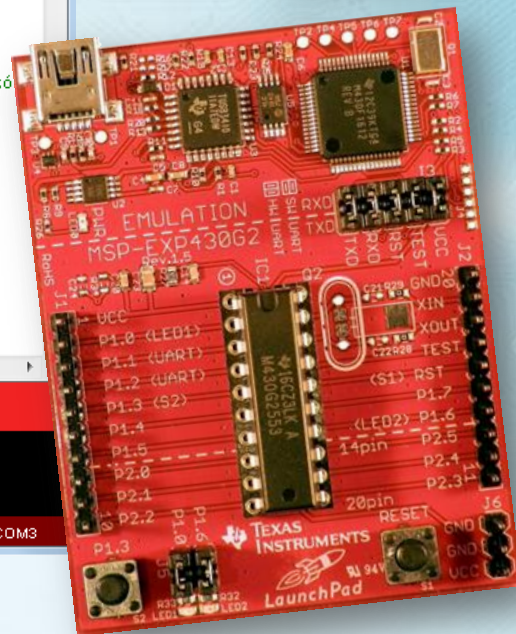
void setup() {
  // Digitális kimenetnek konfiguráljuk a piros LED-hez tartozó
  pinMode(RED_LED, OUTPUT);
}

void loop() {
  digitalWrite(RED_LED, HIGH); // bekapcsoljuk a LED-et
  delay(1000); // várunk egy másodpercig
  digitalWrite(RED_LED, LOW); // kikapcsoljuk a LED-et
  delay(1000); // várunk egy másodpercig
} |

19 LaunchPad w/ msp430g2553 (16MHz) on COM3
```



# Energia



## MSP430 programozás Energia környezetben

Az első lépések ...

# Előfeltételek

## Előfeltételek

- Alapszintű C programozási ismeretek ([Kernighan és Ritchie könyv](#))
- A C könyvtárak és fejléc állományok használatának alapszintű ismerete
- Alapvető ismeretek az órajelgenerátor, a megszakítások a GPIO tulajdonságok és a perifériákkal (időzítők, SPI, I2C) kapcsolatban. Adatlap és korábbi cikkek alapján...

## Szükséges eszközök és szoftverek

### ❖ Hardver

- *USB porttal ellátott PC, Windows (XP, 7, 8), Linux vagy Mac OS*
- *MSP430 Launchpad (v1.5)*

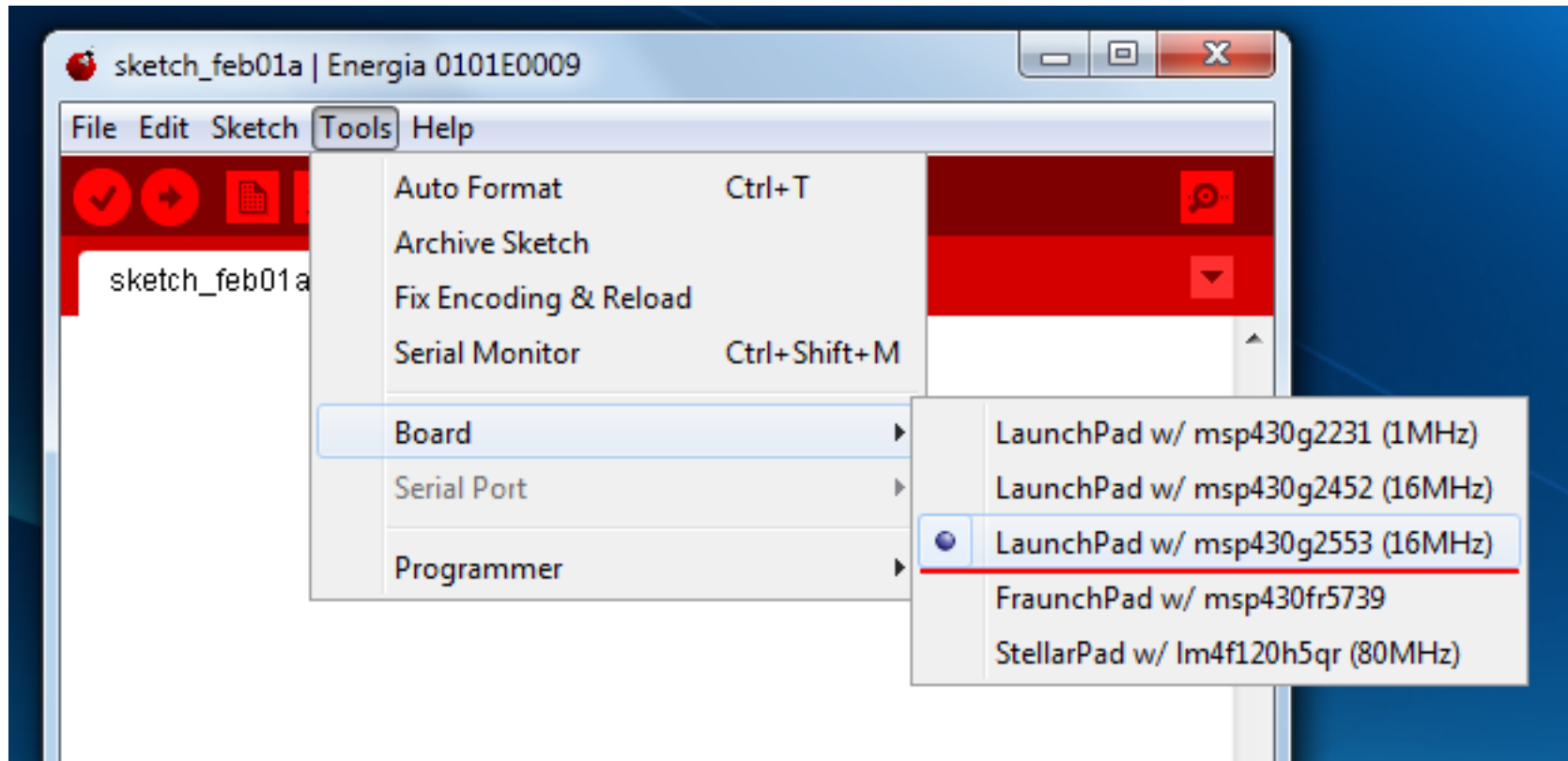
### ❖ Szoftver

- *Windows (XP, 7, 8), Linux vagy Mac OS*
- *Energia (letöltés: <http://energia.nu>)*
- *Launchpad USB meghajtó szoftver (az Energia is tartalmazza!): [EZ430-UART.zip](#)*

## Telepítés

- *Az **Energia** telepítő csomagját bontsuk ki egy alkalmas helyre (pl. C:\)!*
- *Az **EZ430-UART.zip** csomagot bontsuk ki és futtassuk a **DPinst.exe** programot!*

# A kártya kiválasztása

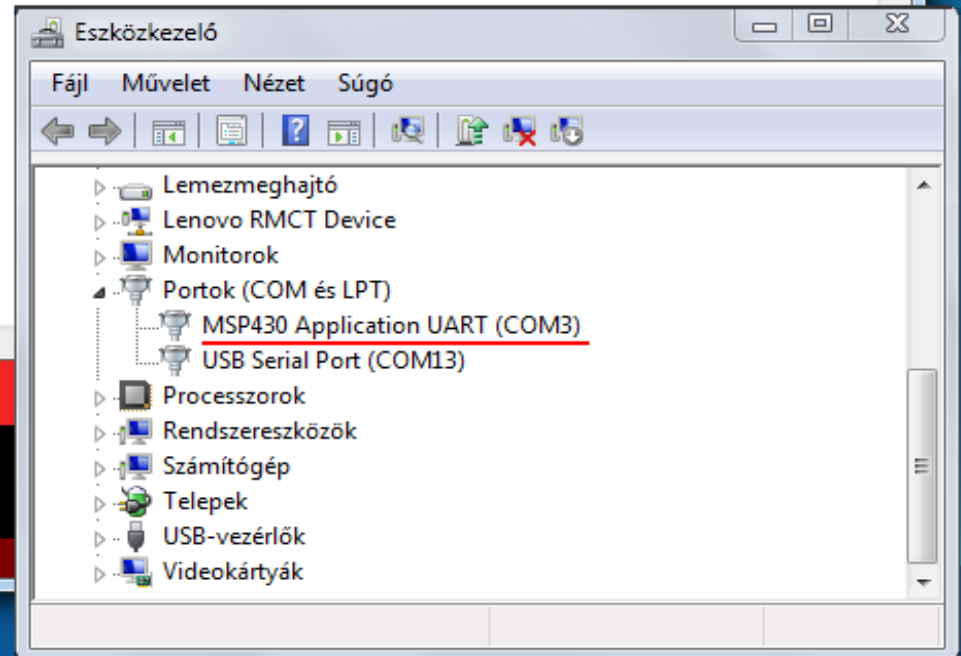
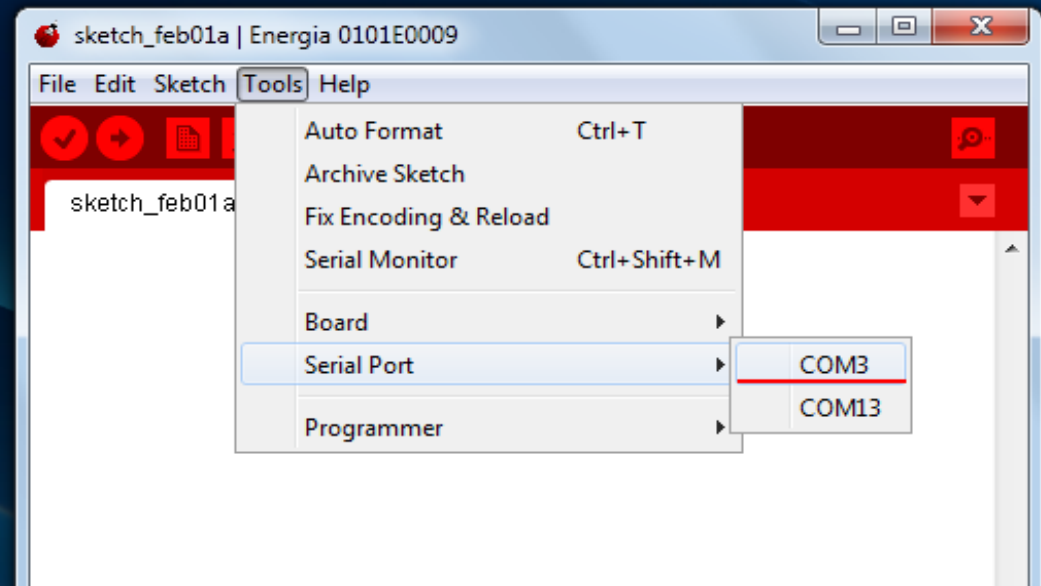


*Csak az első indításkor kell beállítani, azután az Energia megjegyzi.*

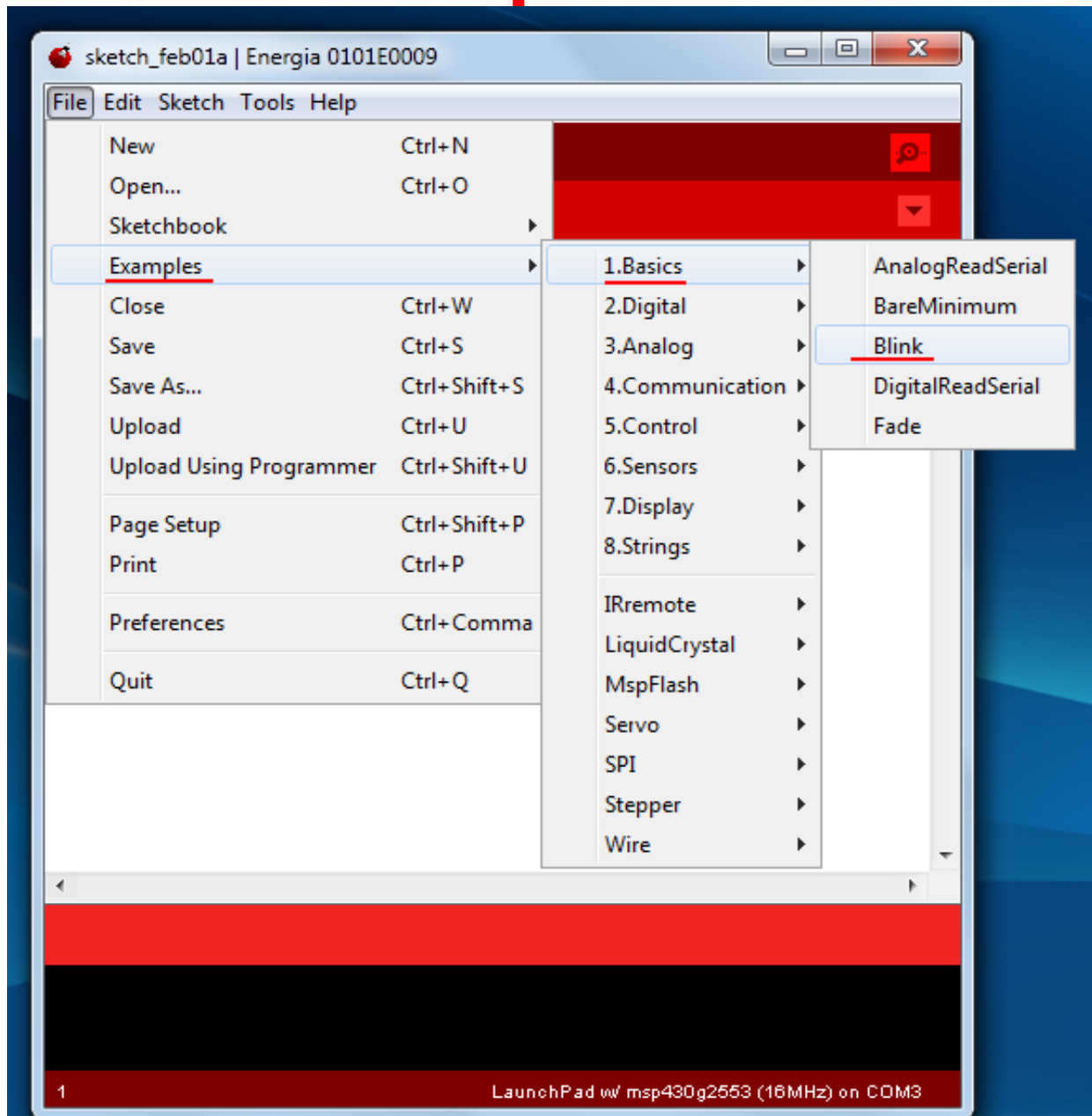
# A kommunikációs port beállítása

Ha több port is elérhető, akkor az Eszközkezelőben kell megnézni, hogy melyik az MSP430 alkalmazási soros port.

A COM és LPT szekcióban kell keresni...



# Az első mintapélda betöltése





# MSP430 Launchpad : Energia Pinout

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

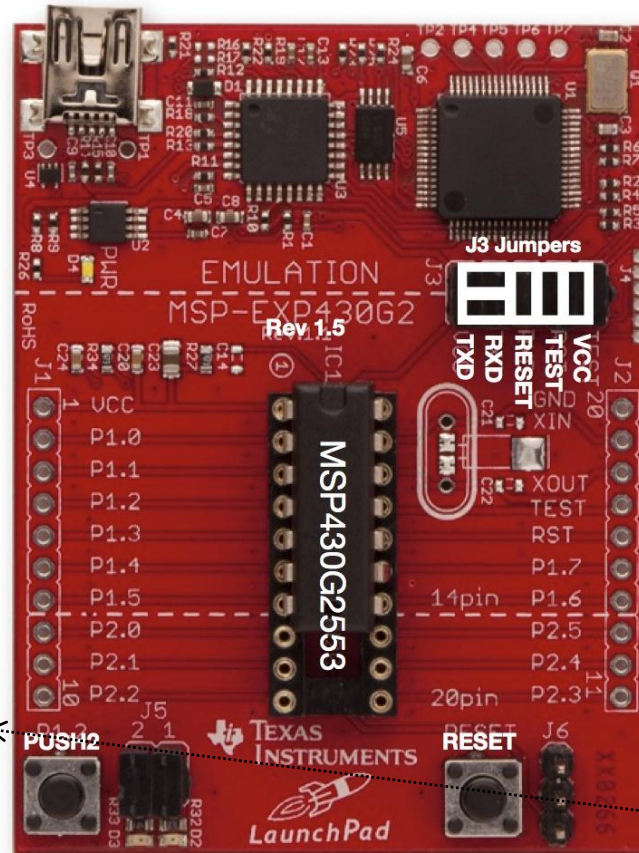


## LaunchPad with MSP430G2553

Revision 1.5

Flash 16 KB  
Serial Hardware

Hardware
Pin number
IPC
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			

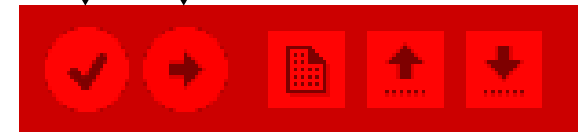
Arduino/Energia logical pin #'s

# Az Energia kezelése



Fordítás/Ellenőrzés

Letöltés



Új

Megnyitás

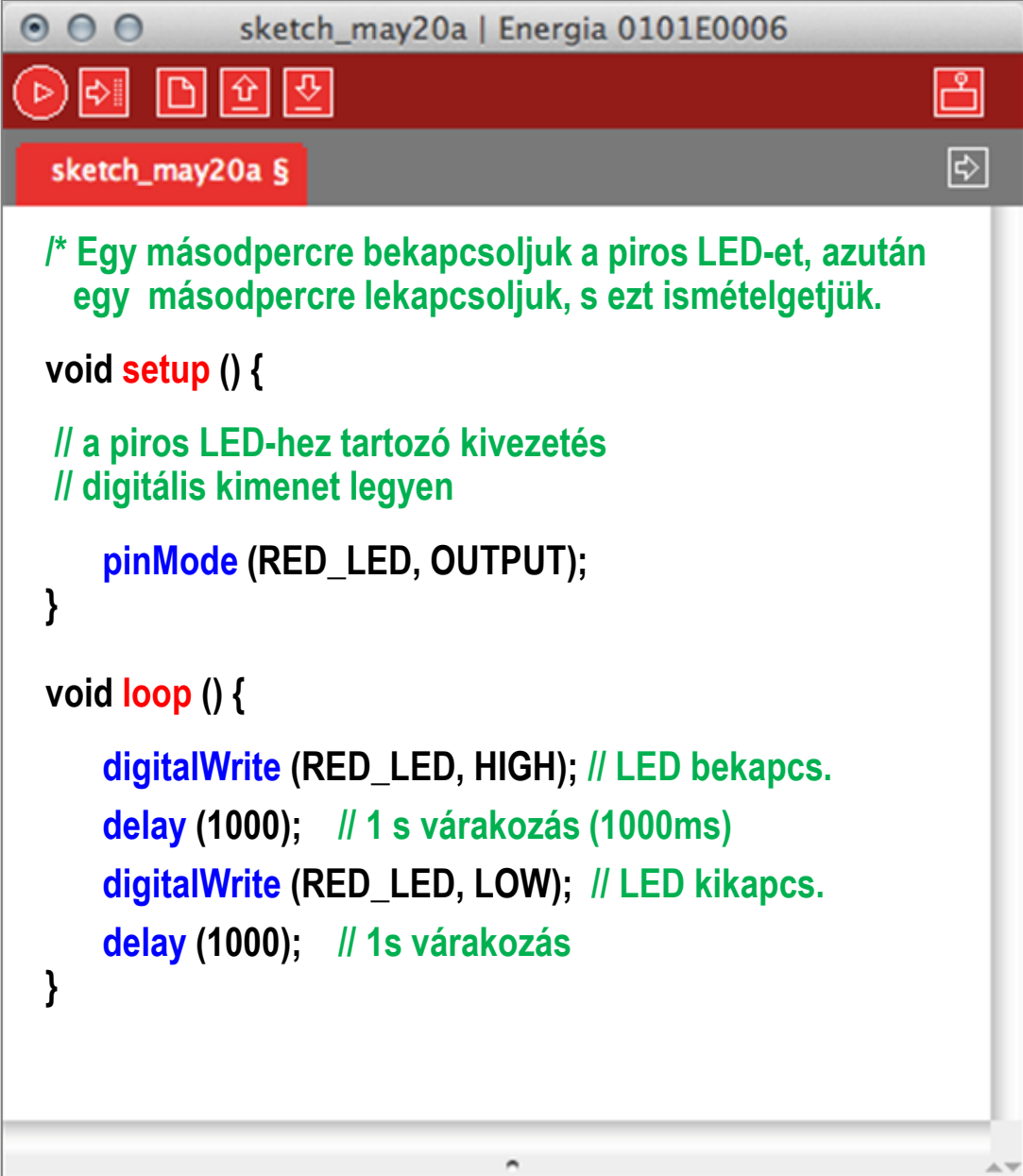
Mentés

## ◆ Letöltés gomb...

- ◆ Elvégzi a fordítást és letölti a programot a céláramkörbe

# Programozás az Energia környezetben

- ◆ Az Energia programok elnevezése *sketch* (vázlat)
- ◆ A sketch tipikusan két függvényt definiál:
  - ◆ `setup()`
  - ◆ `loop()`
- ◆ Vannak C/C++ programok amelyek az Arduino/Wiring könyvtárakat használják
  - A könyvtárak az IDE-vel települnek
- ◆ Ha szükséges, hozzáférhetünk az MCU HW specifikus részeihez, de ez korlátozza a hordozhatóságot
- ◆ Ledvillogtatás az MCU-k 'Hello World' példája
  - ◆ 'Wiring' egyszerűvé teszi
  - ◆ Nem optimalizált példa



```
sketch_may20a | Energia 0101E0006

sketch_may20a §

/* Egy másodpercre bekapcsoljuk a piros LED-et, azután
   egy másodpercre lekapcsoljuk, s ezt ismételtjük.

void setup () {

  // a piros LED-hez tartozó kivezetés
  // digitális kimenet legyen

  pinMode (RED_LED, OUTPUT);
}

void loop () {

  digitalWrite (RED_LED, HIGH); // LED bekapcs.
  delay (1000); // 1 s várakozás (1000ms)
  digitalWrite (RED_LED, LOW); // LED kikapcs.
  delay (1000); // 1s várakozás
}
```



# Energia Referencia kézikönyv

## Structure

- setup()
- loop()

## Control Structures

- if
- if...else
- for
- switch case
- while
- do... while
- break
- continue
- return
- goto

## Further Syntax

- ; (semicolon)
- { } (curly braces)
- // (single line comment)
- /\*\* \*/ (multi-line comment)
- #define
- #include

## Arithmetic Operators

- = (assignment operator)
- + (addition)
- - (subtraction)
- \* (multiplication)
- / (division)
- % (modulo)

## Variables

### Constants

- HIGH | LOW
- INPUT | OUTPUT
- INPUT\_PULLUP | INPUT\_PULLDOWN
- true | false
- integer constants
- floating point constants

### Data Types

- void
- boolean
- char
- unsigned char
- byte
- int
- unsigned int
- word
- long
- unsigned long
- float
- double
- string - char array
- String - object
- array

### Conversion

- char()
- byte()
- int()
- word()

## Functions

### Digital I/O

- pinMode()
- digitalWrite()
- digitalRead()

### Analog I/O

- analogReference()
- analogRead()
- analogWrite() - PWM

### Advanced I/O

- tone()
- noTone()
- shiftOut()
- shiftIn()
- pulseIn()

### Time

- millis()
- micros()
- delay()
- delayMicroseconds()

### Math

- min()
- max()
- abs()

### Comparison Operators

- == (equal to)
- != (not equal to)
- < (less than)
- > (greater than)
- <= (less than or equal to)
- >= (greater than or equal to)

### Boolean Operators

- && (and)
- || (or)
- ! (not)

### Pointer Access Operators

- \* dereference operator
- & reference operator

### Bitwise Operators

- & (bitwise and)
- | (bitwise or)
- ^ (bitwise xor)
- ~ (bitwise not)
- << (bitshift left)
- >> (bitshift right)

# Hol találunk további információt?

## ◆ Energia

- ◆ Honlap: <http://energia.nu/>
- ◆ Letöltés: <http://energia.nu/download/>
- ◆ Wiki: <https://github.com/energia/Energia/wiki>
- ◆ Támogatott kártyák: <https://github.com/energia/Energia/wiki/Hardware>  
(H/W pin mapping)
- ◆ Kezdőlépések: <https://github.com/energia/Energia/wiki/Getting-Started>
- ◆ Támogatói fórum: <http://forum.43oh.com/forum/28-energia/>
- ◆ Magyar cikk: <http://www.hobbielektronika.hu/cikkek/energia....html>

## ◆ Launchpad Boards

- ◆ MSP430: <http://www.ti.com/tool/msp-exp430g2> ([wiki](#)) ([eStore](#))
- ◆ ARM Cortex-M4F: [Launchpad](#) [Wiki](#) [eStore](#)

## ◆ Arduino:

- ◆ honlap: <http://www.arduino.cc/>
- ◆ Referencia: <http://arduino.cc/en/Reference/HomePage>
- ◆ Képregény: <http://www.jodyculkin.com/.../arduino-comic-latest3.pdf>

# Blink demóprogram

```
/*
Blink
Egy másodpercre bekapcsoljuk a piros LED-et, azután egy
másodpercre lekapcsoljuk, s ezt ismételtjük.

Ez a mintaprogram szabadon felhasználható (public domain).
*/

void setup() {
  // Digitális kimenetnek konfiguráljuk a piros LED-hez tartozó ki
  pinMode(RED_LED, OUTPUT);
}

void loop() {
  digitalWrite(RED_LED, HIGH); // bekapcsoljuk a LED-et
  delay(1000); // várunk egy másodpercig
  digitalWrite(RED_LED, LOW); // kikapcsoljuk a LED-et
  delay(1000); // várunk egy másodpercig
}
```

11 LaunchPad w/ msp430g2553 (16MHz) on COM3



# Kérdések és feladatok (2)

3. Hogyan tudjuk megváltoztatni, hogy a másik LED villogjon?

Nézzük meg a kártya kivezetéseinek diagramján, hogy mit kell változtatni!

Kérem, írja le ide: Váltunk a 2. lábról a 14. lábra, hogy a zöld LED villogjon

(Könnyebb út: használja a GREEN\_LED szimbólumot!)

---

4. Módosítsa a programot, hogy másik LED villogjon!

Működik? Igen

# Button mintaprogram

```
const int buttonPin = PUSH2;    // the number of the pushbutton pin
const int ledPin = GREEN_LED;  // the number of the LED pin

// variables will change:
int buttonState = 0;           // variable for reading the pushbutton status

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT_PULLUP);
}

void loop(){
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  }
  else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

## LED vezérlése nyomógommbal

(a forráskód a  
File/Examples/2.Digital  
csoportban található)

Mi történik, ha  
lenyomjuk az SW2  
gombot?



# Kérdések és feladatok (3)

## Button program

1. Próbálja ki a gyári demóprogramot!

Amikor lenyomjuk a gombot (ZÖLD vagy PIROS) LED-et kapcsolja (KI vagy BE)?

Zöld LED-et kapcsolja ki

---

## Vizsgálja meg a kódot!

2. Miben különbözik a 'Button' programban a LED definiálás a 'Blink'-hez képest?

A 'Blink' programban az előre definiált RED\_LED, GREEN\_LED makrókat használtuk  
a 'Button' programban „cont int” típusú változót definiáltunk. Mindkettő használható.

---

3. Miben különbözik a nyomógomb megadása/használata a LED-hez képest?

A Setup()-ban 'input'-ként konfiguráljuk; a loop()-ban digitalRead()-del használjuk

---

A mikrovezérlő melyik lábához van kötve a nyomógomb? Pin 5

Mi a különbség az INPUT és az INPUT\_PULLUP között?

INPUT egyszerű bemeneted definiál – pl. nyomógomb állapot beolvasása.

---

Az INPUT\_PULLUP emellett bekapcsolja a belső felhúzást is;

---

(sok TI  $\mu$ C biztosítja ezt a lehetőséget)

---

# Kérdések és feladatok (4)

4. A standard C nyelvhez hasonlóan használhatunk változókat. Mi az a globális változó, amit a 'Button' példaprogramban használtunk?

A 'buttonState' változó tárolja a nyomógomb állapotát amit a digitalRead() olvas ki.

---

Ezt az értéket használjuk az IF-THEN/ELSE parancs feltételvizsgálata során.

---

Mi lenne hatékonyabb módszer a nyomógomb lenyomásának figyelésére? (Miért fontos ez az MSP430 programozók számára?)

Hatékonyabb lenne a program, ha a nyomógomb állapotának folytonos vizsgálata

---

helyett a CPU energiatakarékos módba kapcsolhatna, ahonnan egy megszakítás

---

ébresztené fel a nyomógomb megnyomásakor. A lekérdezéses módszer nem engedi,

---

hogy a CPU SLEEP módba kapcsoljon.

---

(A későbbiekben majd foglalkozunk ezzel a kérdéssel.)

**Fordítsa meg a Button/LED akciót!**

5. Működik? Igen (muszáj működnie)

```
if (buttonState == HIGH) {  
    // turn LED on:  
    digitalWrite(ledPin, HIGH);  
}  
else {  
    // turn LED off:  
    digitalWrite(ledPin, LOW);  
}
```

LOW

HIGH

# DigitalReadSerial mintaprogram

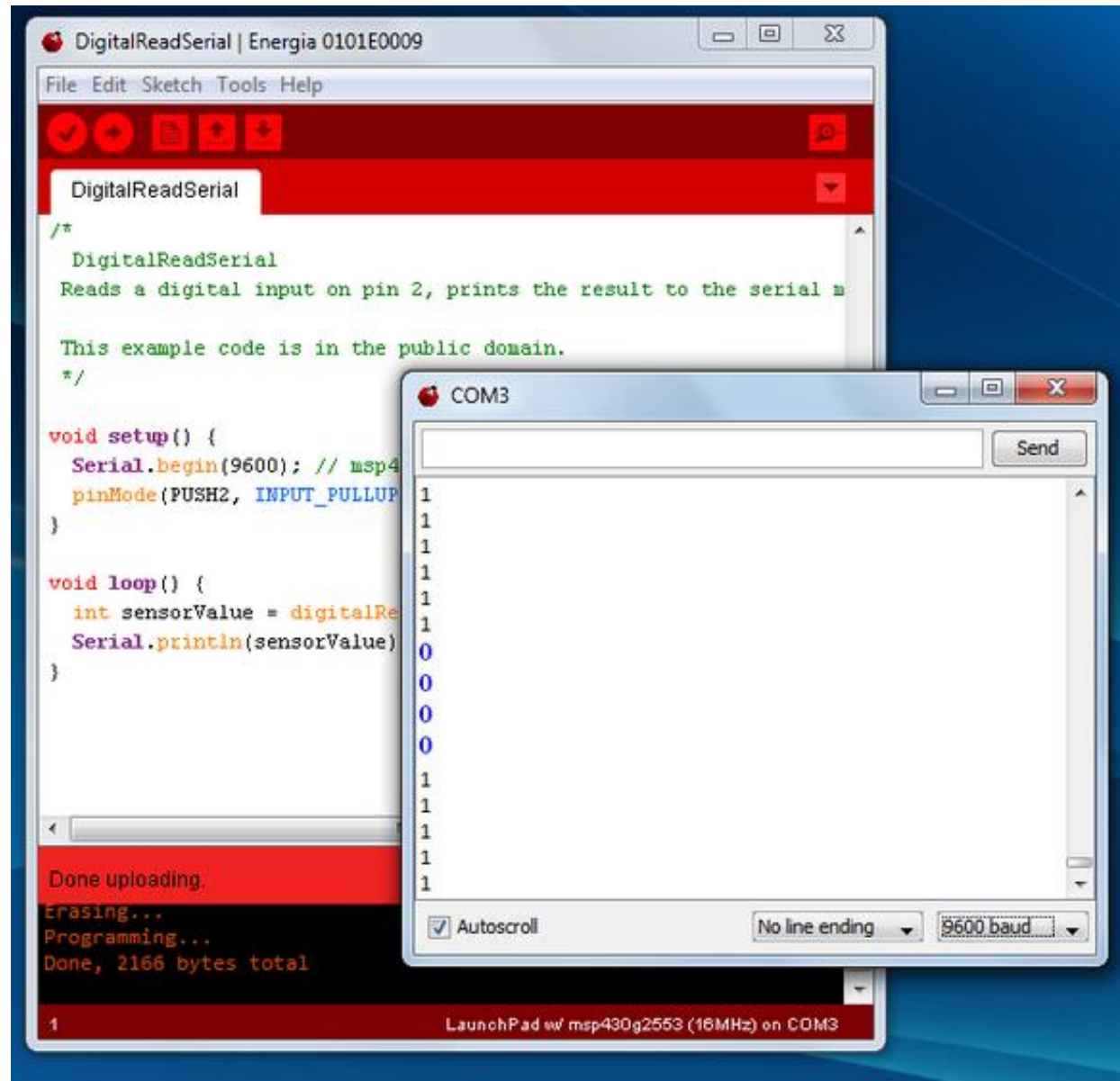
Nyomógomb  
állapotának  
beolvasása és  
soros porton  
keresztüli kiírása

Új függvények:

Serial.begin()

Serial.print()

Serial.println()



# Kérdések és feladatok (5)

## DigitalReadSerial program

4. Látja a számokat a monitoron? \_\_\_\_\_ Igen, ha...

Ha nem lát semmit a Serial Monitor ablakban, akkor mit kell tennie?

**Be kell állítani a soros porti átkötéseket!**

---



# DigitalReadSerial2 (módosítás)

```
/* DigitalReadSerial2
   Reads a digital input on SW2,
   prints the result to the serial monitor
   This example code is in the public domain.
*/

void setup() {
  Serial.begin(9600);
  pinMode(PUSH2, INPUT_PULLUP);
}

void loop() {
  int sensorValue = digitalRead(PUSH2);
  Serial.println(sensorValue);
  delay(500);
}
```

**Várakozás**

Lassítjuk a kiírásokat

# DigitalReadSerial3 (módosítás)

```
/* DigitalReadSerial3
   Reads a digital input on SW2,
   report changes to the serial monitor.
*/
int lastValue = 2;
void setup() {
  Serial.begin(9600);
  pinMode(PUSH2, INPUT_PULLUP);
}

void loop() {
  int sensorValue = digitalRead(PUSH2);
  if(sensorValue != lastValue) {
    lastValue = sensorValue;
    Serial.println(sensorValue);
  }
}
```

Előző állapot  
tárolója

Csak akkor van  
kijelzés, ha az  
új állapot  
különbözik az  
előzőtől!

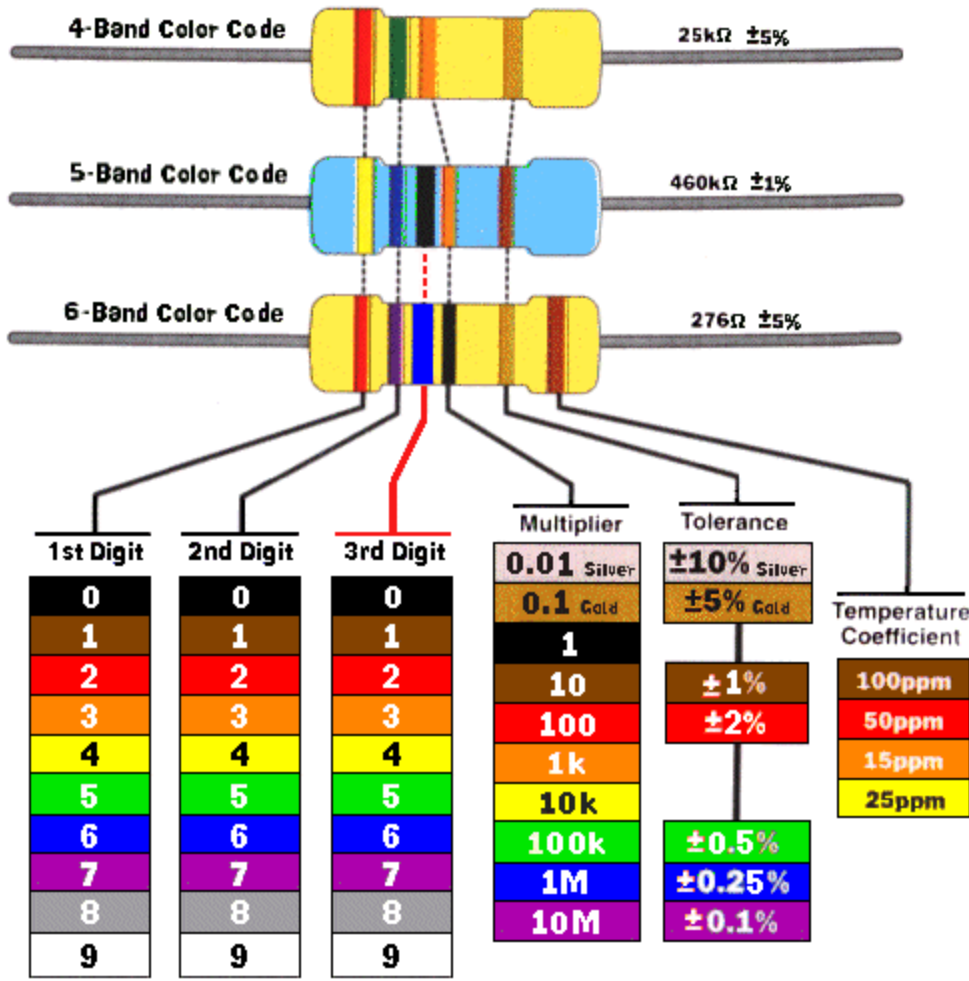


A következő lapokon azt tekintjük át röviden,  
hogy **milyen kiegészítő alkatrészekre** lehet  
szükségünk az első előadásban bemutatott,  
illetve a Hobbielektronika fórumon megjelent,  
az **Energiát** bemutató cikkemben ismertetett  
példaprojektekhez.

# Kiegészítő alkatrészek

## Ellenállások

A leggyakrabban használt értékek:  
470  $\Omega$ , 1 k $\Omega$ , 10 k $\Omega$ ,



## Vezetékek

Leggyakrabban:  
Female-female típus kell,

Max. 15 - 20 cm hosszú  
legyen

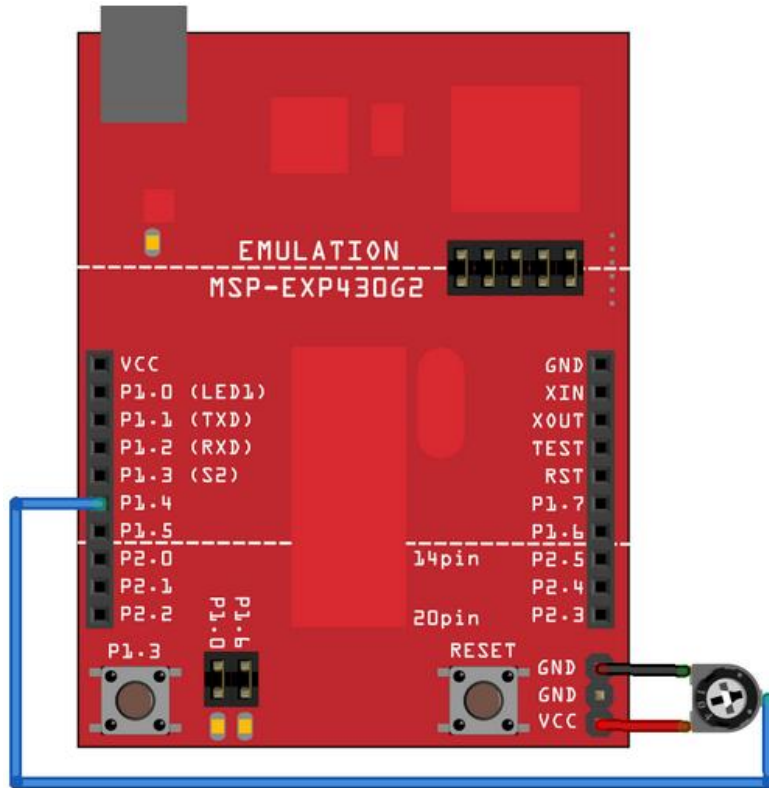


# Kiegészítő alkatrészek

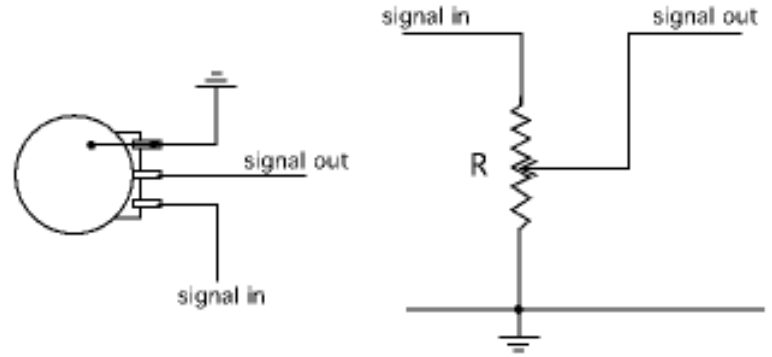
## Potenciométer

Változtatható feszültségosztó

5 k $\Omega$  – 100 k $\Omega$  közötti értékű legyen



Példa: feszültségmérés



# Kiegészítő alkatrészek

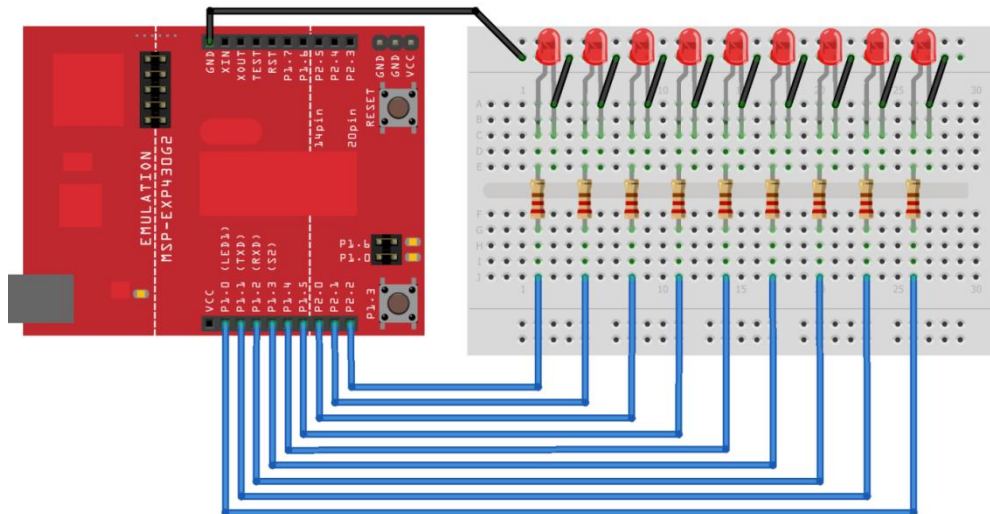
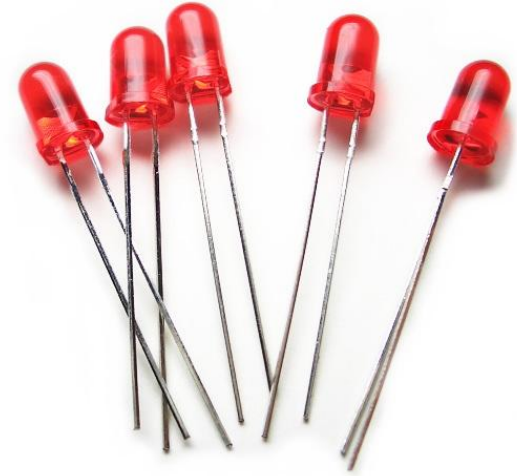
## LED-ek

Tipikusan 5 vagy 3 mm-esek

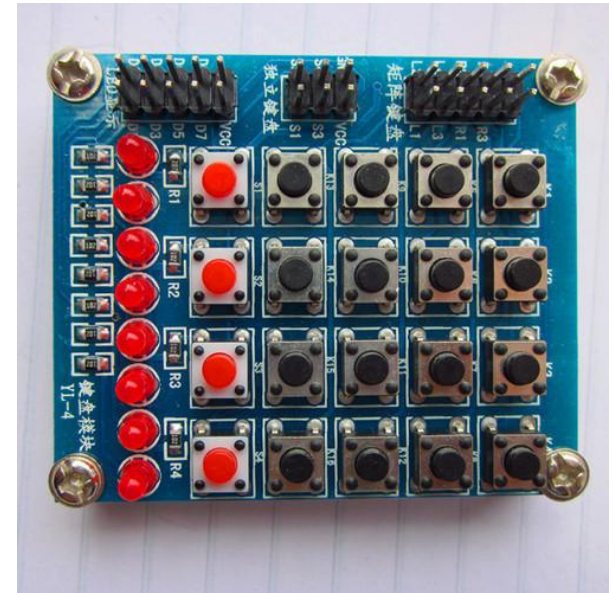
A hosszabb láb az anód (a pozitívabb)

Mindig áramkorlátozó ellenállással

kössük be!



Made with  Fritzing.org



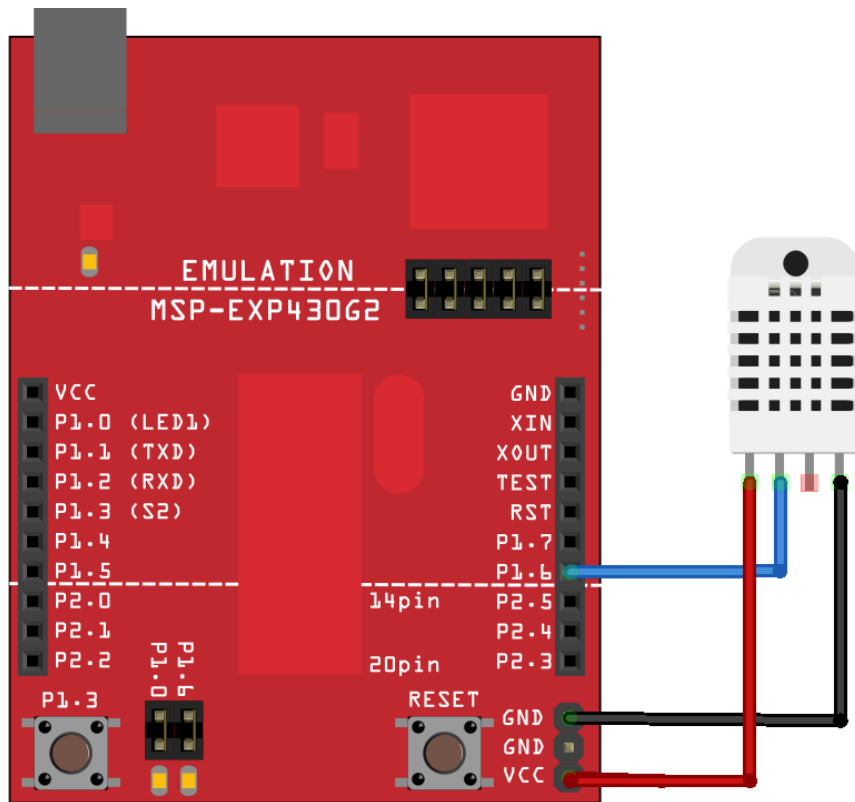
Példa: Knight Rider villogó

Alternatív megoldás

# Kiegészítő alkatrészek

AN2302 (DHT22) szenzor

Hőmérséklet és relatív páratartalom mérése



Made with  Fritzing.org

Minta kapcsolás

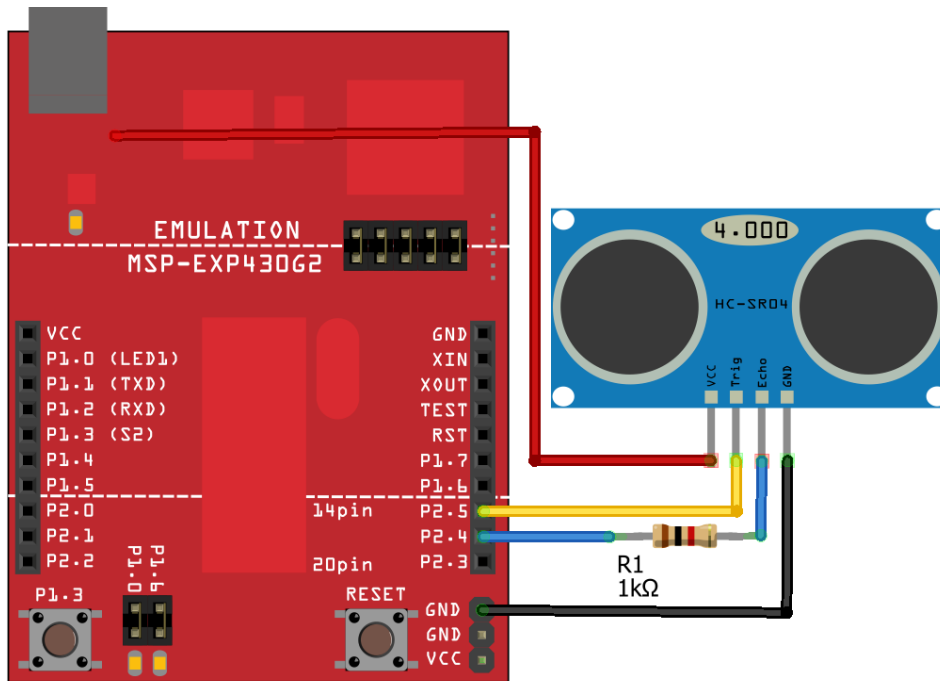
# Kiegészítő alkatrészek

HC-SR04 szenzor

Ultrahangos távolságmérés

2 cm – 400 cm tartományban

5 V-os tápfeszültséget igényel!



Made with  Fritzing.org

Minta kapcsolás

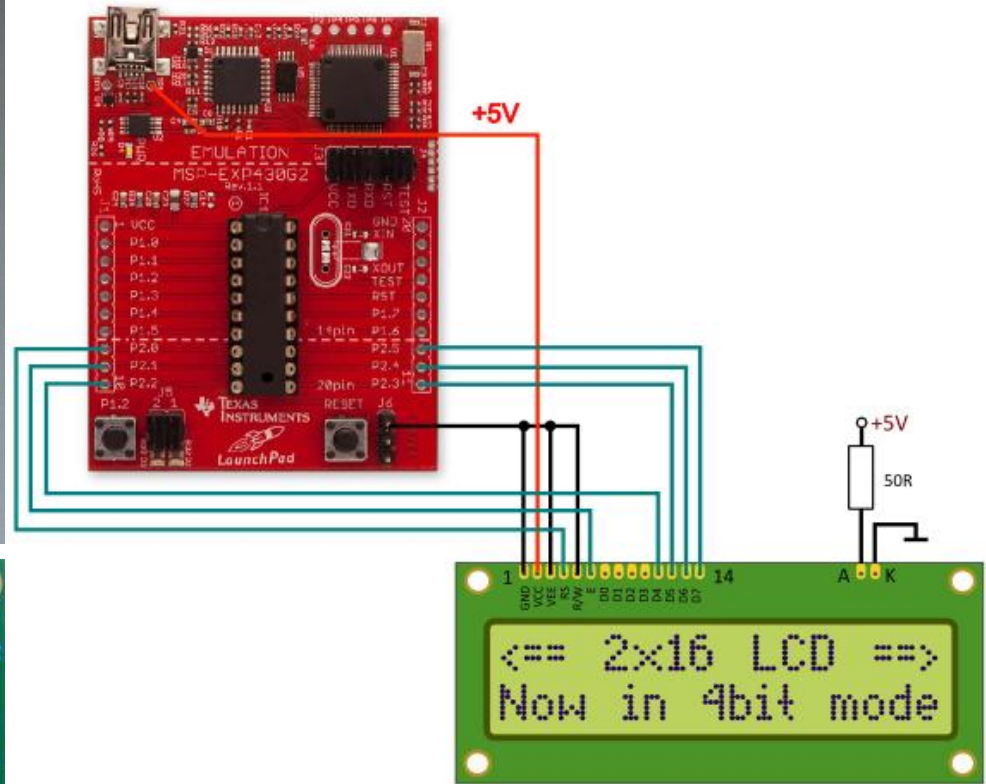


# Kiegészítő alkatrészek

Alfanumerikus kijelző

Ipari szabványnak tekinthető.

Van 5V-os és 3,3V-os változat!



Minta kapcsolás

