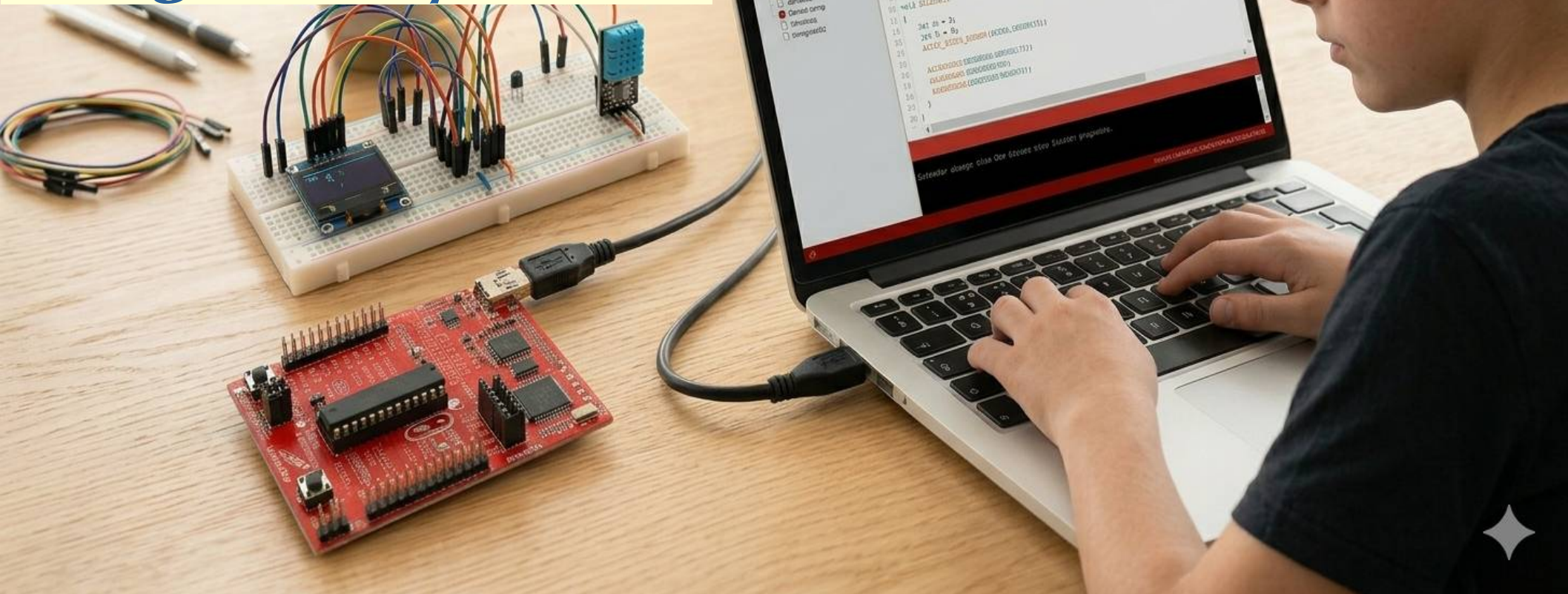
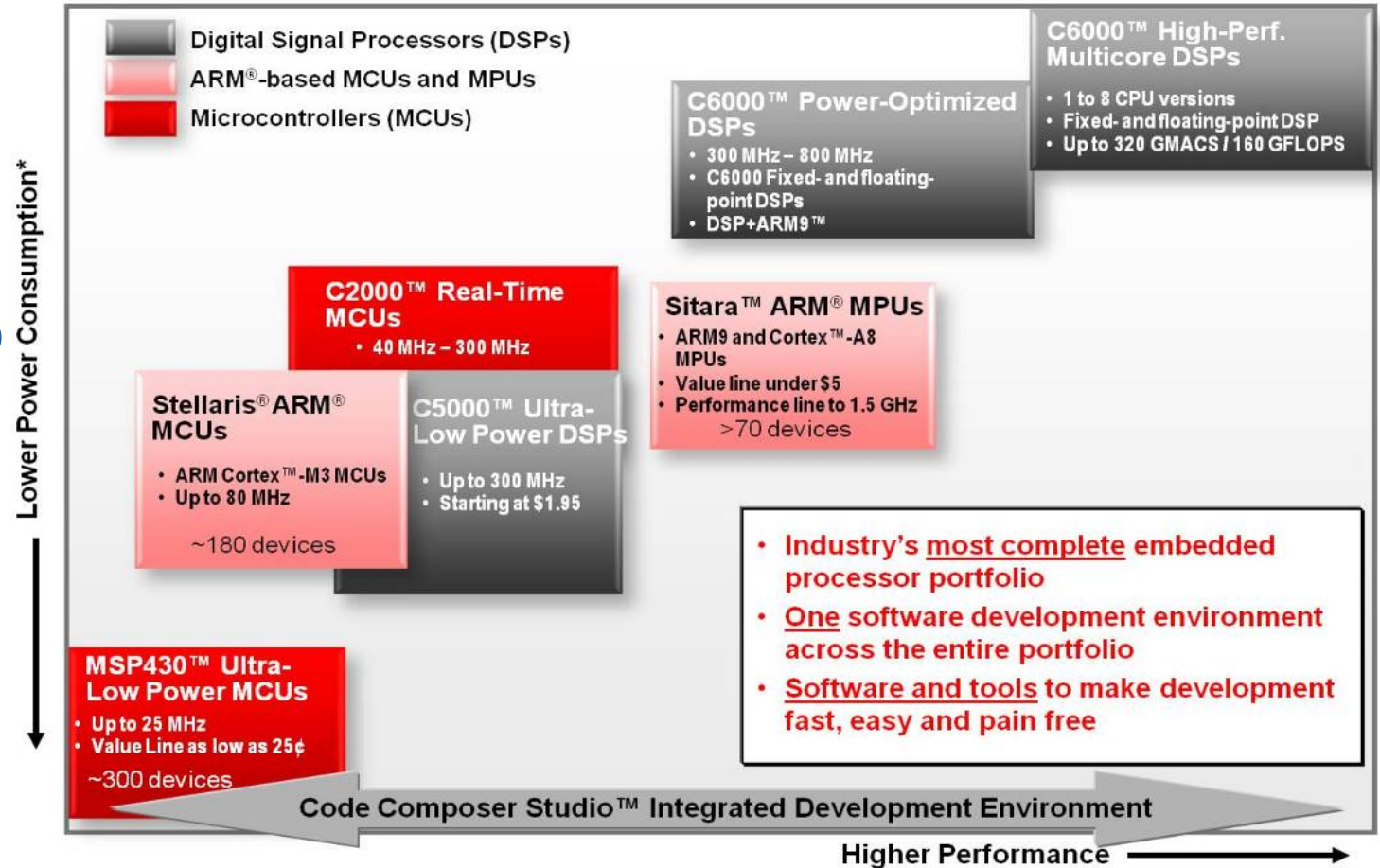


Texas mikrovezérlők programozása Energia környezetben



Ez már történelem!

- ❖ A Texas Instruments kb, 15 évvel ezelőtti mikrovezérlő és mikroprocesszor portfólióját mutatja
- ❖ Bennünket az **MSP430** és a **Stellaris** (későbbi nevén **Tiva C**) MCU család érdekel



MSP430



- > MSP430 MCUs are optimized for ultra-low-power & general purpose applications
- > Up to 25MHz and 128KB Flash
- > These MCUs feature flexible low power modes to enable battery-operated or battery-free applications

C2000



- > C2000 MCUs are optimized for real-time digital control applications
- > Up to 60MHz and 64KB Flash
- > These MCUs feature unique InstaSPIN libraries for efficient motor control

Tiva C Series

(korábban *Stellaris*)



- > Tiva C Series MCUs are based on the ARM® Cortex™-M architecture
- > Optimized for connected and general-purpose MCU applications
- > Up to 80 MHz and 256 KB flash memory
- > Feature a highly integrated platform of communication peripherals for a wide range of connected applications

Hercules



- > Hercules Safety MCUs are ARM Cortex-based and are optimized for safety-critical applications
- > Up to 100 MHz, 384KB Flash with ECC and 32KB RAM with ECC
- > These MCUs features dual lock-step ARM Cortex cores and integrated safety features for safety-critical applications

Olcsó fejlesztőeszközök


Szoftverfejlesztői környezetek

- ❖ **Texas Instruments Code Composer Studio (CCS):** Az Eclipse-alapú gyári fejlesztői környezet, amely az MSPWare szoftvercsomag révén kiterjedt támogatói könyvtárakat és hibakereső/nyomkövető eszközöket biztosít
- ❖ **IAR Embedded Workbench for MSP430:** Egy gyors, de rendkívül hatékony natív környezet, szűkebb szoftveres támogatással
- ❖ **Energia IDE:** Az Arduino ökoszisztémára épülő, nyílt forráskódú keretrendszer, amely egyszerűsített C++ függvénykönyvtáraival könnyű tanulhatóságot nyújt



Energia

- ❖ Az **Energia IDE** egy Arduino-szerű fejlesztőkörnyezet a TI LaunchPad sorozathoz
- ❖ A projektet 2012-ben indította Robert Wessels és Rei Vilo és a Texas Instruments is aktívan támogatta a projektet
- ❖ Támogatott platformok: MSP430, Stellaris LM4F120, TivaC TM4C123, MSP432, CC3200
- ❖ Egyszerű API: pinMode(), digitalWrite(), analogRead(), stb.
- ❖ Könnyű hozzáférés az MSP430 LPM0–LPM4 energiatakarékos módjaihoz: sleep(), sleepSeconds(), WDT-s ébresztés, órajel-skálázás (setClock())
- ❖ Gyors prototípus-készítés, oktatásban ideális
- ❖ Cross-platform: Windows, Linux, macOS
- ❖ 2019 után a projekt gyakorlatilag befagyott, de ma is hasznos régi LaunchPadekhez és oktatási demonstrációkhoz



The screenshot shows the Energia IDE interface. The title bar reads "Blink | Energia 1.8.11E23". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for running, uploading, and other functions. The main editor area displays the following code:

```
1 #define LED RED_LED
2
3 void setup() {
4   // initialize the digital pin as an output.
5   pinMode(LED, OUTPUT);
6 }
7
8 void loop() {
9   digitalWrite(LED, HIGH); // turn the LED on
10  delay(1000); // wait for a second
11  digitalWrite(LED, LOW); // turn the LED off
12  delay(1000); // wait for a second
13 }
```

The status bar at the bottom indicates the board and port: "MSP-EXP430G2 w/ MSP430G2452 on COM6".

Az Energia telepítése és konfigurálása

❖ Telepítés:

- Töltsük le az Energia 1.8.10E23 (12/17/2019) csomagot: <https://energia.nu/download/> és bontsuk ki (pl. a C:\Energia mappába!
- Az **MSP430** kártyákhoz töltsük le és telepítsük a MSP430 FET Drivers 1 0 1 1 meghajtót!

❖ Kritikus feltöltési hiba javítása:

- Nyisd meg a C:\Energia\hardware\energia\msp430\platform.txt állományt
- Keresd meg a **tools.mspdebug.upload.pattern** sort és a második felét írd át így:
`--force-reset \"prog \"{build.path}/{build.project_name}.hex\" \"`
- A **boards.txt** állományban lehet, hogy a mikrovezérlődnél át kell írni a protokollt, például: `MSP-EXP430G2553LP.upload.protocol=tilib`

❖ A Stellaris vagy Tiva C Launchpad kártyákhoz az Energia IDE Tools » Board manager menüjében telepítsük az **Energia TivaC Boards** bővítő csomagot

❖ A kártyához szükség lehet a https://www.ti.com/tool/STELLARIS_ICDI_DRIVERS telepítésére is

EK-MSP430G2 Launchpad MSP430G2553 mikrovezérlő



Az MSP430 termékcsalád

← Ultra-Low Power Performance		— Analog Integration		Easy-to-Use →								
MSP430™ 16-bit RISC CPU All devices feature: <ul style="list-style-type: none"> • 16-bit timers • Watchdog timer • Internal Digitally Controlled Oscillator • External crystal support • <50 nA pin leakage • <6 μs wakeup 	L092 0.9V-1.65V Speed: 4 MHz ROM: Up to 2Kb RAM: Up to 2Kb GPIO: 11	BOR	G2xxx Speed: 16 MHz Flash: 0.5-16Kb RAM: Up to 512b GPIO: 10-24	BOR	F4xx Speed: 8/16 MHz Flash: 4-120Kb RAM: Up to 8Kb GPIO: 14-80	BOR	F5xx/6xx Speed: 25 MHz Flash: 8-256Kb RAM: Up to 18Kb GPIO: 32-83	PMM	BOR	CC430 Speed: 20 MHz Flash: 8-16Kb RAM: Up to 4Kb GPIO: 40	PMM	BOR
		DAC8		ADC10		LCD			SVS			
FRAM Speed 24 MHz FRAM 4-16Kb GPIO 14-28 Non-volatile memory	F1xx Speed: 8 MHz Flash: 1-60Kb RAM: Up to 10Kb GPIO: 14-48	Comp	F2xx Speed: 16 MHz Flash: 1-120Kb RAM: Up to 8Kb GPIO: 10-64	ADC10,12	F5xx/6xx	SD16	PMM	SVS	CC430	SVM	PMM	SVS
		ADC10		Comp		SD16		SVS				
ADC12	Comp	DAC12	DAC12	DMA	F5xx/6xx	Comp	PMM	LDO	CC430	LDO	PMM	LDO
DMA	USART	MPY	OpAmp	MPY	F5xx/6xx	MPY	MPY	MPY	MPY	MPY	MPY	MPY
USCI	USART	SVS	OpAmp	OpAmp	F5xx/6xx	USART	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
USCI	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI	USCI	USCI	USCI	USCI	USCI	USCI
DMA	USART	USART	USCI	USCI	F5xx/6xx	USCI						

Az MSP430 G2xxx (Value Line) sorozat

Főbb jellemzők

CPU frekvencia: 0 - 16 MHz

Tápfeszültség: 1.8V – 3.6V

Flash: 16 kB-ig

RAM: 512 B-ig

kapacitív érzékelő támogatás

Ultra kismegőrzés

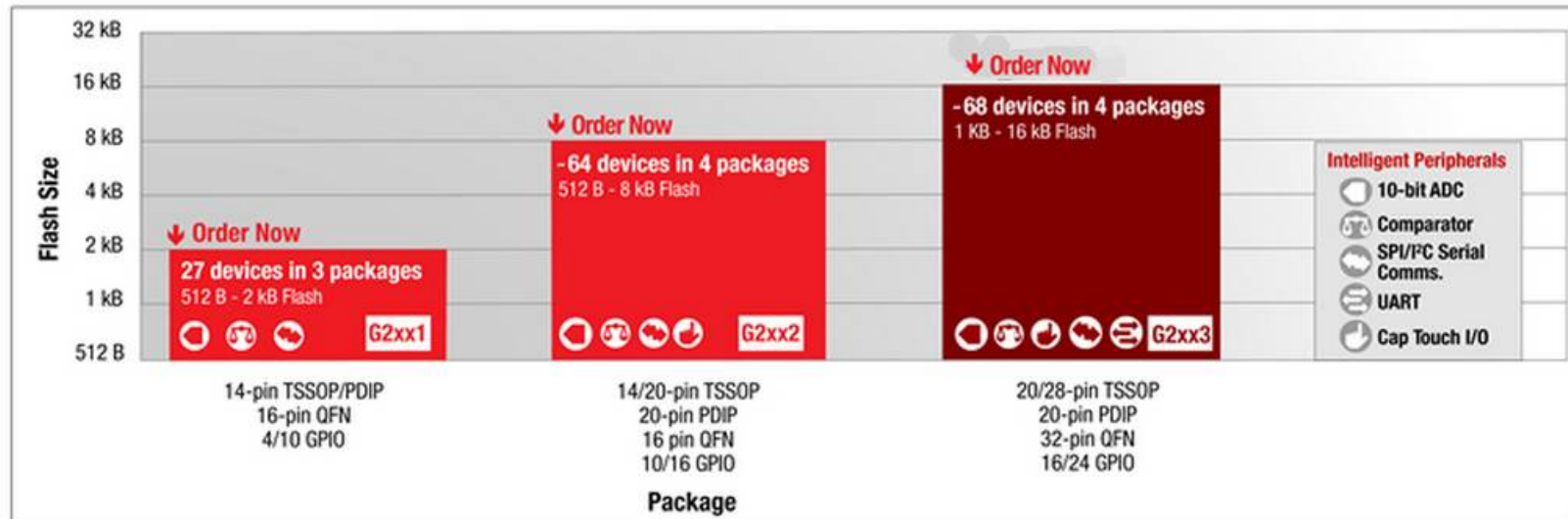
0.1 μ A RAM megőrzés

0.4 μ A Standby mód (VLO)

0.7 μ A valós idejű óra mód

220 μ A /MIPS aktív állapot

Ultragyors feléledés <1 μ s



Felhasznált és ajánlott irodalom

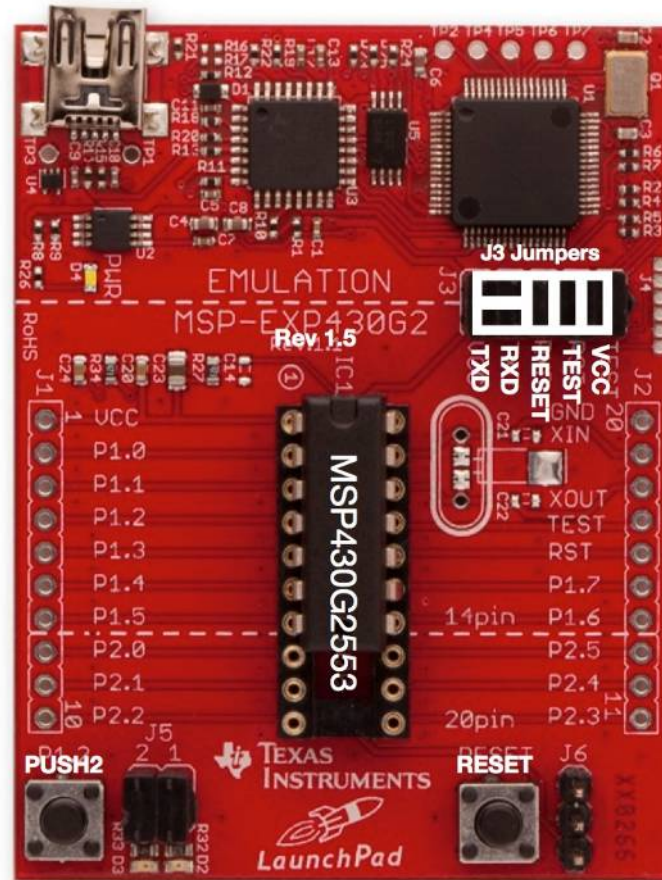
- ❖ [MSP-EXP430G2 LaunchPad Development Kit User's Guide \(Rev. G\)](#)
- ❖ [MSP430F2xx, MSP430G2xx Family User's Guide](#)
- ❖ [MSP430G2x13 and MSP430G2x53 Mixed Signal Microcontrollers Datasheet](#)
- ❖ [MSP430 FET Drivers 1 0 1 1](#)
- ❖ [Launchpad: ismerkedés az MSP430 mikrovezérlőkkel I.](#)
- ❖ [Launchpad: ismerkedés az MSP430 mikrovezérlőkkel II.](#)
- ❖ [Launchpad: ismerkedés az MSP430 mikrovezérlőkkel III.](#)
- ❖ [Launchpad: ismerkedés az MSP430 mikrovezérlőkkel IV.](#)
- ❖ [Tényleg nincs királyi út? I. rész \(Texas mikrovezérlők programozása Energiában\)](#)

LaunchPad with MSP430G2553

Revision 1.5

Flash 16 KB
Serial Hardware

+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



LEDs

Hardware
Pin number

I²C
Serial UART
SPI

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

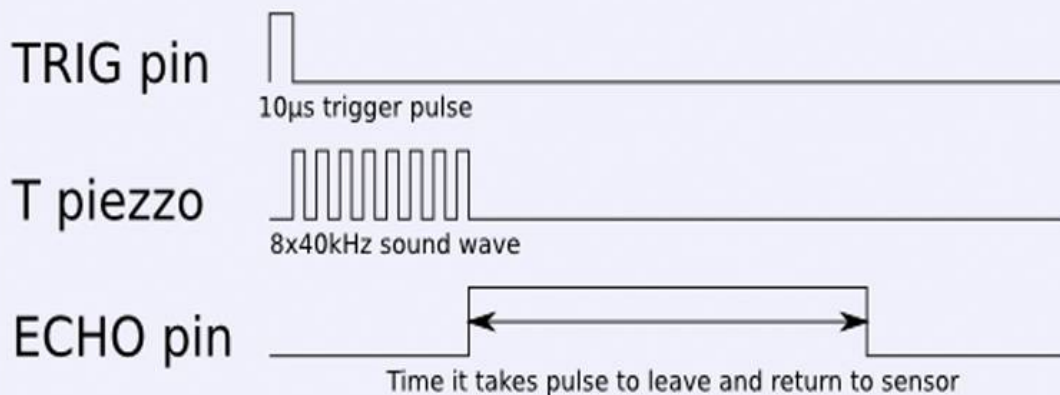
20					GROUND
19	P2_6				XIN
18	P2_7				XOUT
17					TEST
16					RESET
15	P1_7	A7	SDA	MOSI (B0)	
14	P1_6	A6	SCL	MISO (B0)	GREEN_LED
13	P2_5				
12	P2_4				
11	P2_3				

Ultrahangos távolságmérés

A **HC-SR04** modul piezo jeladója az indító impulzus hatására egy 40 kHz-es jelcsomagot sugároz ki. A modul digitális kimenő impulzusának szélessége megegyezik a visszaverődött hang terjedési idejével.



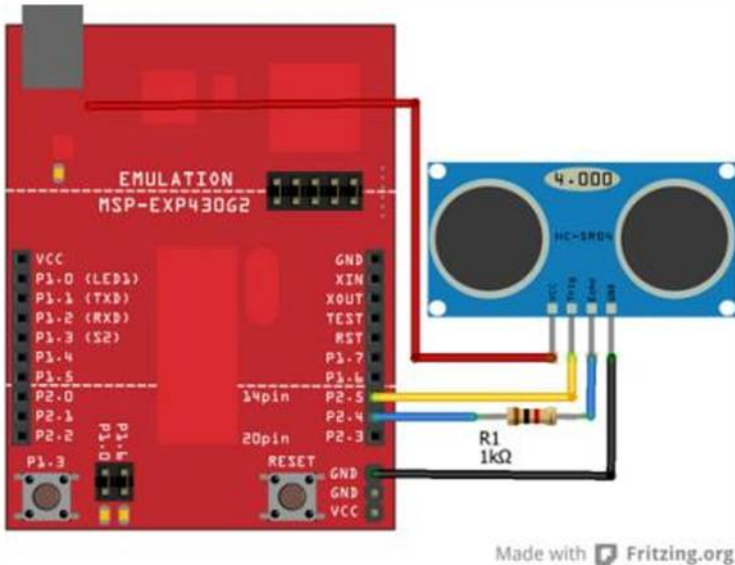
HC-SR04 Timing Chart



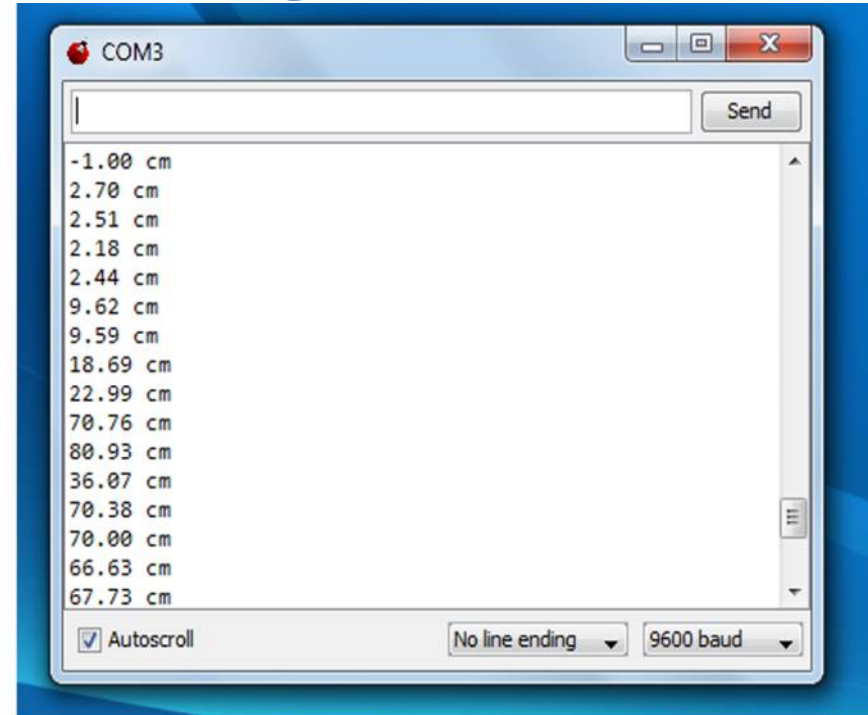
Főbb paraméterek

- Tápfeszültség: 4.5 V – 5.5 V
- Mérési tartomány: 2 cm – 4 m (gyakorlatban inkább 2 m)
- Érzékelési szögtartomány: $\sim 16^\circ$

Ultrahangos távolságmérés



Megjegyzés: az 5 V-os tápfeszültséget a Launchpad kártya TP1 pontjáról vehetjük



unsigned long **pulseIn**(int pin, int value)

Meghatározza a beérkező impulzus szélességét (mikroszekundum egységekben).

pin – a vizsgált bemenet sorszáma

value – az vizsgálandó impulzus polaritása (HIGH vagy LOW)

Sonar.ino – 2/1. oldal

```
#define echoPin P2_4           // Echo bemenet
#define trigPin P2_5          // Trigger kimenet

int maximumRange = 200;      // Legnagyobb távolság cm-ben
int minimumRange = 1;        // Minimális távolság cm-ben
long duration;               // Időtartam [us]
float distance;              // Távolság [cm]void

void setup() {
  Serial.begin(9600);         //Soros kapcsolat 9600 bit/s
  Serial.println("Sonar program");
  pinMode(echoPin, INPUT);    //Impulzus bemenet
  pinMode(trigPin, OUTPUT);   //Vezérlő kimenet
  digitalWrite(trigPin, LOW); //Alaphelyzetben alacsony szint
  pinMode(RED_LED, OUTPUT);   //A beépített piros LED jelző funkciót lát el
  digitalWrite(RED_LED, HIGH);
}
```

Sonar.ino – 2/2. oldal

```
void loop() {  
  delay(1000); //egy kis várakozás  
  digitalWrite(RED_LED, HIGH);  
  digitalWrite(trigPin, HIGH);  
  delayMicroseconds(10);  
  digitalWrite(trigPin, LOW);  
  duration = pulseIn(echoPin, HIGH); //Bejövő impulzus szélességének meghatározása  
  digitalWrite(RED_LED, LOW);  
  distance = duration/58.82; // Kiszámoljuk a távolságot  
  if (distance >= maximumRange || distance < minimumRange) {  
    distance = -1.0;  
  }  
  Serial.print(distance); // A mért távolság kiíratása  
  Serial.println(" cm");  
}
```

Trigger impulzus előállítása

$$d = \frac{t * v}{2}$$

Ahol d a távolság, t az impulzus hossza, v a hang terjedési sebessége (~340 m/s). Mivel t értéke μ s-ban adott, d -t pedig cm-ben mérjük, így $d = t * 34000/2000000$, azaz $d = t/58.82$

MSP-EXP430FR5739 Launchpad MSP430FR5739 mikrovezérlő



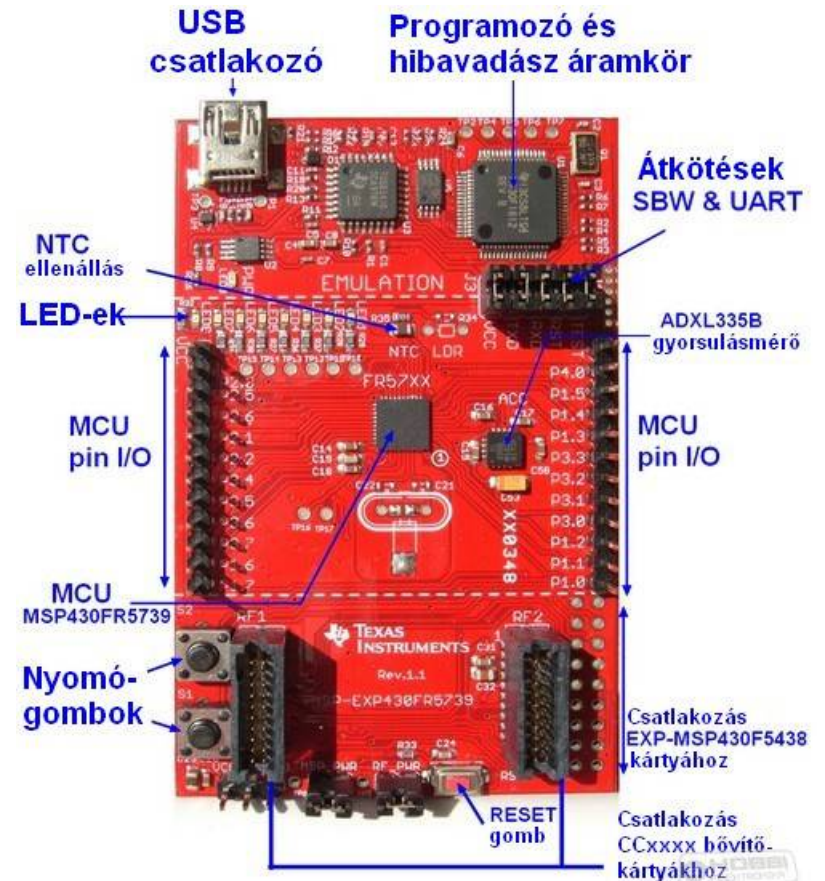
Az MSP430FR5937 (Fraunchpad) kártya

Az MSP-EXP430FR5739 fejlesztői kártya főbb jellemzői:

- ❖ MSP430FR5739 16-bites mikrovezérlő
- ❖ 16 kB FRAM és 1 kB RAM memória
- ❖ max. 24 MHz CPU frekvencia
- ❖ 2x Timer_A blokk, 3x Timer_B blokk
- ❖ 1x USCI (UART/SPI/IrDA/I2C) blokk
- ❖ 6 csatornás 10 bites ADC, 16 csat. analóg komparátor
- ❖ 32 I/O
- ❖ Programozó és hibavadász áramkör (MSP-FET)
- ❖ 8 db LED
- ❖ 3 db nyomógomb (RESET és 2x felhasználói gomb)
- ❖ ADXL335B háromtengelyű gyorsulásmérő
- ❖ NTC ellenállás (hőmérő)

Leírások, ismertetőik

- ❖ [MSP-EXP430FR5739 Experimenter Board User's Guide](#)
- ❖ [MSP430FR5739 Mixed-Signal Microcontrollers Data Sheet](#)
- ❖ [MSP430FR57xx Family User's Guide](#)
- ❖ [MSP430 DriverLib for MSP430FR5xx 6xx Devices](#)
- ❖ [Tényleg nincs királyi út? I. rész](#)



A portkivezetések elérhetősége

Az egyes portkivezetések foglaltságát és elérhetőségét (hol van kivezelve) az alábbi táblázatban foglaltam össze:

A port kivezetések foglaltsága/elérhetősége

Láb	Felhasználás	Merre	Láb	Felhasználás	Merre
P1.0		SV1-1	P2.0	UART RX	SV2-11
P1.1		SV1-2	P2.1	UART TX	SV2-8
P1.2		SV1-3	P2.2		SV2-7
P1.3		SV1-8	P2.3		RF3-10
P1.4	NTC (analóg bemenet)	SV1-9	P2.4		RF3-14
P1.5		SV1-10	P2.5		SV2-10
P1.6		SV2-2	P2.6		SV2-9
P1.7		SV2-1	P2.7	NTC/LDR tápfeszültség	RF3-9

Láb	Felhasználás	Merre	Láb	Felhasználás	Merre
P3.0	ADXL X kimenet	SV1-4	P4.0	S1 nyomógomb	SV1-11
P3.1	ADXL Y kimenet	SV1-5	P4.1	S2 nyomógomb	RF3-13
P3.2	ADXL Z kimenet	SV1-6			
P3.3	LDR	SV1-7			
P3.4	LED5	SV2-6	PJ.0	LED1	TP10
P3.5	LED6	SV2-5	PJ.1	LED2	TP11
P3.6	LED7	SV2-4	PJ.2	LED3	TP12
P3.7	LED8	SV2-3	PJ.3	LED4	TP13

SV1 és SV2 a 12 pólusú tűsoros csatlakozók a bővítőkétyákhoz.

RF3 pedig 2x9 pólusú csatlakozó az EXP-MSP430F5438 kártyához történő kapcsolódáshoz.

Experimenter Board with MSP430FR5739

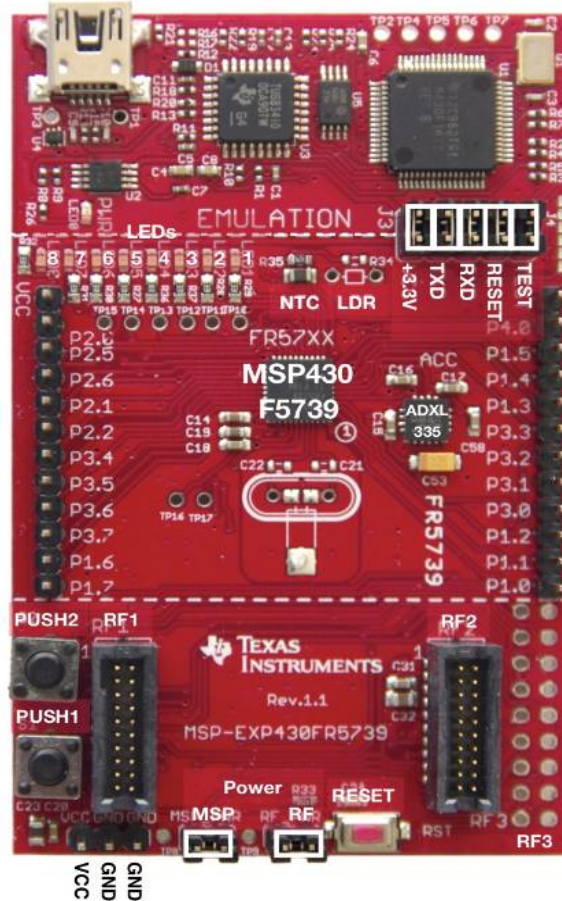
Revision 1.1

FRAM 16 KB
SRAM 1 KB

Serial	hardware
ADC	10 bits
Use pins numbers only!	
	PWM 2 or 23
	PWM 5 or 9
	PWM 6 or 10
	PWM 7 or 11
	PWM 8 or 12

+3.3V					1
	TXD			P2_0	2
				P2_5	3
				P2_6	4
	RXD			P2_1	5
		SCK (B0)		P2_2	6
LED5				P3_4	7
LED6				P3_5	8
LED7				P3_6	9
LED8				P3_7	10
		MOSI (B0)	SDA	P1_6	11
		MISO (B0)	SCL	P1_7	12

LED1	RED_LED			PJ_0	25
LED2				PJ_1	26
LED3				PJ_2	27
LED4	GREEN_LED			PJ_3	28



Hardware
Pin number
Other pin number

I ² C
Serial UART
SPI

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

24				GROUND
23	P4_0			PUSH1
22	P1_5	A0		
21	P1_4	A1		NTC
20	P1_3	A2	CS (B0)	
19	P3_3	A3		LDR
18	P3_2	A4		ACC_Z
17	P3_1	A5		ACC_Y
16	P3_0	A6		ACC_X
15	P1_2	A7		
14	P1_1	A8		
13	P1_0	A9		

29	P4_1			PUSH2
30	P2_7		NTC_ENABLE	ACC_ENABLE

Headers are not compatible with LaunchPad!

Futófény a LED soron

- ❖ A Fraunchpad különlegessége a 8 sorbarendezt LED. Itt egy tömböt használunk a lábak kezelésére

```
// Definiáljuk a LED-ek lábait egy tömbben a PinMap alapján
int leds[] = {P1_0, P1_1, P1_2, P1_3, P3_4, P3_5, P3_6, P3_7};

void setup() {
  for (int i = 0; i < 8; i++) {
    pinMode(leds[i], OUTPUT); // Minden LED lábát kimenetre állítunk
  }
}

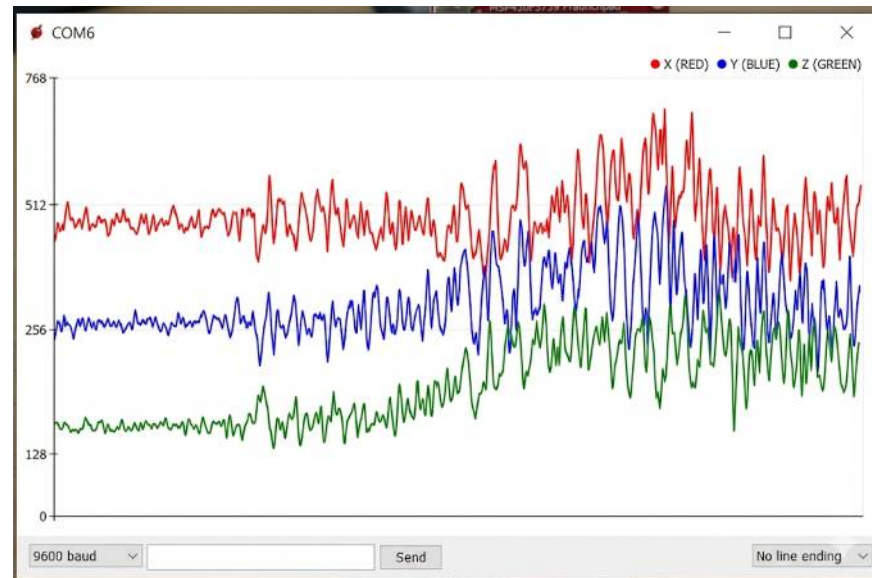
void loop() {
  for (int i = 0; i < 8; i++) {
    digitalWrite(leds[i], HIGH); // Aktuális LED be
    delay(100);                  // Rövid várakozás
    digitalWrite(leds[i], LOW);  // Aktuális LED ki
  }
}
```

Gyorsulásmérő és Serial Plotter

- ❖ Az ADXL335B háromtengelyű gyorsulásmérő analóg kimenetekkel rendelkezik, amelyeket a P3_0, P3_1, és P3_2 kivezetéseken érhetünk el. A gyorsulásmérő tápellátását a P2_7 kimenet magas szintre állításával kapcsolhatjuk be

```
void setup() {  
  Serial.begin(9600);  
  pinMode(P2_7, OUTPUT); // ACC_ENABLE  
  digitalWrite(P2_7, HIGH); // Gyorsulásmérő tápellátása  
}  
  
void loop() {  
  // A tengelyek kiolvasása a specifikus lábokról  
  int x = analogRead(P3_0); // ACC_X  
  int y = analogRead(P3_1); // ACC_Y  
  int z = analogRead(P3_2); // ACC_Z  
  
  // Plotter formátum: érték1, érték2, érték3  
  Serial.print(x); Serial.print(",");  
  Serial.print(y); Serial.print(",");  
  Serial.println(z); // Itt van a lezáró sortörés  
  delay(200);  
}
```

Az adatokat a Serial Plotter ablakban jeleníthetjük meg

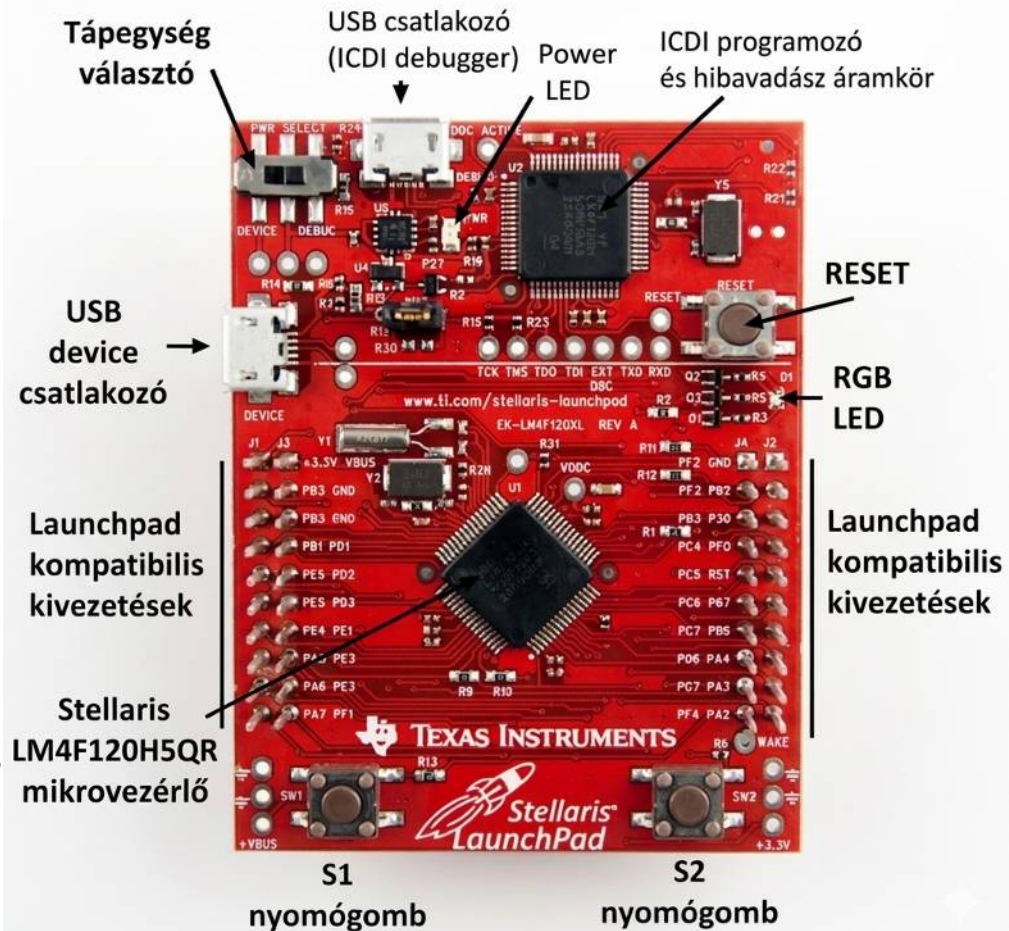


Stellaris és Tiva C Launchpad LM4F120/TM4C123 mikrovezérlő



Stellaris és Tiva C Launchpad kártyák

- ❖ A Texas Instruments **EK-LM4F120XL** kártyában egy 32 bites 80 MHz-es ARM Cortex-M4F magú **LM4F120H5QR** mikrovezérlő ketyeg, ami teljesítményben és bonyolultságban már meghaladja azt a szintet, amit kezdőnek szívesen ajánlana az ember. A teljesség kedvéért azonban erről a kártyáról is szólnunk kell, hiszen az Energia ezt a kártyát is támogatja
- ❖ Az **LM4F120H5QR** mikrovezérlő 256 KB flash programtároló memóriát, 32 KB RAM-ot és 2 KB EEPROM-ot tartalmaz. Két 12-bites ADC, egy USB 2.0 port és számos soros kommunikációs periféria (pl. UART, CAN, SPI, I2C)
- ❖ A kártya újabb kiadása **EK-TM4C123GXL** néven jelent meg, a **TM4C123GH6PM** MCU-val, ami az LM4F120 mikrovezérlő továbbfejlesztett változata (újdonság benne a USB Host/OTG funkció és egy speciális PWM periféria)

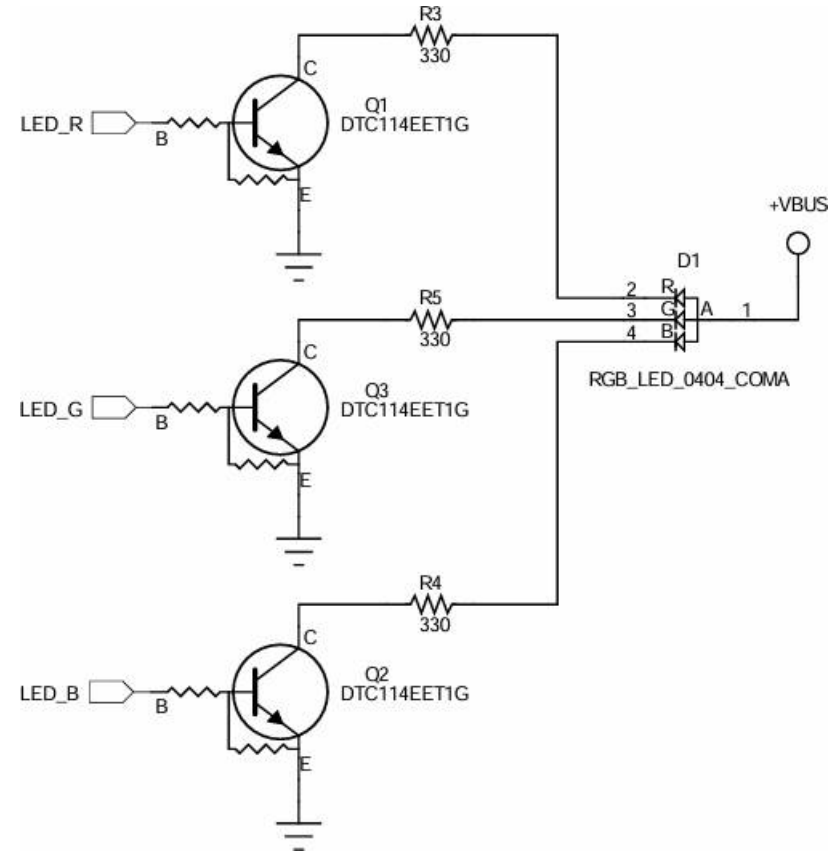


Adatlapok

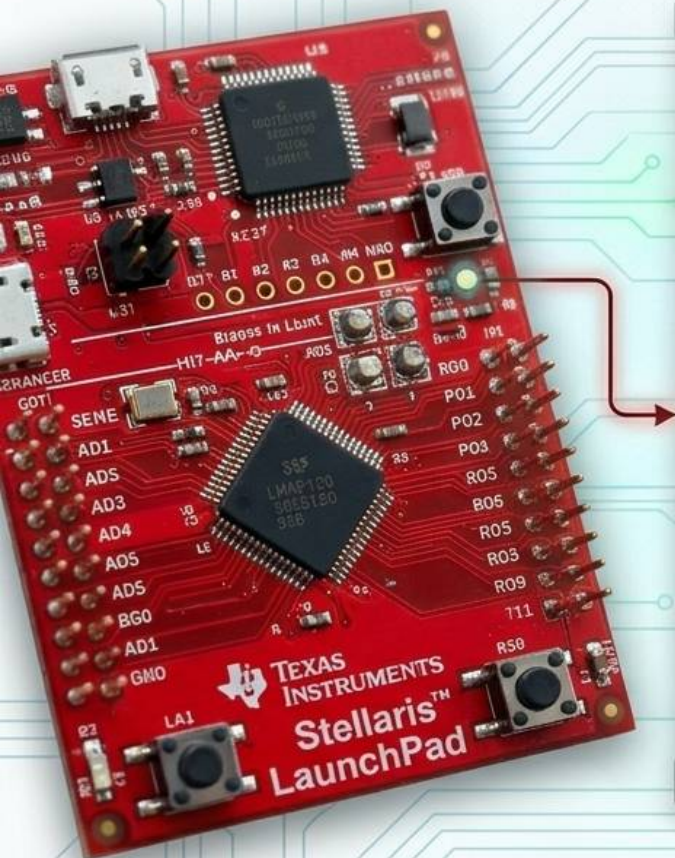
- ❖ [Stellaris LM4F120 LaunchPad Evaluation Board User Manual](#)
- ❖ [Tiva C Series TM4C123G LaunchPad Evaluation Board User's Guide](#)
- ❖ [Stellaris LM4F120H5QR Microcontroller Datasheet](#)
- ❖ [Tiva™ TM4C123GH6PM Microcontroller Datasheet](#)

Az RGB LED vezérlése

- ❖ A Stellaris Launchpad (EK-LM4F120XL) RGB LED-je három GPIO lábra van kötve: a piros (Red) LED a **PF_1**, a zöld (Green) LED a **PF_3**, a kék (Blue) LED pedig a **PF_2** lábon található. Az Energia környezetben ezeket az előre definiált **RED_LED**, **GREEN_LED** és **BLUE_LED** konstansokkal érhetjük el a legegyszerűbben
- ❖ A programban 8 alapszín (black, red, yellow, green, blue, purple, cyan, white) generálásához egy 0-tól 7-ig futó ciklust használunk, ahol a ciklusváltozó bitjei reprezentálják az egyes színek komponenseket (R-G-B)



rgb_ledblink.ino



```
/*
 * Stellaris LaunchPad (LM4F120) RGB LED Színzongora
 * Végigmegy a 8 alapszínen (000-111)
 */
// LED Pins: PF_1 (R), PF_2 (B), PF_3 (G)
void setup() {
  pinMode(RED_LED, OUTPUT);
  pinMode(GREEN_LED, OUTPUT);
  pinMode(BLUE_LED, OUTPUT);
}
void loop() {
  for (int i = 0; i < 8; i++) {
    // i determines color: bit 0(R), 1(G), 2(B)
    digitalWrite(RED_LED, (i & 0x01) ? HIGH : LOW);
    digitalWrite(GREEN_LED, (i & 0x02) ? HIGH : LOW);
    digitalWrite(BLUE_LED, (i & 0x04) ? HIGH : LOW);
    delay(1000); // 1 mp delay
  }
}
```

(000)		[0]
fekete	(Kikapcsolva)	
(001)		[1]
piros		
(010)		[2]
zöld		
(011)		[3]
sárga		
(100)		[4]
kék		
(101)		[5]
viola		
(110)		[6]
cián		
(111)		[7]
fehér		

rgb_control.ino 3/1.

- ❖ A beépített RGB LED vezérlése két nyomógommbal
 - **PUSH1 gomb:** A színárnyalat (HUE) folyamatos állítása 0–360° között (5 mp körülfordulással).
 - **PUSH2 gomb:** A fényerő szabályzása (fokozatosan fel, majd le)
 - **Algoritmus:** szektor-alapú HUE-RGB konverzió, a fényerő **brightness^{2.5}** exponenciális gamma-korrekcióval a természetesebb fényerőérzetért

```
float hue = 0.0;           // Induló szín: Piros
float brightness = 0.5;    // Induló fényerő: közepes
bool brightnessRising = true;
unsigned long lastUpdate = 0;

void setup() {
  pinMode(RED_LED, OUTPUT);
  pinMode(GREEN_LED, OUTPUT);
  pinMode(BLUE_LED, OUTPUT);

  // A gombok beépített felhúzó ellenállással (LOW = lenyomva)
  pinMode(PUSH1, INPUT_PULLUP);
  pinMode(PUSH2, INPUT_PULLUP);
}
```

rgb_control.ino 3/2.

```
void loop() {
  unsigned long currentMillis = millis();
  float deltaTime = (currentMillis - lastUpdate) / 1000.0;
  lastUpdate = currentMillis;

  // 1. Gombok kezelése (5 mp körülfordulási idő)
  // PUSH1 -> HUE változtatás
  if (digitalRead(PUSH1) == LOW) {
    hue += (360.0 / 5.0) * deltaTime; // 360 fok / 5 másodperc
    if (hue >= 360.0) hue -= 360.0;
  }
  // PUSH2 -> Fényerő változtatás (Fokozatosan fel, majd le)
  if (digitalRead(PUSH2) == LOW) {
    float step = (1.0 / 2.5) * deltaTime; // 2.5 mp fel, 2.5 mp le = 5 mp kör
    if (brightnessRising) { brightness += step;
      if (brightness >= 1.0) { brightness = 1.0; brightnessRising = false; }
    } else { brightness -= step;
      if (brightness <= 0.01) { brightness = 0.01; brightnessRising = true; }
    }
  }
  // 2. HUE konvertálása RGB-re (egyszerűsített algoritmus)
  setHue(hue, brightness); delay(10); // Kis pihenő a stabilitásért
}
```

rgb_control.ino 3/3.

```
void setHue(float h, float b) {
  float r, g, bl;
  // Színkomponensek kiszámítása a HUE alapján (0-360)
  float sector = h / 60.0;
  float x = 1.0 - abs(fmod(sector, 2.0) - 1.0);

  if (sector < 1)      { r = 1; g = x; bl = 0; }
  else if (sector < 2) { r = x; g = 1; bl = 0; }
  else if (sector < 3) { r = 0; g = 1; bl = x; }
  else if (sector < 4) { r = 0; g = x; bl = 1; }
  else if (sector < 5) { r = x; g = 0; bl = 1; }
  else                { r = 1; g = 0; bl = x; }

  // 3. Fényerő korrekció (Gamma-szerű korrekció: b^2.5)
  // Ez biztosítja, hogy a "közepes" érték valóban közepesnek tűnjön.
  float adjustedBrightness = pow(b, 2.5);

  // PWM jelek kiküldése (Energia: 0-255)
  analogWrite(RED_LED,    (int)(r * adjustedBrightness * 255));
  analogWrite(GREEN_LED, (int)(g * adjustedBrightness * 255));
  analogWrite(BLUE_LED,  (int)(bl * adjustedBrightness * 255));
}
```