

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

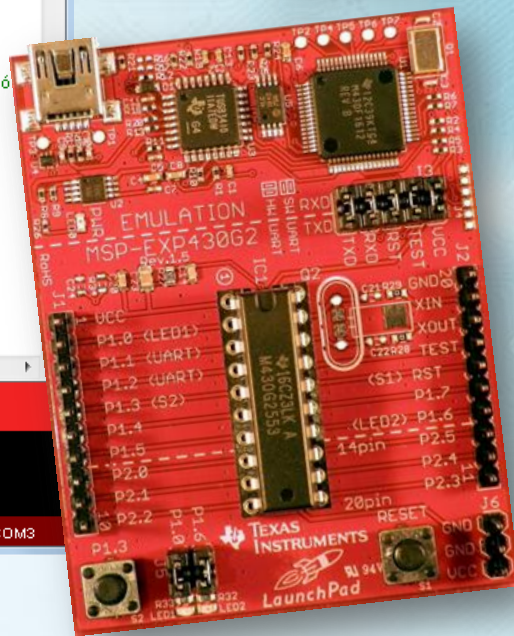
Ez a mintaprogram szabadon felhasználható (public domain).
*/

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 I/O portok kezelése



Egyszerű I/O vezérlés

Digitális I/O

- ✓ `pinMode(pin, mode)` – kivezetés üzemmódjának beállítása
- ✓ `digitalWrite(pin, state)` - kimenetvezérlés
- ✓ `digitalRead(pin)` – bemenet állapotának lekérdezése

Analóg I/O

- `analogReference(ref)` – ADC referenciájának megadása
- `analogRead(chan)` – analóg-digitális konverzió eredménye
- `analogWrite(pin)` - PWM teljesítményvezérlés

Referencia: DEFAULT (VCC), INTERNAL1V5, INTERNAL2V5 or EXTERNAL



Hőmérés a beépített szenzorral

```
/* TemperatureSensor: Hőmérés a beépített hőmérővel. A jobb feloldás  
érdekében a beépített 1.5 V-os referenciához képest mérünk, az  
eredményt Celsius fokokra átszámítjuk. A soros porton kiíratjuk az  
ADC-bőlkapott nyers adatot és az átszámított eredményt is. A mérések  
között egymásodperces szünetet tartunk. */
```

```
int val; //a konverzió eredménye  
float temp; //a hőmérséklet értéke
```

```
void setup() {  
  Serial.begin(9600); //Soros kapcsolat 9600 bit/s  
  analogReference(INTERNAL1V5); //1,5 V-os belső referencia  
  analogRead(TEMPSENSOR); //a beépített hőmérő kiválasztása  
}
```

```
void loop() {  
  delay(1000); //egy kis várakozás  
  val = analogRead(TEMPSENSOR); //a hőmérő leolvasása  
  Serial.print("adc:"); //a nyers adat kiírása  
  Serial.print(val);  
  temp = val*0.413-277.8; //a hőmérséklet kiszámítása  
  Serial.print(" temp:");  
  Serial.println(temp,1); //a hőmérséklet kiírása  
}
```



Hőmérés a beépített szenzorral

```
TemperatureSensor | Energia 0101E0009
File Edit Sketch Tools Help
TemperatureSensor$
int val;          //a konverzió eredménye
float temp;       //a hőmérséklet értéke

void setup() {
  Serial.begin(9600);
  analogReference(INTERNAL1V5); //1,5 V-os belső referencia
  analogRead(TEMPSENSOR);      //a beépített hőmérő kiválasztása
}

void loop() {
  delay(1000);                //egy kis várakozás
  val = analogRead(TEMPSENSOR); //a hőmérő leolvasása
  Serial.print("adc: ");      //a nyers adat kiírása
  Serial.print(val);
  temp = val*0.413-277.8;      //a hőmérséklet kiszámítása
  Serial.print("  temp:");
  Serial.print(temp,1);       //a hőmérséklet kiírása
  Serial.println(" C");
}

Done uploading.
Erasing...
Programming...
Done, 6542 bytes total

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

```
COM3
Send
adc: 748 temp:31.1 C
adc: 761 temp:36.5 C
adc: 756 temp:34.4 C
adc: 756 temp:34.4 C
adc: 758 temp:35.3 C
adc: 756 temp:34.4 C
adc: 756 temp:34.4 C
adc: 756 temp:34.4 C
adc: 750 temp:32.0 C
adc: 747 temp:30.7 C
adc: 747 temp:30.7 C

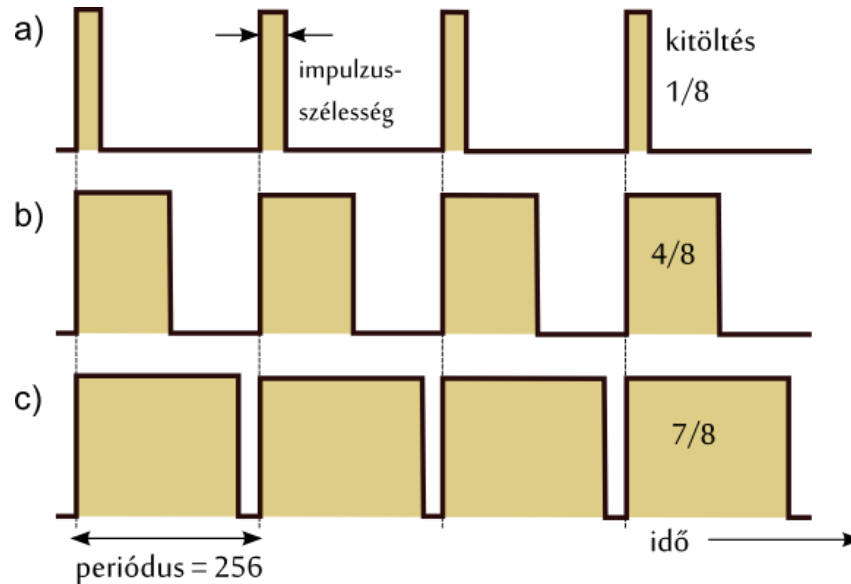
Autoscroll No line ending 9600 baud
```



Analóg vezérlés elve: PWM

PWM = pulse width modulation (impulzus-szélesség moduláció).

A frekvencia állandó,
a kitöltés változtatható 0-255 között.



Az **analogWrite()** függvény csak bizonyos lábakra vonatkozóan használható, de további megkötések is vannak.

Időzítő	Csatorna	Választható kivezetés
Timer0	TA0.1	P1_2, P1_6 vagy P2_6
Timer1	TA1.1	P2_1 vagy P2_2
Timer1	TA1.2	P2_4 vagy P2_5



Fade demóprogram

```
int brightness = 0;    // how bright the LED is
int fadeAmount = 5;   // how many points to fade the LED by

void setup() {
  pinMode(GREEN_LED, OUTPUT); // declare pin 14 to be an output
}

void loop() {
  analogWrite(GREEN_LED, brightness); // set the brightness of pin 14
  brightness = brightness + fadeAmount; // change brightness value

  // reverse the direction of the fading at the ends of the fade:
  if (brightness == 0 || brightness == 255) {
    fadeAmount = -fadeAmount ;
  }
  delay(30); // wait for 30 milliseconds to see the dimming effect
}
```



MSP430G2553 memóriatérkép

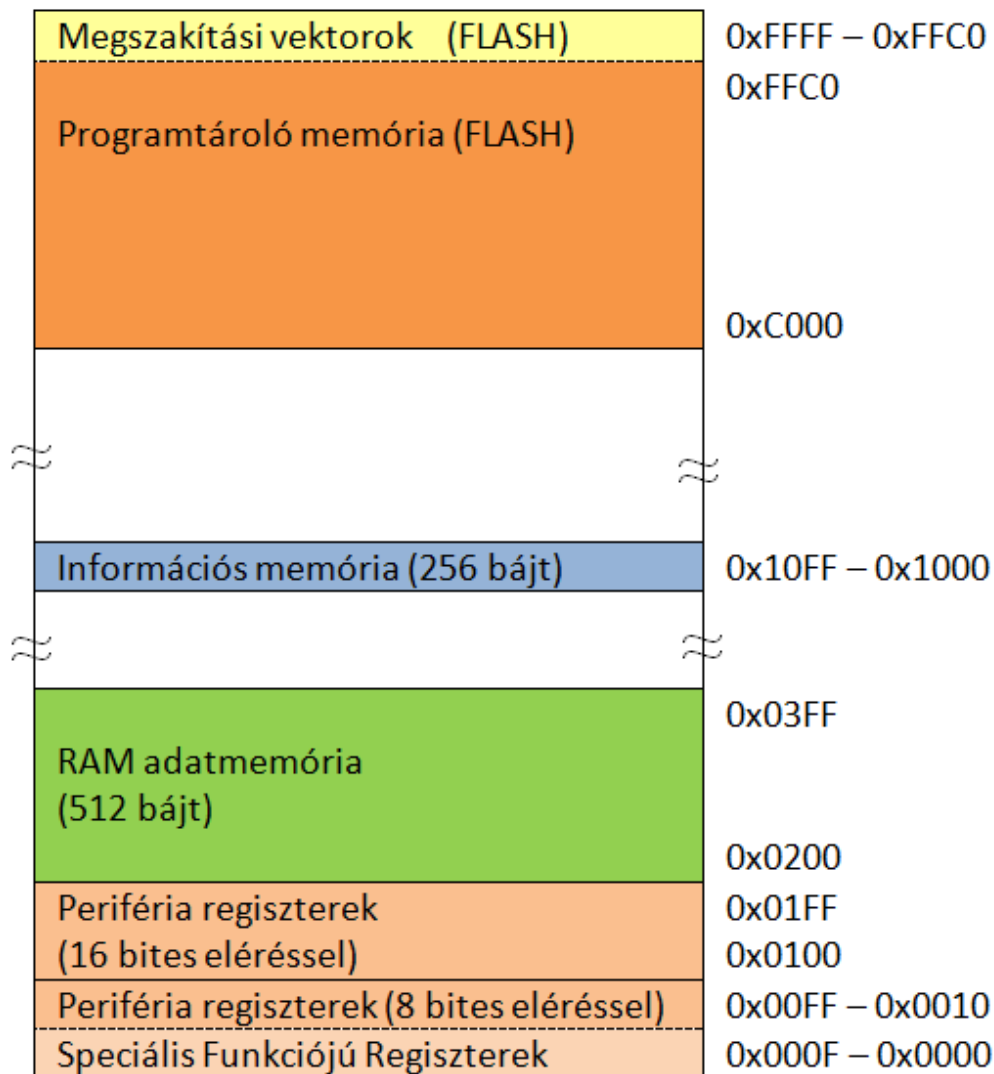
Egyetlen közös, címtartomány
(64 kB)

Flash memória: 16 kbájt, ennek utolsó 32 szava speciális szerepű (reset és megszakítási vektorok)

Információs memória: 256 bájt
Benne gyári kalibrációs adatok

RAM: 512 bájt (változók és veremtár)

A perifériák regiszterei is memóriacímekre vannak leképezve





Megszakítási vektorok

INTERRUPT SOURCE	INTERRUPT FLAG	SYSTEM INTERRUPT	WORD ADDRESS	PRIORITY
Power-Up External Reset Watchdog Timer+ Flash key violation PC out-of-range ⁽¹⁾	PORIFG RSTIFG WDTIFG KEYV ⁽²⁾	Reset	0FFFEh	31, highest
NMI Oscillator fault Flash memory access violation	NMIIFG OFIFG ACCVIFG ⁽²⁾⁽³⁾	(non)-maskable (non)-maskable (non)-maskable	0FFFCh	30
Timer1_A3	TA1CCR0 CCIFG ⁽⁴⁾	maskable	0FFFAh	29
Timer1_A3	TA1CCR2 TA1CCR1 CCIFG, TAIFG ⁽²⁾⁽⁴⁾	maskable	0FFF8h	28
Comparator_A+	CAIFG ⁽⁴⁾	maskable	0FFF6h	27
Watchdog Timer+	WDTIFG	maskable	0FFF4h	26
Timer0_A3	TA0CCR0 CCIFG ⁽⁴⁾	maskable	0FFF2h	25
Timer0_A3	TA0CCR2 TA0CCR1 CCIFG, TAIFG ⁽⁵⁾⁽⁴⁾	maskable	0FFF0h	24
USCI_A0/USCI_B0 receive USCI_B0 I2C status	UCA0RXIFG, UCB0RXIFG ⁽²⁾⁽⁵⁾	maskable	0FFEEh	23
USCI_A0/USCI_B0 transmit USCI_B0 I2C receive/transmit	UCA0TXIFG, UCB0TXIFG ⁽²⁾⁽⁶⁾	maskable	0FFECh	22
ADC10 (MSP430G2x53 only)	ADC10IFG ⁽⁴⁾	maskable	0FFEAh	21
			0FFE8h	20
I/O Port P2 (up to eight flags)	P2IFG.0 to P2IFG.7 ⁽²⁾⁽⁴⁾	maskable	0FFE6h	19
I/O Port P1 (up to eight flags)	P1IFG.0 to P1IFG.7 ⁽²⁾⁽⁴⁾	maskable	0FFE4h	18
			0FFE2h	17
			0FFE0h	16
See ⁽⁷⁾			0FFDEh	15
See ⁽⁸⁾			0FFDEh to 0FFC0h	14 to 0, lowest



A P1 port vezérlő regiszterei

A P2 porthoz is hasonló regiszterek tartoznak...

Regiszter	Cím	A regiszternév rövid magyarázata
P1IN	0x0020	PORT1 input (P1 bemenet)
P1OUT	0x0021	PORT1 output (P1 kimenet)
P1DIR	0x0022	PORT1 direction (P1 adatáramlási irány)
P1IFG	0x0023	PORT1 interrupt flags (P1 megszakítási kérelmet jelző bitek)
P1IES	0x0024	PORT1 edge select (P1 fel- vagy lefutóél választása megszakításhoz)
P1IE	0x0025	PORT1 interrupt enable (P1 programmegszakítás engedélyezés)
P1SEL	0x0026	PORT1 selection (P1 funkcióválasztó bitek)
P1REN	0x0027	PORT1 resistor enable (P1 fel-, vagy lehúzás engedélyezés)
P1SEL2	0x0041	PORT1 selection 2 (P1 további funkcióválasztó bitek)

I
N
T
E
R
R
U
P
T



LED villogtatás – a nehezebb módon

```
void setup() {  
  // P1.6 kimenetre állítása  
  P1DIR = 0x40; //BIT6  
  P1OUT = 0;  
  // watchdog letiltása  
  // WDTCTL = WDTPW | WDTHOLD;  
  // Órajel beállítása  
  // BCSCCTL1 = CALBC1_16MHZ;  
  // DCOCTL = CALDCO_16MHZ;  
}
```

Setup
kód

```
void setup() {  
  // Setup pin for output  
  pinMode (GREEN_LED, OUTPUT);  
  digitalWrite (GREEN_LED, LOW); // LED off  
}
```

```
Void loop() {  
  P1OUT = 0x40; // LED on  
  delay(1000); // wait 1 sec  
  P1OUT = 0; // LED off  
  delay(1000);  
}
```

Végtelen
ciklus

```
void loop() {  
  digitalWrite (GREEN_LED, HIGH); // LED on  
  delay (1000); // wait 1 second  
  digitalWrite (GREEN_LED, LOW); // LED off  
  delay (1000);  
}
```

Miért nem jó ez a program?



LED-ek villogtatása ellenütemben

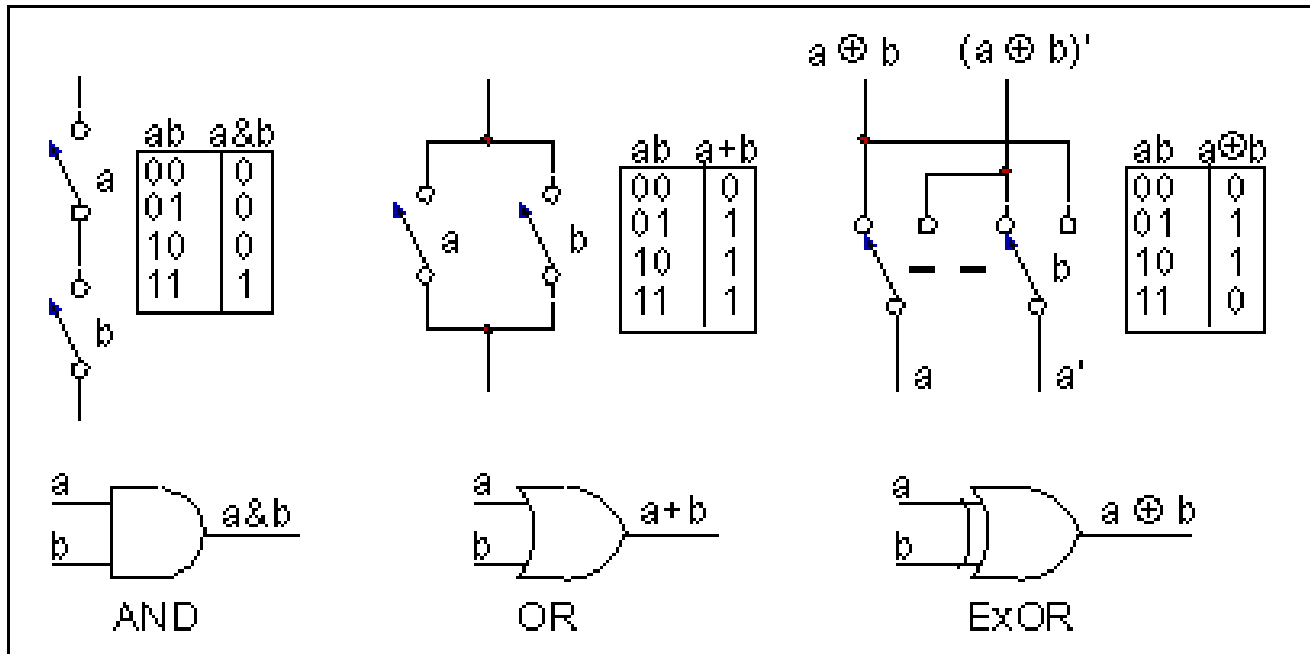
```
void setup() {  
    // P1.6 kimenetre állítása  
    P1DIR = 0x41;  
    P1OUT = 0x01;  
}  
  
void loop() {  
    P1OUT = 0x40; // LED be, LED ki  
    delay(1000); // wait 1 sec  
    P1OUT = 0x01; // LED ki, LED be  
    delay(1000);  
}
```

Miért nem jó ez a program?

```
void setup() {  
    // Setup pin for output  
    pinMode (GREEN_LED, OUTPUT);  
    pinMode (RED_LED, OUTPUT);  
    digitalWrite (GREEN_LED, LOW); // LED ki  
    digitalWrite (RED_LED, HIGH); // LED be  
}  
  
void loop() {  
    digitalWrite (GREEN_LED, HIGH); // LED be  
    digitalWrite (RED_LED, LOW); // LED ki  
    delay (1000); // wait 1 second  
    digitalWrite (GREEN_LED, LOW); // LED ki  
    digitalWrite (RED_LED, HIGH); // LED be  
    delay (1000);  
}
```



Bitműveletek a C nyelvben



$Y = a \& b;$ $Y = a | b;$ $Y = a \wedge b;$



Bit, vagy bitcsoport 1-be állítása

```
P1DIR |= 0x41;           // 0x41 = 0100_0001b
```

Jelentése: $P1DIR = P1DIR | 0x41;$

Elv: $x | 1 = 1$

Példa:

0 x 1 0	1 1 0 x	kiindulási érték
<u>0 1 0 0</u>	<u>0 0 0 1</u>	bitmaszk
0 1 1 0	1 1 0 1	eredmény



Bit, vagy bitcsoport 0-ba törlése

```
P1DIR &= ~0x41;           // ~0x41 = 1011_1110b
```

Jelentése: $P1DIR = P1DIR \& 0xDE;$

Elv: $x \& 0 = 0$

Példa:

0 x 1 0	1 1 0 x	kiindulási érték
<u>1 0 1 1</u>	<u>1 1 1 0</u>	bitmaszk
0 0 1 0	1 1 0 0	eredmény



Bit, vagy bitcsoport átbillentése

```
P1DIR ^= 0x41;           // 0x41 = 0100_0001b
```

Jelentése: $P1DIR = P1DIR \wedge 0x41;$

Elv: $x \wedge 1 = \sim x$ $x \wedge 0 = x$

Példa:

0	1	1	0	1	1	0	0	kiindulási érték
0	1	0	0	0	0	0	1	bitmaszk
<hr/>								
0	0	1	0	1	1	0	1	eredmény



LED-ek villogtatása korrekt módon

```
void setup() {  
    // P1.6 kimenetre állítása  
    P1DIR |= 0x41;    // BIT6 + BIT0  
    P1OUT |= 0x01;    // BIT0  
    P1OUT &= ~0x40;  // BIT6  
}  
  
void loop() {  
    P1OUT ^= 0x41;    // LED be, LED ki  
    delay(1000);      // wait 1 sec  
    P1OUT ^= 0x41;    // LED ki, LED be  
    delay(1000);  
}
```

Miért fölösleges a kihagyított rész?

```
void setup() {  
    // Setup pin for output  
    pinMode (GREEN_LED, OUTPUT);  
    pinMode (RED_LED, OUTPUT);  
    digitalWrite (GREEN_LED, LOW); // LED ki  
    digitalWrite (RED_LED, HIGH);  // LED be  
}  
  
void loop() {  
    digitalWrite (GREEN_LED, HIGH); // LED be  
    digitalWrite (RED_LED, LOW);    // LED ki  
    delay (1000);                    // wait 1 second  
    digitalWrite (GREEN_LED, LOW); // LED ki  
    digitalWrite (RED_LED, HIGH);   // LED be  
    delay (1000);  
}
```



Programmegszakítások

A főprogram megszakítja a tevékenységét és a megszakítást kiszolgáló függvény lefut.

Kapcsolódó függvények:

- `interrupts()` - megszakítások engedélyezése
- `noInterrupts()` - megszakítás tiltása
- `attachInterrupt()` - kiszolgáló fv. hozzárendelése megszakításhoz
- `detachInterrupt()` - kiszolgáló fv. hozzárendelése megszüntetése

`attachInterrupt(PUSH2, blink, FALLING);`

forrás függvény esemény (LOW, CHANGE, RISING, FALLING)



Egyszerű megszakítás használata

```
volatile int state = HIGH;
```

```
void setup() {  
    pinMode(RED_LED, OUTPUT);  
    pinMode(PUSH2, INPUT_PULLUP),  
    digitalWrite(RED_LED, state);  
    attachInterrupt(PUSH2, blink, FALLING); //Gomnyomáskor megszakítás  
}
```

```
void loop() {  
    LPM4; // ultra kisfogyasztású módba lép  
}
```

```
void blink() { // Megszakítás kiszolgálása  
    state = !state;  
    digitalWrite(RED_LED, state);  
}
```

Miért nem tökéletes a program?