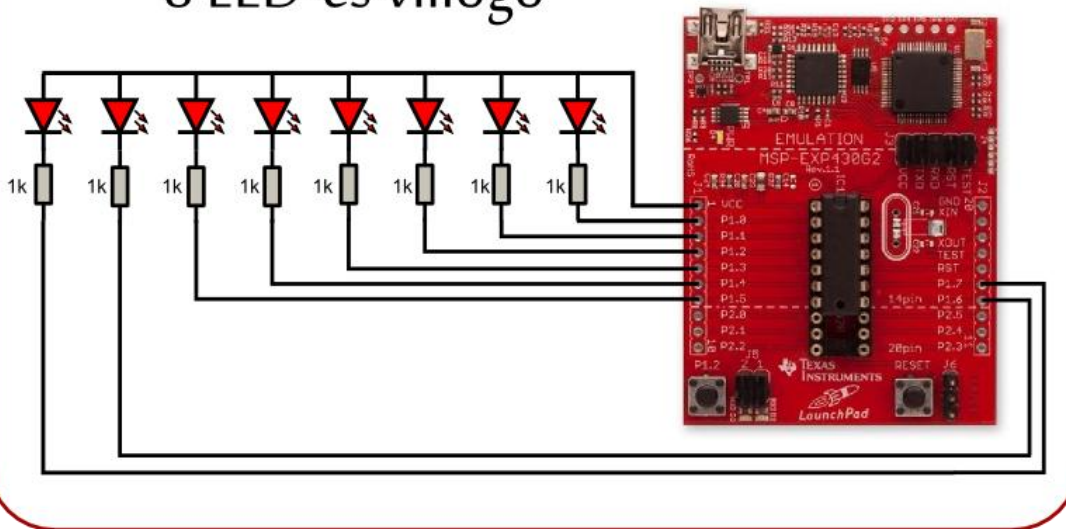


A Hobbielektronika csoport bemutatója

2013. május 20.

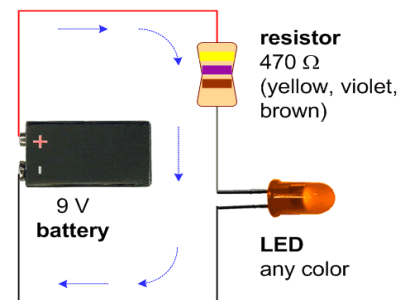
Debreceni Megtestesülés Plébánia

8 LED-es villogó



Ha a fényemittáló diódán (LED) nyitóirányú áram folyik, a dióda fényt bocsáj ki.

Az MSP430 Launchpad kártya mikrovezérlője a megírt program szerint tevékenykedik, egy táblázatban tárolt bitminták szerint kapcsolgatja ki és be a LED-eket. A bitminták között szünetet tartunk (késleltetés).



Knigh Rider villogó (C nyelvű program)

Fejlesztői környezet: IAR Embedded Workbench

```
#include "io430.h"
#include "intrinsics.h"
const char a[]={0x01,0x03,0x07,0x0E,0x1C,0x38,0x70,0xE0,0xC0,0x80};
char i;
void main(void) {
    WDTCTL = WDTPW + WDTHOLD;           //Letiltjuk a watchdog időzítőt
    P1DIR = 0xFF;                       //P1 minden bitje legyen digitális kimenet
    P1OUT = 0xFF;                       //kezdőállapot beállítása
    while(1) {
        for(i=0; i<10; i++) {
            P1OUT = ~a[i];              //bitminta kirakása a P1 portra
            __delay_cycles(100000);    //Várakozás kb. 50 ms ideig
        }
        i=10;
        while(i) {
            i--;                        //a ciklus addig fut, amíg i!=0 teljesül
            P1OUT = ~a[i];              //ciklusváltozó léptetése
            P1OUT = ~a[i];              //bitminta kirakása a P1 portra
            __delay_cycles(100000);    //Várakozás kb. 50 ms ideig
        }
    }
}
```

Speciális fényeffektus és Knight Rider villogó kombinálása (C nyelvű program)

```
#include "io430.h"
#include "intrinsics.h"
const char a[]={0x01,0x03,0x07,0x0E,0x1C,0x38,0x70,0xE0,0xC0,0x80};

void knight_rider(int n) {
int i,j;
P1OUT = 0xFF; //kezdőállapot beállítása
for(j=0; j<n; j++) {
for(i=0; i<10; i++) {
P1OUT = ~a[i]; //bitminta kirakása a P1 portra
__delay_cycles(100000); //Várakozás kb. 50 ms ideig
}
i=10;
while(i) { //a ciklus addig fut, amíg i!=0 teljesül
i--; //ciklusváltozó léptetése
P1OUT = ~a[i]; //bitminta kirakása a P1 portra
__delay_cycles(100000); //Várakozás kb. 50 ms ideig
}
}
}

void startup_sequence(void) {
unsigned char flag=4,up=1,counter = 0;
unsigned char LED_Array[] = {0x81,0x42,0x24,0x18 };
while (counter <12) {
counter++;
P1OUT = 0xFF; //kezdőállapot
if(up) {
while(flag) {
P1OUT = ~LED_Array[flag-1];
__delay_cycles(31000);
flag--;
}
up=0;
}
else {
while(flag<4) {
P1OUT = ~LED_Array[flag];
__delay_cycles(31000);
flag++;
}
up = 1;
}
}
P1OUT = 0xFF; //kilépés, LED-ek törlése
}

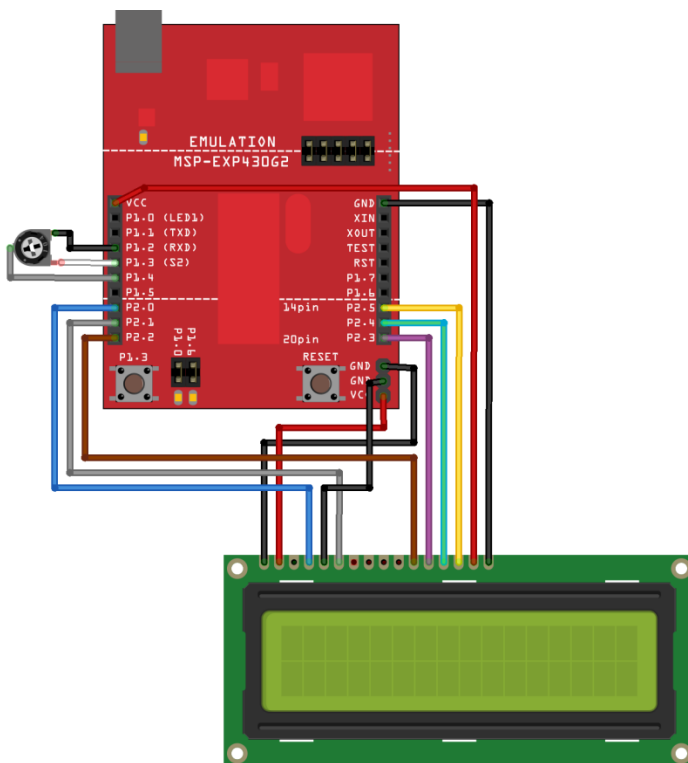
void main(void) {
WDTCTL = WDTPW + WDTHOLD; //Letiltjuk a watchdog időzítőt
P1DIR = 0xFF; //P1 minden bitje legyen digitális kimenet
P1OUT = 0xFF; //kezdőállapot beállítása
while(1) {
startup_sequence();
knight_rider(10);
__delay_cycles(400000);
}
}
```

Feszültségmérő LCD kijelzéssel

Az Energia IDE beépített könyvtárai segítségével egyszerűen kezelhetjük a perifériákat. Az **analogRead(A4)**; függvényhívással megmérjük az A4 bemenetre kapcsolt feszültséget (0 – 3.5 V közötti feszültséget kapcsolhatunk rá). Az eredményt voltokra számítjuk át, majd kijelezzük egy 16x2 karakteres LCD modul segítségével.

Az LCD kijelzőt a **LiquidCrystal** programkönyvtár segítségével kezeljük. A LiquidCrystal objektumosztály a Print osztály leszármazottja, így az LCD-re történő kiíratásra a print metódust használhatjuk.

A program bemutatásához egy, a tápfeszültség és a föld közé kötött potmétert használunk, s az A4 bemenetre a potméter csúszkáját kötjük.



Made with Fritzing.org

A program listája:

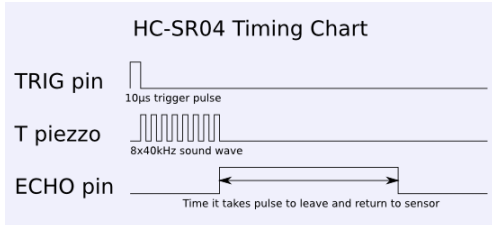
```
#include <LiquidCrystal.h> //Becsoljuk a programkönyvtárat
int i, k;
//Inicializáljuk a programkönyvtárat az alábbi lábkiosztással
LiquidCrystal lcd(P2_0, P2_1, P2_2, P2_3, P2_4, P2_5);

void setup() {
  lcd.begin(16, 2); //Beállítjuk a sorok és oszlopok számát)
  lcd.print("Analog voltmeter"); //Kiírunk egy szöveget az első sorba
  pinMode(P1_2,OUTPUT);
  digitalWrite(P1_2,LOW); //GND
  pinMode(P1_3,OUTPUT);
  digitalWrite(P1_3,HIGH); //VCC
}

void loop() {
  int sensorValue = analogRead(A4); //Mérés az A4 bemeneten
  // Átszámítjuk a 0 - 1023 közötti voltra (0 - 3.5V):
  float voltage = sensorValue * (3.5 / 1023.0);
  lcd.setCursor(0, 1); //Kurzor a második sor elejére
  lcd.print(voltage,3); //Kiíratjuk az eredményt
  lcd.print(" v");
  delay(2000);
}
```

Ultraszónos 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.

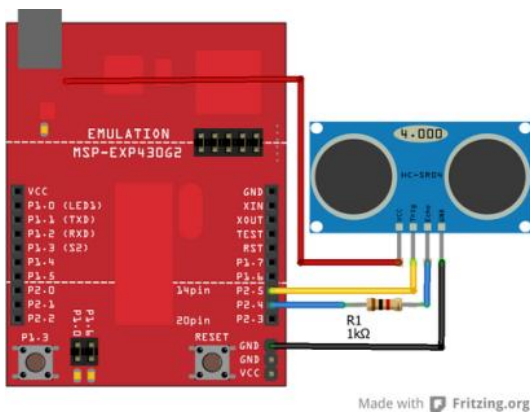


Főbb paraméterek

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



Kapcsolási vázlat



Program (Energia IDE)

```
Sonar | Energia 0101E009
File Edit Sketch Tools Help

Sonar §

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

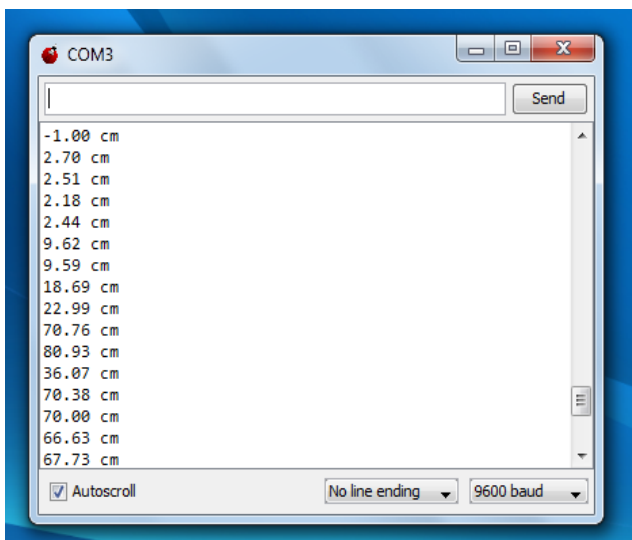
int maximumRange = 200; // Legnagyobb távolság cm-ben
int minimumRange = 2; // Minimális távolság cm-ben
long duration; // Időtartam [us]
float distance; // Távolság [cm]

void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  digitalWrite(trigPin, LOW);
}

void loop() {
  delay(1000);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration/58.2; // Kiszámoljuk a távolságot
  if (distance >= maximumRange || distance < minimumRange) {
    distance = -1.0;
  }
  Serial.print(distance); // A mért távolság kiírása
  Serial.println(" cm");
}

No changes necessary for Auto Format.
Erasing...
Programming...
Done, 6818 bytes total

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

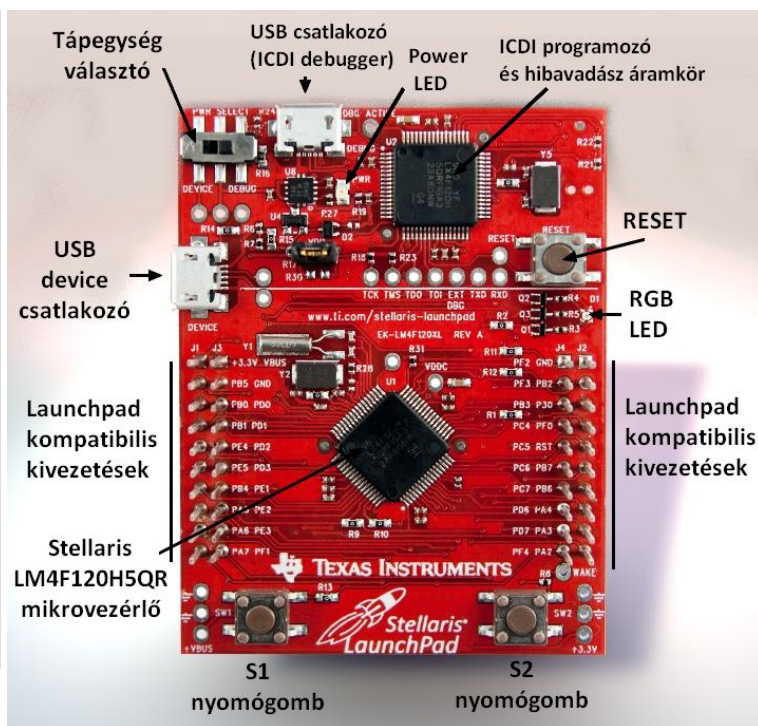
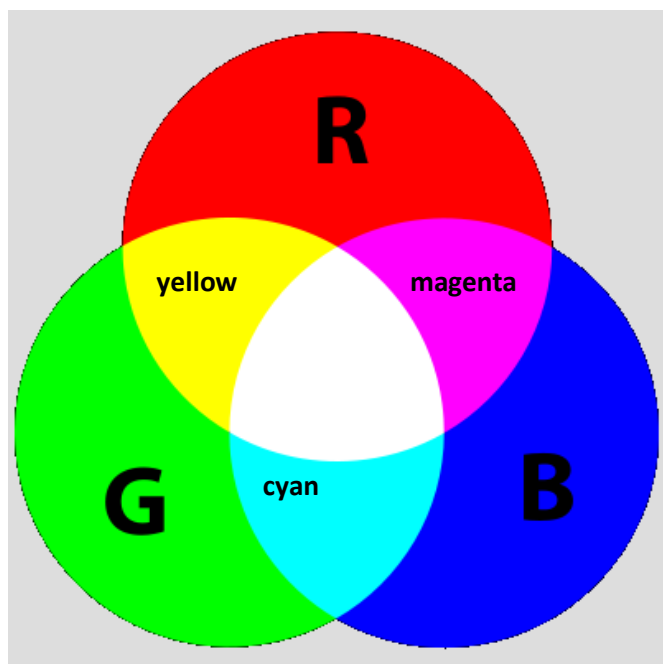


Alkalmazási lehetőségek

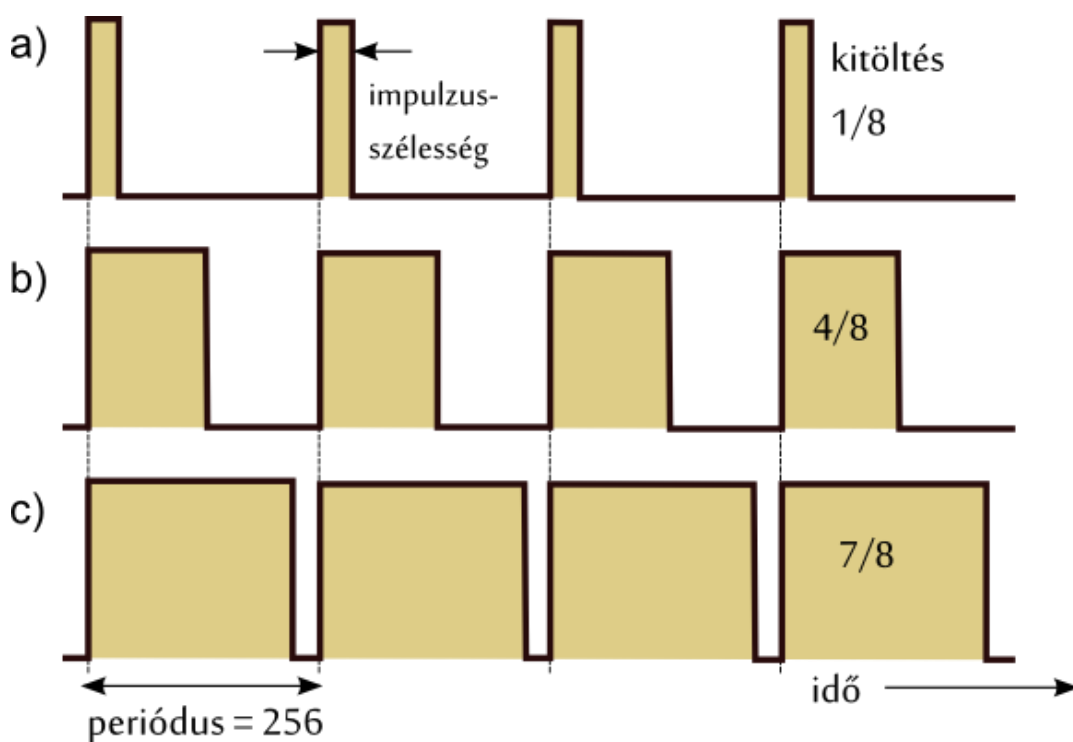
Távolságmérés, folyadékszint ellenőrzés, ütközésgátló, tolató radar

RGB LED vezérlése (folyamatos színátmenettel)

A Texas Instruments Stellaris Launchpad kártya tartalmaz egy RGB LED-et, amelyet a három színkomponens arányainak megválasztásával tetszés szerinti színre be tudjuk állítani (additív színkeverés).



A folyamatos színátmenet biztosításához analóg kimenetekkel kellene vezérelni a LED-eket. Ennek híján a kimenetek gyors fel- és lekapcsolgatásával és a kitöltési tényező változtatásával tudjuk a folyamatosság látszatát kelteni. Ennek a technikának impulzus-szélesség moduláció (angol nevének rövidítésével: PWM).



```

float RGB1[3], RGB2[3], INC[3], float SUM;
int red, green, blue;
void setup() {
//--- kiindulási színek generálása, véletlenszerűen
  randomSeed(analogRead(0));
  for (int x=0; x<3; x++) {
    RGB1[x] = random(256);
    RGB2[x] = random(256);
  }
}
void loop() {
//--- Az 1. színből a 2. színbe történő átmenet lépéseinek meghatározása
  for (int x=0; x<3; x++) {
    INC[x] = (RGB1[x] - RGB2[x]) / 256;
  }
  for (int s=0; s<256; s++) {
    red = int(RGB1[0]);
    green = int(RGB1[1]);
    blue = int(RGB1[2]);
//--- A közelítő szín kijelzése
    analogWrite (RED_LED, red);
    analogWrite (GREEN_LED, green);
    analogWrite (BLUE_LED, blue);
    delay(20);
    for (int x=0; x<3; x++) {
      RGB1[x] -= INC[x]; //közelítünk a másik színhez
    }
  }
do { //--- Új színt konstruálunk, véletlen számok generálásával
  SUM = 0;
  for (int x=0; x<3; x++) {
    RGB2[x] = random(956)-700;
    RGB2[x] = constrain(RGB2[x], 0, 255);
    SUM += RGB2[x];
  }
}
while(SUM < 300); //Nem engedjük túlságosan elsötétedni

```

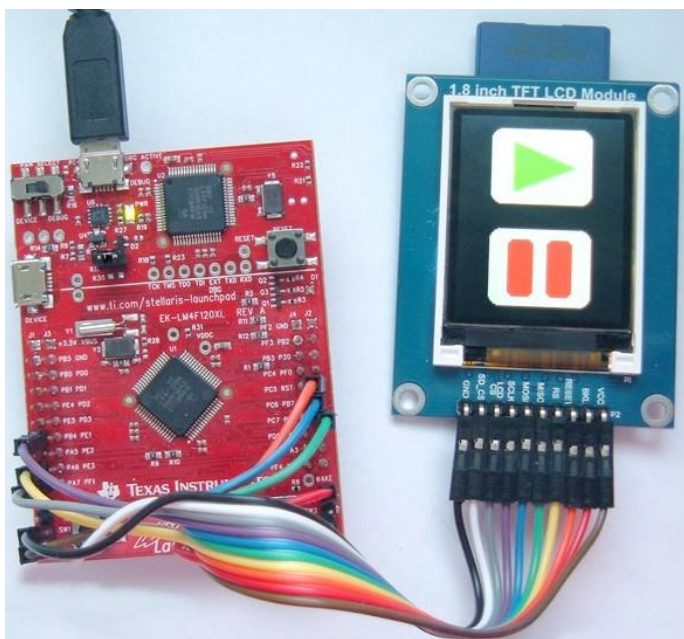

}

Színes 1.8" TFT képernyő vezérlése

Eredetileg az Arduino kártyához készült programkönyvtár kis módosítással más kártyához is felhasználható. A **graphicstest_highspeed** demó a képernyű grafikus képességeiből mutat ízelítőt. A program az Energia fejlesztői környezetben, a Stellaris Launchpad kártyára fordítható le.

Az ST7735R vezérlővel ellátott 1.8" méretű kijelző modul fontosabb paraméterei:

- Felbontás: 128x160
- Jelszint: 3.3 vagy 5 V
- Tápfeszültség: 5 V
- Illesztőfelület: SPI (szinkron soros)
- Periféria: SD kártya foglalat



HIVATKOZÁSOK

1.8" LCD kijelző:

http://www.ebay.com/sch/?_nkw=1.8%22%20LCD%20Arduino%20SD%20card

<http://www.adafruit.com/products/358>

Stellaris Launchpad kártya:

http://processors.wiki.ti.com/index.php/Getting_Started_with_the_Stellaris_EK-LM4F120XL_LaunchPad_Workshop

Adafruit ST7735 Library:

<https://github.com/adafruit/Adafruit-ST7735-Library>

Energia port of Adafruit_ST7735 Library:

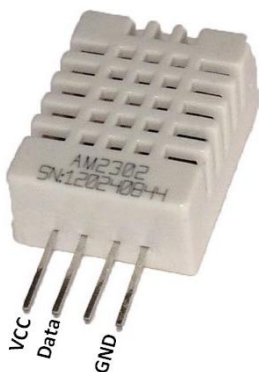
<http://forum.stellarisiti.com/topic/590-energia-library-port-of-adafruit-gfx-and-adafruit-st7735/>

Hőmérséklet és relatív páratartalom mérése

A méréshez az Aosong AM2302/DHT22 szenzort használjuk, melynek digitális kimenőjelét a Gadget Renesas gyártmányú, 32 bites RX63N mikrovezérlővel ellátott Sakura kártya fogadja és továbbítja a számítógépre USB kapcsolaton keresztül. Az adatokat a PC-n futó Processing/Java alkalmazás megjeleníti és kívánságra naplózza.

A SAKURA KÁRTYA FŐBB JELLEMZŐI

- Mikrovezérlő : RX63N(R5F563NBDDFP)
- Tápfeszültség : 3.3V
- Órajel frekvencia : 96MHz
- Digitális ki/bemenet : 55 db
- Analog bemenetek : 16 db
- Flash Memória: 1MB
- RAM : 128KB
- USB: eszköz vagy hoszt
- SD/MMC mikro kártyaolvasó
- Ethernet: 100 mbit (RJ-45)
- Arduino kompatibilis csatlakozók



Az AM2302 SENZOR FŐBB JELLEMZŐI

Felbontás: hőmérséklet 0.1 °C és rel. páratartalom 0.1 %

Kommunikáció: 1-wire, nem szabványos protokollal, 4 bájt adat + 1 bájt ellenőrző összeg.

Mintavételezési gyakoriság: 2 másodpercenként

Tápfeszültség: 3,5 – 5.5 V

Áramfelvétel: átlagosan 0.5 mA

HIVATKOZÁSOK

AM2302 adatlap: www.aosong.com

Adafruit DHT sensor library:

<https://github.com/adafruit/DHT-sensor-library>

Gadget Renesas (Sakura kártya):

www.renesas.com/products/promotion/gr/index.jsp

Sakura Web Compiler: <http://tool-cloud.renesas.com>

Processing fejlesztői környezet: www.processing.org

Temperature/relative humidity logger project:

<http://embedded-lab.com/blog/?p=5453>



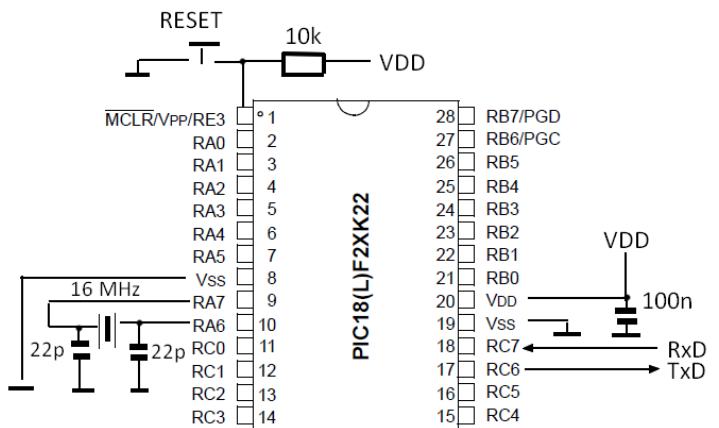


Az **AMICUS** egy Ingyenes és korlátozás nélküli BASIC fordító és integrált fejlesztői környezet Microchip PIC18F25K20 és PIC18F25K22 mikrovezérlőkhöz. Az Amicus18 kártya készen is kapható, de az egyszerű kapcsolás házilag is összeállítható. A mikrovezérlőn kívül csak egy USB-UART átalakítóra lesz szükségünk.

A **PIC18F25K22** mikrovezérlő előnyös tulajdonsága, hogy 3,3 V-os és 5 V-os tápfeszültségen is működőképes (egyszerűbb az illeszkedés a különféle szenzorokhoz).

Főbb jellemzők: 8 bites adatút, 16 MIPS sebesség, 32 kB programmemória, RAM 1536 bájt, EEPROM 256 bájt, 3 db 8 bites, 4 db 16 bites időzítő/számláló, 5 db Captur/Compare/PWM csatorna.

Programbetöltés: DS30 bootloaderrel (nem kell programozó készüléket venni)



Mintapélda: egyszerű LED villogtató program

```
' Flash an LED connected to RB4
Device = 18F25K22
while 1 = 1
    High RB4
    DelayMS 250
    Low RB4
    DelayMS 250
wend
' Create an endless loop
' Illuminate the LED connected to bit-0 of PortB
' wait for half a second
' Extinguish the LED connected To bit-0 of PortB
' wait for half a second
```

Mintapélda: Soros kommunikáció a PC-vel (Hello World)

```
' Transmit text serially (RS232) from the Amicus18 board
;
' view the results on the serial terminal set to 9600 baud 8N1
;
Device = 18F25K22
while 1 = 1
    HRSout "Hello world",13
    DelayMS 1000
wend
' Create an endless loop
' Transmit the text followed by a carriage return
' wait one second. i.e. 1000 Milliseconds
```