

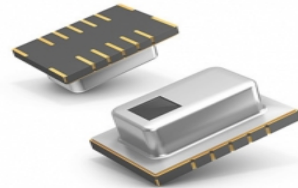


A Panasonic GRID-EYE szenzor 2. generáció használata – 2. rész

Cikkorozatunk előző részében bemutattuk a Panasonic GridEye termoelem mátrix szenzort és annak felhasználási lehetőségeit. Mostani írásunk célja, hogy bemutassuk, hogyan lehet ezt a szenzort egy önállóan működő mini-hőkameraként felhasználni. A cikksorozat második részében a korábban megismert kiértékelő kit Arduino pajzsként való felhasználásával egy mini mikroporcesszoros alkalmazást építünk, amivel érzékelni tudjuk az emberi kéz közelségét és bekapcsolunk segítségével egy relét, ezzel szimulálva egy gesztusvezérelt elektronikai rendszert (jelenlét érzékelésen alapuló vezérlés). Másik alkalmazásként egy PC alapú mozgáskövető világítást mutatunk be.

A GRID-EYE szenzor

Mint azt már az olvasó tudja, a Panasonic GRID-EYE eszköze egy 8X8-as MEMS technológiára épülő hőelem mátrix, azaz 64 különálló szenzorral képes abszolút hőmérsékletet detektálni az objektum által kibocsátott infravörös sugárzás érzékelésével.



A Grid-EYE képes a hőmérséklet és a hőmérsékleti gradiens észlelésére és egyszerű, kifelbontású (8X8=64 pixeles) hőkép felvételére is.

A végtermék fejlesztés megkönnyítésére készített, a szenzort és soros kommunikációs interfészt tartalmazó kártya kialakítása alkalmassá teszi azt ARDUINO DUE miniszámítógéphez illeszkedő „shield”-ként való használatra.



Fontos felhívunk a figyelmet arra, hogy a GridEye kiértékelő kit 3.3V-os kártya, így csak az Arduino DUE számítógépekkel működik.

Az UNO-hoz már csak fizikai kialakítása miatt sem alkalmazható, a MEGA verziók, habár rendelkeznek a megfelelő csatlakozókkal amelyek látszólag illeszkednek is a kártya fizikai csatlakozásaihoz, azok magasabb, (5V-os) feszültségszintje miatt nem kompatibilisek a 3.3V-os GridEye Evaluation kit-el.

A feladat

A megvalósítandó szimulációs feladat során az Arduino számítógéppel az I²C buszon keresztül kiolvassuk a 64 pixel hőmérsékleti adatait.

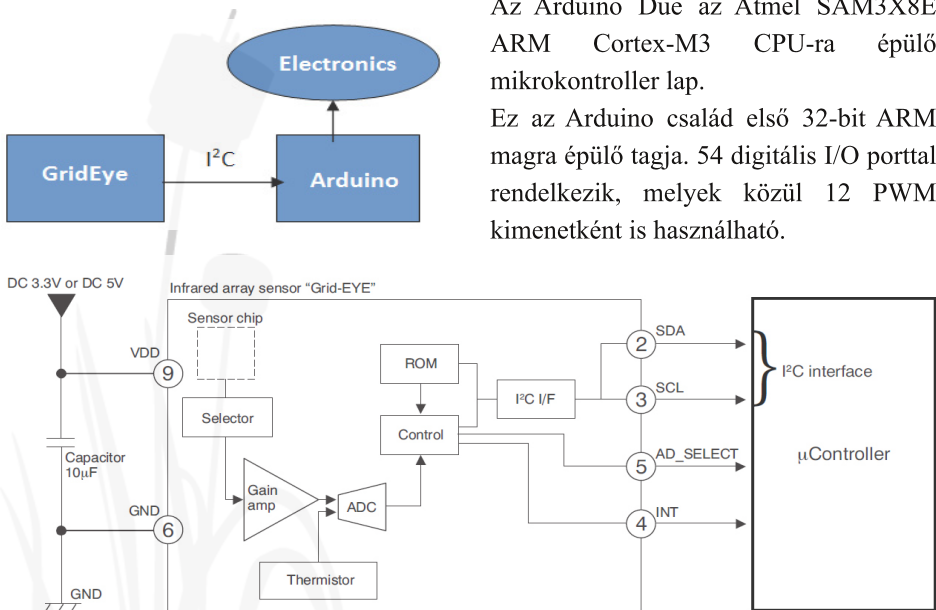
A lapon található egy termisztor is, mely a NYÁK környezeti hőmérsékletét méri, természetesen ennek adatára is szükség van.

A kéz érzékelését az egyszerűség kedvéért úgy végezzük, hogy azon pixelek hőmérsékletadatait, melyek a a szenzor környezeti hőmérsékletnél nagyobb értékkel rendelkeznek összeátlagoljuk, és amennyiben ez az érték 5°C-kal meghaladja a szoba hőmérsékletétől függően beállított határértéket, úgy feltételezzük a hőforrás jelenlétét és egy digitális kimenet logikai magas szintre állításával egy relét kapcsolunk.

Az Arduino DUE számítógép

Az Arduino Due az Atmel SAM3X8E ARM Cortex-M3 CPU-ra épülő mikrokontroller lap.

Ez az Arduino család első 32-bit ARM magra épülő tagja. 54 digitális I/O porttal rendelkezik, melyek közül 12 PWM kimenetként is használható.



Meghajthatunk vele például egy szervomotort, mely segítségével követjük a hőforrás mozgását. Mint említettük a korábbi verzióktól eltérően a DUE 3.3V-os kártya, mely azt is eredményezi, hogy az I/O pinek maximális jelszintje is csak 3.3V.

Emiatt gondoskodni kell a TTL szinthez való illesztésről, például SN74HC125N vonalmeghajtó alkalmazásával, mely kétirányú feszültség illesztést tesz lehetővé (3.3V- 5V).

Panasonic GridEye könyvtár

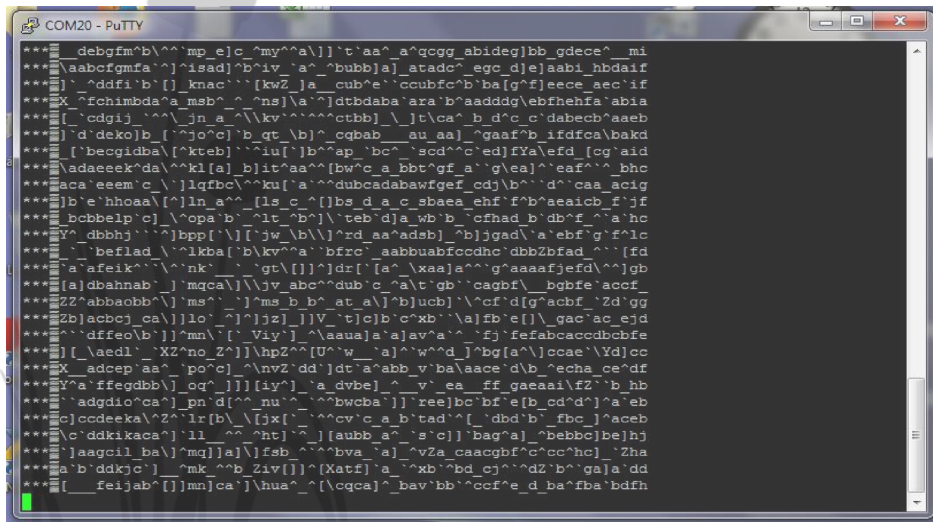
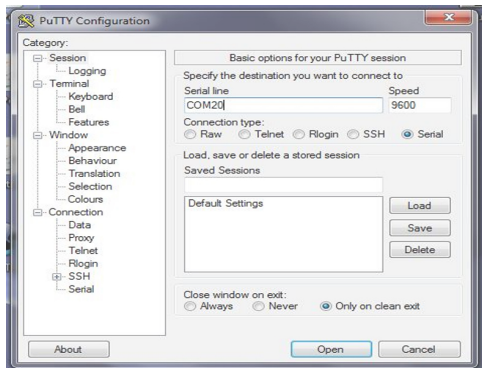
A Panasonic támogatói oldalról letölthető a GridEye Evaluation Kit V1.0 számára szükséges Arduino Library, melyet az Arduino IDE számára úgy tehetünk elérhetővé, hogy bemásoljuk azt az <arduino könyvtár>/libraries helyre.

Az IDE újraindításával hivatkozhatunk a GridEye.h, a GE_SoftUart.h és a GE_SoftUartParse.h állományra.

A GridEye szenzor adatstruktúrája

A soros porton másodpercenként 10 alkalommal érkező nyers adatok első ránézésre ijesztőek, azonban a struktúra ismeretében könnyen feldolgozhatóak.

A soros port terminál programmal (pl. PuTTY) való vizsgálata az alábbi eredményt mutatja:



135 byte adat érkezik másodpercenként tízszor, melyből néhány vezérlő karakter kiszűrése után a 64 pixel és a termisztor hőmérsékleti adata is kinyerhető az alábbi adatstruktúra szerint:

- 3 byte fejléc ***
- 2 byte NTC hőmérséklet HL,HH (12 bites előjeles)
- 64 X 2 byte (HL HH) 128 byte (12 bites kettes komplementer kódú)
- CR LF (\r \n)

A feladat így már nem annyira bonyolult, az adatstruktúra nagyon logikus. Arra kell csak odafigyelnünk, hogy az alsó (HL) és a felső (HH) byte-ok felcserélt sorrendben érkeznek: (HL, HH).

A hőmérsékleti adatok meghatározásához szükséges konverzió az alábbiak alapján lehetséges.

Az NTC 12 bites előjeles ábrázolása :

- XXXX S _ _ _ _ _
- Ha S=1, akkor a hőmérséklet negatív, ha S=0 akkor pedig pozitív
- A 12 biten tárolt bináris értéke decimálissá alakítás után a 0.0625°C felbontással felszorozva alakítható valós hőmérsékleti adattá:
11 bit = 1024 (211), így a teljes tartományban (-1023 ..+1024) -63 .. +64°C közti hőmérséklet ábrázolását teszi lehetővé.

▪ Példa: HH = 0x01 HL = 2A -> bináris konverzió -> 0000 0001 0010 1010 -> + hőmérséklet $1*2+1*8+1*32+1*256=298$ (decimális) Hőmérséklet = $298*0.0625 = 18,6^{\circ}\text{C}$

▪ A pixel adatok kettes komplementer kódban érkeznek, kicsit más a feldolgozás menete:

XXXX S _ _ _ _ _

A 12. bit itt is az előjelre utal, azonban nemcsak előjelként funkcionál, hanem értéke is részt vesz a kalkulációban: amikor értéke = 1, akkor negatív a hőmérséklet, ilyenkor tehát a decimális érték 2048 helyett -2048 lesz: (-212).

A megvalósításhoz szükséges ARDUINO mintakód az alábbi címen elérhető:

http://www.endrich.hu/mintaprogram/ENDRICH_ARD

[UINO GRIDEYE Elektronet.ino](http://www.endrich.hu/mintaprogram/ENDRICH_ARD)



A program másodpercenként egy, vagy akár tíz alkalommal beolvassa a szenzorról érkező adatokat az I²C buszon keresztül, és ezeket hőmérsékleti adatokká alakítja.

A könnyebb áttekinthetőség érdekében a virtuális soros (esetünkben COM14) porton keresztül kiíratjuk ezeket, és az Arduino IDE beépített terminálján meg is jelenítjük.

```

ENDRICH_ARDUINO_GRIDIYE_forSalesTraining | Arduino 1.6.9
File Edit Sketch Tools Help

ENDRICH_ARDUINO_GRIDIYE_forSalesTraining
.....
Endrich demonstration of Panasonic GridEye sensor
based on original GRIDIYE Demo program of Panasonic
extended with own code for measurement formatting
Explains the usage of the data coming out of GridEye sensor
and displaying this data as explicit temperature values in [C]
(c) 2016
.....

#include <Wire.h>
#include "Arduino.h"
#include <GridEye.h>
#include <GE_SoftTart.h>
#include <GE_SoftTartParse.h>
.....
variable value definition
.....
static uint16_t Main_Delay = 85;
grideye GE_GridEyeSensor;
uint8_t auxInBuf[2]; /* NTC thermistor temperature value */
short g_ahRawTemp[64]; /* Temperature values of 64 pixels */
char buf[2];

.....
Function: GE_SendDataToPC
Description: Used to Send data to PC
Input: None
Output: None
Others: None
.....
void GE_SendDataToPC( void )
{
}

.....
Function: GE_SendDataToPC
Description: Used to Send data to PC
Input: None
Output: None
Others: None
.....
void GE_SendDataToPhone( void )
{
}

```

```

COM14 (Arduino Due (Programming Port))
.....
[Measurement start]
Raw thermistor data : HH:1 HL:11010111
Thermistor temperature : [C]: 29.44
24.50 24.75 23.50 25.25 26.25 24.25 24.00 23.75
26.00 25.00 23.75 24.25 24.75 25.00 24.75 24.75
25.25 24.00 23.75 24.50 24.25 25.25 24.75 25.25
27.00 25.00 25.50 24.00 24.75 25.50 24.75 26.75
[Measurement end]
[Average temperature of all pixels :24C]

[Measurement start]
Raw thermistor data : HH:1 HL:11010111
Thermistor temperature : [C]: 29.44
24.50 25.25 24.00 25.00 26.50 23.75 23.25 23.00
24.50 24.25 23.25 24.00 24.00 24.25 22.75 24.00
25.25 25.00 23.50 23.25 23.25 23.00 23.00 23.00
25.75 26.00 26.00 24.00 24.00 22.75 24.25 24.25
24.50 24.50 24.00 24.75 25.75 23.00 24.25 24.25
26.00 24.50 24.25 23.25 24.00 25.50 24.75 25.25
26.25 23.75 24.00 23.75 24.50 25.25 24.50 25.00
27.00 25.50 24.50 24.75 24.50 25.25 24.75 25.25
[Measurement end]
[Average temperature of all pixels :24C]

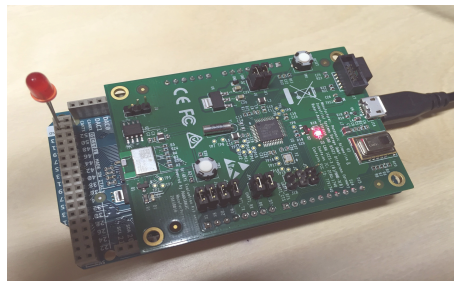
Autoscroll Both NL & CR 57600 baud

```

Amennyiben a szenzor 64 pixelének hőmérsékleti átlagértéke meghaladja az előre beállított küszöbértéket (a programban `temperature_treshold = 25°C`), akkor jelen van a szenzor látóterében egy hőszugárzó tárgy, jelen esetben a kezünk. (Nem szükséges a mozgás, elég a jelenlét, ez a nagy előnye a GridEye szenzornak pl. a PIR vagy a radar szenzor technológiával szemben.) Tovább árnyalható az algoritmus azzal, hogy csak azok a pixelek vesznek részt az átlagolásban, melyek hőmérsékletértéke meghaladja a GridEye beépített termisztorának mért értékét.

Amennyiben pozitív eseményt érzékelünk, az Arduino egyik szabadon használható digitális I/O portjára logikai „1” értéket írunk ki (GPIO 53 a

mintaprogramban). Ezzel tesztelés közben egy a GPIO63 és a GND lábak közé illesztett LED-el ellenőrizhetjük a helyes működést.



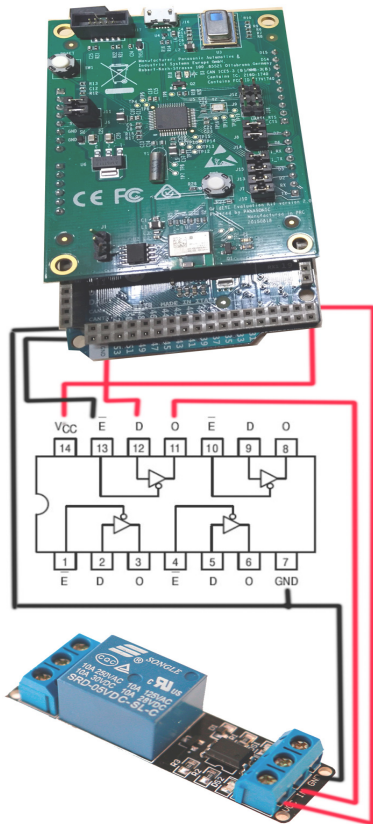
Fentebb említettük, hogy az Arduino DUE 3GPIO lábai 3.3V-os kimeneti jelszintet adnak ki logikai „1 – HIGH” állapotban. Mivel ez általában nem elegendő a kereskedelmi forgalomban leginkább elérhető 5V-os relék

vezérlésére egy vonali 3.3V-5V szintkonverter beépítése ajánlott.

Egyik lehetséges megoldás a SN74HC125N logikai szintillesztő IC-vel, mely egy 14 lábú tokban 4 csatonás megoldást ad, melyből a mintapéldában csak egyet használunk. Egy relé meghajtásához általában elegendő az Arduino DUE által szolgáltatott teljesítmény, azonban több relé egyidejű használatakor ezek külső táplálása szükséges, ekkor ne feledkezünk el a tápegység és az ARDUINO „GND” lábak közösítéséről és esetleges túlfeszültségvédő dióda alkalmazásáról sem. A bemutatott megoldással tehát egy 64 pixeles felbontású mini hőkamera jelenlétérzékelésre és ennek alapján egy – akár hálózati feszültségű – külső áramkör relén keresztül vezérlésére adtunk megoldást.

A GridEye evaluation kit PC-vel való használata

A GridEye Evaluation kit nemcsak Arduino Shieldként, hanem USB portján keresztül PC-vel összekötve is használható. Ilyenkor a megfelelő device driver alkalmazásával az USB-re csatlakoztatott eszköz virtuális COM (soros) porton keresztül érhető el. Az adatstruktúra fenti leírása itt is érvényes, valamilyen PC-s fejlesztőeszközzel megírható az adatok vizualizációját szolgáló szoftver. Egy ilyen megoldás



készen letölthető a Panasonic WEB oldaláról is, de mi is készíthetünk egyet. Én a Lazarus Free Pascal rendszert használtam, ingyenessége és a Delphi környezet kényelmi funkcióinak elérhetősége miatt.

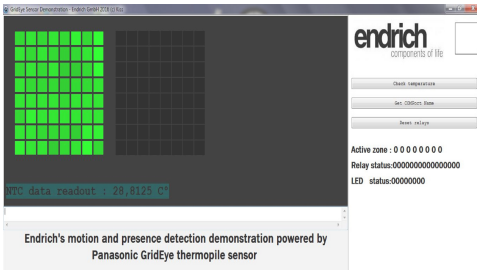
A teljes szoftvert nem célok bemutatni, azonban néhány szóban összefoglalom.

Az alkalmazás, amit ilyen módon készíthetünk egy mozgáskövető (lámpa)kapcsoló. A környezeti hőmérséklettől eltérő pixelek kiszűrésével a jelenlét és a mozgás

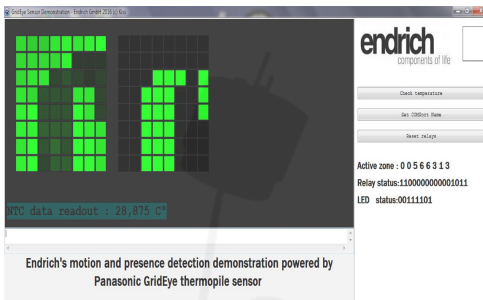
érzékelhető, a helyzetét változtató „hot-spot” – ot követjük 8 digitális relé kapcsolatásával. Ily módon elérhetjük, hogy a fény kövesse a mozgó céltárgyat, pl. egy emberi kezét (gesztusvezérlés).

Példák

Nincs semmi a szenzor előtt:



Kéz a szenzor előtt (bal oldalon a szenzor egyszerűsített hőképe, jobbra a kiemelt céltárgy képe):



A teljes rendszer blokkdiagramja az ábrán látható. A nyolc függőleges LED csík fénye követi a kézmozgást, vagy a szenzor előtt elhaladó emberek mozgását.

Cikksorozatunk harmadik, azaz a következő és egyben befejező részében a GridEye szenzorral megvalósítunk egy önállóan futó Arduino Due alkalmazást, mely a mini hőkamera képét egy 8X8-as RGB LED mátrix-on jeleníti meg, miközben a céltárgy mozgását egy szervomotorral igyekszünk követni.

