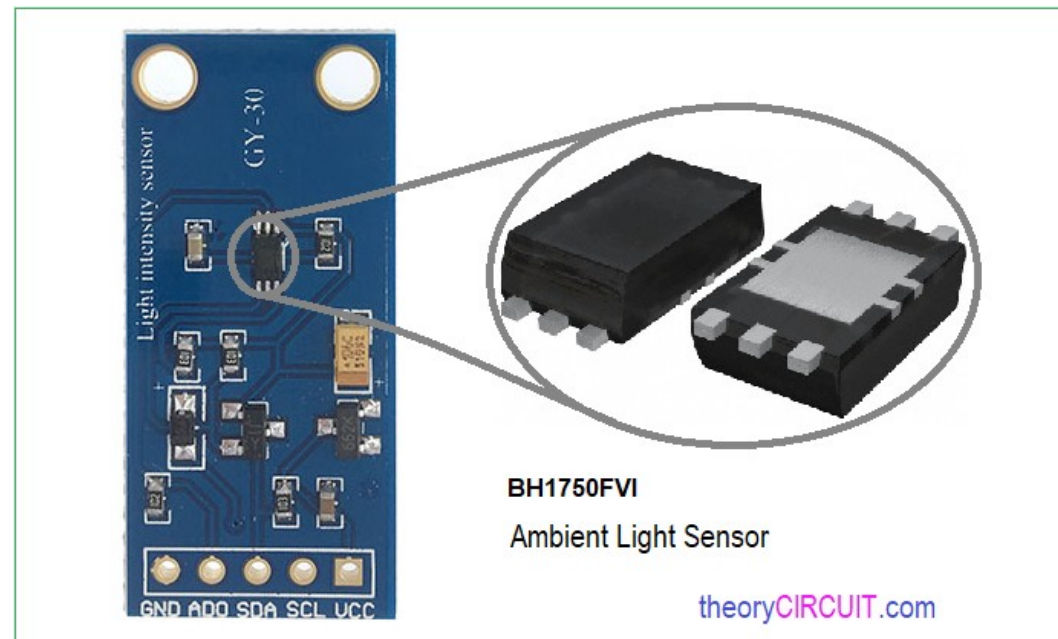


Tarján Péter
Nyíregyházi Egyetem

Szenzorok és ötletek Arduinós mérésekhez

BH1750 megvilágítás

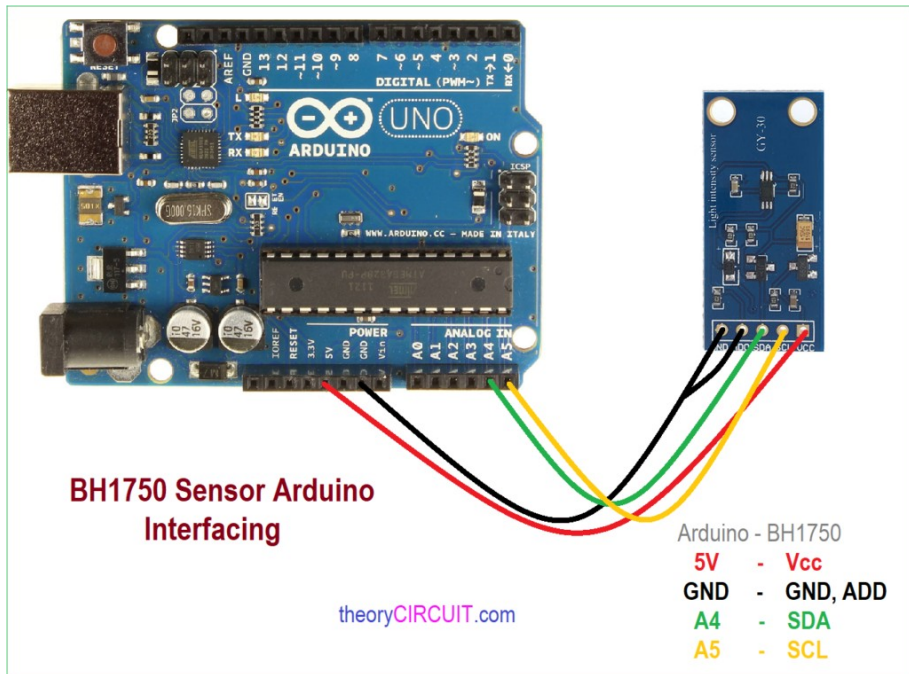
- Spektrális érzékenysége az emberi szemhez hasonló
- Közvetlenül luxban adja vissza a megvilágítás értékét
- Lineáris válasz
- Egyszeri vagy folytonos mérés
- alacsony felbontású mód:
4 lx pontosság, 16 ms mérési idő
- magas felbontású mód: 1/0.5
lx pontosság, 120 ms mérési idő



Cím az I2C buszon:
0x23 vagy 0x5C

Ára: \$1 (ebay)

BH1750 kapcsolás és példa



```
/* Csatlakoztatás Arduino Unóhoz vagy Nanóhoz:
```

```
VCC  <-> 3V3
```

```
GND  <-> GND
```

```
SDA  <-> A4/SDA
```

```
SCL  <-> A5/SCL */
```

```
// a szükséges függvénykönyvtár
```

```
#include <BH1750FVI.h>
```

```
// Inicializáljuk a szenzort
```

```
BH1750FVI FenySzenzor(BH1750FVI::k_DevModeContHighRes);
```

```
// a setup
```

```
void setup() {
```

```
// soros portti kommunikáció indítása 115200 baud sebességgel
```

```
    Serial.begin(9600);
```

```
// beindítjuk a szenzort
```

```
    FenySzenzor.begin();
```

```
}
```

```
void loop(){
```

```
// kiolvassuk a szenzort
```

```
    uint16_t Megvilagitas = FenySzenzor.GetLightIntensity();
```

```
// kiírjuk a soros portra a kapott értéket
```

```
    Serial.print("Megvilágítás: ");
```

```
    Serial.print(Megvilagitas);
```

```
    Serial.println(" lx");
```

```
// várunk 2 másodpercet a szenzor újabb kiolvasása előtt
```

```
    delay(2000);
```

```
}
```

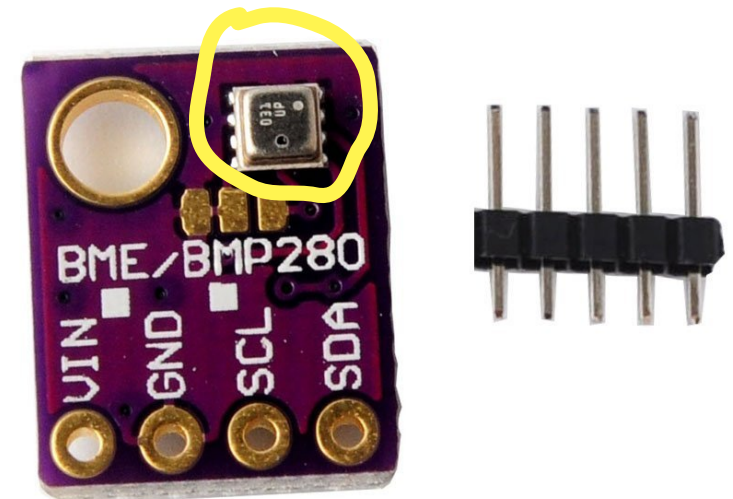
BH1750 ötletek

- különböző fényforrások fényerősségének összehasonlítása
- $1/r^2$ -es távolságfüggés igazolása
- lézer fényeloszlásának vizsgálata
- lézernyaláb divergenciájának meghatározása
- ergonómia ellenőrzése
- növények életkörülményeinek vizsgálata

BME280

légnyomás+hőmérséklet+páratartalom

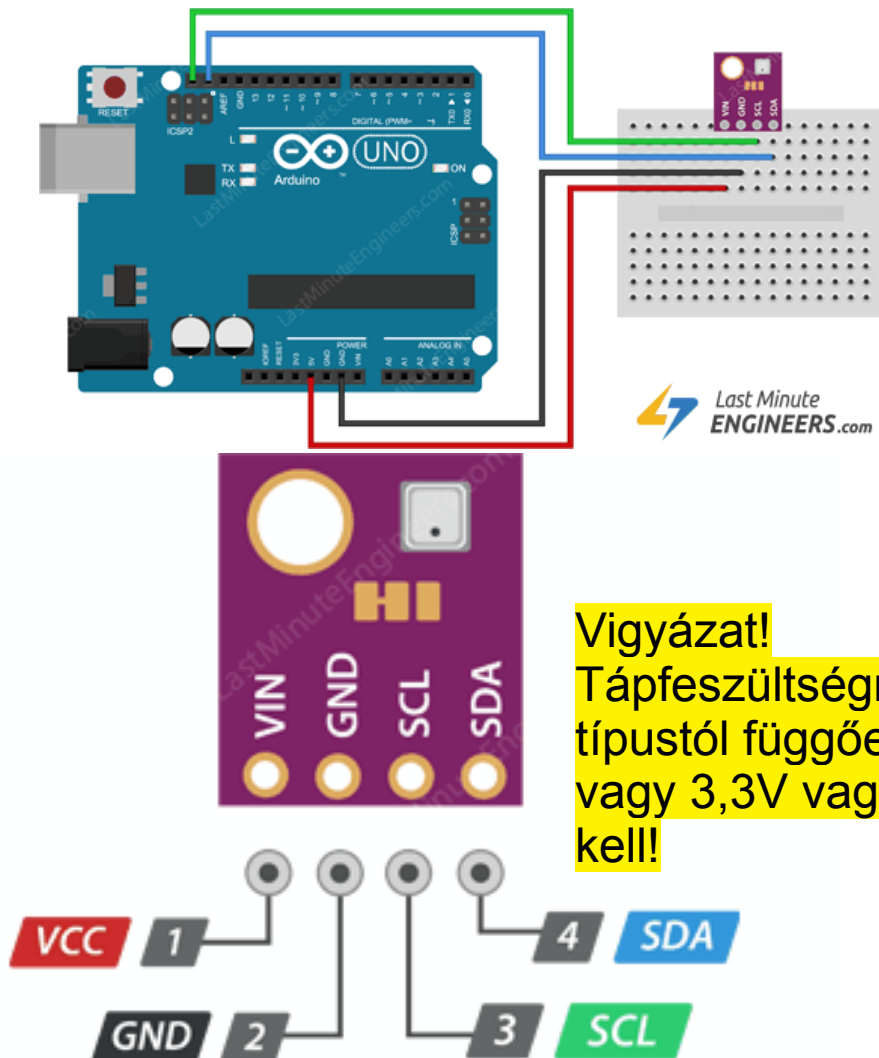
- sokoldalú, nagy pontosságú kombinált szenzor
- elődei: BMP080 (p, T), BMP180 (p, T), BMP280 (p, T)
- légnyomás: 30-110 kPa, 0,1 kPa abszolút pontosság (magasságmérés: 1 m pontosság)
- hőmérséklet: -40-85 °C, 1 °C abszolút pontosság
- páratartalom: 0-100%, 3% abszolút pontosság
- a mért értékek *felbontása*, a mérés érzékenysége még jobb



Cím az I2C buszon:
0x76 vagy 0x77

Ára: \$3,80 (ebay)
(BMP280: \$1)

BME280 kapcsolás és példa



```
// a szükséges függvénykönyvtárak
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
// a tengerszinti légnyomás megadása
// (magasság számításához)
#define SEALEVELPRESSURE_HPA (1013.25)
Adafruit_BME280 bme;

void setup() {
  Serial.begin(9600);
  // ha nincs a szenzor a 0x76 címen, akkor hibaüzenet...
  if (!bme.begin(0x76)) {
    Serial.println("Nincs meg a BME280 szenzor, nézd át a
    kapcsolást!");
    // ... és végtelen ciklusban itt várunk
    while (1);
  }
}

void loop() {
  Serial.print("Hőmérséklet = ");
  Serial.print(bme.readTemperature());
  Serial.println(" *C");
  Serial.print("Légnyomás = ");
  Serial.print(bme.readPressure() / 100.0F);
  Serial.println(" hPa");
  Serial.print("Tengerszint feletti magasság = ");
  Serial.print(bme.readAltitude(SEALEVELPRESSURE_HPA));
  Serial.println(" m");
  Serial.print("Páratartalom = ");
  Serial.print(bme.readHumidity());
  Serial.println(" %");
  Serial.println();
  delay(2000);
}
```

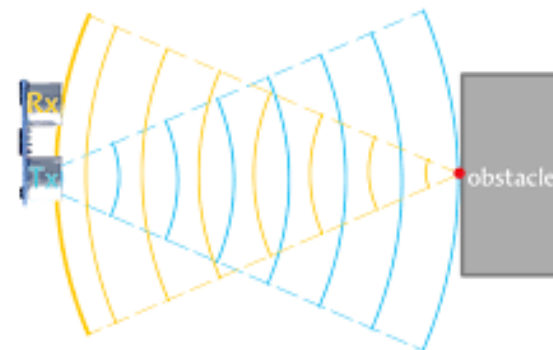
BME280 ötletek

- levegőhőmérséklet-monitorozás
- légnyomás mérése emeletenként
- relatív páratartalom/hőmérséklet összefüggése
- időjárás-állomás
- áramlási sebesség mérése
- ergonómia ellenőrzése
- növények életkörülményeinek vizsgálata
- újabb szenzor: BME680 levegőminőséget is mér, \$13.

HC-SR04 távolság*

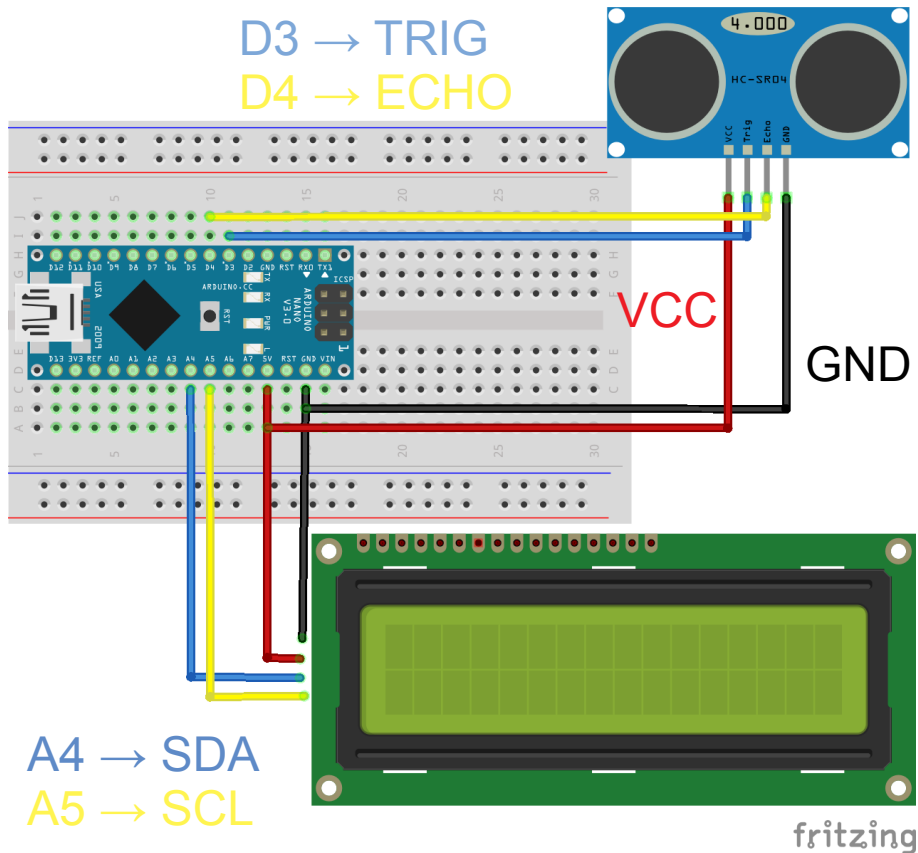
(*valójában
visszatérési idő)

- általában távolságmérésre használt olcsó, jó pontosságú ultrahangos szenzor
- 3 cm – 500 cm hatótávolság (a minimális hatótávolság alatt rossz, de sajnos hihető értékeket ad vissza!)
- hangkúp szöge $< 15^\circ$
- felbontás: 0,3 cm
- ultrahang-frekvencia: 40 kHz



Ára: \$1 (ebay)

HC-SR04 kapcsolás és működés

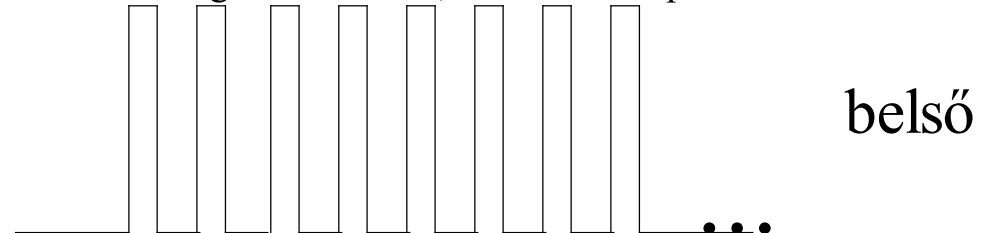


16x2 karakteres i2c kijelző
(cím az i2c buszon: 0x27)

10 μ s-os impulzus a szenzor TRIG bemenetén indítja a mérést...



...a szenzor generál 8 rövid, 40 kHz-es impulzust...

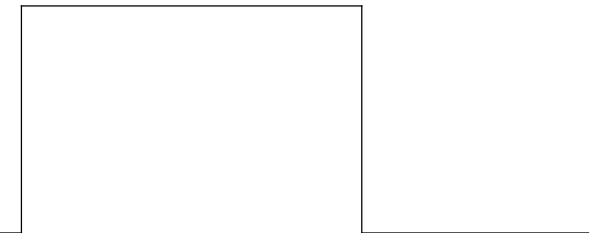


... a szenzor megvárja a visszhangot, majd az ECHO kimenetét annak megfelelő ideig magasra húzza. Ezt mérjük!

később:

ECHO

...



HC-SR04 példa

```
#define tu_trig 3 // az UH szenzor TRIG bemenetére kötött tű
#define tu_echo 4 // az UH szenzor ECHO bemenetére kötött tű
#include <Wire.h> // az i2c kijelzővel kommunikációhoz
// Innen kell beszerezni az i2c LCD függvénykönyvtárat
// https://bitbucket.org/fmalpartida/new-liquidcrystal/downloads
#include <LiquidCrystal_I2C.h>
// az LCD kijelző címe az i2c buszon 0x27
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
unsigned long ido = 0;

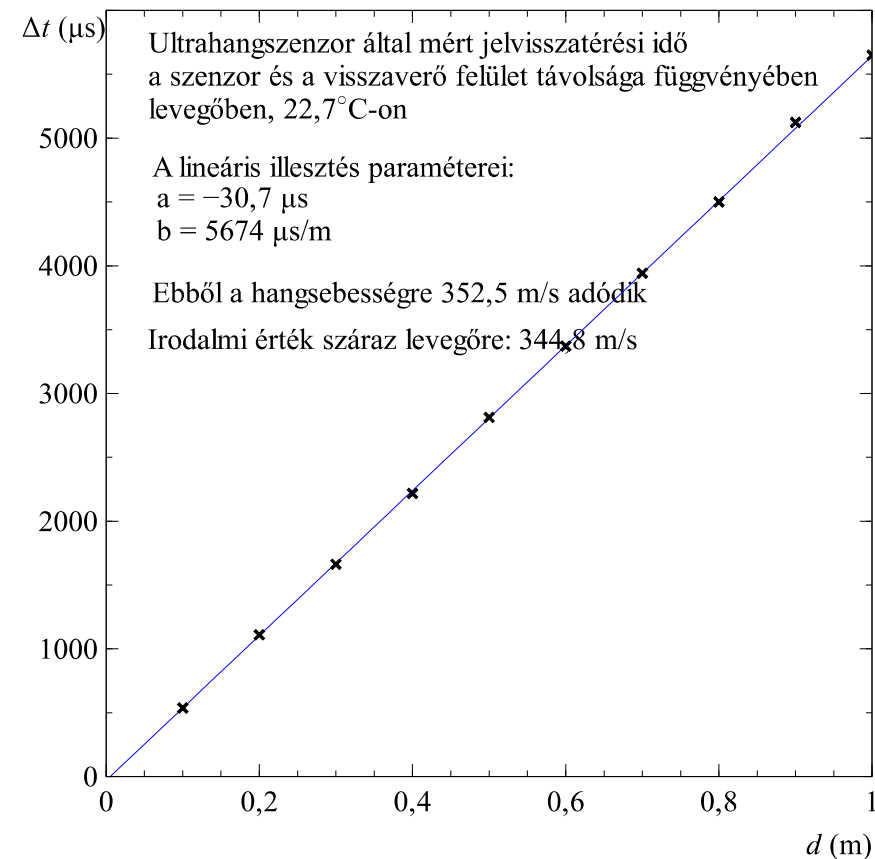
void setup()
{
    // a trig tű kimenet lesz, az echo tű bemenet
    pinMode(tu_trig, OUTPUT); pinMode(tu_echo, INPUT);
    // elindítjuk a kommunikációt az LCD kijelzővel,
    // megmondjuk, hogy a mérete 16 karakter x 2 sor
    lcd.begin(16, 2);
    lcd.backlight(); // bekapcsoljuk a kijelző háttérvilágítását
    Serial.begin(9600);
}

void loop()
{
    // a mérés az időmérés függvény meghívásával történik
    ido = visszhangido(tu_trig, tu_echo);
    Serial.println(ido);
    lcd.clear(); // a kijelző törlése
    lcd.setCursor(0, 0); // a kurzor pozicionálása
    lcd.print("Eltelt ido: ");
    lcd.setCursor(6, 1); // a mérési eredmény a 2. sorba kerül
    lcd.print(ido); // a mérési eredmény kiírása
    lcd.setCursor(13, 1);
    lcd.print("us"); // u, mint mikro-
    delay(2000);
}
```

```
// mikrosec-ban (us) adja vissza az időt
// ezt a függvényt hívjuk meg a loopból a méréshez
unsigned long visszhangido(int tu_t, int tu_e)
{
    // 0-ra húzzuk a szenzor TRIG bemenetét 10 us-ig
    digitalWrite(tu_t, LOW);
    delayMicroseconds(10);
    // majd 1-re húzzuk a szenzor TRIG bemenetét 10 us-ig
    digitalWrite(tu_t, HIGH);
    delayMicroseconds(10);
    digitalWrite(tu_t, LOW);
    // megmérjük, az ECHO tű mennyi ideig volt magasan
    // annyi ideig tartott, amíg visszajött a visszhang
    unsigned long ido = pulseIn(tu_e, HIGH);
    if (ido != 0)
    {
        return ido; //mértékegység: mikroszekundum
    }
    else
    {
        return -1;
    }
}
// persze a hangsebesség ismeretében egyből a
// távolságot is kiíráthatnánk
```

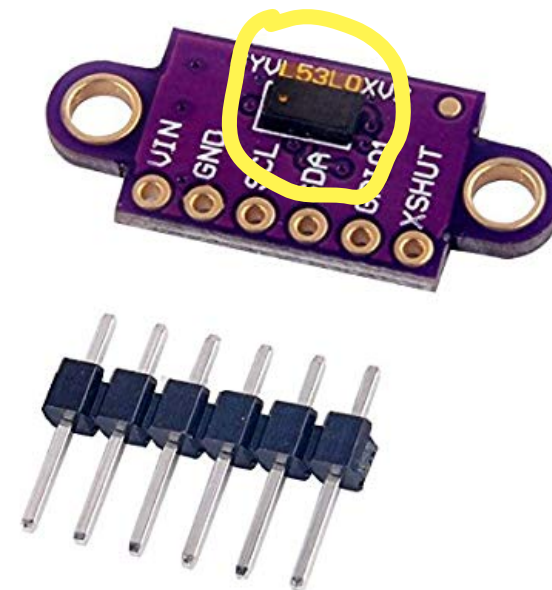
HC-SR04 ötletek

- távolságok automatikus detektálása
- hangsebesség mérése
- folyadékszint mérése
- robot tájékozódás
- fix reflektortávolsággal: hőmérsékletmérés
- szélerősségmérés?



VL53L0X távolság

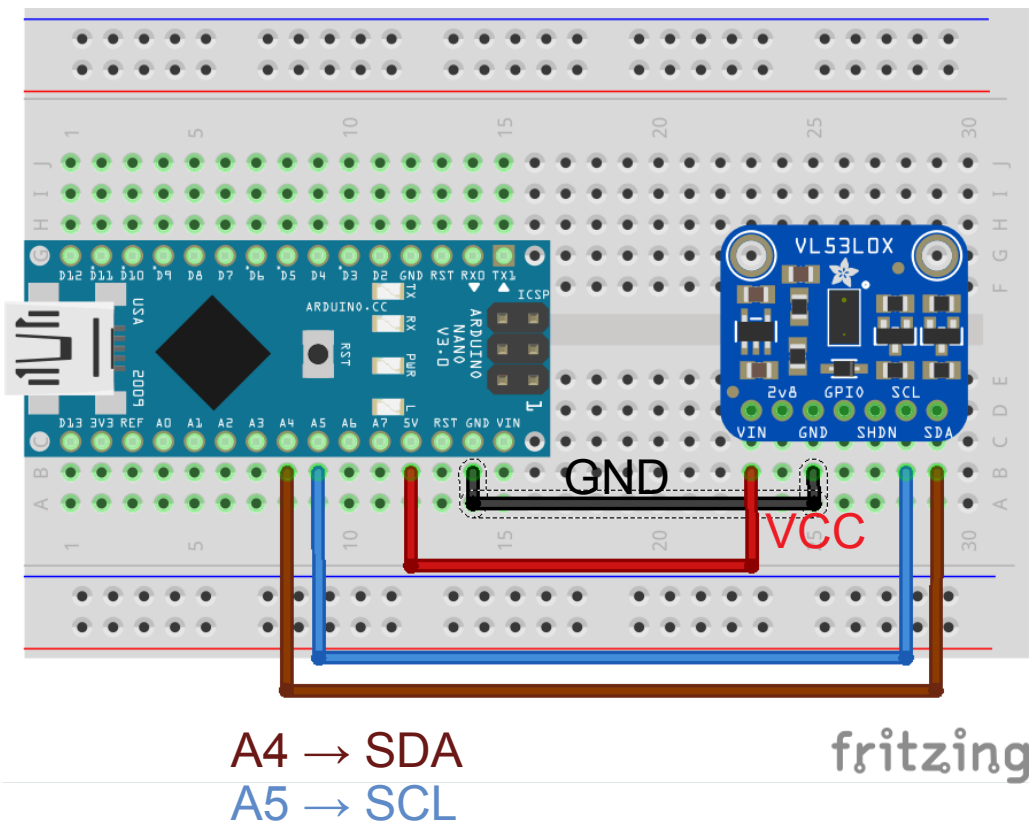
- jelenleg a legkisebb lézeres távolságmérő szenzor
- akár 200 cm hatótávolság
- 940 nm hullámhosszú infravörös lézer
- visszatérési időt mér, nem visszavert intenzitást
- 3-9% pontosság (mérési mód, zavaró tényezők függvénye)
- mérési idő: 20-200 ms (tipikusan: 30 ms)



Cím az I2C buszon:
0x29

Ára: \$5 (ebay)

VL53L0X kapcsolás és példa



```
// a szükséges függvénykönyvtár
#include "Adafruit_VL53L0X.h"
Adafruit_VL53L0X lox = Adafruit_VL53L0X();

void setup() {
  Serial.begin(115200);
  // nem megyünk tovább, amíg nincs soros kommunikáció
  while (!Serial) {
    delay(1);
  }
  Serial.println("VL53L0X teszt");
  if (!lox.begin()) {
    // ha nincs kapcsolat a szenzorral, megállunk
    Serial.println(F("Nincs kapcsolat a szenzorral!"));
    while(1);
  }
  Serial.println(F("VL53L0X távolságmérés példa\n\n"));
}

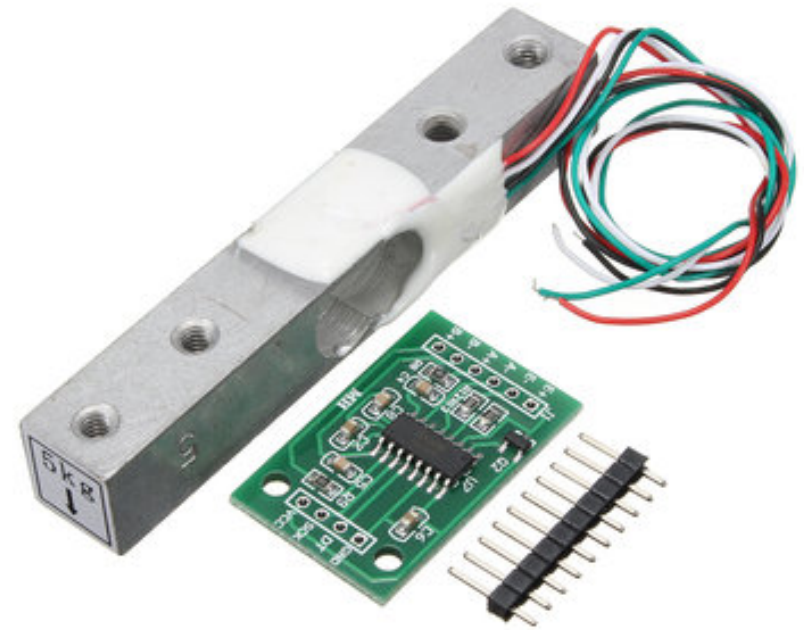
void loop() {
  VL53L0X_RangingMeasurementData_t measure;
  Serial.print("Mérés indítása... ");
  lox.rangingTest(&measure, false);
  // ha nem hibás a mért adat...
  if (measure.RangeStatus != 4) {
    // ...akkor kiíratjuk.
    Serial.print("Távolság (mm): ");
    Serial.println(measure.RangeMilliMeter);
  } else {
    Serial.println(" hatótávolságon túl! ");
  }
  // 1 másodperc szünet két mérés között
  delay(1000);
}
```

VL53L0X ötletek

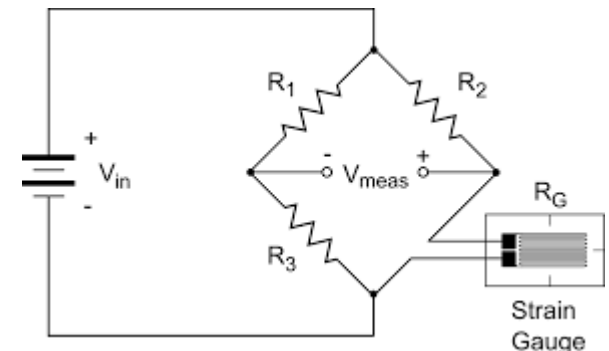
- távolságok automatikus detektálása
- ~~fénysebesség meghatározása~~
- robot tájékozódás
- folyadékszint mérése
- törésmutató meghatározása?

Mérlegcella súly, erő

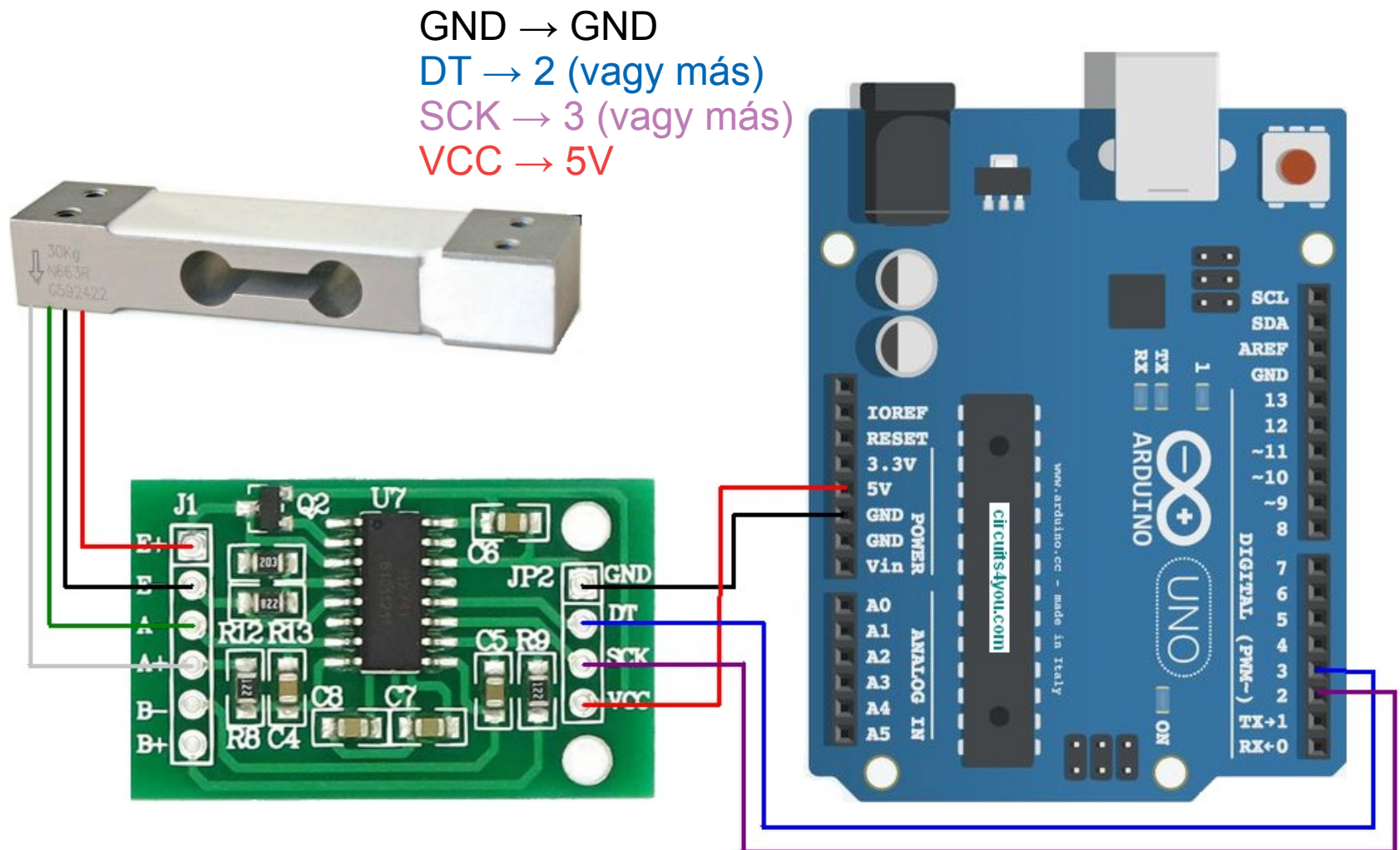
- pontos erőmérést tesz lehetővé
- sok verzióban kapható, különböző méréshatárok
- deformációt érzékel, nyúlásmérő bélyegekkel
- a nyúlásmérő bélyegek ellenállásváltozását egy Wheatstone-híd konvertálja feszültséggé
- a kis feszültségjelet egy 24 bites felbontású analóg-digitális átalakító dolgozza fel



Ára: \$5-\$20 (ebay)



Mérlegcella kapcsolás



Mérlegcella példa

```
#include <HX711_ADC.h>
#include <EEPROM.h>

//HX711 konstruktor (DO/DT/SIG/DATA tű, SCK tű):
HX711_ADC LoadCell(2, 3);
const int eepromAdress = 0;
long t;

void setup(){
    float calValue; // kalibrációs konstans
    // calValue = 1; // beállítható itt
    // vagy ha korábban a kalibrációs programot futtattuk, akkor
    // kiolvassa EEPROM-ból
    EEPROM.get(eepromAdress, calValue);
    // üzenetek a soros portra
    Serial.begin(9600); delay(10); Serial.println();
    Serial.println("Kezdés");
    // az erőmérő inicializálása & tárázása
    LoadCell.begin();
    // javítja a tárapontosságot, ha adunk neki időt
    long stabilisingtime = 3000;
    LoadCell.start(stabilisingtime);
    if(LoadCell.getTareTimeoutFlag()) {
        Serial.println("Időtúllépés, biztos jó a vezetékezés?");
    }
    else {
        LoadCell.setCalFactor(calValue); // kalibráció alkalmazása
        Serial.println("Tárázás kész.");
    }
}
```

```
void loop() {
    // ezt legalább mintavételenként
    // meg kell hívni
    LoadCell.update();

    // adatok símitása és kiírása
    if (millis() > t + 500) {
        float i = LoadCell.getData();
        Serial.print("Load_cell output val: ");
        Serial.println(i);
        t = millis();
    }

    // ha üzenet jön a soros porton, tárázzunk
    if (Serial.available() > 0){
        float i;
        char inByte = Serial.read();
        if (inByte == 't')
            LoadCell.tareNoDelay();
    }

    // az utolsó tárázás hiba nélkül lefutott?
    if (LoadCell.getTareStatus() == true) {
        Serial.println("Tárázás kész.");
    }
}
```

Mérlegcella ötletek

- erő/súly/tömegmérés
- Newton-törvények ellenőrzése
- tapadási/súrlódási együttható mérése
- Hooke-törvény ellenőrzése
- kötélerők mérése

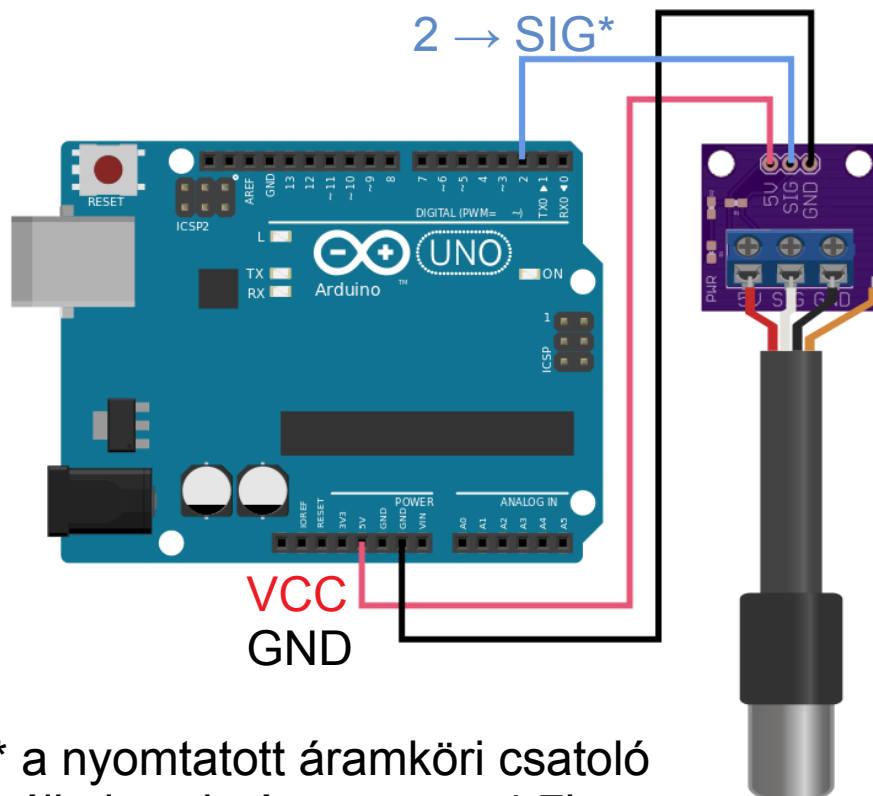
DS18B20 hőmérséklet

- olcsó, pontos, kalibrált, digitális kimenetű hőmérsékletszenzor
- vízálló változatban is
- mérési tartomány: $-55-125\text{ }^{\circ}\text{C}$
- pontosság: $0,5\text{ }^{\circ}\text{C}$ ($-10\text{ }^{\circ}\text{C}$ és $85\text{ }^{\circ}\text{C}$ között)
- minden szenzornak egyedi sorozatszám van, egyszerre több is használható



Ára: \$1,20-\$3,20 (ebay)

DS18B20 kapcsolás és példa



```
// szükséges függvénykönyvtárak
#include <OneWire.h>
#include <DallasTemperature.h>
// A SIG vezetékét az Arduino 2-es tűjére kötjük
#define ONE_WIRE_BUS 2
// létrehozuk az osztály egy példányát
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
```

```
void setup(void){
  // soros kommunikáció indítása
  Serial.begin(9600);
  // indítjuk a szenzor(ok)at
  sensors.begin();
}
```

```
void loop(void) {
  // a sensors.requestTemperatures() bekéri a hőmérsékletet
  sensors.requestTemperatures();
  Serial.print("T = ");
  // a buszon levő első szenzor (0. index) által mért hőmérséklet
  Serial.print(sensors.getTempCByIndex(0));
  delay(1000);
}
```

* a nyomtatott áramköri csatoló nélkül szükség van egy 4,7k-s ellenállásra a táp- és az adatvonalak között

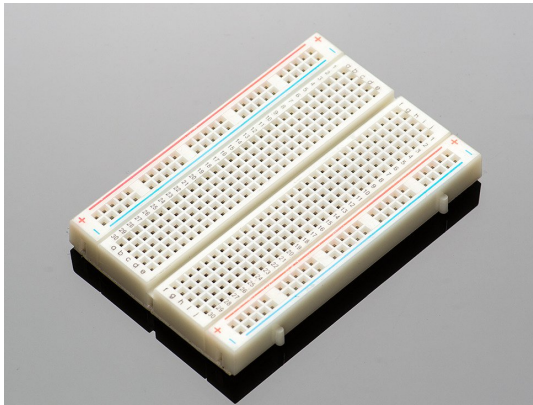
* a tápvezeték elhagyható, az adatvezeték magában is el tudja látni táppal a szenzort

DS18B20 ötletek

- folyadékok hőmérsékletének mérése
- gáztörvények igazolása
- hőterjedés vizsgálata
- olvadáspont, forráspont meghatározása
- kalorimetria
- hőtágulás vizsgálata

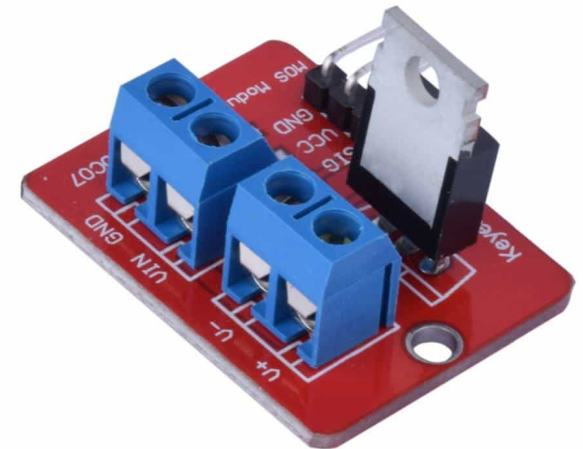
Egyéb hasznos cuccok

MOSFET modul
MOSFET module



próbapanel
breadboard

5V-os feszültségnövelő
5V voltage booster



ceruzaelem-tartó
AA battery box

LEDsupply
FOR ALL YOUR LED PROJECT NEEDS



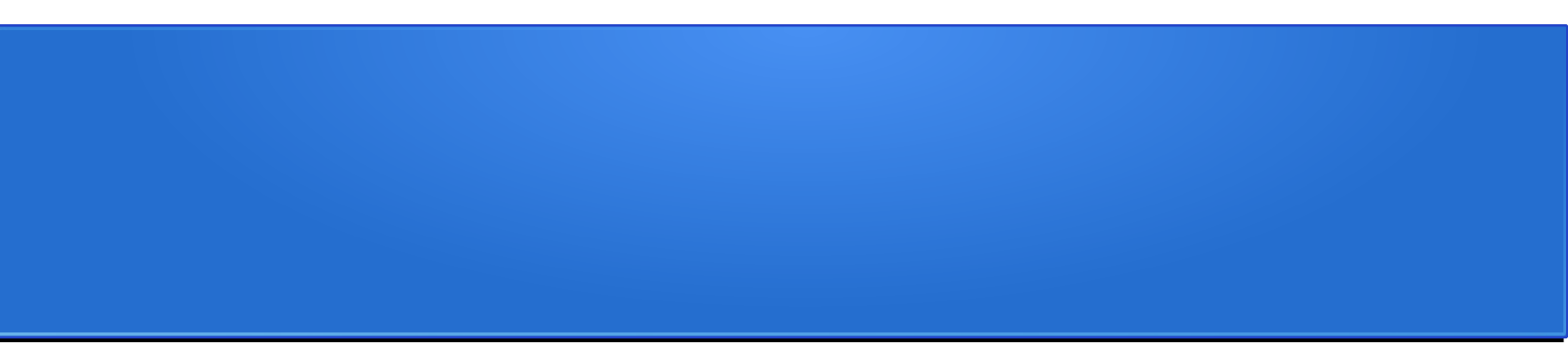
16x2 karakteres LCD (i2c!)
16x2 character LCD i2c



vezetékek
Dupont wire (male-male,
male-female, female-female)

Összefoglalás

- Nagyon sokféle fizikai mennyiség mérésére létezik elektronikus szenzor
- A szenzorok jelentős része egészen olcsó
- Az Arduino platform elterjedtsége és népszerűsége miatt szinte minden szenzorhoz létezik a használatát segítő függvénykönyvtár és példakód
- Gyakran nincs szükség jelentős programozói tudásra
- Szenzorok kreatív felhasználásával és kombinálásával akár összetett fizikai mérésekre is lehetőség nyílik
- Ha a diákok kezébe adjuk az eszközöket, az elektronika passzív fogyasztóiból aktívan bütykölő, érdeklődő, újat létrehozó embereket nevelhetünk.



Köszönöm a figyelmet!

Kérések, kérdések, óhajok, sóhajok:

Tarjan.Peter@nye.hu