***Teljesítménymérés Arduino-val***

Szeretnék egy univerzális teljesítménymérőt készíteni Arduinoval. Ez a végső célom.

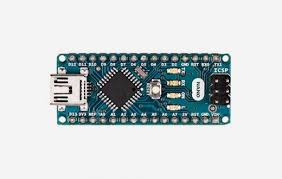
Legelőször szerettem volna valahogyan egyenfeszültséget mérni. Sikerült.

Majd valahogyan egyenáramot mérni. Sikerült.

Ezek után már jött magától a teljesítménymérés. Idáig jutottam.

Hogyan is sikerült mindez? Kezdjük a legelején!

1. ***Az Arduino***

Az Arduino egy nyílt elektronikai platform, aminek célja, hogy könnyen használható hardverekből és szoftverekből álljon. A hardver egy kézzel fogható alkatrész, a szoftver egy írt program. Ezek az Arduino lapok nyílt forráskódúak, ami azt jelenti, hogy szabadon használható, másolható, terjeszthető és módosítható. Ezek könnyen használhatóak, programozhatóak ezért bárki kipróbálhatja: diákok, hobbiprogramozók, programozók, szakemberek. Az Arduino boardok (lapok) bemenetei képesek feldolgozni a külső jeleket, mint például egy mozgásérzékelő jelét vagy hőmérséklet szenzor jelét vagy az én esetemben a feszültséget. A kimenetek segítségével pedig LED-eket lehet fényadásra bírni, LCD kijelzőkre lehet írni vagy akár egy hangmodult is meg lehet szólaltatni. De akár még egy Wi-Fi modul segítségével az internetre is lehet csatlakozni. Több típusa létezik:



3. ábra Arduino MEGA

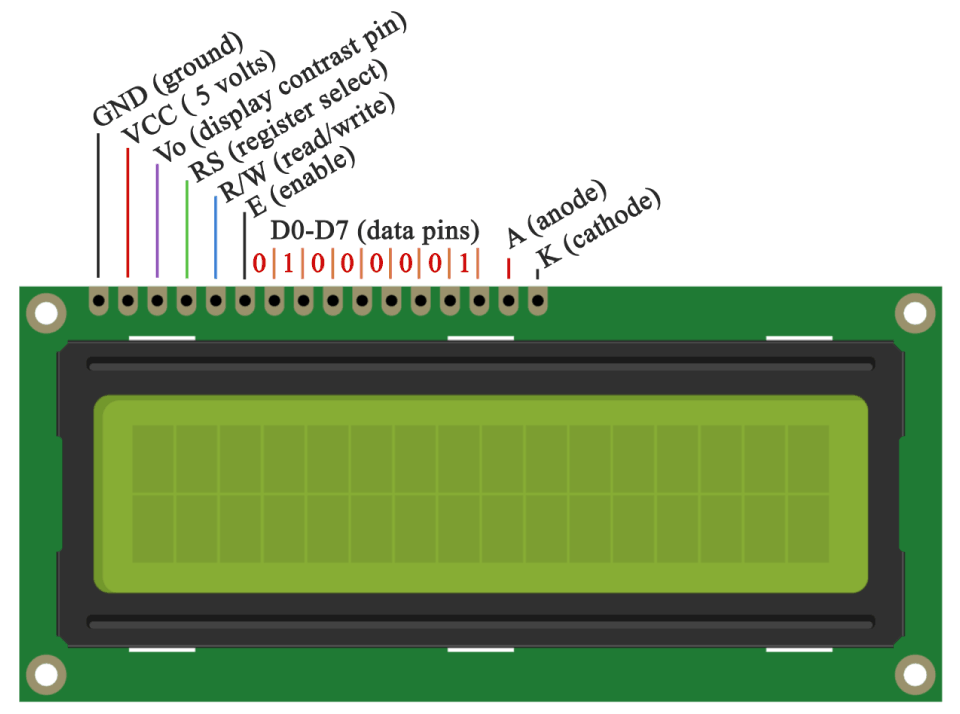
2. ábra Arduino NANO

1. ábra Arduino UNO

Ezeken általában megtalálható egy USB bemenet, ami a tápellátásért és a gép-Arduino kommunikációért felel. Tápellátásnak van még egy bemenete, ami az Arduino Nano típusú boardra nem fért rá, hiszen azért Nano, hogy kisebb helyet foglaljon, kisebb projekteknél is könnyen lehessen elhelyezni. Az Arduinok ajánlott bemeneti feszültsége 7-12V közé esik. Vannak rajta digitális lábak. Ezek bemenetként és kimenetként is használhatók egyaránt, ezen kívül vannak analóg lábai, amik csak bemenetként használhatók, valamint ott vannak még a táplálásra és egyebekre szolgáló lábak. Helyet kap a boardokon még az órajelért felelős quartzkristály, feszültségszabályzó, reset gomb és az egésznek a lelke az IC-k és a hozzá tartozó 2X3 kivezetés ami segítségével frissíteni illetve újratelepíteni lehet az IC-t. Az IC vagyis integrált áramkör félvezető lapon kialakított kis áramkör.

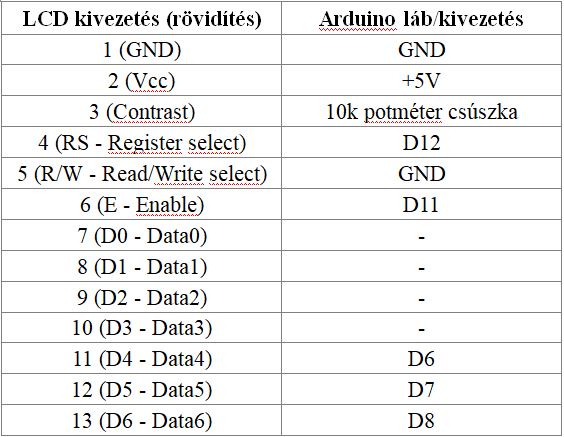
1. ***LCD kijelző***

Az LCD kijelző adatok kiírására használják. Sok helyen lehet vele találkozni például zsebszámológép, digitális karóra, digitális lázmérő.

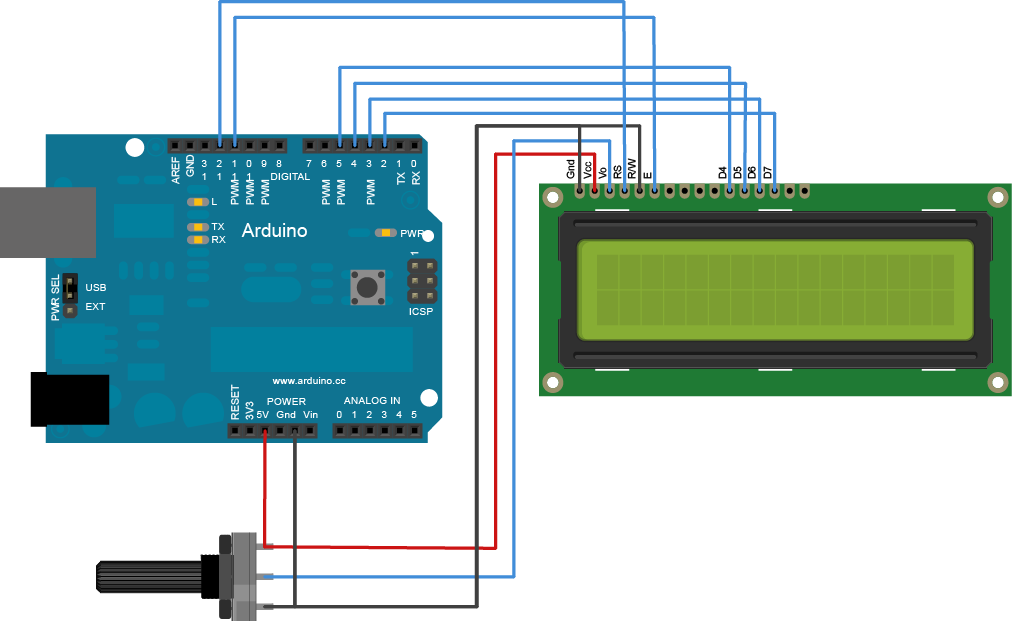


4. ábra 2x16-os LCD kijelző.

**Bekötése:**

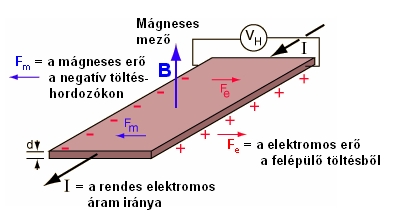
A digitális lábakat használom kimenetként, azon kapja az adatot, hogy mit írjon ki. A kontraszt beállítását egy potenciométerrel oldottam meg, mert így bármikor könnyen meg lehet változtatni a kontraszt mértékét. A potenciométer egy változtatható ellenállás. Végül a betáplálást, vagyis az 5V és a GND-t ráadom.

5. ábra LCD és Arduino kapcsai



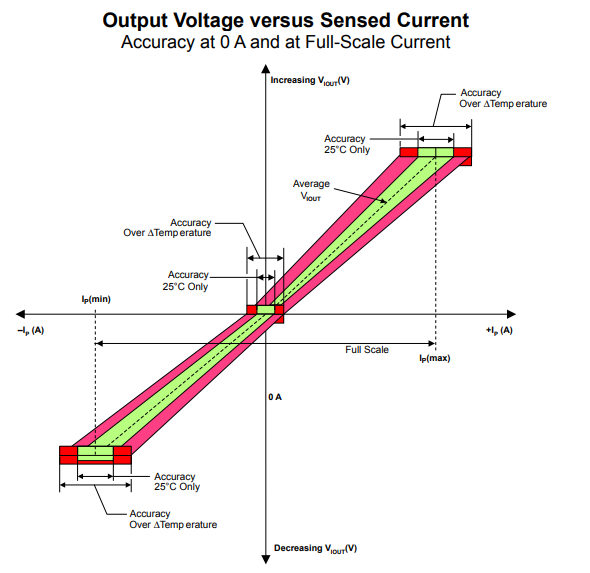
6. ábra LCD bekötése

1. ***ACS712***

Ezt a szenzort az áram mérésére használják. Működése a Hall effektuson alapszik. Ha egy vezetőben áram folyik és az a vezető mágneses térben van, akkor potenciálkülönbség jön létre. Ezt a feszültséget nevezik Hall feszültségnek. Tehát az áramkörünkbe sorba bekötjük a szenzorunkat, rákötjük az 5V és GND-t végül az out lábat bekötjük az Arduino egyik analóg bemenetére és az itt kapott Hall feszültséget az Arduino megméri. Ennek van egy ”mVperAmp” tényezője ami az 5A-es modulnál 185, a 20A-es modulnál 100, a 30A-es modulnál pedig 66. [művelet: Amps = ((Voltage - ACSoffset) / mVperAmp); azaz f(x)=a\*x+b; 1/185\*mért érték – (1/185)\*eltolási érték; a 185 egy korrigálási tényező; az eltolási érték az a szám, amit le kell vonni, mivel 0A-es terhelés esetén 2,5V-ot, azaz 2500mV-ot mérünk az OUTPUT lábon.] Érdekessége még, hogy 25°C-on méri a legpontosabb értéket.

8. ábra Hall effektus

7. ábra ACS712

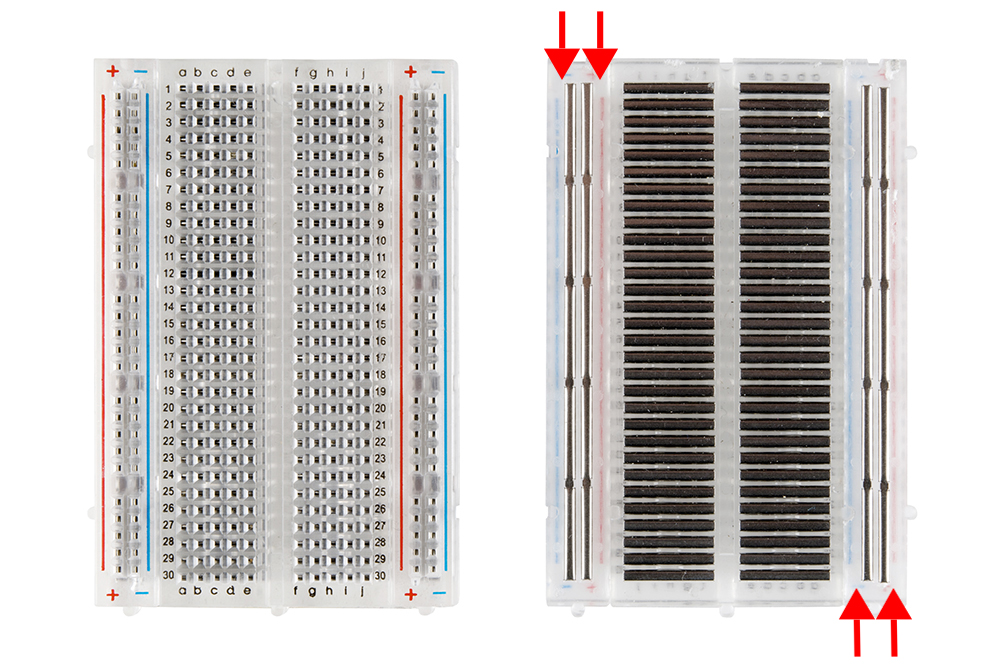


9. ábra ACS712 lábak bekötése

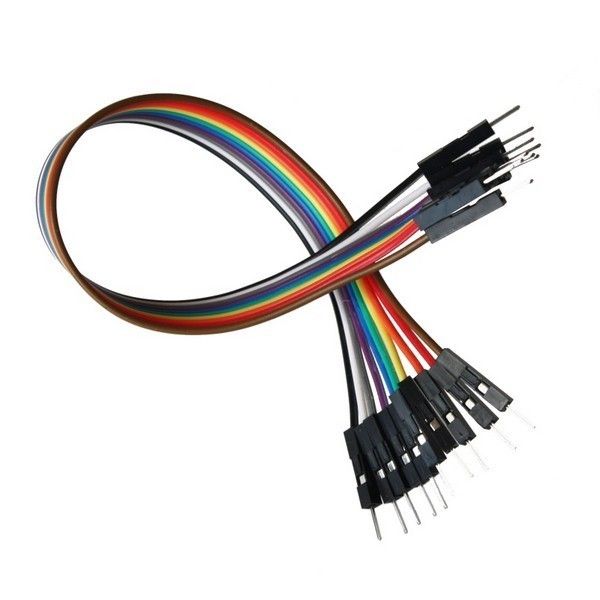
1. ***Egyéb alkatrészek***

**Breadboard:**

Ez az alapja az egésznek. Minden erre van rárakva a feszültségosztótól kezdve az LCD kijelzőn át a potenciométerig. Hátránya, hogy a kötések illeszkedése nem a legjobb, sőt kifejezetten rossz a műszer szempontjából mivel ezzel kisebb-nagyobb eltérések keletkeznek. Hogyha ezt figyelembe vesszük, akkor nagyon pontos. Ez a mérési eredményekből is látszik. Ez tényleg csak arra jó, hogy letesztelje az ember a prototípusát. Később szeretném ezt egy nyáklapra cserélni a végső műszernél.



**Vezeték:**

****A breadboard-hoz, Arduino és az LCD kijelző kapcsaihoz illő vezeték

1. ***A teljesítménymérő működése***

A teljesítménymérő működése a feszültségmérő és árammérőn alapszik, mivel P=U\*I, tehát feszültség szorozva az árammal. Ez valahogy így néz ki a parancsban:

Watts = vin \* Amps;

Ez a parancs egyszerűen csak összeszorozza a kettő értéket. Végül már csak ki kell írni.

lcd.clear(); -Törli az LCD kijelzőt

lcd.setCursor(0, 0); -xy(0,0), legelső pozícióra helyezi a kurzort

lcd.print("INPUT P= "); -Kiírja a kijelzőre az idézőjel közötti részt

lcd.print(Watts); -Megkeresi azt az értéket ami egyenlő a Watts-al és kiírja.

lcd.print(" W"); -Kiírja a kijelzőre az idézőjel közötti részt

delay(2000); -2 másodperc múlva ugrik a következőre

Így végül a kijelzőn megjelenik az ”INPUT P= (a mért érték) W”

Na de akkor ehhez kell a feszültség és az áram is.

A feszültségmérő

vout = (value\*5.0)/1024.0; - 5.0V a referencia feszültség, és 10 bit a chip felbontása (2^10=1024)

vin = vout/(R2/(R1+R2)); - feszültségosztó képlettel kiegészítve a mért érték eltárolva a „vin” változóba. Így ~55V-ig mérhetünk feszültséget. Nem vagyunk az 5V engedélyezett bemenetig korlátozva. A „value” az az érték, amit az „A0” analóg bemeneten mérünk.

Az árammérő

Voltage = (RValue \* 5000) / 1024; - ugyan az a megfontolás, mint a Volt-mérésnél, csak itt mA-ben (milliAmper) történik a számolás..

Amps = ((Voltage - ACSoffset) / mVperAmp); - A valós érték pedig az ACS712 működéséből fakad. Mivel szintén az 5V rendszerfeszültségtől kap táplálást és 0A mért áramnál az Analog bemenetre (A1-re) kötött láb 2,5V-ot, azaz 2500mV-ot ad. Ezt le kell vonnunk a mért értékből. Ez után a Hall elem típusára vonatkozó értékkel kell osztani. Jelen esetben ez 185.

1. ***A teljes program:***

#include "LiquidCrystal.h" /LCD működéséhez szükséges ún. header fájl, majd jön a deklarációs rész

//bekotjuk az LCD-t (RS,E,D4,D5,D6,D7)

const int rs = 7, en = 8, d4 = 9, d5 = 10, d6 = 11, d7 = 12; /LCD lábainak leosztása integer változóba

LiquidCrystal lcd(rs, en, d4, d5, d6, d7); // LCD függvény az inicializáláshoz, felhasználva ehhez a változókat

const int analogInputA = 1; //Árammérésre az Analóg 1-es bemenetet használjuk

const int analogInput = 0; //voltmérésre az A0 bemenetet

float vout = 0.0; //lebegőpontos számok deklarálása a feszültség és egy sorral lejjebb, az árammérés értékeinek számolására, eltárolására

float vin = 0.0; //kezdőérték 0, minden esetben

float R1 = 100000.0; //előtét ellenállások értékei, ezek a feszültségosztó számolásához szükségesek.

float R2 = 10000.0;

float value = 0;

int mVperAmp = 120; // use 100 for 20A Module and 66 for 30A Module miliV/A

float RValue = 0;

int ACSoffset = 2500; //Az ACS712, 0A-nál leadott feszültség értéke

float Voltage = 0.0; //feszültség értékéhez változó

float Amps = 0.0; //áram értékéhez változó

float Watts = 0.0; //teljesítmény értékéhez változó

//---------------- Ezután a setup() szekció jön, amely mindig a futás első lépéseként végrehajtódik

void setup() {

// put your setup code here, to run once:

pinMode(analogInput, INPUT); //ez digitális I/O bemenetnél lenne érdekes, ez törölhető

pinMode(analogInputA, INPUT); //ez digitális I/O bemenetnél lenne érdekes, ez törölhető

lcd.begin(0, 0);

lcd.print("DC WATT-METER");

delay(3000); //késleltetés

}

// a ciklikusan lefutó programrész következik

void loop() {

// put your main code here, to run repeatedly:

//read the value at analog input

// -------

value=0; //kezdeti érték inicializálás, minden ciklusban

RValue = 0; //kezdeti érték inicializálás, minden ciklusban

for (int i=0; i<1000; i++){

value = value+analogRead(analogInput);

RValue = RValue+analogRead(analogInputA);

} // minden értéket mielőtt véglegesítek, ezerszer megmérem és átlagolom a ciklusból kilépve

value = value/1000; // átlagolás utolsó fázisa

RValue = RValue/1000; // átlagolás utolsó fázisa

// feszültség számítás…

vout = (value\*5.0)/1024.0;

vin = vout/(R2/(R1+R2)); //előtéttel módosított képlet

// Amp

Voltage = (RValue \* 5000) / 1024; // feszültséget kapunk

Amps = ((Voltage - ACSoffset) / mVperAmp);

Watts = vin \* Amps; //teljesítmény számítás

lcd.setCursor(0,0); // eredmények kiíratása sorrendben, késleltetésekkel

lcd.print(" ");

lcd.setCursor(0,0);

lcd.print("INPUT U= ");

lcd.print(vin);

lcd.print(" V");

delay(2000);

lcd.setCursor(0,0);

lcd.print(" ");

lcd.setCursor(0,0);

lcd.print("INPUT I= ");

lcd.print(Amps);

lcd.print(" A");

delay(2000);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("INPUT P= ");

lcd.print(Watts);

lcd.print(" W");

delay(2000);

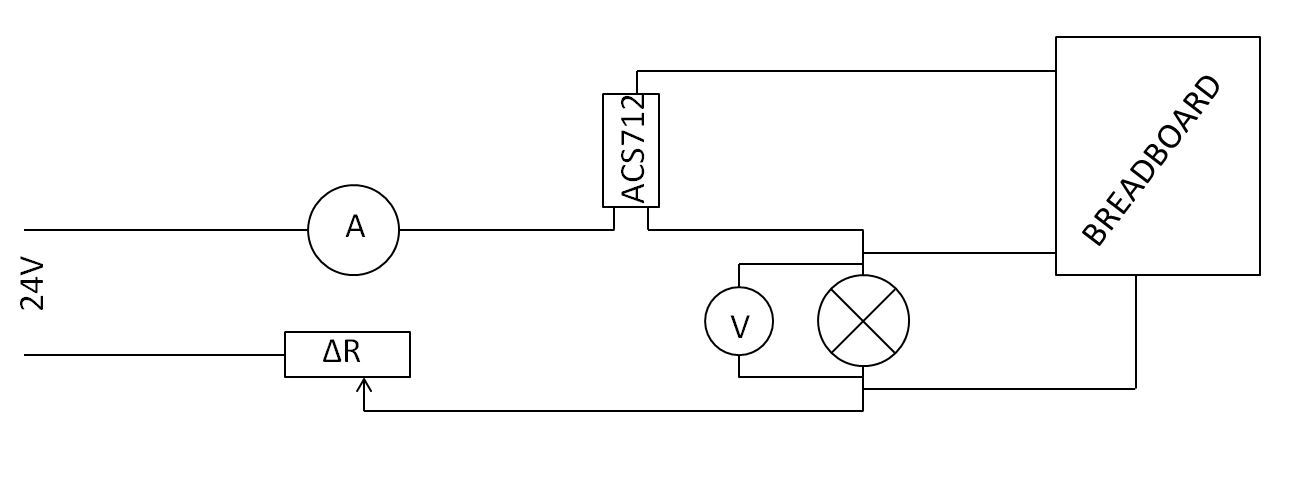
}

1. ***Teszt és ellenőrzés:***

A mért eredményekből látszik, ha összehasonlítjuk a digitális mérőműszer és az általam készített műszer áram értékeit, akkor azt vesszük észre, körülbelül a 6-os mérési pont körül mérünk teljesen azonosnak tekinthető áram értékeket. Előtte ez az érték az ellenőrző műszer értékéhez viszonyítva kevesebb, majd utána több. Az elektronikus ellenőrző műszerünknél meredekebb az Arduino műszerünk jelleggörbéje. A meredekséget alapvetően a gyárilag megadott 185-ös (1/185) érték adja meg, a kezdeti érték eltérését pedig az It=0 értékhez tartozó Ueltolási feszültség (pontosabban Ue/185, lásd excel). Az ellenállás változása a hőmérséklet függésében is befolyásolja a mért eredményeket, mivel olyan változtatható ellenállást használtam, ami könnyen melegszik, illetve az izzó is melegszik. Ezt a későbbiekben szeretném kiküszöbölni.

Mérés értékelés: A mérés során – a mellékletben láthatóan – pontos műszer hiányában a digitális műszer értékeit vettük alapul. Ezáltal a gyárilag megadott 185-ös érték helyett 120-as értékre állítottunk. Ezért a végleges műszer kialakításánál egy pontos műszer alapján állítjuk be. A HALL elem jelleggörbéje alapján a pontosság függ a környezet hőmérsékletétől.

1. ***Ellenőrző mérés kapcsolási rajza***



1. ***Források***

Arduino : https://electro.blog.hu/2017/12/24/mi\_az\_az\_arduino

-https://www.youtube.com/watch?v=BYoA4dnIhhs

Képi :

Arduino UNO -https://www.module143.com/arduino-uno-r3-atmega328p-microcontroller-board

Arduino Nano -https://store.arduino.cc/arduino-nano

Arduino Mega -https://pandaelectronicsbd.com/shop/arduino-mega-2560-rev3/

-LCD kijelző : https://howtomechatronics.com/tutorials/arduino/lcd-tutorial/

acs712 -http://henrysbench.capnfatz.com/henrys-bench/arduino-current-measurements/acs712-current-sensor-user-manual/

Breadboard -https://learn.sparkfun.com/tutorials/how-to-use-a-breadboard/all

vezeték -http://hobbielektronikabolt.hu/spd/CABL001/10-db-Szines-Breadboard-Jumper-Kabel-20cm-Arduino