

UH-zongora - zenélés mikrokontrollerrel

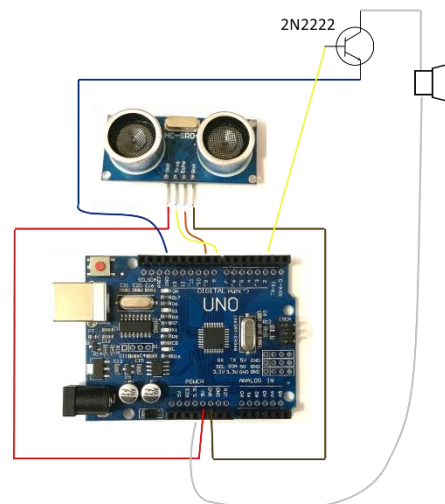
Beszeda Imre: 61. Fizikatanári Ankét, Szeged, 2018.március 14-17, eszközkiallítás

A hangszer alapja egy ú.n. mikrokontroller (Arduino), ami egy ultrahangos távolságmérő szenzorral (HC-SR04) érzékeli, hogy van-e előtte akadály (pl. tenyér). Ha igen, megméri a távolságát, és a távolságtól függő hangmagasságot szólaltat meg: 0-10cm között a C hangot, 10-15cm között a D hangot, és így tovább...

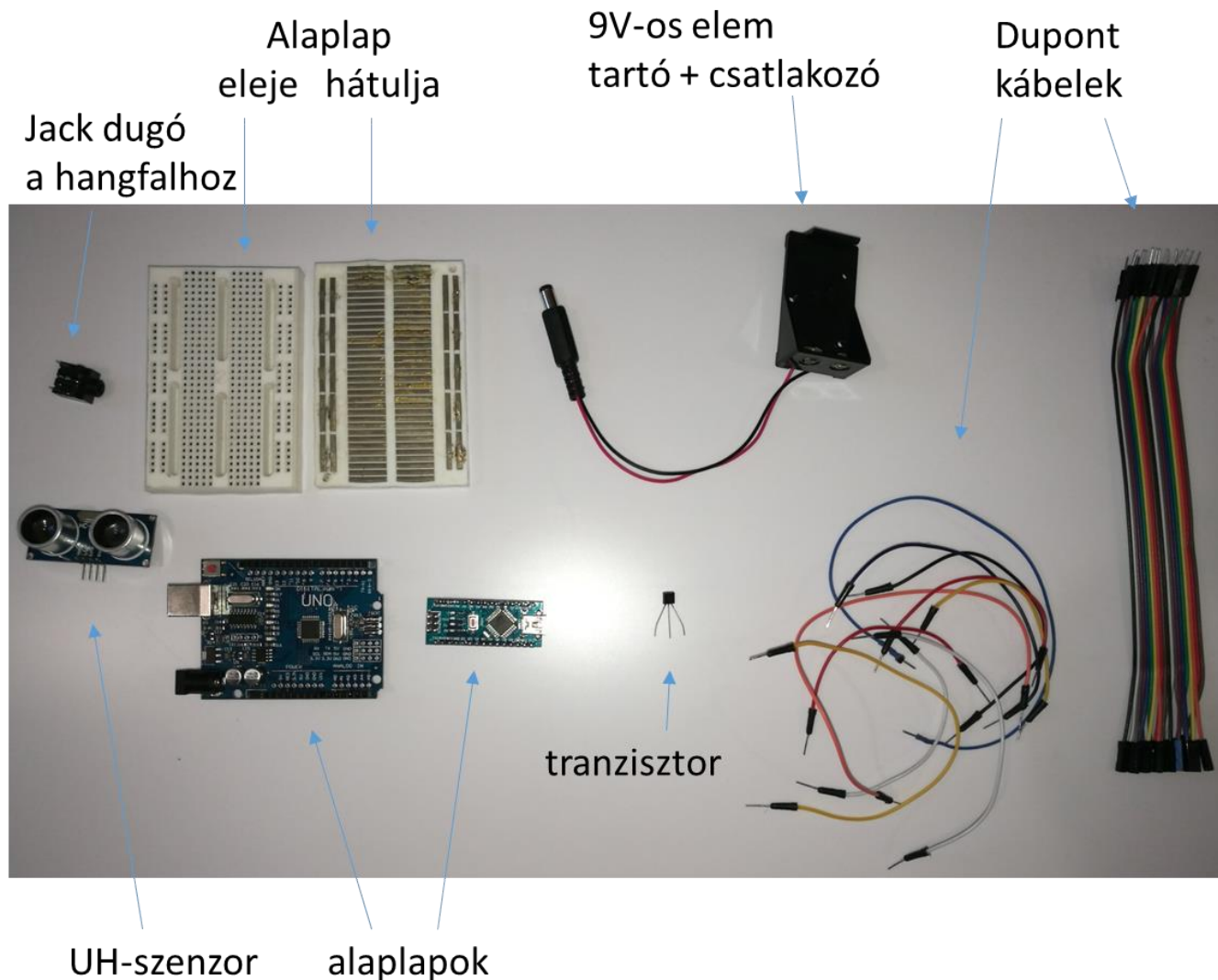
Az áramkör:

Ha nem akarunk nagy hangerőt, akkor egy egyszerű piezo csipogó (piezo-buzzer) is megteszi, ami közvetlenül kapcsolható a digitális kimenet és a GND közé, nincs szükség a tranzisztorra mint erősítőre.

A szükséges eszközöket pl. az eBay-en érdemes beszerezni, mert kb tized annyiért megkapjuk, ingyenes szállítással, cserébe több hetes szállítási határidővel.

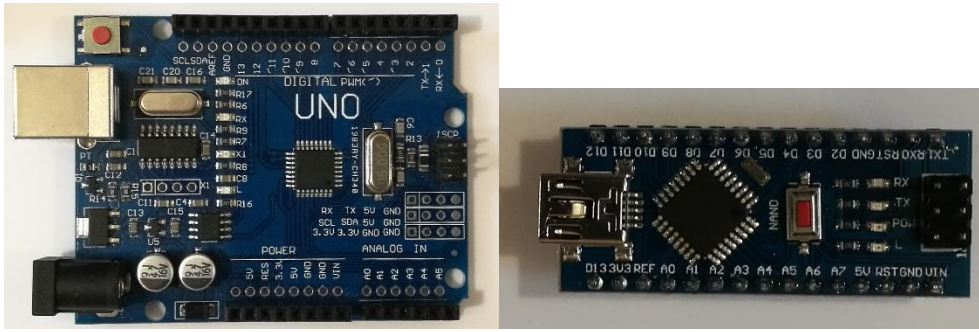


Amire szükségünk lesz/lehet – az alkatrész lista:



Nézzük részletesen:

Mindenek előtt szükséges egy Arduino UNO vagy NANO alaplap (Arduino Uno board, ha pl eBay-en keressük):

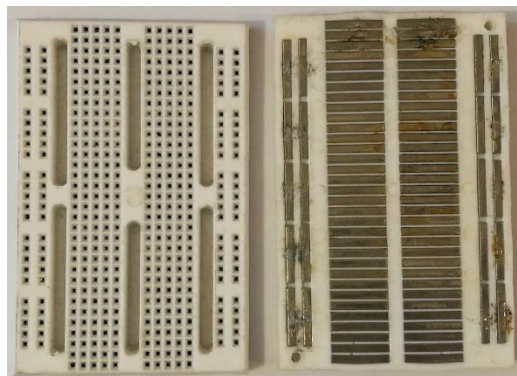


Ha Nano alaplapot veszünk, akkor eleve USB kábellel együtt érdemes megvenni, mert az áramellátása csak azon át oldható meg.

Az Uno alaplap 9V-os elemről is működtethető, így önmagában is használható.

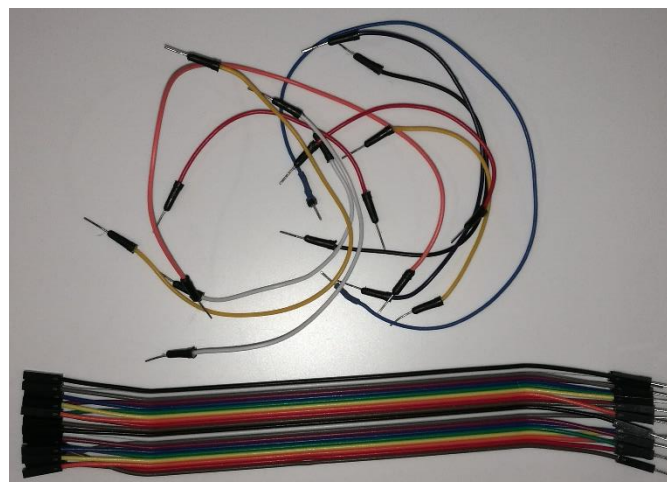
Dugaszolós próbapanel (breadboard):

A bal oldali a szemből nézet, a jobb oldali a hátoldalát mutatja – ezen látszik, hogy mely lyukak vannak egymással összeköttetésben.

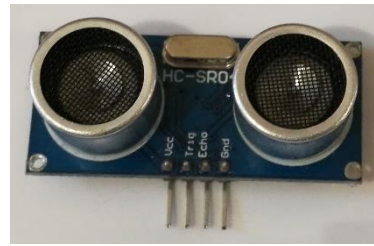


Ettől kisebbek és nagyobbak is kaphatók, a célnak megfelelő pólusszámút érdemes venni.

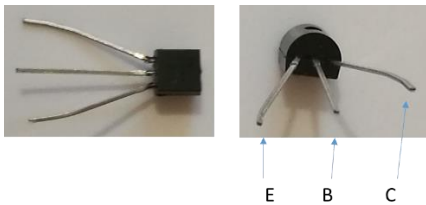
Kell néhány db 10-20cm hosszúságú Dupont jumper kábel (Dupont jumper cable), apa-apa és apa-anyanya végűek is:



A HC-SR04 jelű ultrahang szenzor (ultrasonic sound sensor):



Valamilyen általános tranzisztor, viszonylag nagy terhelhetőségű, pl 2N2222 NPN tranzisztor:



Az áramkör összerakása:

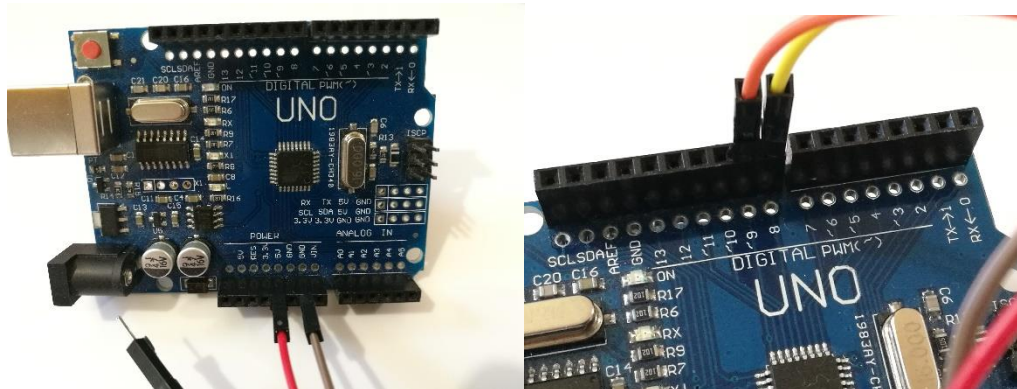
4 db, kb 20cm hosszúságú, apa-anya végű vezetéknek az anya-végeit rányomjuk az UH-szenzor kivezetéseire, az ábrának megfelelő színekkel: piros = +5V, barna (esetleg fekete v. sötétkék) = GND (= föld, vagy nulla), sárga = Trig, narancssárga = Echo.



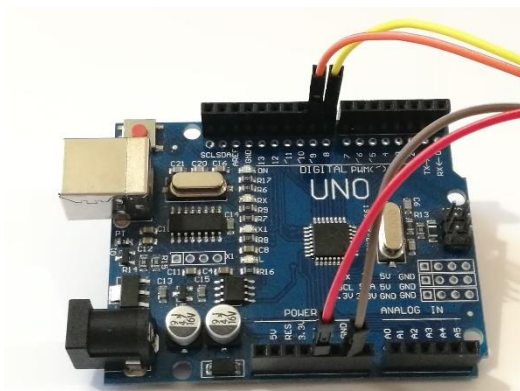
A vezeték másik végeit az alaplap megfelelő ki/bemeneteihez (pin-jeihez) csatlakoztatjuk:

A pirosat a +5V-hoz, a barnát a GND-hez,

a sárgát pl a digitális 8-hoz, a narancsot a 9-hez:



Ezek együtt:

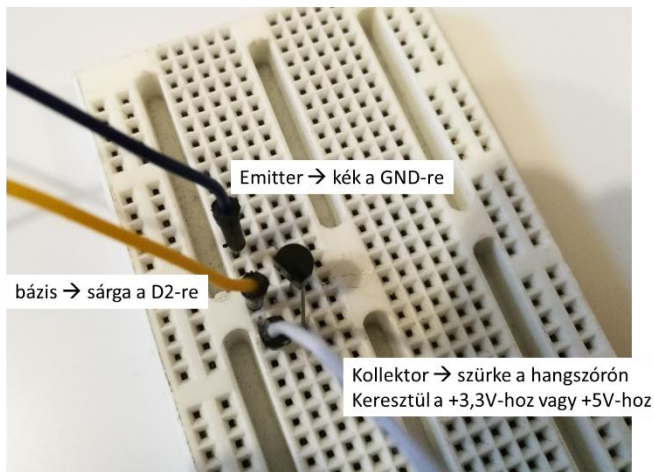
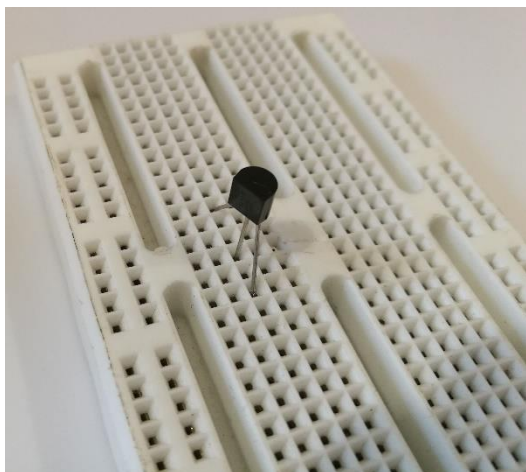


A hangot pl a digitális D2 lábon fogjuk kiküldeni, ekkor a piezo buzzer + lába jön ide, a másik lába a GND-hez.

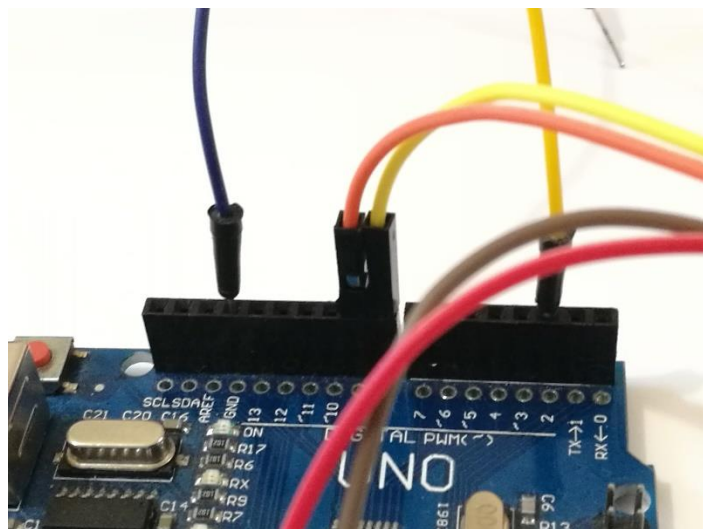
Ha inkább egy komolyabb hangszórót vagy pl 2.1-es hangfalat szeretnénk, amin nagyobb áram folyhat (az alaplap digitális lábai max 40mA árammal terhelhetők, folyamatos terhelésnél inkább csak 20mA), akkor egy tranzisztort kell közébeiktatni kapcsolóként, aminek a bázisára megy a jel, és ez vezérli (kapcsolgatja) az emitter (E)-kollektor (C) közötti áramot, ami a tranzistor megfelelő választásával tetszőlegesen nagyra választható (pl. a 2N2222 kb 0,6A = 600mA áramot visel el).

A tranzistor a próbapanelon:

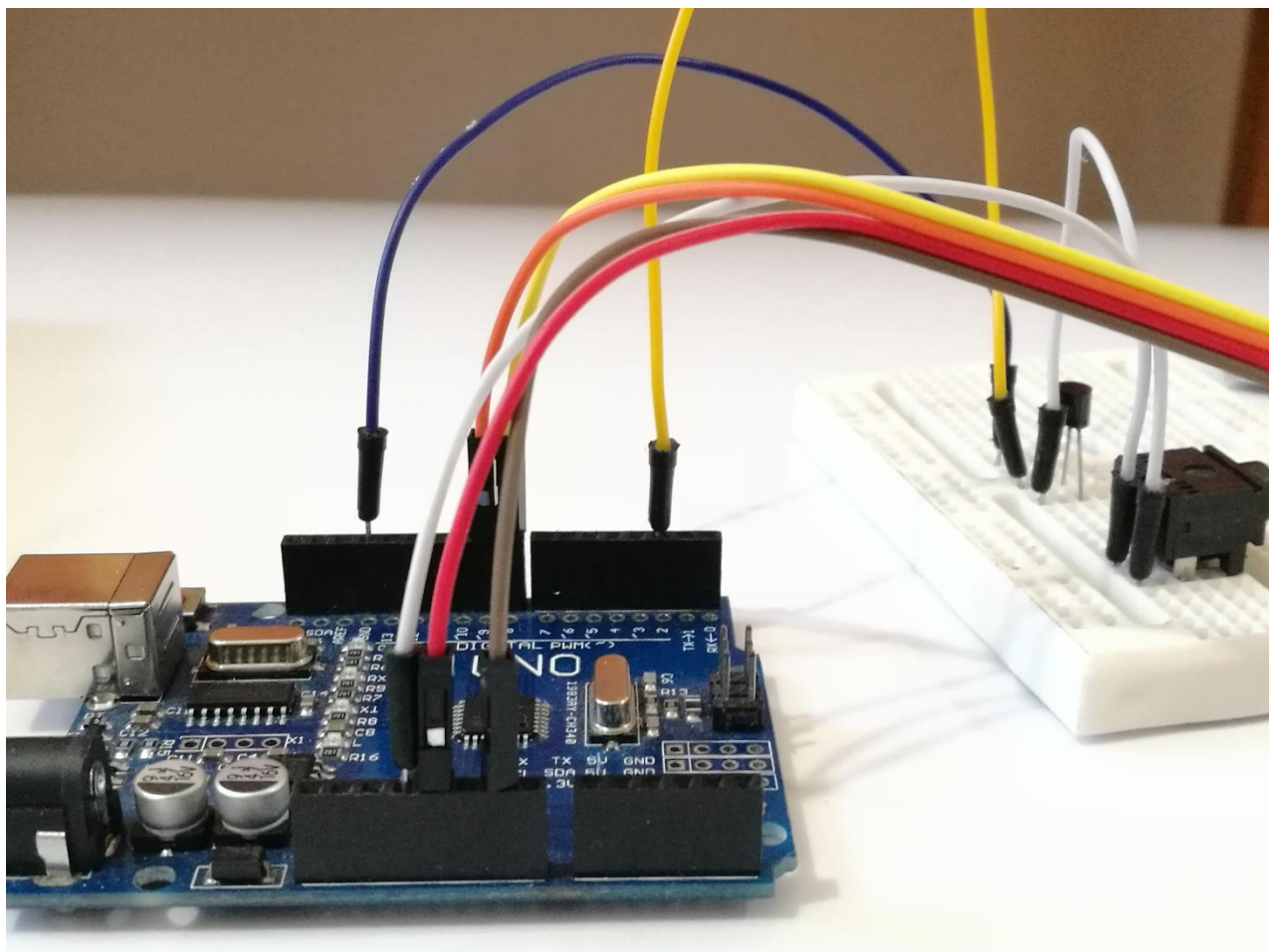
és a hozzá érkező vezetékek:



A vezetékek másik végei az alaplapon:

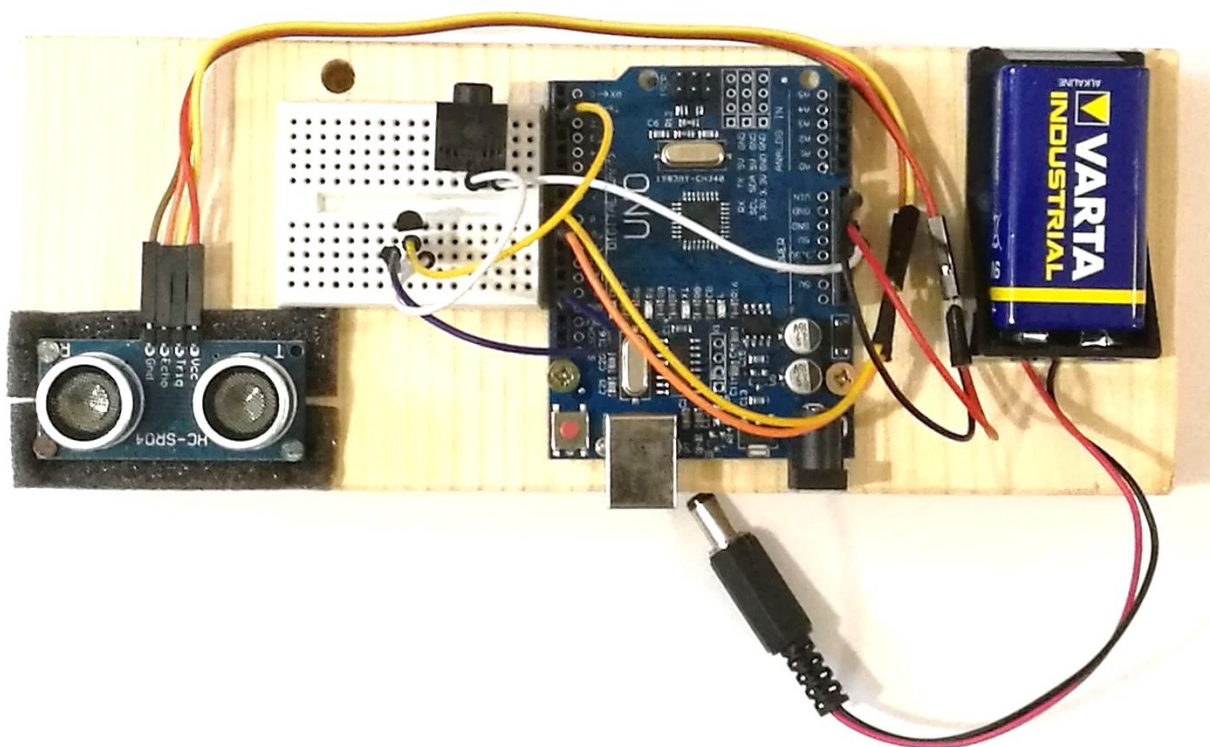


És végül az összerakott teljes áramkör; a szürke (fehér) vezeték a hangszóró csatlakozóján keresztül megy a +3,3V-os csatlakozóhoz:

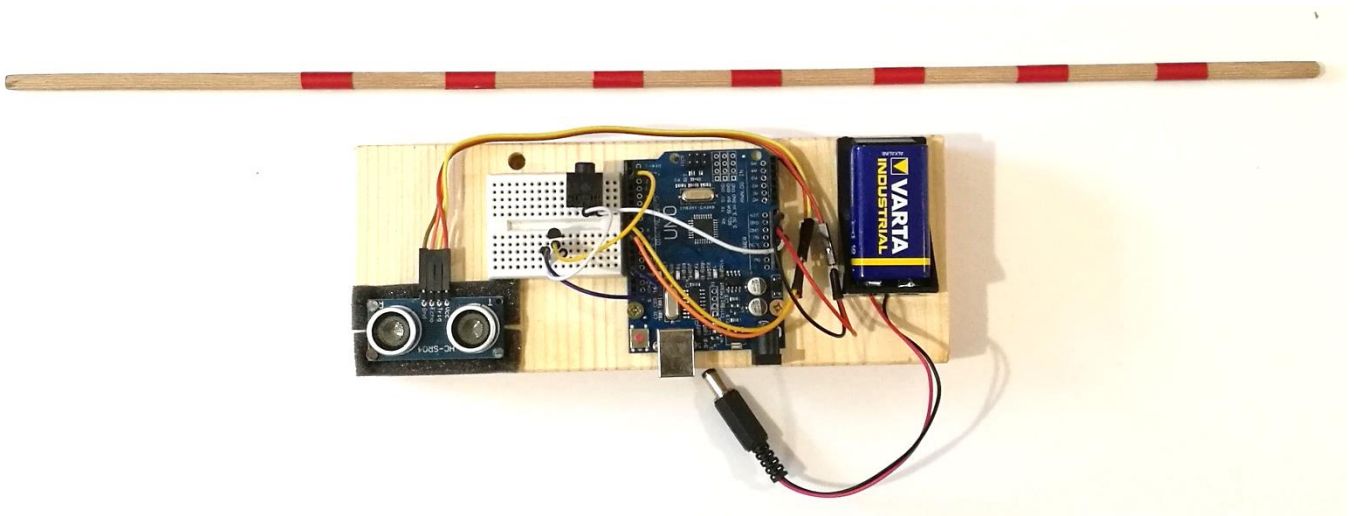


A jack-dugót egy régi hangkártyából forrasztottam ki, de nyilván meg lehet venni.

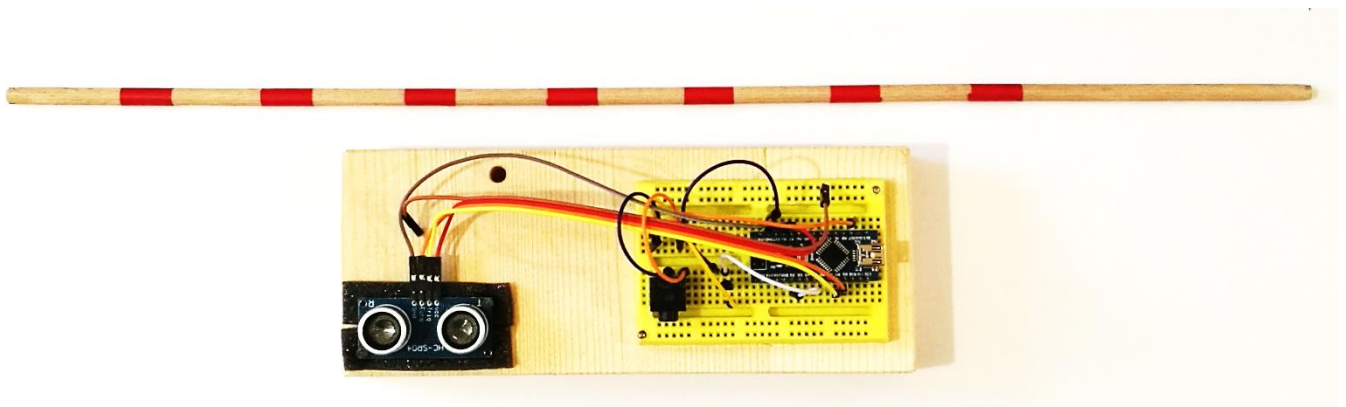
Utána az egészet rárajuk egy lapra, hogy hordozható legyen:



Mivel a távolságmérés a 0 - 45cm közötti tartományra lesz beállítva, ezért készítünk egy ilyen hosszúságú pálcát, 5-5cm-es osztásközzel:



Ugyanez az eszköz NANO alaplappal elkészítve:



A legvégén a magasságjelző pálcát beállítjuk az erre a célra készített furatba:



Ezek után meg kell írni a programot, ami előállít és kiküld egy négyszögimpulzust az UH-szenzor Trig lábára, figyeli az Echo (=visszahang) lábán a visszajövő (az akadályról visszavert) jelet és a hangsebességből kiszámolja a távolságot az UH-szenzortól, és a távolság értékétől függően kiküld a D2 digitális kimenetre egy adott frekvenciát.

A programkód:

A program gyakorlatilag egy C nyelvű kód. A // jelek utáni szöveget a fordító figyelmen kívül hagyja, itt magunk számára helyezhetünk el infókat, megjegyzéseket, hogy pl hetek múlva is emlékezzünk arra, hogy az adott sor vagy adott utasítás mit is csinál. (A program elején /* ... */ jelek közötti rész ugyanezt csinálja, csak itt több egymás utáni sor a megjegyzés.)

A program utasításainak magyarázatát ld. a sorokhoz fűzött kommentekben!

```
/* zenélés HC-SR04 jelű UH szenzorral:
```

```
az UH-szenzor elé helyezzük a tenyerünket: a szenzor 10cm - 45cm között méri a távolságot
és 5cm-es lépésközzel változtatja/szólaltatja meg a különböző magasságú hangokat:
```

```
a távolságmérés elve:
```

```
egy kiküldött ping jel visszaverődik a detektorral szemben levő akadályról, a szenzor veszi a visszavert jelet
a hang 340m/s sebességgel terjed a levegőben; a szenzor méri az oda-vissza terjedési időt
ezen adatokból meghatározható az objektum távolsága
```

```
az áramkör:
```

```
* a szenzor +V lábát rákapcsoljuk az Arduino +5V-ra, GND = föld,
* a TRIG lábát pl. a digital pin 8 - hoz, az ECHO-t pedig a pin 9 - hez,
* a piezo buzzer a GND és a digital pin2 közé megy. (Ha a buzzer helyett hangszórót akarunk, akkor azt egy
 tranzisztossal (pl 2N2222) kell kapcsolgatni, mert a digitális kimenet max 40mA áramot bír ! Ekkor a bázist
 vezérli a pin2-re kimenő jel, az emitter a földön van, a hangszóró pedig a kollektor és a +5V közé.)
*/
```

```
//a 4-es oktáv hangjainak definiálása:
```

```
#define C4 262 //1908,396946564885, pl a C4 hang 262Hz-et jelent (négyszögjel)
#define D4 294 //1700
#define E4 330 //1515
#define F4 349 //1433
#define G4 392 //1276
#define A4 440 //1136
#define B4 494 //1012
#define C5 523 //956
```

```
// konstansok és változók definiálása:
```

```
const int pingPin = 8; // a szenzorra kimenő jelet a 8-as digitális lábán fogjuk kiküldeni
const int echoPin = 9; // szenzor input - a visszajövő jel a 9-es lábára érkezik
const int speakerPin = 2; // hangszóróra kimenő jelet a 2-es digitális lábán küldjük majd ki
long duration, cm; // a „duration” a kimenő és visszajövő jelek közti idő-különbséget jelöli
```

```
void setup() // itt azok a parancsok, beállítások vannak, amiket csak egyszer kell megmondani
{
  pinMode(pingPin, OUTPUT); // itt a pingPin lábát beállítjuk kimenetre
  pinMode(echoPin, INPUT); // az echoPin lábát pedig bemenetre
```



```

}

void loop()    // az itt következő utasításokat folyamatosan ismételteti a végtelenségig
{
// a program működése:
// négyszögjel kiküldése, a visszatérés idejének mérése, ebből távolság kiszámítása, majd a hang megszólaltatása,
// és ugyanezt ismételtjük a LOOP-ban a végtelenségig (amíg áramellátás van)

// 5 mikroszekundum hosszúságú négyszögjel (LOW – HIGH - LOW) impulzusokat küldünk ki.
// Hogy tiszta, egyértelmű HIGH jelünk legyen, ezért először egy rövid (2us-os) LOW kimenetet adunk:
digitalWrite(pingPin, LOW);    // a pingPin lábra LOW (=nulla) kimenetet adunk
delayMicroseconds(2);        // várunk 2 mikroszekundumot (eddig nem csinál semmit a program)
digitalWrite(pingPin, HIGH);   // a kimenetet HIGH-ra állítjuk
delayMicroseconds(5);        // és várunk 5 mikroszekundumot, majd
digitalWrite(pingPin, LOW);    // ismét nullára állítjuk a kimenetet, így kiment az 5us hosszúságú négyszögjel

// a HIGH impulzus akadályig tartó oda-vissza időtartama adja a ping jel kiküldésétől a visszaérkezésig eltelt
// időtartamot (mikroszekundum egységekben); a visszatérő impulzust az ECHO lábon fogadjuk:
duration = pulseIn(echoPin, HIGH);

// az időtartamot átszámítjuk távolságra:
cm = duration / 29 / 2;
// Ezek után a távolság függvényében különböző hangmagasságokat szólaltatunk meg 20ms ideig; a hangot a
speakerPin nevű kimenetre küldjük.
if (cm < 10)    tone(speakerPin, C4, 20); // ha a mért távolság < 10cm, akkor a C4 hangot szólaltatjuk meg
else if (cm < 15) tone(speakerPin, D4, 20); // ha a távolság 10-15cm közötti, akkor a D4 hangot
else if (cm < 20) tone(speakerPin, E4, 20); // és így tovább...
else if (cm < 25) tone(speakerPin, F4, 20);
else if (cm < 30) tone(speakerPin, G4, 20);
else if (cm < 35) tone(speakerPin, A4, 20);
else if (cm < 40) tone(speakerPin, B4, 20);
else if (cm < 45) tone(speakerPin, C5, 20);
}

```

Megjegyzés: A leírás abszolút kezdők számára készült – elnézést azoktól, akik ezen a szinten már túlvannak !