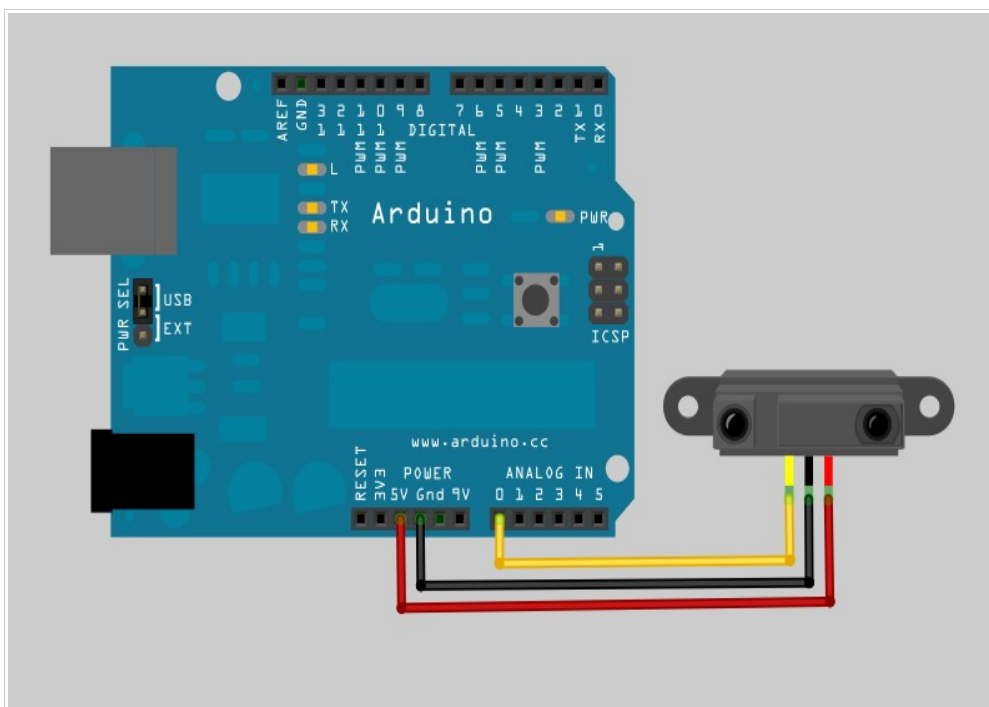




Sharp 3-40 cm, Sharp 10-80 cm si Sharp 15-150 cm

Acesti trei senzori sunt extrem de similari, difera doar gama distantelor in care sunt utili. Conectarea la Arduino este identica, iar codul folosit este aproape identic.

Pentru ca mufa de conectare este mai complicat de folosit altfel, recomand achizitionarea unui cablu special JST, care intra direct in mufa si ofera ca iesire 3 fire (ROSU – alimentare – VCC Arduino, NEGRU – masa – GND Arduino, si ALB – semnal – un pin analogic Arduino).



Imediat dupa ce ai conectat senzorul, urmatorul pas ar fi sa vezi valorile date de senzor, fara a mai aplica nici o procesare suplimentara. Pentru aceasta, incarca pe placa Arduino un program care afiseaza valorile senzorului, ca mai jos.

```
void setup() {
  Serial.begin(9600);
}
void loop() {
  int valoareSenzor = analogRead(0);
  Serial.print("Valoare Senzor: ");
  Serial.println(valoareSenzor, DEC);
}
```

Deschide Serial Monitor (Tools -> Serial Monitor) si apropiind si departand mana de senzor



urmareste ce se intampla cu valorile citite. Pentru toti senzorii vei observa ca atunci cand nu ai nici un obiect in fata senzorului valoarea citita este mica, in zona 10-20-30. Pe masura ce mana ta se apropie de senzor, valoarea incepe sa creasca, pana la aproximativ 630 cand mana ta ajunge la 10 cm de senzor (valoarea pentru senzorul 10 – 80 cm). Daca apropii mana si mai mult, valoarea incepe iarasi sa scada.

Pentru marea majoritate a proiectelor, acest cod este suficient. Sigur, te intrebi probabil "si unde e distanta, ca deocamdata sunt doar niste numere ???". Asa este, ai dreptate. Cea mai simpla varianta ar fi sa-ti etalonezi singur cateva valori care te intereseaza (spre exemplu, masori experimental ca la distanta de 25 de cm ai o valoare citita de 432). O varianta ceva mai complicata este sa analizezi graficul valorilor din datasheet-urile senzorilor si sa faci o determinare matematica prin calcule, alegand puncte de referinta din grafic (folosind de exemplu metoda celor mai mici patrate). De remarcat ca pentru fiecare dintre senzori caracteristica (modul cum variaza valoarea citita de Arduino cu distanta pana la obiect) este diferita.

Pentru senzorul 10-80 cm spre exemplu, functia matematica de mai jos (determinata matematic plecand de la caracteristica din datasheet) aproximeaza distanta in centimetri in functie de valoarea citita de Arduino (este valabila doar pentru senzorul 10-80 cm).

```
int readDistance() {
    float volts = analogRead(0)* ((float) 5 / 1024);
    float distance = 65*pow(volts, -1.10);
    return distance;
}
```

Datorita faptului ca emit lumina infrarosie foarte des, in mod automat senzorii Sharp sunt destul de mari consumatori de curent si sunt afectati de perturbatii in sursa de alimentare (cum ar fi, de exemplu, situatia unui robot alimentat cu o baterie mai putin capabila – sa spunem o baterie dreptunghiulara de 9 V – si la care atata motoarele cat si senzorii merg pe aceeasi baterie). Intr-o situatie de acest gen, valorile indicate de senzori pot fi mai mult sau mai putin eronate. Pentru a evita acest lucru, este bine ca intotdeauna senzorii sa fie alimentati separat (dintr-o alta baterie sau sursa de energie) decat motoarele sau alti mari consumatori de energie. Alta abordare este sa faci medierea valorilor citite de la senzor, ca mai jos.

```
int readDistanceMediata() {
    int sum = 0;
    for (int i=0; i<10;i++){
        float volts = analogRead(0)* ((float) 5 / 1024);
        float distance = 65*pow(volts, -1.10);
        sum = sum + distance;
        delay(5);
    }
    return (int)(sum / 10);
}
```

In acest exemplu am ales sa mediez 10 valori ale senzorului, lucru care ajuta foarte mult pentru situatiile cand valorile citite de la senzor nu sunt stabile. Evident, poti alege sa mediezi un alt numar de valori. Cu cat mediezi mai multe valori, cu atat senzorul este mai stabil, dar mai lent. Cu cat mediezi mai putine, cu atat senzorul este mai rapid.



Alte abordari pentru a face ca citirile senzorilor sa fie cat mai exact (abordari la nivel electric, de data aceasta) sunt sa adaugi in circuit unul sau doi condensatori, cu rolul de a netezi fluctuatiile tensiunii de alimentare.

De ce nu merge ?

–Asa cum spuneam si mai sus, senzorii Sharp sunt mari consumatori de curent. Daca sursa ta de alimentare nu este suficient de capabila, atunci valorile citite de senzori nu vor fi corecte. Spre exemplu, daca ai Arduino alimentat doar prin USB, atunci vei obtine valori mai mici decat cele reale. De asemenea, daca ai alimentat din aceeasi sursa de tensiune si motoarele robotului tau, si Arduino, iar sursa ta de alimentare nu este foarte capabila, atunci vei observa ca valorile citite de senzori sunt afectate de activitatea motoarelor (spre exemplu, cand motoarele pleaca de pe loc, acestea consuma mult curent, lasand senzorii fara curent). Exista mai multe solutii pentru a corecta aceasta problema. Cea mai simpla este ca intotdeauna sa alimentezi Arduino separat si motoarele separat, folosind o sursa de alimentare capabila (in mod normal, 6 baterii R6 de 1.5V fiecare, sau o baterie LIPO reprezinta surse de alimentare capabile; o baterie patrata de 9V inasa, NU este niciodata o sursa capabila). Alta varianta este ca pur si simplu sa mediezi valorile intoarse de senzori (vei avea o latentă in citiri, dar erorile vor fi diminuate). In sfarsit, cea de-a treia solutie este din zona hardware si presupune conectarea in paralel cu alimentarea senzorului a unui condensator electrolitic de cativa zeci de microfarazi. Daca vrei sa aprofundezi aceasta abordare, iti recomand o cautare folosind Google dupa "sharp sensor capacitor filtering".