

Height measurement device

Introducere

Proiectul reprezintă un dispozitiv care poate **măsura** cu acuratețe **înălțimea** persoanei care stă sub acesta.

Mi s-a părut interesantă folosirea **senzorilor** într-un mod care ține de mai mult decât doar mișcare, iar astfel obținem o unealtă folositoare pentru **colectarea de date**, fie într-un scop mai larg (măsurarea mediei de înălțime a unui sample group) fie unul mai simplu (observarea creșterii unui copil și asigurarea că aceasta are loc în parametri normali). Aceste date vor putea, după, să fie accesate și organizate printr-o **aplicație**.

Descriere generală



Inițial, **senzorul laser** va înregistra înălțimea la care se află față de sol. Printr-un calcul de forma $h_{\text{sol}} - h_{\text{obiect}}$ vom obține înălțimea dorită (h reprezintă distanța de la senzor la sol, respectiv obiect). La plasarea unui obiect/individ sub senzor, **led-ul** și **buzzer-ul** se vor activa, indicând detecția acestuia, iar **display-ul** va afișa înălțimea. Prin apăsarea unui **buton**, se vor trimite datele obținute prin **Bluetooth** pe aplicație, unde vor putea fi accesate în mod convenabil. **Microcontroller-ul** este center piece-ul ce leagă toate acestea.

Ideal, realizez cererea datelor tot prin aplicație, deoarece poziționarea butonului ar putea fi inconvenientă. În acel caz, butonul ar putea fi folosit pentru resetarea înălțimii față de sol în caz că dispozitivul este mutat.

Hardware Design

- [Arduino UNO](#)
- [senzor ToF VL53L0X](#)
- [display led](#)
- [modul Bluetooth HC-05](#)



Pinii legați sunt astfel:

Piesă	Legături
LCD	GND → GND VCC → 5V SDA → A4 SCL → A5
HC-05	GND → GND VCC → 3.3V TXD → D2 RXD → D3
Senzor	GND → GND VIN → 5V SCL → SCL SDA → SDA

Software Design

Deoarece am folosit un microcontroller Arduino am lucrat în Arduino IDE și am descărcat biblioteci pentru display-ul LCD și pentru senzor.

Implementarea este simplă: se realizează o măsurătoare inițială, pentru a determina înălțimea la care ne aflăm, iar apoi calculăm constant înălțimea obiectului de sub senzor, dacă se află ceva. Prin aplicație, când dorim să obținem înălțimea, la apăsarea unui buton va apărea un pop-up cu rezultatul obținut pe aplicație. Aceste date după aceea vor putea fi vizualizate într-o listă de date.

[proiect.ino](#)

```
#include "Adafruit_VL53L0X.h"
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>
Adafruit_VL53L0X lox = Adafruit_VL53L0X();
LiquidCrystal_I2C lcd(0x27, 16, 2);
VL53L0X_RangingMeasurementData_t sup_measure;
SoftwareSerial mySerial(2, 3); // RX, TX
void setup() {
  Serial.begin(9600);
  while (!Serial) {
    delay(1); // wait for serial port to connect. Needed for native USB
  }
  // bluetooth init
  pinMode(12, OUTPUT);
  digitalWrite(12,HIGH);
  mySerial.begin(9600);
  //lcd init
  lcd.begin(16, 2);
  lcd.backlight();
  // senzor init
  pinMode(13, OUTPUT);
  digitalWrite(13,HIGH);
  if (!lox.begin()) {
    Serial.println(F("Failed to boot VL53L0X"));
    while (1);
  }
  // masuratoare initiala a suprafetei
  lox.rangingTest(&sup_measure, false);
  if (sup_measure.RangeStatus != 4) {
    Serial.print("Distance (mm): ");
    Serial.println(sup_measure.RangeMilliMeter);
  } else {
    Serial.println(" out of range ");
  }
  lcd.setCursor(0, 0);
  lcd.print("dist sup: ");
  lcd.setCursor(11, 0);
  lcd.print(sup_measure.RangeMilliMeter / 10);
  delay(5000);
}
void loop() {
```

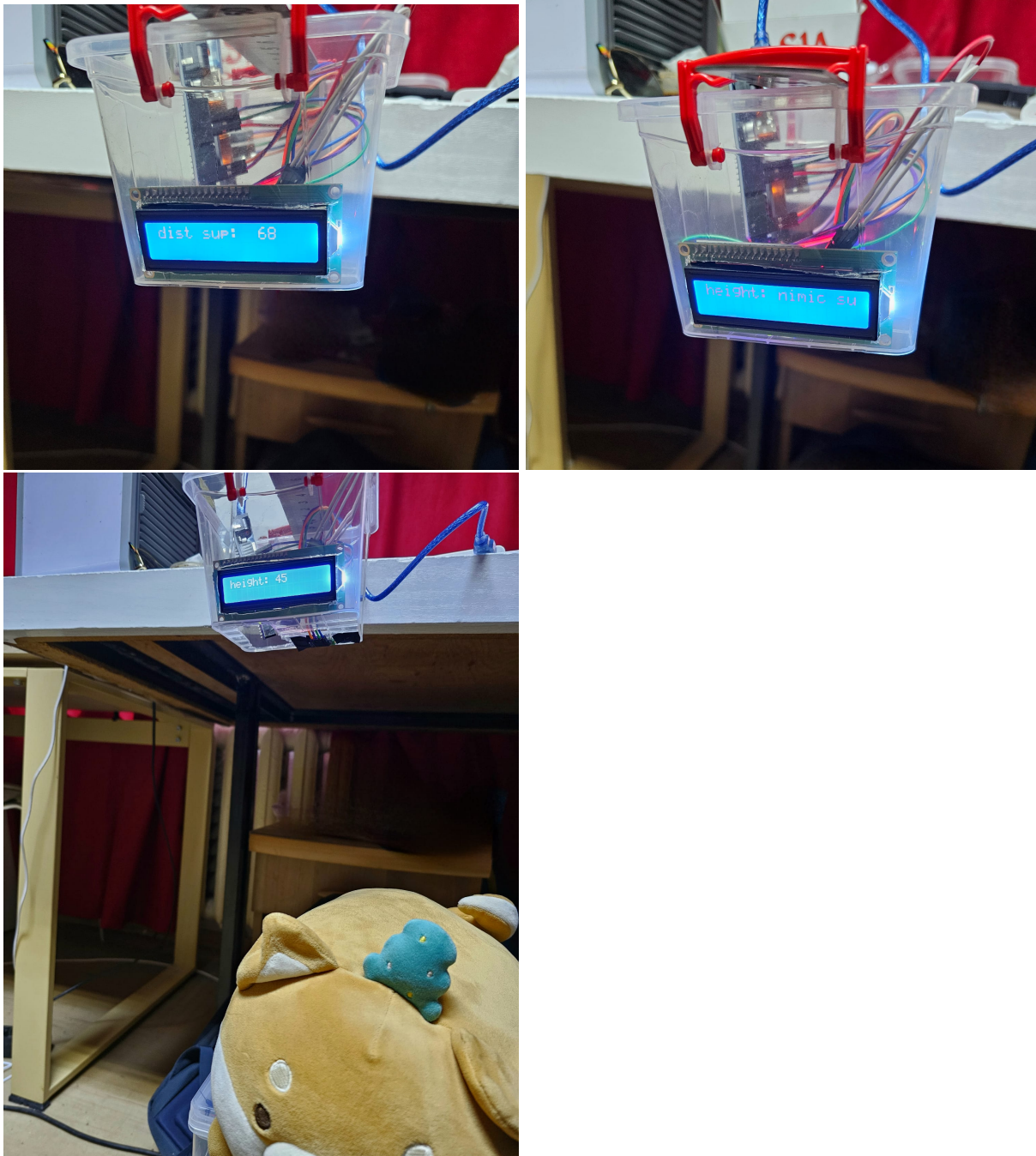
```
VL53L0X_RangingMeasurementData_t measure;
lox.rangingTest(&measure, false);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("height: ");
lcd.setCursor(8, 0);
int heightDiff = (sup_measure.RangeMilliMeter -
measure.RangeMilliMeter) / 10;
if (heightDiff > 0)
    lcd.print(heightDiff);
else
    lcd.print("nimic sub");
if (mySerial.available()) {
    char c = mySerial.read();
    Serial.println(c);
    if (c == 'l'){
        Serial.println("trimit inaltime");
        mySerial.println(heightDiff);
    }
    Serial.write(c); // send from Bluetooth to Serial
    //Serial.print("Bluetooth received: ");
}
if (Serial.available()) {
    char s = Serial.read();
    mySerial.print(s); // send from Serial to Bluetooth
    //Serial.print("Serial received: ");
}
delay(1000);
}
```

Aplicația a fost realizată cu MIT App Inventor. Mai jos sunt interfața, respectiv schema bloc folosită pentru funcționalități.



Rezultate Obținute

Aparatul realizează măsurători față de obiectele aflate sub, însă senzorul fiind destul de mic e fie nevoie de o poziționare foarte bună a obiectelor mai subțiri fie nu par să fie bine percepute.



Din nefericire, senzorul de distanță de **ALLEGEDLY** 2m era doar de 1m. Prin urmare putem măsura copiii sub 4 ani sau obiecte random din casă. Nu am un copil la îndemână, deci am improvizat.

Concluzii

Overall, a fost o experiență nice cu un rezultat tangibil, pe care intenționez să-l expandez în viitor. Dacă obțin un senzor care chiar își respectă descrierea, intenționez să fac o combinație și cu un cântar pentru un full-on BMI calculator.

M-am distrat și cu crearea aplicației și încercarea de unelte noi, out of my comfort zone.

Download

[heightmeasurement.zip](#)

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună 😊.

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul).
Exemplu: Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

Bibliografie/Resurse

Am folosit mai multe resurse în proces dar acestea au fost cele mai folositoare:

[Resursă proiect](#) [Resursă aplicație](#)

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/tdicu/maria.movileanu>



Last update: **2024/05/27 06:58**