Digital Meter

Ufuk BAL, Hasan Ali ÖZKAN, Mehmet Kadri GOFRALILAR, Hasan ARCAS

Abstract:

Measuring distances between two points is usually troublesome for people, for example, a person might want to measure the height of the room, or the distance between two balconies, or distance between two buildings and so on. For safety reasons, especially last two situations are quite problematic. Also after measuring, writing down these measurements in order to keep record is yet another task itself. This task requires you to carry a pen and a paper (or your mobile could work as well). But our solution will bring a new approach to these kinds of tasks.

Our solution consists of Arduino elements in order to build the device, C programming language using Arduino IDE [1], which is used in order to make the Arduino elements work, and finally, a mobile app written in Dart programming language by using Flutter [2] in order to interact with users.

Usage of this system is quite easy. In order to measure the distance between points (x, y); place the device in the x point and then point the laser to the y point by using joystick. Then, with one click, you can measure the distance and keep record of the measurements in the application.

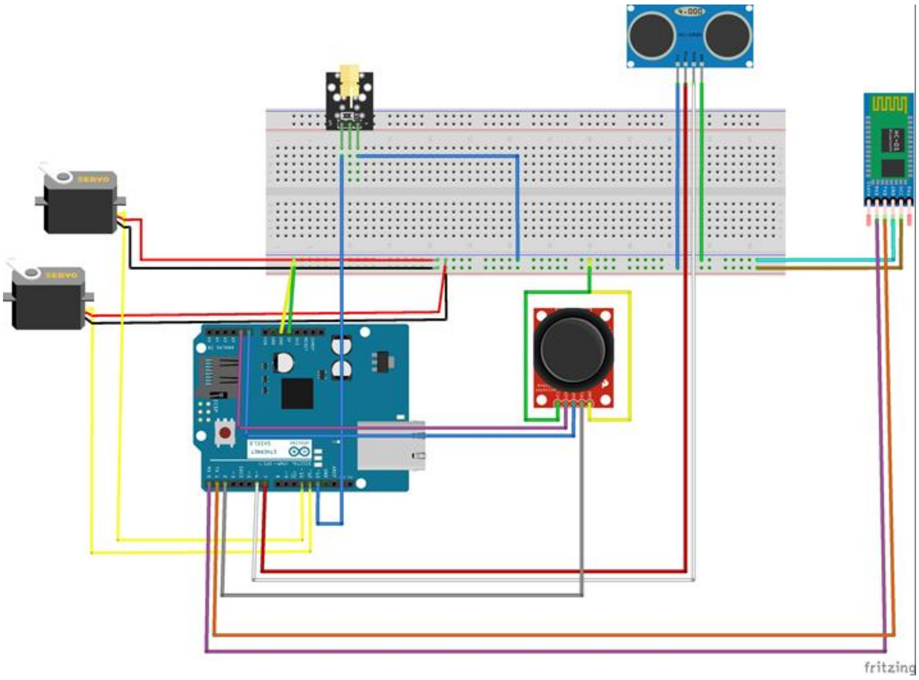
Keywords: Distance, Digitality, Meter, Measurement, Arduino [3], Mobile Application

1. Introduction:

Measuring distances can be difficult or bothersome for many people. Nowadays, everyone prefers handling this kind of tasks from where they are sitting. Accuracy of the measurement is also an important matter. Since it is harder to measure in an accurate way by using conventional methods, a digital method would possibly be more reliable and faster.

Nonetheless, the main disadvantages of the existing method can be summarized as follows: (a) measurements are usually not too precise; (b) task itself is bothersome; (c) users need to use a pen and a paper in order to write down the measurements to remember; (d) users might need to measure a distance between two points where the second end is out of reach (such as a ceiling, distance between two balconies etc.). Therefore, we decided that a digitalized method should be developed to handle with these disadvantages.

2. Modeling:



- Method:

- Components:

- Arduino UNO [4]: Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button.

- 2 Servo Motors (micro servo 9G SG90) [5]: A Servo Motor is a small device that has an output shaft. This shaft can be positioned to specific angular positions by sending the servo a coded signal. As long as the coded signal exists on the input line, the servo will maintain the angular position of the shaft.

- Joystick (2 dimensional) [6]: Arduino joystick is a device that can be used to measure the X-axis, Y-axis and Z-axis direction. It is also called the game console. It can be considered as a combination of a potentiometer and one button.

- Bluetooth Module HC06 [7]: HC-06 is a Bluetooth module designed for establishing short range wireless data communication (<100 meters).

- Laser Diode [8]: This module emits a small intense focused beam of visible red light. This module can be used in laser toys, wireless telecommunications, signal equipment, etc. Operating voltage: 5V. Required current for module to be turn on: Less than 40mA. Output power: 5mW.

- Ultrasonic Sensor HC-SR04 [9]: The HC-SR04 ultrasonic sensor uses SONAR to determine the distance of an object just like the bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package from 2 cm to 400 cm or 1" to 13 feet.

- Breadboard [10]: A breadboard is a solderless construction base used for developing an electronic circuit and wiring for projects with microcontroller boards like Arduino.

- Process:

We have combined 2 Servo Motors by using a glue and we have fixed the combined motors into our Breadboard in order to make the system more stable in a surface.

We have pasted together the laser diode and the head part of one of the servo motors and we have used glue again in order to paste the ultrasonic sensor on top of the laser diode.

In order to make all the parts communicate with each other we have used a Breadboard and used several electrical wires to connect the components to the breadboard.

Make the components communicate with each other was an important step, because some of the components work together, for example when we move the joystick the servo motors should rotate in order to direct the laser diode, which will turn on and off when we click the joystick.

After connecting the several parts to the breadboard, we connected some wires from the components into the Arduino UNO because the "brain" of the system is embedded inside it.

Finally we've connected the Bluetooth Module into the system in order to take the measurement made from the Ultrasonic Sensor and send these results into our Android Smartphone.

3. Implementation:

```
#include <SoftwareSerial.h>
#include <Servo.h>

long duration, cm, inches;
SoftwareSerial BTserial(0,1); // RX | TX

Servo s1,s2;
void setup() {
  pinMode(6,INPUT); //  Input pin for distance sensor.
  pinMode(7,OUTPUT); // Output pin for distance sensor.
  s1.attach(11);  //defining the pins for the servo's
  s2.attach(12);  //defining the pins for the servo's
  pinMode(2,INPUT);   //SW pin status
  digitalWrite(2,HIGH);
  Serial.begin(9600);  //Start Seria Monitor
  BTserial.begin(9600);
}


int p1=90,p2=90;   // initial position of both servos in degree
```

```
//This function is used for move to servo motors.
int change(int pos,int t)
{
    pos=pos+t;                    // Increment/decrement the Servo angle
    if (pos>180)                 //  maximum anlgle of servo is 180 degree
    pos=180;
    if(pos<0)                   // minimum angle of servo is 0 degree
    pos=0;
    return(pos);                //return the change of position
}

//This function is used for getting distance and convert it to cm.
int distance(){

  digitalWrite(7, LOW);
  delayMicroseconds(5);
  digitalWrite(7, HIGH);
  delayMicroseconds(10);
  digitalWrite(7, LOW);
  pinMode(6, INPUT);
  duration = pulseIn(6, HIGH);
  cm = (duration/2)*0.0343;
  delay(250);   //for stability

  return cm;
}
```

```
void loop()
{

  int t1=0,t2=0;  //rate of increment/decrement of angle
  int a=analogRead(A0); // reads analog x-y readings of joystick
  int b=analogRead(A1);
  int dist = digitalRead(7);

  //when joystick is moved away from the center
  if(a<=450 or a>=550)
  {
    t1=map(a,0,1023,10,-10);
    p1=change(p1,t1);  //change the servo's current position

  }
  if(b<=450 or b>=550)
  {
    t2=map(b,0,1023,-10,10);
    p2=change(p2,t2);  //change the servo's current position

  }
   s1.write(p1); // rotate the servo's if the joystick is moved
   s2.write(p2);


 if(digitalRead(2)==LOW){
  int a = 0;
   for(int i =0; i< 5; i++){
    a = distance();
    for(int j = 0; j< 60; j++){
      Serial.println("\n");
    }
  }
  Serial.println(a);
}
  delay(50);          // for Stability
}
```

We have included the libraries that we are going to use, and we have defined the variables for the different measurements, the Bluetooth module and the 2 Servo Motors.

In the “ setup() ” part we have declared the different pins that we have used to connect the components into our Arduino and finally we have finished the setup of our Bluetooth module.

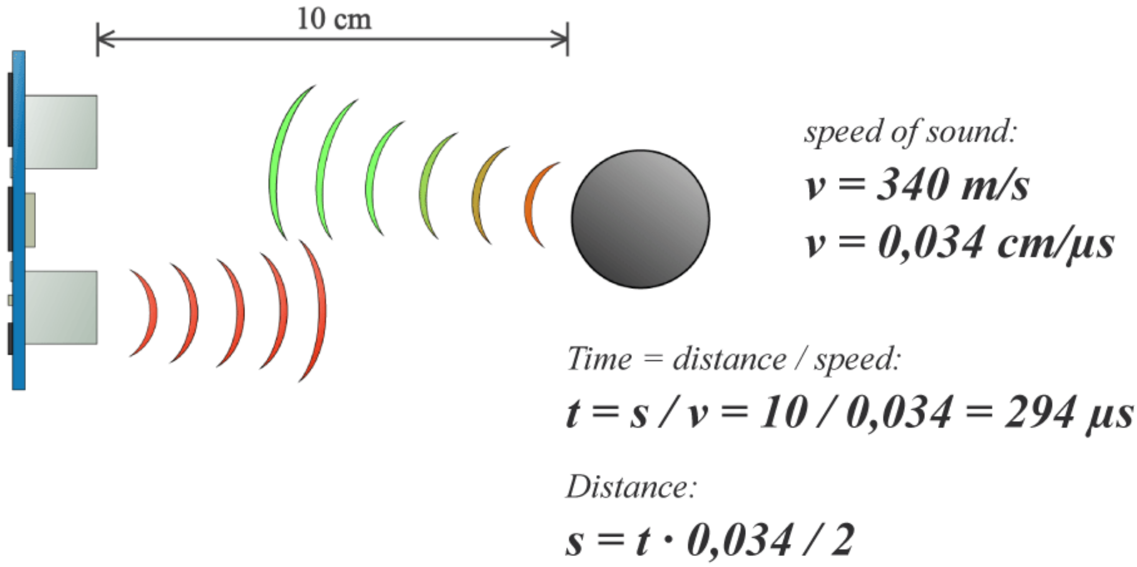
We have declared an initial position for our Servo motors in terms of degrees.

In the “ loop() ” part we are continuously reading the data received from the joystick (x and y positions) and we read the data received from the Ultrasonic Sensor.

After that we check the angle value received from the joystick and we decide which Servo motor to move based on the angle value.

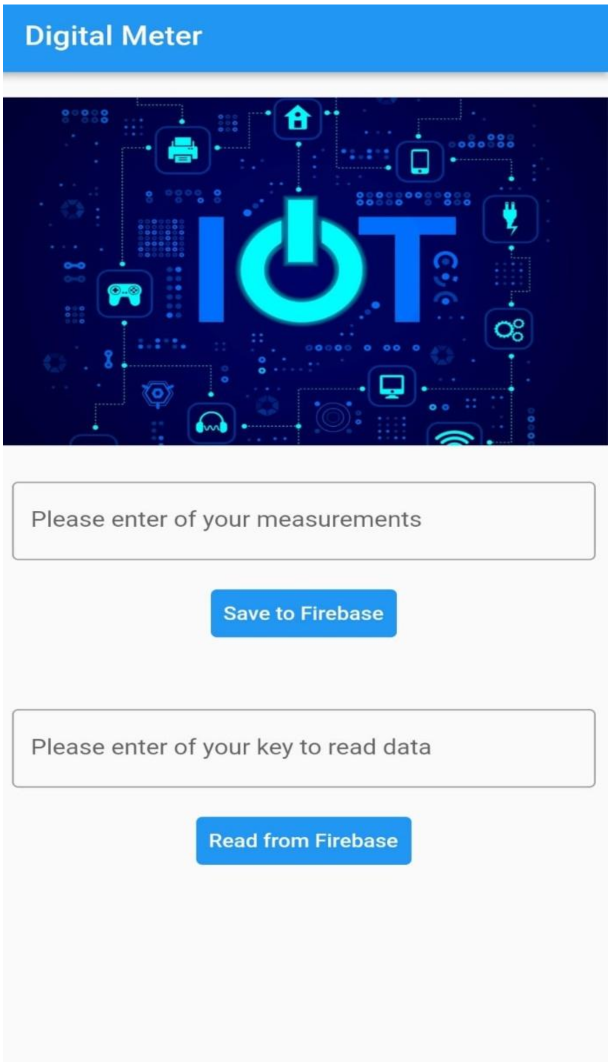
After that we read if the user clicks into the joystick, and if it is that the case, we change the status of the laser diode from ON to OFF or from OFF to ON.

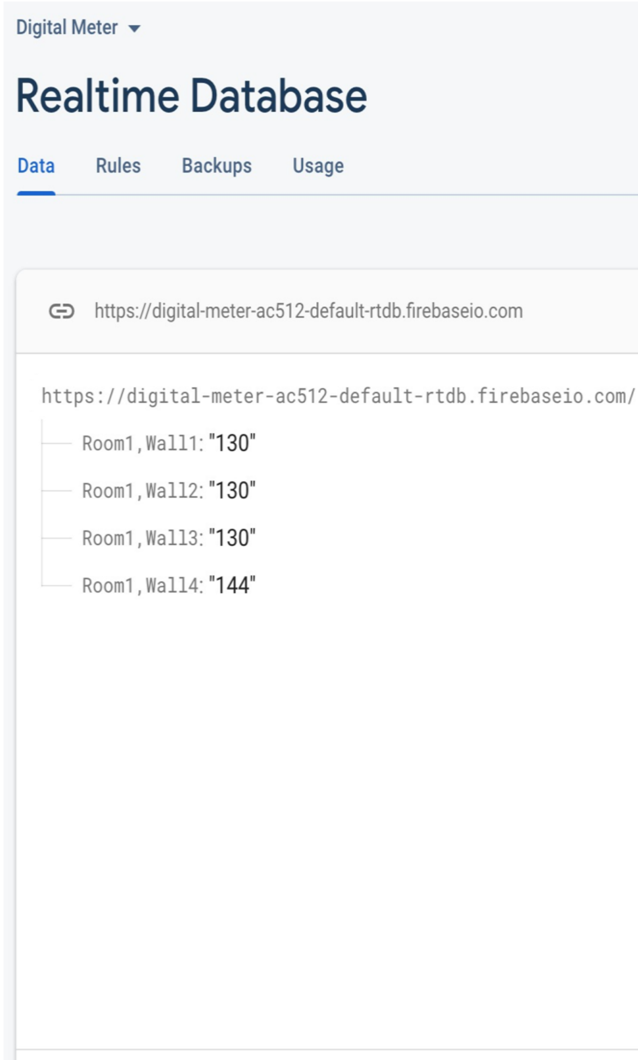
The method to calculate the distance is called “distance()” and it executes the following formula:



Finally, we write the distance value taken from the Ultrasonic Sensor into our app by using the Bluetooth Module.

4. User Interface:





We have decided to use Flutter to make an Android App for several reasons:

-Flexibility
-Easy to implement
-Works well with Firebase
-User Friendly

The most important reason is the compatibility with the Google's platform Firebase, that we have used in the back-end of our app in order to store the values received from our Arduino system.

The reason of that is because both Flutter and Firebase are Google's development tools.

The App is quite straightforward:

A measure value comes from Arduino and we can see the measure in terms of centimeters (cm).

After that we have the chance of saving that measurement made into Firebase by entering a name of the measurement made.

When we log in into Firebase , we can go to the app created and we can check the measurements made from our app.

Choosing a name is important because Firebase system save the values in a "Hash map" style where the keys of our maps are the names of the measurements and the values of each key is the correspondent measurement value.

5. Conclusion:

In this project, a new, digitalized approach is proposed for measuring distances. The proposed approach brings utility and precision with it. The main advantages are to be able to measure distances that are hard to measure in an accurate way by using conventional methods more accurately and being able to record these measurements.

References:

1. https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics

2. https://docs.flutter.dev/get-started/install

3. https://www.arduino.cc/

4. https://store.arduino.cc/products/arduino-uno-rev3

5. https://store.arduino.cc/products/feetech-rc-micro-servo-1-8kg-cm-metal-gears-analog-servo-motor

6. https://store.arduino.cc/products/grove-thumb-joystick

7. https://www.robotistan.com/kablolu-hc06-bluetooth-serial-modul-karti-hc06-bluetooth-to-serial-port-m?#ins_sr=eyJwcm9kdWN0SWQiOiIxMjAxOSJ9

8. https://www.robotistan.com/lazer-sensor-650nm-5v-5mw

9. https://eu.mouser.com/ProductDetail/OSEPP-Electronics/HC-SR04?qs=wNBL%252BABd93PqZEhuhHkuOw%3D%3D&gclid=Cj0KCQjwnNyUBhCZARIsAI9AYlFwCPcPzChF7hddicvt8AeuplnXCMumliRbHBtLzWMt_otAhGMZC3IaAj8sEALw_wcB

10. https://www.robotistan.com/breadboard-2