ACTIVITY 10: Sensors - Touch sensor & Sonar Sensor

Jason Harvey Lorenzo, April 23, 2024
ENGG 122.02, Department of Electronics, Computer, and Communications Engineering
School of Science and Engineering, Ateneo de Manila University
Quezon City, 1108, Philippines
jason.lorenzo@obf.ateneo.edu

***Abstract*—The project focuses on creating a distance measuring circuit incorporating touch and ultrasonic sensors, complemented by a DHT11 sensor for temperature measurement, enhancing distance accuracy. The touch sensor detects changes in capacitance upon skin contact Conversely, the ultrasonic sensor emits waves and detects their echo to gauge distance. Utilizing the echo duration and the speed of sound, the sensor accurately calculates the distance to the object. The provided code orchestrates these sensor functionalities. Upon touch detection, it triggers the ultrasonic sensor to emit a pulse, measures the echo duration, and computes the distance, concurrently reading the air temperature. The results are relayed to the serial monitor for real-time observation. The formula derivation rationalizes the conversion of the calculated distance from meters to centimeters. The Arduino setup diagram illustrates the components' connections, facilitating sensor integration and data collection.**

***Index Terms*—Arduino, sensors, touch, sonar, DHT**

I. Introduction

This project involves the construction a distance measuring circuit that integrates touch and ultrasonic sensors. Additionally, a DHT11 sensor is incorporated to measure air temperature, contributing to more precise distance calculations. The touch sensor operates by detecting changes in capacitance upon contact with the skin, producing a LOW signal when untouched and a HIGH signal when touched. On the other hand, the ultrasonic sensor emits ultrasonic waves and measures their echo to determine distance. By calculating the time taken for the echo to return and considering the speed of sound, the sensor's distance from the object is determined. The TRIG pin triggers ultrasonic wave emission, while the ECHO pin receives and processes the echo signal. This combination of sensors enables efficient and reliable distance measurement in the circuit.

II. Program Listing

A. Source Code

```
#include <DHT.h>

const int TOUCH_PIN = 2;
const int ECHO_PIN = 11;
const int TRIG_PIN = 12;
const int TEMP_PIN = 3;

DHT dht(TEMP_PIN, DHT11);

void setup() {
  Serial.begin(9600);
  dht.begin();
  pinMode(TOUCH_PIN, INPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(TRIG_PIN, OUTPUT);
}

void loop() {
  int TOUCH_STATE =
    digitalRead(TOUCH_PIN);

  if (TOUCH_STATE){
    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);

    long duration =
      pulseIn(ECHO_PIN, HIGH);

    float Tc = dht.readTemperature();
    long v_sound = 0.6*Tc + 331.5;;
    float cm = ((0.6*Tc + 331.5)
      *duration)/20000;

    Serial.print("temperature:■");
    Serial.print(Tc);
    Serial.print("■C■");
    Serial.print("speed■of■sound:■");
    Serial.print(v_sound);
    Serial.print("m/s■");
    Serial.print("distance:■");
    Serial.print(cm);
    Serial.println("cm");
    while(TOUCH_STATE){
      TOUCH_STATE =
        digitalRead(TOUCH_PIN);
    }
  }
}
```

B. How It Works

This code implements the sensors to detect touch and measure temperature and distance. In the setup() function, the serial communication is initialized for outputting to the serial monitor, and the DHT sensor is initiated to read temperature data. Additionally, the pins for the sensors are set as input and output accordingly.

Within the loop() function, the code continuously checks the state of the touch sensor. If the sensor is touched, the ultrasonic sensor is triggered to emit a pulse. The duration between the pulse and its echo is then measured using pulseIn() function. Meanwhile, the temperature is read using the DHT sensor. Based on the temperature, the speed of osund is calculated using a formula. Then, using the duration of the echo and the calculated speed of sound, the distance of the object is computed.

The temperature reading and calculated values are sent to the serial monitor for observation. Finally, a while loop waits until the touch sensor is no longer touched before proceeding to the next iteration of the loop, ensuring that distance measurement is only taken the instance the sensor is touched.

III. Formula Derivation

The linear equation of the velocity of sound (V_s) in air medium is given as:

$$V_s = 0.6(T_c) + 331.5 \tag{1}$$

where (T_c) is the air temperature in °C

From this, the distance can be determined utilizing the fundamental relationship between distance (D_{total}) and velocity, as depicted in eq. 2.

$$D_{total} = V_s t \tag{2}$$

Here, the variable t represents the total duration of sound propagation. For the intended application of this activity, the calculated distance corresponds to the combined round-trip distance covered by the ultrasonic pulse, encompassing both the journey to the object and its return. To accommodate this consideration, the value of D in Equation 2 is halved, reflecting the one-way distance between the sensor and the object.

$$D = \frac{1}{2} V_s t \tag{3}$$

This equation provides the distance measurement in meters, as V_s is expressed in meters per second (m/s) and t is in seconds. However, given that the HC-SR04 ultrasonic sensor typically operates within a range of 2 to 400 centimeters, it's more practical to utilize centimeters as the unit of measurement. Additionally, the pulse duration measured by the sensor falls within the microsecond range. To address this, we can employ dimensional analysis, as illustrated in Equation 4.

$$\begin{aligned} D_{cm} &= \frac{1}{2} V_s t \frac{m \cdot \mu s}{s} \times \frac{100cm}{1m} \times \frac{1s}{1000000\mu s} \\ D_{cm} &= \frac{V_s t}{20000} \end{aligned} \tag{4}$$

Substituting eq. 1 to eq. 4 yields the distance formula used in the code above.

IV. Arduino Setup

The Arduino setup, depicted in Fig. 1, comprises an Arduino UNO connected to a laptop via a USB cord for power supply and program uploading, DHT11, along with touch and ultrasonic sensors. The pins of the sensors are connected to the Arduino as specified in the code. On the background, sample outputs from the serial monitor can also be seen.

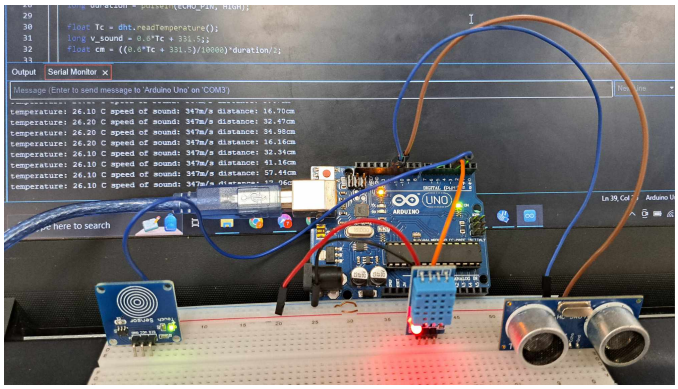

Fig. 1. Arduino setup with DHT11, touch and ultrasonic sensors