ECGR4161 – Introduction to Robotics
Lab Assignment #3
Shengkai Xu
02-26-2021

Objective:

In this lab everything is straightforward. The goal is to the test the usability and component setup of the ultrasonic sensor. Student are instructed to using onboard micro controller MSP432P401R from the TI robotic kit wiring with HC-SR04 ultrasonic sensor. The result need to show five reading from the out put display of Energia IDE and with one median reading after above reading. To test the accuracy of the median reading, one hand blocking action was done to the ultrasonic sensor during the five normal distance reading. It emulated the inconsistent reading in between other four. The median result should be a reading that is usable for real world application inregards to that one error reading caused by the hand block.

URL of the video:
https://youtu.be/imBYFDec1sI

Commentary:

There was many incorrect reading value when I connected the ultrasonic sensor directly to the micro controller board. Because of the voltage requirement and supply different between the board and ultrasonic sensor, the logic level converter was need to convert the logic value. Using logic level converter it eliminated the voltage fluctuation causing the error reading.

Code Snippet:

```
#undef min
#undef max
#include <algorithm>

const int trigPin = 32;  //This is Port Pin 3.5 on the MSP432 Launchpad
const int echoPin = 33; //This is Port Pin 5.1 on the MSP432 Launchpad

constexpr size_t SAMPLE_IDX = 5;

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  Serial.begin(9600);
  Serial.println("HC-SR04 Testing...");
}

long get_sonar_val() {
  long samples[SAMPLE_IDX];
  long centimeters;

  for (int i = 0; i < SAMPLE_IDX; ++i) {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(5);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);

    samples[i] = pulseIn(echoPin, HIGH);
    centimeters = samples[i] / 58;

    Serial.print("Distance = ");
    Serial.print("      ");
    Serial.print(centimeters);
    Serial.println(" centimeters");

    delay(1000);
  }

  std::sort(samples, samples + SAMPLE_IDX);

  if (SAMPLE_IDX % 2 == 1) {
    return samples[(SAMPLE_IDX / 2) + 1];
  } else {
    return ( samples[(SAMPLE_IDX / 2) + 1] + samples[(SAMPLE_IDX / 2) + 1] ) / 2;
  }
}

void loop() {
  long pulseLen;
  long centimeters;

  pulseLen = get_sonar_val();
  inches = pulseLen / 148;
  centimeters = pulseLen / 58;

  Serial.print("Median distance = ");
  Serial.print("      ");
  Serial.print(centimeters);
  Serial.println(" centimeters");
  delay(5000);
}
```