Sharp GP2Y0A21YK0F
Distance sensor on Beaglebone green

Cmpt433
Spring 2022

Kwan Baden Leung - kyl42@sfu.ca
Xizhuo Liu - xizhuol@sfu.ca
Supriya Dua - sda92@sfu.ca
Gaurav Modi - gmodi@sfu.ca

Table of contents:

1.Description:

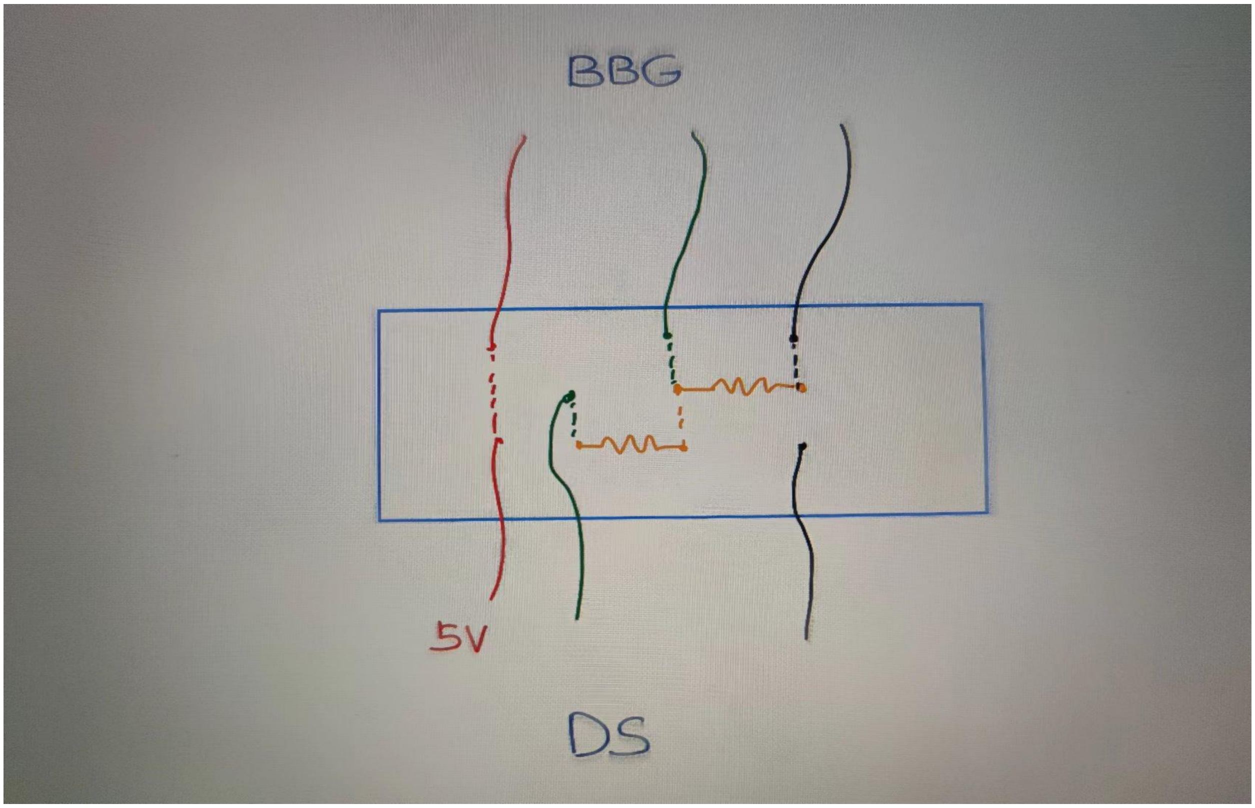
GP2Y0A21YK0F is a distance measuring sensor unit, composed of an integrated combination of PSD (position sensitive detector) , IRED (infrared emitting diode) and signal processing circuit. The distance sensor mainly needs 5V power supply, and its detection range is between 100mm and 800mm. The following guide will lead you how to connect the distance sensor to beaglebone and ensure that the circuit is connected correctly.

2. Wiring:

Required devices:

- three male/female connectors
- two 10kΩ resistors, defined as R1 and R2
- one sharp GP2Y0A21YK0F IR distance sensor
- beaglebone green

The sensor uses Analog-to-Digital converter(A2D) to read the voltage, and the wiring need to meet the following requirements(otherwise it may fire your board or potentiometer, so make sure your connection is correct):



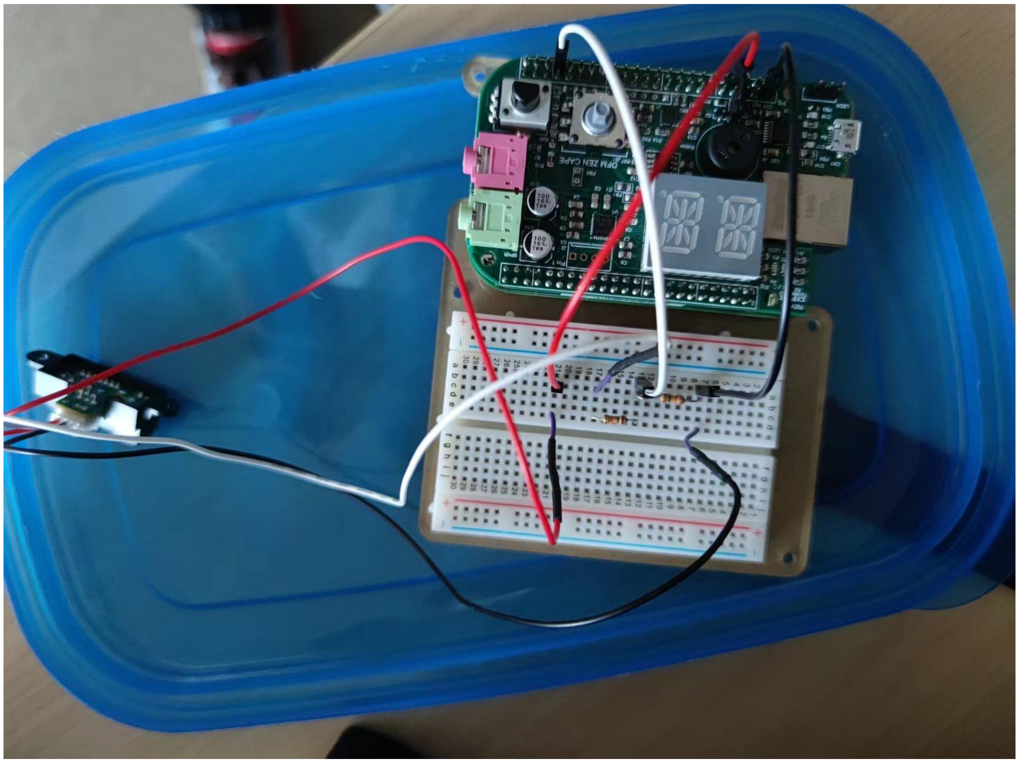
- The ***black wire***(ground) from distance sensor connects to one side of the 10kΩ R1 resistor, and one black male-female wire connects to **P9 PIN 2**(ground)

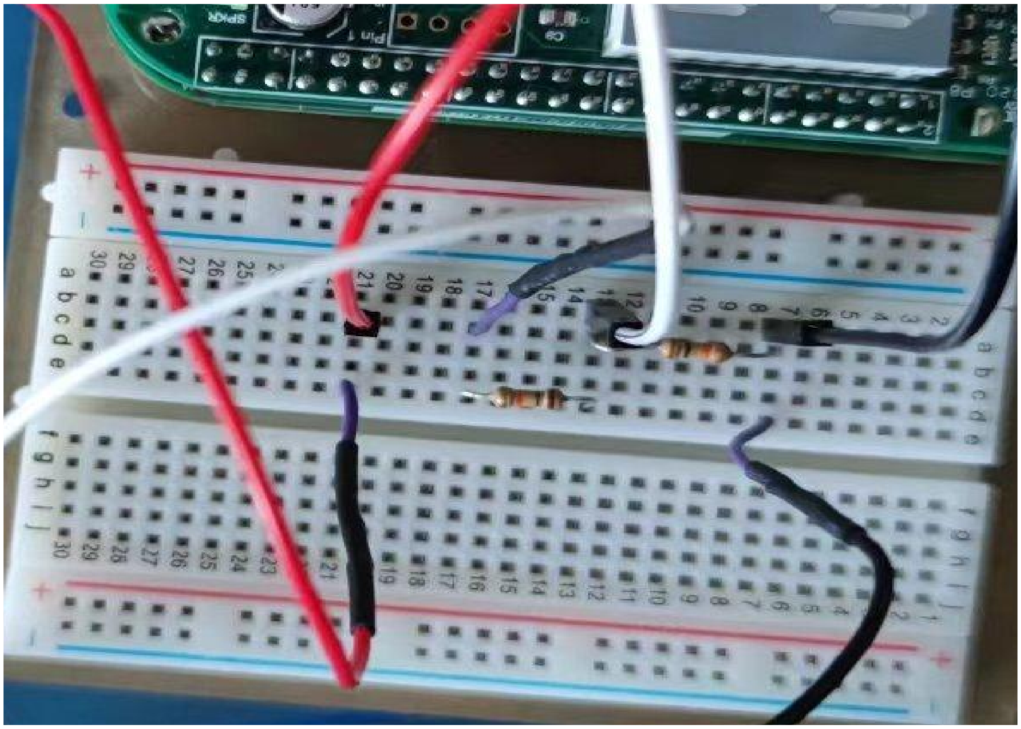
- The ***red wire***(power) from distance sensor connects directly with the red male-female wire, and one red male-female wire connects to **P9 PIN 7**(SYS 5V power)

- The ***green wire***(data) from distance sensor connects to one side of the 10kΩ R2 resistor, and one green male-female wire connects to **P9 PIN 40**(AIN1, or any analog pin you want, except AIN0 used for potentiometer)

- The other side of the R2 resistor connects with the R1 resistor to the same slot (same column on breadboard) with the wire that connects to P9 PIN 40. Their purple strip should be facing towards each other.

The diagram of the physical wiring part is shown below:





3. Checking sensor voltage via terminal:

Through the resistance of two resistors, the voltage of the 5V supply can be limited to the range of 0 to 1.8V, which will not fire our BBG!

To view the status of the distance sensor, we can view it under folder `/sys/bus/iio/devices/iio:device0` , typed with following command:

```
(bbg)$ cd /sys/bus/iio/devices/iio\:device0

(bbg)$ cat in_voltage1_raw
```

because we connect our data wire to the AIN1 pin on BBG

- The value will between 0 to 4095

Troubleshooting:

1) If you can only read 0 and 1 from your A2D pin, there is a high chance that your wire connection is wrong, and it may damage some of your device. Check if your potentiometer is still alive by following command:

   ```
   (bbg)$ cat
   /sys/bus/iio/devices/iio\:device0/in_voltage0_raw
   ```

   see whether you can get voltage readings at around 3000 to 4000.

2) Check if your wires are loose. Gently nudging downwards while holding the wires' end (not the rectangular part) may help strengthen the connection. An extension header is recommended if you want a more sturdy connection

3) If you are changing the position of the potentiometer but the value being read from in_voltage0_raw does not change, ensure the hardware is correctly connected. For example, if you are using the Zen cape, ensure it is fully seated on the BBG (virtually none of the P8/P9 pins should be visible between the BBG and Zen).

(The followings(4,5) are for troubleshooting general A2Ds from Brian Fraser's guides)

4) If the file `/sys/bus/iio/devices/iio:device0/in_voltage0_raw` does not exist, then double check the following in `/boot/uEnv.txt`:
   - Enabled the UBoot overlays:

   ```
   enable_uboot_overlays=1
   ```

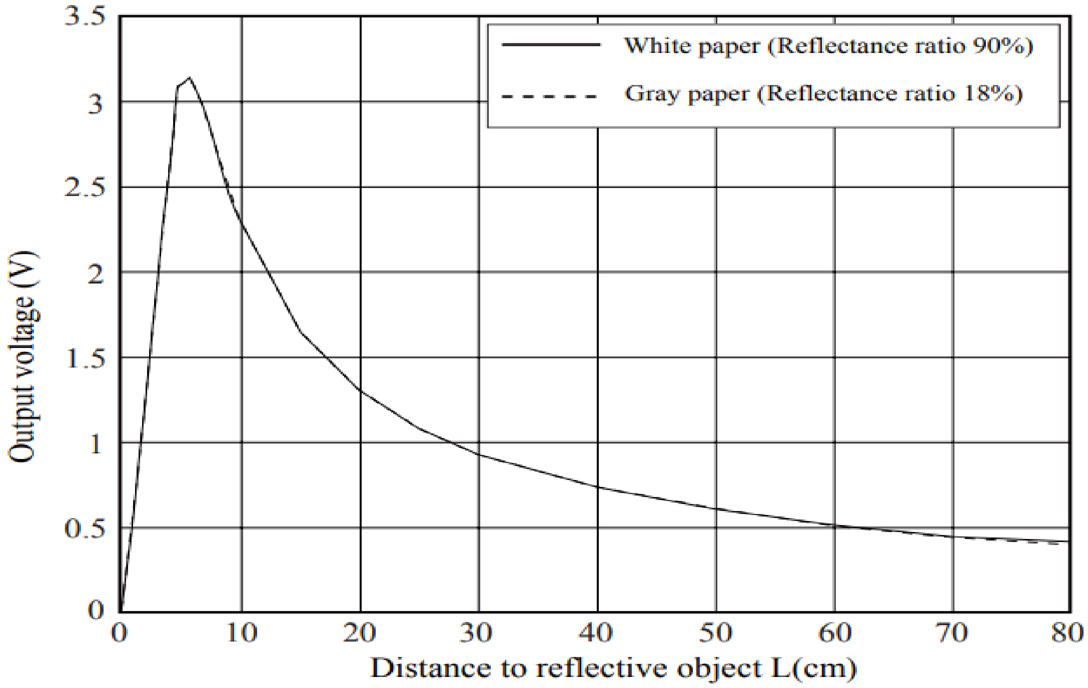
5) Commented out (with a # in front) the A2D disable command:

   ```
   #disable_uboot_overlay_adc=1
   ```

4. Algorithm for distance sensor:

1. we can use the formula to obtain the value of voltage from A2D with formula:
   `voltage=(A2D_value/4095)*1.8`

2. Once we have the voltage, we need a way to calculate distance. We refer to the datasheet of the distance sensor, the relationship between voltage and distance is shown below:



Take voltage as input value x, and distance as output value y, we can use linear approximation function to calculate the distance, and the formula will be:

```
   distance = 29*(voltage^{-1.7})
```

(reference from marker guides website)

3. The C code for calculating the distance will be similar to following idea:

```
#define A2D_FILE_VOLTAGE1
"/sys/bus/iio/devices/iio:device0/in_voltage1_raw"
#define A2D_VOLTAGE_REF_V 1.8

unsigned int SHARPDistanceSensor::getVoltageValues(){
      this->sensorFD=fopen(A2D_FILE_VOLTAGE1,"r");
      if(!this->sensorFD){
      printf("ERROR: UNABLE TO OPEN VOLTAGE FILE\n");
      exit(-1);
      }

      //get reading
      int a2dReading=0;
      int itemsRead=fscanf(this->sensorFD,"%d",&a2dReading);
      if(itemsRead<0){
```

```
      printf("ERROR: UNABLE TO READ VALUES FROM VOLTAGE INPUT FILE\n");
      exit(-1);
      }

      fclose(this->sensorFD);
      return a2dReading;
}

double SHARPDistanceSensor::getSensorValues(){
    //use voltage graph to train the data
    //in order to calculate the distance
    double ret = 0;
    //double average_correspondVoltage=0;
    double correspondVoltage=0;


    for(int i = 0; i < 10; i++){
        int a2dValue = this->getVoltageValues();
        Double voltage
=((double)a2dValue/A2D_MAX_READING)*A2D_VOLTAGE_REF_V;

        correspondVoltage += voltage*5/1.8*0.9;
    }
    correspondVoltage /= 10;


    //linear approximation: Distance=29.988*pow(voltage, -1.173);
    ret=29.988*pow(correspondVoltage, -1.0);
    if((ret <= 10)||(ret >= 80)){
        ret = 0;
    }

    return ret;
    //return correspondVoltage;


}
```

4. **Note:** although the measurement range stated in the datasheet is between 10 to 80 cm, the distance sensor still has voltage in the range of 0 to 10 cm and >80 cm.

Reference:

1. “GP2Y0A21YK0F - Pololu.” [Online]. Available: https://www.pololu.com/file/0J85/gp2y0a21yk0f.pdf. [Accessed: 13-Apr-2022] (for distance sensor datasheet)

2. “Using the IR sensor GP2Y0A21YK...,” *WoodUino.ca*, 14-Feb-2015. [Online]. Available: http://woodsgood.ca/projects/2015/02/02/using-the-ir-sensor-gp2y0a21yk/. [Accessed: 13-Apr-2022].

3. A. B. de Bakker, “IR distance sensor Arduino Tutorial (sharp GP2Y0A21YK0F),” *Makerguides.com*, 02-Mar-2022. [Online]. Available: https://www.makerguides.com/sharp-gp2y0a21yk0f-ir-distance-sensor-arduino-tutorial/. [Accessed: 13-Apr-2022].

4. B. Fraser, “Guides,” *A2D Guide - Guides | CMPT433 - Spring 2022*. [Online]. Available: https://opencoursehub.cs.sfu.ca/bfraser/grav-cms/cmpt433/guides/files/A2DGuide. [Accessed: 13-Apr-2022].