Measurement of water's weight and height with Arduino Mega

Overview

This is a system that has the main purpose to use for measuring and monitoring 3 things every 10 minutes including:

1. Distance
2. Weight of water
3. Height of water level

In addition, the data is sent to mobile via SMS and also written to MicroSD card.

This system is designed to work for Arduino mega with 5 external IoT devices which are ultrasonic sensor called HC-SR04, weight scale called (Load Cell), HX711 amplifier broad, MicroSD card adapter, and GPRS+GPS module (A7 Ai-Thinker).

Library

This system is required 3 libraries consisted of 2 Arduino libraries and 1 contributed library including:

1. SD – this library allows reading from and writing to SD cards.
2. SPI (Serial Peripheral Interface) – this library allows communicating with SPI devices.
3. HX711 – this library allows HX711 24-bit analog-to-digital converter for load cell to work. (src: https://github.com/bogde/HX711)

Materials

1. Arduino Mega
2. Ultrasonic sensor (HC-SR04)
3. Weight scale (Load Cell)
4. HX711 amplifier board
5. MicroSD card adapter
6. GPRS+GPS module (A7 Ai-Thinker)
7. MicroSD card
8. Sim card
9. Arduino IDE

Installation

How to install Arduino IDE?

For Linux: https://www.arduino.cc/en/Guide/Linux (see also the Arduino playground page https://playground.arduino.cc/Learning/Linux)

For macOS X: https://www.arduino.cc/en/Guide/MacOSX

For Windows: https://www.arduino.cc/en/Guide/Windows

How to install IoT devices?

- Arduino Mega and HX711 amplifier

```
Arduino Mega pin 6 -> HX711 CLK
Arduino Mega pin 5 -> HX711 DOUT
Arduino Mega pin 5V -> HX711 VCC
Arduino Mega pin GND -> HX711 GND
```

Figure 1: Connection between Arduino Mega and HX711

- Arduino Mega and A7 Ai-Thinker

```
Arduino Mega TX3 -> A7 Ai-Thinker RX
Arduino Mega RX3 -> A7 Ai-Thinker TX
A7 Ai-Thinker 5V -> A7 Ai-Thinker PWR
```

Figure 2: Connection between Arduino Mega and A7 Ai-Thinker

- HX711 amplifier and Load Cell

```
HX711 E+ -> Load Cell RED
HX711 E- -> Load Cell BLACK
HX711 A- -> Load Cell WHITE
HX711 A+ -> Load Cell GREEN
```

Figure 3: Connection between Load Cell and HX711

- Arduino Mega and MicroSD card adapter

```
Arduino Mega pin 53 -> MicroSD card adapter SS
Arduino Mega pin 51 -> MicroSD card adapter MOSI
Arduino Mega pin 50 -> MicroSD card adapter MISO
Arduino Mega pin 52 -> MicroSD card adapter SCK
```

Figure 4: Connection between Arduino Mega and MicroSD card adapter

How to calibrate the scale

1. Call set_scale() with no parameter.
2. Call tare() with no parameter.
3. Place a known weight on the scale and call get_units(10).
4. Divide the result in step 3 to your known weight. You should get about the parameter you need to pass to set_scale.
5. Adjust the parameter in step 4 until you get an accurate reading. [1]

Description of code

This program is understandably separated into 4 parts including:

1. Libraries' import and variables' declaration part – importing 3 libraries that is stated above, defining constant variables and normal variables, declaring objects, and setting some pins. In addition, it is set calibration factor to -2150 that works for 88 grams tube

2. Setup part (setup()) – this is the main part that is used to initialize and set the initial value of variables. Moreover, it is used to set pin mode, calibrate the scale, initialize the SD library, and set the maximum baud rate of Arduino Mega as 115200.

3. Execution part (loop()) – this is another main part that is used to actively control the Arduino Mega board for execution by calling the external functions.

4. Addition function part

   This part consists of 4 functions including:

   - calculateDistance() is used to calculate the actual distance by applying the raw data that stems from HC-SR04 with this formula; distance = time x speed of sound.

```
Speed of sound = 340 m/s = 34000 cm/s = 34 cm/ms = 0.034 cm/µs
time = distance / speed
distance = time x speed
distance apart (cm) = time x speed / 2 = time x 0.034 / 2
therefore, d = duration*0.034/2.0
```

Figure 5: Distance Formula's Proof

   - calculateWeightHeight() is used to calculate the weight and height of water by applying the raw data that stems from Load Cell and HX711 amplifier board with 2 solutions.
     For weight, using scale.get_units();
     For height, using
     - bF = weightAir – w
     - v = bF / (p*g)
     - height = v / (pi * r * r)

     Note: bF – buoyant force
     w – weight of water that get from scale.get_units()
     weightAir – 88.40
     v – volume of cylinder
     p – density of water
     g – gravity
     pi – value of Pi (3.14159)
     r – radius of hanging tube

```
According to the equation
  Fb = (weight in the air) - (weight in the water)
We keep the weight in the air before running the code
After we change 'weightAir' variable above, we can run the code and calculate Fb - Buoyant force

According to the equation
  Fb = pVg
We know the Fb, density of the water, and g.
Therefore, we can calculate Volume of the sunk part of the hanging tube with this -> V = Fb/(pg)

According to the equation
  v = Pi*r^2*height
We can cal culate the volume of cylinder shape object.
Since we know the volume and the radius of the hanging tube, we can get the height of the sunk tube.
  height = v/(Pi*r^2)
```

Figure 6: Height Formula's Proof [2]

- writeSD() is used to record the data into MicroSD card by using SD library.
- SendSMS() is the function that uses AT command to send the data to mobile phone via SMS.

```
Serial3.println("AT+CMGF=1");    // set input and output format of messages to Text mode.
delay(5000);                    // Delay to let the system finish the work
Serial3.print("AT+CMGS=\"");     // Begin to send SMS to given number
Serial3.print(phone_no);         // Phone number here
Serial3.write(0x22);             // "
Serial3.write(0x0D);             // /r
Serial3.write(0x0A);             // /n                /r/n = new line in Windows
```

Figure 7: Example of AT command

Reference

[1] https://github.com/bogde/HX711

[2] https://www.khanacademy.org/science/physics/fluids/buoyant-force-and-archimedes-principle/a/buoyant-force-and-archimedes-principle-article