Computer Systems and Networks Assignment Two

Andrew Helen

20086819

Remote Tank Volume Monitoring

Aim

Develop a working prototype/ proof of concept of an IoT system to remotely monitor the volume of a liquid in a tank.

System Overview

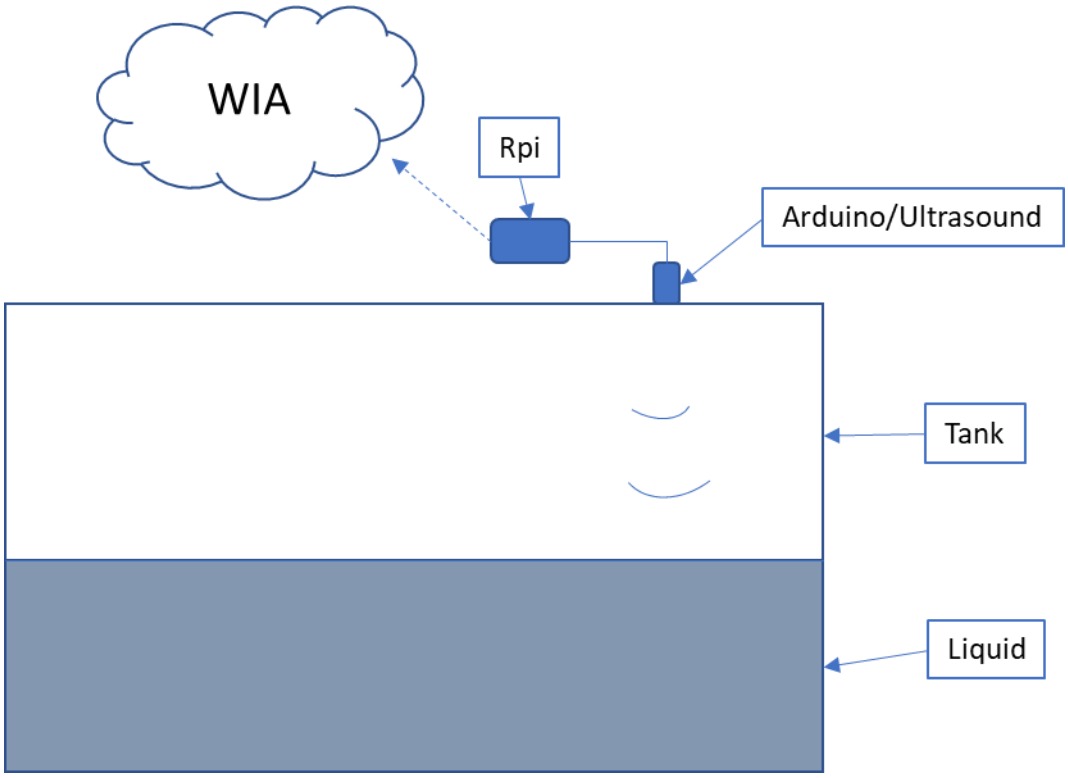

Fig 1. System overview

Fig 1. above gives an overview of where the various components fit in the system. A ultrasound sensor connected to an Arduino UNO measures the height of the liquid in the tank. This is then sent to the Raspberry Pi (Rpi) which processes the data. Since the volume in the tank is related to the height of the liquid in the tank a look up table can be used to calculate the volume stored in the tank. This lookup table is stored in a MySQL database on the Rpi.

Once the volume of liquid is known the Rpi then sends this to the WIA IoT platform where it can be represented graphically and accessed remotely.

IoT Layers

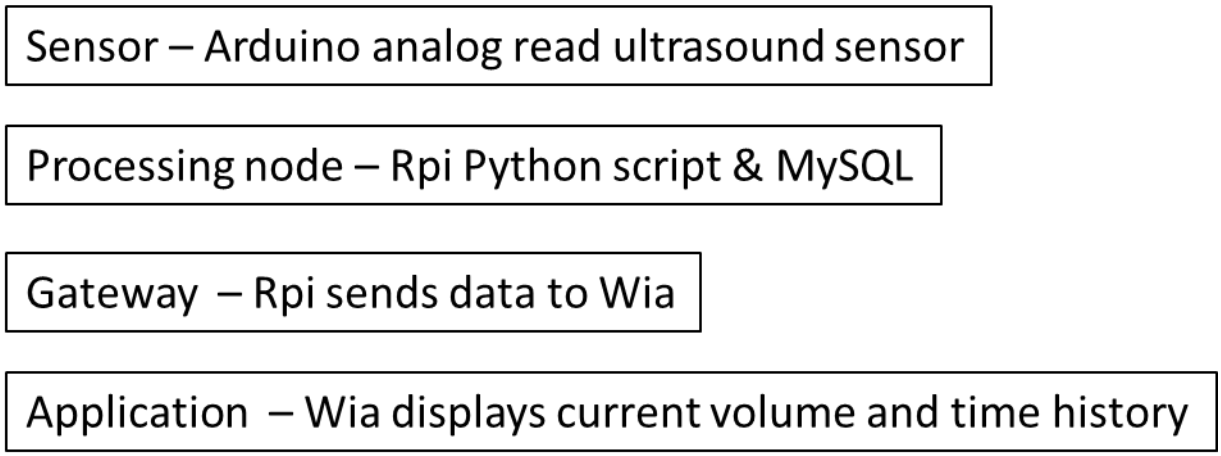

Fig 2. Layers of the IoT System.

Software flowchart

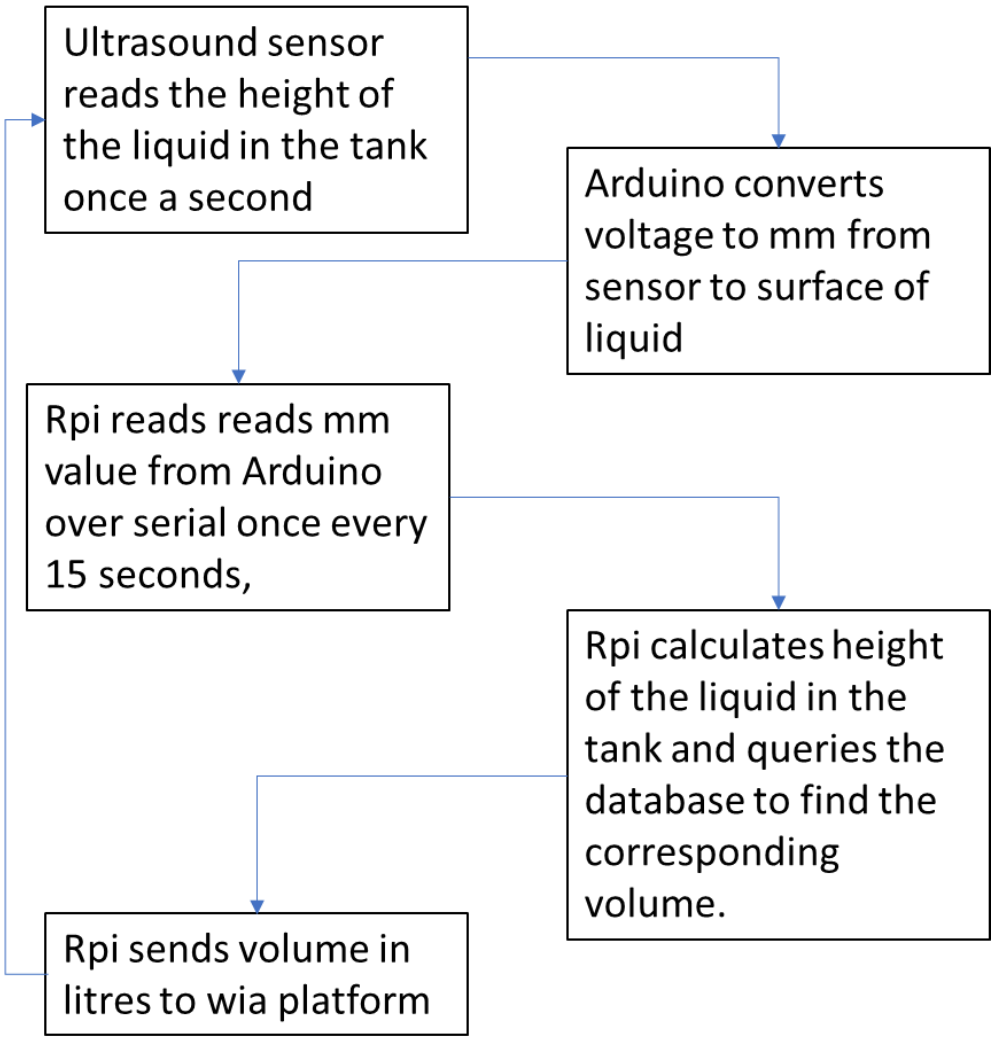

Fig 3. Software flowchart

Arduino connection diagram

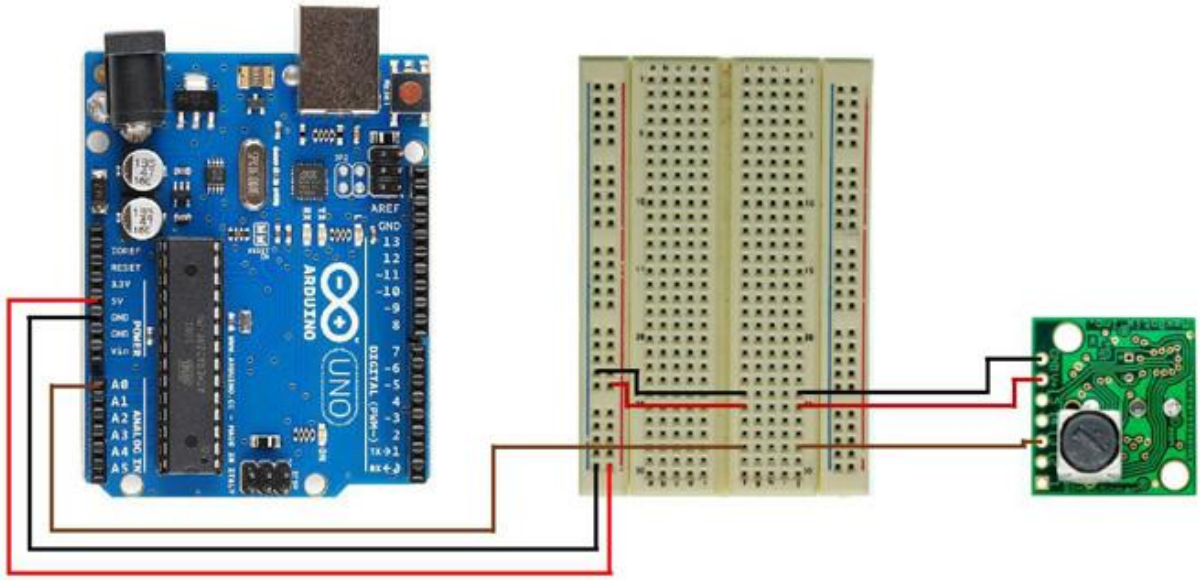

Fig 4. Arduino connection diagram. (maxbotix.com, 2020)

Arduino Sketch

HRLV_MB1013_analog_read | Arduino 1.8.10

File Edit Sketch Tools Help

HRLV_MB1013_analog_read

```
const int anPin = 0; // use analog pin 0
long anVolt, mm;


void setup() {
  Serial.begin(9600);

}

void read_sensor() {
   anVolt = analogRead(anPin); // read raw voltage
   mm = anVolt * 5; // convert adc value to mm
}

void loop() {
  read_sensor();
  {Serial.println(mm); // use serial print instad of write to send mm as a string instead of byte
  delay(1000); // one second wait between readings
  }
 }
```

Fig 5. Arduino Sketch

Python Script

```
 GNU nano 2.7.4                                          File: tank_volume.py

#!/user/bin/env python

# import libraries
import serial
import time
import mysql.connector
from wia import Wia

# connect to local database containing tank voilume  map
db= mysql.connector.connect(user='admin',password='admin',host='localhost',database='tank_level')

# set db cursor
c = db.cursor(buffered=True)


wia = Wia()

wia.access_token = "                              "

# set serial port to communicate with arduino
port = "/dev/ttyACM0"

# start serial
sl = serial.Serial(port, 9600)
sl.flushInput()
sensorHeight = 3000 #distance in mm from bottom of empty tank to sensor
while True:
        if sl.inWaiting()>0:
                inputValue=sl.readline().strip().decode("utf-8") # read string from serial
                measuredHeight = (int(inputValue)) # convert string to int
                height =sensorHeight - measuredHeight # subtract to get the height of the liquid in the tank
                print height
                c.execute("select volume from levelMap where height=%s",(height,)) # query db for tank volume
                result = c.fetchone() # return result of query
                volume = ( "{}".format(result[0])) # format query result into a string
                print volume
                wia.Event.publish(name="Level", data=volume) # push volume to wia
                time.sleep(15) # wait  to prevent flooding wia
                sl.flushInput() # clear serial input
```

Fig 6. Python Script

Wia Dashboard

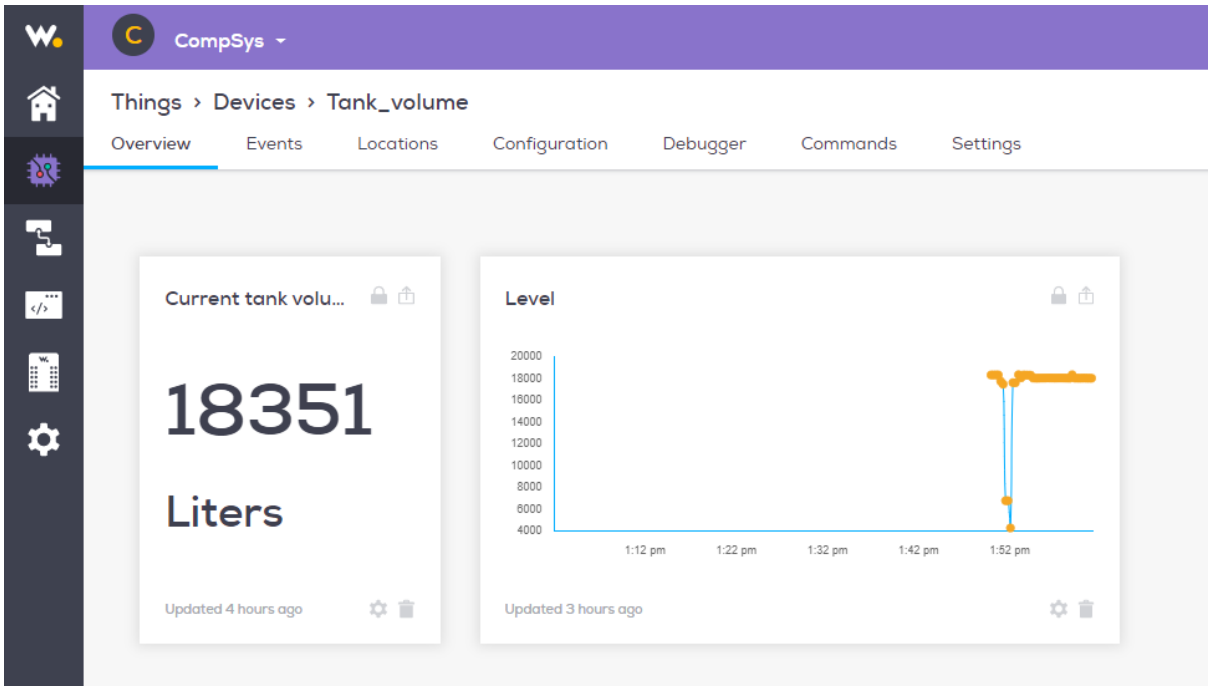

Fig 7. Wia dashboard

References

Maxbotix.com 2006, *How to use an Ultrasonic Sensor with Arduino [with Code Examples], viewed 3/1/20,* < https://www.maxbotix.com/Arduino-Ultrasonic-Sensors-085/>