FLOOD ALERTING SYSTEM USING RASPBERRY PI & THINGSPEAK -IOT PLATFORM

Dinesh Kumar R, Paneerselvam M, Surendar R, Ganesh V, Mrs. Kasirathi. N*

Final year UG Student, *Assistant Professor-III ,Department of Electrical and Electronics Engineering
Velammal Engineering College,Chennai-600099

ABSTRACT

Flooding is considered one of the most devastating natural disasters in the world. In countries like India with climatic conditions occurrence of heavy rain fall and subsequent discharge of water leads to Flood. Flooding creates major damages to life , their habitats and the economy By installing of flood alerting systems near major waterways vital information can be provide so that lives and property can be protected. Normal Weather monitoring and alerting systems are not quick and accurate enough to predict floods in time to prevent personal or environmental damages. The government has to spend tons of money in flood mitigation plans to help the victims and also to reduce the number in the long run damages that can occur after flooding. Since Most of the flood alerting systems involves high cost they are deployed on select locations based on priority.In this project we make use of a cost effective system using raspberry pi board and sensors, to measure flood flow rate and rise of water level in revvers and water bodies and alert government authorities and people instantly by transmitting information using IOT. In the present work we have used thingspeak-IOT platform . The data can be accessed from android smart phones using thingsView mobile application at any time from anywhere in the world.

1 .INTRODUCTION

The drastic climatic changes due to the effect from various human activities such as pollutions, cutting of innumerable trees and too much of gas emission are the some of the main reason for natural disasters that occur in worldwide . The most common factor that cause major damage to life, property and country's economy is the flood .Flooding is brought on by an increased quantity of water in lake or river when it is overflowing. When a dam fractures and abruptly releasing a massive quantity of water not only houses and property are damaged , sewage overflow and chemical spillage also leads to a variety of diseases afterwards. To manage these kind of situations and alert people understanding of increased water level and speed of water flow are valuable for discovering potential seriousness of the flood. Use of IoT based-system is attracting attention of researchers since in this system data from various devices are collectively sensed by sensors and provide various services without human intervention [1]. Lo et al. [2] have proposed an automatic system to monitor the flood based on the remote cyber surveillance systems . For measureing water level at every second Azid et al [3] have used pressure sensor. Jana Priya et al. [4] and Satria et al. [5] have demonstrated the idea and implemented a flood monitoring and alert system using different sensors.

This project presents the details of how the data - like flood speed and flood level are collected from sensors and made available on cloud by using Raspberry pi and Thingspeak-an IOT platform. The data from the IOT cloud can be accessed by android smart phones at anytime from anywhere in the world using the mobile app thingshow.

2 Flood Alerting :Basic Concepts

The proposed flood alerting system to check the flood speed and level basically consists of a microcontroller, ultrasonic sensor, flow meter, and a web server. Fig.1 shows the block diagram of the system. YF-S201 Hall effect sensor is used to gauge the flow rate of flood. The microcontroller used in the present system is Raspberry Pi 4B development board. The hall effect flow sensor - physically placed in water stream is used to measure the flow rate. An analog pulse generated Hall effect flow sensor is sent to the raspberry pi . pi board is programmed to read the signal. In the same way an ultrasonic distance sensor -which measures the time of travel by echo signal gives the water level of flood in river- The Pi board is also programmed to send the data to a cloud are using an external web server named as Thingspeak so as to alert the general public.

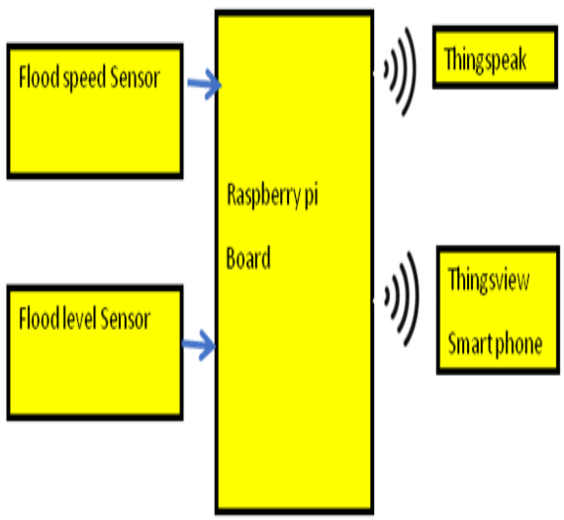

Fig.1 Flood alerting system-.

Advantages of the proposed system:

Improves flood alerting system

Intelligent management of the services .

Due to automation it will reduce the time tocheck the parameters.

Low maintenance.

Real time information on the website.

3. System hardware design

The Raspberry Pi 4B board is the central module of the whole Flood alerting system. Its main parts include: main processing chip unit, memory, power supply HDMI Out i.e. VGA display, Ethernet port, and USB ports. Sensor components are connected to Raspberry pi through various GPIO pin 16 (or GPIO23) which will be discussed in the following section

3.1Raspberry Pi3 Model B

It is a fantastic platform for building build automation systems. The Raspberry Pi3 model B board [6] is used as a "hub" for automation systems, connecting to open-source hardware parts like sensors and actuators. Raspberry Pi3 Model B is capable of doing all jobs that normal desktop computer does.

The following are the main features of Raspberry Pi3ModelB [7]

It is faster and more powerful than its predecessors.
It has built -in wireless and Bluetooth connectivity,
Due to this feature it is the ideal IoT solutions.
It is Built on the latest Broadcom 2837 ARMv8 64bit processor
It has 1.2GHz QUAD Core Broadcom BCM2837 64bit ARMv8 processor,
BCM43438 Wi-Fi on board,
Bluetooth Low Energy (BLE) on board.
The Raspberry Pi3B model is shown in fig.2

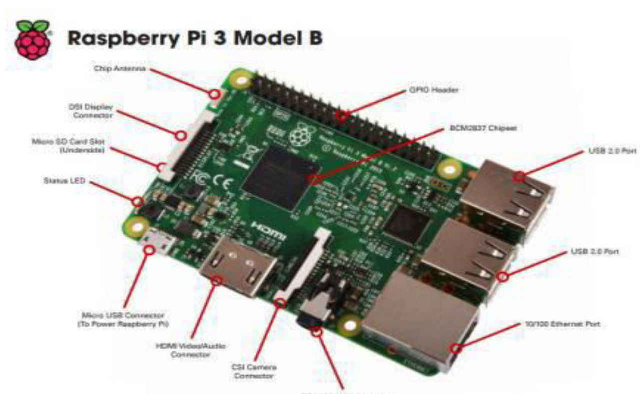

Figure .2 Raspberry Pi 3 Model B module

Raspberry Pi 3Model B run on Linux based OS. It boots and runs from the SD card. It does not have any internal memory other than the ROM. It has an SD card slot which is capable of reading up to 32 GB. All 28 - GPIO pins in raspberry Pi3 Model B are programmed using Python. The I/O devices like sensors and all i/o pins connected to GPIO pins in raspberry pi board based on the requirements. Rapberry pi GPIO pin diagram is shown in fig.3

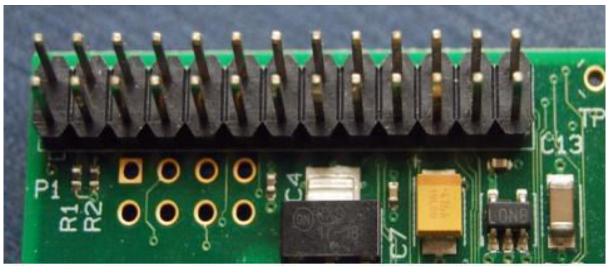

Figure 3 GPIO pin diagram

3.2 Flood Level Sensor

Flood level Sensor LV-MaxSonar-EZ [8,9]used to find the flood level is an ultrasonic rangefinder. Interfacing of this sensor with raspberry pi is discussed below,

LV-MaxSonar EZ is a 7 stick module . In this stick numbers 1,2 & 3 are respectively connected to GND-(-ve) Vcc and GPIO 15 of raspberry pi biard as shown in fig 2. Since the sensing range of this sensor is 15-645 cm it can be used to measure the level of water in rivers flood.

The distance /level is obtained by multiplying the speed of the sound and elapsed time. That is the distance is 34300x elapsed time.

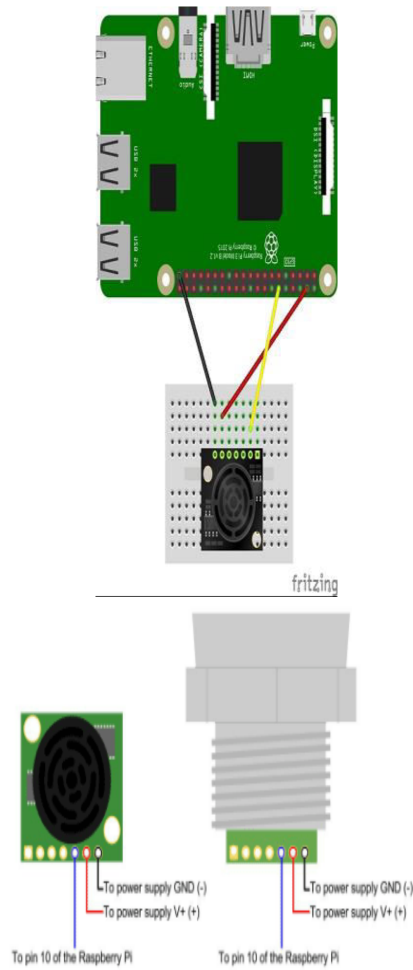

Figure 4: Interfacing Diagram-Ultrasonic Sensor MB1040 LV- MaxSonar with Raspberry-PI.

3.3 Flood Flow rate Sensor YF-S201

The flow rate of flood is measured with the help of Hall Effect sensor type flow meter. Hall Effect is defined as the production of a voltage difference across an electrical conductor, perpendicular to an electric current passing through the conductor and a magnetic field perpendicular to the current.

In Hall Effect based flow rate sensor the output voltage is varied in response to a magnetic field. Hall Effect sensors are used for many applications which includes current detection, speed detection ,proximity switching and positioning.

A flow rate of 1-30 l/m can be measured using this sensor and it can withstand a water pressure of 2.0Mpa. In the present work we have used pulse type Flow sensor YF-S201[10] . It contains a wheel which rotates when water flows through it. The water flow speed can be measured by counting the number of revolutions made by the wheel in the sensor. In the present work the flow rate sensor is interfaced with Raspberry Pi board. The reading obtained is in the form of a pulse. If pulse count gets increased it is identified that flow speed is faster. If it is less count, it is assumed that revolution is slow because the water flow is slow on passing across the flow sensor. Fig.5 shows the flow sensor interfaced with Raspberry pi board.

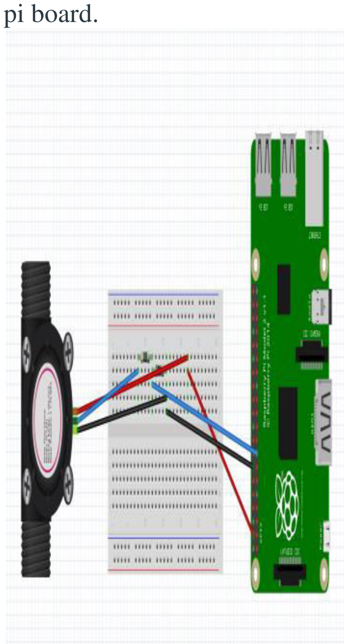

Figure 5 Interfacing of Flow meter sensor with Raspberry pi

4. Design Implementation

The sensor reading were accessed from sensors using Raspberry Pi platform with the following specifications:

• SBU Raspberry Pi 2 900 MHz Quad Code ARM Cortex-A7

• Bootable SanDisk Ultra 8GB micro SD Card

• Sensors- LV-MaxSonar and YF-S201

Raspbian Os installation programming for sensors procedure for connecting Raspberry pi with laptop using putty an dXming server sre discussed in the following section

4.1 OS Installation in pi

Download OS from link raspberrypi.org [11]

Extract zip file to Memory card on computer or laptop

Then insert SD card in to R pi SD card slot.

Connect USB power cable, Connect HDMI to VGA cable and to Monitor, Connect Mouse and Keyboard to USB of R Pi

After powering it will ask to install OS.

We can select multiple OS. Select Rassbian OS. Select Keyboard type ZA. Click on Install. It will take 20 to 30 Mins.

4.2 Steps for Programming

Connect raspberry pi Hardware setup, give supply

Click on R pi symbol (Strawberry/like start in windows)

Open Python 2 (IDLE)

Click on File

Select new file

Untilled file will be open

Write your code in this file

4.3 Steps for connecting Raspberry pi with laptop

Though Raspberry Pi is "Pocket Size PC", for debugging & project purposes its difficult to carry an additional display. Moreover we may not have HDMI display at places where we projects.. So, we need to connect the Raspberry pi to a laptop.

The procedure to connect Pi to laptop is explained briefly in the following section. More details for readers are given in reference 12.

1.After Burning the raspbian OS to sd Connect Rp to my home router using ethernet cable.

2.install Xming and putty in my laptop and start them.

3.Enable X-11 forwarding in putty, and login to raspberry pi's ip address from 2.

4.try to load 'lxsession'

To do this simply:

1. In PuTTY: go to **Connection->SSH->X11**
2. Tick **Enable X11 forwarding**
3. Change **X Display location** to -> localhost:0
4. Run xming, connect to RPi with PuTTY
5. In the shell, type startlxde

Now raspberry can be accessed from laptop using x ming server. The program in Python for both flood flow rate sensor and level sensor which are given in Appendix can be run and viewed from the x mingserver.

5 OPEN SOURCE IOT PLATFORM - THINGSPEAK

There are various Open Source Cloud Platforms available Like Google Cloud Platform, ThingWorx, Thing Speak etc .In the present work we have used ThingSpeak. ThingSpeak is an open source Internet of Things (IoT) application and API to store . It retrieves data from different things using the HTTP protocol over the Internet or via a LAN. ThingSpeak empowers the making of sensor logging applications, area following applications, and an informal community of things with notices". Thing Speak was initially propelled by ioBridge in 2010 as an administration in help of IoT applications. Thing Speak has coordinated help from the numerical registering programming MATLAB from Math Works, permitting ThingSpeak clients to investigate and envision transferred information utilizing Matlab without requiring the buy of a Matlab permit from Math works.

5.1 Procedure for creating a Channel in ThingSpeak

1. Sign In to ThingSpeak™ using your Math Works® Account, or create a new Math Worksaccount.
2. Click Channels>My Channels.
3. On the Channels page, click New Channel.
4. Check the boxes next to Fields 1–3. Enter these channel setting values:
Name: Flood flow rate
Field 1: L/m
Field 2:
Field 3:

5. Click Save Channel at the bottom of the settings.
You now see these tabs:
6. Private View: This tab displays information about your channel that only you can see.
7. Public View: If you choose to make your channel publicly available, use this tab to display selected fields and channel visualizations.
8. Channel Settings: This tab shows all the channel options you set at creation. You can edit, clear, or delete the channel from this tab.
9. Sharing: This tab shows channel sharing options. You can set a channel as private, shared with everyone (public), or shared with specific users.
10. API Keys: This tab displays your channel API keys. Use the keys to read from and write to your channel.
11. Data Import/Export: This tab enables you to import and export channel data.

Next Steps Your channel is available for future use by clicking Channels>My Channels.
For further details authors are advised to refer mathworks website [13]

The program in Python for communication sensor data from Raspberry pi with the ThinSpeak are given in Appendix. Circuit diagram of the flood alerting system drawn using fritzing software is shown in figure 6 Flowchart shown in figure 7 represents the working of Flood monitoring and alerting system Prototype setup of Flood alerting system is shown in fig.8

IJTRET "Voice of Innovative Minds"
International Journal of Trendy Research in Engineering and Technology
Volume 4 Issue 6 Oct' 2020
ISSN NO 2582-0958

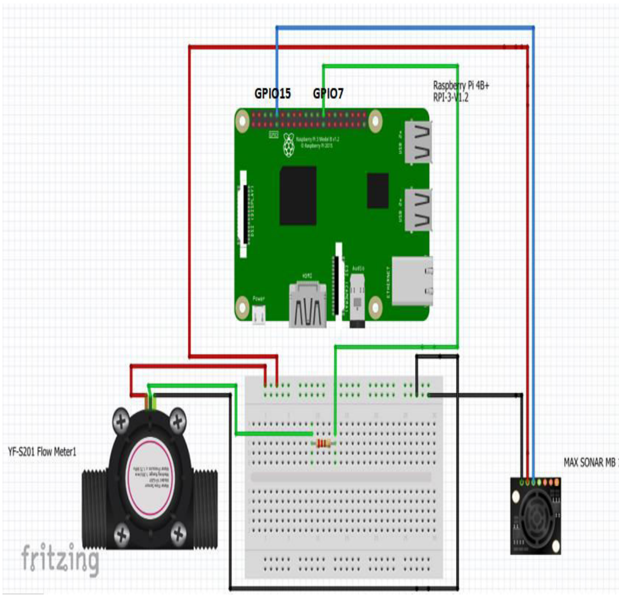

Fig.6 Circuit diagram of the flood alerting system drawn using Fritzing software

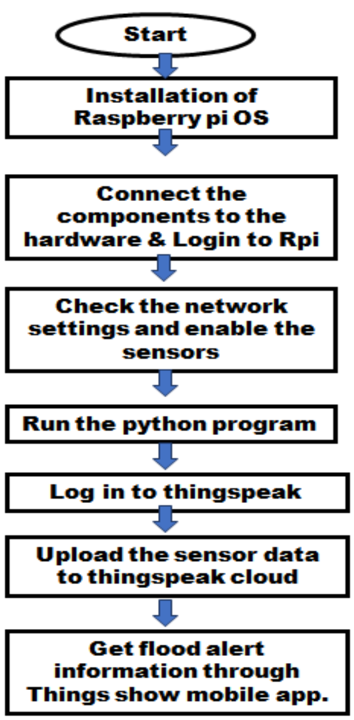

Fig.7 Flowchart -working of flood alerting system.

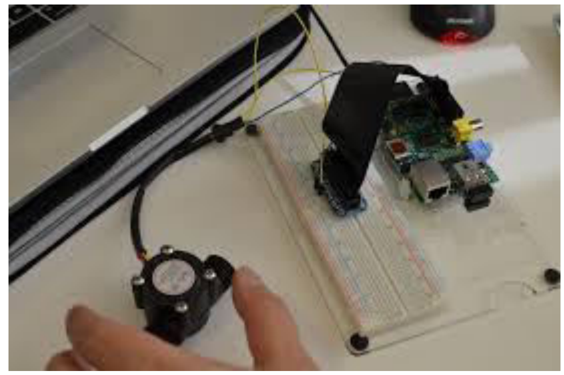

Fig.8 Prototype model of Flood Alerting System

The above figure shows the prototype model to measure flood level and flow rate.

.

6.RESULTS

This online flood alerting system using Raspberry Pi4 Model B is very useful for government authoriries and general public in different aspects.. Our system is designed to measure the flood level and flood speed and the concerned authority can access the information from anywhere and anytime to get the details of the situation accordingly they can take the decision on this immediately.

After the task of interfacing the sensors with raspberry pi and using appropriate python coding the results were obtained .Then the remote correspondence between raspberry pi and Thingspeak web server are successfully achieved.

From the present project work , the following can be obtained

- Water level in the river.
- Flood flow rate
- Transmit the information to the cloud.
- Accessing of data at any time from any where in the world.

- A real time data transmission and accessing system,

The screen shot of Flow rate measurements and timestamps which are printed out to the console for every 5 seconds in shown in Fig. 9

```
Timestamp:2019-05-2022:00:41.267727
 Flow rate:  :25.06000
Timestamp:2019-05-2022:00:46.278729
 Flow rate:  :25.20000
Timestamp:2019-05-2022:00:51.283864
 Flow rate:  :25.208200
```

Figure 9 Screenshot of Flow rate measurements.

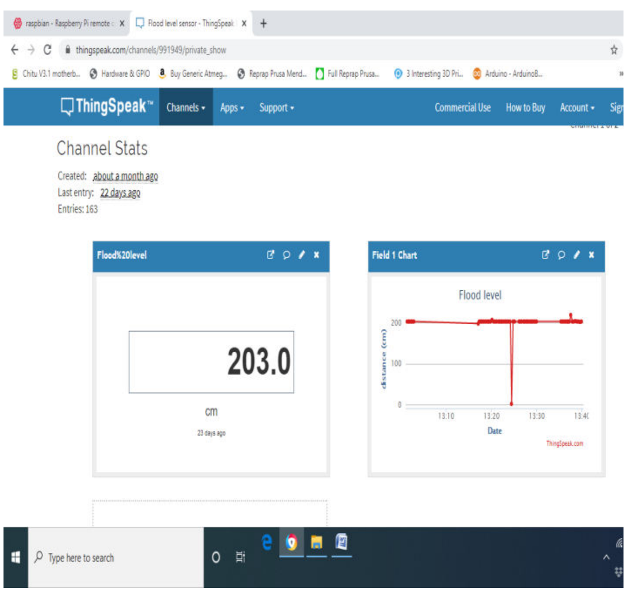

Figure 10 Flood flow rate measured on Thingspeak Windows IOT-Platform) (a) Numerical Data (b) Graphical Data

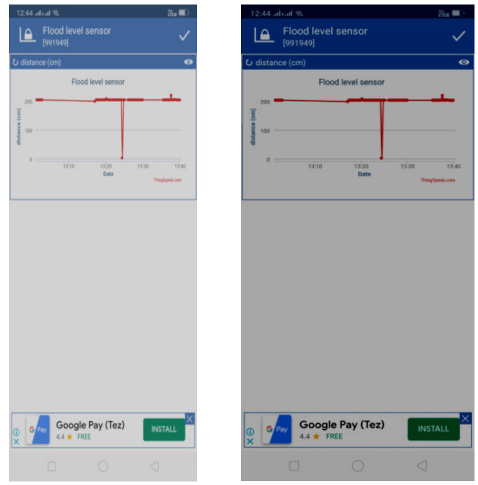

Figure 11: Thingsview Smart phone-IOT App: (a) Water level data (b) Flood flow rate data

7. CONCLUSION

This project deals about the monitoring of flood speed and flood level remotely from anywhere in the world using internet through a personal computer or Smartphone and alert the general public. In this project raw data loaded in the cloud can be visualized in graphical format with in a very short span of time at a remote desk/ mobile app.

We have used Raspberry pi and MALAB's Thingspeak IOT platform along with the sensors for the first time to monitor and alert flood situation. In this system, sensors work with low input power and detects the flood level in the river with a high accuracy. It can detect even an inch raise of water level and give alert. We have tested this system by real time for water flow rate and water level measurement successfully. If the water level increases along with increase of flood flow speed , it will send an alert immediately.

The project can be scaled up to a larger implementation with more channels for visualization by using other IOT platforms like Amazon-AWS , Google -Cloud IoT and Microsoft Azure.

REFERENCES

1. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," IEEE Communications Surveys Tutorials, vol. 17, no. 4, pp. 2347–2376, Fourthquarter 2015.

2.S. W. Lo, J. H. Wu, F. P. Lin, and C. H. Hsu, "Cyber surveillance for flood disasters," Sensors (Switzerland), 2015.

3.S. Azid, B. Sharma, K. Raghuwaiya, A. Chand, S. Prasad, and A. Jacquier, "SMS based flood monitoring and early warning system," ARPN Journal of Engineering and Applied Sciences, 2015.

4.S. J. Priya, S. Akshaya, E. Aruna, J. A. M. Julie, and V. Ranjani, "Flood monitoring and alerting system," International Journal of Computer Engineering & Technology (IJCET), vol. 8, no. 2, p. 15, Mar 2017.

5. D. Satria, S. Yana, R. Munadi, and S. Syahreza, "Prototype of google maps-based flood monitoring system using arduino and gsm module," International Research Journal of Engineering and Technology (IRJET), vol. 4, no. 10, Oct 2017.

6. SimonMonk ,Raspberry Pi Cookbook, published by O'Reilly, 2014

7. Raspberry Pi: Making Amazing Projects Right from Scratch! by Packt Publishing (2016)

8.https://www.maxbotix.com/documents/HRLV-MaxSonar-EZ_Datasheet.pdf

9.http://www.maxbotix.com/Raspberry-Pi-with-Ultrasonic-Sensors-144/ .

10.https://files.amperka.ru/store-media/products/water-flow-sensor/media/ YF-S201.pdf

11. https://www.raspberrypi.org/downloads/

12.http://www.straightrunning.com/XmingNotes/pixming.php

13.https://in.mathworks.com/help/thingspeak/collect-data-in-a-new-channel.html

Annexure I

```
# flood alert project-part1
import time
from datetime import datetime
import RPi.GPIO as GPIO
import sys
from urllib.request import urlopen

class FlowMeter():
    ''' Class -represents the flow meter sensor which handles input pulses
        and calculates current flow rate  measurement in L/min
    '''

    def __init__(self):
        self.flow_rate = 0.0
        self.last_time = datetime.now()

    def pulseCallback(self, p):
        ''' Callback that is executed with each pulse
            received from the sensor
        '''

        # Calculate the time difference since last pulse recieved
        current_time = datetime.now()
        diff = (current_time - self.last_time).total_seconds()

        # Calculate current flow rate
        hertz = 1. / diff
        self.flow_rate = hertz / 7.5

        # Reset time of last pulse
        self.last_time = current_time

    def getFlowRate(self):
        ''' Return the current flow rate measurement.
            If a pulse has not been received in more than one second,
            assume that flow has stopped and set flow rate to 0.0
        '''

        if (datetime.now() - self.last_time).total_seconds() > 1:
            self.flow_rate = 0.0

        return self.flow_rate

def main():
    ''' Main function for repeatedly collecting flow rate measurements
        and sending them to the thingspeak API
    '''
```

```
    # Configure GPIO pins
    INPUT_PIN = 7
    GPIO.setmode(GPIO.BOARD)
    GPIO.setup(INPUT_PIN, GPIO.IN, pull_up_down=GPIO.PUD_UP)

    # Init FlowMeter instance and pulse callback
    flow_meter = FlowMeter()
    GPIO.add_event_detect(INPUT_PIN,
                          GPIO.RISING,
                          callback=flow_meter.pulseCallback,
                          bouncetime=20)

    # Begin infinite loop
    while True:

        # Get current timestamp and flow meter reading
        timestamp = str(datetime.now())
        flow_rate = flow_meter.getFlowRate()
        print('Timestamp: %s' % timestamp)
        print('Flow rate: %f' % flow_rate)
        baseURL= 'https://api.thingspeak.com/update?api_key=%s' % "SL**********4V03"
        f=urlopen(baseURL + "&field1=%d"%(flow_rate))
        print(f.read())
        f.close()
        # Delay
        time.sleep(5)

if __name__ == '__main__':
   main()
```

Annexure II

```
# Flood alert projec- part 2 flood level measurement
import RPi.GPIO as GPIO
import time
import sys
from urllib.request import urlopen

# Use BCM GPIO ref.
# instead of physical pin numbers Gpio pins
GPIO.setmode(GPIO.BCM)

# Define GPIO to use on Pi
GPIO_TRIGECHO = 15

print ("Ultrasonic Measurement")

# Set pins as output and input
GPIO.setup(GPIO_TRIGECHO,GPIO.OUT)  # Initial state as output


# Set trigger to False (Low)
GPIO.output(GPIO_TRIGECHO, False)

def measure():
  # This function measures a distance
  # Pulse the trigger/echo line to initiate a measurement
    GPIO.output(GPIO_TRIGECHO, True)
    time.sleep(0.00001)
    GPIO.output(GPIO_TRIGECHO, False)
  #ensure start time is set in case of very quick return
    start = time.time()

  # set line to input to check for start of echo response
    GPIO.setup(GPIO_TRIGECHO, GPIO.IN)
    while GPIO.input(GPIO_TRIGECHO)==0:
        start = time.time()

  # Wait for end of echo response
    while GPIO.input(GPIO_TRIGECHO)==1:
        stop = time.time()

    GPIO.setup(GPIO_TRIGECHO, GPIO.OUT)
    GPIO.output(GPIO_TRIGECHO, False)

    elapsed = stop-start
    distance = (elapsed * 34300)/2.0
    time.sleep(0.1)
    return distance


try:

    while True:

        distance = measure()
        print ("Distance:",distance,"cm")
        baseURL= 'https://api.thingspeak.com/update?api_key=%s' % "5A**********HF54"
        f=urlopen(baseURL + "&field1=%d"%(distance))
        print(f.read())
        f.close()
        time.sleep(1)

except KeyboardInterrupt:
    print("Stop")
    GPIO.cleanup()
```