PHASE_3

PROJECT: IOT based Flood Monitoring and Alerting System

INTRODUCTION:

The extreme 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. This project presents the details of how the data - like flood level and rain intensity are collected from sensors and made available on cloud and sending alert messages by using Raspberry pi , Thingspeak-an IOT platform and a Global System for Mobile communication (GSM) and short message service (SMS) to relay data from sensors to computers or directly alert the People of that area through their mobile phone. The data from the IOT cloud can be accessed by android smart phones at anytime from anywhere in the world using the mobile app things view.

PROPOSED SYSTEM:

The proposed flood alerting system to check the flood level basically consists of a Raspberry pi which detects the water level and rain intensity using an Ultrasonic Distance Sensor (HC-SR04) and rain sensor respectively. The ultrasonic distance sensor –which is used to measures the time of travel by echo signal gives the water level of flood in river. The Rain Sensor is used to show the Rain status that it’s raining or not, and its rain intensity Value.

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.

By applying an Internet of Thing can help people to monitor the flood water level and rain intensity via thingsview app in smartphone together with the alerting system for incoming flood. A wireless sensor node connected with Raspberry pi is used which consist of ultrasonic sensors and rain sensor to collect data and sent them via cloud to be viewed in Thingsview application. GSM Module is used to send alert SMS to the People When the water level reaches a certain level of hazards, the device will generate an alarm system with three different colours of LEDs indicating three levels of detection for flood level and send Alert notification to the people on incoming flood in that area.

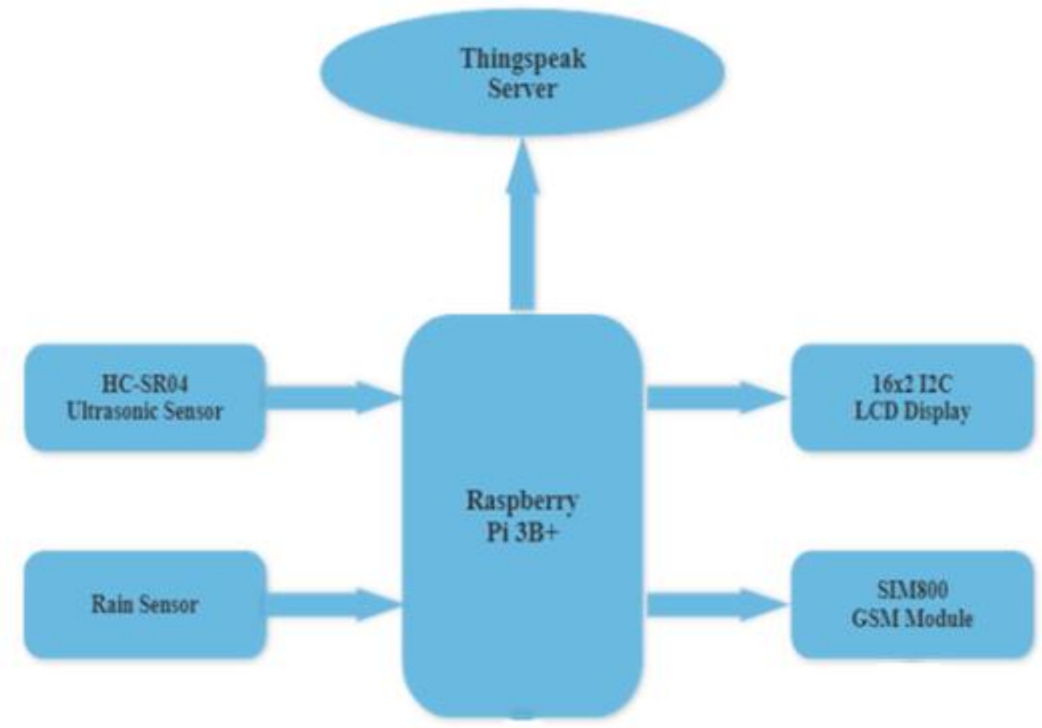

Fig-1: Flood Monitoring and Alerting System

COMPONENTS:

A. Raspberry Pi3 Model B+

Raspberry pi acts as the Processing Unit of this system. Raspberry pi detects the water level and rain intensity using an Ultrasonic Distance Sensor (HC-SR04) and rain s ensor respectively. It has 4 USB ports and the ports have been aligned with the Ethernet connector to make more streamlined design. The Raspberry Pi 3 range, boasting a 64-bit quad core processor running at 1.4GHz, dual-band 2.4GHz and 5GHz wireless LAN, Bluetooth 4.2/BLE, faster Ethernet, and PoE capability via a separate PoE HAT.

B. HC-SR04 Ultrasonic Sensor

The ultrasonic Sensor is used as inputs to the Raspberry pi and power supply of 5V is used to power up the system to function well. The ultrasonic distance sensor is used to detect the flood level at a high prone area of flood (maximum is 4m away from it).

C. Rain Sensor Module

The Rain Sensor is used to detect the rainfall or water drops. It works as a switch. Normally the switch is open condition. This sensor is consisting of mainly two parts, one is Sensing Pad and another one is the Sensor Module. When rainfall or water drops fall on the Sensing Pad surface, then the switch will be closed. The Sensor Module reads data from the sensor pad and processes the data and converts it into a digital/analog output. So, the sensor can provide both types of output Digital output (DO) and Analog output (AO).

D. PCF8591 ADC/DAC Module

PCF8591 is an 8-bit analog to digital or 8 bits digital to analog converter module meaning each pin can read analog values up to 256. It also has LDR and thermistor circuit provided on the board. This module has four analog input and one analog output. It works on I2C communication, so there are SCL and SDA pins for serial clock and serial data address. It requires 2.5-6V supply voltage and have low stand-by current.

E. SIM 800 GSM- Module

GSM modem is a wireless modem. It works on wireless network. This modem works like a dialup modem and sim is required for communication. In dialup modem the dat a is send or received through the fix telephonic line but in GSM modem data is send or received though the radio waves.

F. 16x2 I2C LCD Display

The PCF8574 device is an 8-bit I/O expander for the two-line bidirectional bus (I2C) is designed for 2.5-V to 5.5-V VCC operation. A typical I2C LCD display consists of a Hitachi's HD44780 based character LCD display and an I2C LCD adapter.

A regular LCD requires a lot of wires (parallel interface) to be connected with a Micro controller. The Serial LCD backpack built on PCF8574 IC uses the I2C bus to convert the parallel interface to a serial one. This needs only 2 wires SDA & SCL. The I2C backpack can be soldered on to the LCD. The I2C device has a HEX address by which a microcontroller can communicate with it. This is set by the 3 bits A0, A1, A2.

The device will generate an alarm system with three different colors of LEDs indicating three levels of detection for flood level and send Alert notification to the people on incoming flood in that area.

PROGRAM:

```
import random
import time


# Simulated flood data source (replace with real sensor data)
def get_flood_data():
    # Simulate flood data with random values (0 for no flood, 1 for flood)
    return random.randint(0, 1)


# Function to send alerts
def send_alert(message):
    # Replace this with code to send actual alerts (e.g., SMS, email, etc.)
    print("ALERT:", message)


# Main flood monitoring loop
while True:
    flood_data = get_flood_data()


    if flood_data == 1:
        send_alert("Flood detected! Take immediate action.")


    time.sleep(60)  # Check flood data every 60 seconds
```