Project Title: FLOOD MONITORING AND EARLY WARNING SYSTEM

Phase 1: Project Definition and Design Thinking

Project Definition:

In order to monitor water levels and issue early flood warnings via a public platform, the project entails placing IOT sensors close to water bodies and flood-prone areas. By delivering prompt alerts to the general public and emergency response teams, the goal is to improve flood preparedness and response. In this project, the goals are set, the IOT sensor network is designed, the warning platform is created, and the goals are then integrated using Python and IOT technology.

Introduction:

A flood is when water overflows and engulfs land. Flooding can happen when water from water bodies—such as a river, lake, or ocean—overflows and destroys levees, allowing part of the water to escape its regular confines. It can also happen when rainwater collects on wet ground and causes an area flood. While seasonal variations in precipitation and snowmelt will affect the size of lakes and other bodies of water, these changes are unlikely to be significant unless they cause property to flood or domestic animals to drown.

Despite significant technological advancement, humans are still unable to combat natural disasters. The truth is that there is no way to prevent or abolish natural disasters. However, the technology has advanced greatly in order to save lives. This project is entirely focused on warning the public about the impending flood so they can leave the high-risk area before it occurs.

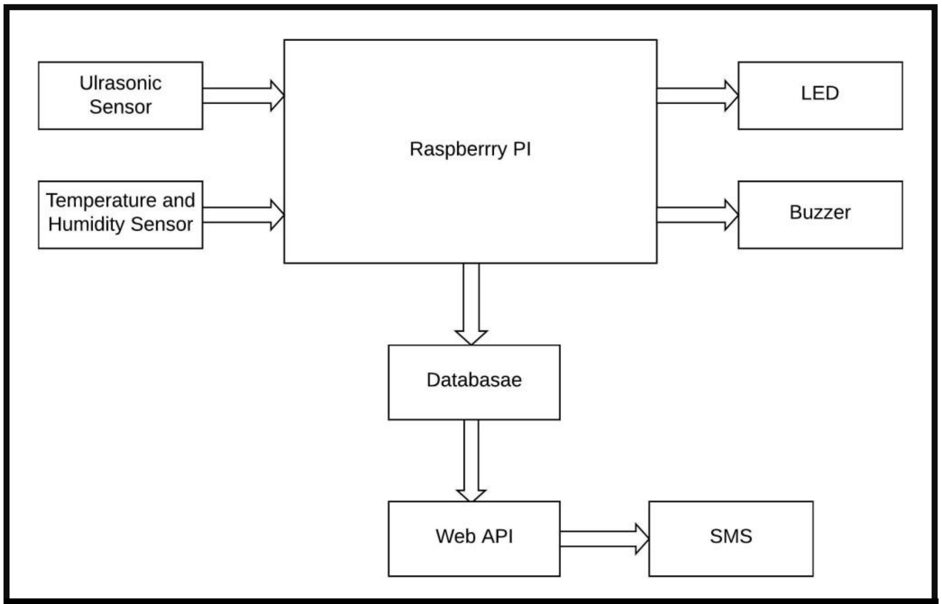
Objectives:

The main objectives of the project are:

- To read the real time temperature and humidity of the environment continuously.
- To give early information about the flood.
- To warn the people about the flood through SMS system using web API.
- To detect the level of water in real time.

IOT Sensor Network Design and Early Warning Platform:

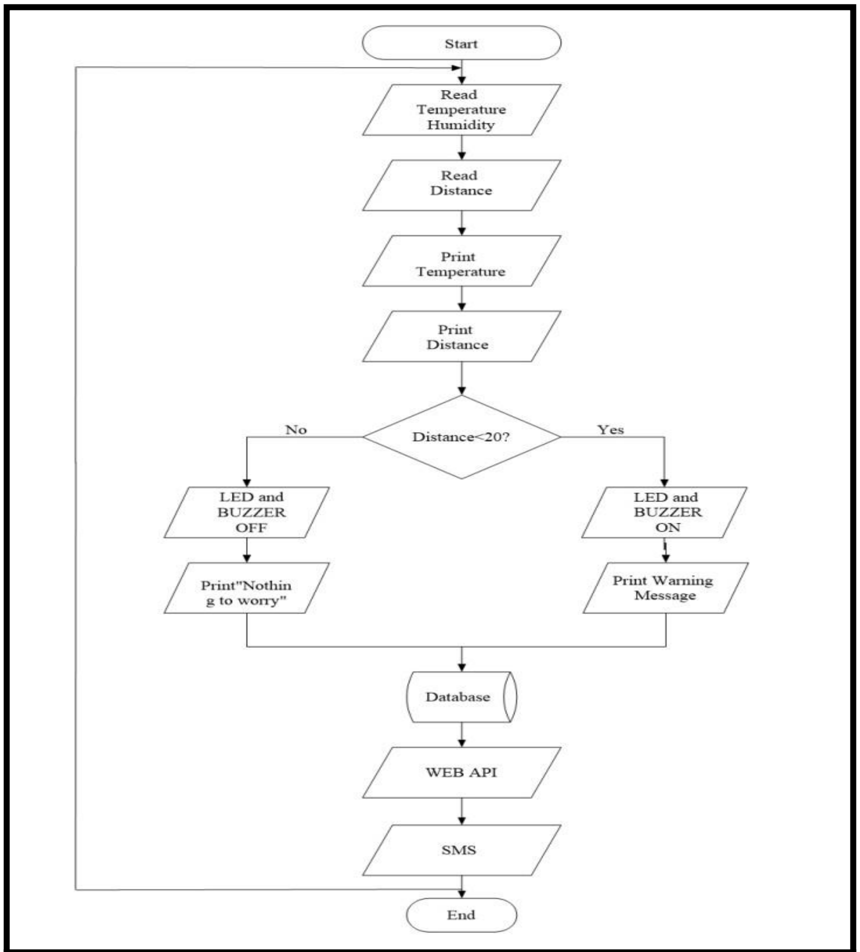
The sensors placed at the different places reads the data which are manipulated through the microcontroller and the values of the sensors are displayed. The values obtained repeatedly are send to database and through the web API, warning message is sent into the phone.



System Block Diagram

The project's brain is the raspberry Pi that was used. It is in charge of gathering, processing, storing, and communicating the data from the sensors, and then carrying out the events, in that order. Raspberry reads the information from the temperature and humidity sensor (DHT11) and the ultrasonic sensor (HCSR04). The Raspberry Pi then processes the sensor value that was obtained and displays it. The level of water is determined using the value from the ultrasonic sensor. The distance between the ultrasonic sensor and the river has a fixed threshold value. With changes in the water level, the distance measurement from the ultrasonic sensor is updated repeatedly. The led and buzzer will turn on if the distance value is less than the predetermined threshold value, signaling that there is a high likelihood that a flood will occur. The LED and buzzer will remain off if the distance value exceeds the predetermined threshold value, indicating that there is no need for concern. The local terminal of the Raspberry Pi also shows the temperature, humidity, and separation between the sensor and the river.

The values of the sensors are obtained repeatedly in the certain interval of time. So the real time values of the sensors are obtained. The values obtained are uploaded to the local server of the Raspberry Pi using the MySQL database. The data obtained in MySQL database from the Raspberry pi are date and time, temperature, humidity, distance of ultrasonic sensor and river and the remarks regarding the flood. The date and time is auto incremented since it doesn't require any sensor input data. The values of temperature and the humidity changes corresponding to the changes in the environmental temperature and humidity and gets updated in the database table. The main role here is of the ultrasonic sensor. The value of the ultrasonic sensor is updated repeatedly in certain interval of time and shows the distance. If the value of the distance is less than the threshold value then the warning message regarding flood will be displayed in the remarks and if the value of the distance is greater than the threshold value then remarks will display default message. The data in the database table are updated automatically every 6 seconds.



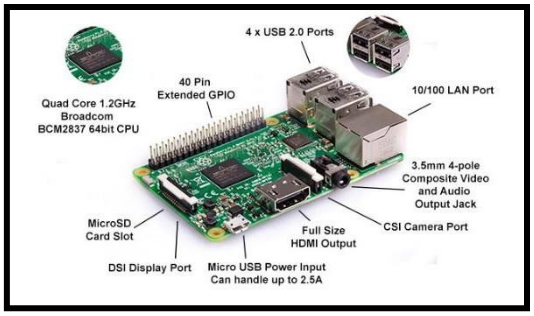
Flow Chart

Now, the main motive of the system to alert the people about the coming flood is done by the web API. The data from the database is linked to the web API. What the web API does is, it continuously keeps on reading the value of sensors from the data base. And when the value of distance becomes less than the threshold value the web API indicates it so by changing the color the trigger used there. The contact or phone number of the residents are also uploaded in the web API so, it quickly informs the local people about flood by sending the warning SMS to the people whose numbers are registered in it.

Hardware Description:

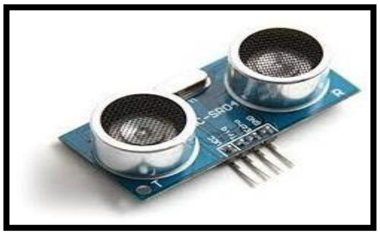
➢ Raspberry Pi:

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. The Raspberry Pi has the ability to interact with the outside world, and has been used in a wide array of digital maker projects, from music machines and parent detectors to weather stations and tweeting birdhouses with infra-red cameras.


4 x USB 2.0 Ports
40 Pin Extended GPIO
Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
10/100 LAN Port
3.5mm 4-pole Composite Video and Audio Output Jack
MicroSD Card Slot
Full Size HDMI Output
CSI Camera Port
DSI Display Port
Micro USB Power Input Can handle up to 2.5A

➢ Ultrasonic Sensor:

The sensor head emits an ultrasonic wave and receives the wave reflected back from the target. Ultrasonic Sensors measure the distance to the target by measuring the time between the emission and reception.

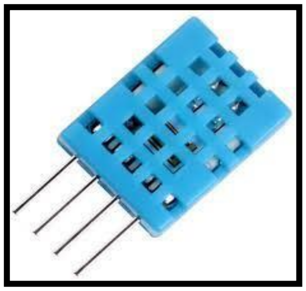


- **Buzzer:**

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or key stroke

- **Temperature and Humidity Sensors (DHT11):**

The DHT11 temperature and humidity sensor is a nice little module that provides digital temperature and humidity readings. It’s really easy to set up, and only requires one wire for the data signal.



- **LED:**

The circuit consists of a power supply (the Raspberry Pi), an LED that lights when the power is applied.

Software Description:

- ### Python:

Python is an interpreted, object-oriented programming language that has gained popularity because of its clear syntax and readability.

- ### Raspbian:

Raspbian is a Debian-based computer operating system for Raspberry Pi. There are several versions of Raspbian including Raspbian Buster and Raspbian Stretch.

- ### MySQL Database :

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL uses a standard form of the well-known SQL data language. It is very friendly to PHP, the most appreciated language for web development. The value obtained from external sensors can also be uploaded to the MySQL database and through which it can be monitored in web page, mobile application, SMS, etc.

- ### Cayenne:

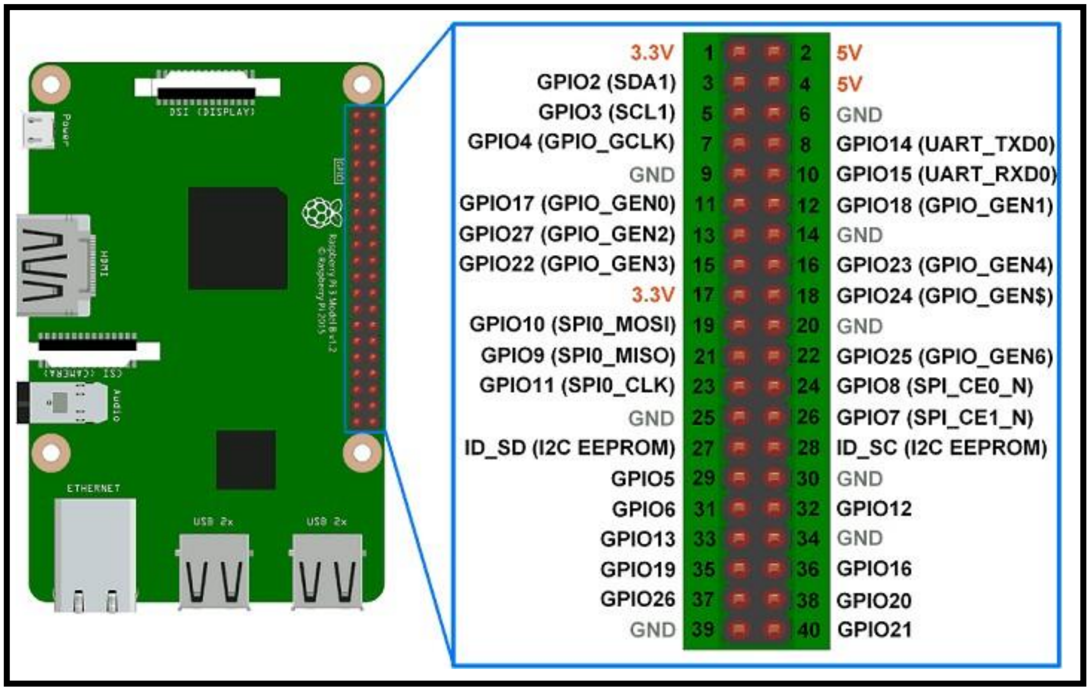
Cayenne is a very useful web API, through cayenne any database can be linked it any web sites, mobile app, SMS system etc. This is very applicable since it is free of cost of very easy to use.

Design And Implementation:

- ### Raspberry Pi:

Every complex embedded system needs a microcontroller or a microprocessor. This project uses a Raspberry Pi because it was the best fit for the needs of the project.

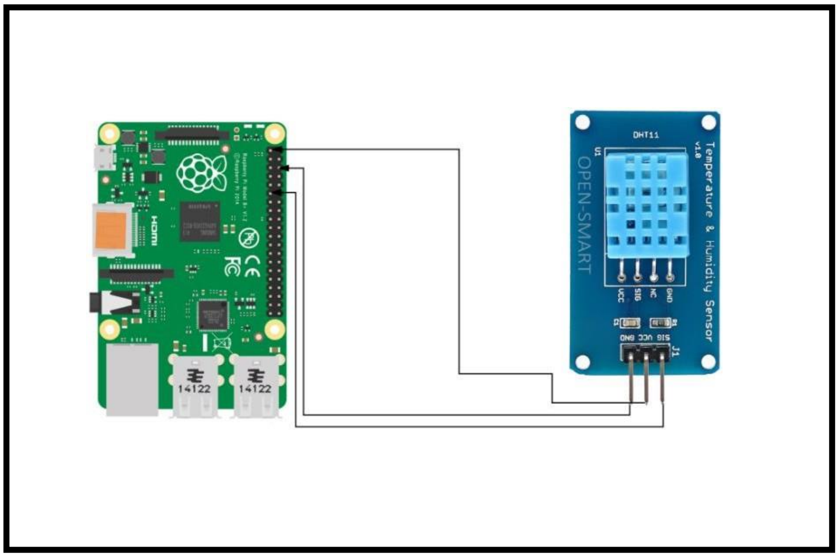
It is an ultra-cheap minicomputer with 5.5 cm width and 9 cm length. It consists of a component named System on Chip (SoC) which comprises of single core CPU with a supportive processor for computing floating points, GPU and RAM with 512 MB size (SD-RAM). Moreover, it consumes less power, which is just around 5 Retracted -7 watts.



3.3V	1	2	5V
GPIO2 (SDA1)	3	4	5V
GPIO3 (SCL1)	5	6	GND
GPIO4 (GPIO_GCLK)	7	8	GPIO14 (UART_TXD0)
GND	9	10	GPIO15 (UART_RXD0)
GPIO17 (GPIO_GEN0)	11	12	GPIO18 (GPIO_GEN1)
GPIO27 (GPIO_GEN2)	13	14	GND
GPIO22 (GPIO_GEN3)	15	16	GPIO23 (GPIO_GEN4)
3.3V	17	18	GPIO24 (GPIO_GEN$)
GPIO10 (SPI0_MOSI)	19	20	GND
GPIO9 (SPI0_MISO)	21	22	GPIO25 (GPIO_GEN6)
GPIO11 (SPI0_CLK)	23	24	GPIO8 (SPI_CE0_N)
GND	25	26	GPIO7 (SPI_CE1_N)
ID_SD (I2C EEPROM)	27	28	ID_SC (I2C EEPROM)
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21

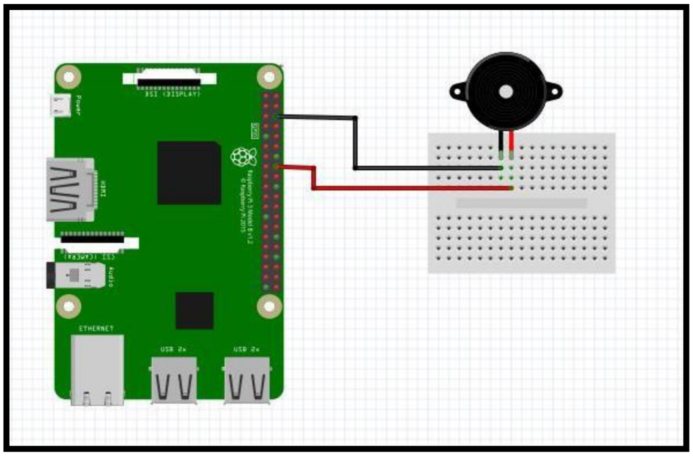
- ➢ **DHT11 Sensor:**

The DHT11 temperature and humidity sensor is a nice little module that provides digital temperature and humidity readings. It's really easy to set up, and only requires one wire for the data signal. These sensors are popular for use in remote weather stations, soil monitors, and home automation systems. Programming the DHT11 and connecting it to a Raspberry Pi is pretty simple too.



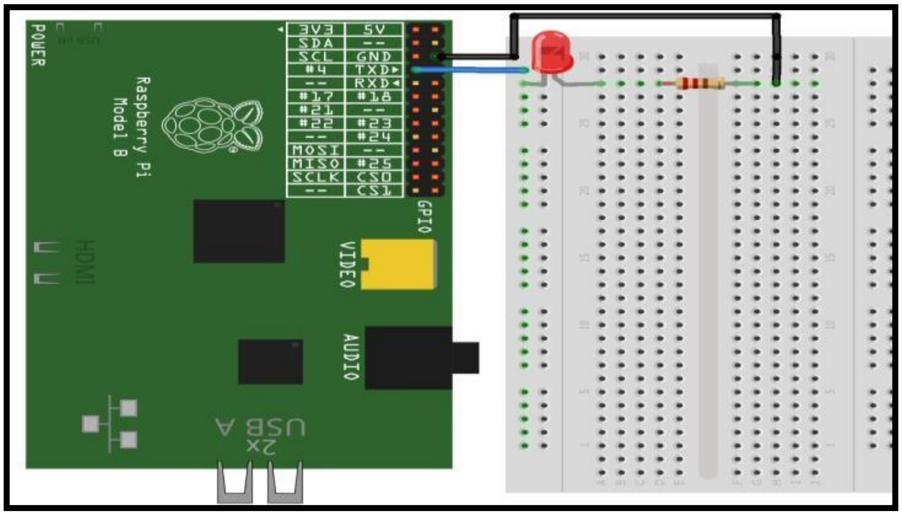
➢ Buzzer:

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or key stroke.



➢ LED:

The circuit consists of a power supply (the Raspberry Pi), an LED that lights when the power is applied, and a resistor to limit the current that can flow through the circuit. You will be using one of the ‘ground’ (GND) pins to act like the ‘negative’ or 0 volt ends of a battery. The ‘positive’ end of the battery will be provided by a GPIO pin. Here we will be using pin 18. When they are ‘taken high’, which means it outputs 3.3 volts, the LED will light.



➢ Ultrasonic Sensor With pi:

There are four pins on the ultrasound module that are connected to the Raspberry .The pins are connected as following example:

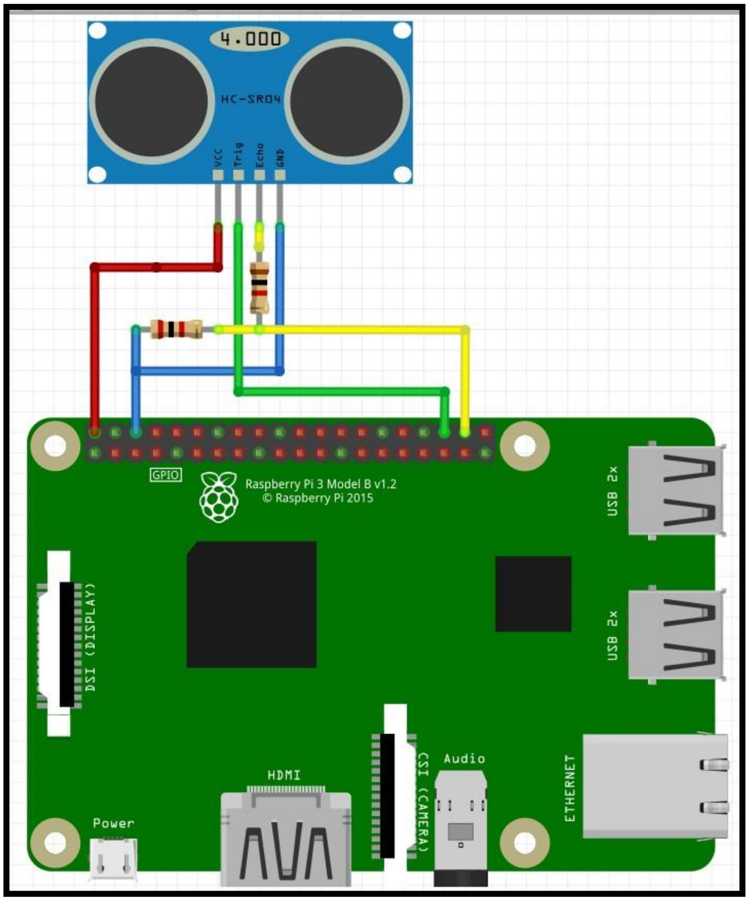
VCC to Pin 2 (VCC)

GND to Pin 6 (GND)

TRIG to Pin 12 (GPIO18)

connect the 330Ω resistor to ECHO. On its end you connect it to Pin 18 (GPIO24) and through a 470Ω resistor you connect it also to Pin6 (GND).

The GPIO pins only tolerate maximal 3.3V. The connection to GND is to have an obvious signal on GPIO24. If no pulse is sent, the signal is 0 (through the connection with GND), else it is 1. If there would be no connection to GND, the input would be undefined if no signal is sent (randomly 0 or 1), so ambiguous. Below here is the structure as a circuit diagram:



Integration Approach:

Ultrasonic and water level sensors are used to detect the rise in water level. for monitoring changes in temperature and humidity Temperature and humidity sensors are used. The microcomputer reads the data from the DTH11 and HC-SR04 and analyzes it to determine the water level. The microcomputer activates the LED and buzzer if the water level is below the specified threshold value. Additionally, the database receives the information downloaded from the microcomputer. In a database table, the values of the sensors that are updating in real time can be seen. The trigger is set, and the database table's content is now connected to the web API (Application Programming Interface). Now, the trigger is set off when the water level reaches the threshold value, and the web API sends the SMS to the associated phone number.