IOT BASED FLOOD DETECTING SYSTEM

A Project report submitted in partial fulfilment

of the requirements for the degree of B. Tech in Electronic and Communication Engineering

By

D.NAVEEN NISCHAL(18BEC011)

N.MAHANTESH(18BEC025)

B.PRAMODH (18BEC007)

SHOBARAJ (18BEC044)

KAVYA F (18BEC021)

Under the guidance of

DR. JAGADEESHA R BHATT

ASST.PROFESSOR

DEPARTMENT OF

ELECTRONICS AND COMMUNICATION ENGINEERING

Dharwad

ज्ञानेन विकासः

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DHARWAD

CONTENTS:

ABSTRACT

Flood is one of the natural disaster that cannot be avoided totally.Every year death rate due to flood increases because of absence of early warning.To solve this problem this project demonstrates the idea and implementing of a Flood Detecting System through IOT technology.This system comprises of three parts.First part is detecting the water level using ultra sonic sensor.The second part is sending the information to the cloud using blynk.The third part is sending the alert message through Blynk.IoT is the technology used for monitoring, collecting, controlling and connecting the system to worldwide, which is the more efficient and advanced solution for accessing the information in the world.This system proposes a real-time flood detecting system.

INTRODUCTION

The Internet of Things is the network of physical objects or things embedded with electronics,software ,sensors and network connectivity,which enables the objects to collect and exchange data.IOT allows objects to be sensed and controlled remotely across existing network infrastructure ,creating opportunities for more direct integration between the physical world and computer based systems and resulting in improved efficiency,accuracy and economic benefit.

IoT heightens the degree of automation which creates an essential impact in reducing the damage occurring due to natural disasters.For example, floods are one of the most destructive natural disaster.So in this lets see how we can reduce damage caused due to floods with the help of Internet of Things.

Flood is a natural disaster where an area of land that gets instantly submerged in water. Flood may occur in many areas in different ways due to overflow of streams, rivers, lakes or oceans or as a result of excessive rain. A warning system is necessary to take precautionary measures and be more prepared to overcome its effects .

In this modern era, there are multiple systems working and are deployed at different locations but the alert notification is passed to government agencies and this ends up in slowing down the process. The reason behind this is that floods are very spontaneous disasters and government agencies have to follow multiple steps before reaching a decision. In this case, awareness among the people is very necessary along with the government officials. So, that a combined and better result will be achieved.

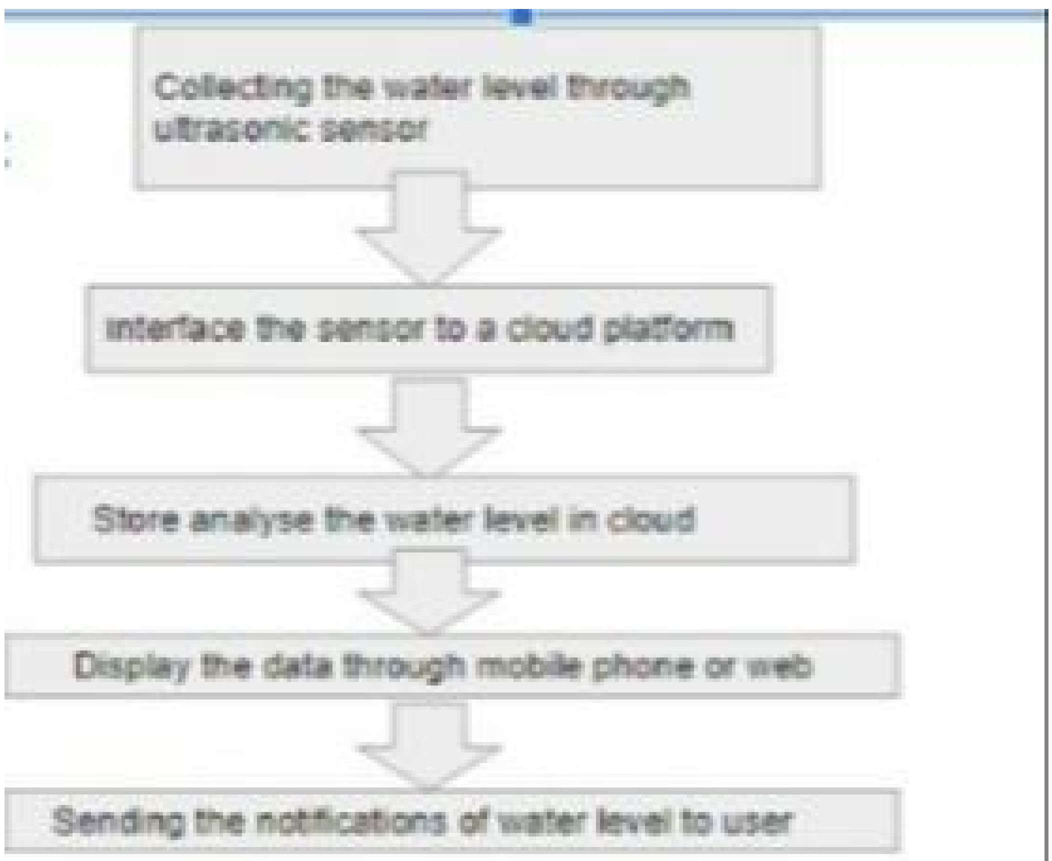
In my proposed system the water level can be monitored in real-time using a cell phone, no matter if you are in a meeting, outside your home, etc. You will be able to see everything in real-time. You can check the App at any time and find out about the water level, if it’s increasing or decreasing. Different levels can be defined. Alert messages are generated automatically when the desired conditions are met.The sensor that measures water level is an ultrasonic sensor

The aim of the project is to develop a smart, affordable and portable flood monitoring system with advanced facility of storing and displaying the measurement of water level in cloud.

PROPOSED MODEL OF SYSTEM:

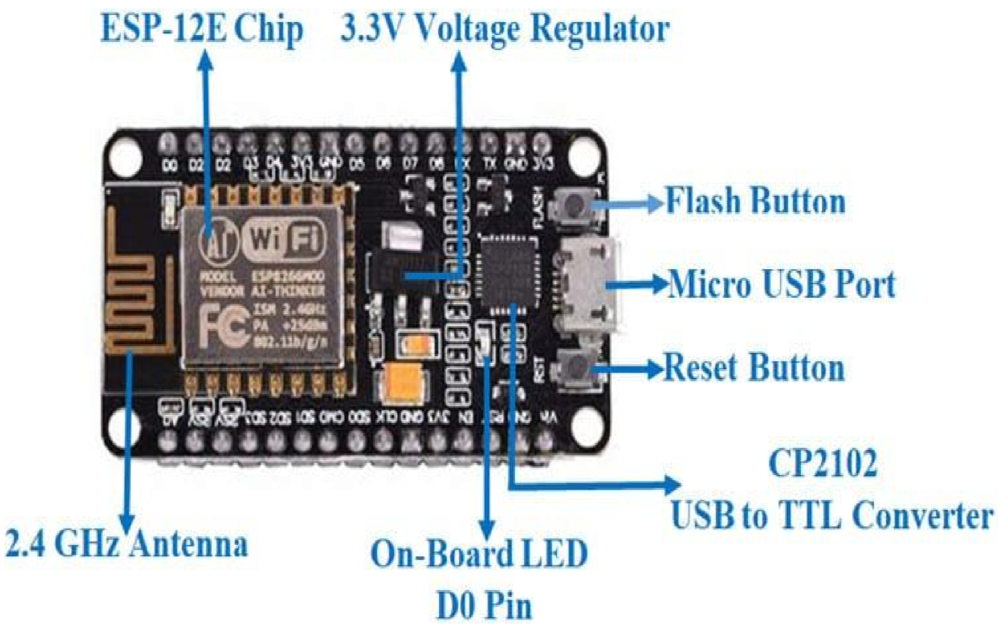
The Flood Detecting System is divided into software and hardware modules.In this project we are using the NODEMCU as the main controlling unit in the system.The ultrasonic sensor is placed to determine the water level.The sensor values gathered from the senor are given to main controlling unit and it get signals forms sensor and transmit those sensor values to the cloud through wifi module .The framework continuously uploads the sensor data to cloud.

Flow chart:



NODEMCU:

The NodeMCU ESP8266 development board comes with the ESP-12E module containing ESP8266 chip having Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency. NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi / Bluetooth and Deep Sleep Operating features make it ideal for IoT projects.NodeMCU can be powered using Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface.



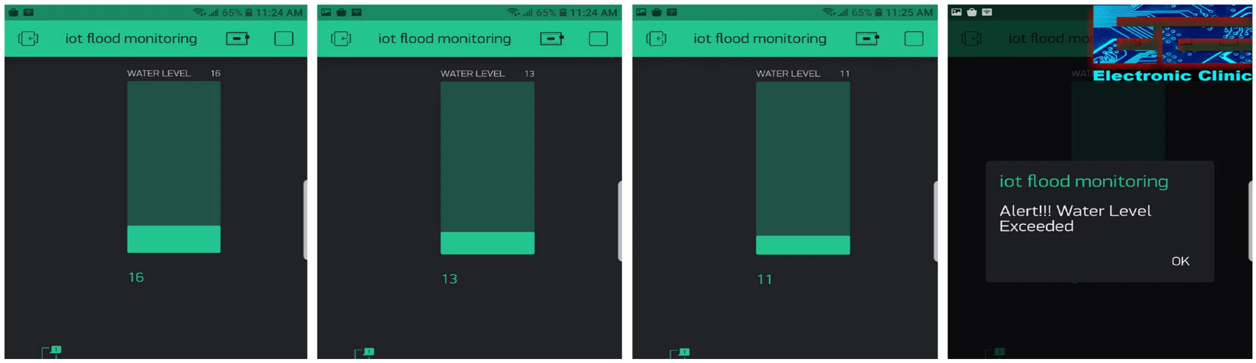
ULTRASONIC SENSOR:

The ultrasonic sensor works on the principle of ultrasound waves which is used to determine the distance to an object. An ultrasonic sensor generates high-frequency sound waves.. By using the time required for the echo to reach the receiver, we can calculate the distance to an object we are targeting.Ultrasonic distance sensor consists of two ultrasonic transducers. Among them, one acts as a transmitter which converts the electrical pulse of a microcontroller into ultrasonic sound pulse and the receiver receives the transmitted pulses. If it receives them, then it produces an output pulse whose time period can be used to determine the distance from the object.

Distance= (Time x Speed of Sound in Air (340 m/s))/2

BLYNK:

Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, visualize it and do many other cool things. There are three major components in the platform. Blynk Server - responsible for all the communications between the smartphone and hardware.



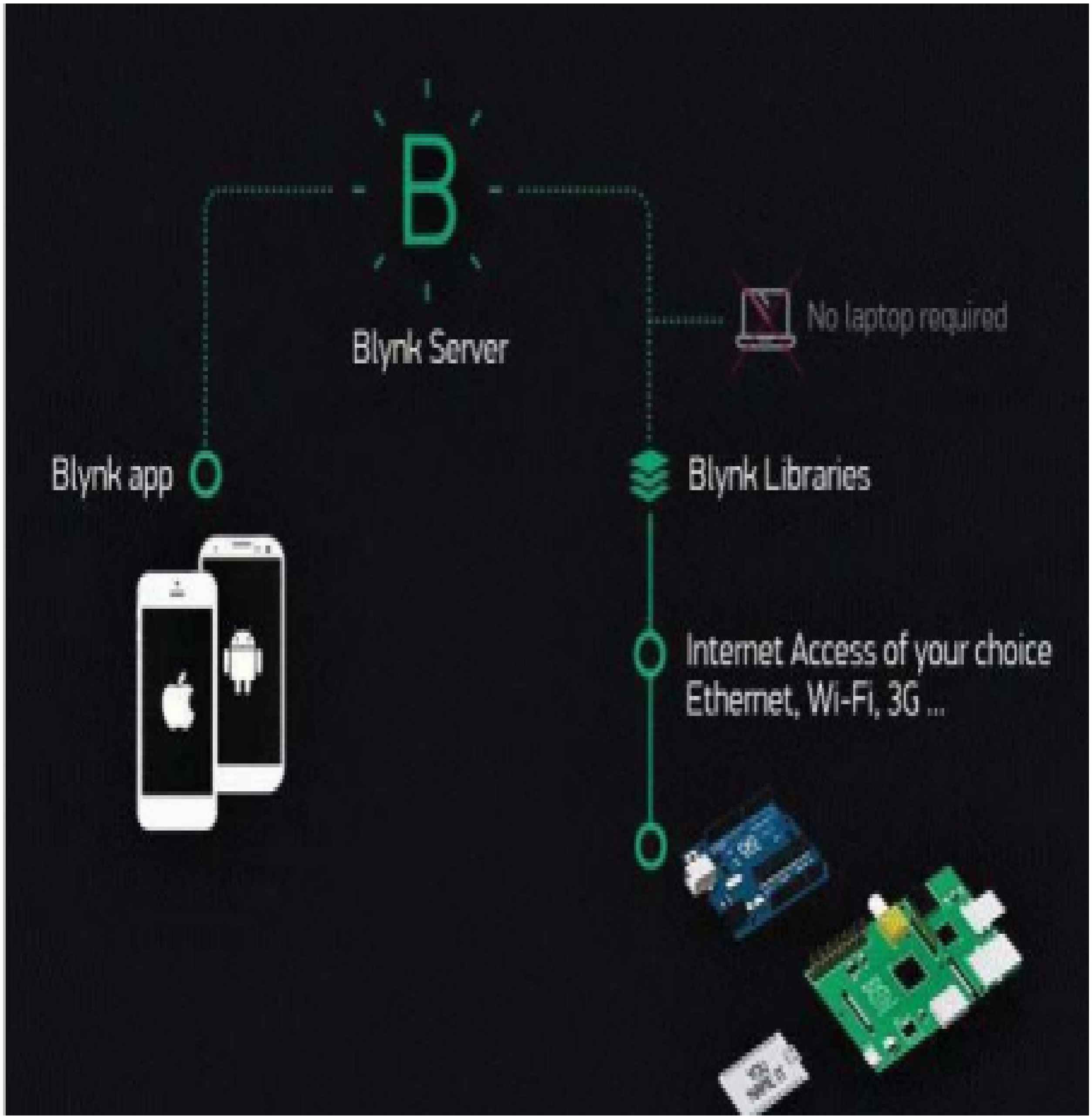
WORKING OF THE SYSTEM:

The system consisted of two subsystems:First subsystem consisted of an ultrasonic sensor which was interfaced with NODEMCU.

The second subsystem consists of a blynk app module and things peak these are platforms where the readings obtained from sensors are displayed.

The number of people who have access to smartphones is increasing exponentially. This arises a need to create an app through which the weather parameters can be monitored . The app used in this paper is blynk, it is available free of cost and has a customizable user interface.

The working of blynk app is represented in fig. below. In the app, a button widget is placed in the screen and is configured as a virtual pin. Once the wifi connection is established, the project turns online and when the button is pressed the LED connected glows indicating that NodeMCU can communicate with the app. When the testing is successful, the next task is to display the real time water level in the app. The code for the respective sensor is uploaded, and the water level value will be automatically displayed received from the sensor.



B
Blynk Server
No laptop required
Blynk app
Blynk Libraries
Internet Access of your choice
Ethernet, Wi-Fi, 3G ...

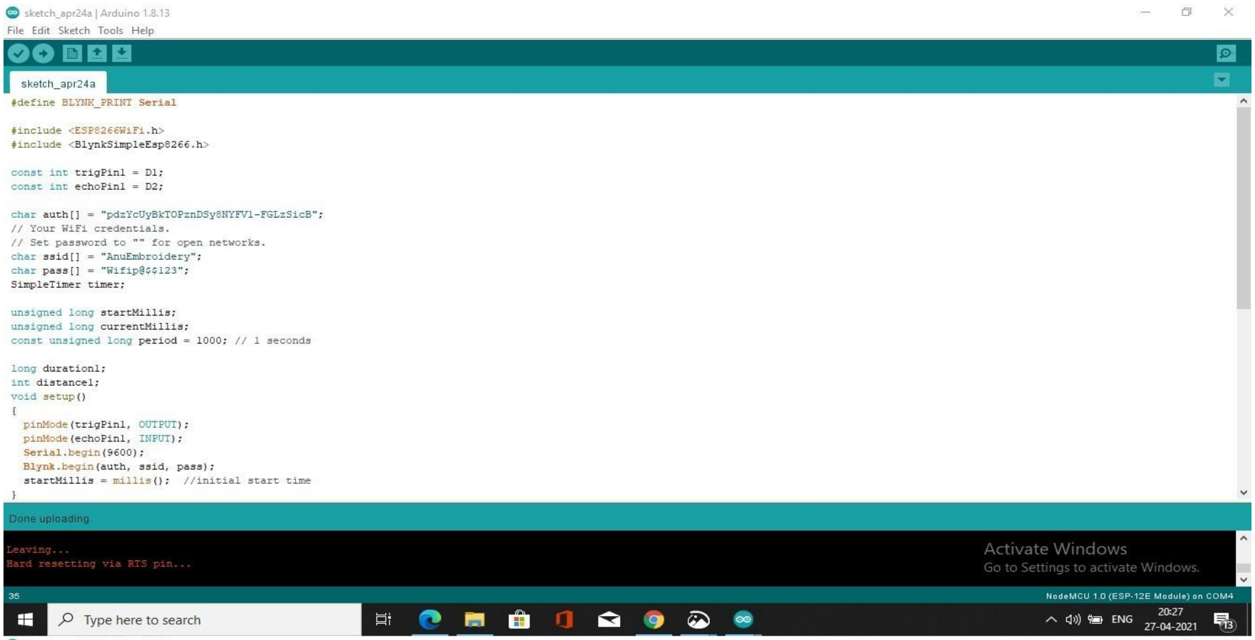
IMPLEMENTATION OF SYSTEM ON HARDWARE PLATFORM:

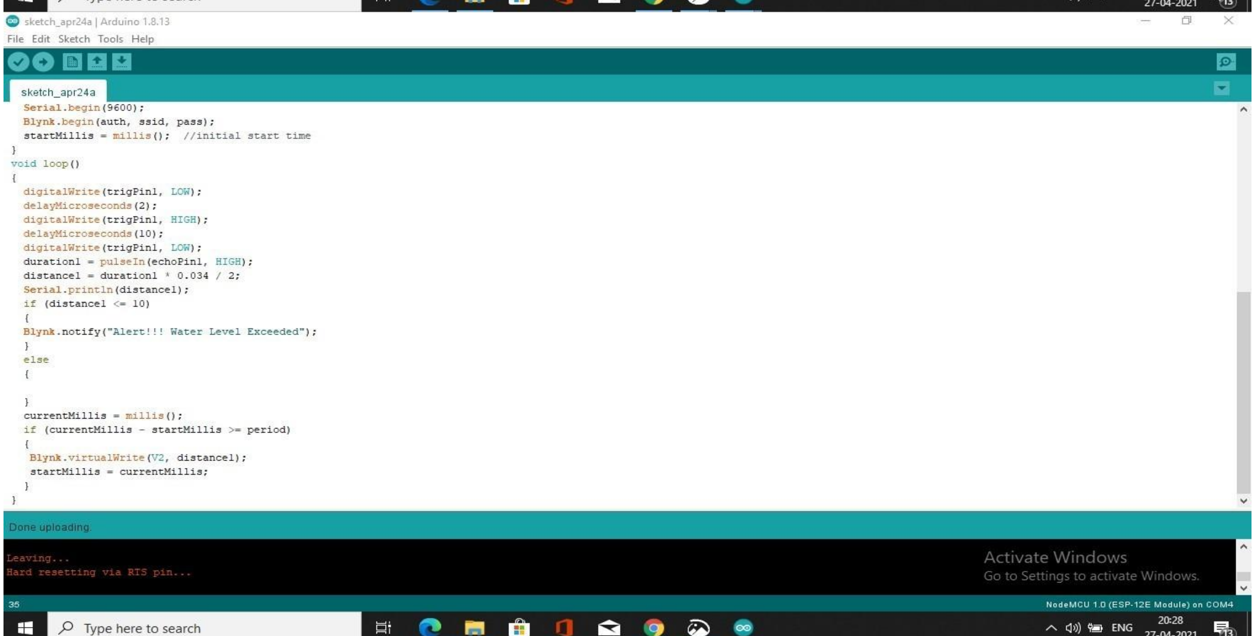
Here we are interfacing NodeMCU ESP8266 ,Ultrasonic sensor as shown in circuit diagram.

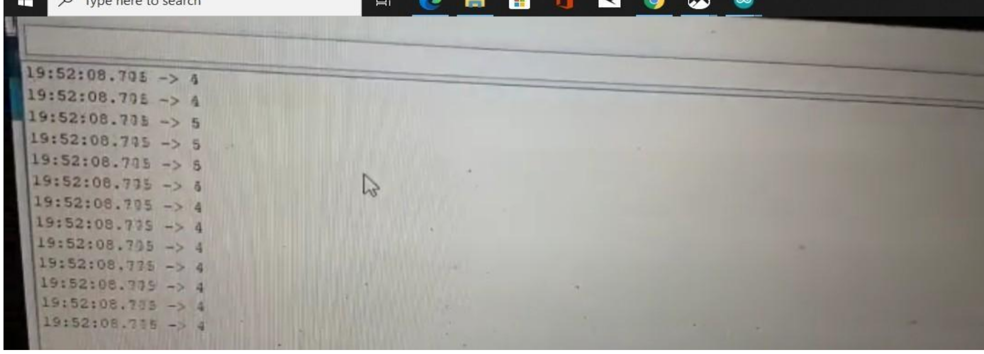
Interfacing HC-SRO4 Sensor to Nodemcu

Ultrasonic HC-SR04	ESP8266
Vcc pin	Vin pin
Trig pin	D1
Echo pin	D2
GND pin	GND

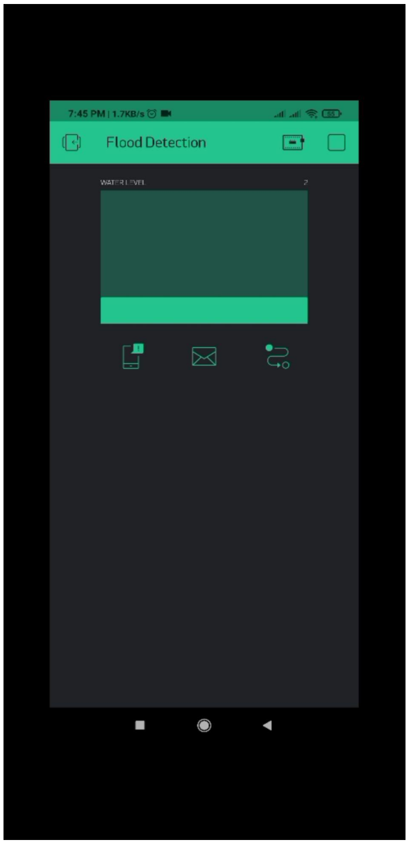
RESULTS:

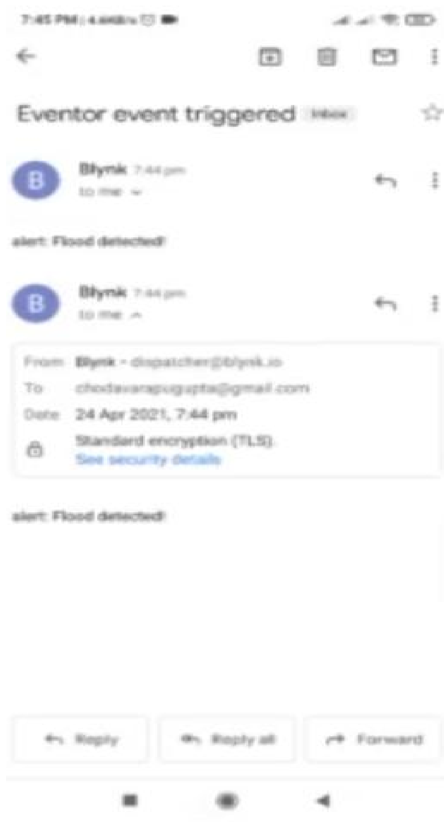

sketch_apr24a | Arduino 1.8.13
File Edit Sketch Tools Help
sketch_apr24a
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
const int trigPin1 = D1;
const int echoPin1 = D2;
char auth[] = "pdzYcUyBkTOPznDSy8NYFV1-FGLzSicB";
// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "AnuEmbroidery";
char pass[] = "Wifip@$$123";
SimpleTimer timer;
unsigned long startMillis;
unsigned long currentMillis;
const unsigned long period = 1000; // 1 seconds
long duration1;
int distance1;
void setup()
{
pinMode(trigPin1, OUTPUT);
pinMode(echoPin1, INPUT);
Serial.begin(9600);
Blynk.begin(auth, ssid, pass);
startMillis = millis(); //initial start time
}
Done uploading.
Leaving...
Hard resetting via RTS pin...
Activate Windows
Go to Settings to activate Windows.
NodeMCU 1.0 (ESP-12E Module) on COM4
Type here to search
ENG
20:27
27-04-2021


sketch_apr24a | Arduino 1.8.13
File Edit Sketch Tools Help
sketch_apr24a
Serial.begin(9600);
Blynk.begin(auth, ssid, pass);
startMillis = millis(); //initial start time
}
void loop()
{
digitalWrite(trigPin1, LOW);
delayMicroseconds(2);
digitalWrite(trigPin1, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin1, LOW);
duration1 = pulseIn(echoPin1, HIGH);
distance1 = duration1 * 0.034 / 2;
Serial.println(distance1);
if (distance1 <= 10)
{
Blynk.notify("Alert!!! Water Level Exceeded");
}
else
{
}
currentMillis = millis();
if (currentMillis - startMillis >= period)
{
Blynk.virtualWrite(V2, distance1);
startMillis = currentMillis;
}
}
Done uploading.
Leaving...
Hard resetting via RTS pin...
Activate Windows
Go to Settings to activate Windows.
NodeMCU 1.0 (ESP-12E Module) on COM4
Type here to search
ENG
20:28
27-04-2021


19:52:08.705 -> 4
19:52:08.705 -> 4
19:52:08.705 -> 5
19:52:08.705 -> 5
19:52:08.705 -> 5
19:52:08.705 -> 5
19:52:08.705 -> 4
19:52:08.705 -> 4
19:52:08.705 -> 4
19:52:08.705 -> 4
19:52:08.705 -> 4
19:52:08.705 -> 4
19:52:08.705 -> 4

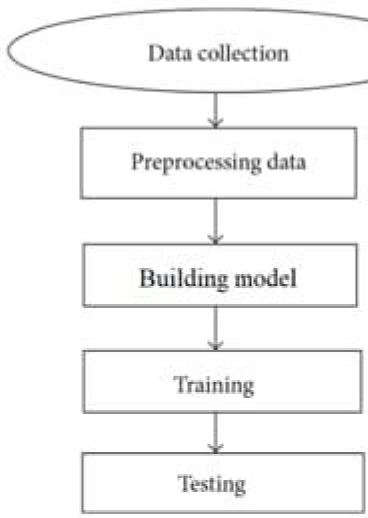
BLYNK RESULT:





IMPLEMENTATION IN SOFTWARE USING MACHINE LEARNING:

FLOW CHART:



FLOOD PREDICTION CODE:

https://github.com/NaveenNischal123/IOT-PROJECT

SCIKIT LEARNING (SKLEARN) FOR ML

Scikit-learn is a machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy.

The algorithms used for prediction are Logistic regression,Decision Tree classification, KNN Classifier.

Using KNN Classifier technique:

The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems.It classifies the data point on how its neighbor is classified. KNN classifies the new data points based on the similarity measure of the earlier stored data points.

After applying KNN Classifier model to our data the predicted results are shown below fig

.RESULTS:

1. KNN Classifier

```
[4] clf = neighbors.KNeighborsClassifier()
    knn_clf = clf.fit(x_train,y_train)

[5] #Let's predict chances of flood
    y_predict = knn_clf.predict(x_test)
    print('predicted chances of flood')
    print(y_predict)

     predicted chances of flood
     [1 0 0 0 0 0 1 0 1 0 0 0 0 0 1 1 1 1 1

[ ] #Actual chances of flood
    print("actual values of floods:")
    print(y_test)

     actual values of floods:
     70      1
     46      1
     102     0
     9       0
     116     0
     92      0
     42      1
     12      0
     14      1
     72      0
     34      0
     17      0
     78      0
     86      0
     109     1
```

:

```
17      0
78      0
86      0
109     1
15      1
1       1
108     0
22      1
89      0
65      0
5       0
24      1
85      0
Name: FLOODS, dtype: int64
```

```
[ ] from sklearn.model_selection import cross_
```

```
[ ] knn_accuracy = cross_val_score(knn_clf,x_t
```

```
[ ] knn_accuracy.mean()

    0.7083333333333334
```

The mean accuracy of KNN Classifier is 0.708

2)Using Logistic Regression:

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of the target or dependent variable is dichotomous, which means there would be only two possible classes.Large sample sizes are required for logistic regression to give adequate numbers in the two classifications of the reaction variable. The more illustrative factors, the bigger the example size required.

After applying Logistic Regression model to our data the predicted results are shown below fig

RESULTS:

```
[ ] from sklearn.model_selection import cross_
    from sklearn.linear_model import LogisticR

    lr = LogisticRegression()
    lr_clf = lr.fit(x_train_std,y_train)

    lr_accuracy = cross_val_score(lr_clf,x_tes
```

```
[ ] lr_accuracy.mean()
```

```
0.625
```

```
[ ] y_predict = lr_clf.predict(x_test_std)
    print('Predicted chances of flood')
    print(y_predict)
```

```
Predicted chances of flood
[1 0 0 1 0 0 1 0 1 0 0 1 0 0 1 1 1 1 1
```

```
[ ] print('Actual chances of flood')
    print(y_test.values)
```

```
Actual chances of flood
[1 1 0 0 0 0 1 0 1 0 0 0 0 0 1 1 1 0 1
```

```
[ ] from sklearn.metrics import accuracy_score
    print("\naccuracy score: %f"%(accuracy_sco
    print("recall score: %f"%(recall_score(y_t
    print("roc score: %f"%(roc_auc_score(y_tes
```

```
accuracy score: 83.333333
recall score: 88.888889
roc score: 84.444444
```

The accuracy of Logistic Regression is 84%

3)Using Decision Tree Algorithm:

Decision tree learning is one of the predictive modelling approaches used in statistics, data mining and machine learning. It uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). The final variables in a DT contain a discrete set of values where leaves represent class labels and branches represent conjunctions of features labels.

RESULTS:

3. Decision tree classification

```
[ ] from sklearn.tree import DecisionTreeClass
    dtc_clf = DecisionTreeClassifier()
    dtc_clf.fit(x_train,y_train)
    dtc_clf_acc = cross_val_score(dtc_clf,x_tr
    dtc_clf_acc
```

```
array([0.71875    , 0.64516129, 0.61290
```

```
[ ] #Predicted flood chances
    y_pred = dtc_clf.predict(x_test)
    print(y_pred)
```

```
[1 1 0 0 0 0 1 0 1 0 0 1 1 1 1 0 0 1 1
```

```
[ ] #Actual flood chances
    print("actual values:")
    print(y_test.values)
```

```
actual values:
[1 1 0 0 0 0 1 0 1 0 0 0 0 0 1 1 1 0 1
```

```
[ ] from sklearn.metrics import accuracy_score
    print("\naccuracy score:%f"%(accuracy_scor
    print("recall score:%f"%(recall_score(y_te
    print("roc score:%f"%(roc_auc_score(y_test
```

```
accuracy score:70.833333
recall score:77.777778
roc score:72.222222
```

The accuracy of Decision Tree is 70%

CONCLUSION:

Flood forecasting and the issuing of flood warnings are effective ways to reduce damage. The proposed system will be efficient because it has better coordination of monitoring, communication and transmission technologies which are adaptable to background conditions.Sensors are installed to monitor the water level using IDE (Integrated Development Environment) received data and result analysis will be sent to the end user through Wi-Fi. NodeMCU controller used to control the sensors, and it receives the data from sensors, and sends it to end users through the cloud. This type of model can be used for reducing the damage caused by the floods .

REFERENCES:

1)H. N. Do, M. Vo, V. Tran, P. V. Tan, and C. V. Trinh, “An Early Flood Detection System Using Mobile Networks,” International Conference on Advanced Technologies for Communications (ATC), pp. 599–603, 2015.

2)M. Mousa, X. Zhang, and C. Claudel, “Flash Flood Detection Proceedings of the International Conference on Intelligent Computing and Control Systems (ICICCS 2019) IEEE Xplore Part Number: CFP19K34-ART; ISBN: 978-1-5386-8113-8 978-1-5386-8113-8/19/$31.00 ©2019 IEEE 1485 in Urban Cities Using Ultrasonic and Infrared Sensors,” IEEE Sensors Journal, vol. 16, no. 19, pp. 7204–7216, 2016

3)M. Khalaf, A. J. Hussain, D. Al-Jumeily, P. Fergus, and I. O. Idowu, “Advance Flood Detection and Notification System based on Sensor Technology and Machine Learning Algorithm,” ” pp. 105–108, 2015.

4)K. R. Dashpute, V. B. Gaikwad, and S. S. Sawkar, “Flood detection using iot,” no. 2, pp. 1289–1292, 2018.

5)S. Fakhruddin and Y. Chivakidakarn, "A case study for early warning and disaster management in Thailand," International Journal of Disaster Risk Reduction, vol. 9, pp. 159-180, 20