IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 9, Issue 5, May 2021

DOI 10.17148/IJIREEICE.2021.9525

# IOT based Smart Irrigation System Using Raspberry pi

**Mayanka Sikarwar**

Student, Dept. of Electrical Engineering, MITS Gwalior, M.P., India

**Abstract –**India's population is around 1.39 billion (April 2021 estimation).The rising population growth need for increased agriculture production. Irrigated agriculture has been important source to increase agriculture production.IOT based smart irrigation system is to create an IOT base automated irrigation mechanism which turns the water motor pump ON and OFF as command received through IOT platform..Secondly, Our economy is mostly affected by agriculture. Just to check the level of water farmer have to go to farm and start water pump. In this paper, I made a smart irrigation system which helps the farmer to control their water pump which is on farm from any place .they need not required to go to farm to check the level of water in soil or to start or stop the water pump. They can monitor their farm from anywhere in the world.

**Key Words:** Raspberry pi, Sensor, Smart Irrigation, Temperature and Humidity, jQuery, Python

## I.INTRODUCTION

As our technology becomes increasingly connected through the rise of IOT and cloud Computing, we have not only benefited our lives but our planet as well . Smart farming has taken the world by storm as farmer have learned to embrace the world to IOT. These farmers have been able to precisely monitor the conditions of their produce as well as control their resources. Agri-Tech companies like Green house have able to solve the difficulty of a rapidly growing population's need for sustainable food and farming.

The raspberry pi is the main component of the system . Python programming language and jQuery (library of java script) is used for automation purpose. The raspberry pi model is programmed such that if the temperature and humidity goes above the threshold value motor will be turned ON and if it goes below threshold value motor will be turned ON.

## II. COMPONENT DESCRIPTION

**A. Raspberry pi** - Raspberry Pi is a mall sized single board computer which can do entire job that an average desktop computer do like spread sheets, Word processing, Internet, Programming, Games etc. The GPIO pins on Raspberry pi allow you to connect your Pi up to all sorts of electronic devices. This guide talks you through how to set up and control the inputs and outputs using python. GPIO pins are metal spikes that stick out of a Raspberry Pi. In this project Raspberry Pi 3 B+ model is used. Number of GPIO pins in this model is 40 and number of input or output pins are 28

Pins-

- Pins are either power points or i/o pins
- Power pins are hard wired to 0v(ground), +3.3v or +5v. You can't change them.
- I/O pins can be set to either an input or an output
- Inputs read a digital value into the Raspberry pi(e.g. has a switch been passed)
- Output send a digital value out from the Raspberry pi(e.g. switch an LED ON or OFF)

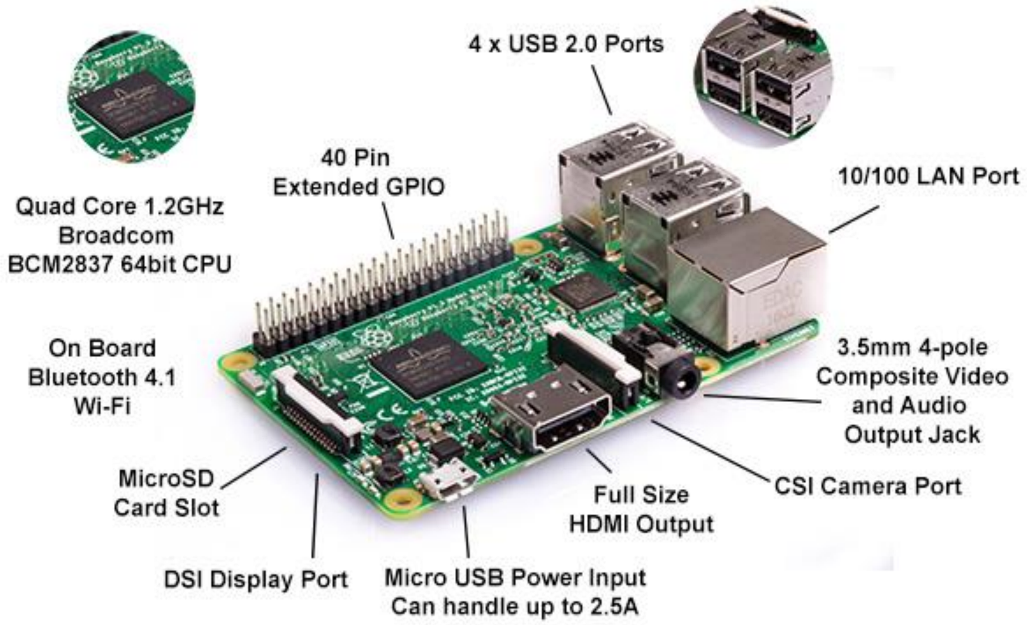
4 x USB 2.0 Ports
40 Pin Extended GPIO
10/100 LAN Port
Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
On Board Bluetooth 4.1 Wi-Fi
3.5mm 4-pole Composite Video and Audio Output Jack
MicroSD Card Slot
CSI Camera Port
Full Size HDMI Output
DSI Display Port
Micro USB Power Input Can handle up to 2.5A

Fig. 1- Raspberry pi

**B. DHT Sensor-** For monitoring the temperature and humidity we use the DHT11 sensor. The DHT11 detects the water vapor by measuring the electrical resistance between two electrodes. The humidity sensing component is a moisture holding substrate with electrodes applied to the surface. When water vapor is absorbed by the substrate, ions are released by the substrate which increases the conductivity between the electrodes. The change in resistance between the two electrodes is proportional to the relative humidity. Higher relative humidity decreases the resistance between the electrodes, while lower relative humidity increases the resistance between the electrodes. The principle in the humidity sensor is they consist of a humidity sensing component, a NTC temperature sensor (or thermistor) and an IC on the back side of the sensor. For measuring humidity, they use the humidity sensing component which has two electrodes with Humidity moisture holding substrate between them.

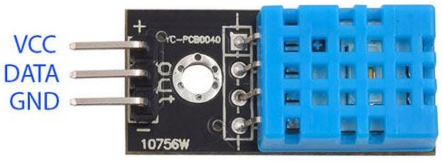

Fig. 2- DHT Sensor

**C. Motor Driver-** Motor driver L298N is used in this project . It is a high power driver module for driving DC motor. Here it is used to drive water pump. Instead of motor driver we can use relay as well.

***Note****-I have used dc motor driver because i used dc motor pump here ,for Ac supply use relay or ac motor driver.*

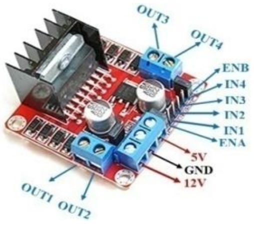

Fig 3- Motor Driver

DOI 10.17148/IJIREEICE.2021.9525

**D. Water Pump-** The pump is a submersible water pump, meaning that it must be completely submerged in a water reservoir in order to pump water. Specification-

1- Operating Voltage : 2.5 - 6V
2-Operating Current : 130 - 220mA
3-Flow Rate : 80 - 120 L/H
4-Maximum Lift : 40 - 110 mm
5-Outlet Outside Diameter: 7.5 mm
6-Outlet Inside Diameter: 5 mm

## III. Methodology-

The complete setup of the transmitter and receiver sections is shown in Figure 4. Hardware connection is shown in figure 5. The power supply of the microprocessor (Raspberry Pi) is turned on. After turning on the temperature and humidity sensor collects the temperature and humidity values and sends them to the raspberry pi. If system is not connected with cloud, click on connect with cloud. As user click on connect with cloud, the raspberry pi microprocessor sends this information to the cloud PUBNUB through WIFI module. Now the value of temperature and humidity will be displayed on the user webpage. If temperature is higher than threshold value user can start the motor which is on farm just by clicking start button from his device (Mobile or computer). Similarly as temperature become low he can stop the motor. User Interface is shown in figure 7.Similarly we can manage soil moisture level by using soil moisture sensor.

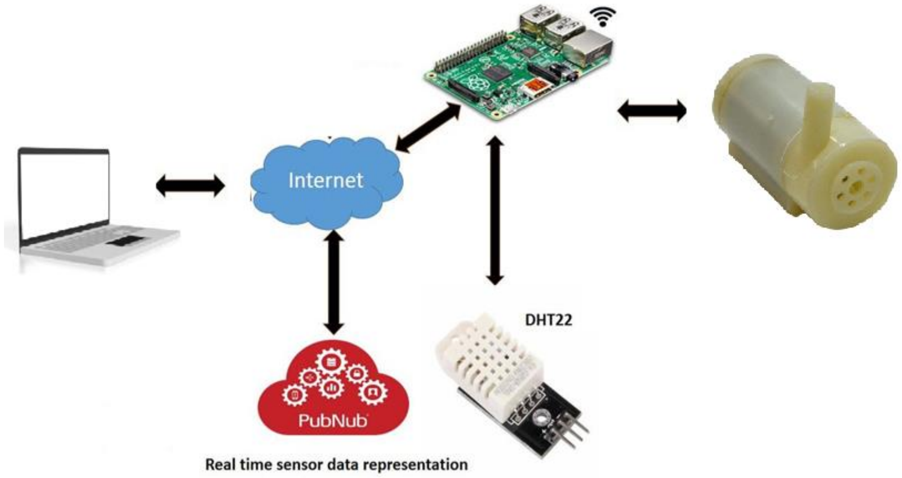

Fig. 4- Transmitter and Receiver section

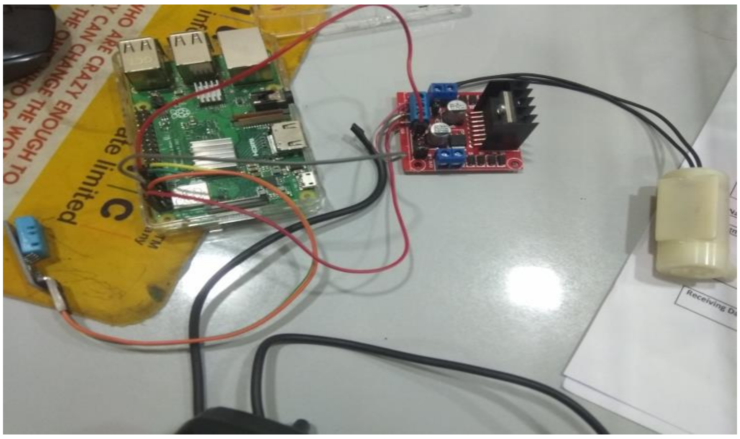

Fig. 5- Hardware Connection

DOI 10.17148/IJIREEICE.2021.9525

## IV.RESULT

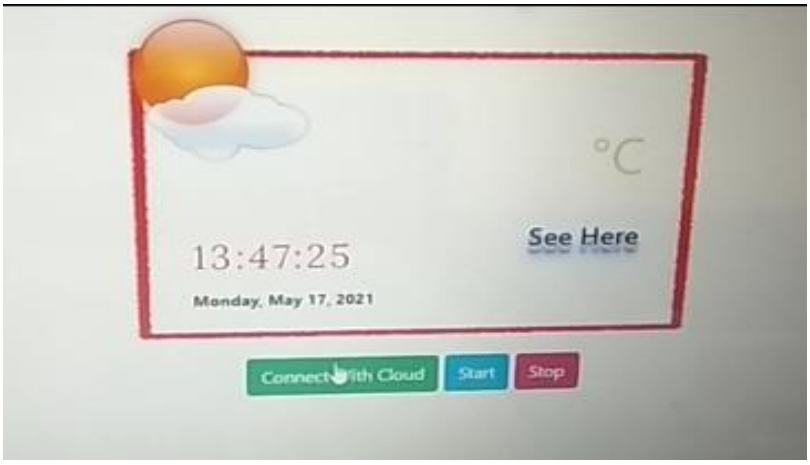

Fig 6. User Interface ( when not connected with cloud)

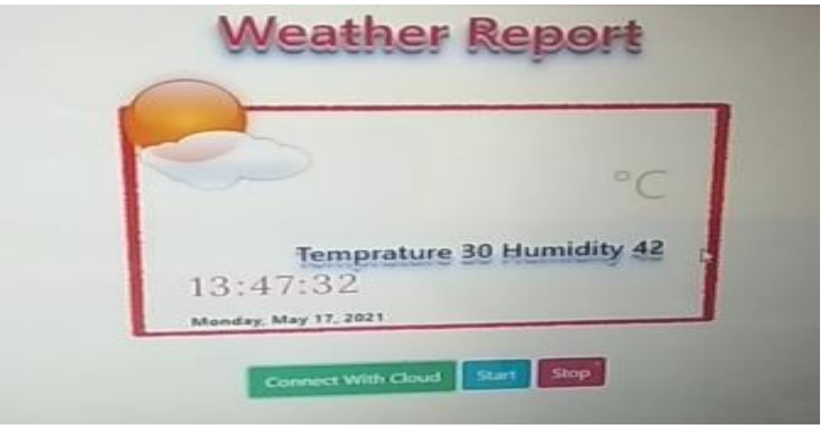

Fig 7. User Interface (when connected with cloud)

## V. JQUERY CODE

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>PubNub Example</title>
</head>
<body>
   {# The Button #}
   <button id="button_id_on">Press Me ON!</button>
   <button id="button_id_off">Press Me OFF!</button>
   {# PubNub JavaScript SDK #}
   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
   <script src="https://cdn.pubnub.com/sdk/javascript/pubnub.4.8.0.js">
   </script>

   {# Press button to publish #}
   <script>
      $(document).ready(function(){
         $("#button_id_on").on("click", function()
        var channel = 'numericrp';
        var pubnub= new PubNub({
          subscribeKey:'sub-c-99a0ff36-0419-11e9-ba8a-aef4d14eb57e',
          publishKey: 'pub-c-39811f57-53a9-4f4d-8aaa-83631e8f34c9',

        });

          alert("Test: Send message to pi");//Testing only
          pubnub.publish({
            channel: channel,
            message: 'on'
          });
      });

      $("#button_id_off").on("click", function(){

                              var channel = 'numericrp';

                              var pubnub= new PubNub({
                                 subscribeKey:'sub-c-99a0ff36-0419-11e9-ba8a-aef4d14eb57e',
          publishKey: 'pub-c-39811f57-53a9-4f4d-8aaa-83631e8f34c9',
                              });

                                alert("Test: Send message to pi");//Testing only
                                pubnub.publish({
                                  channel: channel,
                                  message: 'off'
                                });
      });
   });
  </script>
</body>
</html>
```

## VI. PYTHON CODE

```
import RPi.GPIO as GPIO
import dht11
import time
import datetime
from pubnub.pubnub import PubNub
from pubnub.pnconfiguration import PNConfiguration
from pubnub.callbacks import SubscribeCallback
from pubnub.enums import PNStatusCategory

# initialize GPIO
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.cleanup()
GPIO.setup(18, GPIO.OUT)

# read data using pin 14
instance = dht11.DHT11(pin=17)
pc=PNConfiguration()
#pc.subscribe_key="sub-c-c0ef660a-9ece-11e8-9a7c-62794ce13da1"
#pc.publish_key="pub-c-41a12f2d-6208-45aa-aa2b-bb08116d820c"
pc.subscribe_key="sub-c-99a0ff36-0419-11e9-ba8a-aef4d14eb57e"
pc.publish_key="pub-c-39811f57-53a9-4f4d-8aaa-83631e8f34c9"
pc.ssl=True
pubnub = PubNub(pc)

# Listen for Messages on the Market Order Channel
channel = 'numericrp'


class DatabaseSync(SubscribeCallback):
    def __init__(self):
        self.t=0
        self.h=0
    def message(self,pubnub,message):
        print(message.message)
        if(message.message=='Start'):
         GPIO.output(18,True)
        elif(message.message=='Stop'):
         GPIO.output(18,False)

        result = instance.read()
        if(result.temperature!=0):
         self.t=result.temperature
         self.h=result.humidity
        val="{},{}".format(self.t,self.h)
        pubnub.publish().channel(channel).message({'data':val}).pn_async(show)


def show(msg,stat):
    if(msg and stat):print(msg.timetoken,stat.status_code)
    else:print("Error",stat and stat.status_code)

print('listining...')
pubnub.add_listener(DatabaseSync())
pubnub.subscribe().channels(channel).execute()
```

## VII. CONCLUSION

This paper provides a good approach to save a water usage and make irrigation system better. Raspberry pi and Cloud (PUBNUB) is an important part of system which handle the processing and working. In this proposed system I have used only DHT sensor To make it more efficient we can use soil moisture sensor to indicate moisture level in the soil , buzzer as an alarm to notify user, water sensor to detect water level of the well, Flame sensor to indicate fire in the farm.

## ACKNOWLEDGMENT

I would like to give special thanks to Mr.Sandeep Sappal, Prof. Praveen Bansal, and Prof.Nipun Gupta for their support and guidance. They helped me in every step of my project, wherever I faced difficulty. I am greatly thankful.

## REFERENCES

[1] Prabhu, Boselin and Pradeep, M. and Gajendran, E., “An Analysis of Smart Irrigation System Using Wireless Sensor Network” Star Vol.5 Issue 3(3).
[2]Attar, S, & Sudhakar , K.N.Real Time monitoring of Agriculture Activities Using Wireless Sensor Network.
[3] S.Darshan, T.Sangavi, A.Soundharya ,Sheena Mohan, Sukanya Desikan “Smart Irrigation Syatem” IOSR Journal of Electronics and Communication Engineering(IOSR- JECE), Volume 10, Issue 3, Ver.II(May-June 2015).
[4]]Jainshkumar Aughan, ParveenSultana H, International Journal of Scientific & Engineering Research Volume 9, Issue 6, June-2018
[5] Chandan Kumar,pramitee behera”A low cost Smart Irrigation Control System”, International conference on Eectronics and Communication System(ICECS 2015) IEEE 1146
[6] https:// www.google.com