

# Smart Secured Real Time Agriculture Monitoring System

J. Srirampavan<sup>1\*</sup>

<sup>1</sup>Assistant Professor, Department Of ECE, QIS College Of Engineering And Technology, Ongole, India.

\*Corresponding Author E-Mail: [Srirampavan@Gmail.Com](mailto:Srirampavan@Gmail.Com)

## Abstract

Embedded systems in Agriculture play a vital role in unifying the work involved and improve conservations. Designing a smart as well as a cost efficient and more user-friendly system will be idealistic challenge. The following system that has been proposed is designed with those ideal constraints in mind. It consists of a Raspberry pi3 as a gateway that links the sensor networks with the cloud. To improve security an MQTT protocol is used for cloud connectivity. The communication between the sensor networks is managed by NRF24L01. The Sensor network is a separate entity that can used like a plug and play device and is built by a micro controller with a LCD display and an interfaced GPS. Multicasting is also possible between sensor networks and the gateway. The processed data from the sensor networks is sent through NRF24L01 to the gateway. The gateway further processes and encapsulates the data and through MQTT the data gets stored on the cloud. This cloud data can be accessed through computer or mobile device

**Keywords:** GPS, LCD, MQTT, NRF24L01.

## 1. Introduction

Agriculture, considering the foundation of Indian Economy, Since Without Agriculture Living is Impossible since Agriculture makers are the primary wellspring of sustenance for us we develop a data sharing system about knowledge for agriculture technology. Typically, in these making nation's uneducated agriculturists cases to utilize more water than required by manual techniques in this manner abusing them. With the target that it is basic to instruct the ranchers utilizing advancement soil sogginess sensors are for the most part required in those conditions to show that the agriculturists when it is depended upon to water and when not required the developing erraticism of implanted programming requires another, more able course of action approach. A trademark decision is to utilize settled part-based design; regardless, its arrangement to plan of implanted programming has been immediate and stacked with challenges.

The main aim of this research is to simplify the automation in agriculture field by using embedded system development. However, the approach for the proposed research is a very complex process and requires a deep integration of various technologies with in. We are integrating the sensor node with the gateway (Raspberry pi) by using wireless communication protocol to store on raspberry and send that data on cloud platform to analyze the sensor data for the future study or use that data for future investment. Storing data on cloud is always helpful for analyzing past data for future use. All the data or the current status of the sensors and the appliances would be securely transferred to the user or on the webpage that will be accessible to the user locally or globally.

The protocol used in this project is MQTT, its fast (compared to UDP) compared to other protocol, also it transfers the data securely i.e. without data loss & without any delay. Another major issue is associated with the hardware architecture as most of the embedded system software has to work with the targeted hardware correctly in order to maintain the functionality. There are various

technologies and collection methods of data are available and each has its own architecture which works with its own format. Thus, it is a highly challenging task to make the generated code work with the targeted hardware and related software or protocol. So, with the same approach the further research can be carried out to make this concept work with minority hardware options and come up with the one unique solution for the system.

## 2. System details

### Proposed System

Integrating the sensor node with the gateway (Raspberry pi) by using wireless communication protocol to store on raspberry and send that data on cloud platform to analyze the sensor data for the future study or use that data for future investment. Storing data on cloud is always helpful for analysing past data for future use .All the data or the current status of the sensors and the appliances would be securely transferred to the user or on the webpage that will be accessible to the user locally

The protocol used in this project is MQTT, its fast (compared to UDP) compared to other protocol, also it transfers the data securely i.e. without data loss & without any delay

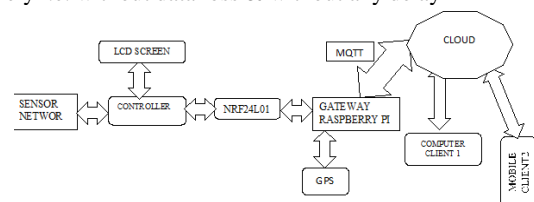
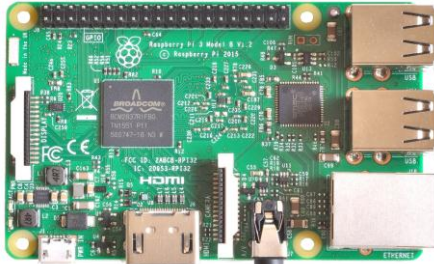


Figure 1: Block diagram

### Hardware Specifications

*Raspberrypi-3 model B*

Based on the most recent Broadcom 2837 ARMv864bit processor the Raspberry Pi 3 display B is quicker and more intense than its ancestors. It has enhanced power administration to help more effective outer USB gadgets and now accompanies worked in remote and Bluetooth network. To take full preferred standpoint of the enhanced power administration on the Raspberry Pi 3 and offer help for much more intense gadgets on the USB ports, a 2.5A connector is required. The below figure 1 shows raspberry pi 3 model B.



**Figure 2:** Raspberry pi 3 model B

*Arduino ATmega328P controller board*

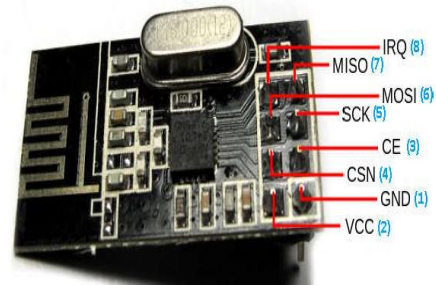
Arduino Uno as the focal control chip of the sensor focus. This chip depends upon ATmega328P (by Atmel) which is a 8-bit Advanced RISC Architecture with 32KB In System Self-Programmable Flash, 1KB EEPROM, 2KB SRAM. It has 14 automated data/yield pins (of which 6 can be utilized as PWM yields), 6 clear information sources, a 16 MHz quartz pearl, a USB alliance, a power jack, an ICSP header and a reset get. The Uno board can be filled by strategies for the USB alliance or with an outside power supply. The power source is picked in this way and can be modified with the Arduino Software (IDE). The underneath figure 2 indicates Arduino ATmega328P controller.



**Figure 3:** Arduino ATmega328P controller

*NRF24L01 transceiver module*

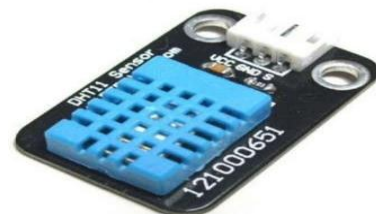
The module can utilize 125 unprecedented channels which gives a credibility to have a course of action of 125 self-governing working modems in a one place. Each channel can have up to 6 addresses, or every unit can talk with up to 6 units in a comparable time. The power utilization of this module is simply around 12mA amidst transmission, which is even lower than a lone LED. The working voltage of the module is from 1.9 to 3.6V, however the good thing is that diverse pins continues 5V method of reasoning, so we can without a doubt interface it to an Arduino without using any basis level converters. Three of these pins are for the SPI correspondence and they ought to be related with the SPI pins of the Arduino, however observe that each Arduino board have various SPI pins. The pins CSN and CE can have related with any propelled stick the Arduino board and they are utilized for setting the module in standby or dynamic mode, and moreover to switch between transmit or command mode.



**Figure 4:** NRF24L01 transceiver module

*Temperature and humidity sensor*

By using the select propelled signal securing method and temperature and dampness distinguishing development, it ensures high steadfastness and awesome whole deal soundness. This sensor fuses a resistive-type suddenness estimation part and a NTC temperature estimation section, and interfaces with a first class 8-bit microcontroller, offering brilliant quality, brisk response, against hindrance limit and cost-reasonability. The change coefficients are secured as ventures in the OTP memory, which are used by the sensor's internal banner recognizing process. Its little size, low power usage and up-to-20-meter hail transmission settling on it the best choice for various applications, including those most asking for ones.



**Figure 5:** DHT11 temperature and humidity sensor

*Soil moisture sensor*

This sensor can be used to test the sogginess of soil, when the earth is having water insufficiency, the module yield is at anomalous state, and else the yield is at low level. By using this sensor, one can thus water the bloom plant, or some different plants requiring modified watering system. Module triple yield mode, electronic yield is essential, basic yield more exact, serial yield with amend readings.

The Soil Moisture Sensor is used to evaluate the volumetric water substance of soil. Soil clamminess sensors measure the volumetric water content by suggestion by using some other property of the earth, for instance, electrical security, dielectric consistent, or correspondence with neutrons, as a mediator for the sogginess content. The Soil Moisture Sensor uses capacitance to evaluate dielectric permittivity of the including medium. In soil, dielectric permittivity is a segment of the water content. The sensor makes a voltage relating to the dielectric permittivity, and thusly the water substance of the soil.

The sensor midpoints the water content over the entire length of the sensor. There is a 2-cm zone of effect concerning the level surface of the sensor, yet it has by zero affectability at the exceptional edges. The figure above shows the electromagnetic field lines along a cross-section of the sensor, speaking to the 2-cm zone of effect.



**Figure 6:** Soil moisture sensor

### 3. Workdone

#### Interfacing Soil Moisture Sensor with Arduino

The equipment is containing an Arduino microcontroller (here an Arduino Uno R3) and a pre-wired soil sogginess sensor module. The soil sogginess sensor module, worked around the LM393 comparator, gives a dynamic low (L) level yield when the earth is dry (overseen by a pre-stetted bind respect). This mechanized yield (wet soil → L/dry soil → H) is guided to one I/O terminal (D2) of the Arduino microcontroller.

This soil moisture sensor has two tests through which current goes in soil, by then read the assurance of soil for examining moistness level. We understand that water make the earth more inclined to electric conductivity coming to fruition less assurance in soil where then again dry soil has poor electrical conductivity in this way more protection in soil. Utilizing these properties of vitality, the sensor is shaped. Inside the sensor there are gear for evaluating the affirmation and changing over it into voltage as yield.

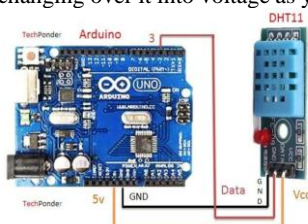


Figure 7: Interfacing soil moisture sensor with Arduino

#### Interfacing Temperature and Humidity Sensor with Arduino

The DHT11 recognizes water vapor by evaluating the electrical assurance between two anodes. The Humidity detecting part is a dampness holding substrate with anodes connected to the surface. Exactly when water vapor is consumed by the substrate, particles are released by the substrate which fabricates the conductivity between the cathodes. The adjustment in protection between the two anodes is with respect to stickiness. Higher relative moistness reduces the protection between the anodes, while letting down relative protection manufactures protection between the cathodes. The DHT11 measures temperature with a surface mounted NTC temperature sensor (thermistor) fused with the unit.

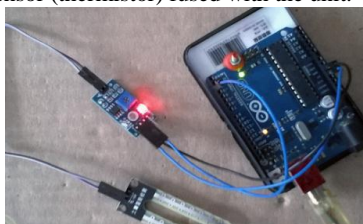


Figure 8: Interfacing temperature and humidity sensor with Arduino

#### Interfacing NRF24L01 Module with Arduino, Raspberry Pi

The NRF24L01 is a single chip, 2.4GHz transceiver with an embedded baseband protocol engine, which is used for ultra-low power wireless applications. This device is intended for operation in the worldwide ISM frequency band at 2.400 - 2.4835GHz. A MCU (microcontroller) and not a ton of outside inert parts are required to format a radio framework with the nRF24L01. The nRF24L01 is created and worked through a Serial Peripheral Interface (SPI.). The radio front end uses GFSK alteration. It has customer configurable parameters like rehash channel, yield power and air information rate. The air information rate fortified by the NRF24L01 is configurable to 2Mbps. The high air information rate

joined with two forces sparing modes makes the NRF24L01 especially sensible for ultra-low power plots.

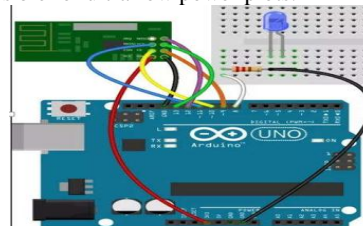


Figure 9: Interfacing NRF24L01 module with Arduino

Web accessibility in a key bit of various little scale controller wanders, particularly those using the atmega328p with Arduino ide/avr-gcc. NRF24L01 modules give a shoddy and strong remote association. In many undertakings, an additional avr/Arduino is joining to a PC with web relationship for partner the exercises to the web. Here I will pack my association with raspberry pi as online interface for various endeavors which utilizes nrf24L01 to remotely exchange information.

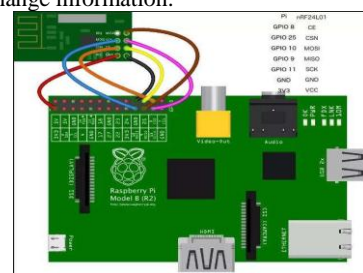


Figure 10: Interfacing NRF24L01 module with raspberry pi

### 4. Results

#### Soil Moisture Sensor with Arduino(Sender)

Interface DHT11 temperature and humidity sensor and soil moisture sensor with Arduino ATmega328p and sending sensing data values through NRF24L01 to raspberry pi

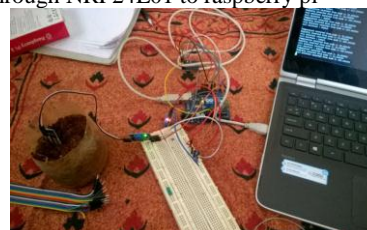


Figure 11: Interfacing of sensors and getting information to Arduino

#### NRF24L01 Module With Raspberry Pi (Receiver)

At receiver, sensor data collected by using NRF24L01 on Raspberry pi. Steps to install RF24 libraries at raspberry pi

```
wget
http://tmrh20.github.io/RF24Installer/RPi/install.sh.
chmod +x install.sh
./install.sh
```

After that by using make file and program (rf24receiver.cpp) file we have to perform following commands to receive sensing data from sensor node via RF24 network

```
Sudo make
Sudo ./rf24receiver
```

After above command execution you will see some a printout of some radio details, followed by a dot every 2 seconds while the Pi patiently waits for data until the Arduino start sending.



Figure 12: Receiver module at raspberry pi

### Cloud Connectivity

The basic C file to read sensor value, by using shell scripting we can read sensor data and send that data on cloud platform. For that it need to create .sh files to read, send, data to cloud

```
pi@raspberrypi:~$ cd bipin/gadgetkeeper/
pi@raspberrypi:~/bipin/gadgetkeeper$ ls
event_trigger.sh  read_soilmoisture.sh  set_value_h.sh
get_value.sh      read_temperature.sh   set_value.sh
MQTT              rf24libs              set_value.sh.save
mqtt_listner.py   send_humidity.sh      set_value_soil.sh
read_humidity.sh  send_humidity.sh.save TMP_FILE
read_sensor.sh    send_soilmoisture.sh tmp.txt
read_sensor.sh.save send_temperature.sh
pi@raspberrypi:~/bipin/gadgetkeeper$
```

Figure 13: Shell script files

Here in /home/pi/bipin/gadget keeper directory we created all read, send, set value .sh files to upload data on cloud Now we will see all .sh file one by one following To read sensor data, following commands are used \$ sudo nano read\_sensor.sh

```
pi@raspberrypi:~/bipin/gadgetkeeper
GNU nano 2.2.6 File: read_soilmoisture.sh
#!/bin/bash
#For DHT11 model
SCRIPT="/home/pi/bipin/gadgetkeeper/rf24libs/rf24receiver"
#For DHT22 model
#SCRIPT="/root/gadgetkeeper/Adafruit_DHT 22 4"
#For AM2302 model
#SCRIPT="/root/gadgetkeeper/Adafruit_DHT 2302 4"
SOILMOISTURE=$(SCRIPT | grep "soil" | awk -F " " '{print $3}')
echo "$SOILMOISTURE"
```

Figure 14: Sensor script

In read\_sensor.sh script we will provide RF network output path as a input to the script to read the all sensor data Set the execute permission with the following command.

\$ sudo chmod +x read\_sensor.sh

Finally we can run the command as follows and read the output.

\$ sudo ./read\_sensor.sh

To read only temperature value use the following script

\$ sudo nano read\_temperature.sh

To read only humidity and soil moisture sensor data.

Sudo ./ read\_humidity.sh

```
pi@raspberrypi:~/bipin/gadgetkeeper
GNU nano 2.2.6 File: read_sensor.sh
#!/bin/bash
#For DS18B20 model
SCRIPT="/home/pi/bipin/gadgetkeeper/rf24libs/rf24receiver"
#For DHT22 model
#SCRIPT="/root/gadgetkeeper/Adafruit_DHT 22 4"
#For AM2302 model
#SCRIPT="/root/gadgetkeeper/Adafruit_DHT 2302 4"
TEMPERATURE=$(SCRIPT)
#echo "$TEMPERATURE"
TEMPERATURE=$(SCRIPT)
echo "$TEMPERATURE"
```

Figure 15: Temperature reading script

Sudo ./read\_soilmoisture.sh

### Data Uploading Using API

To upload data on cloud using API, first we need create account on cloud platform. Here we are using Gadget keeper cloud platform

after creating account, we need to create new Things and in things we need to create different properties to store data. Also we need to provide our account API key, thing ID, Property ID on our script are as follows

### API Interfacing Script

Now we will be able to upload the temperature values to Gadget keeper using the following script

\$ sudo nano send\_temperature.sh

Here we are mentioning path for set\_value.sh , in that file we will provide details about cloud API key , Thing ID and Property ID to uploading data on cloud. For creating set\_value file following command has to type on command line

\$ sudo nano set\_value.sh`

after that command we need to write script in set\_value.sh and need to provide api key , thing id , property id, URL properly. By using “CURL” command we are able to upload sensor data on clouplatform.

\$ sudo ./send\_temperature.sh

### Reading Values from API

To read value from API we need to create another script get\_value.sh by using following command \$ sudo nano get\_value.sh

In that script also we need to provide api key, thing id , property id and url of your cloud platform . And by using “CURL” command we are able to read current value from API. For uploading we need to use “curl -PUT” command in script. For reading current value from api we need to use “curl -GET” command in script.

Now the script can be scheduled to upload data to API periodically (every2 min). Add the following cron job using sudo crontab –e command.

\*/2\*\*\*\*/home/pi/bipin/gadgetkeeper/send\_temperature.sh>/dev/null

By using cronjob we will able to update the sensor data on cloud continuously. For event create we need to create event script by using following command. Will set threshold value on trigger script as follows, so that we will get notification through email or SMS \$ sudo nano event\_trigger.sh

### MQTT Client Preparation Using MQTT Protocol

Create a directory using command sudo mkdir MQTTC d MQTT

By using shell scripting, we are able to upload data on cloud platform, for this please follow from point 5.3 to 5.3.2.1. Here we are using gadget keeper as cloud platform. By using API and MQTT protocol we are able to upload sensor data on gadget keeper. In this case we are using gadget keeper broker (api.gadgetkeeper.com) and we need to install mosquito client on raspberry pi to send data on cloud. Following picture shows uploaded sensor data on gadget keeper cloud platform

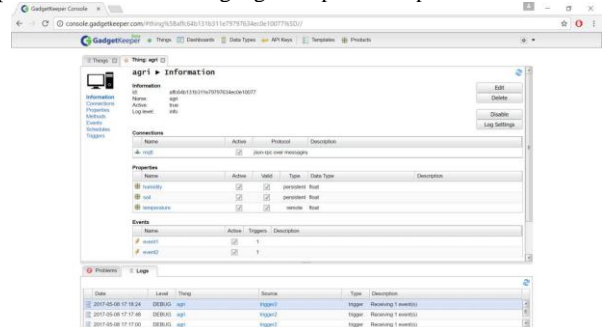


Figure 16: Creating account on cloud platform

## 5. Conclusion

The development of a system for smart agriculture can greatly benefit from the knowledge of the soil and water dynamics. Besides the soil moisture monitoring, to assess the plants water needs for its proper and healthy development, it was also important to assess the possibility of natural resources' usage optimization. Experimental tests on a real case scenario at a farm allowed the identification of some limitations, mainly related to energy consumption. Ongoing developments address the storage of historical information on the cloud concerning an agriculture field. The availability of a large amount of valuable data on the cloud enables the creation of intelligent services with very high potential, such as data correlation between different cultures and/or fields, plant disease estimation through the application of machine learning techniques, or determination of the most appropriate land cultivation according to soil conditions. In this thesis we are able to send data on cloud by using MQTT protocol as raspberry pi acts as a gateway, which would get data from sensor node. Also we will be able to add Transport Layer Security through MQTT protocol. In advancement, we can be able to add more security as video surveillance on farm and we will be able to provide ideal climate & water to the crop or fruits trees by analyzing previous stored data on cloud as per the respective crops and fruit trees

## References

- [1] Kaewmard N & Saiyod S, "Sensor Data Collection and Irrigation Control on Vegetable Crop Using Smart Phone and Wireless Sensor Networks for Smart Farm", 2014 IEEE Conference on Wireless Sensors (ICWiSE), (2014).
- [2] Gutiérrez J, Villa-Medina JF, Nieto-Garibay A & Porta-Gándara, MÁ, "Automated irrigation system using a wireless sensor network and GPRS module", IEEE transactions on instrumentation and measurement, Vol.63, No.1,(2014), pp.166-176.
- [3] Jayaraman PP, Palmer D, Zaslavsky A & Georgakopoulos D, "Do-it-Yourself Digital Agriculture applications with semantically enhanced IoT platform", IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), (2015), pp.1-6.
- [4] Wang Q, Terzis A & Szalay A, "A novel soil measuring wireless sensor network", IEEE Instrumentation and Measurement Technology Conference (I2MTC), (2010), pp.412-415.
- [5] Corke P, Wark T, Jurdak R, Wen H, Valencia P & Moore D, "Environmental wireless sensor networks", Proc. IEEE, Vol.98, no.11, (2010), pp.1903– 1917.
- [6] Mirabella O & Brischetto M, "A hybrid wired/wireless networking infrastructure for greenhouse management", IEEE Trans. Instrum. Meas., Vol.60, No.2, (2011), pp.398–407.
- [7] Sales N, Remédios O & Arseino A, "Wireless sensor and actuator system for smart irrigation on the cloud", IEEE 2nd World Forum on Internet of Things (WF-IoT), (2015), pp.693-698.