REMOTE PLANT WATERING AND MONITORING SYSTEM BASED ON IoT

Hemant Kuruva[1], Balumuri Sravani[2]
[1,2]Student, IV B.Tech., Department of ECE, G. Pullaiah College of Engineering and Technology, Kurnool, AP, India

Abstract: ***In the growing busy life of people, taking care of the houseplants has become a tough task. Professional life has taken such a toll on our lifestyle that taking care of houseplants is of least priority. This paper devises a new approach to solve this problem, which monitors the temperature, humidity, soil moisture and waters the plant remotely. Apart from all these functions, this paper also discusses an approach to visualize the data obtained by the sensors through pie charts and bar graphs.***
Index Terms: Arduino, AWS IoT, Raspberry Pi, Remote plant watering, Soil moisture sensor, Temperature and Humidity sensor, Water level sensor

I. INTRODUCTION

In today's life, it has become a tedious task to take care of plants at home and hence, the authors have come up with a system which can tackle this problem. Growing plants is a time consuming and effort takingduty as watering them is a daily task that is needed to be done to keep the plants alive.It is very usual that people often leave the plants to their own and this becomes worse when they go out of station. In such circumstances, they take help from a third person or they simply give up growing them.

Remote plant watering and monitoring system can help people with growing and monitoring it. As the system is remotely controllable, the owner can moderate the watering according to the atmospheric conditions existing at that point of time. Remote plant watering and monitoring system waters the plant, monitors the temperature and humidity of the surroundings, measures moisture of the soil and estimates the amount of water required for the plant and waters the plant remotely.

It further stores all data in Dynamo DB through AWS IoT. The data so collected is visualised using graphs and charts to give a better understanding of the plant's life and watering statistics to the user. All the sensors, actuators and water pump are controlled by Raspberry Pi running a nodejs application. For this requirement, the proposed system prefers to use Johnny-Five library. To communicate with Amazon Web Services, the authors used AWS IoT SDK for javascript.

II. BLOCK DIAGRAM

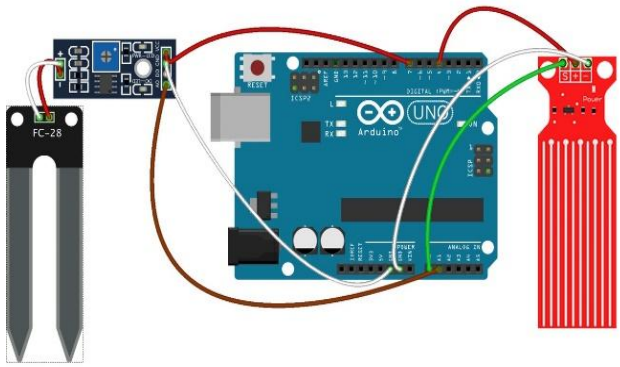

Fig. 1: Block diagram of Arduino

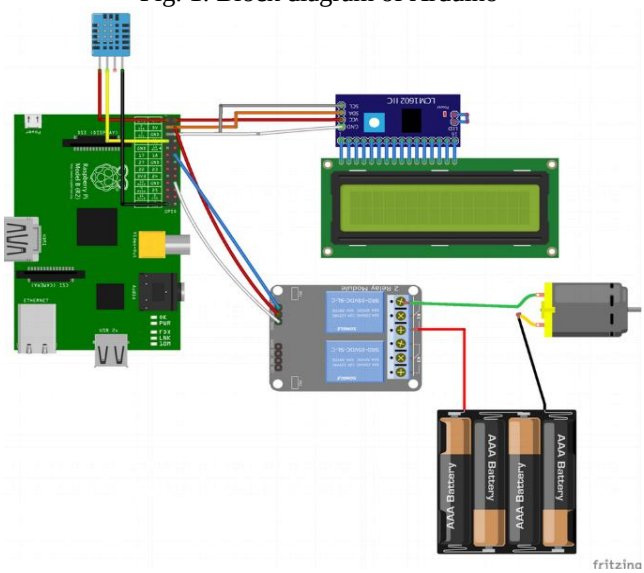

Fig. 2: Block diagram of Raspberry Pi

Hardware components:

1. Raspberry Pi 1 Model B
2. Arduino UNO
3. DHT11 Temperature & Humidity Sensor
4. YL-69 Soil Moisture Sensor
5. Water Level Sensor
6. Standard LCD 16x2
7. Water Pump Motor
8. 5V Relay
9. 4xAA battery holder

III. SOFTWARE IMPLEMENTATION

Initializing Dynamo DB

Dynamo DB stores the data collected by sensors.
Go to Dynamo DB website.
Create a new table with the following attributes:

- Table Name: remotewater_sensor_data
- Partition Key: key
- Click 'Add Sort key', add timestamp

Note: Data type of key and timestamp must be String.

Initializing AWS IoT

Open a web browser and go to AWS IoT Console Page.
Follow the steps below:

- Create a 'thing' with the name 'raspi-water-pump'
- Create a new policy with the name pump-policy, Action iot:*, resource *. Select Allow, click 'Add statement' and then click 'Create'.
- Create a certificate by using'one-click certificate create'. Download the public key, private key, and the certificate.
- Select the created certificate.
- Perform the following procedure:
    - Click 'Actions'> Activate button the certificate
    - Click 'Actions'> Attach a policy and name it as 'pump-policy'which was created and click 'Attach'.
    - Click 'Actions'> Attach a thing with the name'raspi-water-pump', Click 'Attach'.

Create a 'rule' with the following values:

- Name:storeInDynamoDB
- Attribute: *
- Topic Filter: sensor/data
- Choose an action: 'Insert message into a database table'
- Table name:remotewater_sensor_data
- Hash key value: ${topic()}
- Range key value: ${timestamp()}
- Role Name > Click 'Create a new role'.
- Click 'Allow' button from the next page that will open up and select it from Role Name dropdown

IV. HARDWARE IMPLEMENTATION

A. Initializing Arduino

The values received from Water level sensor, and soil moisture sensor are analog. Connectingthem directly to Raspberry Pi can be inefficient, hence we prefer to use Arduino to inputanalog values from these sensors and then send them to Raspberry Pi as digital values. Raspberry Pi updates the Arduino source code.
Connect the Arduino to PC and upload Standard Firmata.
Procedure to upload Standard Firmata:

- Open Arduino IDE on your PC
- Go to File > Examples >Firmata>StandardFirmata
- Upload sketch to Arduino (Ctrl+U)

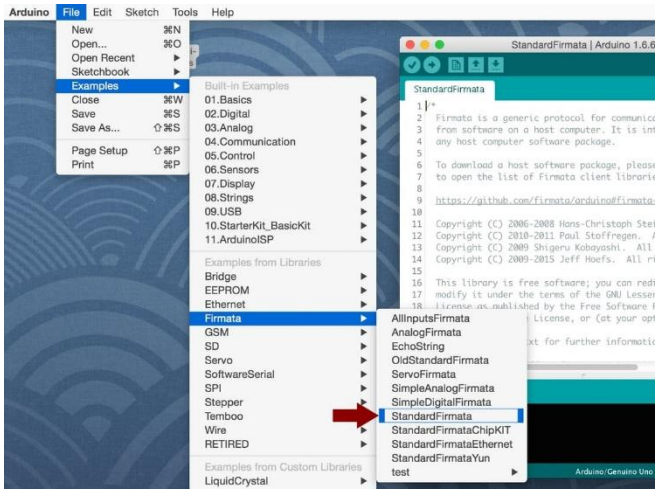

Fig. 3: Standard Firmata example on Arduino IDE

Connections between Soil Moisture and Arduino should be done in the following manner:

- PIN 7 ----> VCC (power)
- GND ---->GND (ground)
- AO ---->(analog output)

Connections between Arduino and Water level sensor should be done in the following manner:

- PIN 4 ---->(+)
- GND ---->(-)
- A0 ---->S

Next, use a USB cable to connect Arduino with Raspberry Pi.
Place thewater level sensor in the water tank and soil moisture sensor in theflowerpot.

B. Initializing Raspberry Pi

Connect Raspberry Pi withLCD screen, DHT11 temperature & humidity sensor and relay to switch water pump.

Connecting DHT11 temperature & humidity sensor to Raspberry Pi

- GPIO4 (PIN 7) ----> S
- 3V3 (PIN1) ---->(+)
- GND (PIN 25) ---->(-)

Connecting LCD Screen

- 5V (PIN2) ---->VCC
- GND (PIN6) ---->GND
- GPIO2 (PIN 3) ---->SDA
- GPIO3 (PIN 5) ---->SCL

Connecting Relay to switch Water pump

- GPIO18 (PIN 12) ---->S
- 5V (PIN4) ---->(+)
- GND (PIN 20) ---->(-)

Connecting the battery and relay to the water pump.

- Out of the two wires present on water pump, one must bewired to relay, the other with battery pack cable.

- The freebattery cable must be connected to the relay.

Refer to Figure 2.

V. RESULTS AND CONCLUSIONS

A. Output in DynamoDB table

We run the nodejs application on Raspberry Pi to collect sensor data and control the water pump remotely. Internet connection to the Raspberry Piis necessary.

Either Ethernet or WiFI would suffice the purpose.

Node should be installed on Raspberry Pi before following with the procedure further.

Connect theRaspberry Pi and run the following command:
git clone git@github.com:demirhanaydin/waterpi-node.git
cd waterpi-node
npm install

Go to AWS IoT page, download the public key, private key, and certificate files,and place them under certs folder. Download the root CA certificate file from root certificate. Save the file as rootCA.pem and place it under ‘certs’ folder.

Open device.js file with a text editor and update the following lines with the file paths. Get the IOT_DEVICE_URL from AWS IoT platform by clicking the thing that was created. It should be like https://XXX.iot.us-east-1.amazonaws.com where XXX is the IOT_DEVICE_URL.

Run the following command:
sudo node index.js

On the successful completion of the above procedure, sensor values will be printed on the LCD screen.

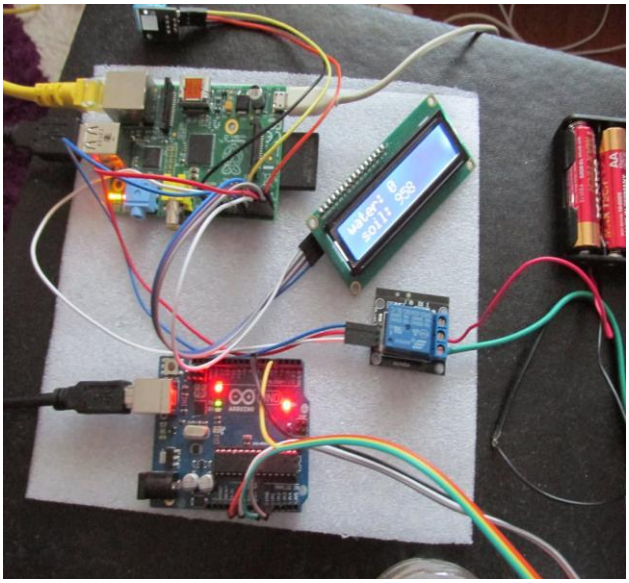

Fig. 4: Sensor values on LCDscreen

Fig. 5: Temperature and Humidity values on LCD

The values can further be seen on the DynamoDB table. The application sends sensor data to IoT platform every 20 seconds.

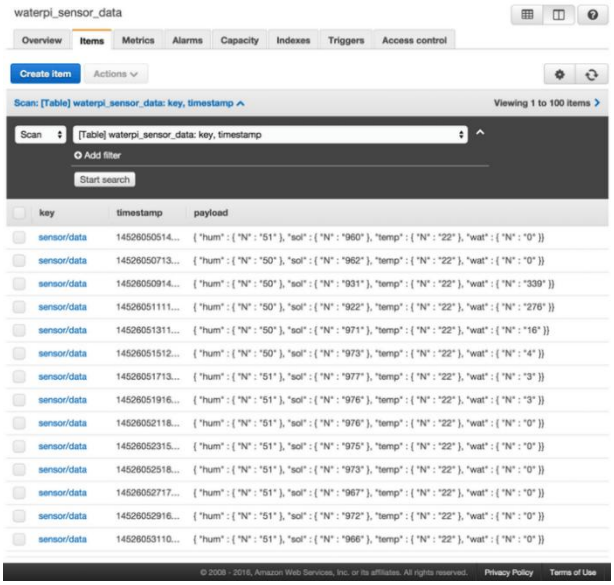

Fig.6: Sensor values on DynamoDB table

B. Output on web application

The web application visualizes the data present on Dynamodb table by using charts. The web application can also control the water pump remotely. Whenever the user clicks ‘Start’ or ‘Stop’ button, it registers an event to AWS IoT platform. Raspberry Piinputs this event and processes itstarts or stops the pump as per the user action.

Install Ruby on the computer to continue with the procedure further.

Open the Terminal and run the following commands:
git clone git@github.com:demirhanaydin/waterpi-web.git
cd waterpi-web
bundle install

Place the public key, private key, certificate, and rootCA files under ‘certs’ folder. Open boot.rb and update the lines

with adequate values.

Note down theaws access key and secret key from Security Credentials page on AWS Management Console.

Run the project with:
ruby app.rb

Open a web browser and go to http://localhost:4567/
Collected sensor data from DynamoDBwill be available in the form of charts.

Fig.7: Web interface of the 'Remote plant watering and monitoring system based on IoT'

In this paper, the authors have successfully demonstrated the practical implementation of Remote plant watering and monitoring system based on IoT.

REFERENCES

[1]. F. H. Moody, J. B. Wilkerson, W. E. Hart and N. D. Sewell, "A Digital Event Recorder for Mapping Field Operations," Applied Engineering in Agriculture, Vol. 20, No. 1, 2004, pp. 119-128.

[2]. K. A. Noordin, C. C. Onn and M. F. Ismail, "A Low-Cost Microcontroller-Based Weather Monitoring System," CMU Journal, Vol. 5, No. 1, 2006, pp. 33-39.

[3]. D. K. Fisher, "Automated Collection of Soil-Moisture Data with a Low-Cost Microcontroller Circuit," Applied Engineering in Agriculture, Vol. 23, No. 4, 2007, pp. 493-500. [Citation Time(s):1]

[4]. G. Vellidis, M. Tucker, C. Perry, C. Kvien and C. Bednarz, "A Real-Time Wireless Smart Sensor Array for Scheduling Irrigation," Computers and Electronics in Agriculture, Vol. 61, No. 1, 2008, pp. 44-50. doi:10.1016/j.compag.2007.05.009

[5]. D. K. Fisher and H. Kebede, "A Low-Cost Microcontroller-Based System to Monitor Crop Temperature and Water Status," Computers and Electronics in Agriculture, Vol. 74, No. 1, 2010, pp. 168-173. doi:10.1016/j.compag.2010.07.006 [Citation Time(s):1]

[6]. Arduino, "An Open-Source Electronics Prototyping Platform," 2012. http://www.arduino.cc [Citation Time(s):2]

[7]. D. Bri, H. Coll, M. Garcia and J. Lloret, "A Multisensor Proposal for Wireless Sensor Networks," 2nd International Conference on Sensor Technologies and Applications, Cap Esterel, 25-31 August 2008, pp. 270-275.

