2020 IEEE International Conference for Innovation in Technology (INOCON)
Bengaluru, India. Nov 6-8, 2020

Implementation of Smart Urban Farming using Raspberry Pi, Arduino and Node-RED Platform

Dhiraj Sunehra, SMIEEE
Department of ECE
JNTUH College of Engineering
Jagtial, India
dhirajsunehra@yahoo.co.in

M. Srinidhi
Department of ECE
JNTUH College of Engineering
Jagtial, India
sreenidhimanda.75@gmail.com

***Abstract*—Internet of things in recent times has been the driving power to the globe for its affordable and easily executable solutions. It is in this line a well-organized system has been developed for urban farming. Urban farming simply refers to producing food in urban spaces. City farming has gained a greater significance in the recent days, as accessing an appreciable amount of food for sustainability has been a grave concern for many families across the world. Also, with massive urbanization and population growth, the fertile land is declining day to day. Hence urban farming is likely to be the most feasible solution to avoid food scarcity. However, it has to be done in a way that it is more beneficial. The system propounded here aims at monitoring and maintaining the essential growth parameters like light, temperature, etc. prevailing to a plant whereby assuring a maximum yield.**

Keywords—Raspberry Pi3, Urban Farming, IoT, Node-RED platform, Arduino Uno, MQTT Protocol

I. INTRODUCTION

Urban farming has been considered prominent in many major cities. There are many motives that led to the development of urban agriculture. The most usual one is the rapid rise in urban population. This is because an increasing number of people migrate towards towns and cities casting around the employment added on with the greater availability of assets in metropolitan areas. As a result, there arose global food crisis, a situation where in the food produced was not enough to feed the entire population. Hence to avoid such turmoil circumstances in the days ahead, an economically fair and energy efficient solution has been developed. It employs Arduino Uno microcontroller that reads all the crucial growth parameters like light, soil moisture, etc. and sends the data to Raspberry Pi which then communicates with the server end using Message Queuing Telemetry Transport (MQTT) protocol thereby helping the user to pertain them as per the requirement.

II. LITERATURE REVIEW

Nilesh Patel and Pawan Kale proposed a smart agriculture monitoring system which monitors different agriculture parameters and transmits them wirelessly to both transmitter and receiver. It also works quite well in transmitting real values to a remotely placed receiver [1]. Tien Cao-Hoang and Can Nguyen Duy propounded an IoT system that is based on wireless sensor network. The system has many sensor nodes and a gateway which helps in monitoring the environmental data for agriculture using a webpage [2]. Rohini Shete and Sushma Agrawal developed a standalone system that provides a dynamic data sheet about the parameters of the city environment using a Raspberry Pi which communicates directly either with a LAN or Wi-Fi module [3].

III. HARDWARE DESCRIPTION

Figure 1 shows the block diagram of the smart urban farming system. It consists of various sensors such as Temperature and Humidity (DHT) sensor, Light Dependent Resistor (LDR) sensor, Soil Moisture sensor. These sensors are interfaced to the Arduino Uno board. The Raspberry Pi collects the sensor data from the Arduino and uploads the data on to the server for display on the webpage. The Node-RED platform is a server-side integrated development environment (IDE) that allows user to wire all the hardware devices on a GUI platform.

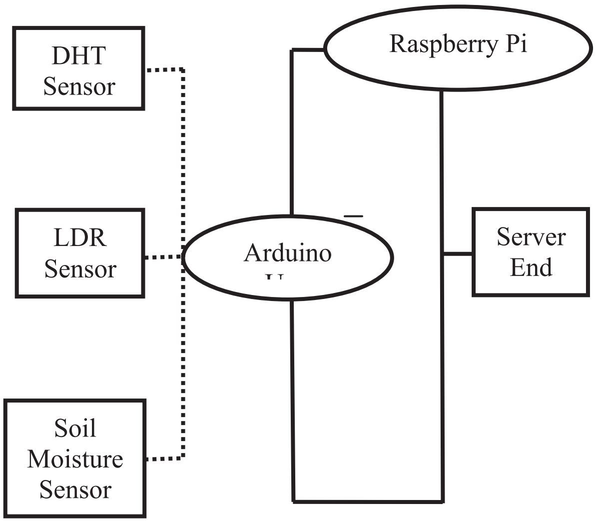

Fig. 1. Block Diagram of Smart Urban Farming System

A. *Salient Features of Raspberry Pi3 Model B+*

Figure 2 shows Raspberry Pi 3 B+ board. It is the third-generation single board computer. It is featured with Broadcom BCM2837B0 with Cortex-A53 System-on-chip (SoC). It is driven by a high quality 2.5A micro USB power supply. It was setup with 1.4 GHz processor with 1GB SDRAM. It has an on-board Wi-Fi 802.11.b/g/n/ac operating at dual band frequencies (2.5 and 5GHz) and Bluetooth [4]. It is supported with a three times faster Gigabit Ethernet. It comprises of 4 USB 2.0 ports, a full-size HDMI along with PoE facility.

Authorized licensed use limited to: Mukesh Patel School of Technology & Engineering. Downloaded on July 20,2023 at 04:17:51 UTC from IEEE Xplore. Restrictions apply.

There is no inbuilt storage in Raspberry Pi. The operating system for the Raspberry Pi is loaded into the SD card whose storage can range from 8GB to 64GB. It has 40 pins, out of which 26 are GPIO pins including UART, four PWM channels, I2C bus, SPI bus, 5V and GND. It has also has a Camera Serial Interface port to connect Pi camera through which we can capture HD videos and still images.

Fig. 2 Raspberry Pi3 Model B+ Board

B. Arduino Uno

Arduino Uno board consists of ATmega328P microcontroller. ATmega328P is an 8-bit advanced Reduced Instruction Set Computer (RISC) processor and is fabricated as a 28 pin PDIP (pin dual-inline configuration). It has 2KB of SRAM, 32 KB of flash memory, 6 analog pins which has a resolution of 10 bits, and 14 digital input / output pins [5]. Of the 14 digital pins, 6 pins exhibit PWM functionality. It operates on a 16MHz frequency resonator, has a USB connection, and power jack of 5V. It accepts an input voltage ranging from 7V-20V. However, it is powered up using a USB cable with a 9V supply. The Arduino Uno board It is depicted in Fig. 3.

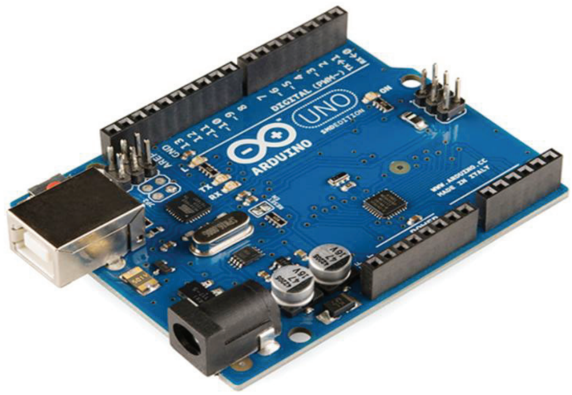

Fig. 3 Arduino Uno Board

C. Temperature and Humidity Sensor (DHT 11)

Figure 4 shows the DHT11 sensor. It is a resistive type humidity and temperature sensor module [6]. It utilizes exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. It makes use of a capacitive humidity sensor and a thermistor. These sensing elements are connected with 8-bit single-chip computer. And the temperature along with relative humidity values are determined with the variations in the resistance.

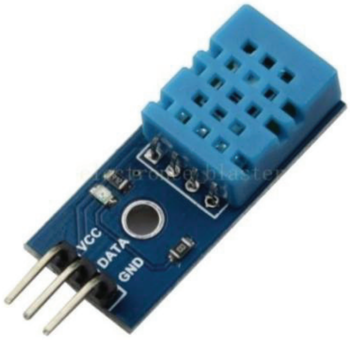

Fig. 4 DHT 11 Temperature and Humidity Sensor

D. Light Dependent Resistor (LDR)

An LDR sensor also called as a photoresistor. It is a passive device that exhibits photoconductivity, a phenomenon in which the resistance value decreases with the incoming luminosity on the sensitive surface of the component. This photo conductive cell creates an electron hole pair as light falls on it and thereby changes its resistance from mega ohms to ohms [7]. A simple LDR sensor is shown in the Fig. 5. Thus, decreased resistance contributes to increased illumination. At times even the wavelength of incident light may exert its impact on the component's sensitivity.

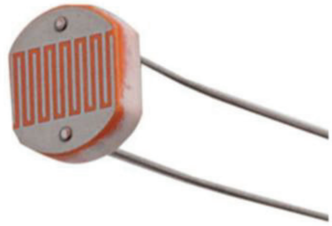

Fig. 5 LDR Sensor

E. Soil Moisture Sensor

Generally, the direct gravimetric measurement of moisture content in free soil turns out to be a complex process as it involves removing, drying and weighing the sample. Hence, soil moisture sensors are employed in a way that they estimate the volumetric water content of the soil by varying the parameters like resistance, dielectric constant etc [8]. Figure 6 depicts the soil moisture sensor.

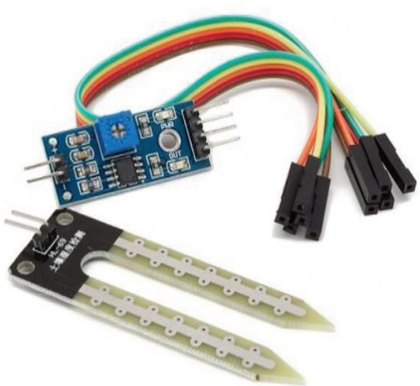

Fig. 6 Soil Moisture Sensor

Authorized licensed use limited to: Mukesh Patel School of Technology & Engineering. Downloaded on July 20,2023 at 04:17:51 UTC from IEEE Xplore. Restrictions apply.

IV. Software Tools

A. *Arduino IDE*

Arduino IDE is an open source software that makes an ease of writing and compiling the code into the Arduino module. This tool runs on various operating system platforms like Windows, Linux and Mac OS X, etc. The written code file saves with the extension of .ino. It runs on the java platform that has inbuilt functions and commands.

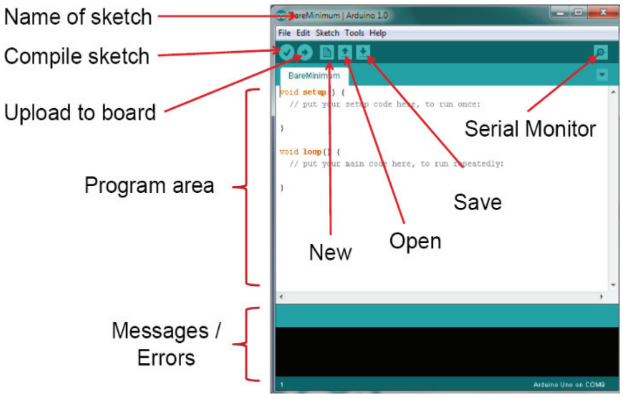

Fig. 7 Arduino IDE

Figure 7 represents the Arduino IDE environment which comprises of two sections namely the editor and compiler. The programs so called sketches are written in the editor section whereas the compiler helps in uploading them into the Arduino modules. The sketch written usually generates a hex file which is then transferred and uploaded. Both C and C++ can be employed in writing the code.

B. *MQTT on Raspberry Pi*

Message Queuing Telemetry Transport (MQTT) was designed by Andy Stanford-Clark (IBM) and Arlen Nipper in 1999. It is a light weight protocol that establishes machine to machine telemetry using publish/ subscribe paradigm and plays an integral role in IoT deployments. Publish/Subscribe is event-driven and enables messages to be pushed to clients. The central communication node is the MQTT broker, which dispatches the messages between the senders and the rightful receivers. For the MQTT protocol to be enacted, an open source message broker is required. And one such broker that best supports the Raspbian is Eclipse Mosquitto [9]. For the mosquitto to be set up on Raspberry Pi, the following commands are to be used:

- sudo apt-get install mosquito –y
- sudo apt-get install mosquito mosquito-clients –y

C. *Node-RED Platform*

Node-RED is a graphical programming language that runs with Node.js and lets the user to wire the hardware devices and online services. It is a server sided IDE and acts as a glue for IoT. It has become so eminent as it connects the hardware devices on a GUI. The nodes created in the IDE perform the specified task with the flows made thus eliminating the code at the front end. These nodes are generally the java script functions [10]. The editor window of Node-RED tool is shown in Fig. 8.

Fig. 8 Node-RED IDE

V. Results and Discussion

The schematic diagram, experimental setup, flowchart and results are discussed here.

A. *Schematic Diagram*

The schematic diagram of the hardware setup is demonstrated in the Fig. 9. It shows the interfacing of various components like GPS, GSM, USB camera, buzzer to the Raspberry Pi board. This is done using Fritzing application in which all the components are imported from Fritzing libraries. The 5V and GND of the Arduino are connected to the soil moisture sensor and DHT 11 sensor. A0 pin of Arduino is connected to the soil moisture sensor. Similarly, A1 pin of Arduino is connected to the LDR. Digital Pin 2 of Arduino is connected to DHT11.

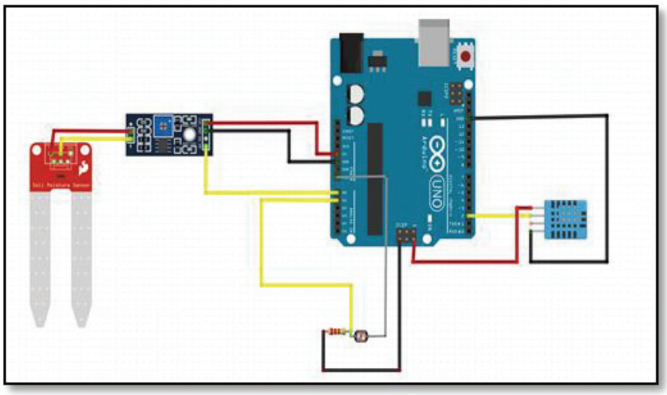

Fig. 9 Schematic Diagram of Smart Urban Farming

Authorized licensed use limited to: Mukesh Patel School of Technology & Engineering. Downloaded on July 20,2023 at 04:17:51 UTC from IEEE Xplore. Restrictions apply.

B. *Experimental Setup*

Initially all the components are powered by 5V-9V. The analog pin from Arduino board say A0 and A1 are connected to the analog pins of soil moisture sensor module and LDR respectively. The digital pin of DHT11 sensor (temperature and humidity sensor) is connected to the pin 2 of the controller. The Vin pins of all these sensors are connected to the 5V supply pins of the controller board. In order to link the Arduino board and Raspberry Pi, the serial port of Arduino board is connected to the USB port of Raspberry Pi 3 B+ model, which itself acts as a computer. All the connections made are represented in the Fig. 10.

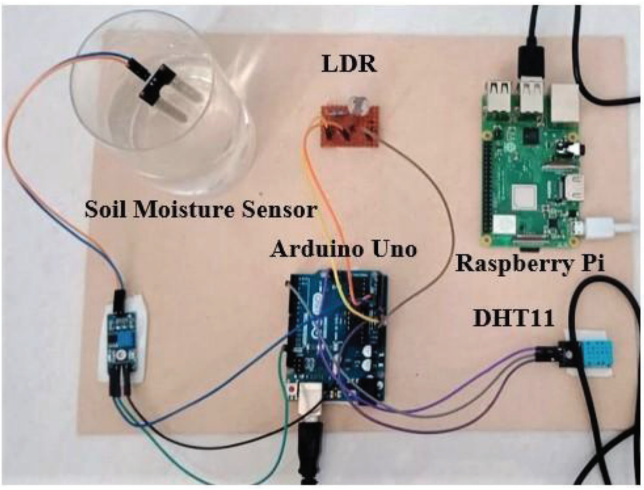

Fig. 10 Experimental Setup

The Node-Red flow created for the demonstration of the system's functionality is illustrated in the Fig. 11.

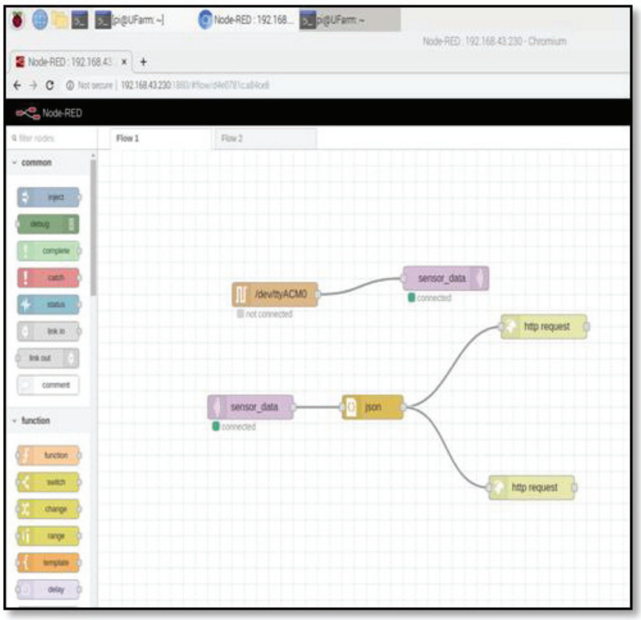

Fig. 11 Node-RED Flow

As Arduino is connected to the sensors, a serial node is used to represent the serial communication between the sensors and the microcontroller. Now Arduino board publishes the data and acts as a MQTT broker. Hence sensor data published by Arduino and subscribed by Raspberry Pi are represented using MQTT nodes. Since the data resembles the json format, json node is used. The data is then hosted on to a local and a global server which are represented using http request nodes in the above Fig.11.

C. *Flowchart of the System*

Message Queuing Telemetry Transport (MQTT) was

Figure 12 describes the flowchart of the smart urban farming system.

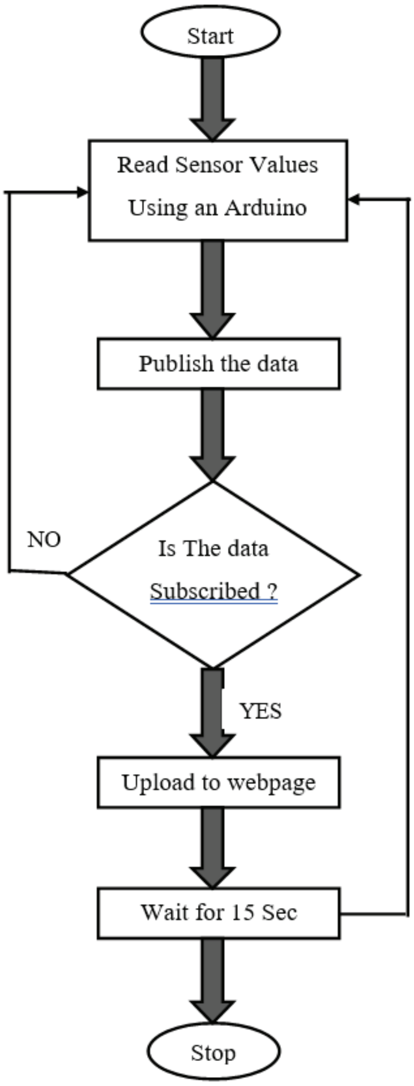

Fig. 12 Flowchart

D. *Results*

Figure 13 displays the sensor data uploaded on a local server. The web page projects the values of temperature, humidity, light and soil moisture at a particular instant including the date and time. The data is processed in such a way that the data is read by the page for every 15 seconds which can be varied. This local server is accessed using the IP address 192.168.43.230 using a remote desktop connection.

Figure 14 displays the results hosted on to a global server. In order to facilitate the user to access the data and monitor the parameters from anywhere across the globe, the data monitored is hosted on to a global server. The updation of results are made simultaneously into the local as well as global server. The global server in this paper is accessed using an IP address 3.17.74.57.

Authorized licensed use limited to: Mukesh Patel School of Technology & Engineering. Downloaded on July 20,2023 at 04:17:51 UTC from IEEE Xplore. Restrictions apply.

URBAN FARM MONITOR

192.168.43.230/Urban_Farm/Ufarm.php

Smart Urban Farming Using Raspberry PI, Arduino UNO And Node-RED

Temperature(° C)	Relative Humidity(%)	Soil Moisture(%)	Light Intensity(%)	Time & Date
29	46	35.97	1.96	08:27:30pm 12.08.2020
28	47	37.24	32.16	08:27:44pm 12.08.2020
26	48	37.05	34.02	08:27:59pm 12.08.2020
29	47	37.34	20.33	08:28:14pm 12.08.2020
28	49	37.05	36.27	08:28:30pm 12.08.2020
28	47	37.73	37.54	08:28:45pm 12.08.2020
25	59	60.51	47.31	09:58:08am 13.08.2020
25	60	67.55	47.12	09:58:22am 13.08.2020
24	61	51.71	48.48	09:58:37am 13.08.2020
24	61	51.71	49.07	09:58:52am 13.08.2020
25	60	52	49.36	09:59:07am 13.08.2020
24	61	52.49	49.95	09:59:22am 13.08.2020
24	61	52.79	51.22	09:59:37am 13.08.2020
25	60	52.79	50.64	09:59:53am 13.08.2020
25	60	52.98	50.93	10:00:08am 13.08.2020

Fig. 13 Data hosted on to a local server

URBAN FARM MONITOR

3.17.74.57/Office_Projects/Urban_Farm/Ufarm.php

Smart Urban Farming Using Raspberry PI, Arduino UNO And Node-RED

Temperature(° C)	Relative Humidity(%)	Soil Moisture(%)	Light Intensity(%)	Time & Date
29	46	35.97	1.96	08:27:31pm 12.08.2020
28	47	37.24	32.16	08:27:45pm 12.08.2020
26	48	37.05	34.02	08:28:00pm 12.08.2020
29	47	37.34	20.33	08:28:15pm 12.08.2020
28	49	37.05	36.27	08:28:30pm 12.08.2020
28	47	37.73	37.54	08:28:46pm 12.08.2020
25	59	60.51	47.31	09:58:08am 13.08.2020
25	60	67.55	47.12	09:58:23am 13.08.2020
24	61	51.71	48.48	09:58:38am 13.08.2020
24	61	51.71	49.07	09:58:54am 13.08.2020
25	60	52	49.36	09:59:09am 13.08.2020
24	61	52.49	49.95	09:59:24am 13.08.2020
24	61	52.79	51.22	09:59:38am 13.08.2020
25	60	52.79	50.64	09:59:53am 13.08.2020
25	60	52.98	50.93	10:00:08am 13.08.2020

Fig. 14 Data hosted on to a global server

Authorized licensed use limited to: Mukesh Patel School of Technology & Engineering. Downloaded on July 20,2023 at 04:17:51 UTC from IEEE Xplore. Restrictions apply.

VI. CONCLUSIONS

In this paper, a smart urban farming system is implemented using various sensors. The temperature, humidity, soil moisture and light illumination are measured and uploaded onto the webpage using the Raspberry Pi. This system therefore exemplifies the gravity for IoT networks. These networks have become so eminent as they enable the controllability of devices over the internet. The system thus developed profoundly provides an ample amount of food to all the people encircling a locality thereby turning out urban farming to be more constructive and viable.

REFERENCES

[1] Rohini Shete, Sushma Agrawal, "IoT based urban climate monitoring using Raspberry Pi," International Conference on Communication and Signal Processing (ICCSP), 6-8 April, 2016, Melmaruvathur, India.

[2] Tien Cao-Hoang, Can Nguyen Duy, "Environmental monitoring system for agriculture application based on wireless sensor network," Seventh International Conference on Information Science and Technology (ICIST), 16-19 April, 2017, Da Nang, Vietnam.

[3] Nilesh R Patel, Pawan D Kale, "Smart design of microcontroller based monitoring system for agriculture," International Conference on Circuit, Power and Computing Technologies (ICCPCT), 20-21 March, 2014, Nagercoil, India.

[4] https://www.raspberrypi.org/

[5] www.arduino.cc

[6] https://datasheetspdf.com/pdf/785590/D-Robotics/DHT11/1

[7] https://components101.com/ldr-datasheet

[8] https://www.smart-prototyping.com/Soil-Hygrometer-Detection-Module-Soil-Moisture-Sensor-For-Arduino.html

[9] Richard L. Halterman, "Learning to Program with Python", 2011 (www.cs.uky.edu/).

[10] https://nodered.org/

Authorized licensed use limited to: Mukesh Patel School of Technology & Engineering. Downloaded on July 20,2023 at 04:17:51 UTC from IEEE Xplore. Restrictions apply.