**Department of Electronic & Telecommunication Engineering**

**University of Moratuwa**

**Logo

Description automatically generated**

**EN2560 IoT Design and Competition**

**Project Report: IoT Empowered Garden Watering System**

**Group 05**

180195A – Y.S Ginige

180205H – H.K.R.L Gunasekara

180241M – D.R Hewavitharana

**Problem**

Most of the plats needed to be watered and maintained a proper moisture level and the sunlight in the soil to be grown up well. Due to busy schedules, people can’t spend considerable amount of time to treat their gardens. Because of that, home gardens are looking less attractive day by day and it eliminates another way to relax their minds.

**Conceptual Overview**

Open Weather map

Node RED

MQTT broker

Node MCU

Watering System

Mobile

Weather Details

Publish

Subscriber

Control

Alerts /Control

**Data Source**

* Open Weather Map API is used to get real time weather information of a given location.
* An NTP server is used to build a local clock.

**Methodology**

Node MCU

(Server + Micro controlling)

MQTT Server

Web Application

NTP Server

Motor Controller

DC Valve

**Node MCU Functionalities**

*Fig: Connectivity diagram between Node MCU and other blocks*.

1. **Web Application and Node MCU Connectivity**

Webserver

(Node MCU)

Webpage

(User)

HTTP request

HTTP reply

Webpage is designed using HTML, CSS and JavaScript.

User commands is sent to the Node MCU as HTTP requests.

* User sets up the location and send it to the Node MCU.
* User can set up the watering mode to either AUTO or MANUAL.

A Clock will be running in order to synchronize the sleeping schedule with the Node MCU. This will be further discussed later.

Webserver is implemented in the Node MCU.

According to the HTTP requests sent by the user, operations are carried out.

* Location set up data sent by the user is sent to the Node RED and receive the weather details.
* Received weather data sent to the webpage.
* According to the selected method, watering is done.

Location set data

sent via MQTT

**Node RED**

**Node MCU**

**Webpage**

*Location Set data*

*Location Set data*

*Weather Data*

*Weather Data*

Weather Data is used to adjust the watering.

**Location setup**

**Watering**

***AUTO*** ***MANUAL***

**Node MCU**

**Webpage**

*MANUAL enable*

*MANUAL enable confirm*

Manual Watering

*Manual Watering*

*Manual Watering Confirm*

**Node MCU**

**Webpage**

*AUTO enable*

*AUTO enable confirm*

AUTO watering happens as programmed.

**Webpage**



**To set up the location**

**Current Location Weather Details**

**Toggle between AUTO and MANUAL**

**When set to MANNUAL, this can be used to manually water the plants**

1. **MQTT Server and Node MCU Connectivity**

Location of the garden (City- Country) is sent to the Node Red through the MQTT server.

MQTT broker sends the published data from Node Red to the Node MCU. These data packets include the Temperature, Humidity, Weather type (Rain, Clouds, Sunny…), and the time zone of the garden.

1. **NTP Server and Node MCU**

NTP server is used to get the UTC time and a local clock runs on the Node MCU. Local time is calculated using the time zone information (how much time needs to be added to the UTC) sent by the Node-Red.

1. **Node MCU - Server**

Server runs on the Node MCU and it is accessible only inside the Local network because router doesn’t have a public IP. This server contains the web page information, and it is capable of handling Get, Push, …… Requests. In this application we only use Get request and their responses to maintain the system.

1. **Node MCU - Micro Controller**

At the same time, Node MCU act as the micro controller of the watering system. Using collected data, it decides water or not, mode of watering and when to water. Additionally, it will sleep occasionally and wake up and rebuild connections automatically. This mechanism helps to reduce the power consumption.

This system has two watering modes. Switching between these two modes can be done only when the Node MCU is awake.

**MANUAL MODE**

User has the full authority to control the watering. When he presses the watering button once, system waters for 5 minutes.

No sleeping mechanism implemented because user has the full control of the system.

**AUTO MODE**

Watering happens twice a day, between 9am-10am and 4pm – 5pm, for a calculated time period.

Watering will not happen if the weather type is Rain or Storm.

**Sleeping Schedule**

Node MCU goes to its light sleep mode at (2, 8, 14, …, 57) minutes in every hour. It sleeps for 3 minutes wake up in (0, 6, 12, …, 54) minutes in every hour. Briefly Sleep- Awake mechanism has a 6 minute time period. It awakes in first 3 minutes and sleeps in the second 3 minutes.

After watering happens, it automatically falls back to the sleeping schedule.

1. **Motor Controller**

When watering happens, Node MCU make its Valve (D4) pin High. This signal is given to a LM298n motor driver module and it switches on the 12V DC water valve.

Node MCU

Motor Controller

DC Valve

**D4 High / Low**

**ON / OFF**

A picture containing indoor

Description automatically generated

*Fig: Hardware setup of the system*

1. **Node-RED**

The Node-RED part of our system has three main stages, which are listed below.

Send weather data to the NodeMCU

Get the garden location

* Using the MQTT Out node, we send the weather details of the given location to the MQTT broker.
* NodeMCU then retrieves the weather details from the MQTT broker.
* To get weather information at the given location, we use the OpenWeatherMap node.
* <https://openweathermap.org/> Provides real-time weather details to the OpenWeatheMap node.
* Using MQTT In node, we get the garden location from the MQTT broker.
* We use <https://test.mosquitto.org/> as our MQTT broker.

Get the weather condition of the location

A picture containing graphical user interface

Description automatically generated

*Fig: Node-Red flow*

**Node-RED to NodeMCU communication protocol stack**

|  |  |
| --- | --- |
| Application Layer | MQTT |
| Transport Layer | TCP |
| Network Layer | IPv4 / IPv6 |
| Physical Layer | Wi-Fi : IEEE 802.11 |

**NTP Server to NodeMCU communication protocol stack**

|  |  |
| --- | --- |
| Application Layer | NTP |
| Transport Layer | UDP |
| Network Layer | IPv4 /IPv6 |
| Physical Layer | Wi-Fi : IEEE 802.11 |

**Webpage to NodeMCU(Web Server) communication protocol stack**

|  |  |
| --- | --- |
| Application Layer | HTTP |
| Transport Layer | TCP |
| Network Layer | IPv4 /IPv6 |
| Physical Layer | Wi-Fi : IEEE 802.11 |