FoodComputer

Techincal Documentation V1

07.11.2022

Nauman Shakir
https://NaumanShakir.com
https://3STechLabs.com

Overview

This technical report aims at defining an architecture for FoodComputer systems. The goals of this project are mentioned below.

Goals

1. Hardware should have internet-based connectivity.
2. Hardware should use a generic communication protocol.
3. Generic WebApp-baseddashboard.
4. Each device should have a unique ID.

Specifications

The system is divided into 3 different layers, which should communicate with each other in real-time.

Layers

- Hardware - Sensor Nodes
- Processing Layer - Server {Ubuntu Server 20.04}
- Application Layer - Smartphone App and WebApp

Components Required(Prototype){Standalone Sensor Nodes}

Using off-the-shelf components to test the first version of the device.

Single Board Computer

- Raspbery Pi 3B+

Sensors

- Si7021 Temperature and Humidity Sensor

Power

- AC to DC Converter
    - 12VDC Generic Adapter

Relay

- 4 Channel Relay Module

Misc

- 2x 12VDC Fans
- 1x AC Light Bulb

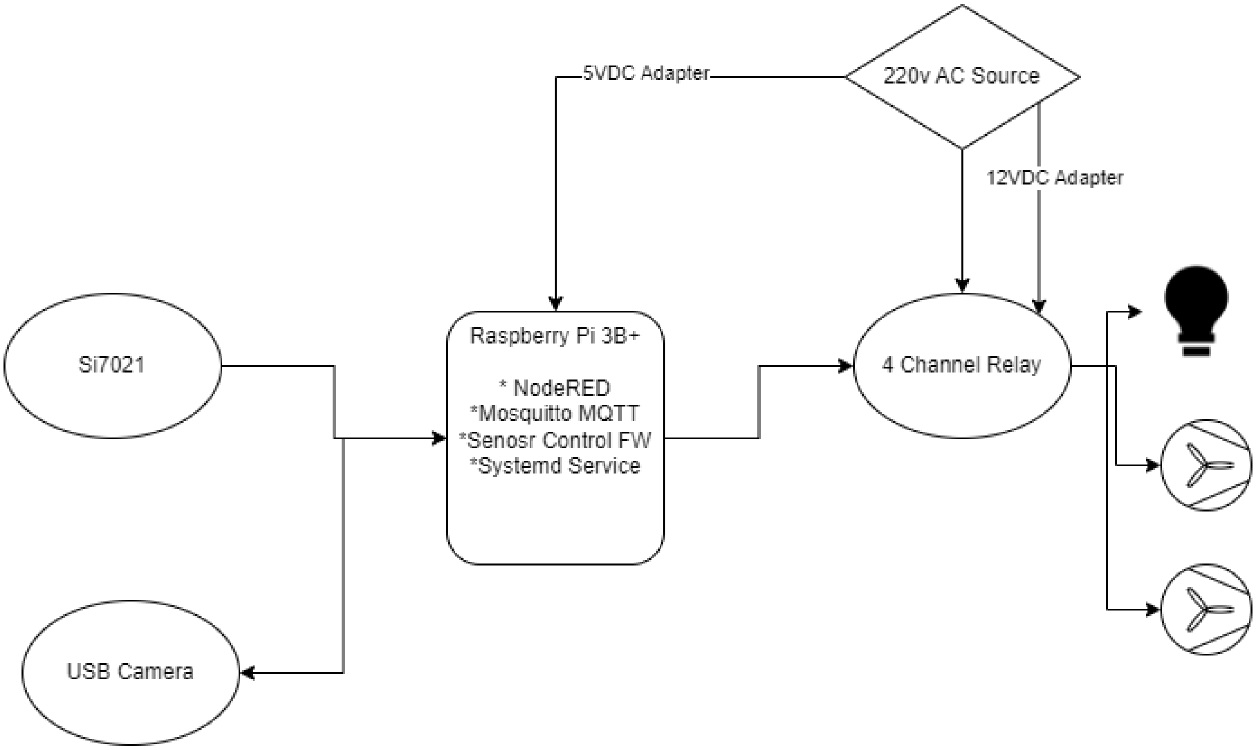
Architecture

The complete project has multiple components

- Sensor Nodes
- IoT Server

Sensor Nodes is a completely enclosed FoodComputer device and the IoT Server is hosted on the Sensor Node itself essentially making it an Edge IoT Sensor Node.

Sensor Nodes Architecture



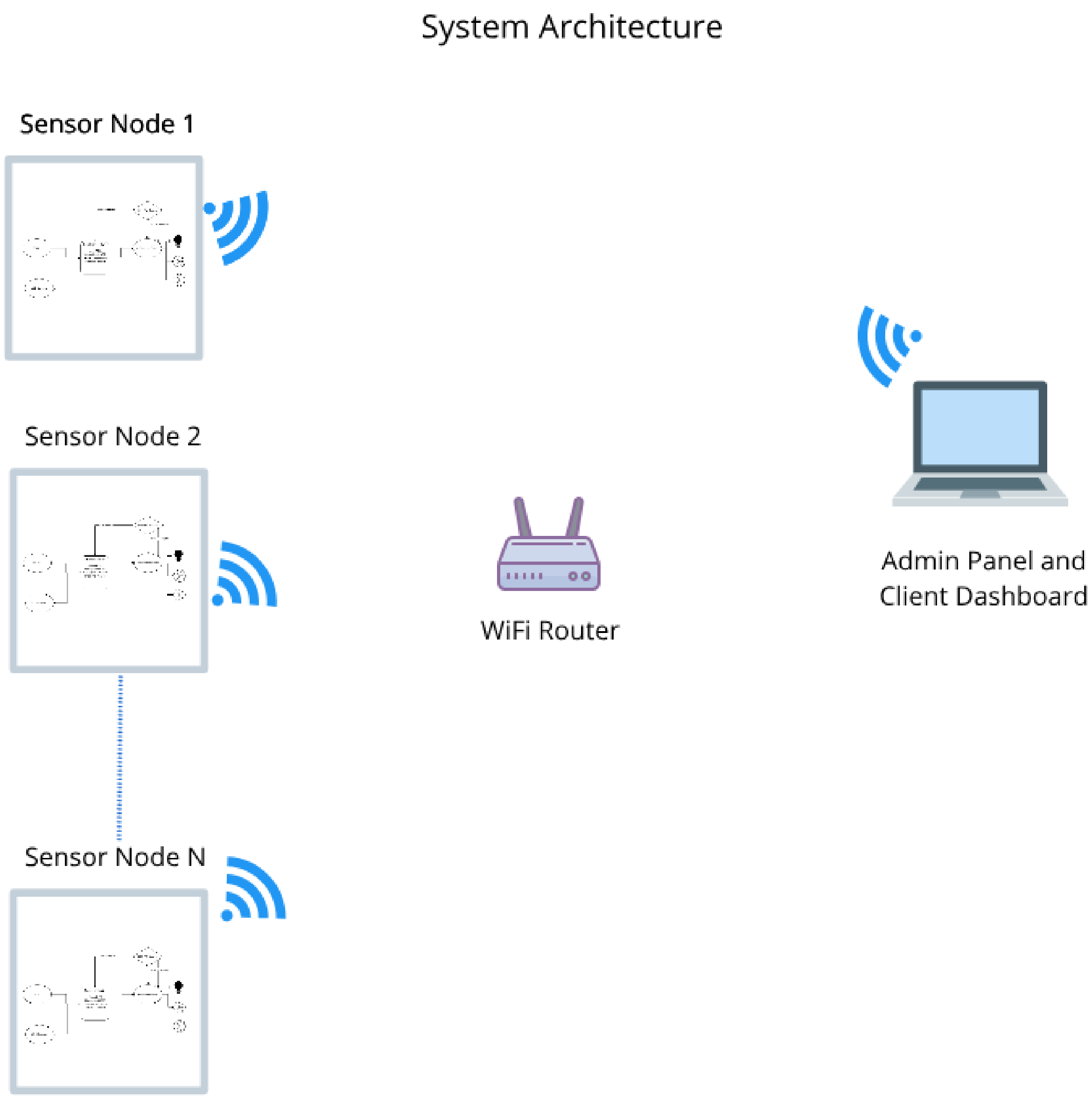
Above is the Sensor Nodes architecture. The sensor node is a standalone FoodComputer node. It contains Raspberry Pi Board with a sensors block, a USB Camera, power management block and a relay block. There could be N number of sensor nodes in the system.

Raspberry Pi is the SBC(Single-board-computer) of Choice in this sensor node because:

- It has WiFi, Ethernet and BLE built-in.
- Has a multi-core processor and can handle data acquisition, data storage, configuration and communication with the server in parallel.
- Have plenty of GPIOs and can be used to add extra sensors in the future.

For the commercial product, the Raspberry Pi could be replaced with an Raspberry Pi CM4 Module that is smaller in form factor. A single PCB could be designed to connect all sensors and relays on one board.

System Architecture



Above shown is the complete system architecture. There could be multiple sensor nodes in the system. A web-app-based dashboard hosted on the sensor node itself allow admins to see the camera feed, sensor values, control the relays and set the different parameters.

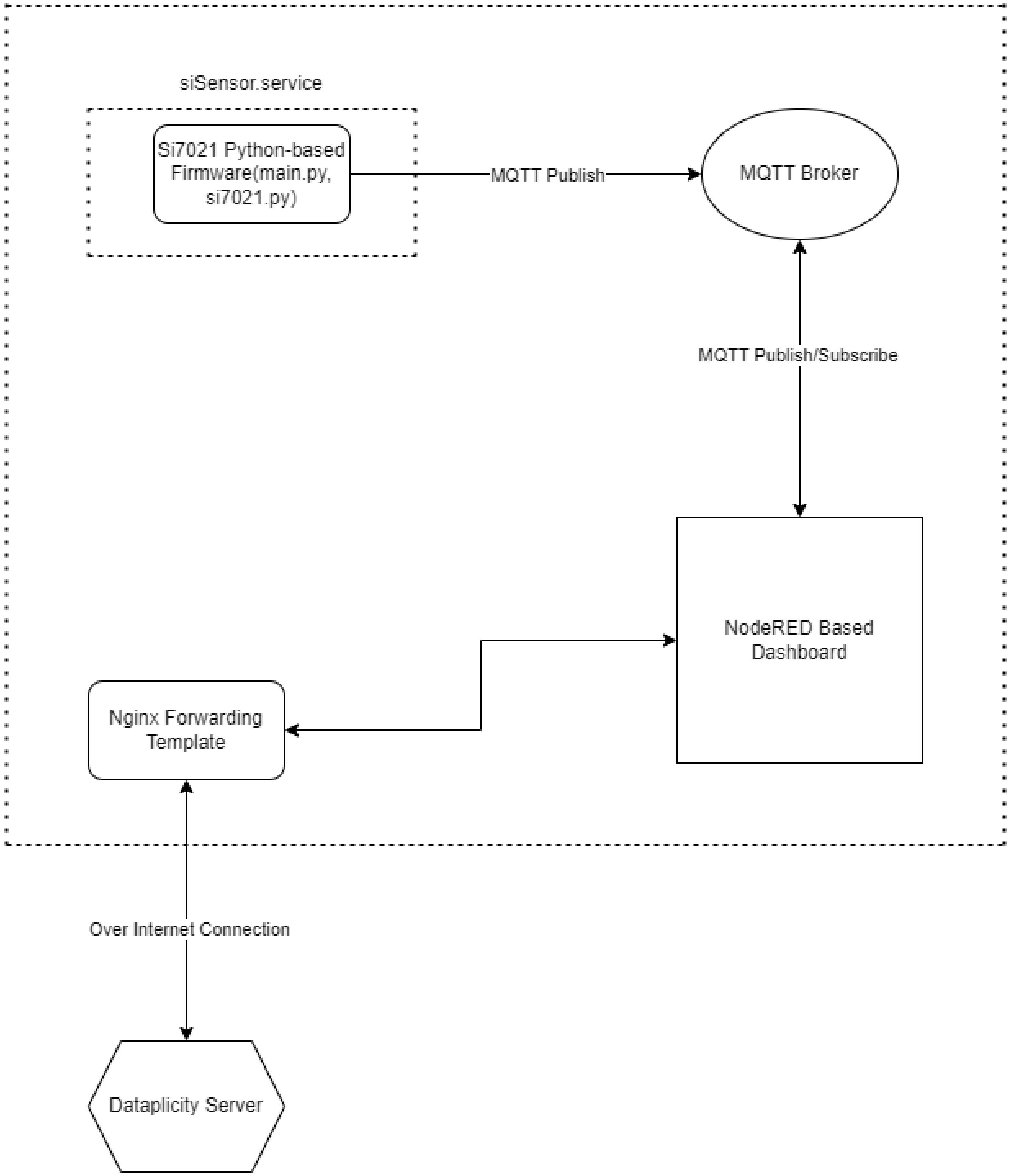
Software Architecture

Multiple Nodes management

The nodes have unique identifiers, the MAC address numbers of Raspberry Pi, which will be used to identify each Node uniquely(important for the MQTT Based Communication)

SensorNodeID;[DataString]

- SensorNodeID is a unique ID of Sensor Node
- [DataString] contains the sensor value(Status)



In short, Food Compter software has four basic components. The main component is the MQTT Broker which is installed on the Raspberry Pi itself and helps in allowing communication between different components. Then we have a python-based firmware to acquire the si7021 sensor data and publish it over the mqtt. The si7021 firmware is running as a systemd service and runs automatically on the boot. The third components is NodeRED dashboard that is susbcirbed to the si7021 data and displays it. Moreover, the dashboard also displays the camera pictures and allows the relay control.

To allow the dashboard access from outside the local network, an nginx template is implemented that forwards all the requests from dataplicty url to localhost:1880(NodeRED).

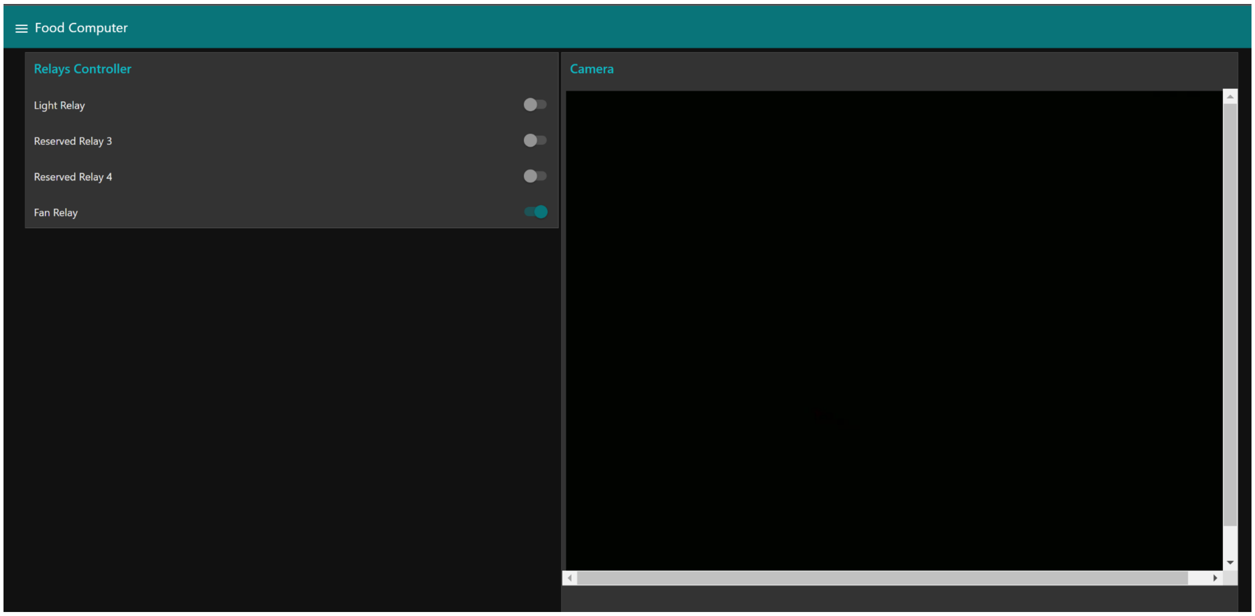
Every component in the software architecture is running as a systemd service and restart itself if something goes wrong.

Dataplicity Integration

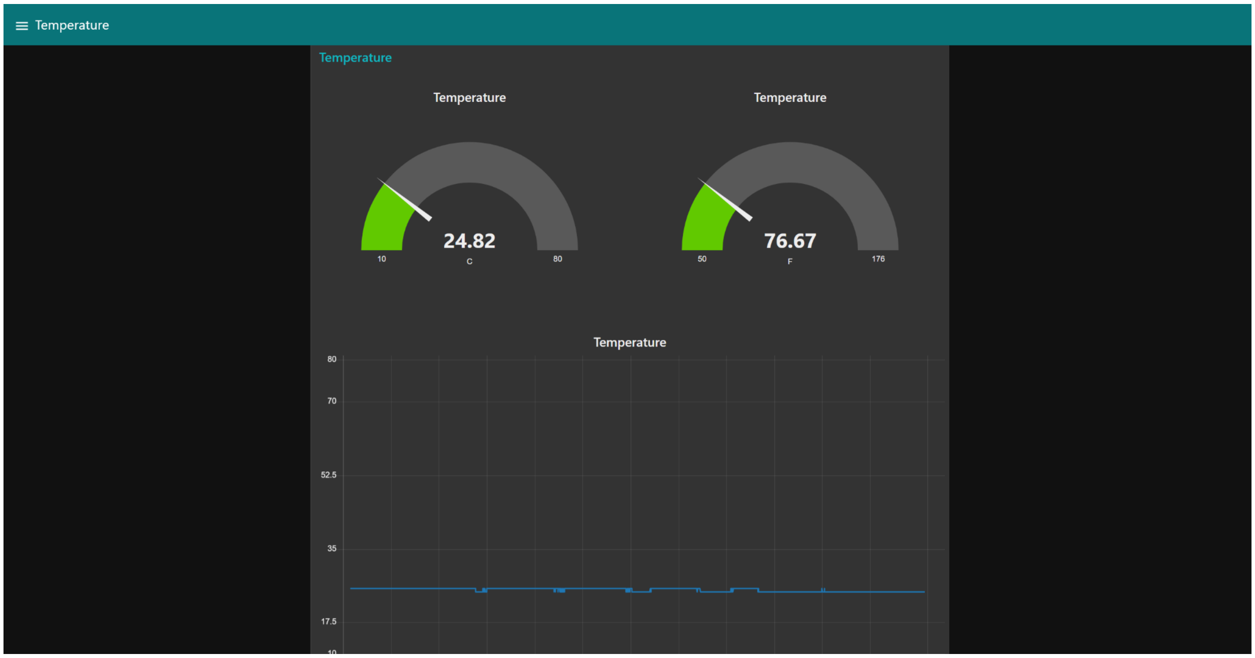
The Datapliciuty client program is running on the raspberry pi and allows remote terminal and the NodeRED dashboard access.

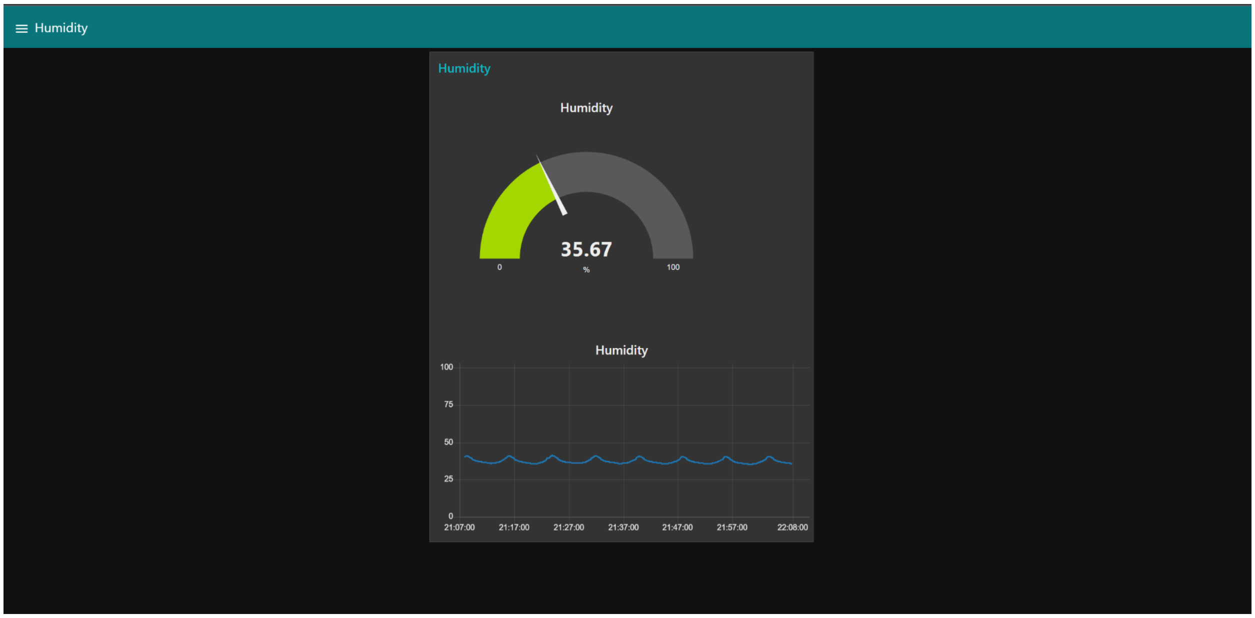
Dashboard

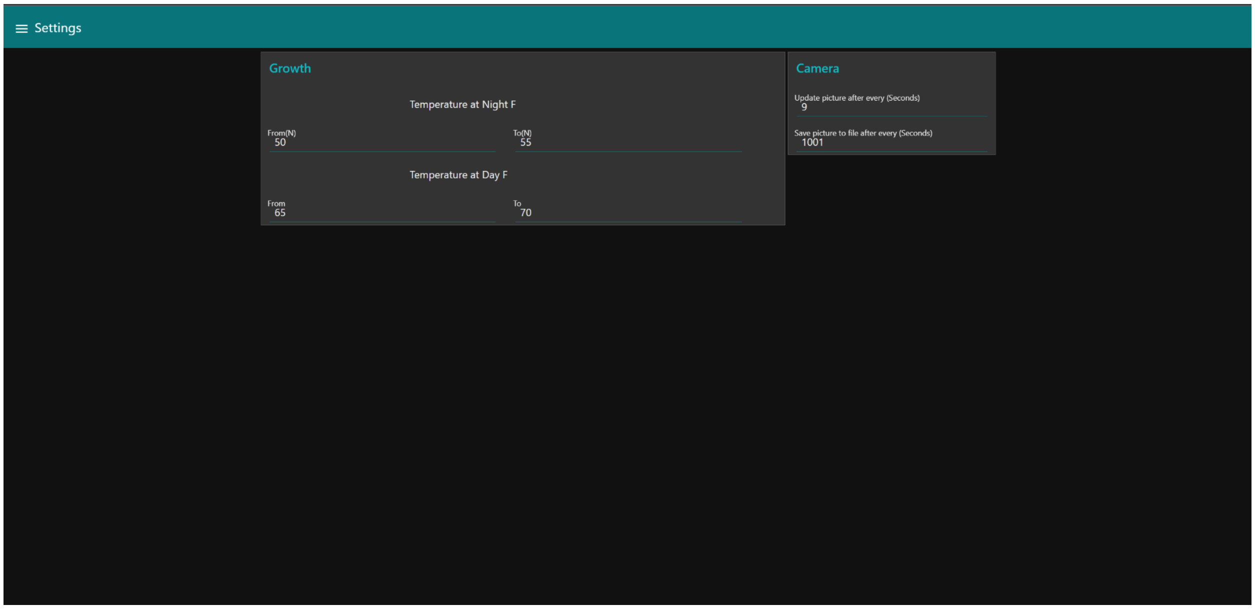
The dashboard UI is developed using NodeRED. The plugin used for UI design is *node-red-dashboard-ui* The screenshots are shown below.



Light Relay ○
Reserved Relay 3 ○
Reserved Relay 4 ○
Fan Relay ●







Growth

Temperature at Night F

From(N)
50

To(N)
55

Temperature at Day F

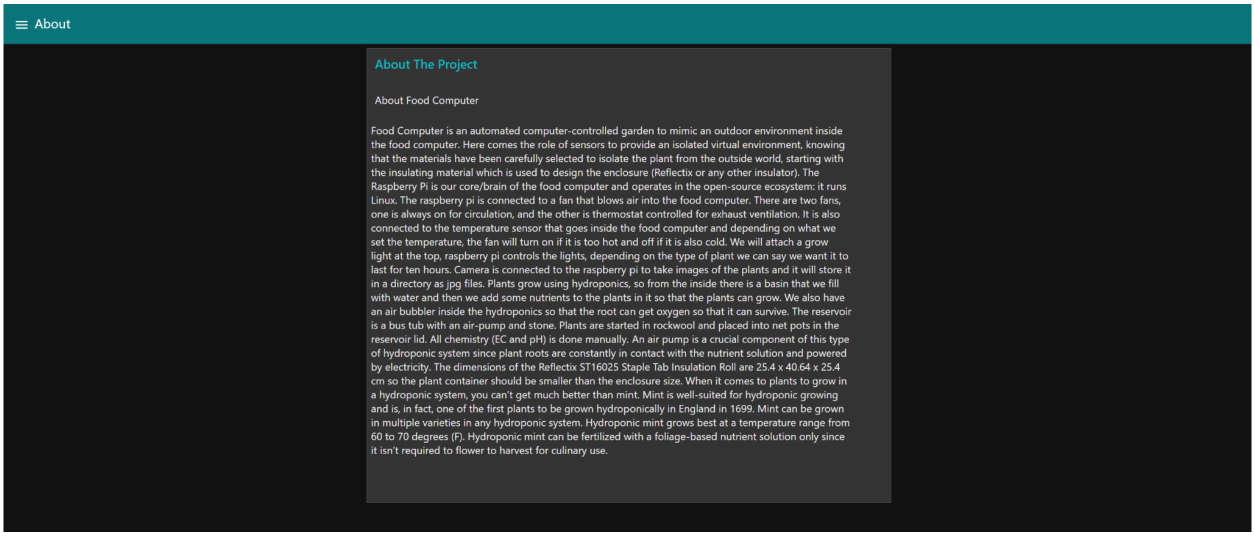
From
65

To
70

Camera

Update picture after every (Seconds)
9

Save picture to file after every (Seconds)
1001



About The Project

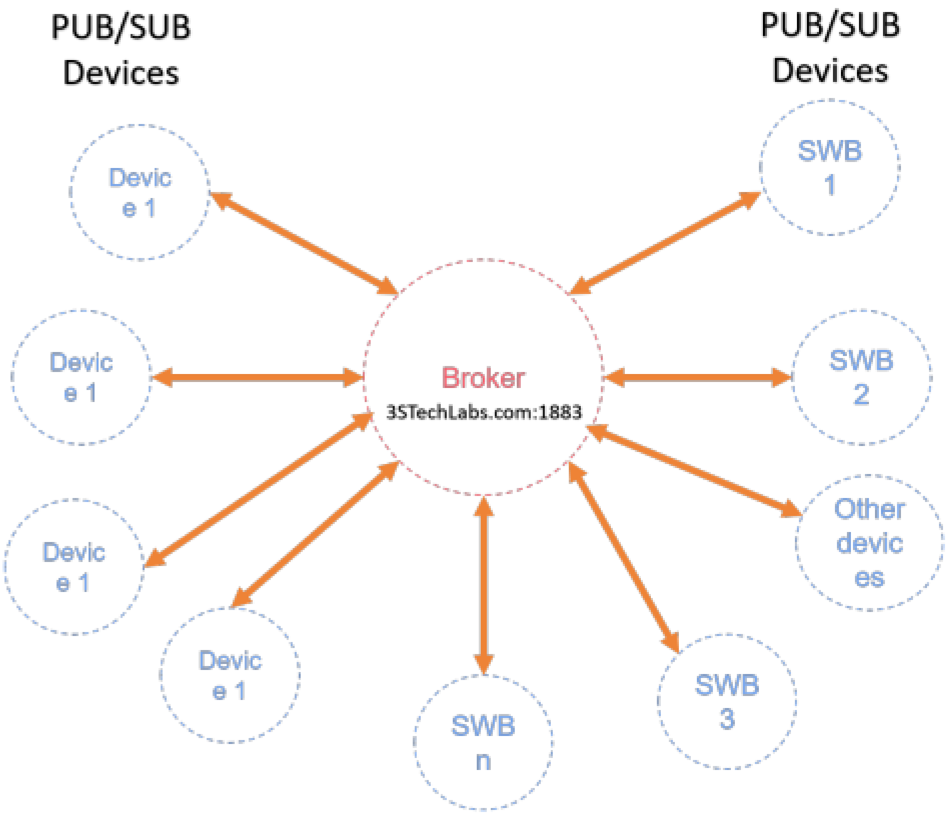
About Food Computer

Food Computer is an automated computer-controlled garden to mimic an outdoor environment inside the food computer. Here comes the role of sensors to provide an isolated virtual environment, knowing that the materials have been carefully selected to isolate the plant from the outside world, starting with the insulating material which is used to design the enclosure (Reflectix or any other insulator). The Raspberry Pi is our core/brain of the food computer and operates in the open-source ecosystem: it runs Linux. The raspberry pi is connected to a fan that blows air into the food computer. There are two fans, one is always on for circulation, and the other is thermostat controlled for exhaust ventilation. It is also connected to the temperature sensor that goes inside the food computer and depending on what we set the temperature, the fan will turn on if it is too hot and off if it is also cold. We will attach a grow light at the top, raspberry pi controls the lights, depending on the type of plant we can say we want it to last for ten hours. Camera is connected to the raspberry pi to take images of the plants and it will store it in a directory as jpg files. Plants grow using hydroponics, so from the inside there is a basin that we fill with water and then we add some nutrients to the plants in it so that the plants can grow. We also have an air bubbler inside the hydroponics so that the root can get oxygen so that it can survive. The reservoir is a bus tub with an air-pump and stone. Plants are started in rockwool and placed into net pots in the reservoir lid. All chemistry (EC and pH) is done manually. An air pump is a crucial component of this type of hydroponic system since plant roots are constantly in contact with the nutrient solution and powered by electricity. The dimensions of the Reflectix ST16025 Staple Tab Insulation Roll are 25.4 x 40.64 x 25.4 cm so the plant container should be smaller than the enclosure size. When it comes to plants to grow in a hydroponic system, you can't get much better than mint. Mint is well-suited for hydroponic growing and is, in fact, one of the first plants to be grown hydroponically in England in 1699. Mint can be grown in multiple varieties in any hydroponic system. Hydroponic mint grows best at a temperature range from 60 to 70 degrees (F). Hydroponic mint can be fertilized with a foliage-based nutrient solution only since it isn't required to flower to harvest for culinary use.

Features of the WebApp

1. Minimalistic design
2. Devices overview.
3. Controlling and managing the relays and other parameters.

What is MQTT?



Take an example of a system in which hundreds of people have smartwatches that can display information about a person's surroundings. And then, there are Android, iOS and Windows devices that can be used to monitor smartwatches to define a set of parameters for smartwatches.

So the best communication protocol is MQTT in a scenario with mixed types of devices, including hardware platforms.

It can handle two-way and parallel communication, and the number of devices connected and communicated via MQTT are limitless. The only limit is server resources. MQTT is also known as pub/sub protocol.

Hence the protocol of choice here is MQTT.

*The benefits of MQTT are its low footprint and fast communication.

Management Pipeline

As there’s no cloud server used, there is no need of CI/CD. The source code and documentation available as a github repository. https://github.com/shouqq/FoodComputer

The deployment and installation instructions can be found here: https://github.com/shouqq/FoodComputer/blob/main/README.md