Smart Buzzer System

Preliminary Report V1

04.28.2020

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

Overview

This preliminary report aims at defining an architecture for the Smart Buzzer System. The goals of this report are mentioned below

Goals

1. Wireless Connectivity.
2. High Availability, fail-safe product.

Specifications

The system is divided into 4 different layers which should communicate with each other in real time.

Layers

- Hardware - Microcontroller
- Processing Layer - Server {Raspberry Pi}
- Transport Layer - MQTT(communication protocol) and REST API
- Network Layer - WiFi

Components Required

The list below is a suggested list of components.

Components Required for Single Buzzer

- ESP8266 NodeMCU**($4.07)**
  - https://www.amazon.com/ESP8266-ESP-12E-Wireless-Internet-Development/dp/B07KNV59WB/ref=sr_1_6?dchild=1&keywords=esp8266+nodemcu&qid=1619590202&sr=8-6
- 3.7v 2000mAh Lipo Battery**($13.50)**
  - https://www.amazon.com/103450-2000mAh-Polymer-Battery-Rechargeable/dp/B06X3VW98X/ref=sr_1_2?dchild=1&keywords=3.7v+2000mah+lipo&qid=1619742445&sr=8-2
- 3.7v LiPo charger**($1.80)**
  - https://www.amazon.com/Lithium-Battery-Charging-Protection-Functions/dp/B07V2SKYJM/ref=sr_1_14?dchild=1&keywords=module%2BTP4056&qid=1619742535&sr=8-14&th=1
- 3.7v to 5v Boost Converter**($10)(pack of 5){qty 1=$2}**
  - https://www.amazon.com/Mini-DC-Voltage-Regulator-5V/dp/B01N6EKHZR/ref=sr_1_17?dchild=1&keywords=3.7v+lipo+charging+module&qid=1619590586&sr=8-17
- Buzzer**($12)(pack of 2){qty 1=$6}**
  - https://www.amazon.com/Uxcell-a15050500ux0137-Intermittent-Electronic-Sounder/dp/B010V4UR6Q/ref=sr_1_22?dchild=1&keywords=arduino+buzzer&qid=1619590955&sr=8-22
- Big Push Button**($11)**
  - https://www.amazon.com/Yiju-Shaped-Illuminated-Self-resetting-Button/dp/B092S2ZVHM/ref=sr_1_5?dchild=1&keywords=dome%2Bpush%2Bbutton&qid=1619591061&sr=8-5&th=1
- On/Off Switch**($5)(pack of 10){qty 1=$0.5}**
  - https://www.amazon.com/Besmelody-Rocker-Toggle-Switch-Position/dp/B076V2LFR5/ref=sr_1_17?dchild=1&keywords=two+pin+on+off+switch&qid=1619591401&sr=8-17

Approx Cost of Single Buzzer = $38.87

Components that are required only 1 in quantity for any number of buzzers

- Raspberry Pi 4 4GB Kit**($114)**
  - https://www.amazon.com/CanaKit-Raspberry-4GB-Starter-MAX/dp/B07XPHWPRB/ref=sr_1_3?dchild=1&keywords=raspberry+pi+4&qid=1619591880&sr=8-3

You will require one Raspberry Pi 4 GB kit which will cost you $114 and a single buzzer device will cost you $38.87

Architecture

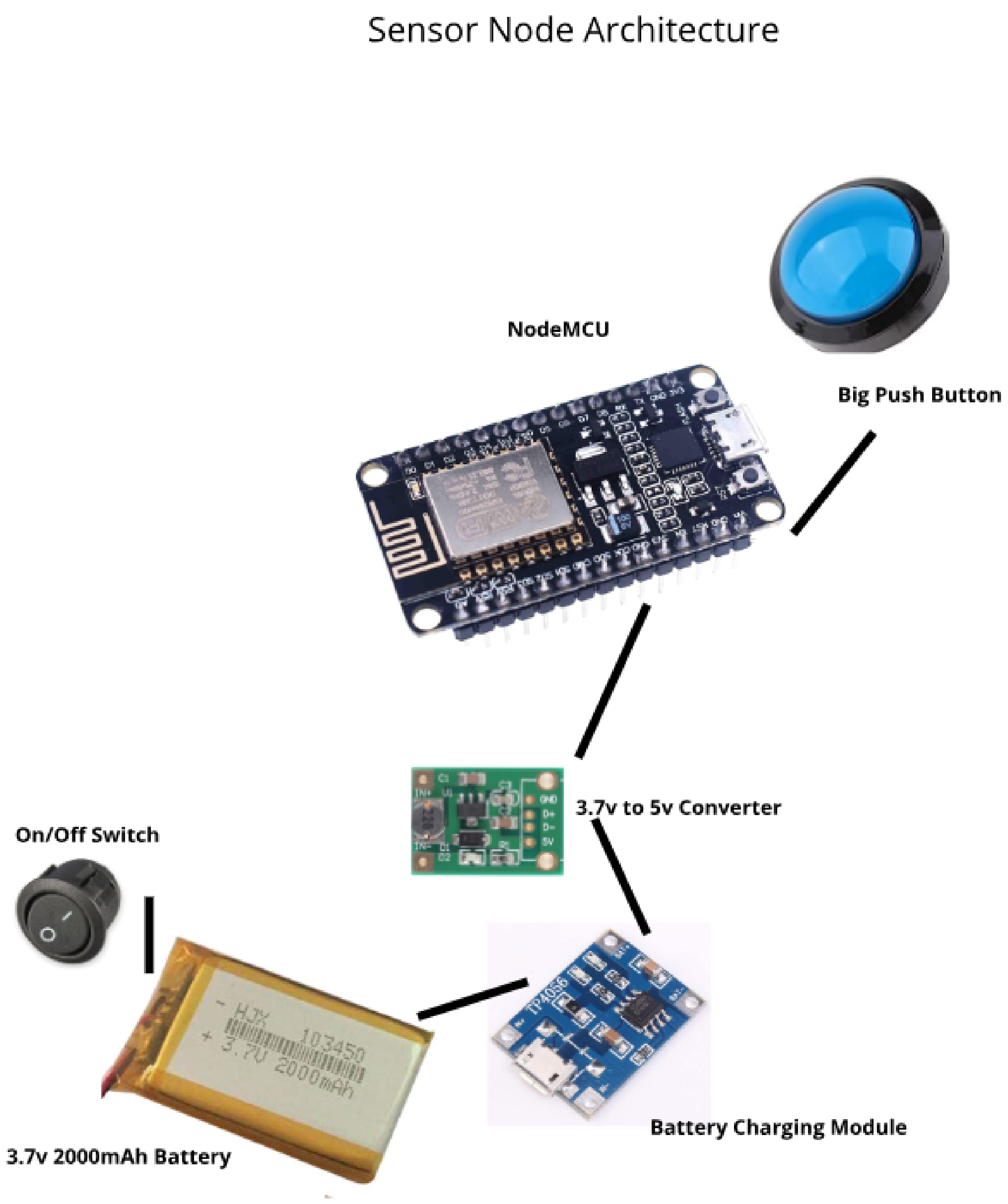
The complete product has

- Sensor Node
- Web Dashboard

A complete System is referred to as a Product in this report.

Everything will go into an enclosure.

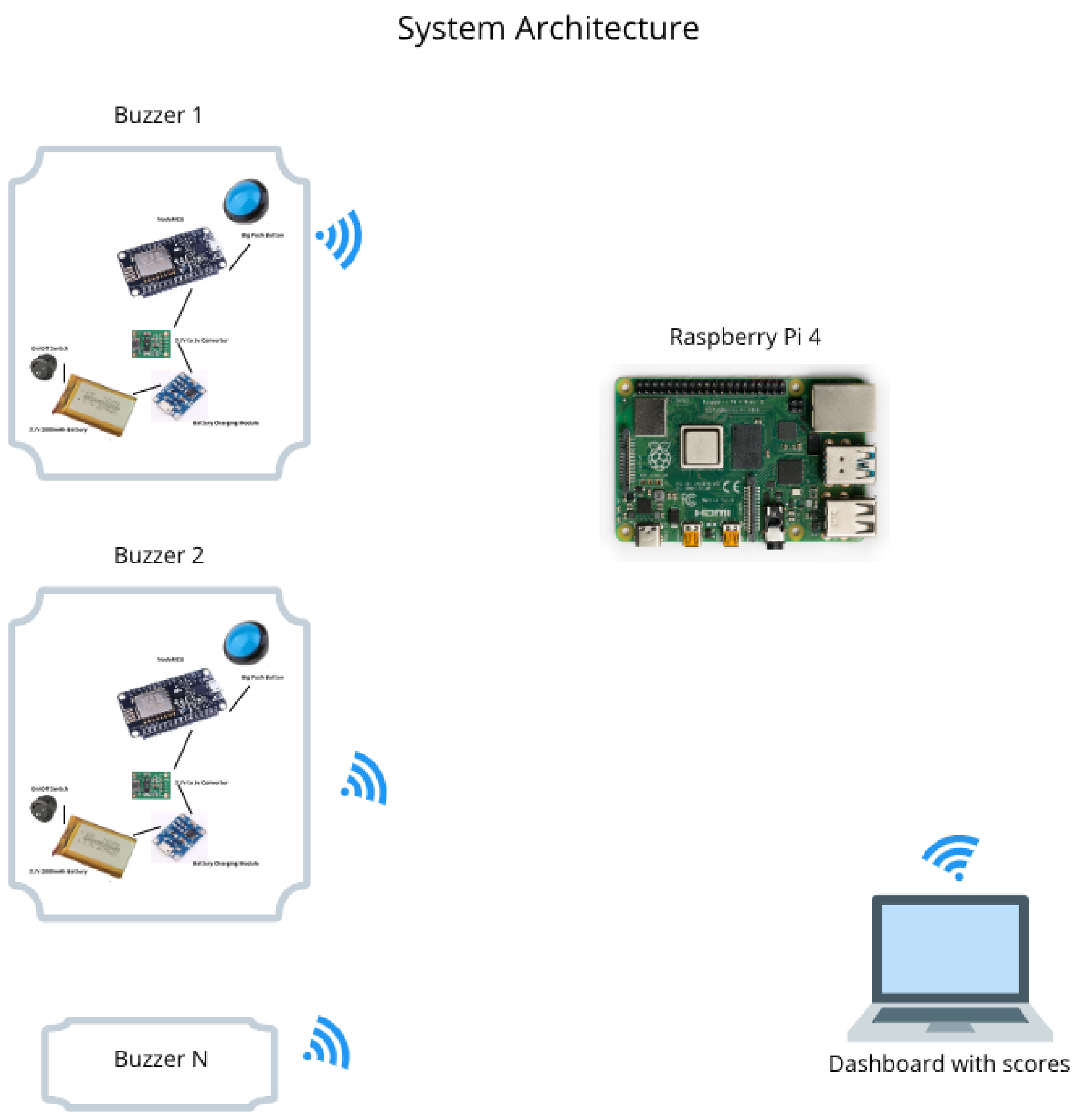
Hardware Architecture - Sensor Node



Above shown is the SmartBuzzerSystem Sensor Node hardware with different working units connected to the master ESP8266.

ESP8266 has WiFi built-in and it will connect to the Raspberry Pi.

Complete Architecture



The complete architecture of the product is shown above. There could be multiple Buzzers devices and all the buzzers will be wirelessly connected to the raspberry pi running the main dashboard. The dashboard will show information like connected buzzers, naming each buzzer and the number of buzzer button presses made. The dashboard will be accessible from any smartphone or laptop as a webapp.

Software Architecture

Buzzer Device

The buzzers will make a WiFi mesh which will allow any number of buzzers to be present in the system.

Multiple Nodes management

The nodes should have unique identifiers which are the MAC address numbers of ESP8266 which will be used to uniquely identify each Node.

SensorNodeID;[DataString]

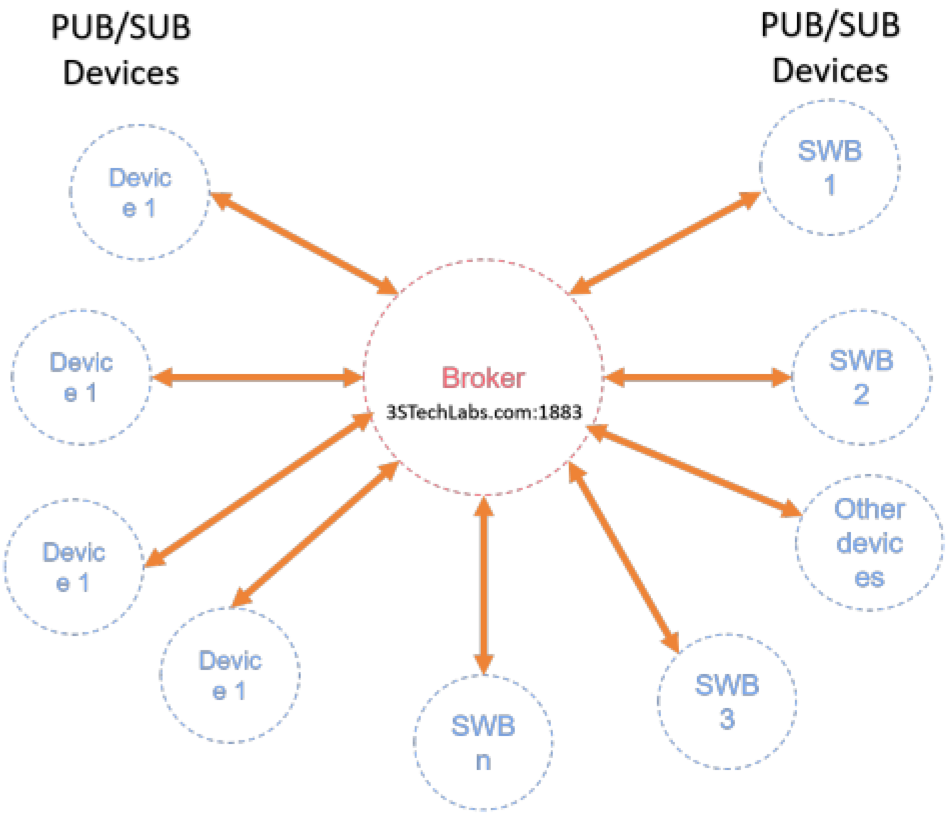
- SensorNodeID is a unique ID of Sensor Node
- [DataString] contains the sensor value(Status)

The communication is done over WiFi and data is shared to the RaspberryPi(and the webapp) over MQTT.

Web Server Integration

The communication should ideally be done using MQTT because of its smaller data packets and fast response.

What is MQTT?



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

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

It can handle two-way and parallel communication and the number of devices that can be connected and communicate 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.

Web App

The webapp will be made using NodeRED/VueJS and will show

- Number of connected buzzers.
- Number of time each buzzer is pressed
- Modifying the name of each buzzer.

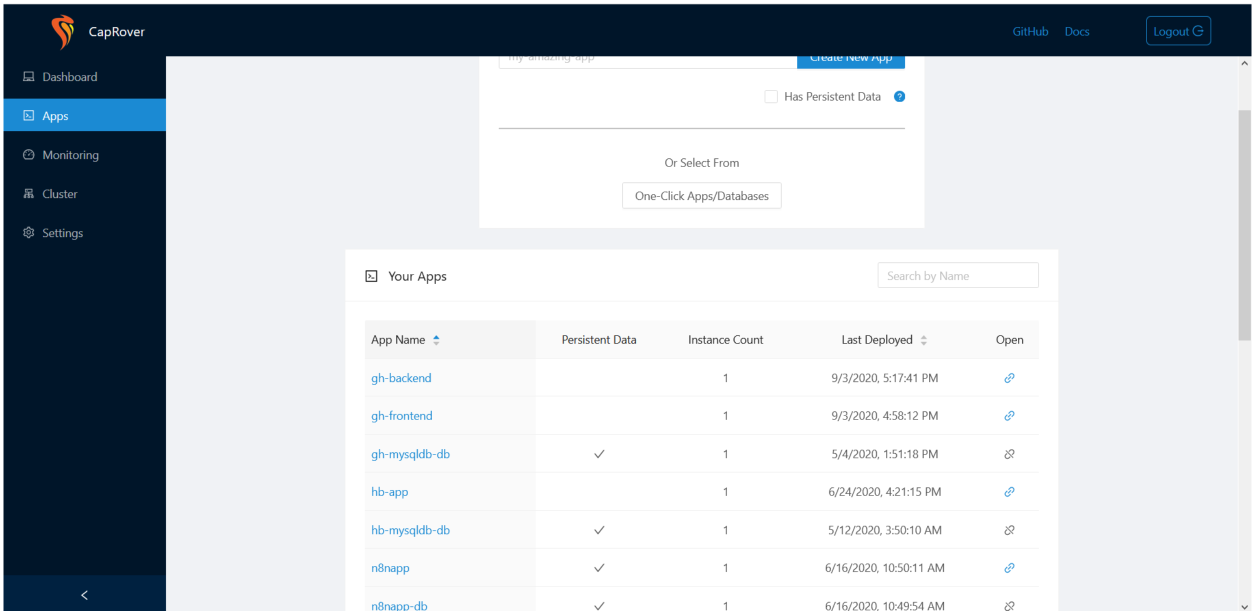
Management and CI/CD Pipeline

We will use Github for delivering the code. The developers from my team will be working on Hardware, Firmware, Backend, Frontend and apps simultaneously and for modularity we will dockerize different components of the app.

For Continuous Integration and Delivery we will be using CapRover running on a bare metal Ubuntu 18.04 instance and each component of the project's github repository will be linked to the respective Containers running in the Caprover. It will allow fully-automated delivery.

We will use our own company's production server running on AWS for testing and delivery during the development time. After that we can transfer the files to your own AWS and details of migration are mentioned in the Terms and Condition section of this report.

My automation engineer will work on this.

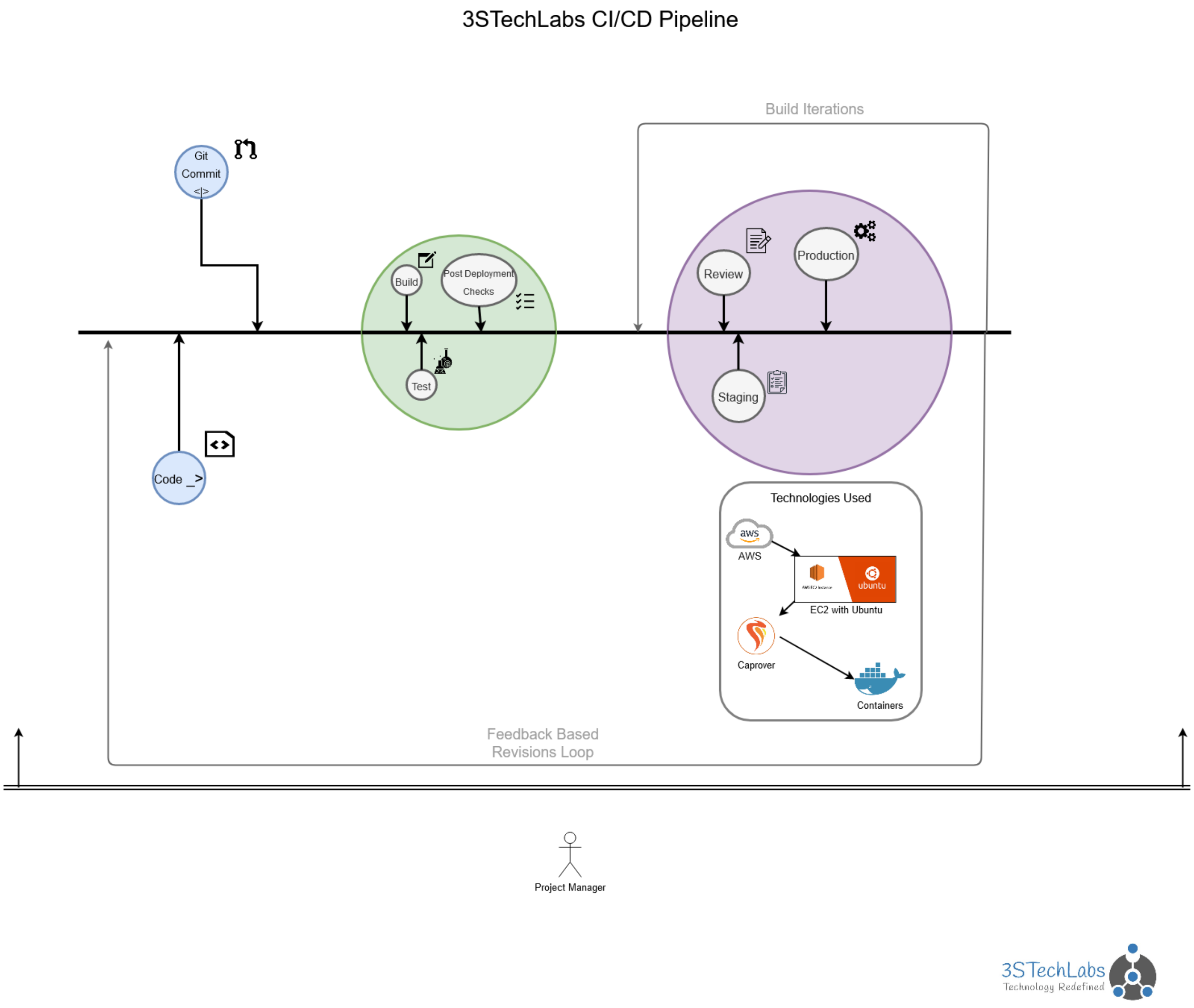


App Name	Persistent Data	Instance Count	Last Deployed	Open
gh-backend		1	9/3/2020, 5:17:41 PM	🔗
gh-frontend		1	9/3/2020, 4:58:12 PM	🔗
gh-mysqldb-db	✓	1	5/4/2020, 1:51:18 PM	⊘
hb-app		1	6/24/2020, 4:21:15 PM	🔗
hb-mysqldb-db	✓	1	5/12/2020, 3:50:10 AM	⊘
n8napp	✓	1	6/16/2020, 10:50:11 AM	🔗
n8napp-db	✓	1	6/16/2020, 10:49:54 AM	⊘

From the picture above you can see that all of the services like backend, frontend and database etc are running in separate containers allowing a smooth delivery pipeline.

3STechLabs' CI/CD Standard Pipeline

Below is our well-tested CI/CD Pipeline for project management and delivery. This has been proven to work for our 10+ Full-Stack IoT Projects and Products.

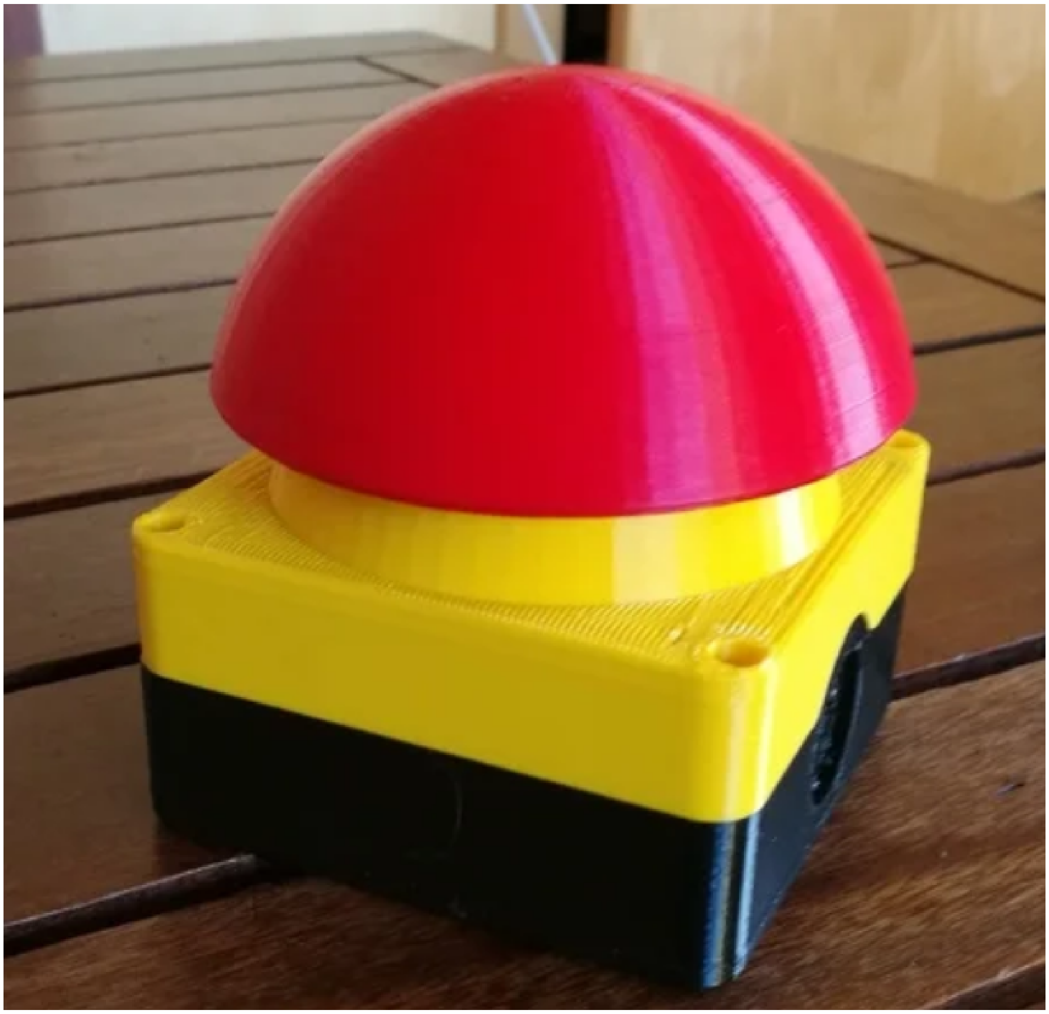


(For a detailed look, please open the image file of above diagram)

Node Casing

Sensor Node in a Case

The casing will be designed in Fusion360 and design will follow your proposed circular design.



Have a look at more pictures: https://www.thingiverse.com/thing:3771410

It is just a sample and we can have a different design as per your requirements.

Questions and Their Answers

2. Could you develop a prototype for us?

- Do you want me to send you the completed prototype to your address? If yes, unfortunately I can't do this at this moment and neither can I commit anything like that because of the current Covid-19 situation the flights are delayed eventually delaying the shipments and in some cases even missing the delivery altogether.
- On other hand, I will provide you
    - Complete Code
    - Circuit diagrams
    - A complete report on how to setup and configure things.
- You can use all the delivered files and documentation to assemble. the prototype yourself.

Milestones Breakdown

1. Sensor Node Design - **Milestone 1** **$250**
    a. Hardware and Firmware + Setup Document
    b. Breadboard Based POC with components mentioned in the list above
    c. Development Time 12 Days
2. Raspberry Pi Gateway Design - **Milestone 2** **$280**
    a. Dashboard Design
    b. MQTT Broker configuration, NodeRED, CapRover, Network configs
    c. Development time 7 Days
3. Casing Design - **Milestone 3** **$140**
    a. Casing design of sensor nodes and Gateway which could be 3D Printed
    b. Design time 5 Days

Maximum Development time 30 Days. Few milestones can be started in parallel to save time. We will be following project management techniques and will implement CI/CD for smooth delivery.

Terms and Conditions

- We will not ship anything physically unless otherwise decided, instead, you will be provided with a setup document with completion of each milestone which will be easy to follow and will contain all of the necessary information.
- The 3D casing design is something that will be finalized at the end of the project because it depends upon the final size of the circuit.
- 3STechLabs will provide all the files necessary for the mass-production of theSmart Buzzer System. If you want 3STechLabs to handle mass-production for you, we will charge for that else we will let you know the trusted vendors that you can contact and can get things manufactured on your own.

In this document, the company refers to 3STechLabs.

If you agree to this report please fill-out the form below.

https://naumanshakir3s.typeform.com/to/jZJlJP

Profile

Name: Nauman Shakir

Company: 3STechLabs

Designation: Founder and Program Manager

Email Address: NaumanShakir3S@gmail.com

Portfolio: https://NaumanShakir.com

I'm a Full-Stack IoT Developer and have done more than 150 hardware projects and running an IoT and Hardware Design House

https://3STechLabs.com

https://facebook.com/3STechLabs

https://Linkedin.com/company/3STechLabs

Freelancing Profiles

https://www.fiverr.com/naumanshakir

https://www.upwork.com/fl/naumanshakir3s