LoneTree Smart Dosing Device

status active

LoneTree Smart Dosing Device

📝 Table of Contents

📜 About

This repo contains

- Backend
- Firmware
- Detailed instructions

Smart Dosing Device

Getting Started

These instructions will get you a copy of the project up and running on your system.

Prerequisites

Things you need to install the FW.

```
- Arduino IDE
```

Installing

A step by step series that tell you how to get the Firmware and Backend running

ESP32 Configuration(Frequency Counter)

You should have Arduino IDE Installed

1. Add ESP32 Board to your Arduino IDE
2. In your Arduino IDE, go to File> Preferences Installing ESP32 Add-on in Arduino IDE Windows, Mac OS X, Linux open preferences
3. Enter `https://dl.espressif.com/dl/package_esp32_index.json` into the “Additional Board Manager URLs” field then, click the “OK” button: Note: if you already have the ESP32 boards URL, you can separate the URLs with a comma(each board will go to neaw line) as follows: `https://dl.espressif.com/dl/package_esp32_index.json,\n http://arduino.esp8266.com/stable/package_esp8266com_index.json`
4. Open the Boards Manager. Go to Tools > Board > Boards Manager…
5. Search for ESP32 and press install button for the ESP32 by Espressif Systems“:
6. That’s it. It should be installed after a few seconds.
7. In your Arduino sketchbook directory, create tools directory if it doesn't exist yet.
8. Unpack the tool into tools directory(present in libs/ESP32FS-1.0.zip) (the path will look like <home_dir>/Arduino/tools/ESP32FS/tool/esp32fs.jar).
9. Close and re-open the Arduino IDE.
10. Now copy the contents of the libs folder to the libraries directory of your Arduino
    1. If you are using windows, the libraries directory will be Documents/Arduino/libraries

ESP32 Node FW Uploading

1. Select ESP32 Dev Module from Tools->Board->ESP32
2. Select the correct port from Tools->Port
3. Then open ESP32_FW.ino file from Firmware/ESP32_FW folder.
4. Select Tools > ESP32 Sketch Data Upload menu item. This should start uploading the files into ESP32 flash file system.
5. Now Upload the Code to your ESP32 Dev Module.

6. Your ESP32 is now ready to be used.

Arduino Nano Configuration

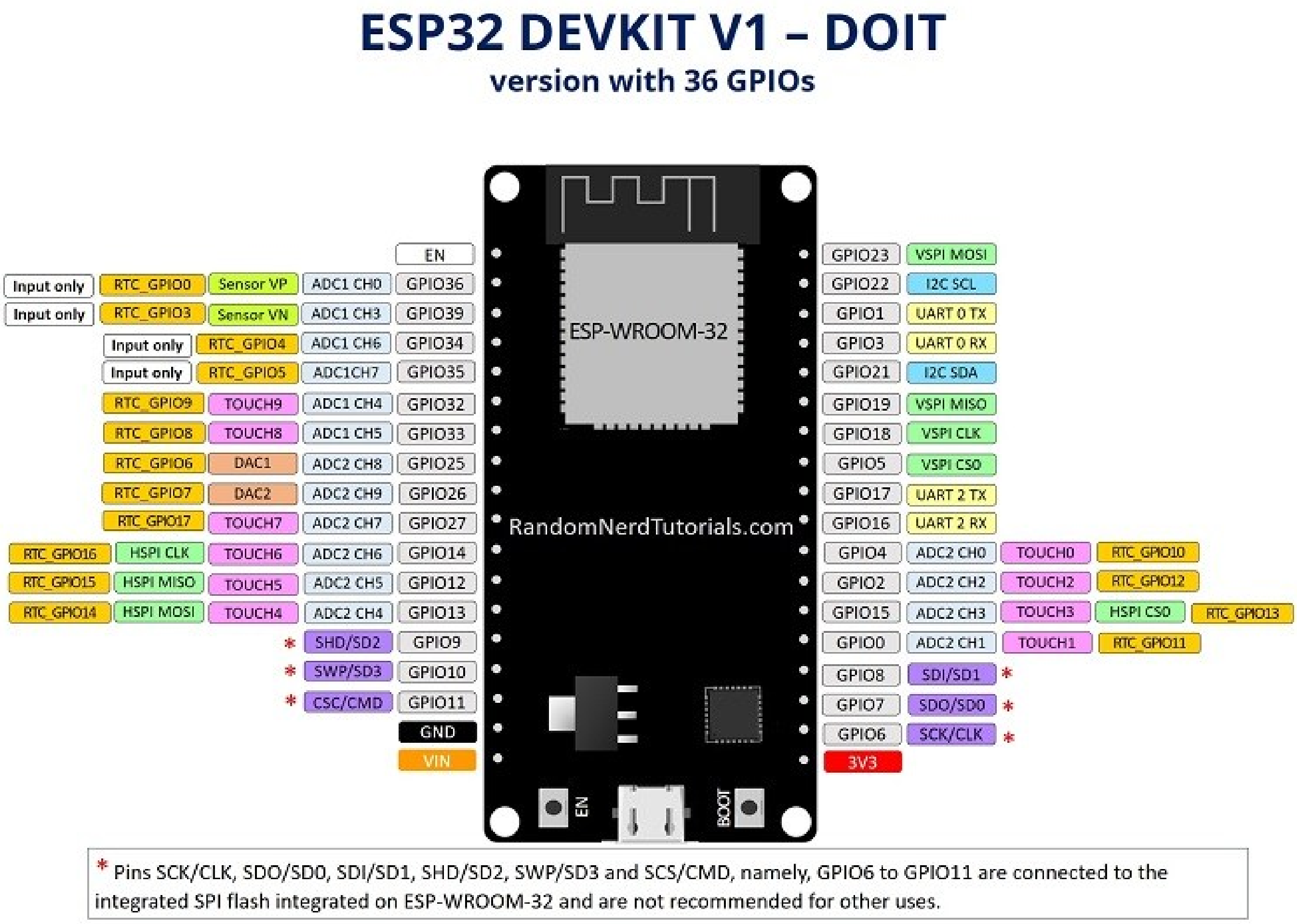
1. Open ArduinoNano_FW.ino file from Firmware/ArduinoNano_FW folder.
2. Select Arduino Nano from Tools->Board
3. Select the correct port from Tools->Port
4. Now Upload the Code to your Arduino Nano.

If the code uploaded succesfully, you can disconnect the USB Cable from Arduino Nano.

💡 Circuit

ESP32 DevKit V1 Pinout

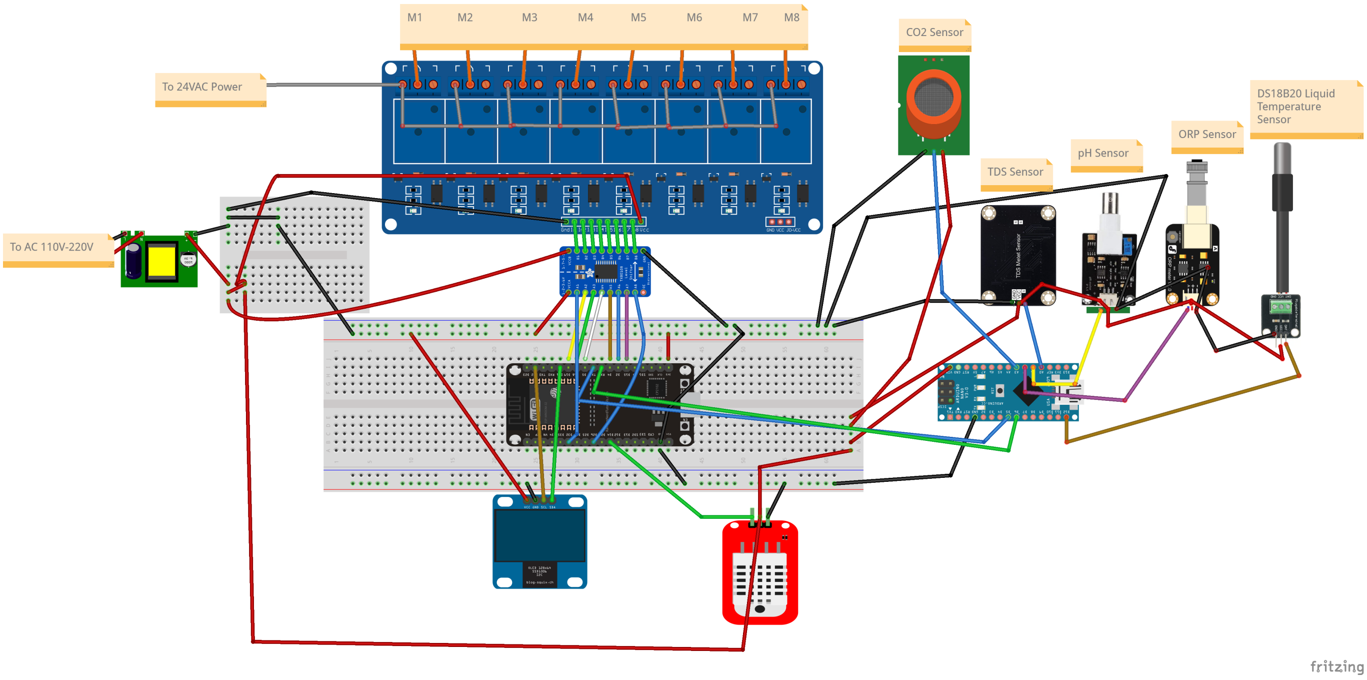
Follow the pinout diagram given below to connect different components to your ESP32 DevKit V1 board.


ESP32 DEVKIT V1 – DOIT
version with 36 GPIOs
ESP-WROOM-32
RandomNerdTutorials.com
EN
Input only RTC_GPIO0 Sensor VP ADC1 CH0 GPIO36
Input only RTC_GPIO3 Sensor VN ADC1 CH3 GPIO39
Input only RTC_GPIO4 ADC1 CH6 GPIO34
Input only RTC_GPIO5 ADC1CH7 GPIO35
RTC_GPIO9 TOUCH9 ADC1 CH4 GPIO32
RTC_GPIO8 TOUCH8 ADC1 CH5 GPIO33
RTC_GPIO6 DAC1 ADC2 CH8 GPIO25
RTC_GPIO7 DAC2 ADC2 CH9 GPIO26
RTC_GPIO17 TOUCH7 ADC2 CH7 GPIO27
RTC_GPIO16 HSPI CLK TOUCH6 ADC2 CH6 GPIO14
RTC_GPIO15 HSPI MISO TOUCH5 ADC2 CH5 GPIO12
RTC_GPIO14 HSPI MOSI TOUCH4 ADC2 CH4 GPIO13
* SHD/SD2 GPIO9
* SWP/SD3 GPIO10
* CSC/CMD GPIO11
GND
VIN
GPIO23 VSPI MOSI
GPIO22 I2C SCL
GPIO1 UART 0 TX
GPIO3 UART 0 RX
GPIO21 I2C SDA
GPIO19 VSPI MISO
GPIO18 VSPI CLK
GPIO5 VSPI CS0
GPIO17 UART 2 TX
GPIO16 UART 2 RX
GPIO4 ADC2 CH0 TOUCH0 RTC_GPIO10
GPIO2 ADC2 CH2 TOUCH2 RTC_GPIO12
GPIO15 ADC2 CH3 TOUCH3 HSPI CS0 RTC_GPIO13
GPIO0 ADC2 CH1 TOUCH1 RTC_GPIO11
GPIO8 SDI/SD1 *
GPIO7 SDO/SD0 *
GPIO6 SCK/CLK *
3V3
EN
BOOT

* Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and SCS/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on ESP-WROOM-32 and are not recommended for other uses.

Complete Circuit Diagram

Here's the complete circuit diagram of the system.



Other Components

```
Other components pin connection details
```

Smart Dosing Device

`Different Components connected to Arduino Nano for Frequency Measurements`

ESP32 With Arduino Nano Connections

`Arduino Nano Connected with ESP32`

Arduino Nano	ESP32
D5	TX2
D6	RX2
GND	GND

TDS Sensor

`Arduino Nano Connected with TDS Sensor`

TDS Sensor	Arduino Nano
S	A0
VCC	VIN
GND	GND

pH Sensor

`Arduino Nano Connected with pH Sensor`

pH Sensor	Arduino Nano
S	A1
VCC	VIN
GND	GND

ORP Sensor

`Arduino Nano Connected with ORP Sensor`

ORP Sensor	Arduino Nano
A	A2
+	VIN
-	GND

CO2 Sensor

`Arduino Nano Connected with CO2 Sensor(MQ-135)`

CO2 Sensor	Arduino Nano
A0	A3
VCC	VIN
GND	GND

DS18B20 Sensor

`Arduino Nano Connected with DS18B20 Sensor`

DS18B20 Sensor	Arduino Nano
A	D12
+	VIN
-	GND

Temperature and Humidity Sensor

`DHT22 Connected with ESP32`

DHT22 Pins	ESP32

DHT22 Pins	ESP32
`VCC`	`5V`
`GND`	`GND`
`Data`	`D12`

8 Channle Relay Module(SSR or Mechanical)

`Relay Connected with ESP32 via Logic Level Shifter`

Relay Module Pins	ESP32
`VCC`	`5V`
`IN1`	`D32`
`IN2`	`D19`
`IN3`	`D18`
`IN4`	`D5`
`IN5`	`D4`
`IN6`	`D2`
`IN7`	`D15`
`IN8`	`D26`
`GND`	`GND`

OLED Display

`OLED Connected with ESP32`

OLED Pins	ESP32
`VCC`	`3.3`
`GND`	`GND`
`SCL`	`D22`
`SCL`	`D21`

💻WebApp

WebApp can be accessed from the link below

- URL: https://lonetree-webapp.production.rehanshakir.com/
    - Username: admin@lontree.com

    - Password: 12345678

Data

Data published from the device would be in the JSON format given below:

```
{
  "macAddress": "",
  "temperature": "",
  "humidity": "",
  "liquidtemperature": "",
  "tds": "",
  "ph": "",
  "orp": "",
  "co2": "",
}
```

MQTT Topics

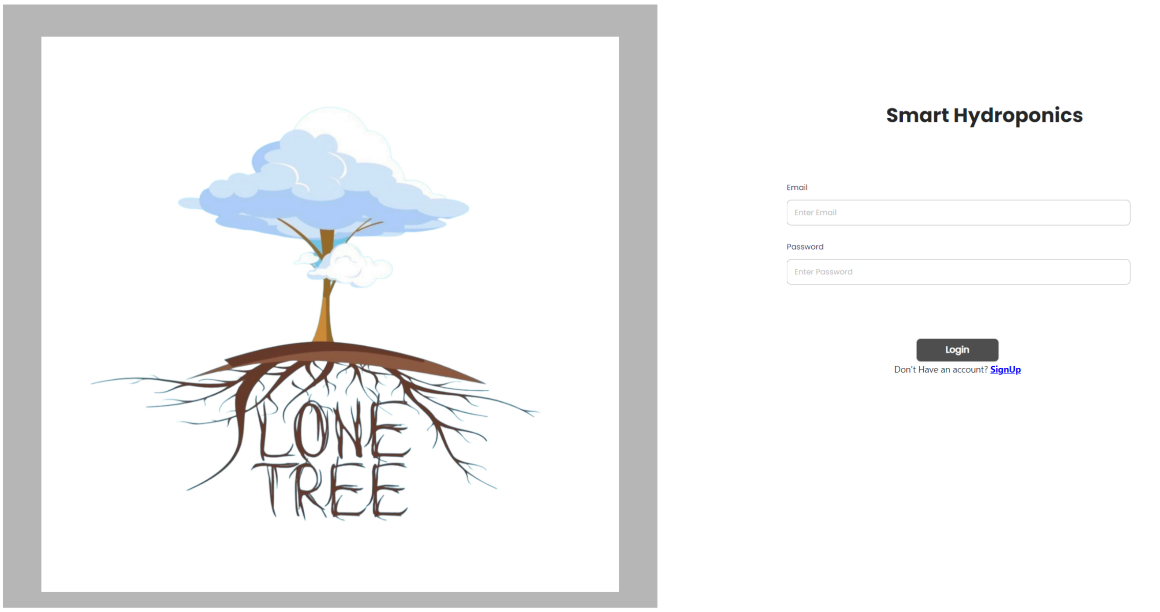
`MQTT Topic Details`

- As seen from ESP32

Topic	Type	Details
`smartdosing/`	`PUBLISH`	`Publishes the Sensors' payload JSON`
`macAddress/dosingControl`	`SUBSCRIBE`	`Control Pumps. Payload PumpNumber,PumpState. Where PumpNumber=1-8 and PumpState=0 or 1`

FrontEnd

- The WebApp is developed using NodeJS, ReactJS and MongoDB.



Smart Hydroponics

Email

Enter Email

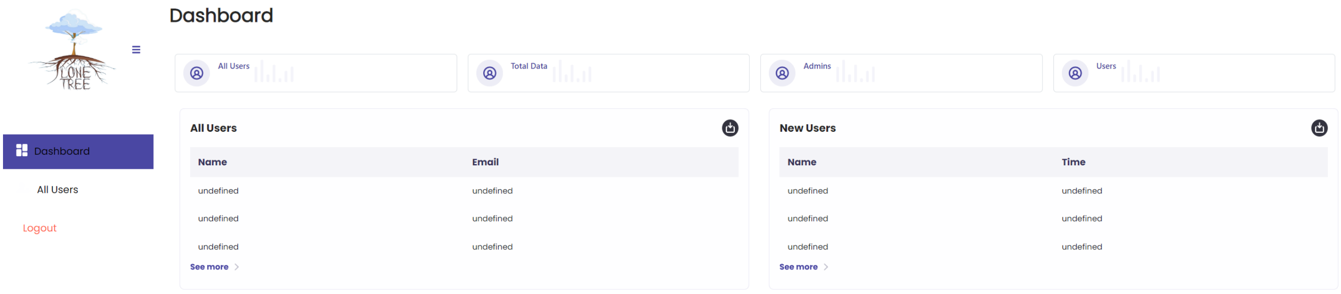
Password

Enter Password

Login

Don't Have an account? SignUp

Sign-up/Sign-in Page



All Users

Name	Email
undefined	undefined
undefined	undefined
undefined	undefined

New Users

Name	Time
undefined	undefined
undefined	undefined
undefined	undefined

Dashboard Home Page

Database

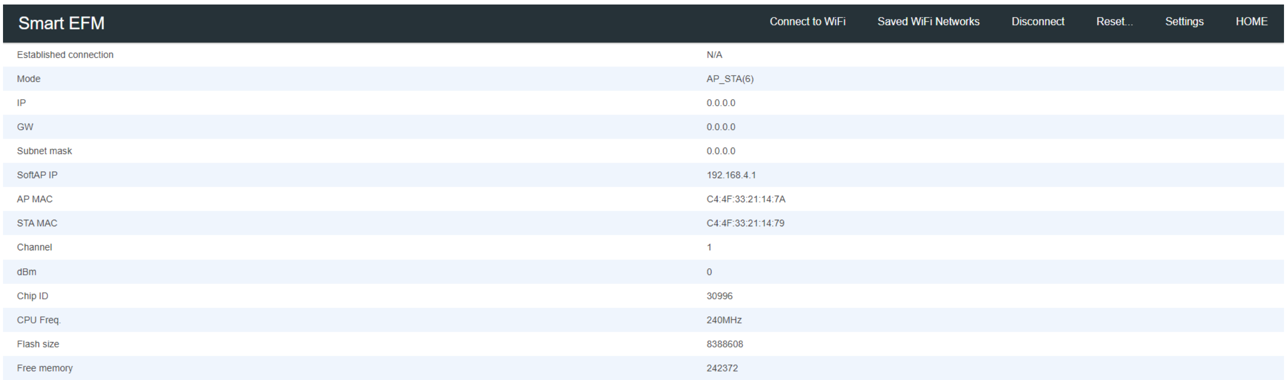
You can access the database from the link below

TO BE ADDED

Usage

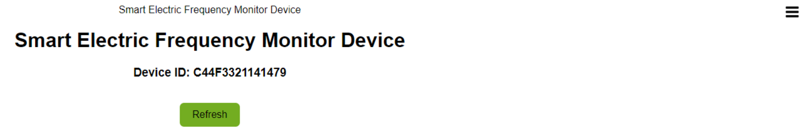
1. Upload the code to your ESP32
2. Upload the code to your Arduino Nano
3. Make the circuit
4. Power on your ESP32, it will present you with an AP named `SmartDosing-abcd` (while `SmartDosing` can be changed in the portal and `abcd` is a unique id for each esp32)
5. Default captive portal password `123456789AP` which can be changed in captive portal.
6. Connect to the ESP32 access point and open the web-browser and navigate to the link `http://esp32.local/_ac`. This link will work on most of the operating systems but if your operating system is not allowing to open it, you may want to check the captive portal IP Address from the serial monitor and can use that IP address inplace of the above mentioned URL.
7. The default access IP Address is `http://192.168.4.1/_ac`

8. You will be presented with a main dashboard as shown below(based on your device)



Established connection	N/A
Mode	AP_STA(6)
IP	0.0.0.0
GW	0.0.0.0
Subnet mask	0.0.0.0
SoftAP IP	192.168.4.1
AP MAC	C4:4F:33:21:14:7A
STA MAC	C4:4F:33:21:14:79
Channel	1
dBm	0
Chip ID	30996
CPU Freq.	240MHz
Flash size	8388608
Free memory	242372

- You can also open `http://esp32.local/` to access the CaptivePortal.

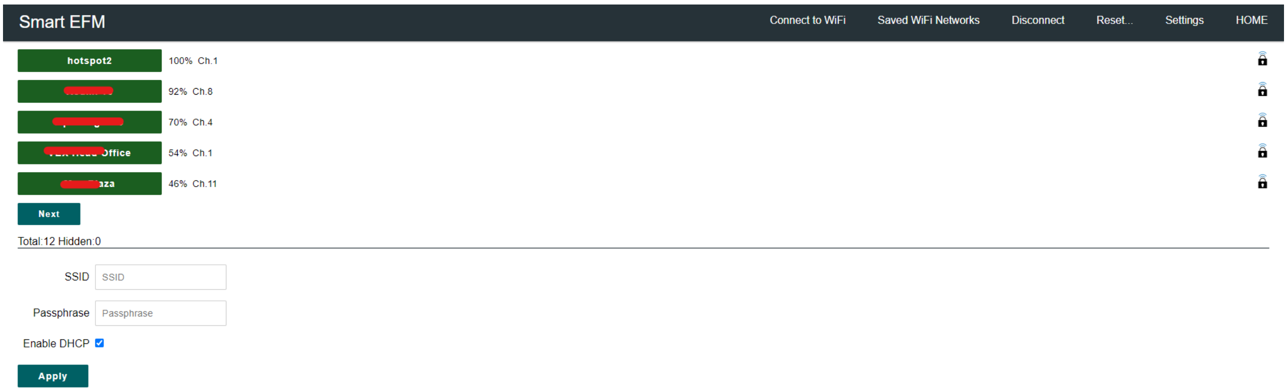


Smart Electric Frequency Monitor Device

Smart Electric Frequency Monitor Device

Device ID: C44F3321141479

9. You can connect to WiFi, from "Connect to WiFi" menu



hotspot2	100% Ch.1	🔒
[redacted]	92% Ch.8	🔒
[redacted]	70% Ch.4	🔒
[redacted] Office	54% Ch.1	🔒
[redacted]za	46% Ch.11	🔒

Next

Total:12 Hidden:0

SSID: SSID

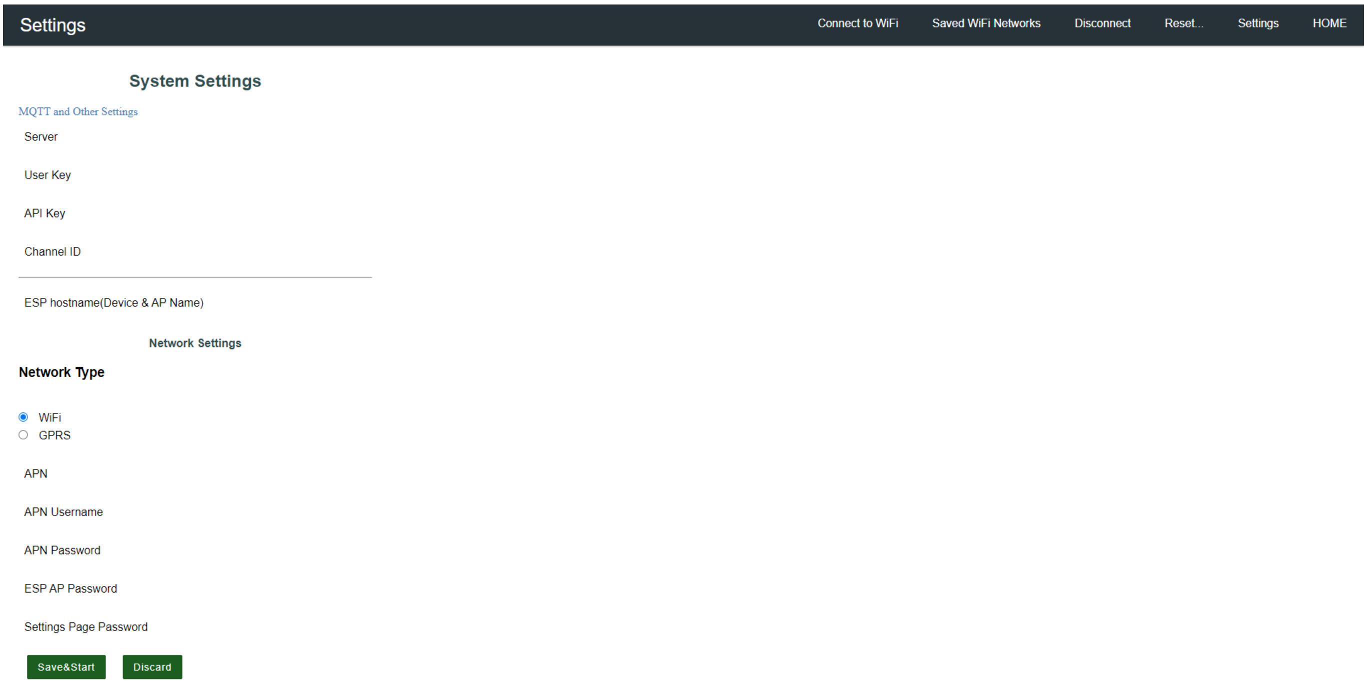
Passphrase: Passphrase

Enable DHCP ☑

Apply

10. Once connected to a WiFi network, you can again access the captive portal using same URL or the IP Address from the Serial monitor.
11. You can open settings page with following default credentials
12. User: **AP Name (SmartDosing)**
13. Password: **admin**

In settings page, you can configure your device. You can set network type (WiFi/GPRS) and you can set network APN details as well.



System Settings

MQTT and Other Settings

Server

User Key

API Key

Channel ID

ESP hostname(Device & AP Name)

Network Settings

Network Type

● WiFi
○ GPRS

APN

APN Username

APN Password

ESP AP Password

Settings Page Password

12. Open the dashboard to see the live data

PCB

TO BE ADDED

Casing

CAEFM CAEFMTOP

List of Components

Following components are used to make this project

1. Microcontrollers
   - ESP32 Dev Module
   - Arduino Nano
2. Sensors
   - TDS Sensor
   - pH Sensor
   - ORP Sensor
   - CO2 Sensor
   - DS18B20 Temperature Sensor Probe

   - Humidity and Temperature Sensor

3. Display

   - OLED

4. Power

   - 5v 5A DC Adapter

5. Misc

   - 8 Channel Relay Module for Pumps
   - Bi-directional logic level shifter

⛏️ Built Using

- Arduino - Embedded Framework and IDE - For Sensor Node Design

✍️ Authors

- @Nauman3S - Development and Deployment