

---

# POWERCLOUD INSTRUCTION MANUAL

---

ILLUSION SOLUTIONS

STUART ANDREWS  
12153983

MARC ANTEL  
12026973

MOTHUSI MASIBI  
12004589

BRANDON WARDLEY  
29005150

October 22, 2016

# Contents

<b>1</b>	<b>Overview</b>	<b>2</b>
<b>2</b>	<b>System Configuration</b>	<b>2</b>
<b>3</b>	<b>Compiling and Installing from Source Code</b>	<b>3</b>
3.1	Pre-requisites . . . . .	3
3.2	Firmware . . . . .	3
3.2.1	Manual Particle Photon Setup . . . . .	3
3.2.2	How to Register Device Onto System . . . . .	4
3.3	Application Server . . . . .	4
<b>4</b>	<b>Starting PowerCloud</b>	<b>5</b>
4.1	Firmware . . . . .	5
4.2	Application Server . . . . .	5
4.3	Web Application . . . . .	5
<b>5</b>	<b>Troubleshooting</b>	<b>9</b>

# 1 Overview

The goal of this project will be to measure the power consumption of machines in easy, cost effective way backed by a powerful cloud based interface. Which will allow the client to interrogate the data in a meaningful way. Providing in depth knowledge of the power consumption. This will allow them to reduce costs and save power leading to our ultimate goal of creating a better world for tomorrow. There is two main components to the project. There will be the software that runs on the particle(Electronic device) and is responsible for logging data to the server. Then the web application that will be responsible for displaying the power consumption. Advanced mathematical analysis and manipulation will be performed on the data. The project will have an optional goal of performing analysis on harmonics. All electronic equipment required will be provided to the students.

## 2 System Configuration

The system is comprised of multiple components, a web application and a Java-MQTT server, with Google's Firebase as the persistence.

- a power usage device,
  - The device consists of multiple separate parts:
    - \* A Particle Photon
      - Uses header pins connected to a bread board to read values.
      - Uses the MQTT messaging protocol to send data to the Java server on port 1883.
      - Uses the Particle API for everything else.
      - The Particle code is C++.
    - \* A multifunction power monitor, used to measure kWh.
    - \* A current transformer, used to acquire accurate and real time current.
    - \* A relay to control the flow of electricity through the device.
    - \* A trip switch to manually turn the device on or off.
- a Java-MQTT server,
  - Uses the Moquette MQTT library to listen, on port 1883, for messages sent from the Device.
  - Uses the Firebase API in order to send and retrieve data, via HTTPS, to and from the database.
- a web application,
  - Uses AngularJS.
  - Uses the Firebase API in order to send and retrieve data, via HTTPS, to and from the database.
- Google Firebase
  - Uses HTTPS to communicate with the Java server and the Web application.

## 3 Compiling and Installing from Source Code

### 3.1 Pre-requisites

**Software** Before proceeding please ensure you have the following software installed on your system:

- NodeJS and NPM
- Bower
- Gulp
- Particle Dev
- Java 8
- Maven

#### Hardware

- A Particle Photon.
- A computer with internet access which does not block incoming and outgoing connections on port 1883 and 80.

### 3.2 Firmware

#### 3.2.1 Manual Particle Photon Setup

Compile and flash the photon with the following commands, please take note that if you run the application server on a local machine, retrieve the IP address and insert it into the firmware's code as shown below.

```
// Callback signature for MQTT subscriptions.
void callback(char* topic, byte* payload, unsigned int length);

// MQTT client.
byte PC_BROKER[] { 178,62,75,151 };
MQTT client(PC_BROKER, 1883, callback);

void callback(char* topic, byte* payload, unsigned int length) {
    Serial.println("Command received:");
    Serial.println((char*)payload);
};
```

Follow the instructions here in order to compile the firmware using Particle Dev.

- Open the firmware using Particle Dev.
- Click on "Compile using Particle Cloud".
- Select the device to flash.
- Click on "flash device".

*If you run into errors, please see the troubleshooting guide at the end of this document.*

### 3.2.2 How to Register Device Onto System

In order to register a new device onto the PowerCloud system, follow the steps:

- Login to <https://power-cloud.tech>.
- Proceed to the profile tab, and click on it.
- Once in the profile tab (see page x, picture y) locate Particle.io Settings.
- Using your Particle login details, login into Particle.io to gain access to the Particle API.
- Now locate Devices on the left hand side, click on it.
- Locate the Add Device tab, click "New".
- Do the following:
  - Enter the name of the device e.g. "PhotonIT444".
  - Enter the appliance that which the device is attached e.g. "IT 4-44".
  - Enter the ID of the Photon, in order to claim it to your Particle account.
  - Enter the threshold of the device, so as to protect your device from potentially high currents that could prove disastrous should they reach the appliance.
  - Select the time interval between readings from the drop down list.

### 3.2.3 How to Register on the Particle.io Site

- Navigate to <https://login.particle.io/console>.
- Click "Sign Up".
- Enter the ID of the Photon, in order to claim it to your Particle account.
- Do the following:
  - Enter an email for your username.
  - Enter a password and verify it.
  - Click sign up.

## 3.3 Application Server

Currently, the *application-server* branch hosts the source code for the Application Server. It needs to be compiled using maven and run on a device which doesn't block incoming and outgoing connections on port 1883.

### Compiling the Application Server

Download the sources from GitHub and compile:

- `$ mvn package`

*If you run into errors, please see the troubleshooting guide at the end of this document.*

## 4 Starting PowerCloud

### 4.1 Firmware

Once the firmware has been compiled and the binary file produced. Flash the photon with the following command:

- *particle serial flash Photon Name \*.ino*

### 4.2 Application Server

Once the application server has been compiled, change your directory to IllusionSolutions/test/, run the application server using the following command

- *java -jar applicationServer-\*.jar*

### 4.3 Web Application

To start the PowerCloud Web application open the console.

- *npm install*
- *bower install*
- *gulp serve*

The web application should now be available on **http://localhost:9000/**

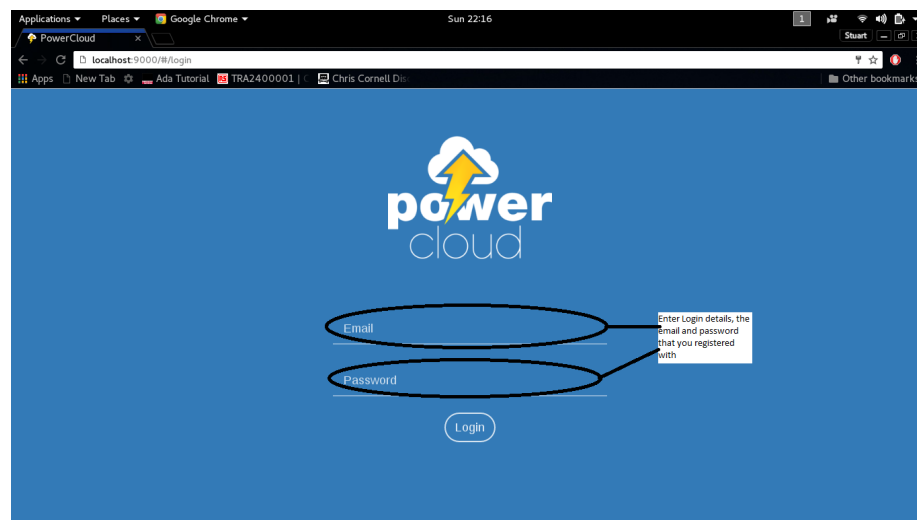


Figure 1: Initial login screen.

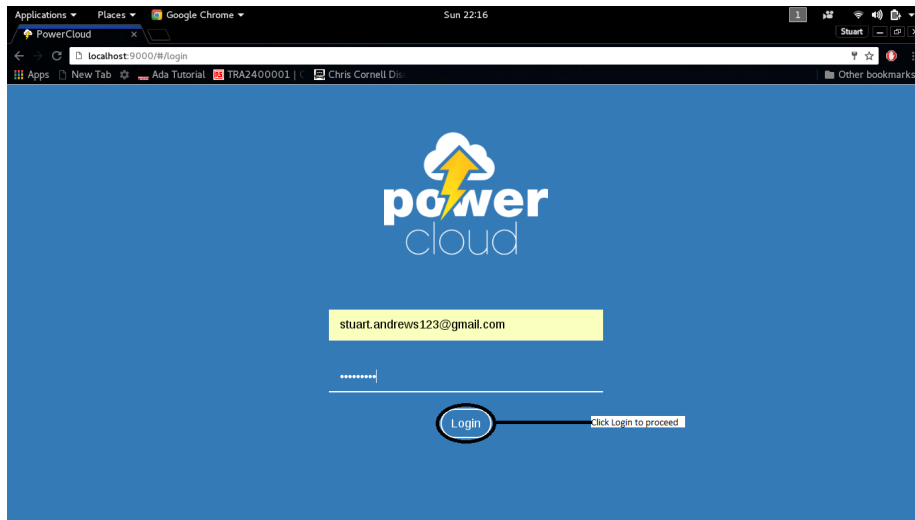


Figure 2: Completed login screen.

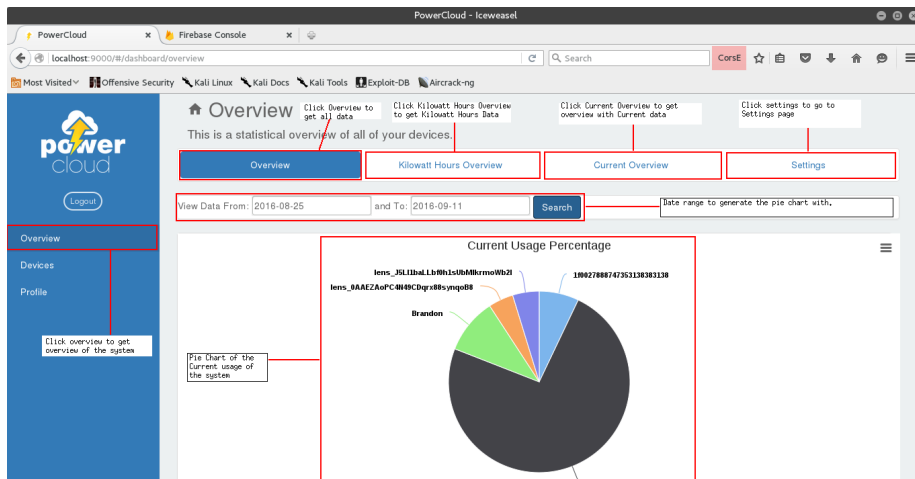


Figure 3: Overview screen.

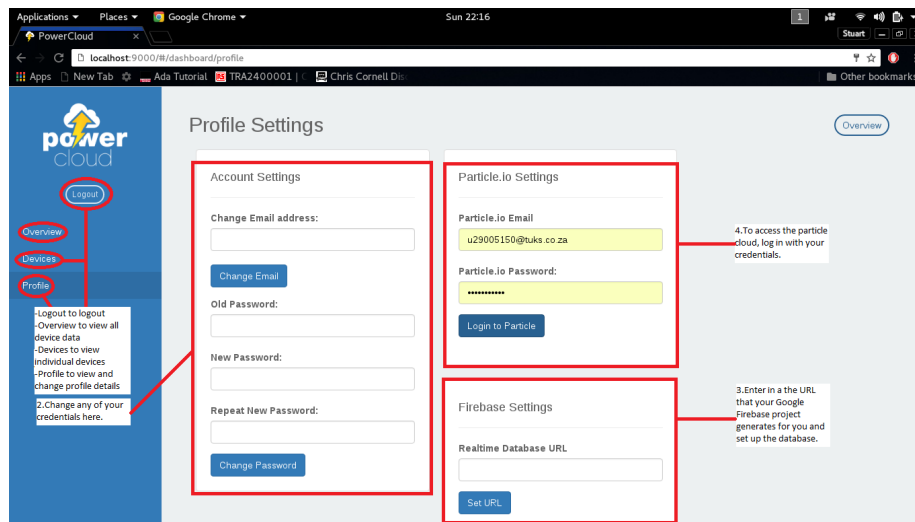


Figure 4: Profile settings.

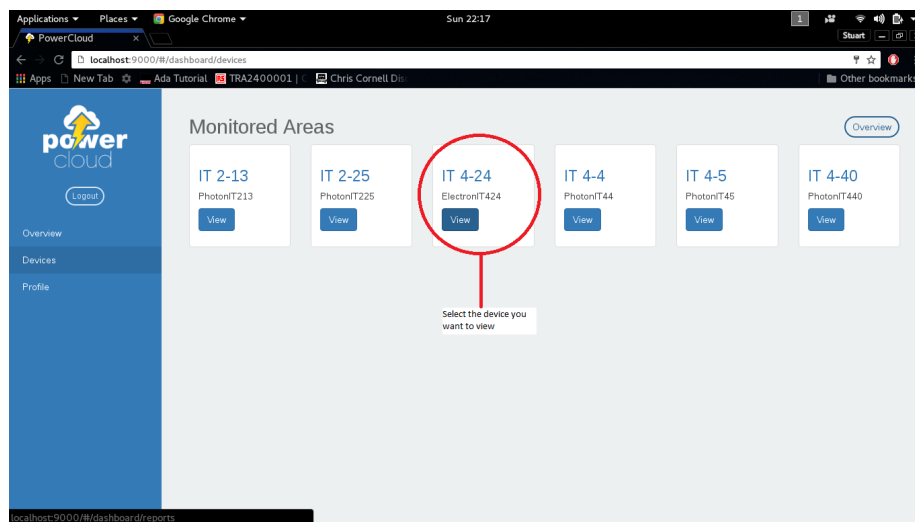


Figure 5: Monitored Areas.



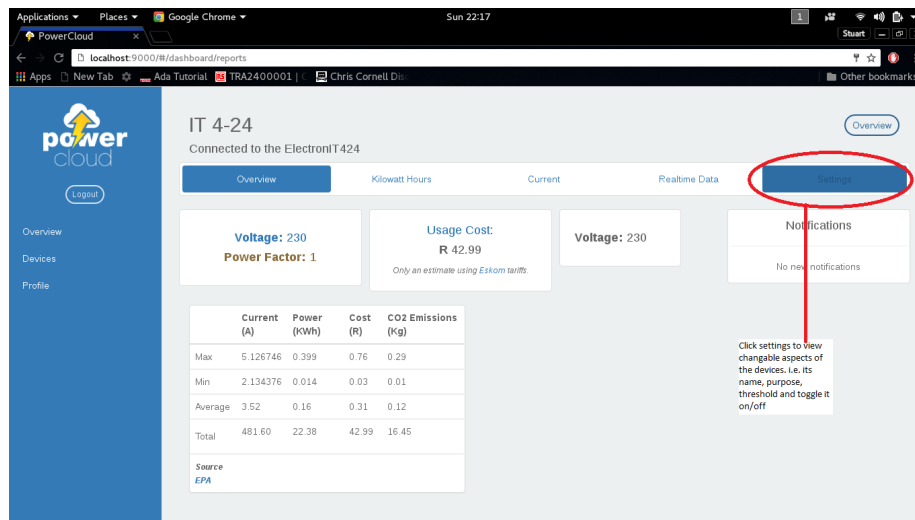


Figure 6: Data display page.

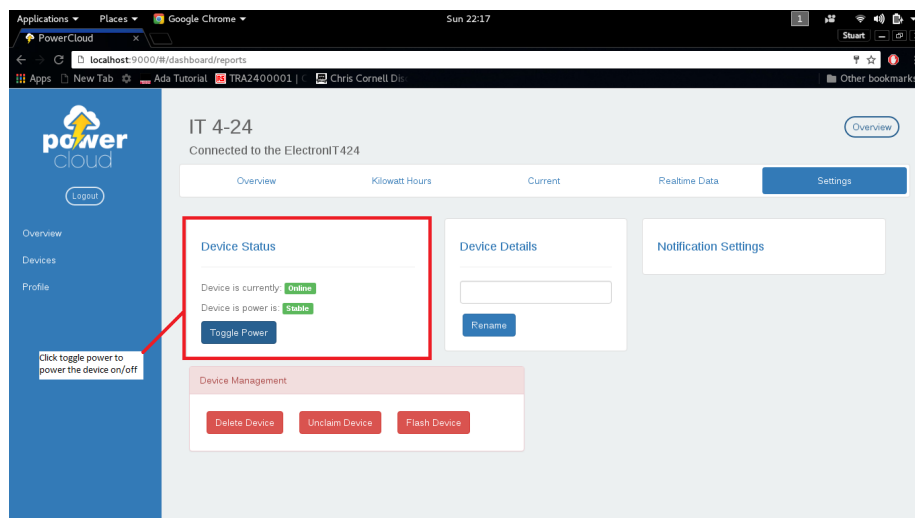


Figure 7: Device settings page.

## 5 Troubleshooting

**I am getting errors when trying to install through the console.**

- Try using sudo before the command. Example: `sudo npm install particle-cli`

**The Web Application won't run and is giving errors when executing the gulp serve command.**

- Make sure you've run `npm install` and `bower install`.
- Make sure you're in the correct directory where the web application is located.