

HOME AUTOMATION

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF ENGINEERING IN COMPUTER SCIENCE & ENGINEERING



Submitted to: Manish Sharma Sir

Project Teacher

**Submitted By:- Keshav Kant
Mishra
(19BCS1887)**

**Project members:-
Harsh Anand (19BCS1895)
Bobby Sharma (19BCS1862)
Keshav Kant Mishra (19BCS1887)
Shubham Kumar (19BCS1883)**

A handwritten signature in black ink, appearing to read "Jha".

**Mentor Signature
(Sudan Jha Sir- 10119)**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Chandigarh University, Gharuan

January,2021

CERTIFICATE

This is to certify that the work embodied in this Project Report entitled "**Home Automation**" being submitted by "**Keshav Kant Mishra**" - UID "**19BCS1887**" , 4th Semester for partial fulfillment of the requirement for the degree of " **Bachelor of Engineering in Computer Science & Engineering** " discipline in " **Chandigarh University** " during the academic session Feb-May 2021 is a record of bonafide piece of work, carried out by student under my supervision and guidance in the " **Department of Computer Science & Engineering** ", Chandigarh University.

APPROVED & GUIDED BY: **Manish Sharma Sir (Project teacher)**

Sudan Jha Sir (Project Mentor)



DECLARATION

I, student of **Bachelor of Engineering in Computer Science & Engineering, 4th Semester**, session: **Feb – May 2021, Chandigarh University**, hereby declare that the work presented in this Project Report entitled "**Home Automation**" is the outcome of my own work, is bona fide and correct to the best of my knowledge and this work has been carried out taking care of Engineering Ethics. The work presented does not infringe any patented work and has not been submitted to any other university or anywhere else for the award of any degree or any professional diploma.

KESHAV KANT MISHRA

(19BCS1887)

Student details and Signature

APPROVED & GUIDED BY: Manish Sharma Sir (Project teacher)

Sudan Jha Sir (Project Mentor)



ACKNOWLEDGEMENT

This project is a synergistic product of many minds and has been accumulated over the last few months. This has been a special project brought to fruition through the efforts of some very special people. Many people contributed enthusiastically to this project, which really came together in the last few weeks before deadline. For their continuous guidance and valuable advice I would like to take this opportunity to thank: Firstly, I would like to thank Chandigarh University who has granted me this opportunity to prepare a project which has helped me to gain knowledge beside my studies and which is also definitely going to be useful in future. I would like to place on record my deep sense of gratitude to my parents for financial wisdom and inspiration that have guided and helped me from day one. My project guide, Mr. Manish Sharma Sir (Project teacher) and my Project mentor Mr. Sudan Jha Sir, who came in and battened down the hatches when things were flying about. And I would like to thank all those who have helped me contribute their valuable insights and time for this project.

TABLE OF CONTENT

1. INTRODUCTION
 - i. What is IoT ?
 - ii. What is Home Automation
2. Project Design and Innovation
 - i. DFD Level 0
 - ii. DFD Level 1
 - iii. Circuit Design
3. Project Implementation
 - iv. Adding a new device
 - v. Creating an account on the IoT Cloud Service
 - vi. Creating a product
 - vii. Configuring Product Hardware
4. Project Implementation and Testing
 - viii. Setting up the Bolt WiFi module
 - ix. Before we move forward
 - Introduction to API
 - Introduction to Anomaly
 - Introduction to Z - Score
 - x. Implementing Circuit Connections
 - xi. Implementation of Coding part
 - Introduction to Ubuntu
 - Introduction to VMware Workstation
 - Introduction to Python
 - Introduction to JavaScript
 - Introduction to Twillo

- xii. Output Explanation
 - Output Part 1
 - Output Part 2
 - Output Part 3
- 5. Final Setup (Completed)
- 6. The END GAME!
 - i. Module and Team member wise distribution of work
- 7. Bibliography

INTRODUCTION

“What is IoT?”

When you Google “what is IoT,” many of the answers are unnecessarily technical. Case in point:

“The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.”

– An unnecessarily technical explanation of IoT

If you just read that and thought, “ok, so what?”, you’re not alone. Most people don’t want to nor need to dive into the nitty-gritty of IoT. In this chapter, we’ll provide you with a simple explanation of the Internet of Things and what it means for you.

Before we jump in, note that “The Internet of Things” and “IoT” can and will be used interchangeably. And a quick tip to sound knowledgeable: avoid saying “the IoT”.

A Simple, Non-Technical Explanation of the Internet of Things

How are you reading this ebook right now? It might be on desktop, on mobile, maybe a tablet, but whatever device you’re using, it’s most likely connected to the internet.

An internet connection is a wonderful thing, it give us all sorts of benefits that just weren’t possible before. If you’re old enough, think of your cell phone *before* it was a smartphone. You could call and you could text, sure, but now you can read any book, watch any movie, or listen to any song all in the palm of your hand.

The point is that connecting things to the internet yields many amazing benefits. We’ve all seen these benefits with our smartphones, laptops, and tablets, but this is true for everything else too. And yes, we do mean *everything*.

The Internet of Things is actually a pretty simple concept, **it means taking all the physical places and things in the world and connecting them to the internet.**

Confusion arises not because the concept is so narrow and tightly defined, but rather because it’s so broad and loosely defined. It can be hard to nail down the concept in your head when there are so many examples and possibilities in IoT.

To help clarify, it’s important to understand the benefits of connecting things to the internet. Why would we even want to connect everything to the internet?

What Is Home Automation?

Home automation is the automatic control of electronic devices in your home. These devices are connected to the Internet, which allows them to be controlled remotely. With home automation, devices can trigger one another so you don't have to control them manually via an app or voice assistant. For example, you can put your lights on schedules so that they turn off when you normally go to sleep, or you can have your thermostat turn the A/C up about an hour before you return to work so you don't have to return to a stuffy house. Home automation makes life more convenient and can even save you money on heating, cooling and electricity bills. Home automation can also lead to greater safety with Internet of Things devices like security cameras and systems. But hold up; what's the Internet of Things?

How Does Home Automation Work?

Home automation works via a network of devices that are connected to the Internet through different communication protocols, i.e Wi-Fi, Bluetooth, ZigBee, and others. Through electronic interfaces, the devices can be managed remotely through controllers, either a voice assistant like Alexa or Google Assistant or an app. Many of these IoT devices have sensors that monitor changes in motion, temperature and light so the user can gain information about the device's surroundings. To make physical changes to the device, the user triggers actuators, the physical mechanisms like light switches, motorized valves or motors that allows devices to be controlled remotely.¹

Home Automation System Components

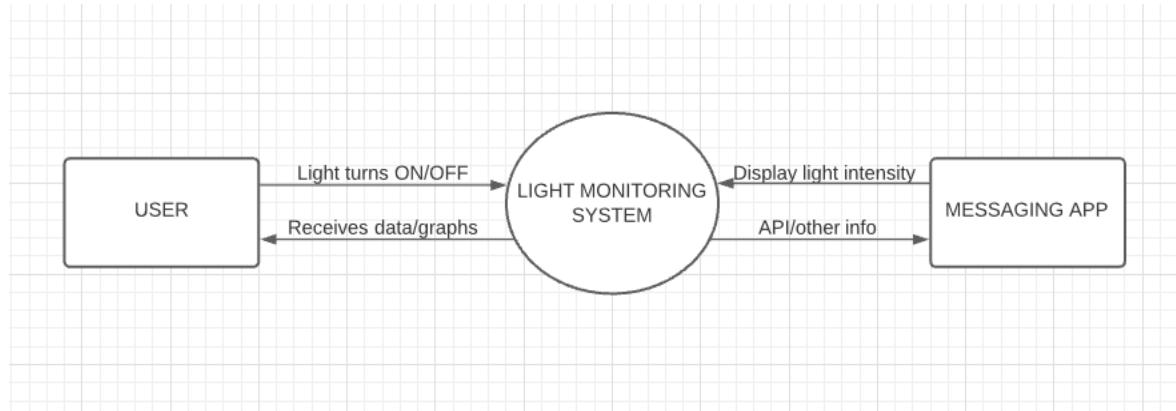
While some home automation systems require hubs, some mobile applications connect directly to a router, which connects directly to an IoT device. Of course, it's preferable when there's no hub, as that's just an added cost on top of the cost of the IoT device itself.

Internet of Things vs. Home Automation

The Internet of Things, commonly known as IoT, refers to any device that's connected to the Internet that isn't normally; for example, a smart light bulb that you can turn on and off via an app. All home automation devices are IoT devices, which can be automated to trigger one another. So while IoT refers to the devices themselves, home automation is what you can do with the IoT devices to make your life just a tad bit easier.

PROJECT DESIGN AND INNOVATION

DFD (Data Flow Diagram) Level 0



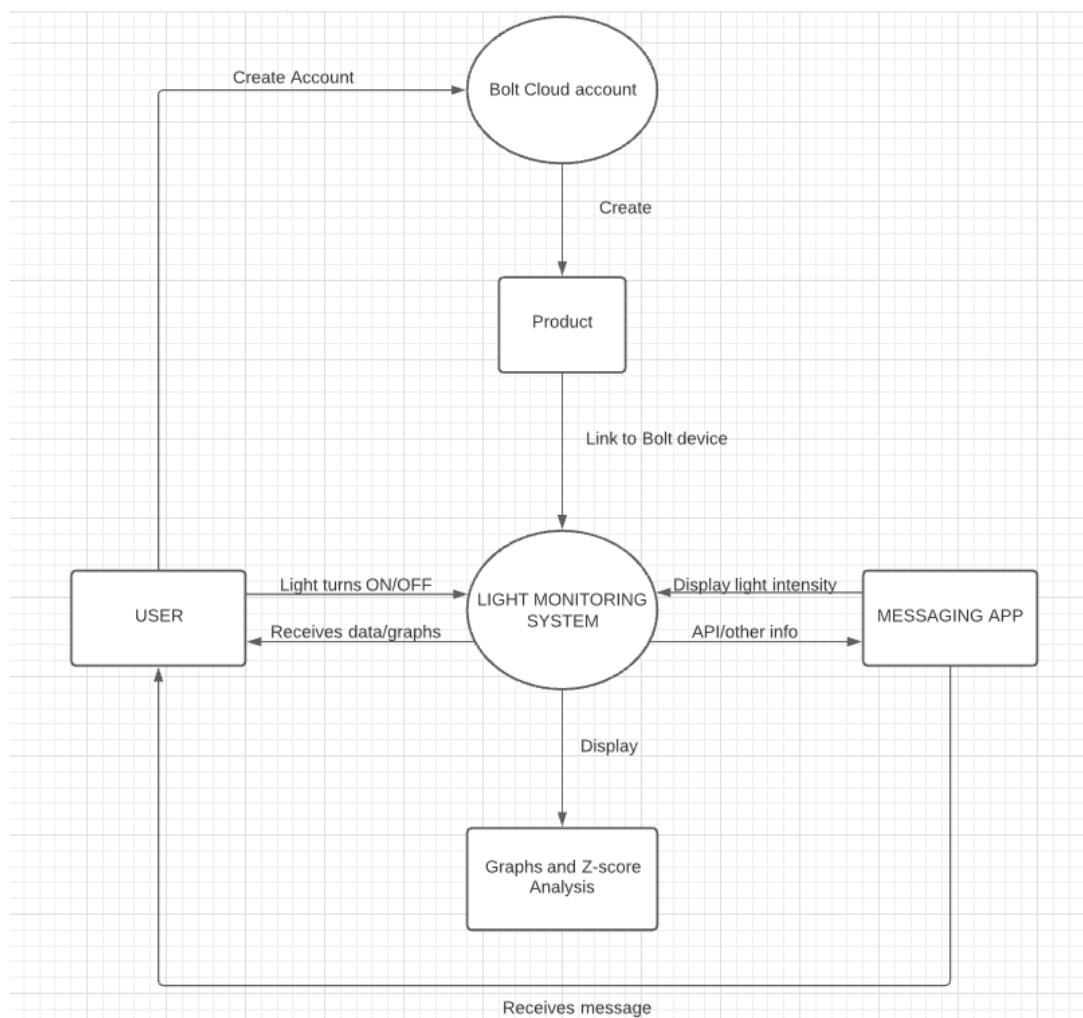
DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.

Here, the rectangle shapes represent the External entity. The arrows connecting the rectangular shapes that is external entity represents data flow or the flow of data between external entities. And finally the circular part represents the Process that will be performed. In this level 0 DFD(Data flow diagram), the User turns the light off/on, then the Light Monitoring system operates and configures specific information or API's. After using the messaging API (Twilio application), the user receives the text message on their smartphone whether the light is on or off and also we can display light intensity data.

DFD (Data Flow Diagram) Level 1

DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. You will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its subprocesses.

After the overview of the design explained in DFD level 0, here is the detailed explanation. User creates a account on the Bolt cloud and just signs up and link or add the product that is the Light Monitoring System to the cloud. After performing these operations the user would be able to see the predictable graph and the Z- Score Analysis of the intensities of light at different timings.

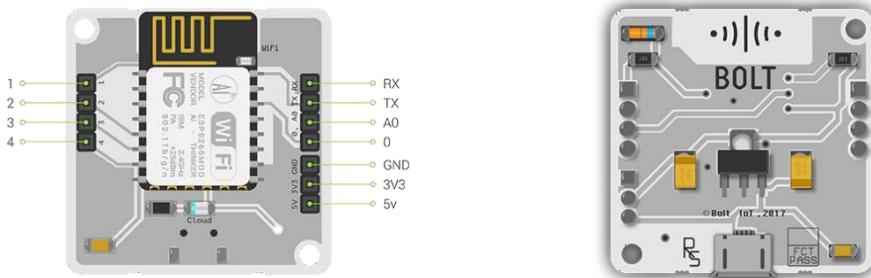


Circuit Design

WiFi Module

Device View

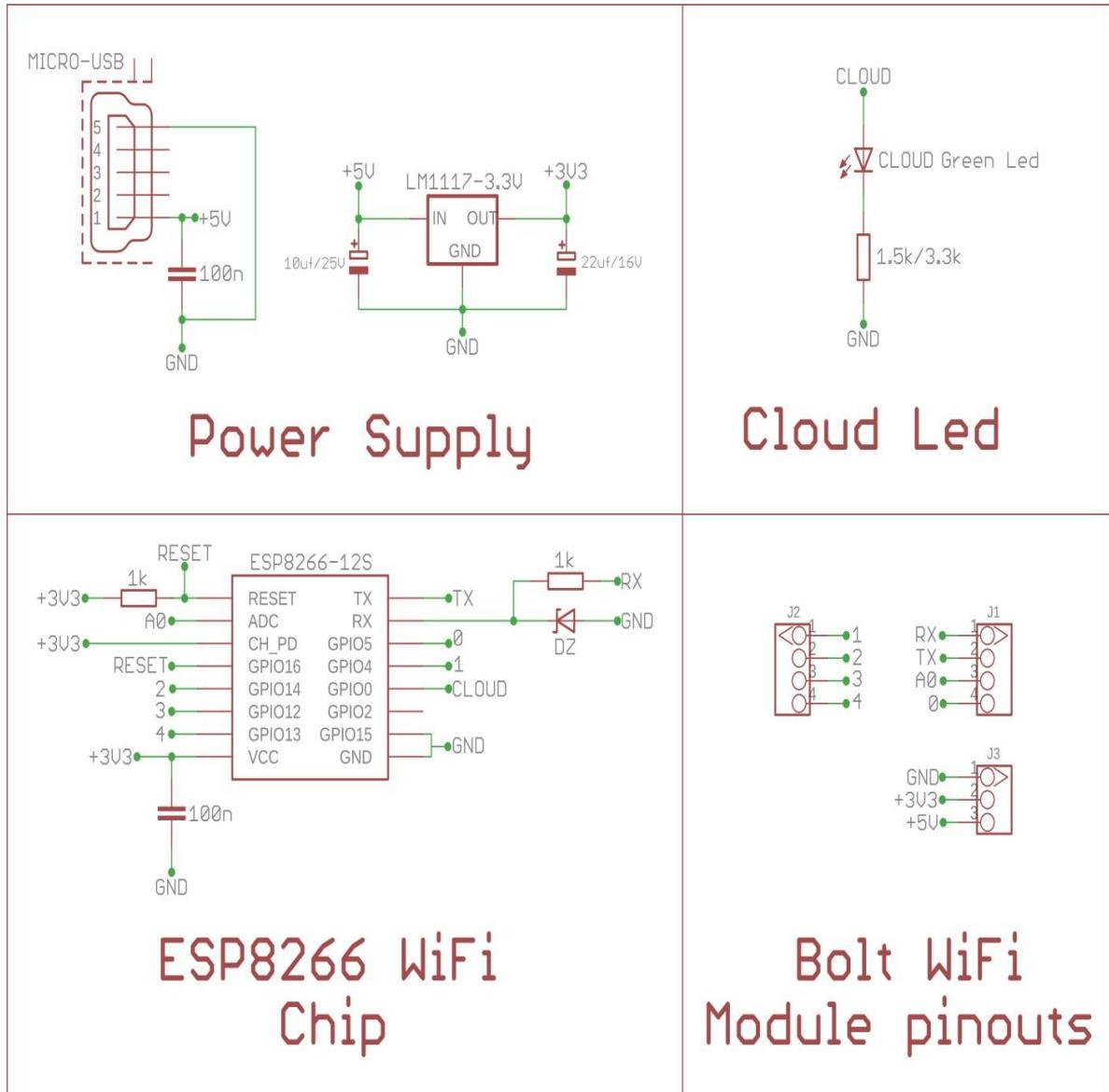
In our project we have used the BOLT WiFi Module.



The Internet of Things (IoT) is nothing new, it has been with us for over a decade, but in this time we have seen the price of devices fall from hundreds of dollars to less than \$10! The ESP8266 (and the newer ESP32) have really shaken the world of IoT from being pretty much a novelty industry (we all remember fridges that tweet and ovens that force us to accept an end user licence agreement) into a citizen science revolution where sensors across the world are monitoring climate change, animal migration patterns and much more.

WiFi Module with ESP8266-12E

ESP-12E is a miniature **Wi-Fi module** present in the market and is used for establishing a wireless network connection for microcontroller or processor. The core of ESP-12E is **ESP8266EX**, which is a high integration wireless SoC (System on Chip). It features ability to embed Wi-Fi capabilities to systems or to function as a standalone application. It is a low cost solution for developing IoT applications.



Pin Configuration

The ESP-12E module has twenty two pins and we will describe function of each pin below.

Pin	Name	Description
1	RST	Reset Pin of the module
2	ADC	Analog Input Pin for 10-bit ADC (0V to1V)
3	EN	Module Enable Pin (Active HIGH)
4	GPIO16	General Purpose Input Output Pin 16
5	GPIO14	General Purpose Input Output Pin 14
6	GPIO12	General Purpose Input Output Pin 12
7	GPIO13	General Purpose Input Output Pin 13
8	VDD	+3.3V Power Input
9	CSD	Chip selection Pin of SPI interface
10	MISO	MISO Pin of SPI interface
11	GPIO9	General Purpose Input Output Pin 9
12	GPIO10	General Purpose Input Output Pin 10
13	MOSI	MOSI Pin of SPI interface
14	SCLK	Clock Pin of SPI interface
15	GND	Ground Pin
16	GPIO15	General Purpose Input Output Pin 15
17	GPIO2	General Purpose Input Output Pin 2
18	GPIO0	General Purpose Input Output Pin 0
19	GPIO4	General Purpose Input Output Pin 4
20	GPIO5	General Purpose Input Output Pin 5
21	RXD0	UART0 RXD Pin
22	TXD0	UART0 TXD Pin

Features and Electrical Characteristics

- Wireless Standard: IEEE 802.11 b/g/n protocol
- Power Transmission:

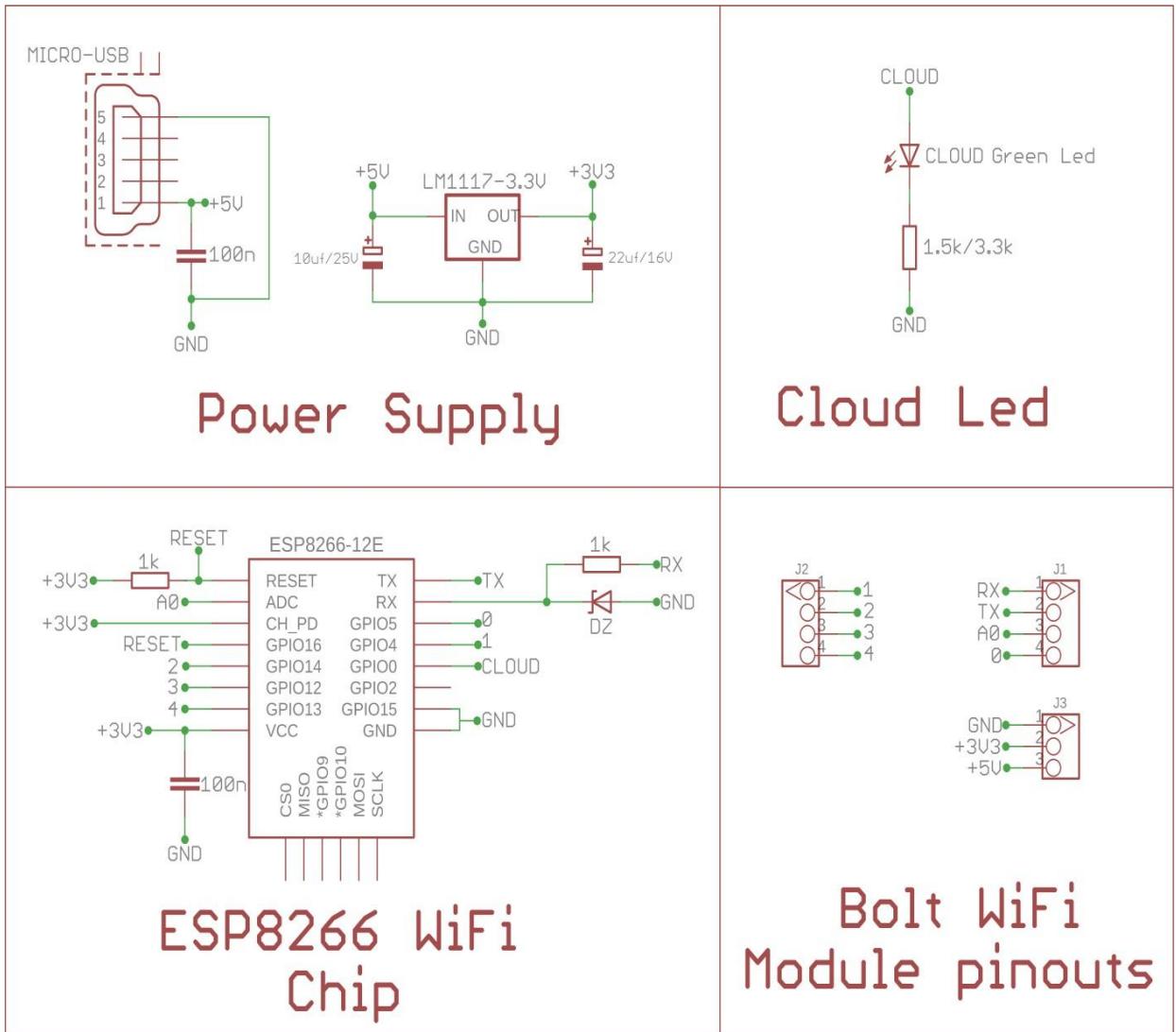
802.11b	+16 ± 2 dBm
802.11g	+14 ± 2 dBm
802.11n	+13 ± 2 dBm

- Frequency Range: 2.412 - 2.484 GHz
- Serial Transmission: 110 - 921600 bps, TCP Client 5
- SDIO 2.0, SPI and UART Interface available
- PWM available
- One ADC channel available
- Programmable GPIO available
- Wireless Network Type: STA / AP / STA + AP
- Security Type: WEP / WPA-PSK / WPA2-PSK
- Encryption Type: WEP64 / WEP128 / TKIP / AES
- Network Protocol: IPv4, TCP / UDP / FTP / HTTP
- Operating Voltage: 3.3V
- Maximum current allowed to draw per pin: 15mA
- Power down leakage current of < 10uA
- Integrated low power 32-bit MCU
- Onboard PCB Antenna
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW
- Operating Temperature: -40°C to +125 °C

Applications

- Weather station
- IoT applications
- Home appliances
- Toys and Gaming applications
- Wireless control systems
- Home automation
- Security ID tags

WiFi Module with ESP8266-12S



The ESP-12S is a WiFi module based on ESP8266, with built-in 32Mbit Flash, in the small SMD16 package. There're also onboard PCB antenna and metal shield. In short, it's a small form factor and fairly high cost effective WiFi module.

With the pre-flashed AT firmware, you can control the module through UART AT commands easily. The module integrates MCU inside, supports other developments like Lua/Micropython/Arduno to achieve flexible and fast prototyping, make it easy to integrate the module into IoT applications.

Features

- 802.11b/g/n Wi-Fi SOC
- Integrated Tensilica L106 ultra-low-power 32-bit micro MCU, run at 80 MHz and 160 MHz frequency, supports RTOS
- Built-in 1-ch 10 bit high precision ADC
- Supports UART/GPIO/ADC/PWM/I2C
- SMD16 package
- Integrated Wi-Fi MAC/ BB/RF/PA/LNA
- Supports multi sleep modes, deep sleep current as low as 20uA
- UART baudrate up to 4Mbps
- Embedded Lwip protocol stack
- Supports STA/AP/STA+AP operation mode
- Supports Smart Config (APP) / AirKiss (WeChat), one key networking
- Supports UART local firmware upgrade and/or remote firmware upgrade (FOTA)
- AT commands for getting started fast
- Supports secondary development, integrated Windows / Linux environment

Applications

The ESP-12S suit for IOT applications such as:

- Smart home
- Intelligent building
- Industrial automation
- Wearable devices
- IP camera
- Intelligent agriculture

Pin Configuration

Pin number	Pin name	function
1	RST	Reset module
2	ADC	A/d conversion result. Input voltage range 0~1V, value range: 0~1024
3	EN	Chip enable pin. Active high
4	GPIO16	GPIO16; can be used to wake up the chipset from deep sleep mode
5	GPIO14	GPIO14; HSPI_CLK
6	GPIO12	GPIO12; HSPI_MISO
7	GPIO13	GPIO13; HSPI_MOSI; UART0_CTS
8	VCC	3.3V power supply (VDD)
9	GND	GND
10	GPIO15	GPIO15; MTDO; HSPICS; UART0_RTS
11	GPIO2	GPIO2; UART1_TXD
12	GPIO0	GPIO0
13	GPIO4	GPIO4
14	GPIO5	GPIO5
15	RXD0	UART0_RXD; GPIO3
16	TXD0	UART0_TXD; GPIO1

General Specifications

Parameters	Details
Connectivity and Processing Module	ESP8266 with custom firmware
MCU	32-bit RISC CPU: Tensilica Xtensa LX106
Power	5V/1A DC via Micro-USB port or 5V and GND pins
Operating Voltage	3.3V
CPU Clock Frequency	80 MHz
MCU Internal Memory	64 KB of instruction RAM; 96KB of data RAM
MCU External Memory	4 MB Flash memory [QSPI]
GPIO pins	5 Digital pins [3.3V logic]
ADC	1 pin 10 bit ADC [0-1V input]
PWM	All 5 Digital pins capable of PWM [Software PWM]
Dimensions	35mm x 35mm
Boot Time	Less then 1 second

Connectivity

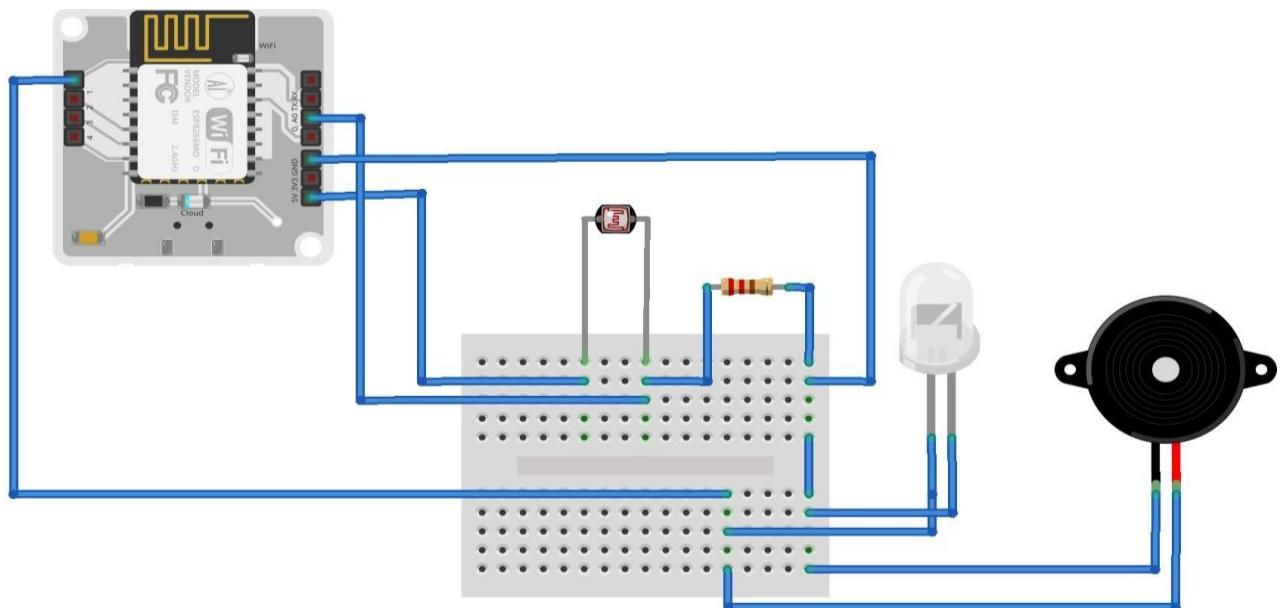
Parameter	Details
WiFi	802.11 b/g/n Automatic AP mode if not connected to WiFi WEP/WPA/WPA2 authentication Only works with 2.4 GHz WiFi
UART	8-N-1 3.3V TTL UART [using TX, RX, GND pins] [2400,4800, 9600,19200 baudrate]
Cloud	Default: Bolt Cloud (https://cloud.boltiot.com) Optional: Custom cloud using Bolt APIs

LED Indicators

Parameter	Details
WiFi LED - WiFi connectivity	1) Slow blinking: Trying to find and connect to WiFi network 2) Fast blinking: User has connected via Bolt IoT app for setup 3) Stable: Connected to WiFi
Cloud LED – Bolt Cloud connectivity	1) Stable: Connected to Bolt Cloud 2) Off: Not connected to Bolt Cloud 3) Dim: Insufficient power/ incorrect boot

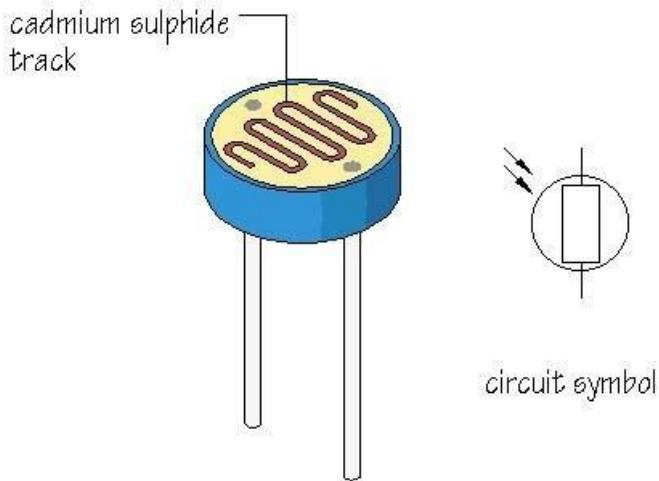
Circuit Connection

The system monitors light intensity and detects anomalous conditions.



Components of Circuit Diagram

LDR (Light Dependent Resistor)



Light dependent resistors, LDRs or photoresistors are often used in electronic circuit designs where it is necessary to detect the presence or the level of light.

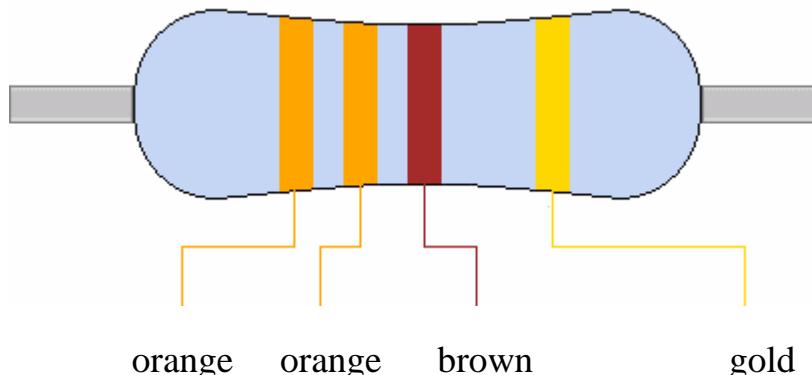
These electronic components can be described by a variety of names from light dependent resistor, LDR, photoresistor, or even photo cell, photocell or photoconductor.

Although other electronic components such as photodiodes or photo-transistor can also be used, LDRs or photo-resistors are a particularly convenient to use in many electronic circuit designs. They provide large change in resistance for changes in light level.

In view of their low cost, ease of manufacture, and their ease of use, LDRs have been used in a variety of different applications. At one time LDRs were used in photographic light meters, and even now they are still used in a variety of applications where it is necessary to detect light levels.

Light dependent resistors are widely available:- they are normally stocked by electronic component distributors, and in view of the way the electronics industry supply chain operates these days, this is the normal way to obtain them. Electronic component distributors large and small will typically have a good selection.

Resistor 330 ohm



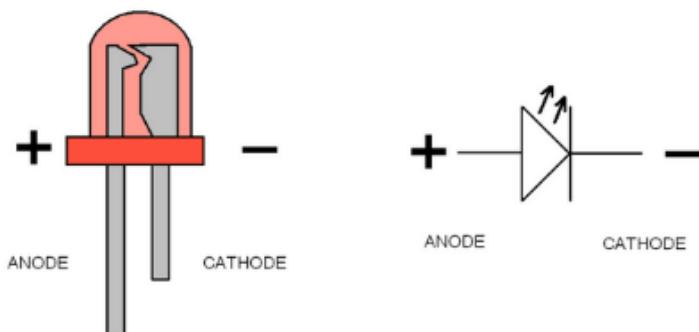
The resistor is a passive electrical component to create resistance in the flow of electric current. In almost all electrical networks and electronic circuits they can be found. The resistance is measured in ohms. An ohm is the resistance that occurs when a current of one ampere passes through a resistor with a one volt drop across its terminals. The current is proportional to the voltage across the terminal ends. This ratio is represented by Ohm's law:

$$R = V/I$$

Resistors are used for many purposes. A few examples include delimit electric current, voltage division, heat generation, matching and loading circuits, control gain, and fix time constants. They are commercially available with resistance values over a range of more than nine orders of magnitude. They can be used to as electric brakes to dissipate kinetic energy from trains, or be smaller than a square millimeter for electronics.

A passive device that resists the flow of electricity. This resistor will provide 330 Ohms of resistance wherever it is placed and will handle 1/4 watts. Use these low value resistors for voltage dividers and where you need to keep the current flow as high as possible.

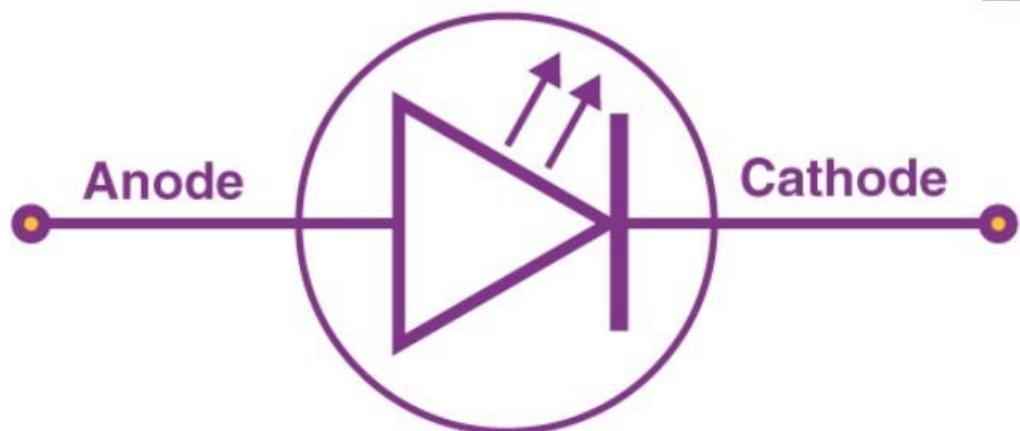
LED (Light Emitting Diode)



A light emitting diode is an electric component that emits light when the electric current flows through it. It is a light source based on semiconductors. When current passes through the LED, the electrons recombine with holes emitting light in the process. It is a specific type of diode having similar characteristics as the p-n junction diode. This means that an LED allows the flow of current in its forward direction while it blocks the flow in the reverse direction. Light-emitting diodes are built using a thin layer of heavily doped semiconductor material. Based on the semiconductor material used and the amount of doping, an LED will emit a colored light at a particular spectral wavelength when forward biased.

LED Symbol

Below is the demonstration of the LED symbol. The symbol is similar to that of the [p-n junction](#) diode. The difference between these two symbols is that the two arrows indicate that the diode is emitting the light.

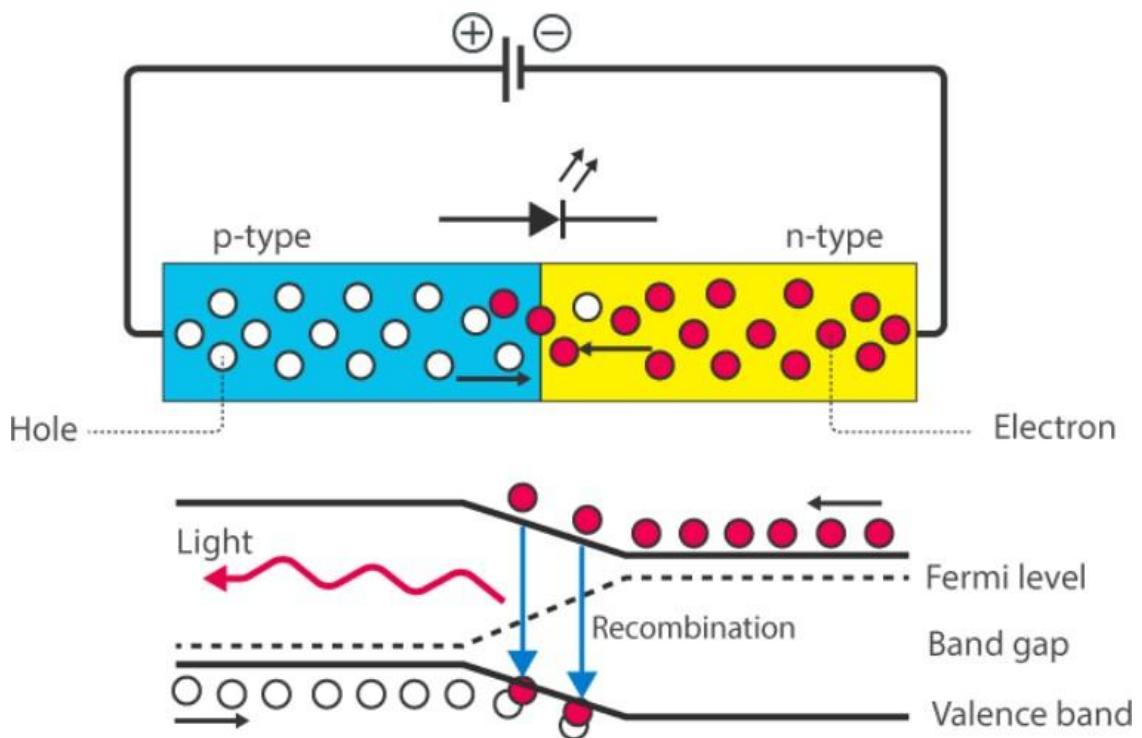


Working Principle of LED

The holes lie in the valence band, while the free electrons are in the conduction band. When there is a forward bias in the p-n junction, the electron which is a part of the n-type semiconductor material would overrun the p-n junction and join with the holes in the p-type semiconductor material. Therefore, regarding the holes, the free electrons would be at the higher energy bands.

When this movement of free electron and hole takes place, there is a change in the energy level as the voltage drops from the conduction band to the valance band. There is a release of energy due to the motion of the electron. In standard diodes, the release of energy in the manner of heat. But in LED the release of energy in the form of photons would emit light energy. The entire process is known as electro-luminescence, and the diodes are known as a light-emitting diode.

In LED, energy discharged in light form hinges on the forbidden energy gap. One could manipulate the wavelength of the light produced. Therefore, from its wavelength, the light color and its visibility or cannot be controlled. The color and wavelength of the light emitted can be determined by doping it with several impurities.

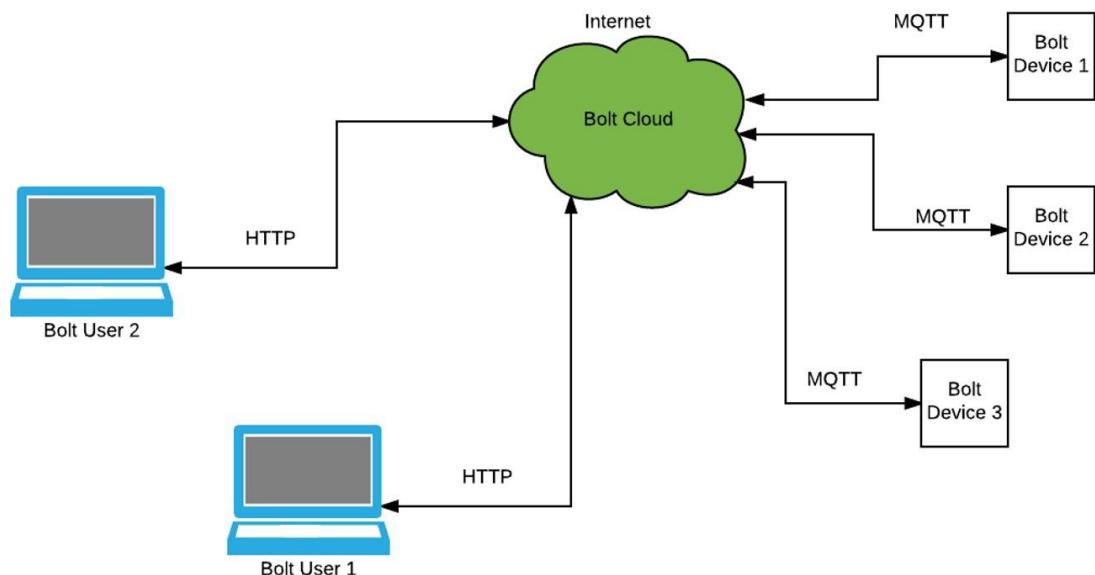


Understanding the IoT cloud architecture

In the previous lesson, we created our account on Bolt Cloud. Before we link our Bolt device to our account, let us understand the Bolt Cloud Architecture, how it works, and why it is important.

The Bolt Cloud is one of the major component in providing the IoT capabilities to the Bolt device. All the Bolt devices connect to the Bolt Cloud out of the box. The Bolt devices are shipped with a firmware that helps it understand how to connect to the Bolt Cloud over the Internet.

Communication chart of Bolt Devices with Bolt Cloud



The communication of Bolt devices with Bolt Cloud happens over the MQTT communication protocol. MQTT stands for Message Queue Telemetry Transport. But why do we need to have a protocol such as MQTT for communication when HTTP & HTTPS protocols are so popular and widely used for communication? Although these protocols are popular, the amount of overhead data that is sent over the Internet for managing the communication is quite a lot. Overhead data is the data which is sent along with the actual message/data which conveys the extra information required to understand the message/data sent. The overhead data varies from protocol to protocol. This is fine in case of systems such as mobile phones, laptops, desktop computers that have the hardware capabilities and the network capabilities to send the extra overhead data.

Most IoT devices and sensors contain limited processing capabilities and constrained Internet bandwidth. Due to these limitations, they send data over the Internet only when required and the data sent is very low in terms of bandwidth usage. Hence using

protocols such as HTTP, HTTPS does not become feasible where the overhead data is more than the actual data itself. MQTT contains very low overhead and hence becomes ideal for IoT communication.

MQTT is a pub-sub messaging protocol. Pub refers to publishing and sub refers to subscribing. There is a central entity, in our case it is the Bolt Cloud. All the Bolt devices connect to the Bolt Cloud and send the data to various channels by publishing the data on their unique channels. The Bolt devices also subscribe to channels so that they can receive commands coming from the Bolt users.

The Bolt Cloud users i.e. people like us communicate with the Bolt Cloud using the HTTPS communication protocol. We can use the Bolt Cloud dashboard to control and monitor our Bolt devices or use the Bolt Cloud APIs in case we want to by-pass the dashboard and access our Bolt devices via programs that we have written. We shall learn more about this in the next modules.

Bolt Cloud receives all the commands to control or request for sensor data from Bolt devices, and sends the commands to the Bolt device. The Bolt device executes the commands, and sends a response back to the Bolt Cloud which in turn forwards it to the user who initiated the command.

In this lesson, we understood the architecture of the Bolt Cloud, how the Bolt devices and users communicate with the Bolt Cloud. In the next lesson, we shall see how the Bolt Cloud works.

PROJECT IMPLEMENTATION

PHASE-1

Adding a new device

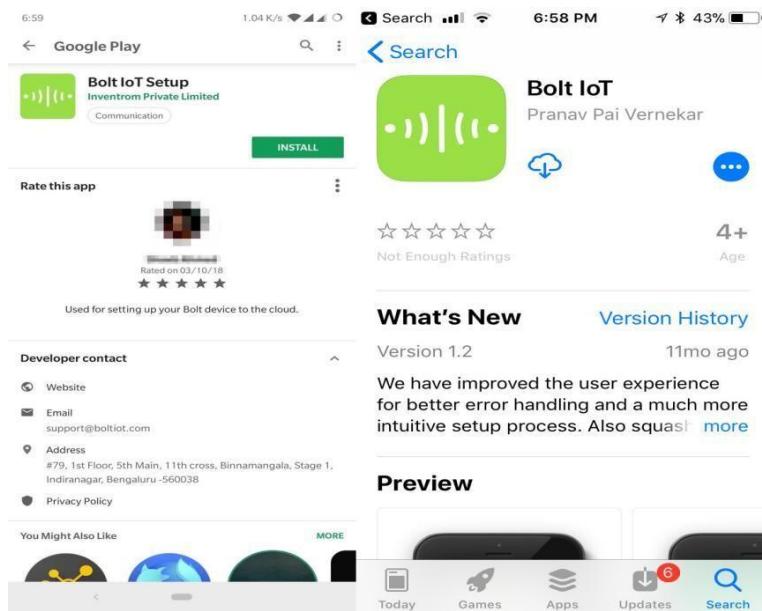
First, power on your device by connecting it to a micro-USB power supply cable. The micro-USB cable is commonly available in any hardware store and looks like this,



You can connect the regular end of the USB cable to your laptop or to a power adaptor which supplied a maximum of 5V and 1 Ampere.

Now that you have powered up the device, the Blue LED on your Bolt will start blinking slowly. This means that the Bolt is transmitting its own hotspot and is ready to be configured.

Now, Install the Bolt IoT app on your Android or iOS device by searching for "Boltiot" on the respective app store and installing it.



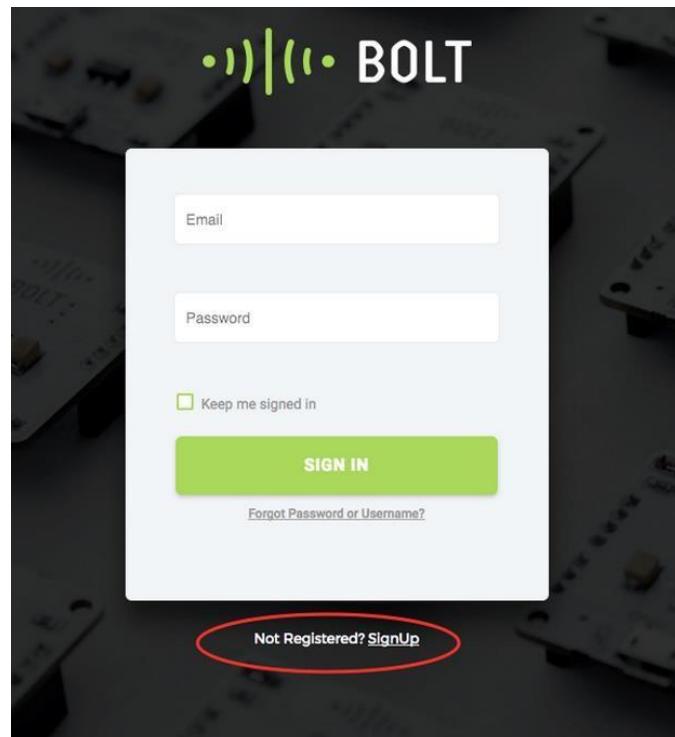
Creating an account on the IoT cloud service

An IoT system requires a hardware as well as a cloud. We looked at Hardware i.e. the Bolt WiFi module in the previous section. In this section we shall learn about the Bolt cloud.

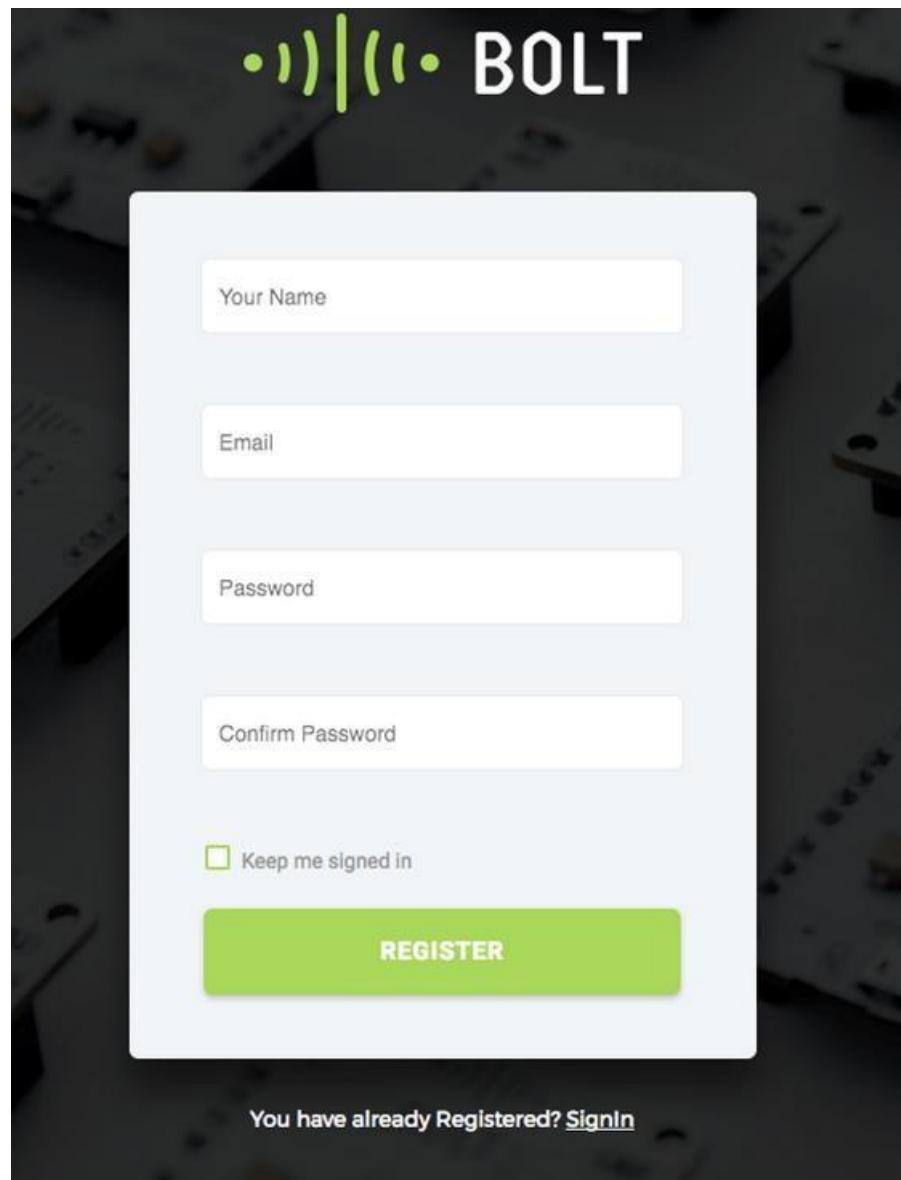
Bolt cloud is a server which lets you communicate with your Bolt WiFi module over the internet. It offers features like receiving and storing the data collected by Bolt Modules, Storing the data, Analyzing it via Data visualization and Machine Learning as well as it lets your program your Bolt modules. We shall learn about each of these features and how to use them more in detail in the workshop. However, before we get started, let us create an account on Bolt cloud. Follow the steps below:

Registration Process

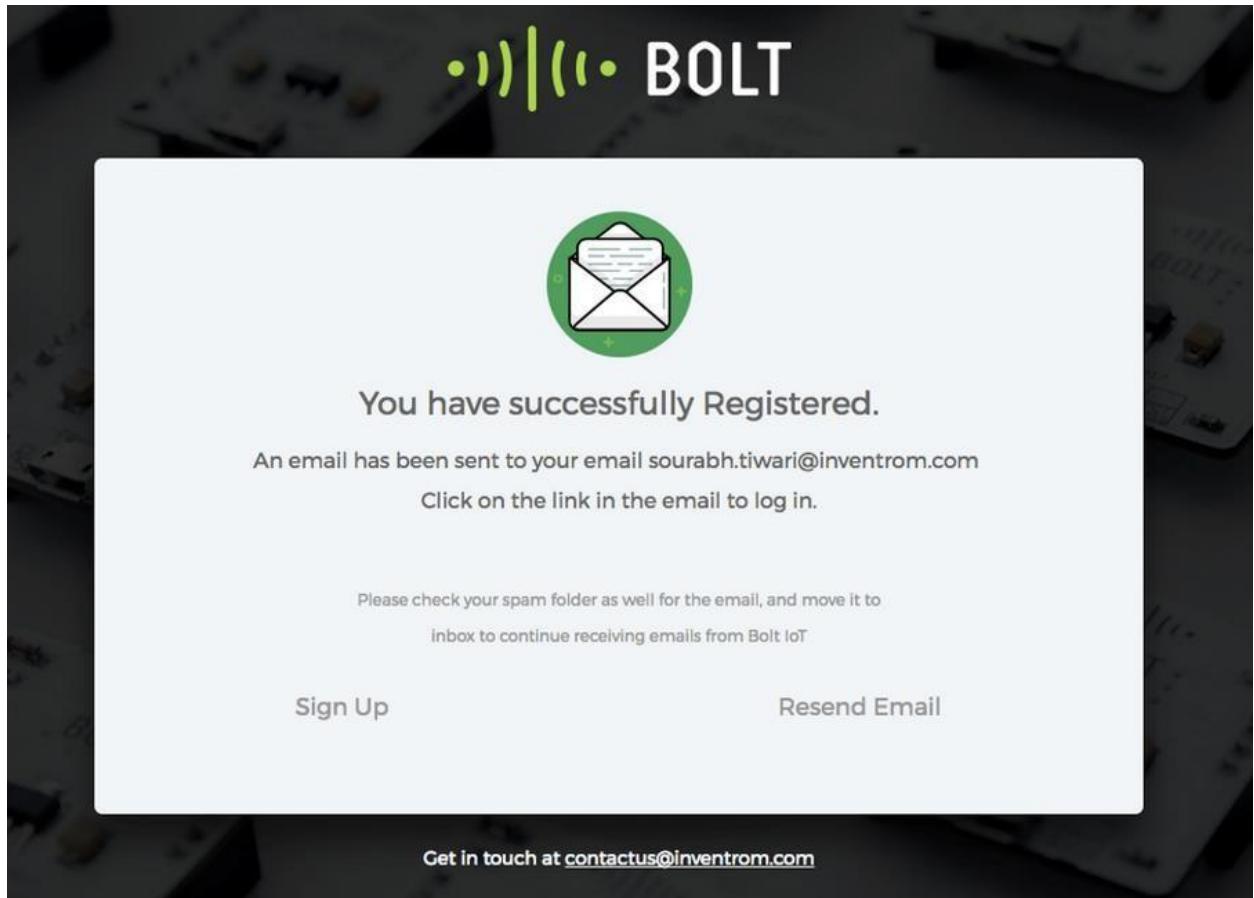
- Step 1: Open www.cloud.boltiot.com on your web browser (Google Chrome recommended)
- Step 2: Click on "SignUp" as shown below



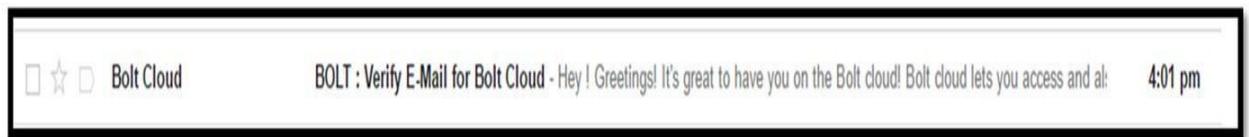
Step 3: Enter your details in the SignUp page shown below. Enter your name, email id (login ID for Bolt cloud),and password in the fields. Do confirm your password by typing it again in the Confirm Password field and then click on the Register button.



Step 4: Verification of your account: If you have entered all the details correctly, you will be successfully registered on Bolt Cloud and you shall see the screen below.



You will be sent a Verification Mail to your mail Id as seen below (Please check your Spam folder if the mail is not found in your Inbox).



Once you click on the button, your email id will be verified and you will be directed to the following page thus completing the registration process on Bolt Cloud.

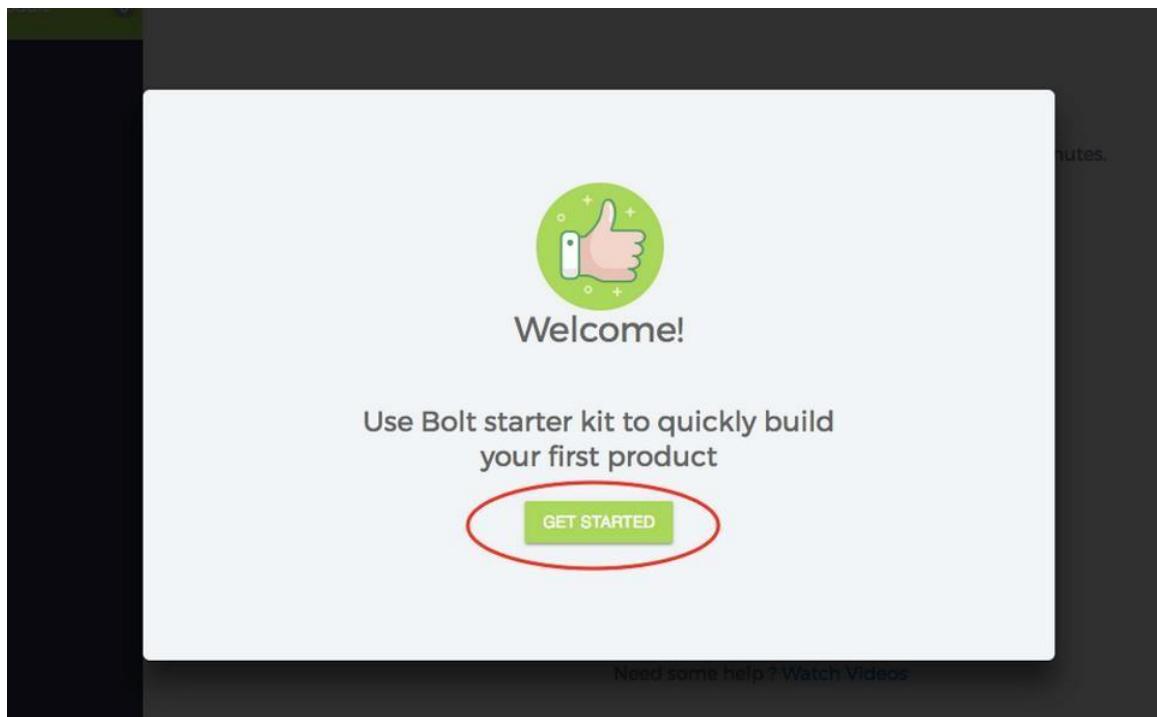


Your Email-id has been verified successfully

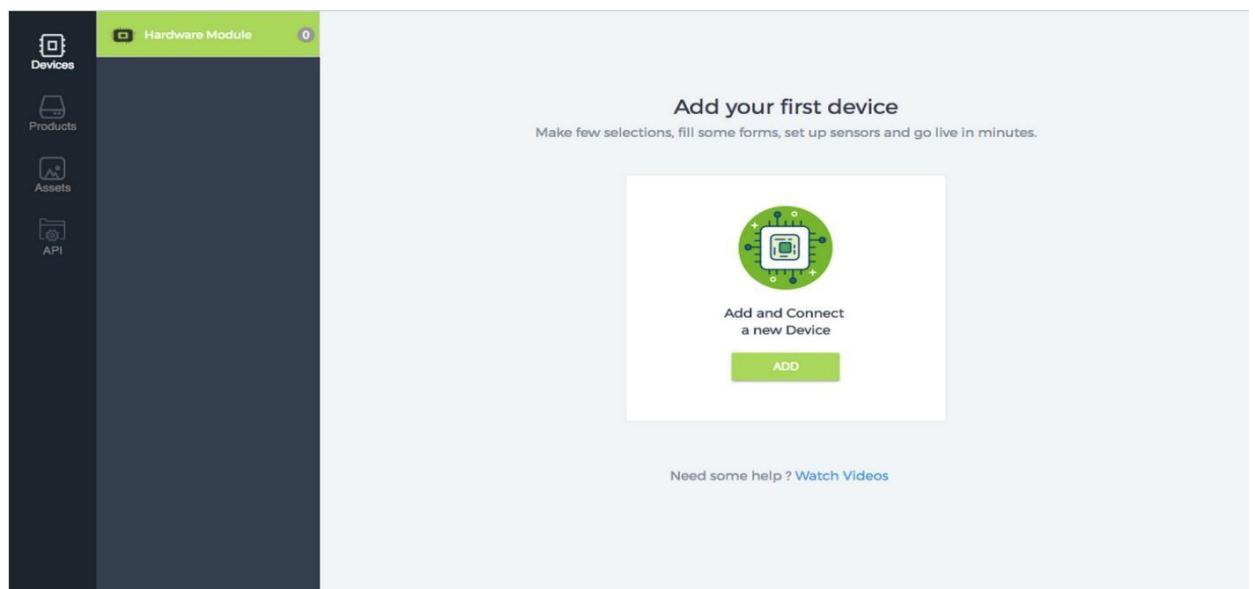
Redirecting ...

Get in touch at contactus@inventrom.com

- Step 5: You will be redirected to your Bolt Cloud Dashboard, with the Welcome message greeting, where you can click on the Getting Started button to view the guide.



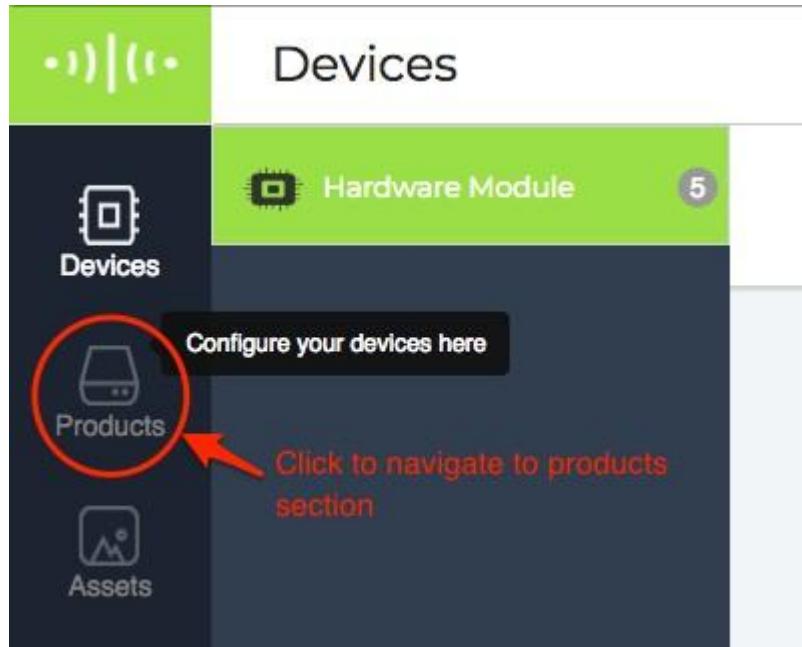
- Step 6: You will be given a basic tour of the Bolt Cloud features on the Dashboard. You can click on the Next & Back button to view the features. You can click the Skip button to skip the tour if you wish to. The tour can be taken anytime by clicking the “Take a Tour” button at the top right corner on your dashboard.
- Step 7: Once the tour is done you shall have the option to add a Bolt device to your dashboard. The process of linking your Bolt device to your Cloud Dashboard will be covered in the next lesson.



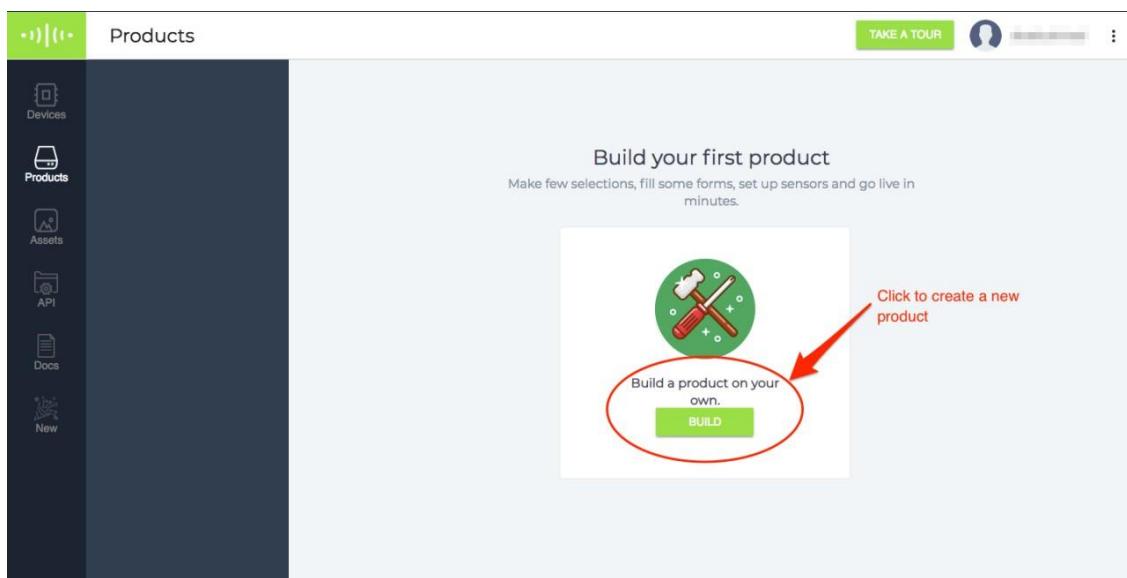
Creating a new product

Products control the behaviour of your Bolt device. You can configure a product to be of certain types. It is also possible to link multiple Bolt devices to a single product but a single Bolt device cannot have multiple products linked to it.

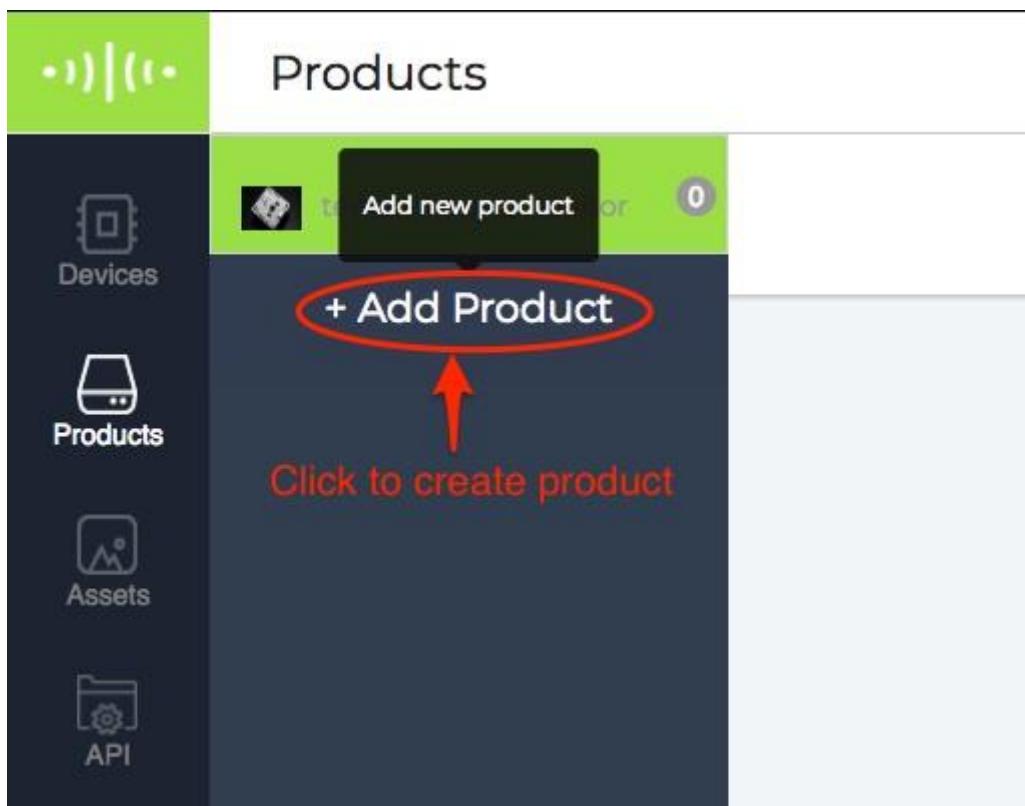
To create a new product, click on the Products tab on your Cloud Dashboard



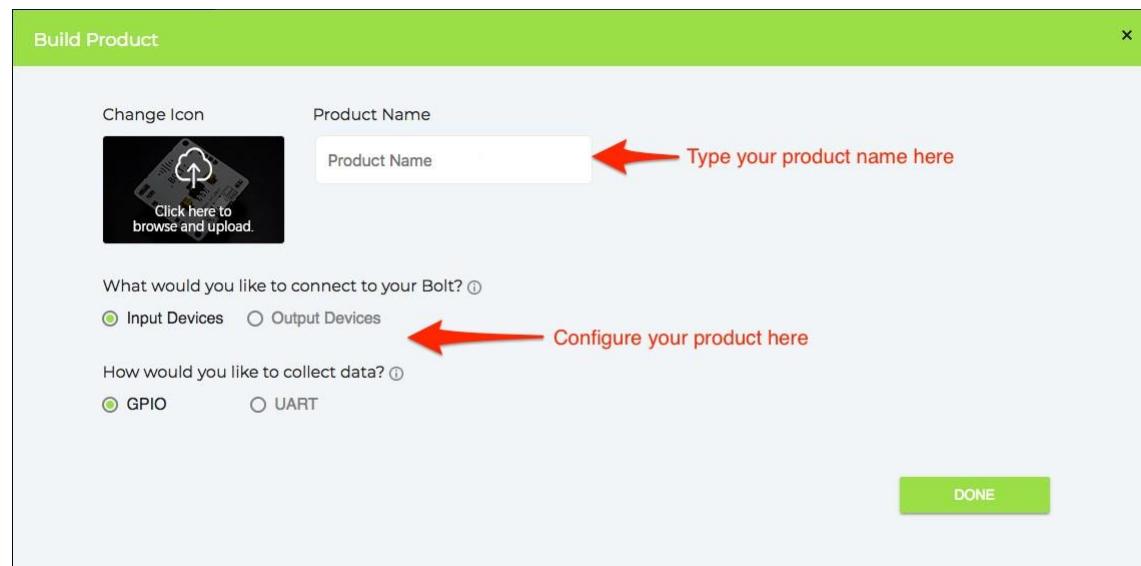
If you are creating your first product, you can click on the "Build" button as shown below.



If you already have created a product and want to create more, you can click on the "Add Product" button next to your product list.



Configure the product in the popup as per your liking.

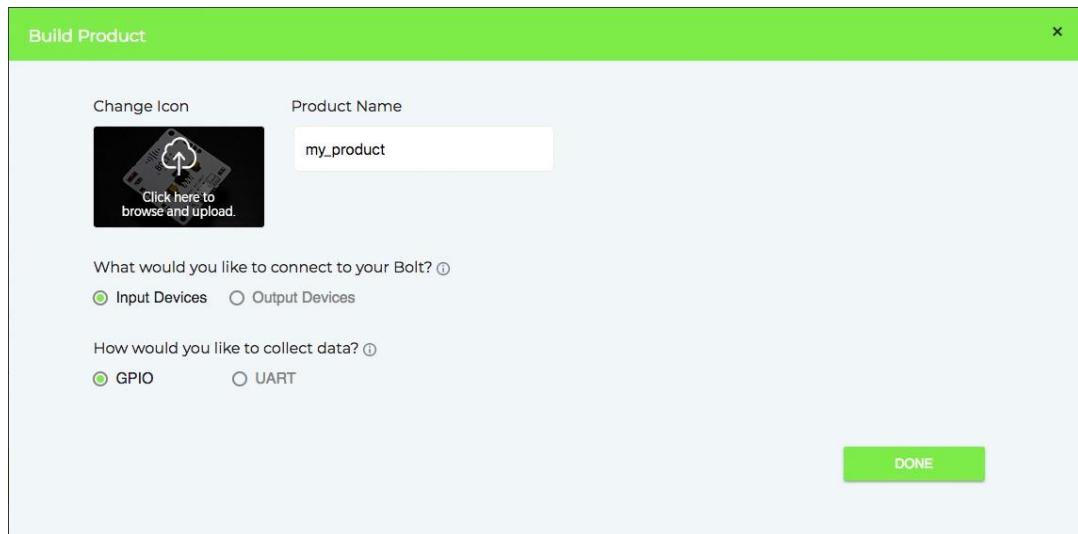


Types of products

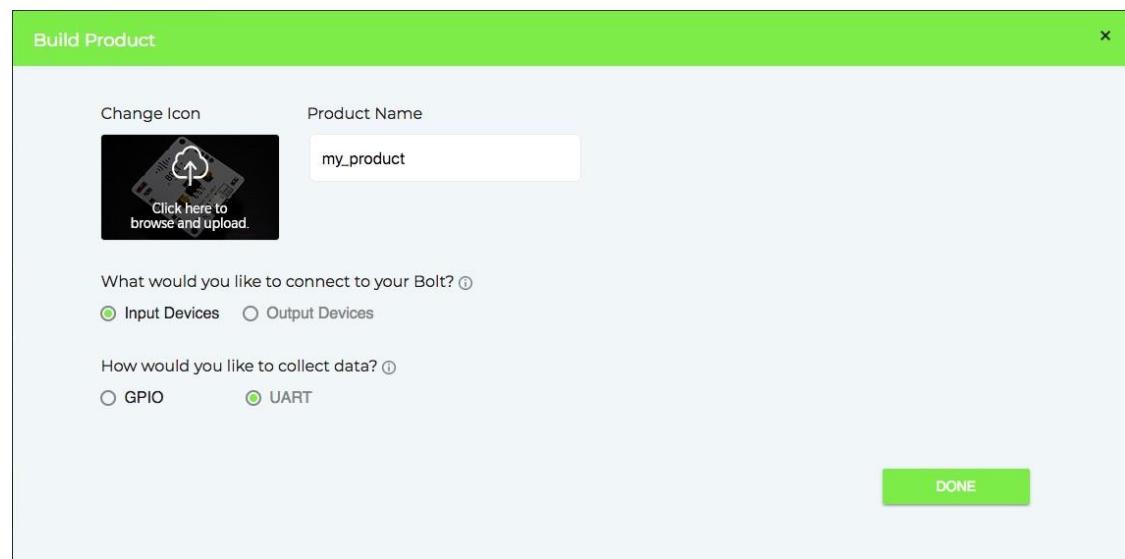
The Bolt Cloud can have three types of products. You can link your Bolt device to a product to be one of three types, i.e.

#Input device

In this type of product, the linked Bolt devices will act as an Input device, i.e. gather data from the connected sensors and send it to the Bolt Cloud at the set interval.



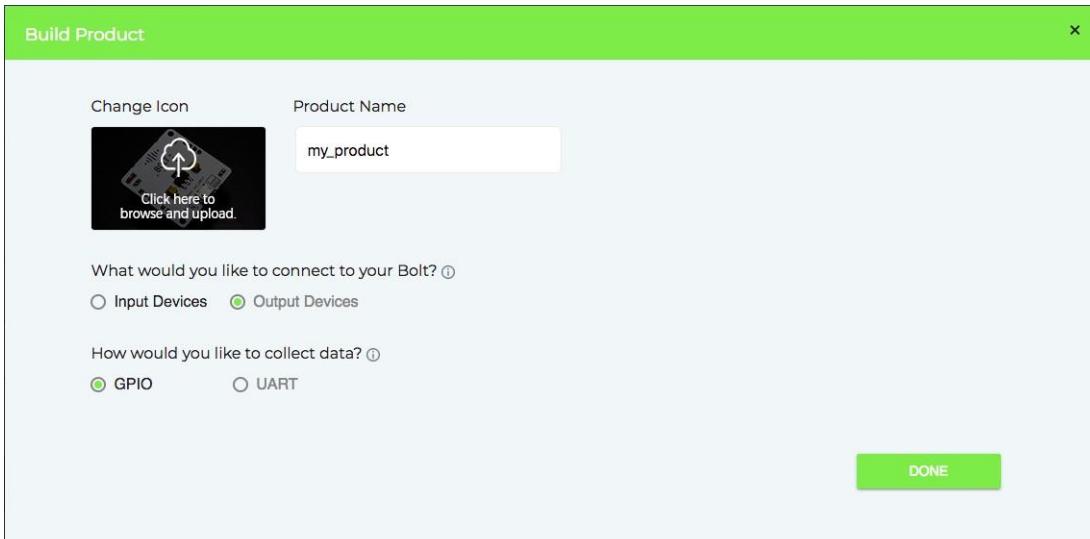
Select GPIO if you're connecting sensors to the Bolt device



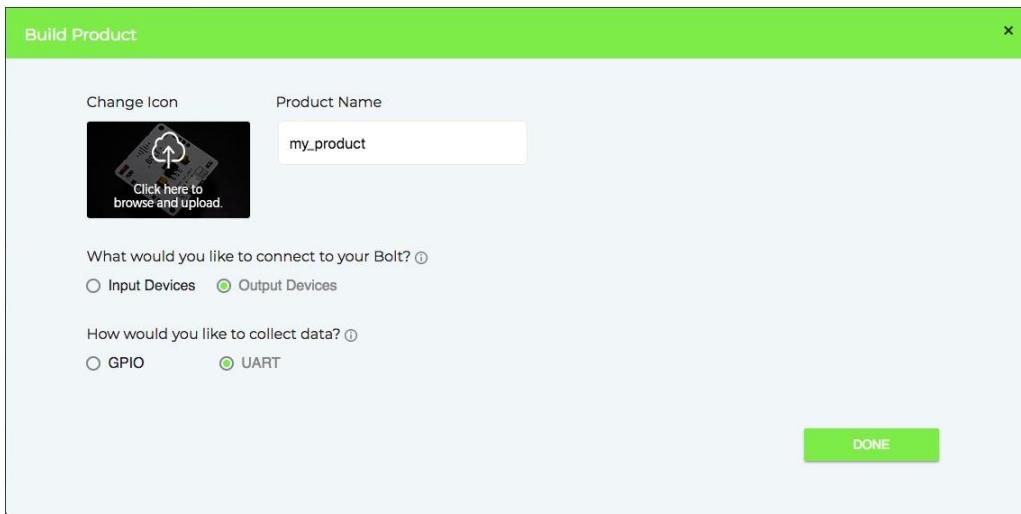
Select UART if you're connecting microcontrollers to the Bolt device

Output device

In this type of product, the linked Bolt devices will act as an Output device, i.e. control actuators connected to the Bolt device.



Select GPIO if you're connecting sensors to the Bolt device



Select UART if you're connecting microcontrollers to the Bolt device

Input and Output device

This type of product enables your device to both send and receive data from the Bolt cloud. Use this type of product when you want the devices linked to the product to control any actuator while at the same time send data to the cloud.

Selecting the product as an Input device will allow your Bolt device to both send and receive data to the Cloud.

Configuring product hardware

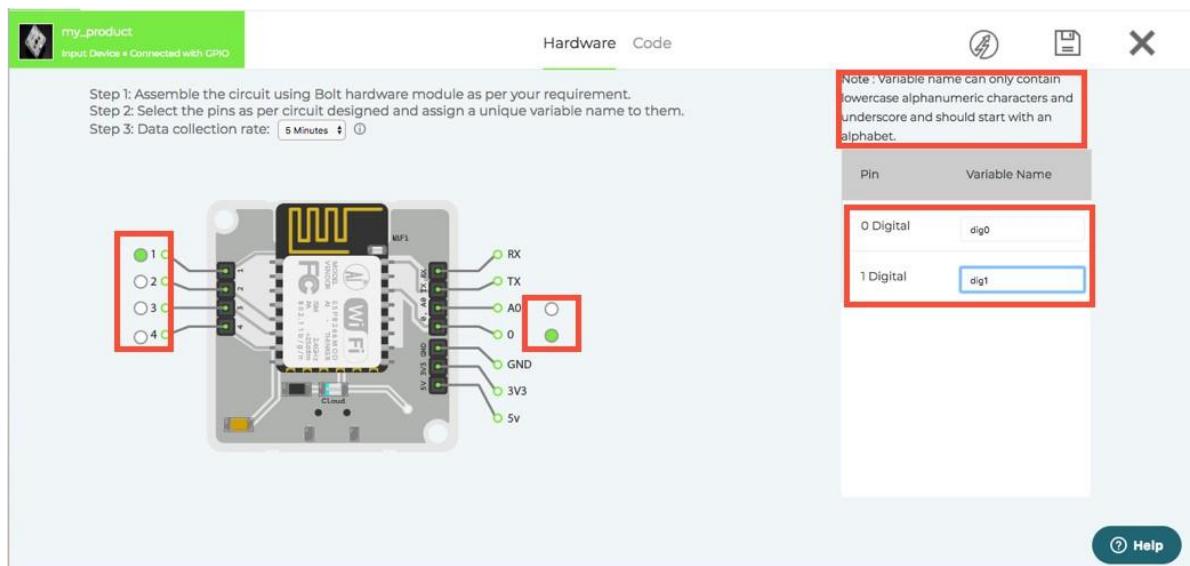
Once you have created a product, it will be listed on the product tab in the Cloud dashboard.

To configure the product, select it in the devices tab, and click on the configure button.

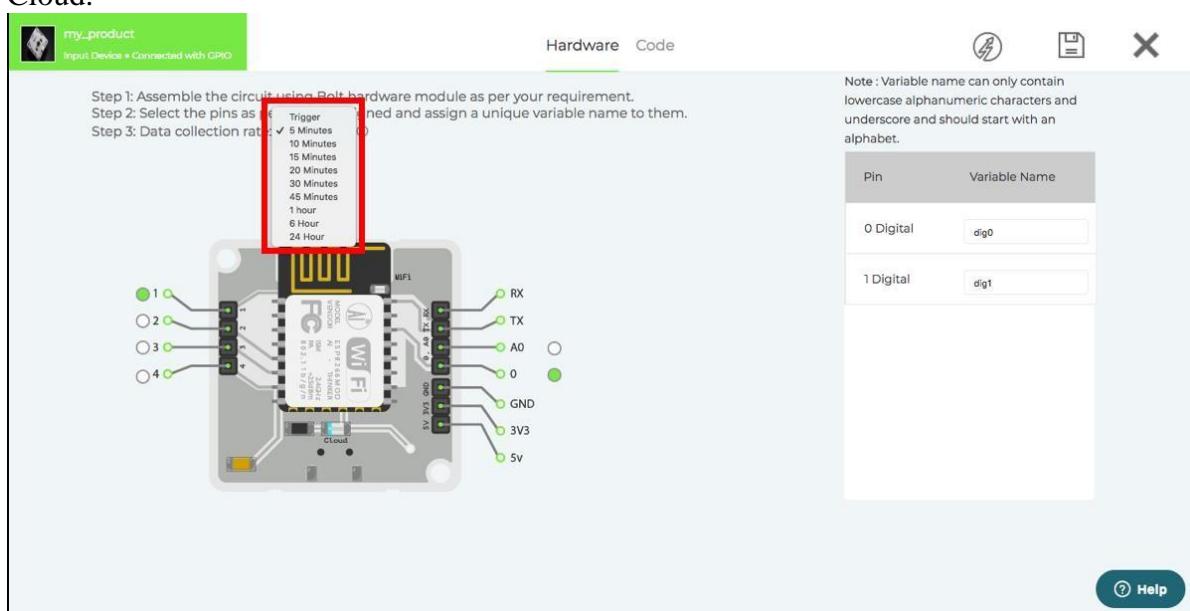


Click on the "More Info" icon and then click on the "Share Device" icon

You will be redirected to the product's hardware configuration page. If you created the product to collect data from the GPIO, then you will be able to configure which GPIO pin to monitor. Selecting a GPIO pin will allow you to assign a specific name to the pin which you can then use in the code configuration. Remember to adhere to the rules while assigning a name to the pin.



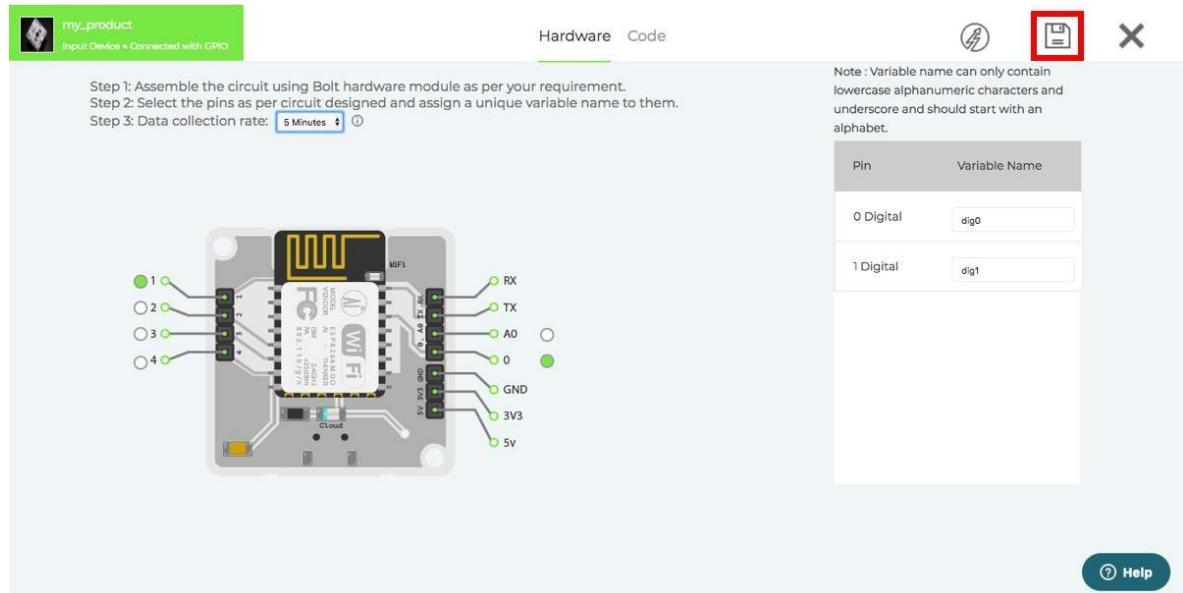
Once you have assigned appropriate names to the pins, select the push data rate. This is the time difference between 2 consequent reading that the device will send to the Cloud.



Selecting "Trigger", will inform the Bolt to push the data to the Cloud, everytime a configured pin on the Bolt changes states from high to low, or low to high. This rate allows you to update the state of the pins to the Bolt Cloud once in every 4 seconds.

NOTE: The trigger feature is only available for Digital GPIO pins. It cannot be applied to the Analog input pins.

You can then save the product hardware configurations by clicking on the save button and selecting yes.



The product hardware configurations will be saved. You can always come back and modify the hardware configurations by deleting the earlier hardware configurations.



my_product

Input Device • Connected with GRC

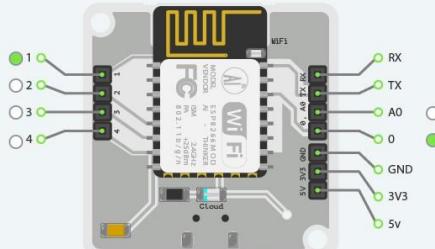
Hardware Code



- Step 1: Assemble the circuit using Bolt hardware module as per your requirement.
Step 2: Select the pins as per circuit designed and assign a unique variable name to them.
Step 3: Data collection rate: 5 Minutes

Note : Variable name can only contain lowercase alphanumeric characters and underscore and should start with an alphabet.

Pin	Variable Name
0 Digital	dig0
1 Digital	dig1

[DELETE CONFIGURATION](#)

Help

PROJECT IMPLEMENTATION

PHASE-2 (COMPLETE)

Setting Up the Bolt WiFi Module



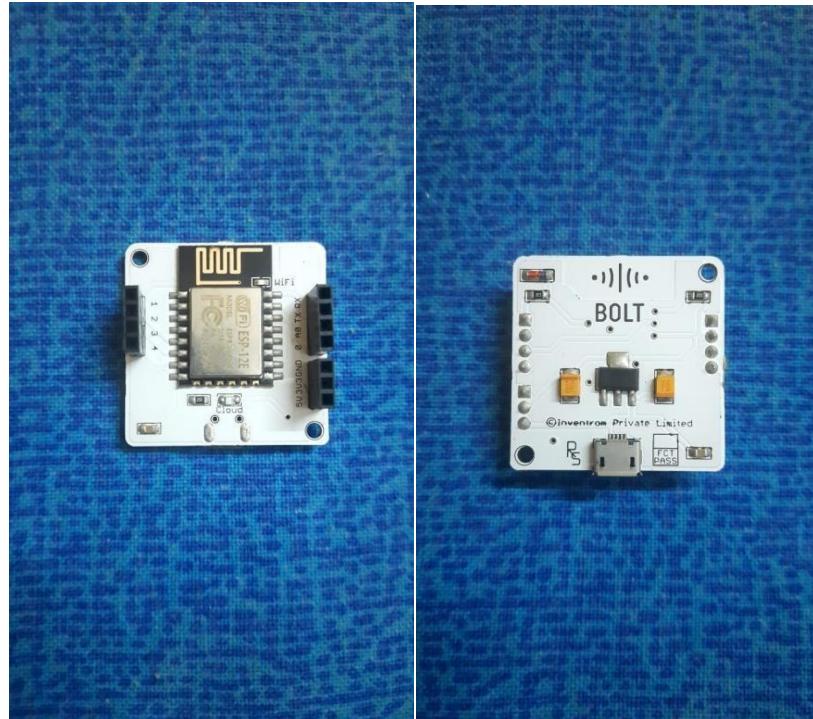
Things used in this project

- Bolt IoT Bolt WiFi Module
- Micro-USB to USB Cable (Generic)
- Bolt IoT Bolt Cloud
- Bolt IoT Android App

Story

This is a handy getting started guide for connecting your Bolt module to the WiFi and linking it to your Bolt cloud account so that you can start playing with it.

The whole process will take less than 10 minutes. The current record for setting up the Bolt device is 2 minutes 15 seconds.



Get a 5V 1A mobile charger which has a micro-usb port and is usually used to charge your Android mobile. You can also use your laptop to power on the Bolt device.



Now that you have all the components, let's start with downloading the Bolt IoT mobile App and installing it in the mobile phone.

Note: Before you get started, check if the WiFi network to which you will be connecting Bolt works at 2.4 GHz. Bolt does NOT support 5 GHz WiFi Networks at the moment.

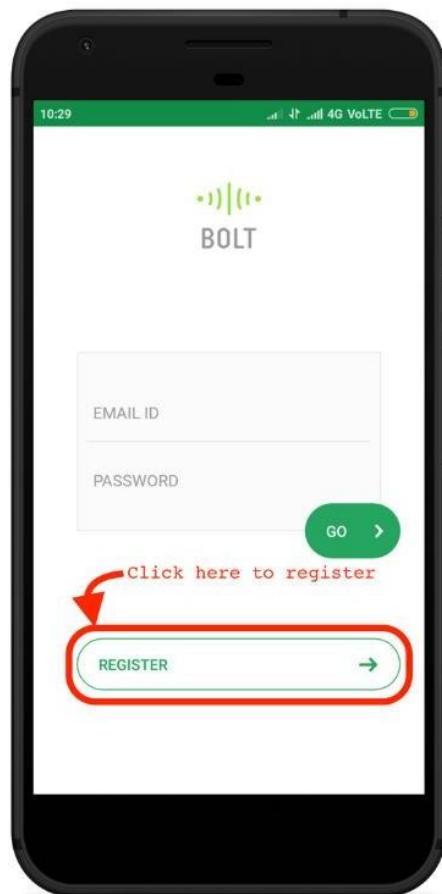
Step 1: Down

Download the 'Bolt IoT' App for Android or iOS. You can either search for 'Bolt IoT'

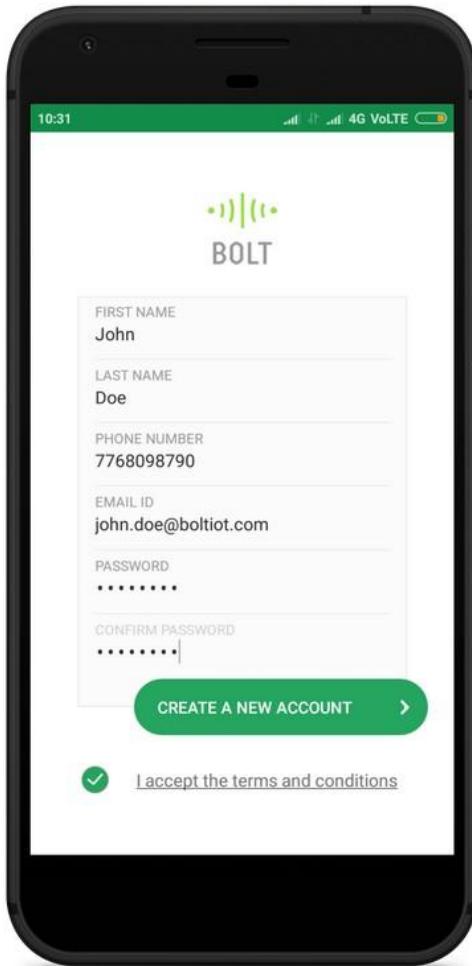
Step 2: Creating an account

You will need to create an account on the Bolt Cloud to control the Bolt device. Did I mention that creating an account on the Bolt Cloud is FREE?

Open the Bolt App and click on 'REGISTER' to create a new account.

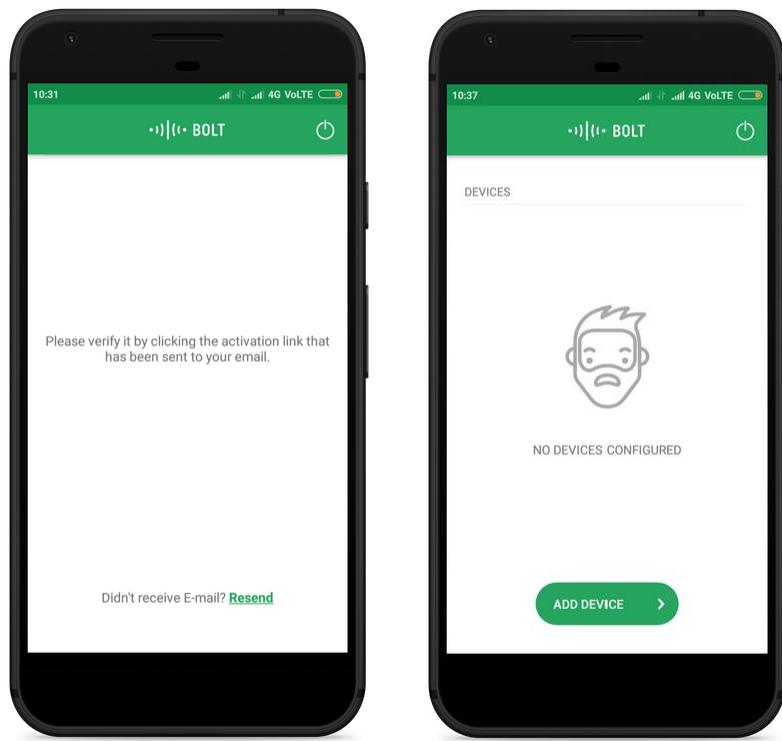


Fill all the required details and agree to the terms and conditions. Finally, click on 'CREATE A NEW ACCOUNT' to create your free account.



You will now receive an email on the email ID that you had given to verify your account. This is to verify that you are an actual human and not a robot.

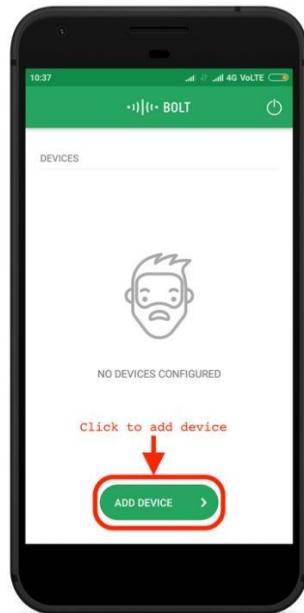
Follow the instructions in the email to activate your Bolt Cloud account. Now, pull down on the screen using your fingers to refresh the mobile App screen which looks something like this below,



Congratulations, you have activated your account. You will then be greeted with a screen that says that you don't have any Bolts Modules linked to your account.

Step 3: Adding Bolt to your account

Now, click on the 'ADD DEVICE' button to setup your Bolt device with your WiFi network.



Power ON the Bolt device by inserting the Micro USB cable into the USB port provided on the Bolt and connecting the other end of the cable to the Android charger or to your laptop.

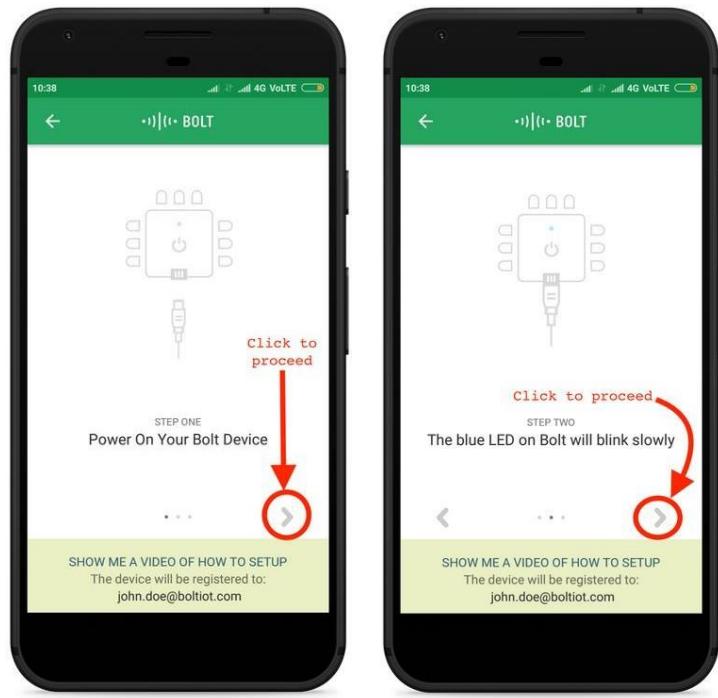
NOTE: There are no ON/OFF buttons on Bolt. Once you power it ON, it is good to go.



Once you have powered ON the Bolt device, blue LED on the Bolt will begin to blink slowly and the green LED will be OFF.

When the blue LED blinks slowly, it means that the Bolt is now transmitting its own WiFi hotspot network and is ready to be setup via the Bolt IoT App on your phone.

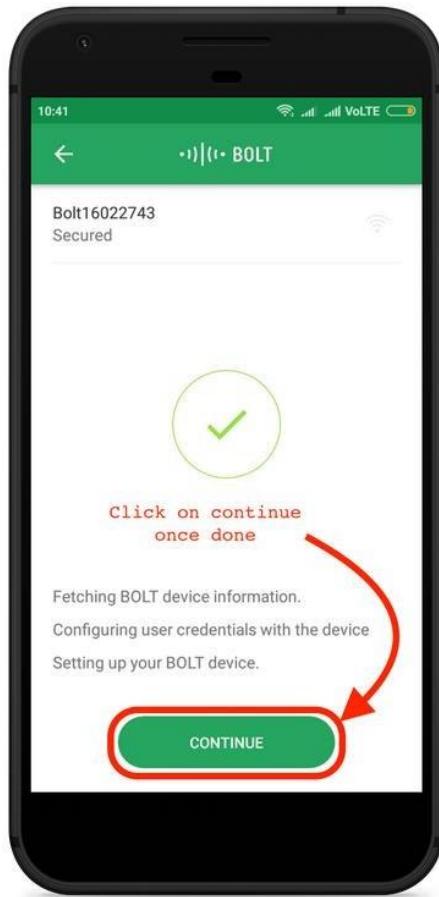
Click on the '>' symbol on the app to progress further.



For Android Only: Before proceeding further, kindly switch OFF your mobile data and switch ON the location service in your mobile if not already and click on the

"READY" button. This is required only in the Android App due to APIs by Google. We do not collect or store any of your location data.

If the Bolt has been detected by the App, it will show a screen similar to the one below. The blue LED on the Bolt will now blink fast. This means that it is under the setup process. To proceed, click on 'CONTINUE'.



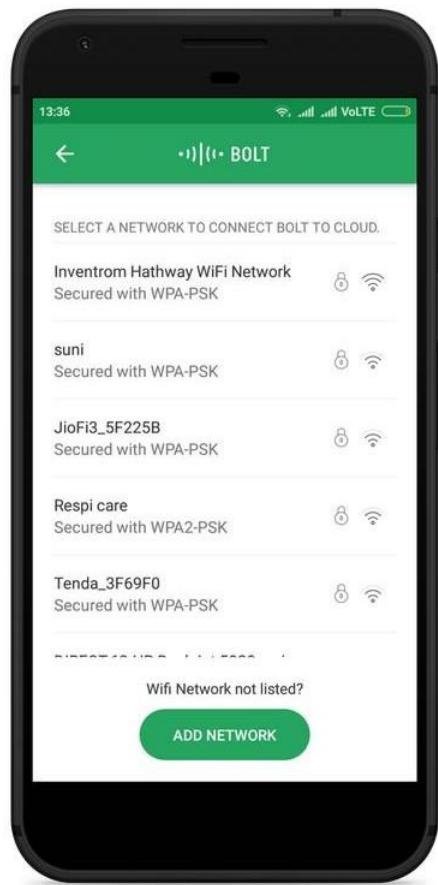
The password for the WiFi hotspot transmitted by the Bolt WiFi Module is: **bolt1234** While setting up with the Android mobile app, your mobile phone will automatically connect to the hotspot when you click next. On an iOS, you will have to connect to the Hotspot by going to your Setting -> WiFi.

Now, we will need to tell the Bolt which WiFi network it has to connect to.

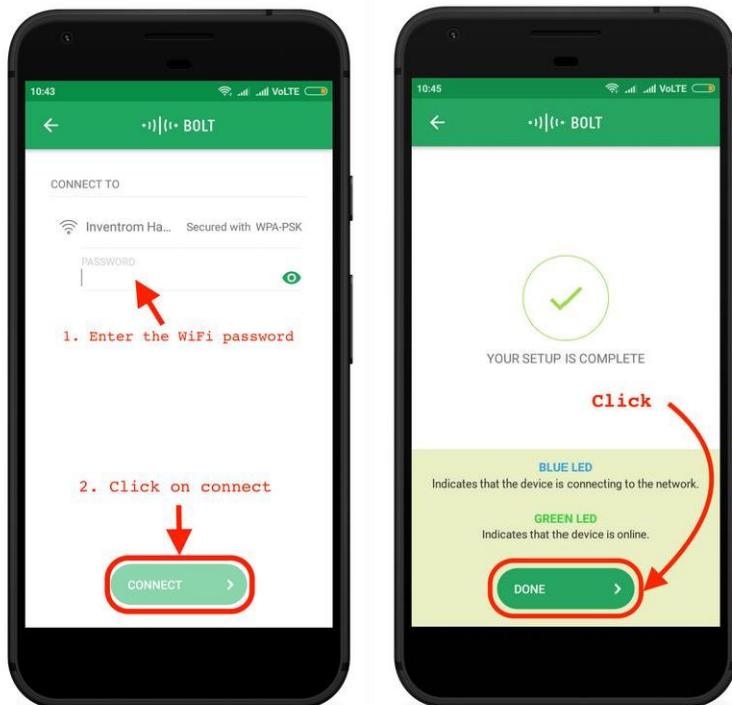
NOTE: The Bolt WiFi module cannot detect 5 GHz based WiFi networks and will not be able to connect to it.

Select the WiFi network to which Bolt has to be connected. You can click on the WiFi name to choose the WiFi network to connect to.

Note: For the iOS App, you will have to manually enter the SSID or the WiFi Name of the device you want to connect to.

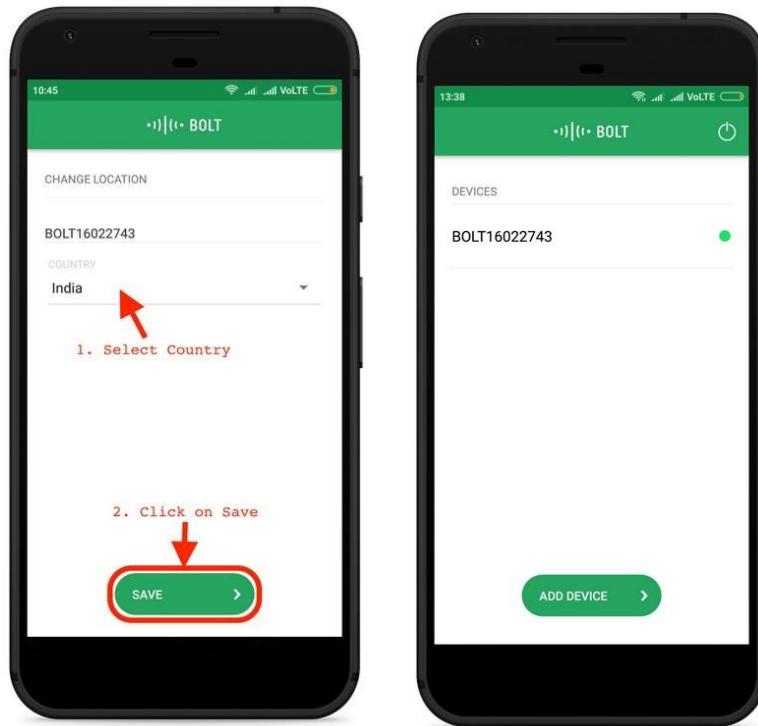


Enter the password of the WiFi network you had chosen in the previous screen.



The App will now send the WiFi credentials to the Bolt WiFi Module.

Now, click on 'DONE' and in the next screen, select your Country and click on 'SAVE' to complete the final step of the setup.



If everything has gone according to plan, the Bolt device will now restart automatically. The blue and green LED on the Bolt device will now be glowing steadily.

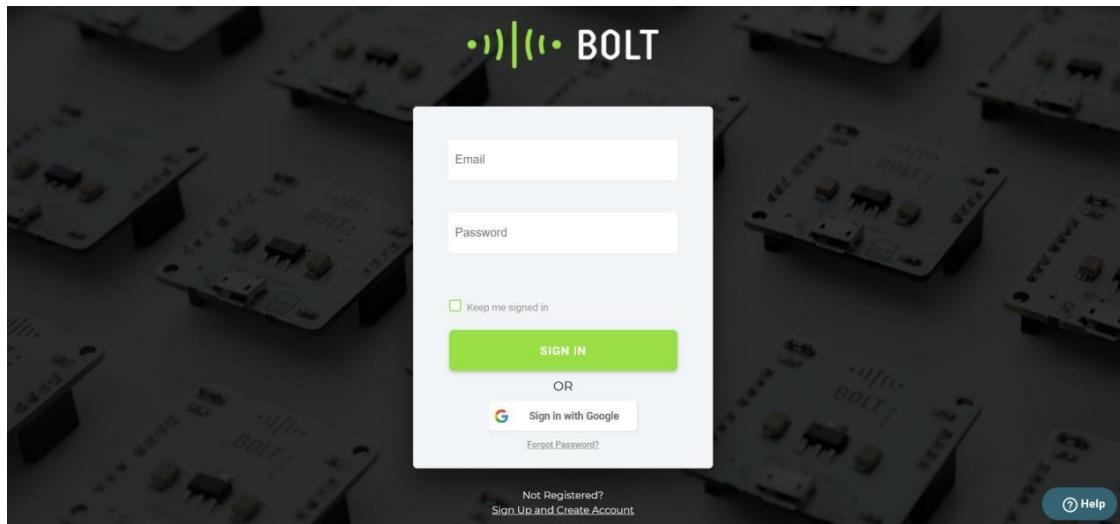
If the Bolt was able to connect to the WiFi network and is connected to the cloud, a green dot will appear beside the Bolt's device ID as shown below.

Debugging:

- If the Bolt is unable to connect to the WiFi network, the green LED will be OFF and the Blue LED will be blinking slowly. This generally happens if you have entered the wrong WiFi credentials (SSID or password) at the time of setup. Please try the setup process once again with the correct WiFi credentials.
- Check if your WiFi Router works at 2.4GHz. Bolt does not support 5GHz at the moment.

Step 4: Accessing the Bolt Cloud to Build IoT Projects

Visit [Bolt Cloud \(cloud.boltiot.com\)](https://cloud.boltiot.com) and login into your registered account using the same email ID and password that you have used on the Bolt IoT mobile App.



You will see your Bolt device with status as 'ONLINE' on your account on the dashboard.

A screenshot of the Bolt Cloud dashboard. On the left is a dark sidebar with icons for Devices, Products, Alerts, Assets, API, Docs, Project Ideas, and Upgrade. The main area shows a table with one row of data. The table has columns for ID, STATUS, PRODUCT, and ACTIONS. The first column contains the ID 'BOLT14917370' with a copy icon. The second column shows the status as 'OFFLINE' with a red dot and the text 'Not Linked'. The third column has an icon for linking. The fourth column contains icons for edit, delete, and more options. A green 'ADD NEW DEVICE' button is located at the top right of the main table area. A 'Help' button is also visible in the bottom right corner.

Congratulations, you have managed to setup the Bolt and link it to your account in less than 10 minutes.

Before we move forward

Important terms

1. Introduction to API

The Bolt Cloud API provides an interface for communication between the Bolt devices and any 3rd party system e.g. mobile application, web server, python programs etc. The API contains very intuitive control, monitoring, communication and utility functions for the Bolt Devices connected to your account. The Bolt Cloud API uses HTTP protocol for the communication and uses the HTTP GET and HTTP POST methods. Hence users can execute actions and retrieve information from Bolt devices programmatically using conventional HTTP requests.

Here are a few use cases of the API:

- Use the API in native apps on iOS and Android to control and monitor Bolt devices over the Internet.
- Pull collect sensor data connect to Bolt device, to any other cloud to run your custom AI algorithms and analytics.
- Connect Bolt Cloud to any VPS (Virtual Private Server) and run your code in any language of your choice. Refer sample codes.
- Remote Operating System: Using the API, Bolt devices can work like a board with an OS i.e. similar to Raspberry Pi or Beagle Bone, with the exception of the OS, which in this case, will reside on a remote VPS (Virtual Private Server). The Bolt will receive data from the sensors and push to the VPS with a Linux OS. The processing will take place on the VPS and it will push the commands to control motors, LEDs, and actuators to the Bolt device. You can use all the features of a Linux OS in this kind of a system.

API Request

This is a simple syntax for making an API request. In this request, the first parameter is the API key and the second parameter is the command that we want to execute and the last parameter is ID of Bolt device. The general syntax for the API request is:

https://cloud.boltiot.com/remote/your_api_key/command|m1=...|m2=...&deviceName=BOLTXXXXXX

The below example typically shows how to "restart" a Bolt device having device name as BOLT916154:

<https://cloud.boltiot.com/remote/44b2de6b-7e68-40e7-a27f-814b58afe008/restart?&deviceName=BOLT9161541>

2. Introduction to Anomaly

Anomaly detection plays an instrumental role in robust distributed software systems. Anomaly detection can:

- Enhance communication around system behavior
- Improve root cause analysis
- Reduce threats to the software ecosystem

Traditional anomaly detection is manual. However, machine learning techniques are improving the success of anomaly detectors. Of course, with anything machine learning, there are upstart costs—data requirements and engineering talent.

Overview

Anomaly detection is any process that finds the outliers of a dataset; those items that don't belong. These anomalies might point to unusual network traffic, uncover a sensor on the fritz, or simply identify data for cleaning, before analysis.

In today's world of distributed systems, managing and monitoring the system's performance is a chore—albeit a necessary chore. With hundreds or thousands of items to watch, anomaly detection can help point out where an error is occurring, enhancing root cause analysis and quickly getting tech support on the issue. Anomaly detection helps the monitoring cause of chaos engineering by detecting outliers, and informing the responsible parties to act.

In enterprise IT, anomaly detection is commonly used for:

- Data cleaning
- Intrusion detection
- Fraud detection
- Systems health monitoring
- Event detection in sensor networks
- Ecosystem disturbances

Anomaly detection with ML

Machine learning, then, suits the engineer's purpose to create an AD system that:

- Works better
- Is adaptive and on time
- Handles large datasets

Despite these benefits, anomaly detection with machine learning can only work under certain conditions.

Unstructured data: what's the anomaly?

Applying machine learning to anomaly detection requires a good understanding of the problem, especially in situations with unstructured data.

Structured data already implies an understanding of the problem space. Anomalous data may be easy to identify because it breaks certain rules. If a sensor should never read 300 degrees Fahrenheit and the data shows the sensor reading 300 degrees Fahrenheit—there's your anomaly. There is a clear threshold that has been broken.

Fraud detection in the early anomaly algorithms could work because the data carried with it meaning. The data came structured, meaning people had already created an interpretable setting for collecting data. Their data carried significance, so it was possible to create random trees and look for fraud.

However, dark data and unstructured data, such as images encoded as a sequence of pixels or language encoded as a sequence of characters, carry with it little interpretation and render the old algorithms useless...until the data becomes structured. Structure can be found in the last layers of a convolutional neural network (CNN) or in any number of sorting algorithms.

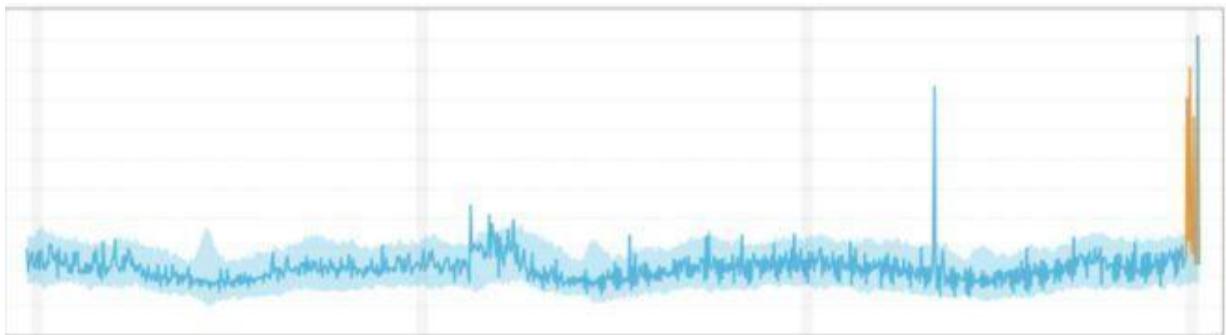
The three different types of time series anomalies

Understanding the types of outliers that an anomaly detection system can identify is essential to getting the most value from generated insights. Without knowing what you're up against, you risk making the wrong decisions once your anomaly detection system alerts you to an issue or opportunity.

Generally speaking, anomalies in your business data fall into three main categories — global outliers, contextual outliers, and collective outliers.

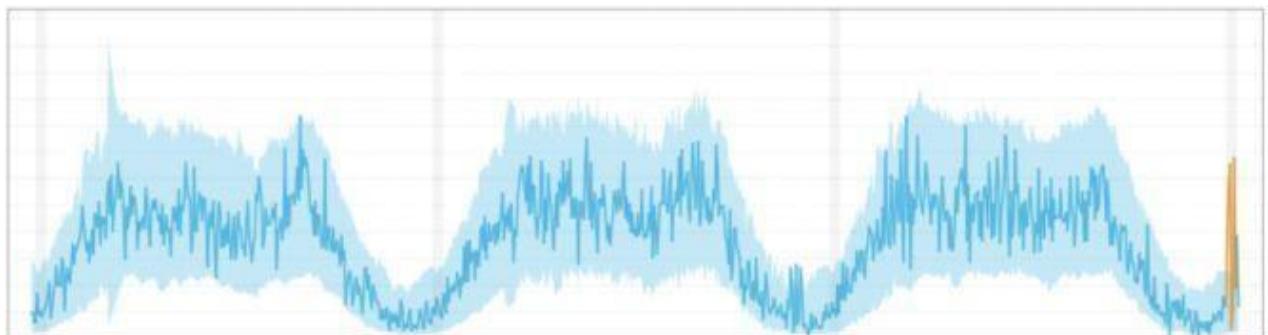
1. Global outliers

Also known as point anomalies, these outliers exist far outside the entirety of a data set.



2. Contextual outliers

Also called conditional outliers, these anomalies have values that significantly deviate from the other data points that exist in the same context. An anomaly in the context of one dataset may not be an anomaly in another. These outliers are common in time series data because those datasets are records of specific quantities in a given period. The value exists within global expectations but may appear anomalous within certain seasonal data patterns.



3. Collective outliers

When a subset of data points within a set is anomalous to the entire dataset, those values are called collective outliers. In this category, individual values aren't anomalous globally or contextually. You start to see these types of outliers when examining distinct time series together. Individual behavior may not deviate from the normal range in a specific time series dataset. But when combined with another time series dataset, more significant anomalies become clear.

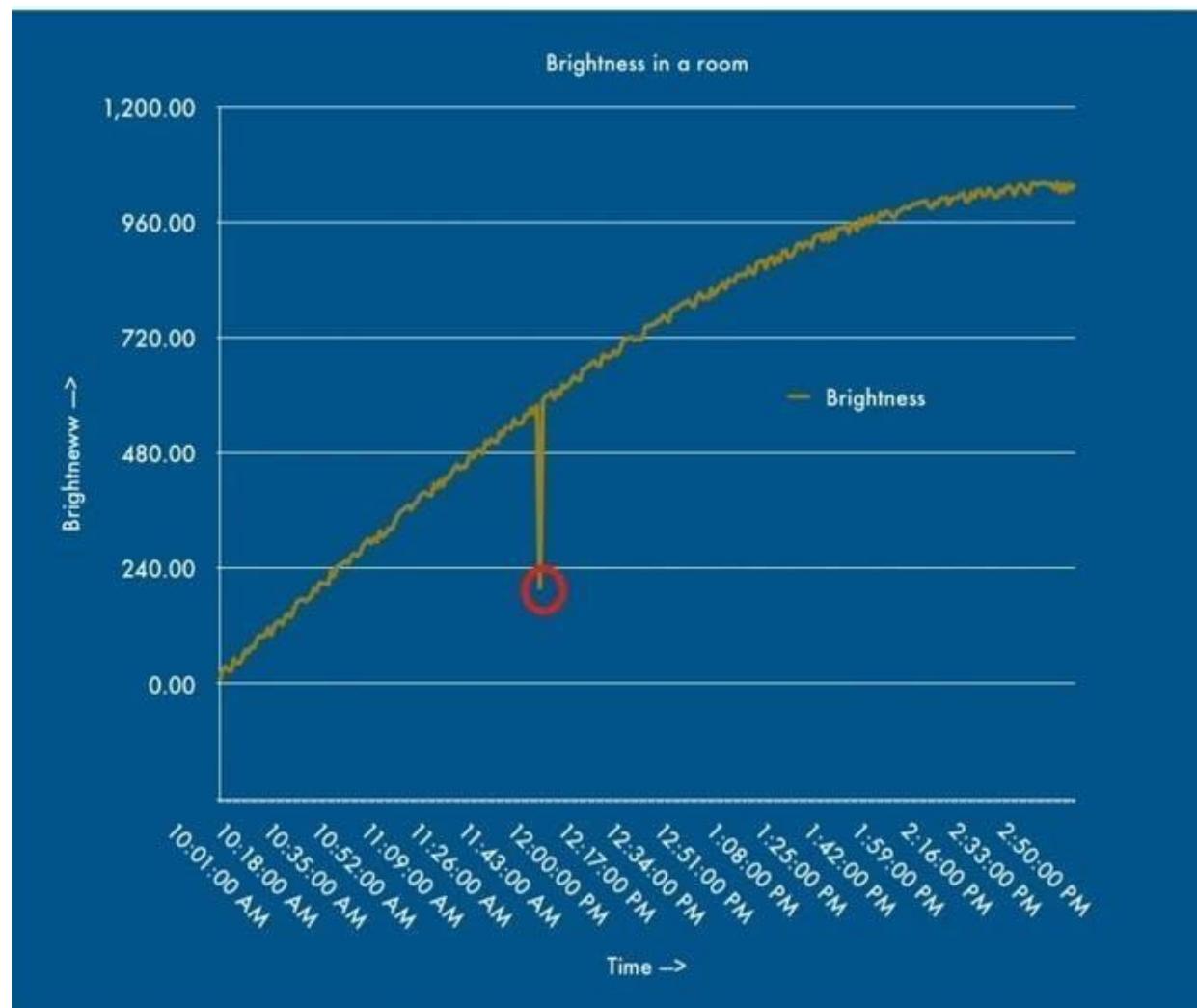


Anomaly detection in our project:-

Anomaly is something that deviates from what is standard, normal, or expected.

In other words an anomaly is an abnormality, a blip on the screen of life that doesn't fit with the rest of the pattern.

Let us consider the following graph -



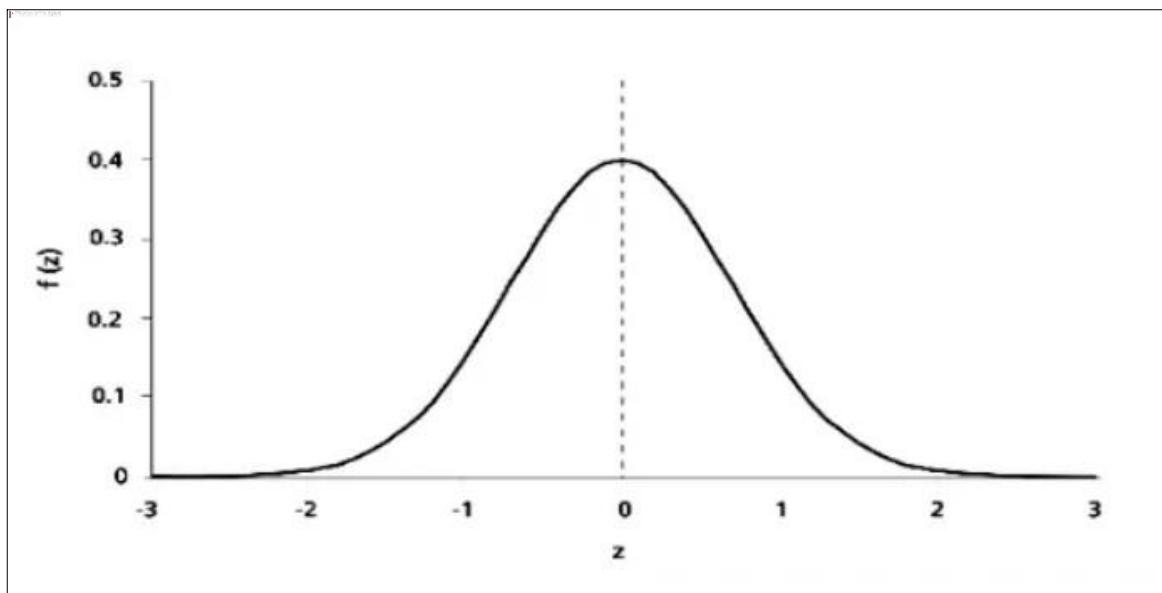
Here the red mark is known as anomaly because that point deviates from the rest of the pattern.

We can declare that red point as an anomaly by setting some threshold values around the pattern by using Z - score analysis.

3. Introduction to Z-Score

A z-score describes the position of a raw score in terms of its distance from the mean, when measured in standard deviation units. The z-score is positive if the value lies above the mean, and negative if it lies below the mean.

It is also known as a standard score, because it allows comparison of scores on different kinds of variables by standardizing the distribution. A standard normal distribution (SND) is a normally shaped distribution with a mean of 0 and a standard deviation (SD) of 1.



A standard normal distribution (SND).

Why are z-scores important?

It is useful to standardize the values (raw scores) of a normal distribution by converting them into z-scores because:

- (a) it allows researchers to calculate the probability of a score occurring within a standard normal distribution;
- (b) and enables us to compare two scores that are from different samples (which may have different means and standard deviations).

How do you calculate the z-score?

The formula for calculating a z-score is $z = (x - \mu)/\sigma$, where x is the raw score, μ is the population mean, and σ is the population standard deviation.

As the formula shows, the z-score is simply the raw score minus the population mean, divided by the population standard deviation.

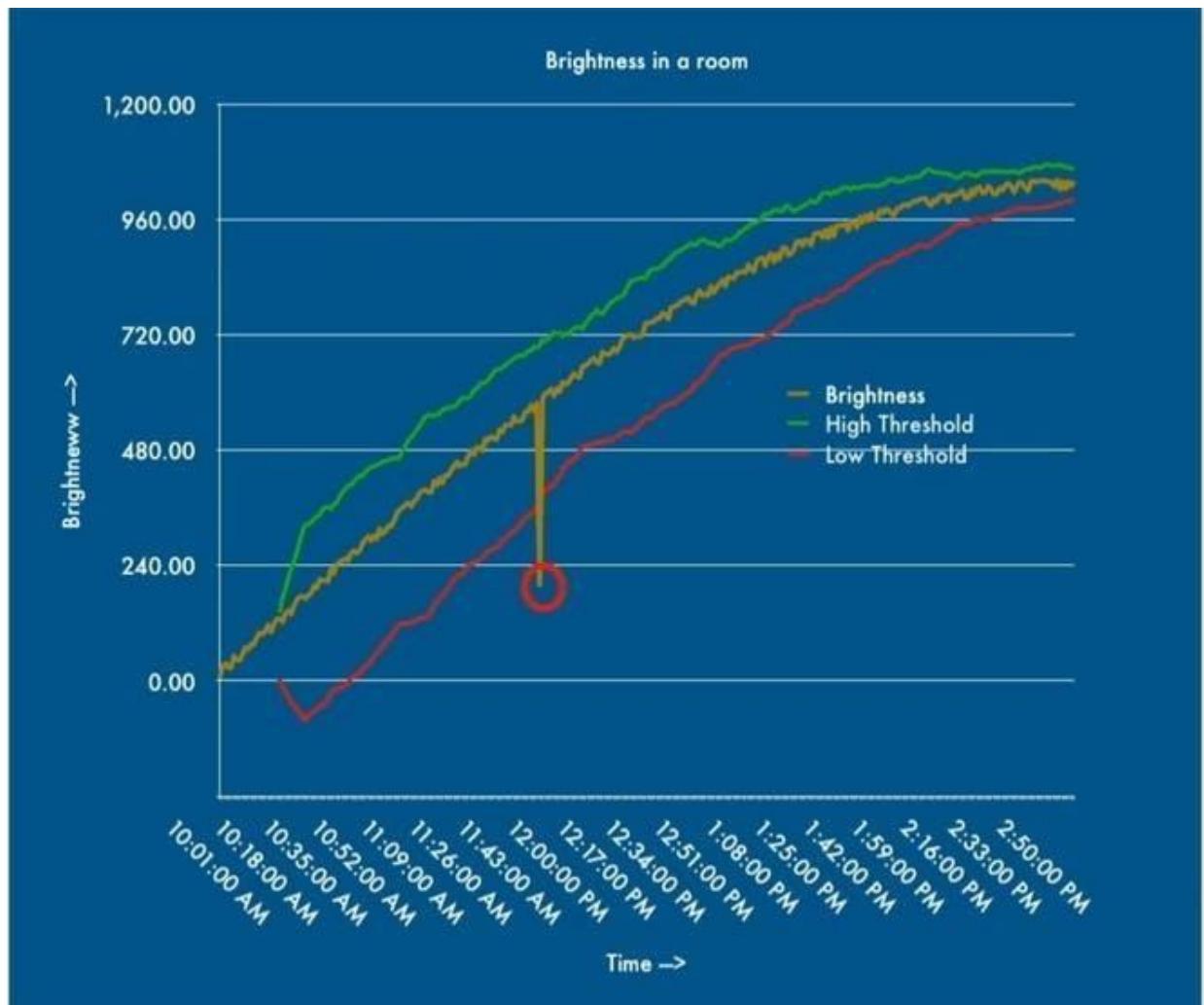
$$Z = \frac{x - \mu}{\sigma}$$

The diagram illustrates the components of the z-score formula. The variable x is labeled "Score" with a red arrow pointing to it. The variable μ is labeled "Mean" with a red arrow pointing to it. The variable σ is labeled "SD" (Standard Deviation) with a red arrow pointing to it.

Applications of Z-scores

The z-score is a useful metric for comparing test results between tests of different scales. For example, the comparison of scores in the SAT vs ACT would be best done using z-scores. As the z-score usefully translates scores in terms of their distribution, rather than rating scale, it is a useful metric for comparing tests of these sorts. Using the z-score technique, one can now compare two different test results based on relative performance, not individual grading scale.

Z-Score Analysis in our Project:-



In the above graph we can point out the anomaly by setting some threshold values.

- The green line signifies high threshold
- The red line indicates low threshold

We can set those threshold values by using below formulas -

- Mean Formula -

$$Mn = \frac{\sum_{i=1}^r Vi}{r}$$

- Z-Score Formula -

$$Zn = C * \sqrt{\frac{\sum_{i=1}^r (Vi - Mn)^2}{r}}$$

- Threshold Formula -

$$Tn = Vi \pm Zn$$

In the above formulas -

- **r** = Frame Size
- **C** = Multiplication Factor

IMPLEMENTING THE CIRCUIT CONNECTIONS

Hardware Connections

Hardware Required

- 1 x WiFi Module
- 1 x Micro USB Cable
- 1 x LDR (2 legged device with a red wave pattern disk on top)
- 1 x 10k Ohm Resistor (brown black orange color code)
- 1 x LED (Light Emitting Diode)
- 1 x Buzzer
- 1 x Breadboard
- 2 x Jumper wires
- Laptop or PC (Personal Computer)

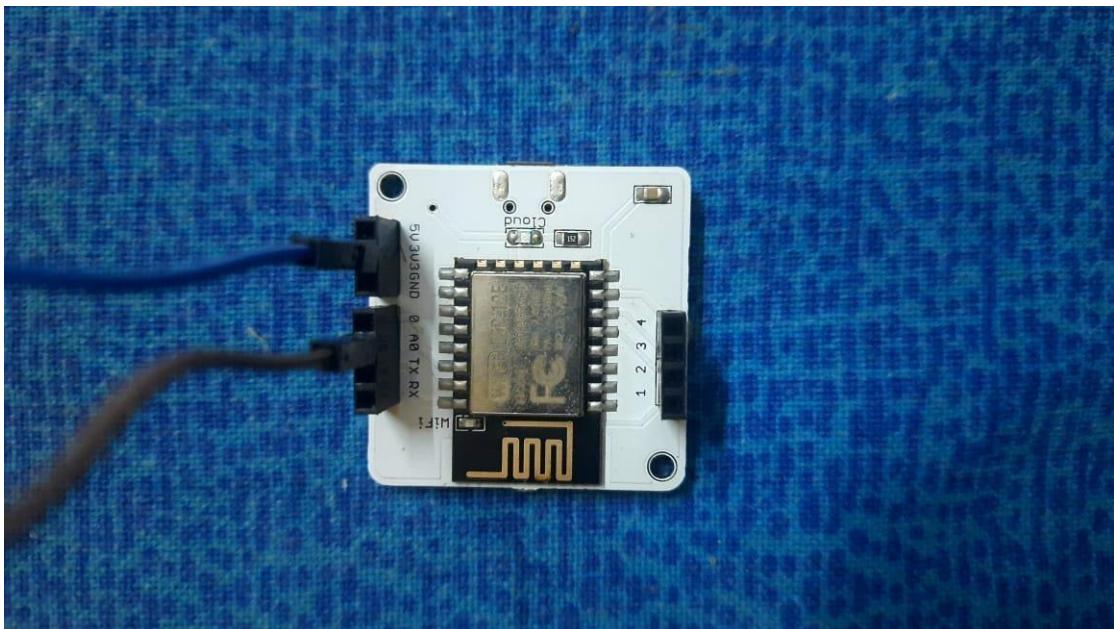
Connecting the LDR Circuit to the WiFi Module

Here are the steps for making the hardware connections:

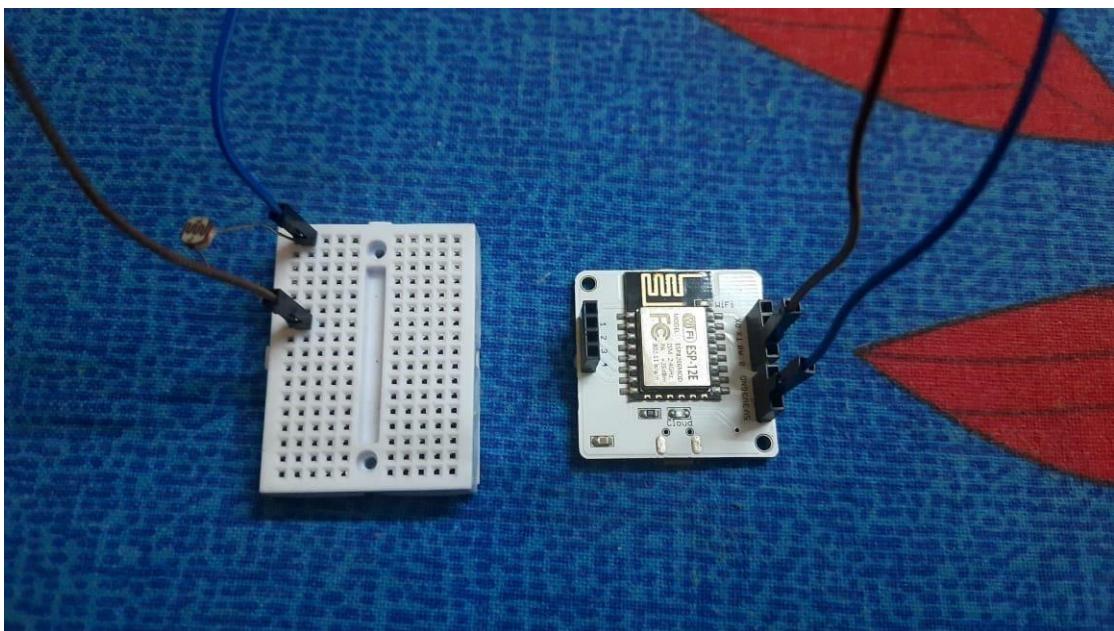
- Step 1: Insert both lead of the LDR into the Breadboard.



- Use Jumper wires to connect the LDR to the WiFi
- Connect/Insert the one lead of the LDR to the WiFi module's 3v3 pin.
- Insert other lead of the LDR into the A0 pin.



Following image shows the correct connections of the LDR to the WiFi module.



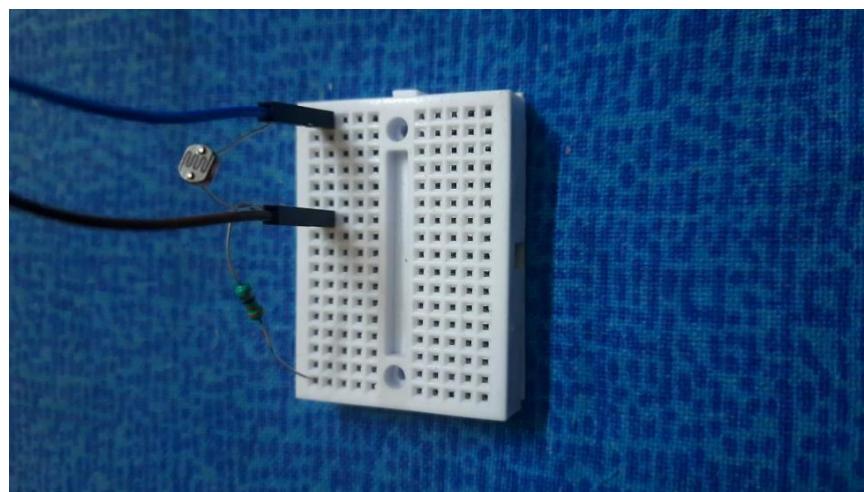
Connect the LDR to WiFi Module as shown in image above.

Note: There is no positive or negative for this and the 10k Ohm resistor. Also, make sure the Bolt module is not powered on while making connections. Always make it a habit to power off the circuit while making connections for your own and the circuit's safety. Double-check all connections before turning it on.

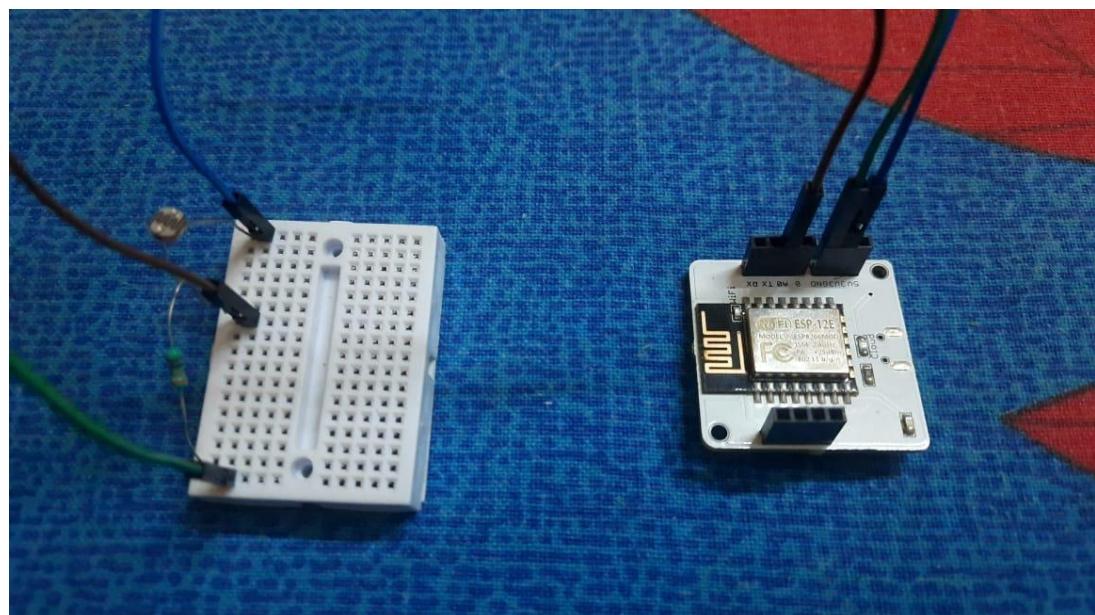
Connecting the Resistor to the WiFi Module

Here are the steps for making the hardware connections:

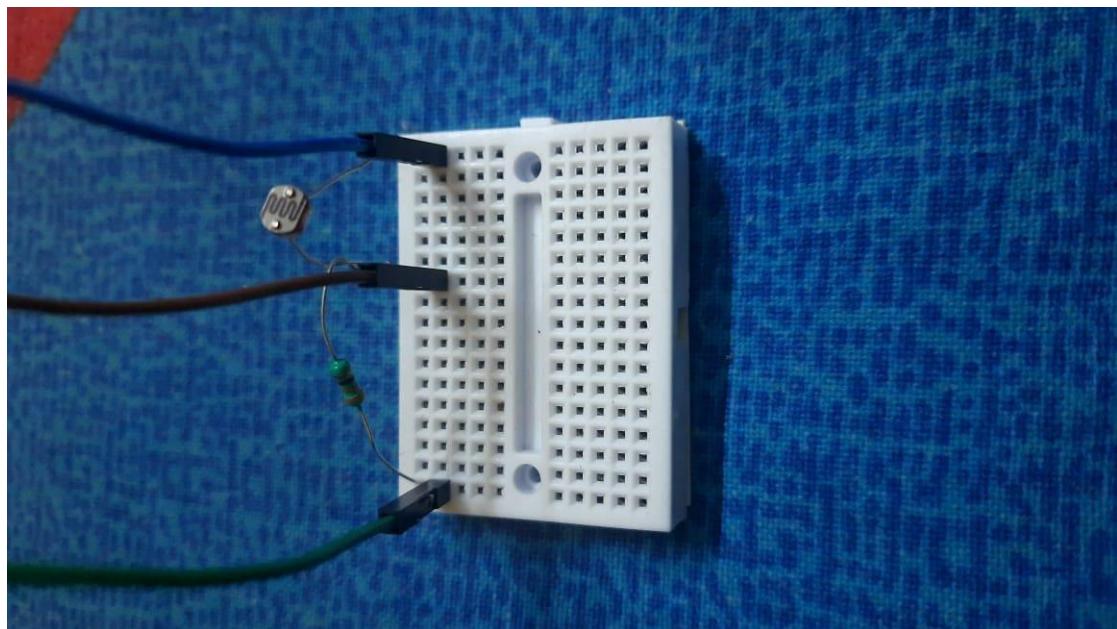
- Step 1 :- Insert one leg of the 10k Ohm resistor into the breadboard.
- Step 2 :- One leg of the 10k Ohm resistor should be wrapped around the other leg of the LDR.



- Step 3 :- Use the jumper wires to connect the 10K Ohm resistor to the WiFi module.
- Step :- Insert that end of the 10K Ohm resistor which has not been connected by the LDR to the GND (Ground) pin of the WiFi module.



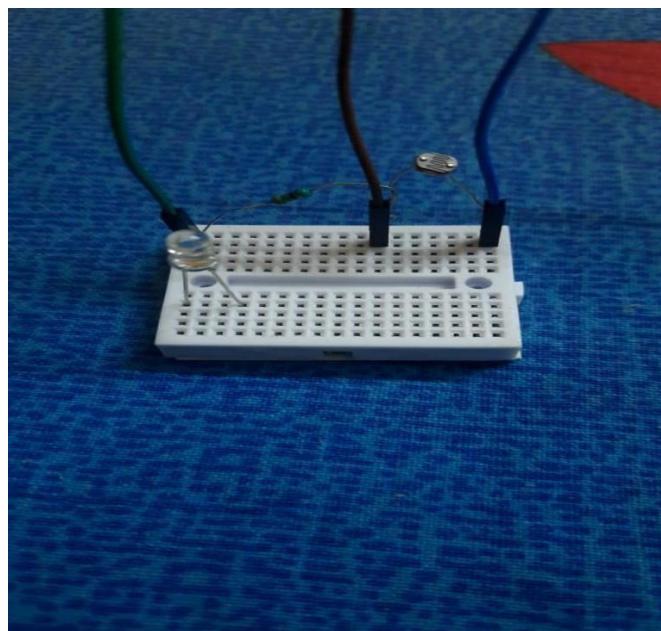
Following figure shows the final implementation of the 10K Ohm Resistor connection.



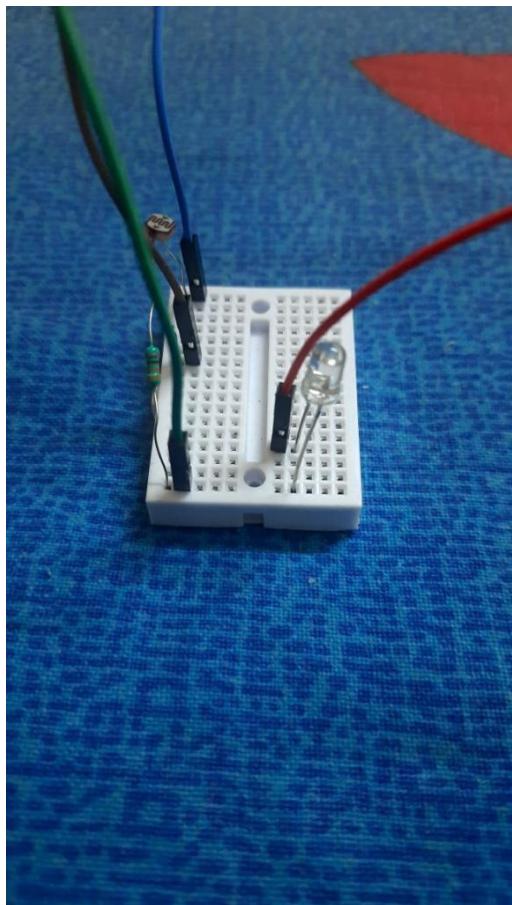
Connecting the LED to the WiFi Module

Here are the steps for making the hardware connections:

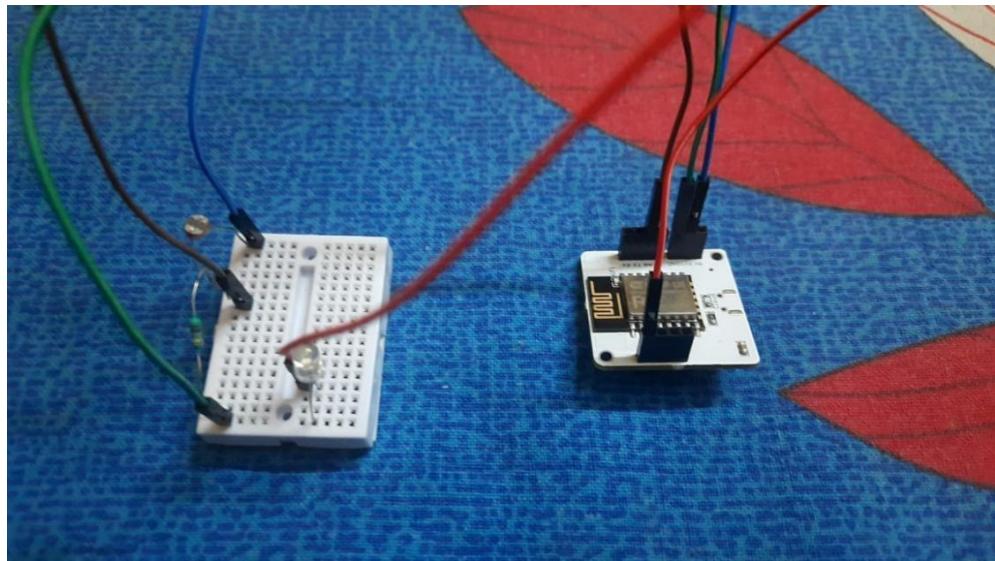
- Step 1 :- Insert the positive end (longer lead) of the LED in the other half of the bread board just as shown below:-



- Step 4 :- Now insert the jumper wire just above the negative lead of the LED.



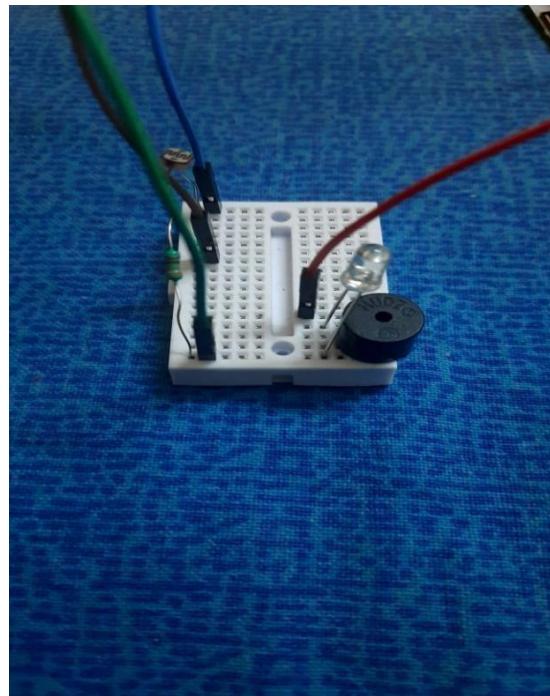
- Step 5 :- With help of the jumper wire connect the LED with the WiFi module that is in the pin mentioned as '1'.



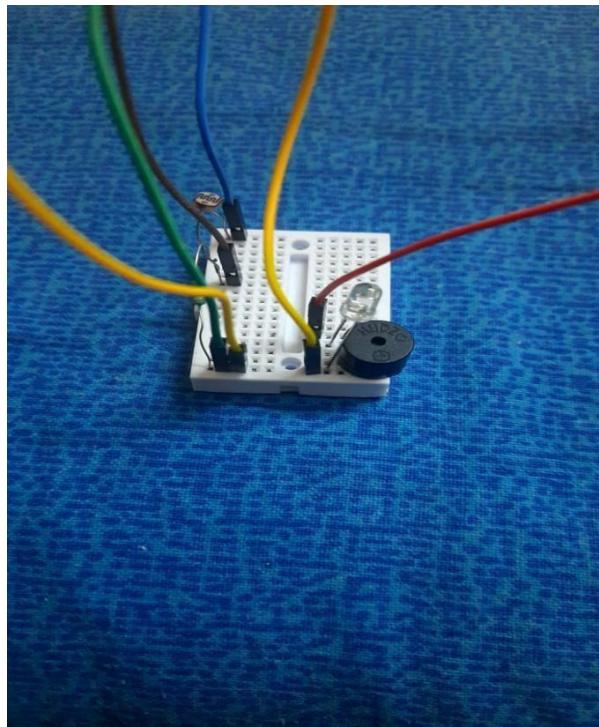
Connecting the Buzzer to the Breadboard

Here are the steps for making the hardware connections:

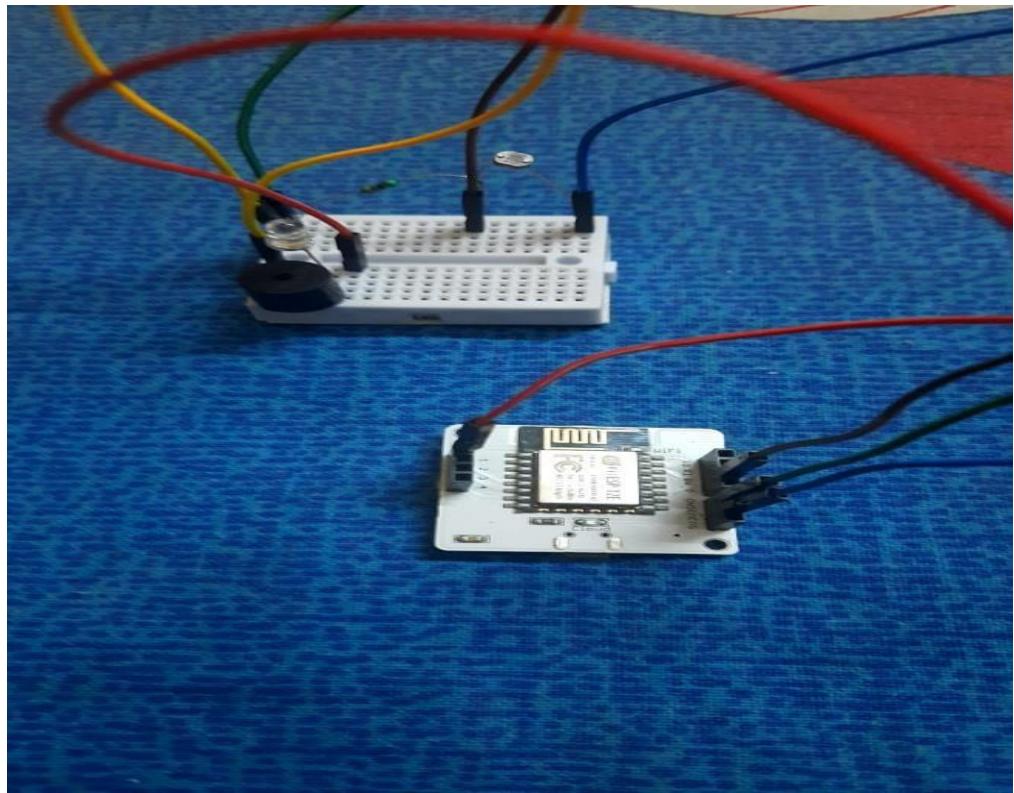
- Step 1 :- Insert the Buzzer ends in the breadboard just below the LED.

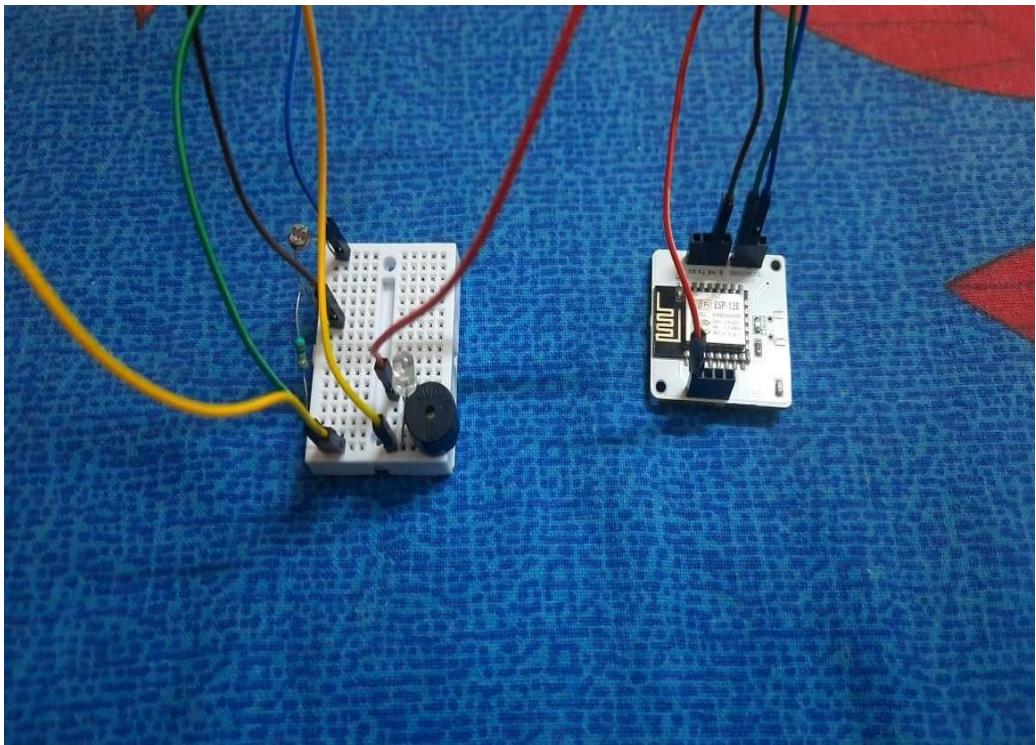


- Step 2 :- Connect the first and the second half of the Breadboard with a Jumper wire.



The following images shows the complete implementation of the circuit connections.





IMPLEMENTATION OF THE CODING PART

Coding Platform/OS (operating system) used:-

- VMware Workstation 12 Player (Ubuntu)

Follow the steps below to install Ubuntu Server using a virtual machine on your local Desktop/Laptop using VmWare Software.

Please read the steps completely mentioned below to install the Ubuntu virtual machine on your local Desktop/Laptop.

Also, note that the OS which the computer is running on will **NOT** be affected in any way.

Minimum Specifications

For installing a virtual machine on your Desktop/Laptop, it should meet the minimum specifications mentioned below,

- OS - Windows 7 or Windows 10
- CPU - Dual Core 2Ghz

- RAM - 4GB
- HDD - Should have at least 10GB free space
- High-Speed Internet Access

Brief Overview

The installation steps are mentioned in brief as below,

1. Download VmWare.
2. Install VmWare.
3. Download Ubuntu Server image.
4. Install Ubuntu Server image on VmWare.

1. Introduction to Ubuntu



Ubuntu is a Linux-based operating system. It is designed for computers, smartphones, and network servers. The system is developed by a UK based company called Canonical Ltd. All the principles used to develop the Ubuntu software are based on the principles of Open Source software development.

Ubuntu defaults installation contains many types of software pre-installed like

- Web Browser (Firefox),
- LibreOffice (Office Suite),
- Thunderbird,
- PDF Reader,
- and many other software.

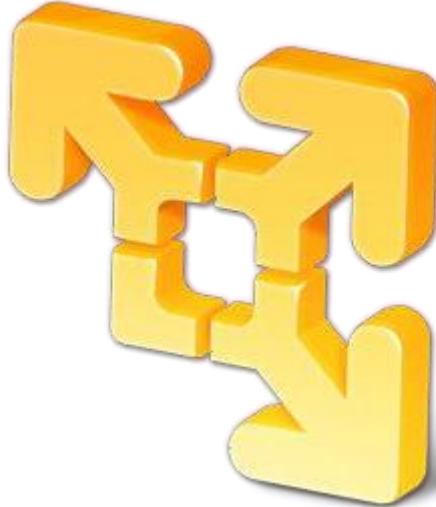
but if you want more types of software, then Ubuntu also contains a wide range of software that can install from Ubuntu Software Center Easily.

let me provide you some more interesting features in points

Some Important features of Ubuntu

- Pre-installed Office Softwares
- Best Internet Experience
- Good looks
- Simplicity
- Reliability
- Stability
- Works Fast
- Very Secure
- Completely Free To Use
- Live mode Supported
- Available For Almost all type of hardware
- Ubuntu Offers Thousand of app. free to use
- Available in many languages.
- Available in many flavors
- Large Community Support
- and much more.

2. Introduction to VMware Workstation



VMware Workstation Pro is a hosted hypervisor that runs on x64 versions of Windows and Linux operating systems[4] (an x86-32 version of earlier releases was available);[3] it enables users to set up virtual machines (VMs) on a single physical machine and use them simultaneously along with the host machine. Each virtual machine can execute its own operating system, including versions of Microsoft Windows, Linux, BSD, and MS-DOS. VMware Workstation is developed and sold by VMware, Inc., a division of Dell Technologies. There is a free-of-charge version, VMware Workstation Player, for non-commercial use. An operating systems license is needed to use proprietary ones such as Windows. Ready-made Linux VMs set up for different purposes are available from several sources.

VMware Workstation supports bridging existing host network adapters and sharing physical disk drives and USB devices with a virtual machine. It can simulate disk drives; an ISO image file can be mounted as a virtual optical disc drive, and virtual hard disk drives are implemented as .vmdk files.

VMware Workstation Pro can save the state of a virtual machine (a "snapshot") at any instant. These snapshots can later be restored, effectively returning the virtual machine to the saved state,[5] as it was and free from any post-snapshot damage to the VM.

VMware Workstation includes the ability to group multiple virtual machines in an inventory folder. The machines in such a folder can then be powered on and powered off as a single object, useful for testing complex client-server environments.

Programming Languages used in this Project:-

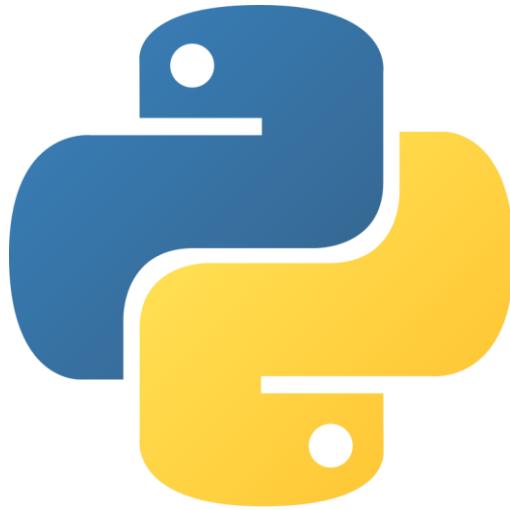
- Python
- JavaScript

1. Introduction to Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Why Python for this Project ?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.



Python Syntax compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

Python Library used in the Project

The Bolt python library provides an easy to use interface of the Bolt Cloud API's.

Follow the steps given below to use the Bolt python library in your code.

- Install boltiot library first by running the below command on terminal/command line.

```
Shell
sudo pip install boltiot
```

- Create your python file which will use the Bolt API library.

```
Shell
touch file_name.py
```

- Import the library in your python code.

Python

```
from boltiot import Bolt
api_key = "ACXXXXXXXXXXXXXXXXXXXXXX"
device_id = "BOLT1234"
mybolt = Bolt(api_key, device_id)
```

Syntax	Parameter(s)	Description
from boltiot import Bolt	None	Import the Bolt Class from the boltiot library.
api_key = "ACXXXXXXXXXXXXXXXXXXXXXX"	Api Key	Assign the Bolt Cloud api key to api_key variable
device_id = "BOLT1234"	Device Id	Assign the device id to device_id variable
mybolt = Bolt(api_key, device_id)	Api key, Device id	Create an object of type Bolt and initialize the api key and device id.

Commands

GPIO Functions

A) **digitalWrite Command** - Write digital output.

Python

```
response = mybolt.digitalWrite('0', 'HIGH')
```

B) **digitalRead Command** - Read digital input.

Python

```
response = mybolt.digitalRead('0')
```

C) analogRead Command - Read analog input.

Python

```
response = mybolt.analogRead('A0')
```

D) analogWrite Command - Write analog output.

Python

```
response = mybolt.analogWrite('0', '100')
```

UART Functions

A) serialBegin Command - Initialize serial communication.

Python

```
response = mybolt.serialBegin('9600')
```

B) serialWrite Command - Send serial data output.

Python

```
response = mybolt.serialWrite('Hello')
```

C) serialRead Command - Read incoming serial data.

Python

```
response = mybolt.serialRead()
```

Utility Functions

A) isOnline Command - Get Bolt device status.

Python

```
response = mybolt.isOnline()
```

B) restart Command - Restart Bolt device.

Python

```
response = mybolt.restart()
```

C) version Command - Get Bolt device version.

Python

```
response = mybolt.version()
```

2. Introduction to JavaScript

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as **LiveScript**, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name **LiveScript**. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

The ECMA-262 Specification defined a standard version of the core JavaScript language.

- JavaScript is a lightweight, interpreted programming language.
- Designed for creating network-centric applications.
- Complementary to and integrated with Java.

- Complementary to and integrated with HTML.
- Open and cross-platform

Client-Side JavaScript

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.

It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field.

The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server.

JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

Introduction to Twilio

What is Twilio?

Suppose your parent has allotted a task to you to check the water level in the plant daily and water them if required. Isn't it very boring and tiresome? What if you get an SMS on your mobile telling that water level in your plant is very less and need water immediately. Sounds Interesting right!! But how can it be done?

This can be achieved with the help of Bolt and a third-party company named 'Twilio'.

So in this lesson, we will learn about Twilio and how to create an account on Twilio.

What is Twilio used for?

Twilio is a third-party SMS functionality provider. It is a cloud communications platform as a service (PaaS) company. Twilio allows software developers to programmatically make and receive phone calls and also send and receive text messages using its web service APIs.

Please note that SMS delivery via a trial Twilio account is not guaranteed to be instant by Twilio. Also, note that SMS will not be sent to numbers which have DND(Do Not Disturb) turned ON.

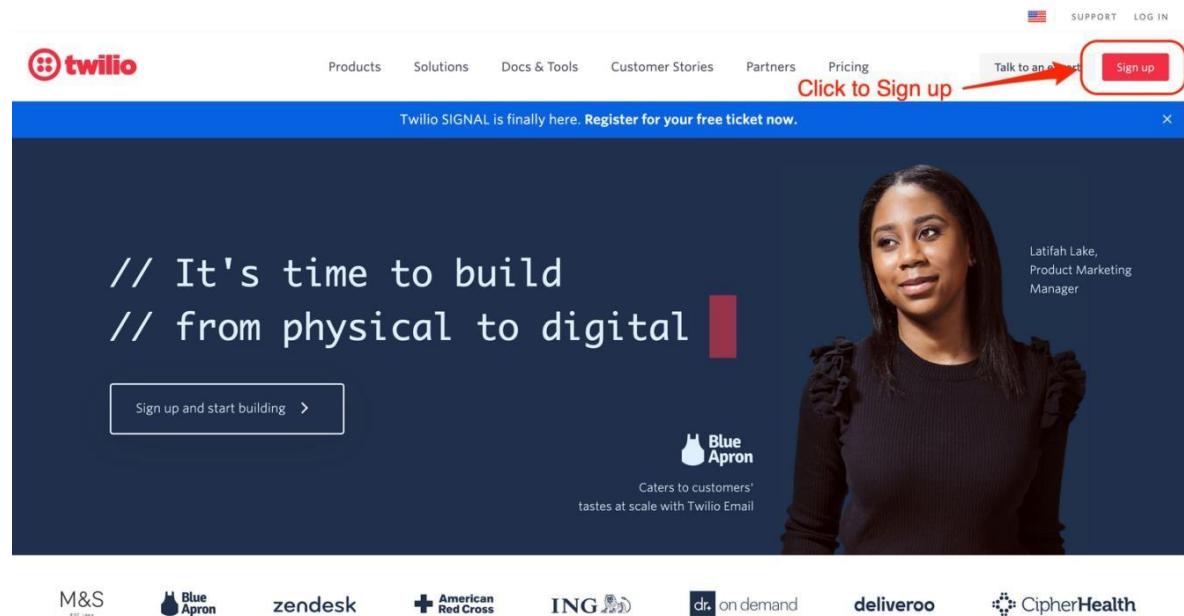
How to create an account on Twilio?

Twilio provides a free trial account so that you can experience its services. We will be using Twilio to send an SMS later on by using a Python program.

Steps to sign up on Twilio

Step 1 - Open Twilio

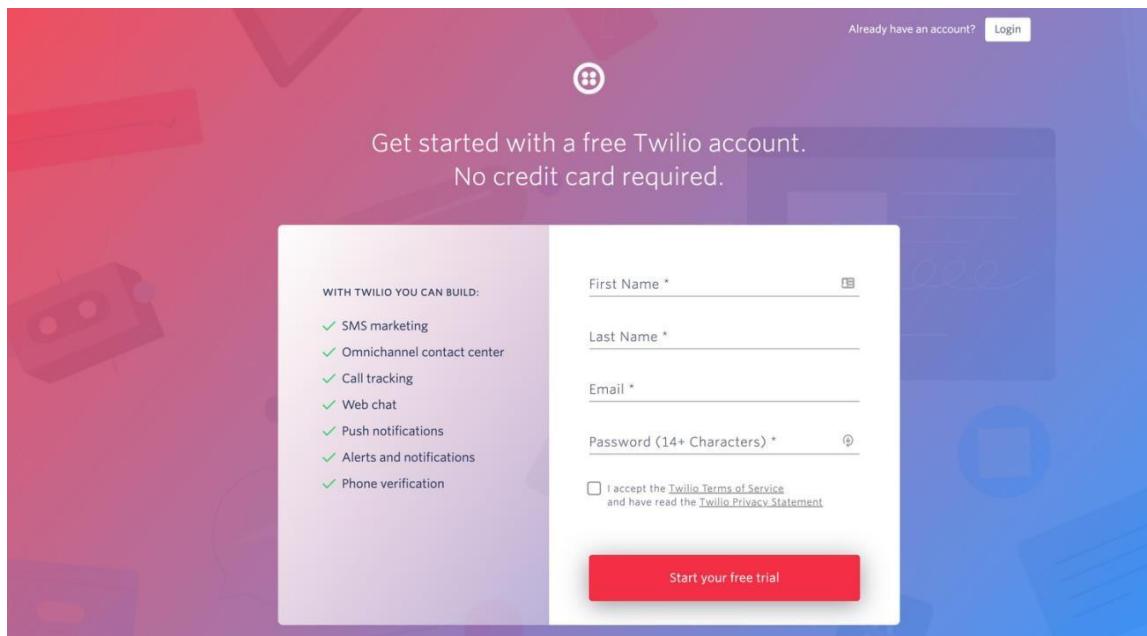
Open www.twilio.com on your browser. Click on the “Sign up” button to go to the Signup page.



Step 2 - Fill in the sign up form

Fill in the sign up form with your details. You will need to enter your first and last name, email ID and a 14 digit password.

Read and accept the checkbox for Terms of Service and Privacy policy of Twilio and then click on “Start your free trial”.



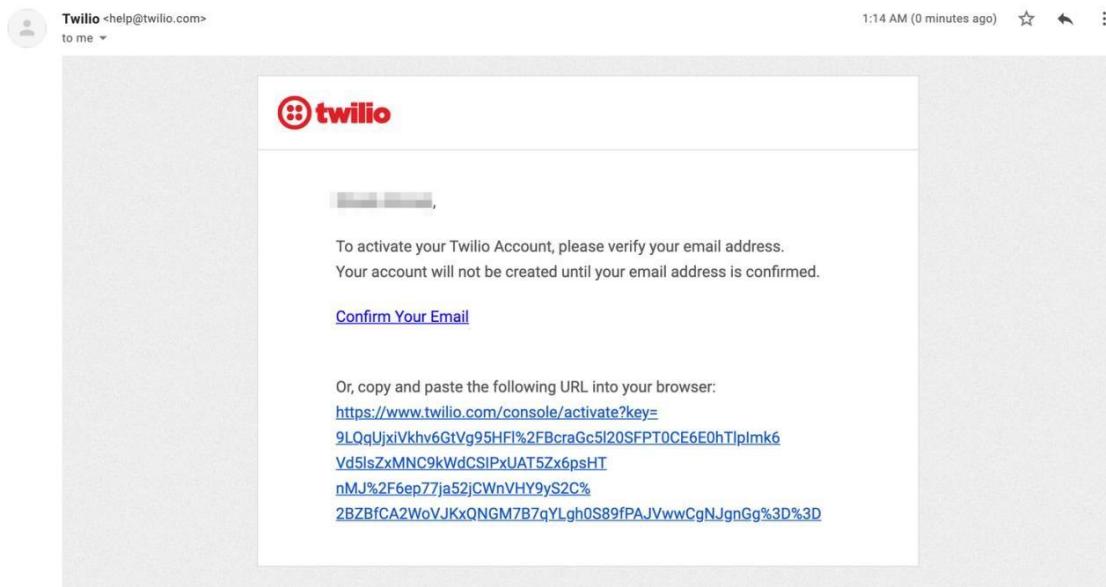
Step 3 - Complete the verification process

To start using Twilio's services, you will need to verify your email address and your mobile number.

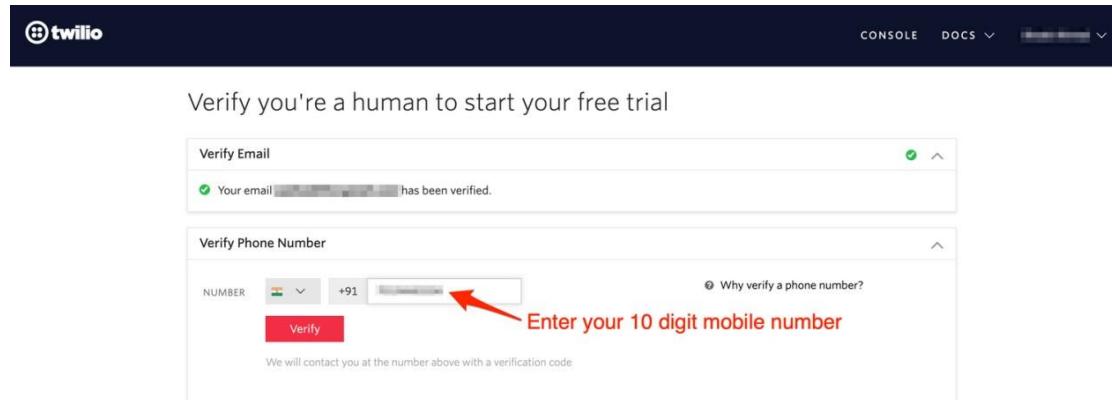
A screenshot of the Twilio verification process. The top navigation bar includes the Twilio logo, "CONSOLE", "DOCS", and a user profile icon. The main content area has a heading "Verify you're a human to start your free trial". Below it is a "Verify Email" section. It shows a message: "We sent an email to [REDACTED]@gmail.com. To continue, please check your email and verify your account." To the right is an envelope icon with a blue circle containing the number "1". Below the envelope are links for "Didn't receive the email?" and "Resend Email". At the bottom of the page, there is a footer with the text "© Twilio, Inc. All rights reserved." and links to "Privacy Policy" and "Terms of Service".

You will need to verify your email address first. Twilio will have sent an email to your inbox with a link to verify your email address. Read the email and follow the instructions given in the email to verify your email address.

Verify your email  



Once you have verified your email address, you will then need to verify your mobile number. For this, in the next screen, enter your 10-digit mobile number and click on “Verify”. They will send an OTP to this number for verification.



Once you have received the OTP code on your mobile number, enter it on the screen and click on “Submit”.

Verify you're a human to start your free trial

Verify Email
Your email [REDACTED] has been verified.

Verify Phone Number
Please enter the verification code we sent to <+91 [REDACTED]>
[Red arrow points to the input field]
Submit

Want to verify with a Call instead of SMS?
Didn't receive a code?
Resend Code (40)

Enter the OTP that you have received

© Twilio, Inc. All rights reserved. [Privacy Policy](#) | [Terms of Service](#)



Now, your account has been verified and you are now one step closer to sending SMSs via Twilio. Now, Twilio will onboard you once your mobile number has been verified.

Step 4 - Onboarding process

For the on-boarding process, you will be asked a few questions on what you plan to do with the Twilio account. Since we will be using the Twilio account to send SMSs via Python code, we will choose the options accordingly.

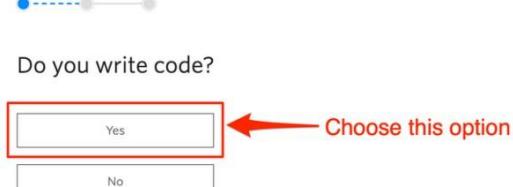
<Message>Welcome! Let's customize your experience!</Message>

Do you write code?

[Red box surrounds the 'Yes' button]
Yes
No

Choose this option

© Twilio, Inc. All rights reserved. [Privacy Policy](#) | [Terms of Service](#)



Firstly, choose the option “Yes” when asked if you write code. Since we will be writing code to send SMS, this is applicable for us.

The screenshot shows the Twilio welcome screen. At the top, there's a navigation bar with the Twilio logo, 'CONSOLE', 'DOCS', and a user icon. Below the navigation bar, a message reads '<Message>Welcome! Let's customize your experience!</Message>'. A progress bar with three dots is shown. The main question is 'What is your preferred language?'. Below the question are several language icons: RUBY, SWIFT, C#, PHP, NODE.JS, PYTHON (which is circled in red), ANDROID, and JAVA. A red arrow points from the text 'Choose this option' to the PYTHON icon. There's also a 'Other' button.

What is your preferred language?



Choose this option

Now, in the next screen, click on Python as we will be writing the code in Python to send the SMS.

The screenshot shows the Twilio goal selection screen. At the top, there's a navigation bar with the Twilio logo, 'CONSOLE', 'DOCS', and a user icon. Below the navigation bar, a message reads '<Message>Welcome! Let's customize your experience!</Message>'. A progress bar with three dots is shown. The main question is 'What is your goal today?'. Below the question are three buttons: 'Use Twilio in a project' (which is highlighted with a red border and has a red arrow pointing to it from the text 'Choose this option'), 'Use Twilio in a student or hackathon project', and 'Explore Twilio'. At the bottom right is a 'Skip to dashboard' button.

What is your goal today?

Use Twilio in a project

Choose this option

Use Twilio in a student or hackathon project

Explore Twilio

Skip to dashboard

© Twilio, Inc. All rights reserved. [Privacy Policy](#) | [Terms of Service](#)

Choose the option “Use Twilio in a project” when asked for the question “What is your goal today?”

A screenshot of the Twilio onboarding welcome screen. At the top, it says "<Message> Welcome! Let's customize your experience! </Message>". Below that, there's a message "One last thing...". A red arrow points from the text "Choose this option" to the first button in a grid of options. The grid contains the following buttons:

- Send or receive a SMS
- Build a contact center with Flex
- Make or receive a phone call
- Add video to your app
- Send WhatsApp messages
- Buy a SIM card
- Build a bot
- Something Else

A "Skip to dashboard" button is at the bottom right.

Select “Send or receive a SMS” when asked, “What do you want to do first?”

When you click on this option, you will then be redirected to the Twilio dashboard.

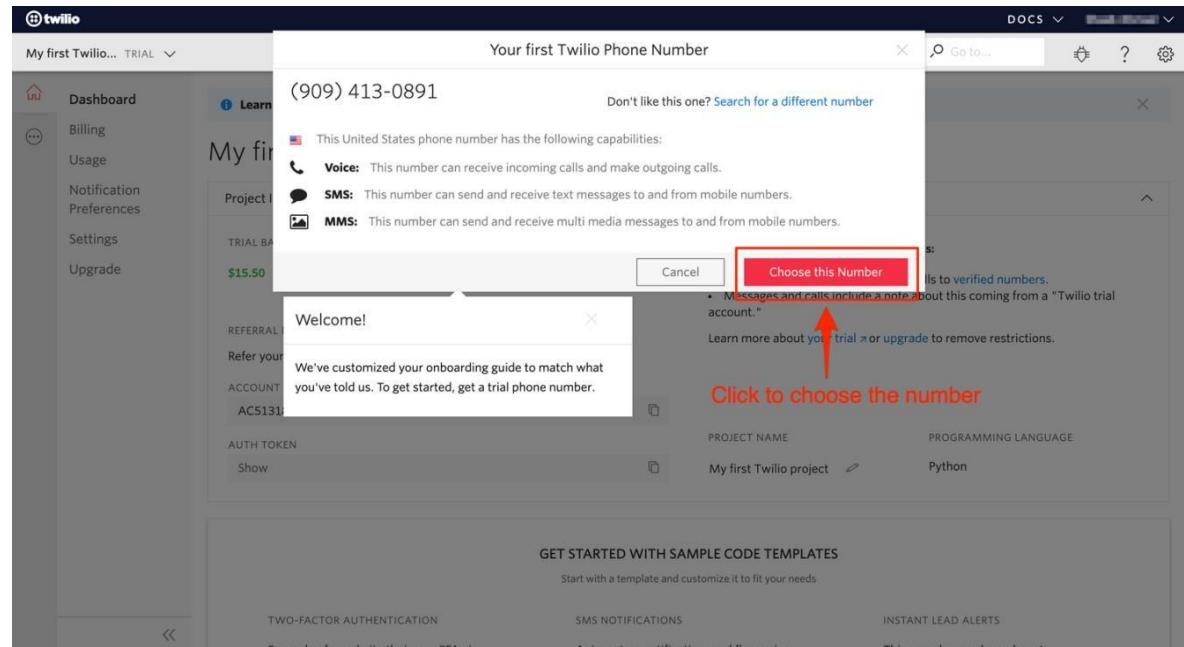
Step 5 - Get credentials

Now the onboarding process is completed, we now need to get the credentials for sending the SMS in our python code.

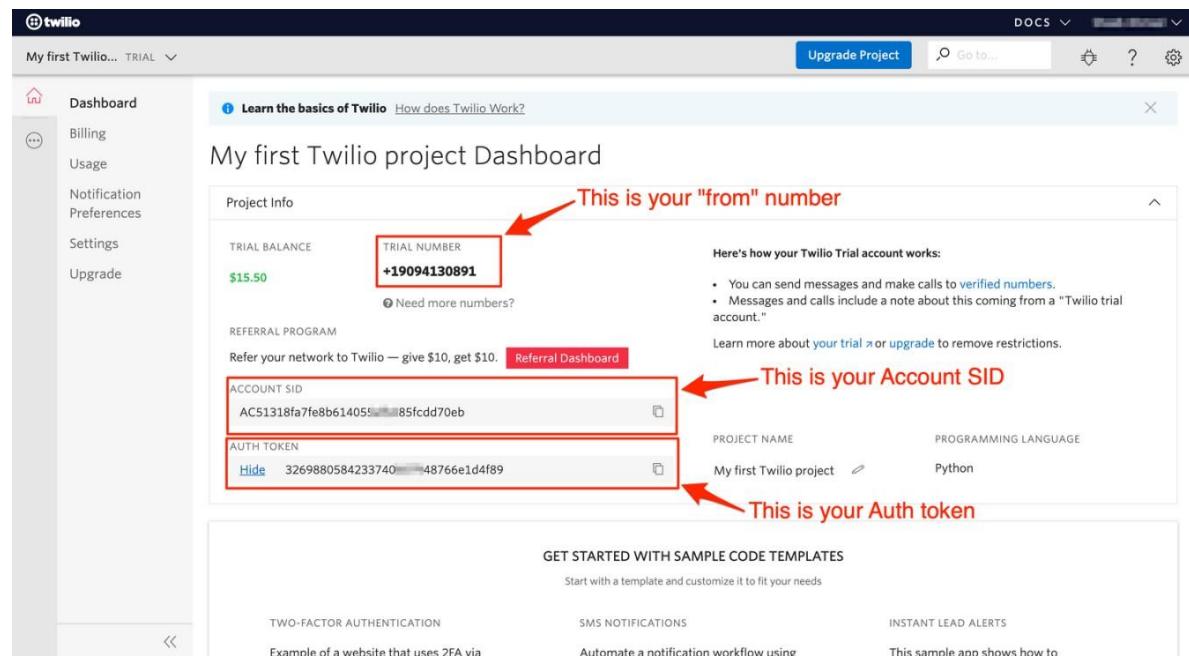
On the dashboard, search for the button “Get a trial number” and click on it. This should be highlighted for you if you have just completed the onboarding process. We need a trial number in order to send an SMS from Twilio.

A screenshot of the Twilio dashboard titled "My first Twilio project Dashboard". On the left sidebar, there are links for Dashboard, Billing, Usage, Notification Preferences, Settings, and Upgrade. The main area shows a "TRIAL BALANCE" of "\$15.50" and a "Get a Trial Number" button. A red arrow points to this button with the text "Click to get a trial number". To the right, there's a section about the trial account and a "GET STARTED WITH SAMPLE CODE TEMPLATES" section. The bottom navigation bar includes links for Two-factor authentication, SMS notifications, Instant lead alerts, and other documentation.

When you click on the button, it will open a popup asking you to choose a number. Once you have finalised the number, click on “Choose this number”.



Now, we have all the credentials required to send an SMS via Twilio. On the dashboard, as highlighted in the picture below, you will need three items.



Your trial number is the “from number” you will be receiving the SMS from.

Your account SID and Auth token are highlighted. In order to view the Auth token, you may need to click on the “Show” button next to it.

Please note these credentials as it will be required in the next lecture where we will write code to send the SMS via a Python program.

Output (Part 1)

The code will initially start printing the following -

```
This is the value 157
Not enough data to compute Z-score. Need -7 more data points
This is the value 125
Not enough data to compute Z-score. Need -6 more data points
This is the value 126
Not enough data to compute Z-score. Need -5 more data points
This is the value 152
Not enough data to compute Z-score. Need -4 more data points
This is the value 124
Not enough data to compute Z-score. Need -3 more data points
This is the value 147
```

After about 100 seconds (10 seconds delay with a frame size of 10), the system will start printing the light intensity values, as per the following -

Note: The values may change depending on the lighting conditions.

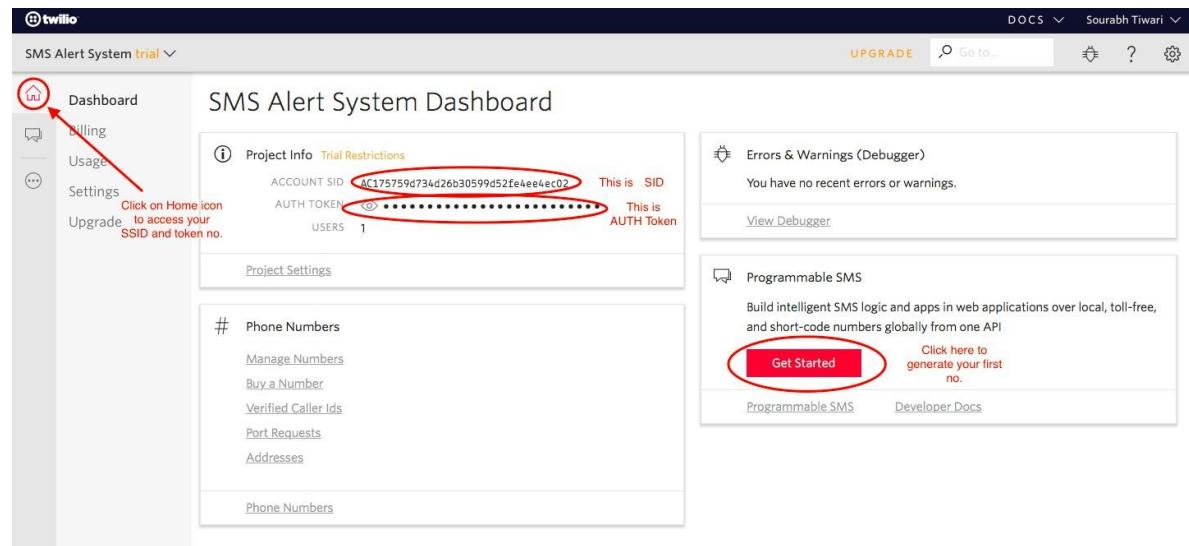
```
This is the value 96
This is the value 133
This is the value 135
This is the value 96
This is the value 89
This is the value 138
This is the value 127
This is the value 112
This is the value 100
This is the value 88
This is the value 104
This is the value 127
This is the value 129
This is the value 90
This is the value 125
This is the value 126
This is the value 92
This is the value 118
This is the value 137
This is the value 113
This is the value 92
```

Requirements for Sending SMS

Twilio is a popular third-party SMS functionality provider. Twilio allows software developers to programmatically make and send and receive text messages using its web service APIs.

Creating an account on Twilio

- Create an account on <https://www.twilio.com/try-twilio>
- After successfully creating the account, click on the home icon on the leftmost bar as shown in the image below. This screen will have your SID and Auth token which we will use later during coding. Copy both and save them in a text file.



- Sender number is a special number which is provided by Twilio. To generate the sender number for your twilio account click on Get Started button. Click on Get a number button.
- A popup will appear. Click on Choose this number button.
- Then you will get a number starting with '+' e.g +7192345678. Copy this number and save to text editor for future references.

That's it. Now you have your SID, auth token and sender number. We will use Bolt Python library and these Twilio values to create our own SMS Alert system.

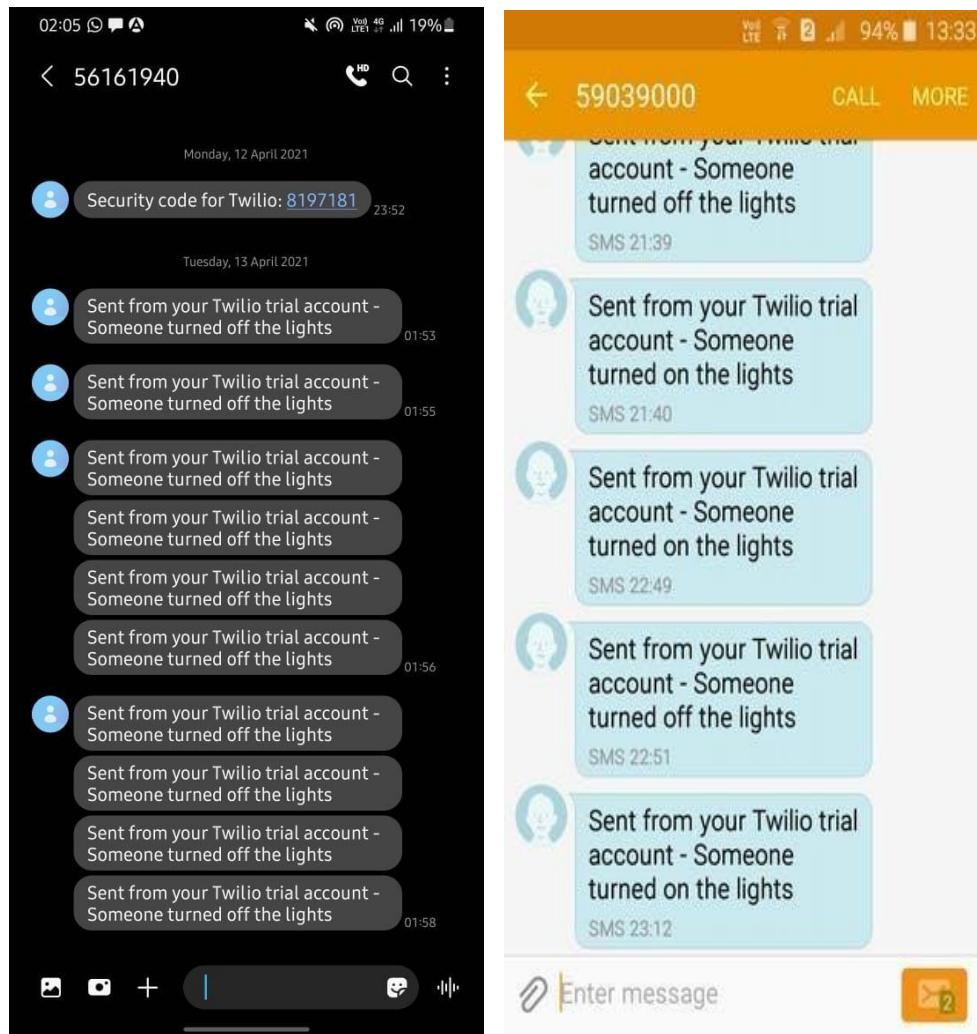
Output (Part 2)

We are using the Z-score algorithm to dynamically change the bounds at which an alert is sent.

So when you move the light source close to or away from the LDR slowly, the bounds also start changing slowly.

But when you move the light source close to or away from the LDR very fast, the bounds do not change fast enough, so the system detects an anomaly and sends an SMS alert while triggering the Buzzer as well as the indicator Led.

The SMS alert will be something like this -



Test Device 1

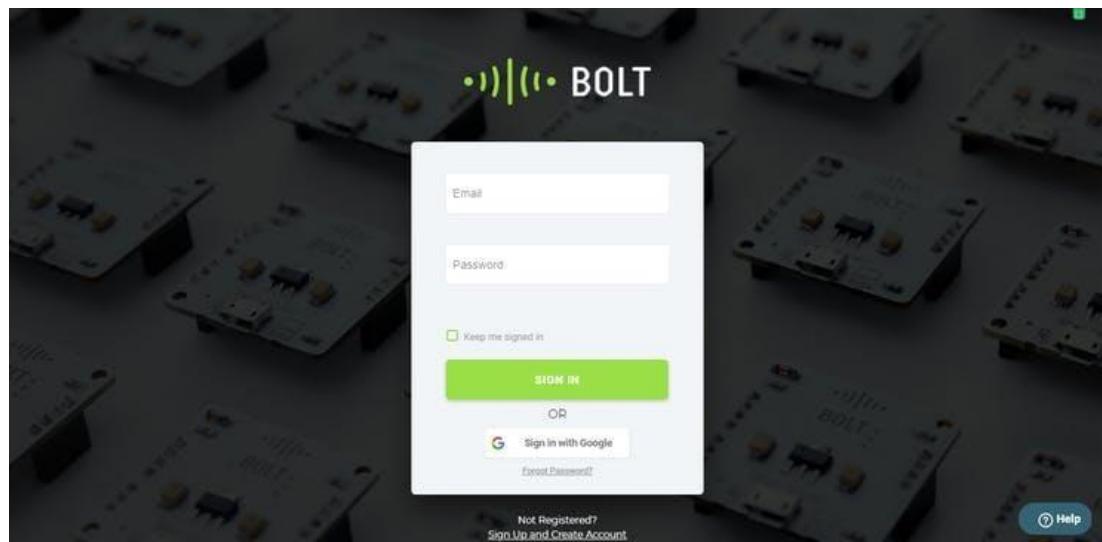
Test Device 2

Output (Part 3)

Data Visualization

The light intensity data can be visualized through Bolt Cloud as follows -

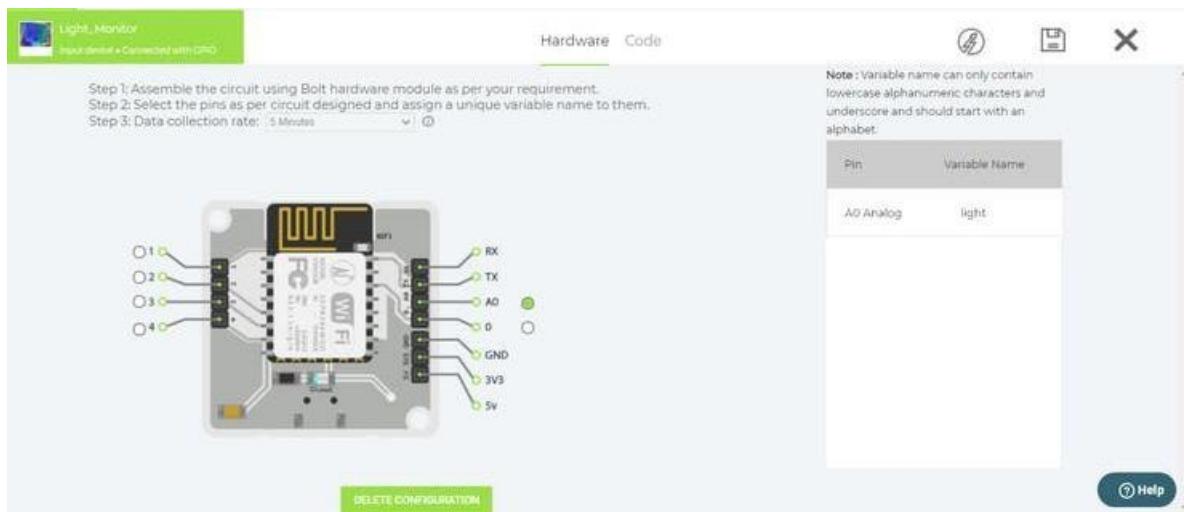
- Just Log in to Bolt Cloud -



- Go to Products tab and Create a Product -

A screenshot of the Bolt Cloud 'Products' tab. On the left is a sidebar with icons for Devices, Products (selected), Alerts, Assets, API, Docs, and New. The main area shows a list of products: 'controlling_LED', 'Buzzer_Control', 'Temp_Monitor', and 'temp_predict'. Each product item has a small thumbnail, a name, and a circular icon with a question mark. Below the list is a dark blue sidebar with a '+ Add Product' button. The main content area contains the text: 'This product is not linked to any devices. Click on ⚙ icon to link devices.' A 'Help' button is at the bottom right.

- Configure the product and save it -



Hardware

```

1 setChartsLibrary("google-chart");
2 setChartTitle("Scatter Graph");
3 setChartType("scatterGraph");
4 setXAxisTitle("TimeStamp");
5 setCrosshair(true);
6 setAnimation(true);
7 plotChart("time_stamp","light");

```

Note : Variable name can only contain lowercase alphanumeric characters and underscore and should start with an alphabet.

Code

```

1 setChartLibrary("google-chart");
2 setChartTitle(" Scatter Graph");
3 setChartType("scatterGraph");
4 setAxisName("Time","Light");
5 setCrosshair(true);
6 setAnimation(true);
7 plotChart("time_stamp","light");

```

Explanation to the Data Visualization Code (JavaScript)

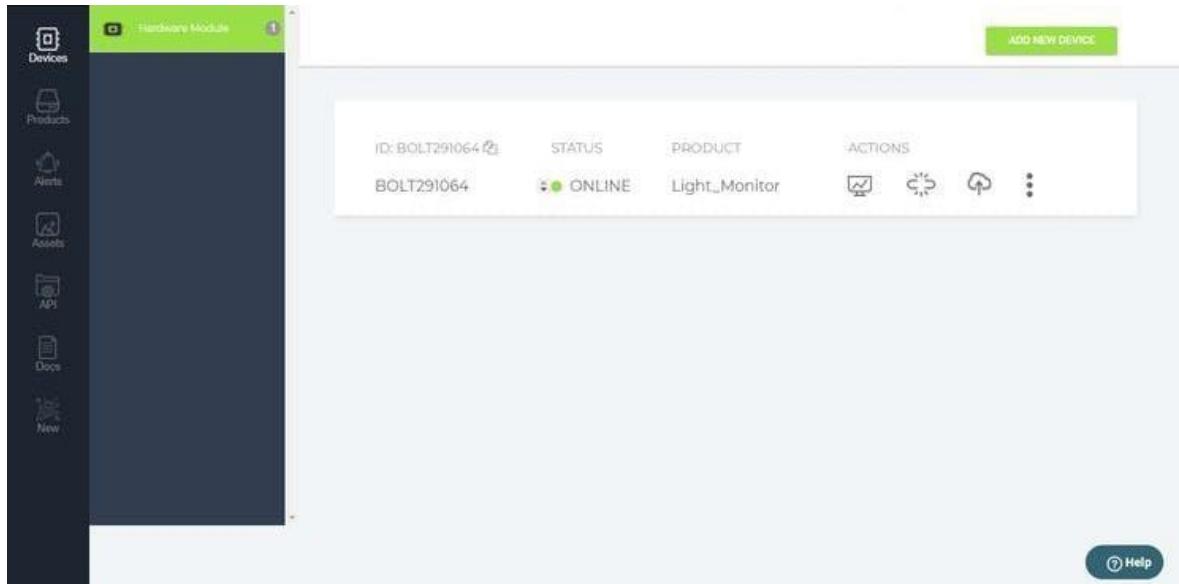
Syntax	Parameter(s)	Description
setChartLibrary(' google-chart');	Library Name	Set the chart library to be used. Bolt Cloud uses 'google-chart' to visualize the data.
setChartTitle(' Your Graph Title');	Graph Name	Set the title of the graph according to your choice.
setChartType(' scatterGraph');	Graph type	Set the type of chart to be plotted. Bolt Cloud offers 8 different types of charts to be plotted i.e Line Graph, Bar Graph, Scatter Graph, Area Graph, Table Graph, Stepped

Syntax	Parameter(s)	Description
<code>setAxisName('X-Axis Name', 'Y-axis Name');</code>	Axis Name	Graph, Histogram Graph and Gauge.
<code>plotChart('time_stamp', 'your_variable_name');</code>	Variable Names	Set the display name for each of the graph axis. Display the graph on the screen for the given variables. First parameter is always 'time_stamp' and second parameter is the variable name that you have given during hardware configuration of the product.
<code>setNumberFormatter('format_type');</code>	Format type	Describes how numeric columns should be formatted. setNumberFormatter() function supports the following options as parameter - 'decimal', 'scientific', 'percentage', 'currencyInDollar', ,

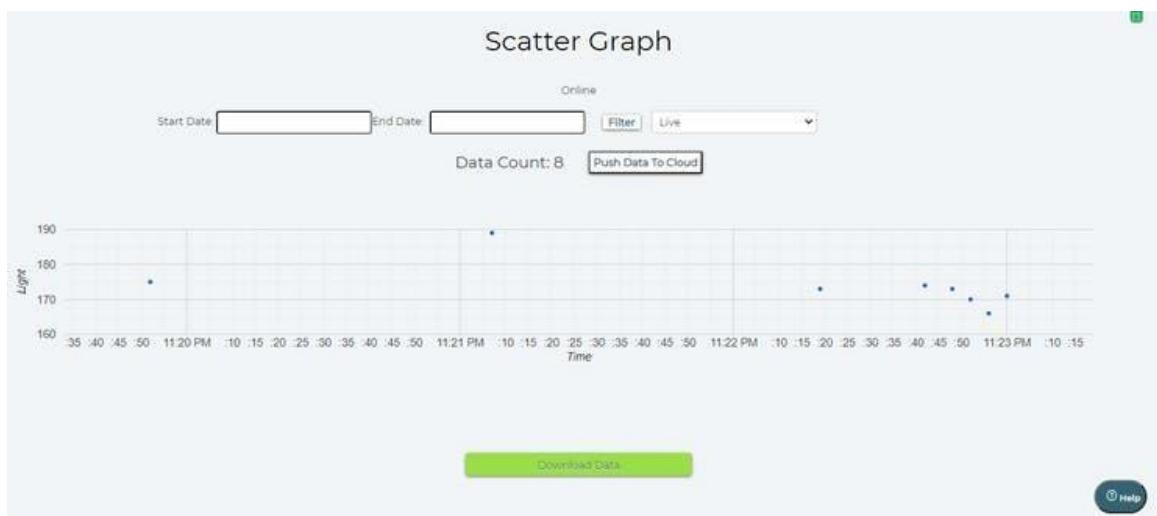
Syntax	Parameter(s)	Description
		' currencyInRupees '. This function should be called before the plotChart() function.
setAnimation(boolean) ;	Boolean type	This function describes how to animate modifications made to a chart, instead of applying them instantly. setAnimation() function supports the following options as parameter- true, false. It is enabled (true) by default and this function should be called before the plotChart() function.
setCrosshair(boolean) ;	Boolean type	Crosshairs are thin vertical and horizontal lines centered on a data point in a chart. When you, as a chart creator, enable crosshairs in

Syntax	Parameter(s)	Description
		<p>your charts, your users will then be able to target a single element. <code>setCrosshair()</code> function supports the following options as parameter- true, false. It is disabled (false) by default and this function should be called before the <code>plotChart()</code> function.</p>
<code>setDateFormatter(' date format');</code> Date Format		<p>Formats the x-axis date value in a variety of ways. <code>setDateFormatter()</code> function supports the following options as parameter- '<code>mm/dd/yyyy</code>', '<code>dd/mm/yyyy</code>'. This function should be called before the <code>plotChart()</code> function.</p>

- Link the product with your Bolt device -



- Finally view this device -

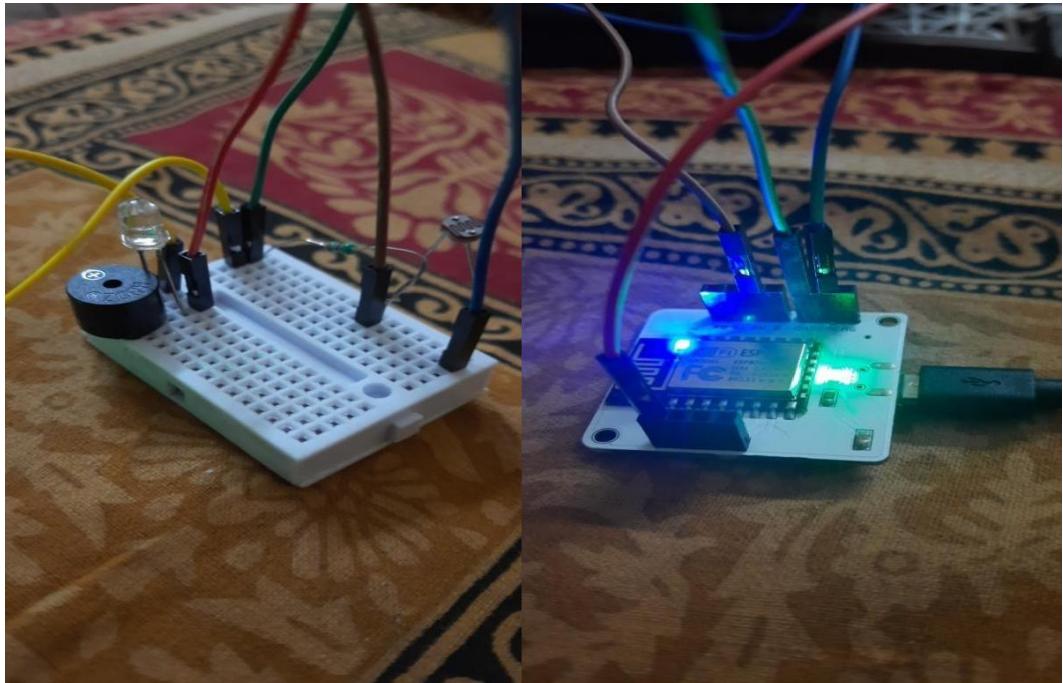


Scatter charts are used to plot data points on a horizontal and a vertical axis in the attempt to show how much one variable is affected by another. Each row in the data table is represented by a marker whose position depends on its values in the columns set on the X and Y axes.

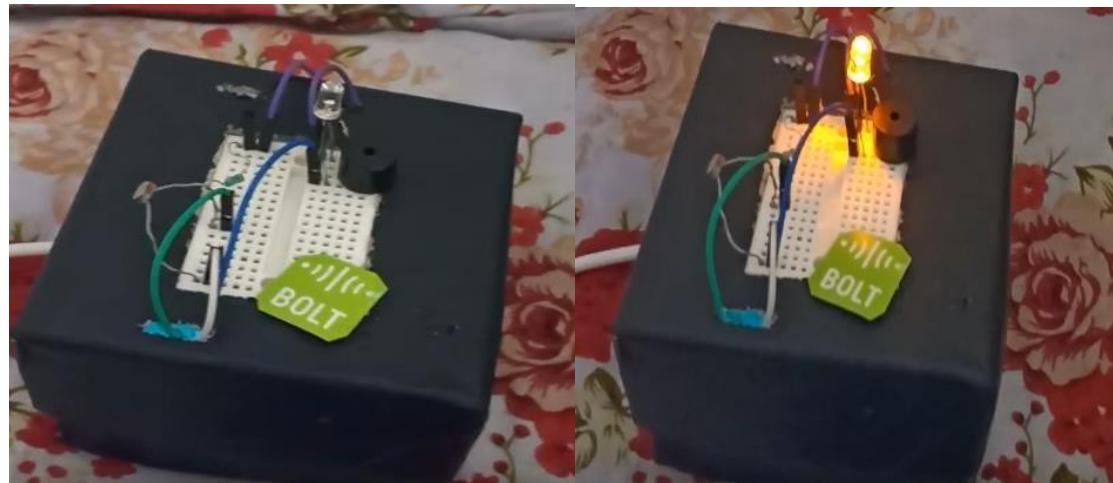
FINAL SETUP (COMPLETED)

Below are some snapshots of our successfully completed project work.

Final setup 1



Final setup 2

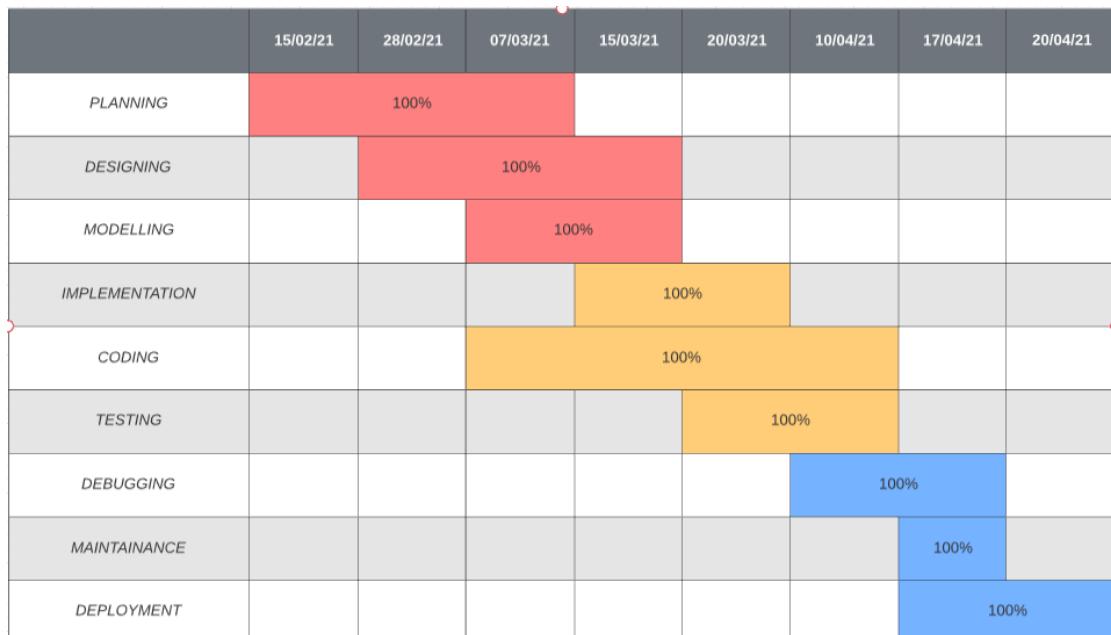


The END GAME!

MODULE & TEAM MEMBER WISE DISTRIBUTION OF WORK

Planning	Designing	Building	Coding	Testing	Development
Harsh Anand			Harsh Anand	Harsh Anand	Harsh Anand
Keshav Kant Mishra	Keshav Kant Mishra	Keshav Kant Mishra	Keshav Kant Mishra		
Bobby Sharma	Bobby Sharma		Bobby Sharma	Bobby Sharma	
Shubham Kumar		Shubham Kumar	Shubham Kumar	Shubham Kumar	

Following is the Gantt Chart showing the whole progress report with team wise work distribution:-



Work Distribution:

 Harsh	 Keshav	 Shubham
Bobby	Bobby	Keshav

BIBLIOGRAPHY

https://www.electronics-notes.com/articles/electronic_components/resistors/light-dependent-resistor-ldr.php

<https://www.newbiehack.com/categories/newbiehack-resistors-330-ohm-quarterwatt-through-hole>

<https://eepower.com/resistor-guide/resistor-fundamentals/what-is-a-resistor/#>

<https://byjus.com/physics/light-emitting-diode/>

<https://components101.com/wireless/esp12e-pinout-datasheet>

<https://www.anodot.com/blog/what-is-anomaly-detection/>

<https://deepai.org/machine-learning-glossary-and-terms/z-score>

https://www.tutorialspoint.com/javascript/javascript_overview.htm

<https://www.leveredge.com/iot-ebook/what-is-iot>

<https://www.security.org/home-automation/>