**IOT**

Alexa Voice Command to Control LED

Project Report

Submitted by:

Nithin Gopal Krishna T.P,

Anvesh P.A,

Nagendra Prasad P,

Apoorva Narahari .

## Problem Statement

## The world is becoming smarter everyday and everyone wants to control their things in a smarter way. If a person can control his devices by just a voice command then it would be an excellent breakthrough in IOT technology.

Abstract

This project is a combination of Voice Technology, Cloud Networking and Embedded system. Here one has to configure an AWS (Amazon Web Services) account. Connect the LED device with the Particle Photon and configure it with wifi. Write a code to control the LED device on the Particle cloud. Create a recipe on the IFTTT Website using ALEXA and the Particle cloud to control the device. By using echosim.io one can a send a voice command which is already registered to control the LED device.

Requirements

Hardware:

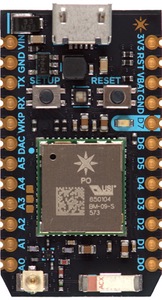
* Mobile Phone / Tablet
* Laptop
* Particle Photon
* 1 LED
* Bread Board
* Micro USB Cable

Software:

* IFTTT Website Registration
* Particle cloud Registration
* AWS(ALEXA) Registration
* Echosim.io
* Particle APP

System Description

### Particle photon



The Photon is a $19 tiny Wi-Fi development kit for creating connected projects and products for the Internet of Things. It's easy to use, it's powerful, and it's connected to the cloud.

Particle's Internet of Things hardware development kit, the Photon, provides everything you need to build a connected product. Particle combines a powerful ARM Cortex M3 micro-controller with a Broadcom Wi-Fi chip in a tiny thumbnail-sized module called the PØ (P-zero).

Particle adds a rock solid 3.3VDC SMPS power supply, RF and user interface components to the PØ on a small single-sided PCB called the Photon. The design is open source and ready to integrate the Photon into the product.

## Amazon Echo

Amazon Echo is a [voice-enabled](https://en.wikipedia.org/wiki/Voice_command_device) [wireless speaker](https://en.wikipedia.org/wiki/Wireless_speaker) developed by [Amazon.com](https://en.wikipedia.org/wiki/Amazon.com). The device consists of a 9.25-inch (23.5 cm) tall cylinder speaker with a seven-piece [microphone array](https://en.wikipedia.org/wiki/Microphone_array). The device responds to the name "Alexa"; this "wake word" can be changed by the user to either "Amazon" or "Echo".

The device is capable of voice interaction, music playback, making to-do lists, setting alarms, streaming podcasts, playing audio books, and providing weather, traffic and other real time information. It can also control several [smart devices](https://en.wikipedia.org/wiki/Smart_device) using itself as a [home automation](https://en.wikipedia.org/wiki/Home_automation) hub.

The device comes with a manual and voice-activated remote control which can be used in lieu of the 'wake word'. Echo's microphones can be manually disabled by pressing a mute button to turn off the audio processing circuit.

Echo requires a Wi-Fi internet connection in order to work. Echo's voice recognition capability is based on Amazon Web Services and the Amazon common voice platform it acquired from [Yap](https://en.wikipedia.org/wiki/Yap_(company)) [Evi](https://en.wikipedia.org/wiki/Evi_(software)" \o "Evi (software)), and [IVONA](https://en.wikipedia.org/wiki/IVONA)

Application

Echo can play music from owner's Amazon Music accounts and has built-in support for the [Pandora](https://en.wikipedia.org/wiki/Pandora_Radio) and [Spotify](https://en.wikipedia.org/wiki/Spotify" \o "Spotify) streaming music services and has support for [IFTTT](https://en.wikipedia.org/wiki/IFTTT) and [Nest Thermostats](https://en.wikipedia.org/wiki/Nest_Labs#Nest_Learning_Thermostat).

Developers can also use the "Smart Home Skill API", a new addition to the Alexa Skills Kit, to easily teach Alexa how to control cloud-controlled lighting and thermostat devices. All of the code runs in the cloud – nothing is on any user device.

## OAuth

Particle tightly adheres to OAuth specifications to ensure secure and correct access to private data and devices. OAuth is an open standard for authorization, providing client applications "secure delegated access" to server resources on behalf of a resource owner

**IFTTT** is a free [web](https://en.wikipedia.org/wiki/World_Wide_Web)-based service that allows users to create chains of simple [conditional statements](https://en.wikipedia.org/wiki/Conditional_(computer_programming)), called "recipes", which are triggered based on changes to other web services such as [Gmail](https://en.wikipedia.org/wiki/Gmail), [Facebook](https://en.wikipedia.org/wiki/Facebook), [Instagram](https://en.wikipedia.org/wiki/Instagram" \o "Instagram), and [Pinterest](https://en.wikipedia.org/wiki/Pinterest" \o "Pinterest).IFTTT is an abbreviation of "If This Then That".IFTTT recipes are widely used and shared by its users.

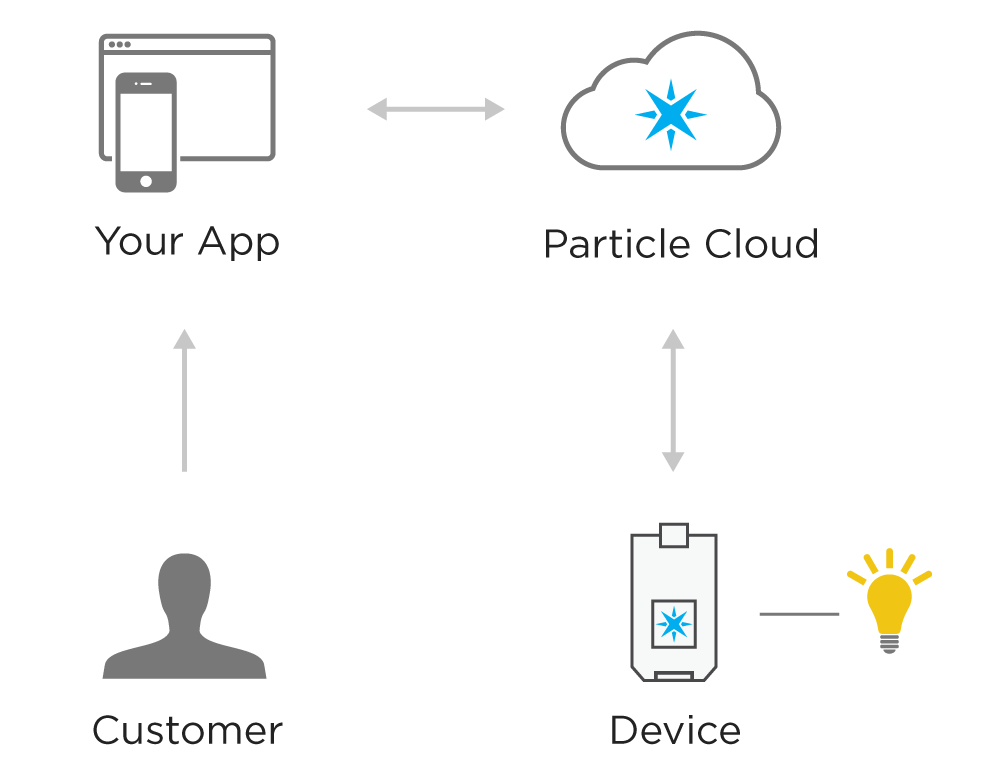
In addition to its web-based application, IFTTT for [iPhone](https://en.wikipedia.org/wiki/IPhone) was launched and contained three channels: [iOS](https://en.wikipedia.org/wiki/IOS" \o "IOS) Photos, Reminders and Contacts. An [iPad](https://en.wikipedia.org/wiki/IPad" \o "IPad) version with iOS notification support was later introduced. Later IFTTT released an [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) version of the app.

IFTTT employs the following concepts.

* **Channels** are the "basic building blocks of IFTTT", they mainly describe a series of data from a certain web service such as [YouTube](https://en.wikipedia.org/wiki/YouTube) or [eBay](https://en.wikipedia.org/wiki/EBay). It can also describe some actions controlled with certain [APIs](https://en.wikipedia.org/wiki/API) like [SMS](https://en.wikipedia.org/wiki/SMS). There are particular triggers and actions in each channel.
* **Triggers** are the "this" part of a recipe. They are the items that "trigger" the action. For example, from an [RSS feed](https://en.wikipedia.org/wiki/RSS_feed), you can receive a notification based on a keyword or phrase.
* **Actions** are the "that" part of a recipe. They are the output that results from the input of the trigger.
* **Recipes** are the [predicates](https://en.wikipedia.org/wiki/Branch_predication) made from Triggers and Actions. For example, if you like any picture in Instagram (trigger), the photo will be sent to your [Dropbox](https://en.wikipedia.org/wiki/Dropbox_(service)" \o "Dropbox (service)) account (action).
* **Ingredients** are basic data made available from a trigger. For example, the data that are available from the email trigger include subject, body, attachment, received date, and the sender’s address.

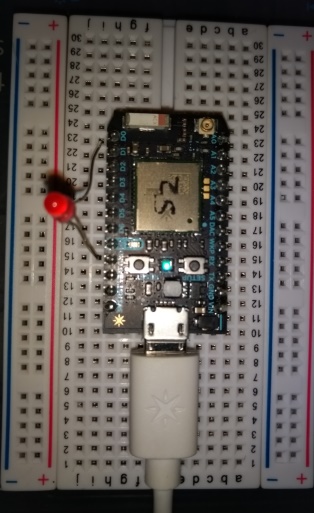
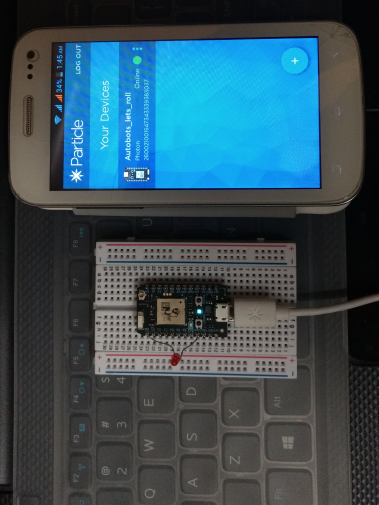
An example "recipe" might consist of sending an [e-mail](https://en.wikipedia.org/wiki/E-mail) message if the IFTTT user [tweets](https://en.wikipedia.org/wiki/Twitter) using a certain hashtag. Or, if the user is tagged by someone on Facebook, then that photo will be added to the user's [cloud-based](https://en.wikipedia.org/wiki/Cloud_storage) photo archive.

Block Diagram

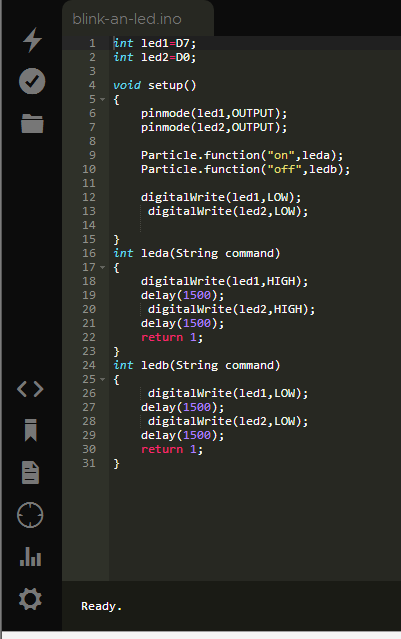


Circuit Diagram

Connect everything together as shown in the image below. The negative (shorter) pin of the LED is connected to ground and the positive (longer) pin is connected to D0.

 Application Code

The application code is loaded onto device. Copy and paste this code into a new application on http://build.particle.io or on Particle Dev. Save this application ,Verify the code, then flash it to the Photon. Now, the LED can be controlled.



Conclusion

The project has successfully completed the aim of controlling the LED features of switching on/off using digital code functions, with the help of Alexa voice command technology, cloud networking and embedded systems in synchronization.