**PROJECT REPORT ON:**

ALEXA VOICE COMMAND TO CONTROL DEVICES

**SUBMITTED TO:**

HED-X

**TEAM NAME:**

Verve

**TEAM MEMBERS:**

Lavanya.G

Mahalakshmi.P

Meghana.CS

Namratha.C

Navya.S

**Under the guidance of**

Surendra Parihar

**ABSTRACT:**

Amazon Alexa enables customers to interact with devices in a more intuitive way using voice. The Alexa Skills Kit is a collection of self-service APIs, tools, documentation and code samples that make it fast and easy for us to add skills to Alexa on devices. This project is a combination of Voice Technology, Cloud Networking and Embedded system. The device must be connected to particle photon and which is configured with wifi.By IFTTT recipe using ALEXA and particle cloud to control the device.

**REQUIREMENTS:**

This project mainly consists of following hardware and software:

1. Hardware:
   1. Android mobile / Tablet PC
   2. Particle Photon
   3. LED
   4. Bread board
2. Software:
   1. IFTTT Website Registration
   2. Particle cloud Registration
   3. AWS(ALEXA) Registration
   4. Echosim.io

**SYSTEM DESCRIPTION:**

In the default mode the device continuously listens to all speech, monitoring for the wake word to be spoken. The device also comes with a manually 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. Echo performs well with a 'good' (low latency) Internet connection which minimizes processing time due to minimal communication round trips, streamable responses and geo-distributed service endpoints.

The system consists of the Particle Photon board as its core. The device to be controlled (LED) is physically connected onto the board through either the digital or analog pins on the board.

The system is controlled using the IFTTT “Do-button” app, thereby allowing the user to control the appliance wireless over the internet.

To achieve this, first a function is declared in the particle cloud space. This function is then designed and developed to control the appliance connected with the board. Once that is done, the function is merely called using the Do-button app with the appropriate parameters.

**IFTTT** is a free web-based service that allows users to create chains of simple conditional statements, called "recipes", which are triggered based on changes to other web services such as Gmail, Facebook, Instagram, and Pinterest. IFTTT is an abbreviation of "If This Then That".

**TASKS THAT CAN BE PERFORMED USING DO BUTTON**

Post to Facebook with One Click

Call Yourself to Get Out of Awkward Situations

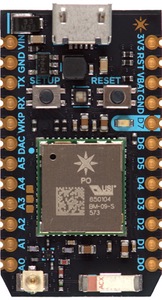
Instantly Add Tasks in Trello

Send Someone an SMS

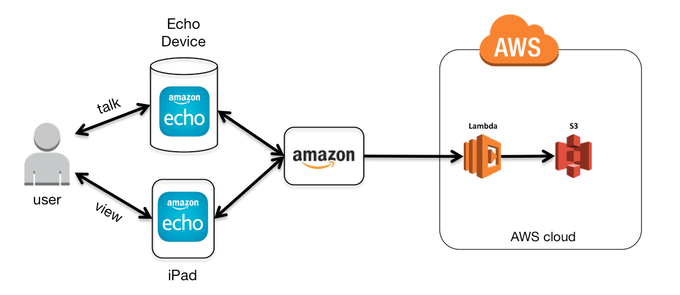
Send a Tweet

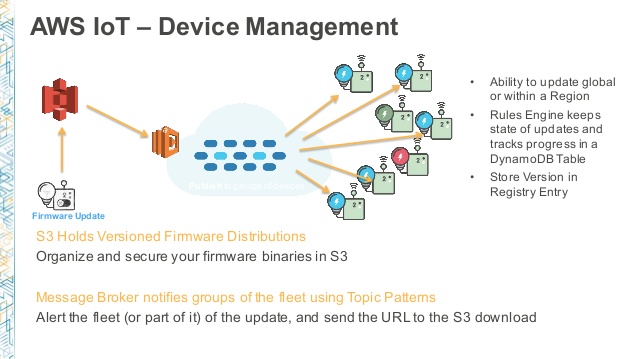
Play a Song

Post to WordPress

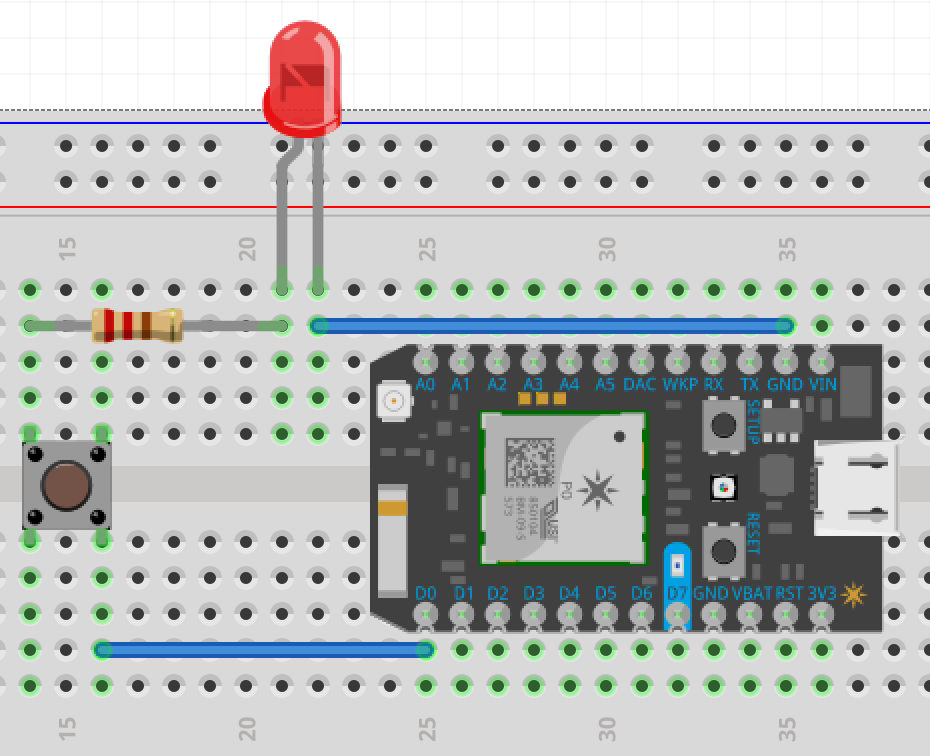
   

**BLOCK DIAGRAM:**





**CIRCUIT DIAGRAM:**



**SOURCE CODE:**

/\* Function prototypes -------------------------------------------------------\*/

int tinkerDigitalRead(String pin);

int tinkerDigitalWrite(String command);

int tinkerAnalogRead(String pin);

int tinkerAnalogWrite(String command);

/\* This function is called once at start up ----------------------------------\*/

void setup()

{

//Setup the Tinker application here

//Register all the Tinker functions

Spark.function("digitalread", tinkerDigitalRead);

Spark.function("digitalwrite", tinkerDigitalWrite);

Spark.function("analogread", tinkerAnalogRead);

Spark.function("analogwrite", tinkerAnalogWrite);

}

/\* This function loops forever --------------------------------------------\*/

void loop()

{

//This will run in a loop

}

int tinkerDigitalRead(String pin)

{

//convert ascii to integer

int pinNumber = pin.charAt(1) - '0';

//Sanity check to see if the pin numbers are within limits

if (pinNumber< 0 || pinNumber >7) return -1;

if(pin.startsWith("D"))

{

pinMode(pinNumber, INPUT\_PULLDOWN);

return digitalRead(pinNumber);

}

else if (pin.startsWith("A"))

{

pinMode(pinNumber+10, INPUT\_PULLDOWN);

return digitalRead(pinNumber+10);

}

return -2;

}

int tinkerDigitalWrite(String command)

{

bool value = 0;

//convert ascii to integer

int pinNumber = command.charAt(1) - '0';

//Sanity check to see if the pin numbers are within limits

if (pinNumber< 0 || pinNumber >7) return -1;

if(command.substring(3,7) == "HIGH") value = 1;

else if(command.substring(3,6) == "LOW") value = 0;

else return -2;

if(command.startsWith("D"))

{

pinMode(pinNumber, OUTPUT);

digitalWrite(pinNumber, value);

return 1;

}

else if(command.startsWith("A"))

{

pinMode(pinNumber+10, OUTPUT);

digitalWrite(pinNumber+10, value);

return 1;

}

else return -3;

}

int tinkerAnalogRead(String pin)

{

//convert ascii to integer

int pinNumber = pin.charAt(1) - '0';

//Sanity check to see if the pin numbers are within limits

if (pinNumber< 0 || pinNumber >7) return -1;

if(pin.startsWith("D"))

{

return -3;

}

else if (pin.startsWith("A"))

{

return analogRead(pinNumber+10);

}

return -2;

}

int tinkerAnalogWrite(String command)

{

//convert ascii to integer

int pinNumber = command.charAt(1) - '0';

//Sanity check to see if the pin numbers are within limits

if (pinNumber< 0 || pinNumber >7) return -1;

String value = command.substring(3);

if(command.startsWith("D"))

{

pinMode(pinNumber, OUTPUT);

analogWrite(pinNumber, value.toInt());

return 1;

}

else if(command.startsWith("A"))

{

pinMode(pinNumber+10, OUTPUT);

analogWrite(pinNumber+10, value.toInt());

return 1;

}

else return -2;

}

**CONCLUSION AND CHALLENGES:**

Challenges faced:

1. Understanding the concept of iot.

2. Usage of particle app.

With all the efforts and challenges we succeeded in getting the desired result of the project and successfully completed the project.

**THANK YOU☺**