**ALEXA VOICE COMMAND TO CONTROL HOME DEVICES**

**A PROJECT REPORT**

***Submitted by***

***THE INCREDIBLES***

**SANGEETA**

**RITHESH.M**

***PURCUING***

**BACHELOR OF ENGINEERING**

***In***

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**GLOBAL ACADEMY OF TECHNOLOGY, BANGALORE**

***Submitting to***

**Hed-x Projects**

**CONTENTS:**

**1.Abstract**

**2.Requirements**

**3.System description**

**4.Block diagram**

**5.Circuit diagram**

**6.Code**

**7.Conclusion and Challenges**

**8.References**

**ABSTRACT:**

The present technology is improving so fast that it provides millions of solution and ways to overcome the modern day problems in an easier and simpler manner. Process are being made to make the things smarter and easier to the consumer. Improving technology made humans to control their things in exciting way based on their perspection as world is becoming smarter and smarter. Here we have shown that a person can control devices at his home or office just by a voice command. The main concern in systems development is the integration of technologies to increase customer satisfaction. This project aims at controlling the devices at home through voice call by understanding the speech or voice of user. The user can make a voice call in order to perform certain actions such as switching lights on/off. Here we use particle photon, it controls the device and we give voice command in Amazon alexa.

**REQUIREMENTS:**

**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:**

Home automation is the use of one or more computers to control basic home functions and features automatically and sometimes remotely. An automated home is sometimes called as smart home. Home automation This Project is designed to control devices at home using Alexa Voice Command. Controlling devices at home or office plays an important role in human life when it comes to their standard of living this provides convenient and hassle free environment.

This project is a combination of Voice technology, Cloud Networking and Embedded system. Here one has to configure AWS account. On the other hand connect a device with photon and we should configure it with wifi. After that we should write a code to control device on particle cloud. Then on IFTTT Website creates a recipe using ALEXA and Particle cloud to control the device. Now by using echoism.io one can send a voice command which is already registered to control the device.

PARTICLE ACCOUNT

It helps to connect the photon to the wifi network. It also helps to read from the or write on to the pins of photon. The read and write can be performed in 2 ways either digital or analog.

Digital read and write takes only 2 values either low or high. Low is for 0 volts and high for maximum value of 255 volts

Analog read and write has values from 0 to 4095 and can take any intermediate values.

PARTICLE APP

The particle app is also used to build a particle by creating a new app and writing appropriate code to perform an specific action.

The code written is then flashed onto the photon and to perform specific action.

AWS.AMAZON

Aws.amazon and echoism account provides permission to access alexa voice command to control home devices.

IFTTT

Ifttt stands for if this than that. This account is used to create a reciepe using ifttt app or Do app.

In this a particular action triggers another particular or specific action.

Example : if gmail then facebook.

Which means if there is a mail received in an account then update a status on facebook.

Thus we can connect alexa and particle/photon using ifttt app.

PHOTON

### FEATURES

Particle PØ Wi-Fi module

Broadcom BCM43362 Wi-Fi chip

802.11b/g/n Wi-Fi

STM32F205RGY6 120Mhz ARM Cortex M3

1MB flash, 128KB RAM

On-board RGB status LED (ext. drive provided)

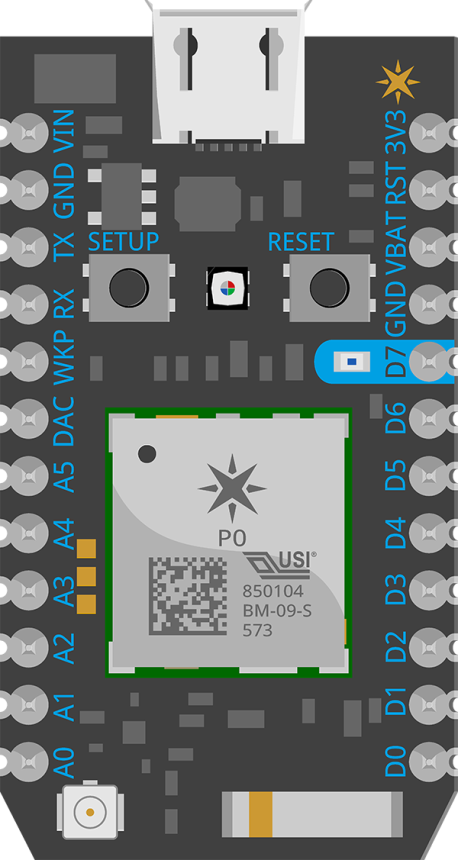
18 Mixed-signal GPIO and advanced peripherals

Open source design

Real-time operating system (FreeRTOS)

Soft AP setup

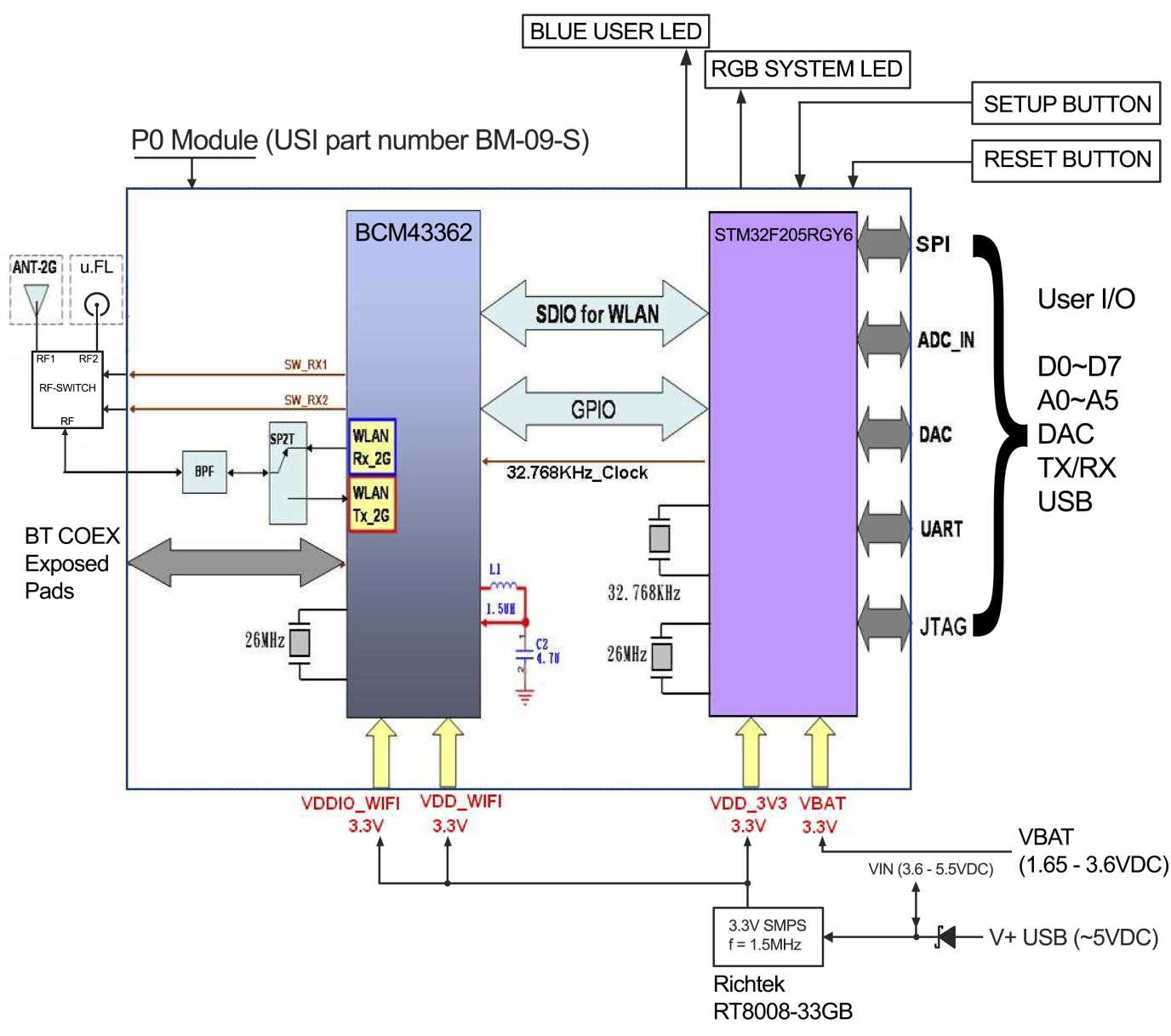
FCC, CE and IC c



Functional description

Interfaces

BLOCK DIAGRAM of photon



JTAG AND SWD

Pin D3 through D7 are JTAG interface pins. These can be used to reprogram your Photon bootloader or user firmware image with standard JTAG tools such as the ST-Link v2, J-Link, R-Link, OLIMEX ARM-USB-TINI-H, and also the FTDI-based Particle JTAG Programmer. If you are short on available pins, you may also use SWD mode which requires less connections.

| **Photon Pin** | **JTAG** | **SWD** | **STM32F205RGY6 Pin** | **PØ Pin #** | **PØ Pin Name** | **Default Internal[1]** |
| --- | --- | --- | --- | --- | --- | --- |
| D7 | JTAG\_TMS | SWD/SWDIO | PA13 | 44 | MICRO\_JTAG\_TMS | ~40k pull-up |
| D6 | JTAG\_TCK | CLK/SWCLK | PA14 | 40 | MICRO\_JTAG\_TCK | ~40k pull-down |
| D5 | JTAG\_TDI |  | PA15 | 43 | MICRO\_JTAG\_TDI | ~40k pull-up |
| D4 | JTAG\_TDO |  | PB3 | 41 | MICRO\_JTAG\_TDO | Floating |
| D3 | JTAG\_TRST |  | PB4 | 42 | MICRO\_JTAG\_TRSTN | ~40k pull-up |
| 3V3 | Power | Power |  |  |  |  |
| GND | Ground | Ground |  |  |  |  |
| RST | Reset | Reset |  |  |  |  |

PIN DESCRIPTION

| **Pin** | **Description** |
| --- | --- |
| VIN | This pin can be used as an input or output. As an input, supply 3.6 to 5.5VDC to power the Photon. When the Photon is powered via the USB port, this pin will output a voltage of approximately 4.8VDC due to a reverse polarity protection series schottky diode between VUSB and VIN. When used as an output, the max load on VIN is 1A. |
| RST | Active-low reset input. On-board circuitry contains a 1k ohm pull-up resistor between RST and 3V3, and 0.1uF capacitor between RST and GND. |
| VBAT | Supply to the internal RTC, backup registers and SRAM when 3V3 is not present (1.65 to 3.6VDC). |
| 3V3 | This pin is the output of the on-board regulator and is internally connected to the VDD of the WiFi module. When powering the Photon via VIN or the USB port, this pin will output a voltage of 3.3VDC. This pin can also be used to power the Photon directly (max input 3.3VDC). When used as an output, the max load on 3V3 is 100mA. NOTE: When powering the Photon via this pin, ensure power is disconnected from VIN and USB. |
| WKP | Active-high wakeup pin, wakes the module from sleep/standby modes. When not used as a WAKEUP, this pin can also be used as a digital GPIO, ADC input or PWM. |
| D0~D7 | Digital only GPIO pins. D0~D3 may also be used as a PWM output. |
| A0~A7 | 12-bit Analog-to-Digital (A/D) inputs (0-4095), and also digital GPIOs. A6 and A7 are code convenience mappings, which means pins are not actually labeled as such but you may use code like analogRead(A7).A6 maps to the DAC pin and A7 maps to the WKP pin. A4,A5,A7 may also be used as a PWM output. |
| DAC | 12-bit Digital-to-Analog (D/A) output (0-4095), and also a digital GPIO. DAC is used as DAC or DAC1 in software, and A3 is a second DAC output used as DAC2 in software. |
| RX | Primarily used as UART RX, but can also be used as a digital GPIO or PWM. |
| TX | Primarily used as UART TX, but can also be used as a digital GPIO or PWM. |

**BLOCK DIAGRAM:**

Voice command

External devices

Particle build

IFTTT

Recipes

AWS Amazon

echoism.io

Photon

STEPS TO BUILD A VOICE CONTROLLED DEVICE USING ALEXA

* Create account in particle.io, ifttt.com and echoism.io.
* Download the particle and ifttt app in smart phone.
* connect the photon to system and set the photon so that it turns to blue.
* Connect this photon to wifi through particle app.
* When photon is connected to wifi network it turns cyan colour.
* Now login to particle.io and select build and create new app and build a suitable code
* CODE

// ------------

// Blink an LED

// ------------

/\*-------------

We've heavily commented this code for you. If you're a pro, feel free to ignore it.

Comments start with two slashes or are blocked off by a slash and a star.

You can read them, but your device can't.

It's like a secret message just for you.

Every program based on Wiring (programming language used by Arduino, and Particle devices) has two essential parts:

setup - runs once at the beginning of your program

loop - runs continuously over and over

You'll see how we use these in a second.

This program will blink an led on and off every second.

It blinks the D7 LED on your Particle device. If you have an LED wired to D0, it will blink that LED as well.

-------------\*/

// First, we're going to make some variables.

// This is our "shorthand" that we'll use throughout the program:

int led1 = D0; // Instead of writing D0 over and over again, we'll write led1

// You'll need to wire an LED to this one to see it blink.

int led2 = D7; // Instead of writing D7 over and over again, we'll write led2

// This one is the little blue LED on your board. On the Photon it is next to D7, and on the Core it is next to the USB jack.

// Having declared these variables, let's move on to the setup function.

// The setup function is a standard part of any microcontroller program.

// It runs only once when the device boots up or is reset.

void setup() {

Particle.function("led", ledtoggle);

// We are going to tell our device that D0 and D7 (which we named led1 and led2 respectively) are going to be output

// (That means that we will be sending voltage to them, rather than monitoring voltage that comes from them)

// It's important you do this here, inside the setup() function rather than outside it or in the loop function.

pinMode(led1, OUTPUT);

pinMode(led2, OUTPUT);

}

// Next we have the loop function, the other essential part of a microcontroller program.

// This routine gets repeated over and over, as quickly as possible and as many times as possible, after the setup function is called.

// Note: Code that blocks for too long (like more than 5 seconds), can make weird things happen (like dropping the network connection). The built-in delay function shown below safely interleaves required background activity, so arbitrarily long delays can safely be done if you need them.

void loop() {

}

int ledtoggle(String command)

{

// To blink the LED, first we'll turn it on...

digitalWrite(led1, HIGH);

digitalWrite(led2, HIGH);

// Wait...

delay(1000);

// Then we'll turn it off...

digitalWrite(led1, LOW);

digitalWrite(led2, LOW);

// Wait...

delay(500);

digitalWrite(led1, HIGH);

digitalWrite(led2, HIGH);

// Wait...

delay(2000);

// Then we'll turn it off...

digitalWrite(led1, LOW);

digitalWrite(led2, LOW);

// Wait...

delay(300);

digitalWrite(led1, HIGH);

digitalWrite(led2, HIGH);

// Wait...

delay(3000);

// Then we'll turn it off...

digitalWrite(led1, LOW);

digitalWrite(led2, LOW);

// Wait...

delay(1000);

return 0;

}

* once the code is written save, compile and flash it on the photon
* After flashing it on photon login to ifttt app.
* Select my reciepe and go to create reciepe.
* “IF THIS THEN THAT” phrase appears, click on “this” and select alexa.
* Click on connect and sign in to amazon account then continue to next to next step.
* Select a say a specific phrase and write the phrase by which the device is controlled.

Example: “led on”.

* Now click on trigger and click on “That” phrase.
* Select particle and click on connect
* The program is coded using a call function hence select call a function icon.
* Select name of the device and click on create action and the reciepe is created.
* Log in to echosim.io and say the phrase that was mention in the reciepe prefixed with alexa trigger. Example: “alexa trigger led on”
* The alexa sends the command to ifttt which is connected to photon and the code works.

CHALLENGES AND CONCLUSIONS

The main aim of the project is that it improves, leading life in smarter way and also saves time and energy. Very interesting thing about this project is that by sitting in any corner at our house, we can control the world.

We can switch on the lights, fan, use washing machine and many other devices at home and outside world by just ordering. This technique can be made even more better by connecting with some other device, for example cellphone and washing machine are connected to the alexa we can control washing machine just by giving instruction(voice) through cell phone .