

ESP32, Blynk et Assistant vocal

L'objectif est de commander vocalement une sortie d'un ESP32. L'idée est d'utiliser des technologies [no-code/low-code](#) comme [Blynk](#) et [IFTTT](#).



Le moyen le plus simple d'interagir vocalement avec l'ESP32 via [Blynk](#) est d'utiliser [IFTTT](#). [IFTTT](#) (*If This Then That*) est un service web gratuit permettant de créer des chaînes d'instruction simples appelées *applets*. Un *applet* intègre un déclencheur (*This*) lié à une action (*That*). [IFTTT](#) fournit un grand nombre de déclencheurs et d'actions.

If This Then That

Les déclencheurs [IFTTT](#) pour assurer une commande vocale sont : Google Assistant et Alexa. Il faudra donc trouver une solution pour Siri en utilisant Webhooks.



My services



Amazon Alexa



Google Assistant



Webhooks



Autre possibilités

- [Alexa et FauxmoESP avec l'ESP32](#)
- [AWS IoT et Alexa avec l'ESP32](#)
- [ESP32 et Google Cloud IoT + Assistant](#)

Blynk

[Mise en oeuvre de Blynk](#)

[Blynk](#) fournit une [API HTTP RESTful](#) qui permet de piloter les entrées/sorties de l'ESP32 (ou tout autre système embarqué pris en charge par Blynk comme Arduino, Raspberry Pi, ESP8266, etc ...). Il est possible d'utiliser des requêtes `GET` et/ou `PUT` :

- Requête `GET` pour lire l'état d'une broche (ici `D1`) :
`http://blynk-cloud.com/auth_token/get/D1`
- Requête `GET` pour écrire sur une broche (ici `D1`) :
`http://blynk-cloud.com/auth_token/update/D1?value=1`
- Requête `PUT` pour écrire sur une broche (ici `D1`) :
`http://blynk-cloud.com/auth_token/update/D1`

Tests (remplacer `XXXXXXXX` par le *token* d'authentification) :

```
$ curl -X GET http://blynk-cloud.com/XXXXXXXX/update/D19?value=1
$ curl -X GET http://blynk-cloud.com/XXXXXXXX/update/D19?value=1

$ curl -X PUT -H "Content-Type: application/json" -d '{"1"}' http://blynk-cloud.com/XXXXXXXX/upda
te/D19
$ curl -X PUT -H "Content-Type: application/json" -d '{"0"}' http://blynk-cloud.com/XXXXXXXX/upda
te/D19
```

On utilisera cet [API HTTP RESTful](#) avec le service [Webhooks d'IFTTT](#) pour piloter l'ESP32 via une requête web (*Make a web request*).

Par contre, il faudra prendre l'adresse du serveur [Blynk](#) `blynk-cloud.com` (car il utilise différents serveurs pour différents pays) :

```
$ ping -c 1 blynk-cloud.com
PING blynk-cloud.com (139.59.206.133) 56(84) bytes of data.
64 bytes from 139.59.206.133 (139.59.206.133): icmp_seq=1 ttl=50 time=41.2 ms

--- blynk-cloud.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 41.218/41.218/41.218/0.000 ms
```

Exemple ESP32 avec [PlatformIO](#) :

- `platform.ini` pour une carte ESP32 wemos et la bibliothèque [Blynk](#) :

```
[env:lolin_d32]
platform = espressif32
board = lolin_d32
framework = arduino
lib_deps =
  blynkkk/Blynk @ ^0.6.7
```

- `main.cpp` :

```
#include <Arduino.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

//#define BLYNK_PRINT Serial

char auth[] = "";
char ssid[] = "";
char pass[] = "";

void setup()
{
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
}

void loop()
{
  Blynk.run();
}
```

Google Assistant

Le moyen le plus simple d'interagir avec l'ESP32 via Google Assistant est d'utiliser [IFTTT](#).

[IFTTT](#) est un service web gratuit permettant de créer des chaînes d'instruction simples appelées *applets*.

On va par exemple créer un *applet* avec un déclencheur sur le service Google Assistant (*This*) avec une action Webhooks vers l'[API HTTP RESTful](#) de [Blynk](#) (*That*) :

Create your own

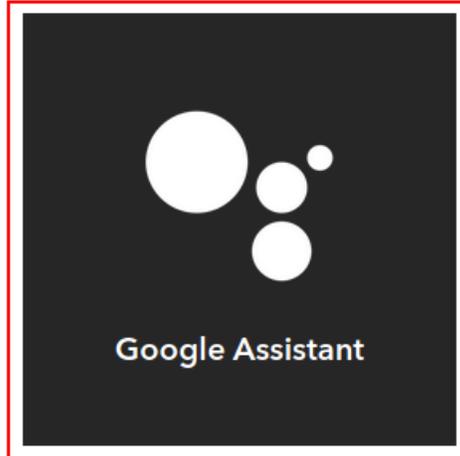
If  This Then That

Il faut choisir le service Google Assistant :

Choose a service

Step 1 of 6

🔍 google assistant



On sélectionne "Say a simple phrase" :



Choose trigger

Step 2 of 6

Say a simple phrase

This trigger fires when you say "Ok Google" to the Google Assistant followed by a phrase you choose. For example, say "Ok Google, I'm running late" to text a family member that you're on your way home.

Say a phrase with a number

This trigger fires when you say "Ok Google" to the Google Assistant followed by a phrase like "Set Nest thermostat to 68." **Use the # symbol to specify where you'll say the number ingredient

Say a phrase with a text ingredient

This trigger fires when you say "Ok Google" to the Google Assistant followed by a phrase like "Post a tweet saying 'New high score.'" **Use the \$ symbol to specify where you'll say the text ingredient

Say a phrase with both a number and a text ingredient

This trigger fires when you say "Ok Google" to the Google Assistant followed by a phrase like "Block time for 'exercise' at 6 PM." **Use the # symbol to specify where you'll say the number ingredient and \$ where you'll say the text ingredient

On complète :



Complete trigger fields

Step 2 of 6

What do you want to say?

Allume la led

What's another way to say it?
(optional)

Allume led

And another way? (optional)

What do you want the Assistant to say
in response?

Ok

Language

French ▼

Create trigger

On va ajouter le *That* :

If  **Then**  **That**

On sélectionne Webhooks :



Connect Webhooks

Step 3 of 6

Integrate other services on IFTTT with your DIY projects. You can create Applets that work with any device or app that can make or receive a web request. If you'd like to build your own service and Applets, check out the IFTTT platform.

Connect

On sélectionne "Make a web request" :



Choose action

Step 4 of 6

Make a web request

This action will make a web request to a publicly accessible URL. NOTE: Requests may be rate limited.

On paramètre l'action :



Complete action fields

Step 5 of 6

URL

Surround any text with "<<>>" to escape the content. [Add ingredient](#)

Say a simple phrase

CreatedAt

The method of the request e.g. GET, POST, DELETE

Content Type

Please select

Optional

Body

Par contre, il faudra prendre l'adresse du serveur Blynk `blynk-cloud.com` (car il utilise différents serveurs pour différents pays) :

```
$ ping -c 1 blynk-cloud.com
PING blynk-cloud.com (139.59.206.133) 56(84) bytes of data.
64 bytes from 139.59.206.133 (139.59.206.133): icmp_seq=1 ttl=50 time=41.2 ms

--- blynk-cloud.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 41.218/41.218/41.218/0.000 ms
```

- Pour une requête `GET` :

URL

```
http://139.59.206.133  
/E100v0010mp0f9k66100v0010mp0  
/update/D19?value=1
```

Surround any text with <<< and >>> to escape the content

Add ingredient

Method

GET



The method of the request e.g. GET, POST, DELETE

- Pour une requête `PUT` :

URL

```
http://139.59.206.133  
/E100v0010mp0f9k66100v0010mp0  
/update/D19
```

Surround any text with <<< and >>> to escape the content

Add ingredient

Method

PUT



The method of the request e.g. GET, POST, DELETE

Content Type (optional)

application/json



Optional

Body (optional)

```
["1"]
```

Surround any text with <<< and >>> to escape the content

Add ingredient

On termine le processus de création :

Review and finish

Step 6 of 6



If You say "Allume la led", then Make a web request

51/140

by **thierryvaira**

Receive notifications
when this Applet runs



Finish

On vérifie l'état `connected` :



If You say "Allume la led", then Make a web request

by thierryvaira

Applets

Applets connect two or more services together and help you do something that you couldn't do with just one service alone. [Learn more](#)

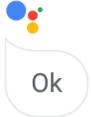
Got it

Connected

Test :



allume la led

[Conditions d'utilisation](#) et [Règles de confidentialité](#)

Plus de fonctionnalités de l'Assistant

PREMIERS PAS

Alexa

Un moyen simple d'interagir avec l'ESP32 via Alexa est d'utiliser [IFTTT](#).

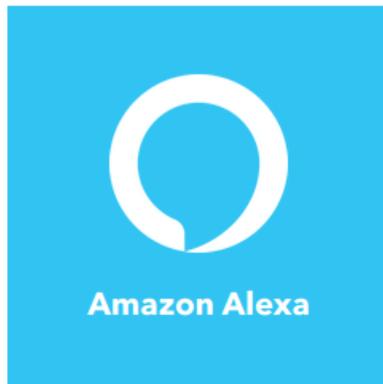
[IFTTT](#) est un service web gratuit permettant de créer des chaînes d'instruction simples appelées *applets*.

On va par exemple créer un *applet* avec un déclencheur sur le service Alexa (*This*) avec une action Webhooks vers l'[API HTTP RESTful](#) de [Blynk](#) (*That*) :

Il faut choisir le service Alexa :

Choose a service

Q Alexa



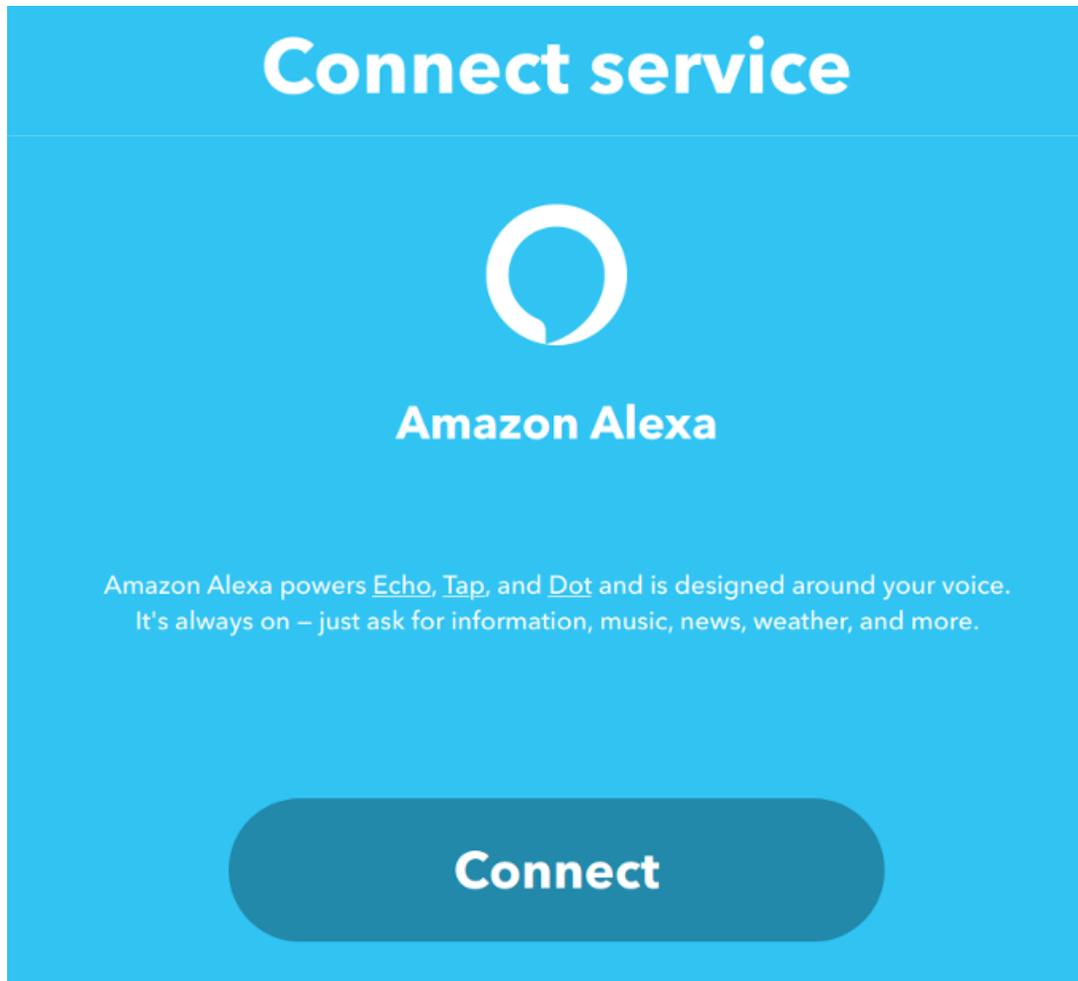
On sélectionne "Say a specific phrase" :

Say a specific phrase

Choose trigger: Say a

This trigger fires every time you say "Alexa trigger" + the phrase that you have defined. For instance, if you set "party time" as the phrase, you can say "Alexa trigger party time" to have your lights loop colors. Please use lower-case only.

Il faut se connecter au service Alexa :



Connect service



Amazon Alexa

Amazon Alexa powers [Echo](#), [Tap](#), and [Dot](#) and is designed around your voice. It's always on – just ask for information, music, news, weather, and more.

Connect

On autorise son compte Alexa :



IFTTT souhaite avoir accès à:

 IFTTT

Nom: Thierry VAIRA

Adresse e-mail: thierry.vaira@orange.fr

Accédez à vos informations Alexa (comme les listes des choses à faire) pour déclencher les recettes que vous créez.

L'autorisation de publier les informations sur Alexa, de les écrire et de les modifier pour préparer les recettes que vous créez.

Autoriser

On va ajouter le *That* en sélectionnant Webhooks :



Step 3 of 6

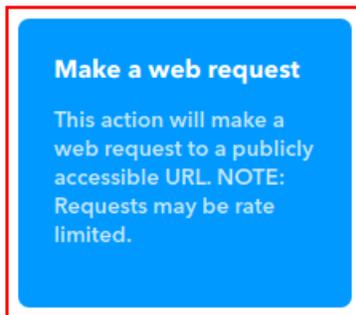
Integrate other services on IFTTT with your DIY projects. You can create Applets that work with any device or app that can make or receive a web request. If you'd like to build your own service and Applets, check out the IFTTT platform.



On sélectionne "*Make a web request*" :



Step 4 of 6



On paramètre l'action :



Complete action fields

Step 5 of 6

URL

Surround any text with "<<>>" to escape the content. **Add ingredient**

Say a simple phrase **CreatedAt**

The method of the request e.g. GET, POST, DELETE

Content Type

Please select ▼

Optional

Body

Surround any text with "<<>>" to escape the content. **Add ingredient**

Create action

Par contre, il faudra prendre l'adresse du serveur Blynk `blynk-cloud.com` (car il utilise différents serveurs pour différents pays) :

```
$ ping -c 1 blynk-cloud.com
PING blynk-cloud.com (139.59.206.133) 56(84) bytes of data.
64 bytes from 139.59.206.133 (139.59.206.133): icmp_seq=1 ttl=50 time=41.2 ms

--- blynk-cloud.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 41.218/41.218/41.218/0.000 ms
```

- Pour une requête `GET` :

URL

```
http://139.59.206.133  
/E100v0010mp0f9k66100v0010mp0  
/update/D19?value=1
```

Surround any text with <<< and >>> to escape the content

Add ingredient

Method

GET



The method of the request e.g. GET, POST, DELETE

- Pour une requête `PUT` :

URL

```
http://139.59.206.133  
/E100v0010mp0f9k66100v0010mp0  
/update/D19
```

Surround any text with <<< and >>> to escape the content

Add ingredient

Method

PUT



The method of the request e.g. GET, POST, DELETE

Content Type (optional)

application/json



Optional

Body (optional)

```
["1"]
```

Surround any text with <<< and >>> to escape the content

Add ingredient

On termine le processus de création et on teste.

Siri

Siri n'est pas disponible directement dans [IFTTT](#). Une solution simple est d'utiliser l'application [Raccourcis](#) pour iOS ce qui permettra d'envoyer une requête web vers un déclencheur Webhooks d'[IFTTT](#).

[IFTTT](#) est un service web gratuit permettant de créer des chaînes d'instruction simples appelées *applets*.

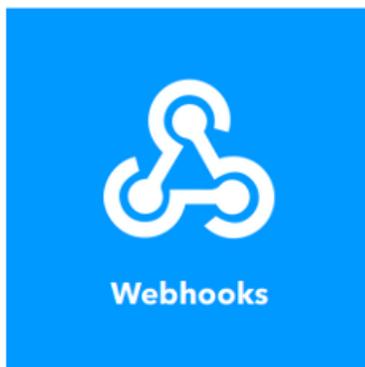
On va par exemple créer un *applet* avec un déclencheur Webhooks (*This*) avec une action Webhooks vers l'[API HTTP RESTful](#) de [Blynk](#) (*That*) :



Il faut choisir le service Webhooks :

Choose a service

Q Webhooks|



Choose a trigger



Webhooks

Receive a web request

This trigger fires every time the Maker service receives a web request to notify it of an event. For information on triggering events, go to your Maker service settings and then the listed URL (web) or tap your username (mobile)

On crée l'évènement déclencheur :

Complete trigger fields



Receive a web request

This trigger fires every time the Maker service receives a web request to notify it of an event. For information on triggering events, go to your Maker service settings and then the listed URL (web) or tap your username (mobile)

Event Name

led_rouge

The name of the event, like "button_pressed" or "front_door_opened"

Create trigger

On va ajouter ensuite le *That* :



Sélectionner Webhooks :



Step 3 of 6

Integrate other services on IFTTT with your DIY projects. You can create Applets that work with any device or app that can make or receive a web request. If you'd like to build your own service and Applets, check out the IFTTT platform.

Connect

On sélectionne "*Make a web request*" :



Choose action

Step 4 of 6

Make a web request

This action will make a web request to a publicly accessible URL. NOTE: Requests may be rate limited.

On paramètre l'action :



Complete action fields

Step 5 of 6

URL

Surround any text with "<<>>" to escape the content **Add ingredient**

CreatedAt

Say a simple phrase

The method of the request e.g. GET, POST, DELETE

Content Type

Please select

Optional

Body

Surround any text with "<<>>" to escape the content **Add ingredient**

Create action

Par contre, il faudra prendre l'adresse du serveur Blynk `blynk-cloud.com` (car il utilise différents serveurs pour différents pays) :

```
$ ping -c 1 blynk-cloud.com
PING blynk-cloud.com (139.59.206.133) 56(84) bytes of data.
64 bytes from 139.59.206.133 (139.59.206.133): icmp_seq=1 ttl=50 time=41.2 ms

--- blynk-cloud.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 41.218/41.218/41.218/0.000 ms
```

- Pour une requête `GET` :

URL

http://139.59.206.133
/E1005v00101mp0f9k661005v00101mp0
/update/D19?value=1

Surround any text with <<< and >>> to escape the content

Add ingredient

Method

GET ▼

The method of the request e.g. GET, POST, DELETE

- Pour une requête `PUT` :

My services



Amazon Alexa



Google Assistant



Webhooks



Et cliquer sur `Documentation` :

Il faut récupérer sa clé (*key*) et surtout l'URL en faisant copier/coller :



Your key is: [REDACTED]

◀ Back to service

To trigger an Event

Make a POST or GET web request to:

`https://maker.ifttt.com/trigger/{event}/with/key/[REDACTED]`

L'URL `https://maker.ifttt.com/trigger/{event}/with/key/XXXXXXXX` est nécessaire pour le raccourci que l'on va créer ensuite.

Test :

```
$ curl -X POST https://maker.ifttt.com/trigger/{event}/with/key/XXXXXXXX
```

Il faut installer l'application [Raccourcis](#) depuis l'*Apple Store* :

Aperçu App Store

Cette app est disponible uniquement dans l'App Store pour iPhone et iPad.



Raccourcis 4+

Apple

N° 73 en Productivité

★★★★★ 4,2 • 778 notes

Gratuit

On crée un nouveau raccourci :

Bibliothèque

Rechercher

The screenshot shows the iOS Shortcuts app library. It features a grid of shortcuts: 'Shazam shortcut' (blue), 'What's a shortcut?' (purple), 'Take a Break' (orange), 'Text Last Image' (green), 'Allume la led' (blue, highlighted with a red box), and 'Créer un raccourci' (white with a blue border). Each shortcut has a three-dot menu icon in the top right corner.

Il faut commencer par ajouter une action "Texte" et coller l'URL `https://maker.ifttt.com/trigger/{event}/with/key/XXXXXXXX` en remplaçant `{event}` par le nom de l'évènement créé dans IFTTT.

Il faut ensuite ajouter simplement l'action "Obtenir le contenu de l'URL".



Dans les réglages du raccourci, il est possible de :

- nommer le raccourci
- l'ajouter aux widgets
- ajouter une phrase Siri "Ajouter à Siri" pour personnaliser la commande vocale
- l'ajouter à l'écran d'accueil

Terminer par et tester.