Tutoriel ESP32

Présentation et Installation

Réalisé par Justin DARNET
Contributeur LaTeX : Florent Chehab

5 juin 2020

Ce guide vous permettra de découvrir la carte électronique connectée ESP32 et de savoir quand, pourquoi et comment l'utiliser.

Introduction

A mesure que le monde qui nous entoure tend à être toujours plus connecté, beaucoup de nouveaux systèmes intelligents apparaissent. Cette révolution du numérique se traduit par la mise à disposition auprès du particulier de plusieurs outils connectés. Le principal objectif de ces objets est d'obtenir l'ensemble des informations que l'utilisateur cherche en quelques secondes seulement. On peut aussi relever les nombreux progrès de la domotique ces dernières années.

Parlons domotique ! Le principe peut être très simple. Dès lors que l'on trouve le moyen de piloter électriquement un appareil, par un relais par exemple pour les systèmes fonctionnant en tout ou rien, il suffit ensuite de piloter ce relais avec une carte connectée au WiFi domestique ; on peut alors contrôler l'activation de l'objet depuis n'importe quel endroit sur terre à condition d'avoir un accès à internet. L'idée est ici de séparer la partie puissance de la partie intelligence/commande. Libre à l'utilisateur ensuite d'ajouter un capteur de présence dans une pièce ou un capteur de luminosité par exemple.

De nombreuses solutions coûteuses et "sécurisées" sont aujourd'hui sur le marché, mais il est tout à fait possible de réaliser ce type d'objet pour quelques euros et un peu de temps personnel.

Cette série de tutoriels sur l'ESP32 va vous familiariser avec l'«IoT» (*Internet of Things*), qui vise à acquérir des données et commander différents objets à distance notamment, en ajoutant de l'intelligence au système. Vous verrez ainsi comment fabriquer votre propre projet IoT.

Bonne lecture, soyez toujours conscients des risques en électronique et assurez-vous de bien connaître votre circuit avant de le tester. Pensez à vous protéger afin de bricoler en toute sécurité !

Table des matières

A Présentation de l'ESP32

L'ESP32 est une petite carte électronique, appelée microcontrôleur, facile à prendre en main grâce à ses ressemblances avec la carte Arduino qui est bien plus répandue. L'ESP32 peut avoir différentes tailles mais la version la plus courante est la suivante :

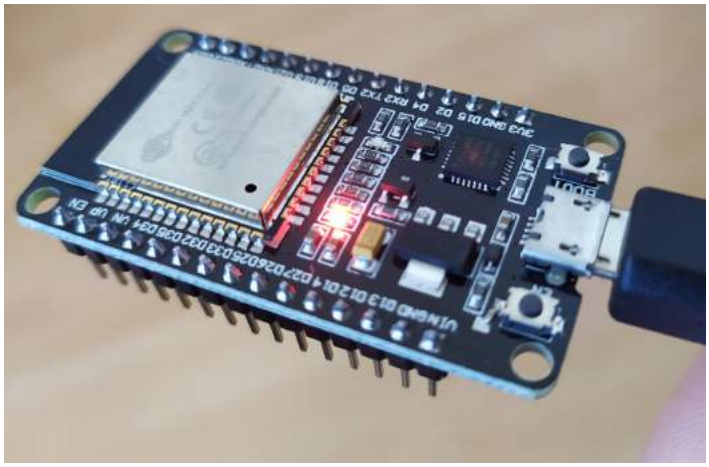


FIGURE 1 – La carte ESP32 DEVKIT WROOM

? Microcontrôleur, Quésako ?

La carte microcontrôleur grand public la plus répandue et très «user friendly» est aujourd'hui la carte Arduino. Plusieurs tutoriels sont aujourd'hui disponibles dans la base des tutoriels FabLab UTC pour utiliser cette carte dans vos projets.

Les cartes microcontrôleur sont des petits circuits dotés d'entrées/sorties programmables. Elles permettent l'acquisition de données provenant de capteurs, l'envoi de signaux de données ou le contrôle de circuits de puissance. De nombreuses utilisations sont possibles grâce à des modules complémentaires qui se branchent sur les broches de la carte (capteurs, leds, écrans, etc.). En somme, la principale limite de ce type de microcontrôleur est l'imagination de l'utilisateur (et parfois la puissance de la carte, bien entendu).

A.1 Utilité de l'ESP32

Comme mentionné précédemment, l'ESP32 est une carte électronique permettant de réaliser des projets «IoT» assez facilement. Elle possède en effet une connectivité assez complète, que nous détaillerons dans la partie suivante. L'ESP32 est assez simple à prendre en main car elle est cousine de la carte Arduino. Il est donc possible de l'utiliser comme la carte microcontrôleur italienne, en utilisant le même langage de programmation, les mêmes modules complémentaires et surtout le même logiciel de compilation (Arduino IDE). Pas de changement fondamental d'interface pour l'utilisateur, ce qui la rend d'autant plus simple à utiliser.

Par ailleurs, on peut relever que sa taille miniature lui est un avantage considérable : elle mesure en effet moins de 3 cm par 5 cm, malgré les technologies qu'elle embarque ! L'ESP32 est sans doute la carte microcontrôleur qui possède le meilleur compromis taille / connectivité / entrées sorties. Cela en fait un outil redoutable pour la miniaturisation des projets connectés

Pour finir elle est aussi tout à fait adaptée à la réalisation de prototypes et non uniquement à la réalisation de projets «finis et définitifs». Elle est une référence dans l'univers maker pour ces nombreuses raisons, il est donc très facile de trouver de l'aide sur Internet pour utiliser cette carte.

Entrons donc dans une étude plus détaillée à présent.

A.2 Connectivité

L'objet «IoT» est très intéressant, cependant il nécessite un accès à Internet afin de pouvoir assurer une communication permanente avec l'utilisateur. L'ESP32 est en ce sens une solution intéressante car elle contient au sein de son architecture un module WiFi intégré. Dans cette mesure, et donc sans avoir besoin de monopoliser des broches en ajoutant un module WiFi externe, l'utilisateur va être capable de connecter son objet IoT à Internet grâce à trois lignes de codes (pour les réseaux domestiques classiques).

Un second atout majeur de l'ESP32 est sa puce Bluetooth intégrée, qui peut aussi être utilisée en mode «BLE» (*Bluetooth Low Energy*) consommant moins d'énergie. La consommation d'énergie est en effet un critère important dans la conception d'objets IoT. Ils requièrent une connexion quasi permanente à Internet et doivent donc rester allumés ou en état de veille à chaque instant afin de pouvoir recevoir les instructions de l'utilisateur. Pour des raisons écologiques et économiques il est ainsi fondamental de réaliser un objet qui consomme le moins d'énergie que possible en état de veille, d'où l'utilité de la technologie BLE notamment.

Côté entrées et sorties, l'ESP32 est semblable aux cartes Arduino et aux cartes Raspberry. On relève en effet l'utilisation des ports GPIO (*General Purpose Input/Output*) qui sont les broches permettant de servir soit d'entrée, soit de sortie selon le programme téléversé.

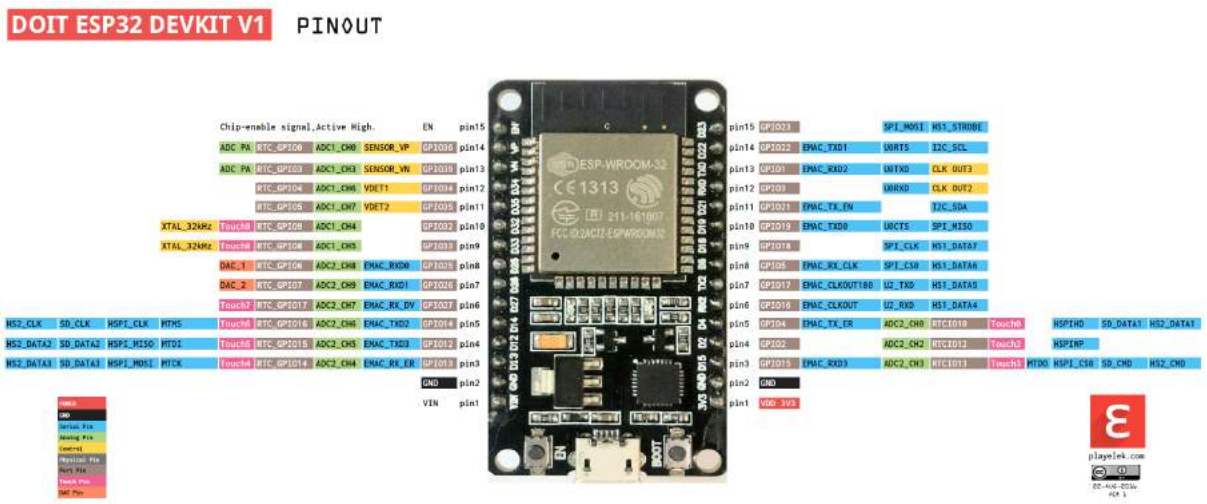


FIGURE 2 – Broches de l'ESP32 DEVKIT

B Comparatif avec d'autres cartes similaires

B.1 ESP8266

L'ESP8266 est, comparée à l'ESP32, une carte minimaliste. L'idée étant de contenir le WiFi et un microcontrôleur sur une seule et même carte. Le nombre d'entrées et sorties est assez limité mais néanmoins suffisant pour réaliser de petits objets «IoT» basiques.

La connectique est simpliste : la carte ne peut être programmée ou alimentée directement via l'ordinateur en USB. Il sera nécessaire d'utiliser une carte Arduino par exemple pour programmer l'ESP8266.

La carte contient très peu de ports GPIO, seul un port analogique est disponible ce qui restreint beaucoup le nombre de capteurs utilisables avec cette carte.

Pour quel type d'utilisation ?

Il est préférable d'utiliser l'ESP8266 pour **des projets «IoT» miniatures, à contrôler par WiFi uniquement, comportant très peu de capteurs** (dont maximum 1 capteur analogique) **et destinés à rester en place longtemps.**

Exemples. *Lampe connectée, Thermomètre connecté, Détecteur de mouvement connecté*

B.2 Arduino UNO

La carte Arduino est la référence en terme de carte microcontrôleur. Elle est très adaptée à la réalisation de projets en tous genres comportant de nombreux capteurs et actionneurs. Elle se décline en plusieurs tailles : l'Arduino Micro, l'Arduino Nano, l'Arduino Uno et l'Arduino Mega. Chez Arduino, plus la carte est grande, plus elle contient de ports, de mémoire et en général, de puissance.

Si l'ESP32 a des dimensions proches de l'Arduino Nano, elle n'a cependant pas les mêmes spécificités. L'ESP (32 ou 8266) est vouée à ajouter de la connectivité aux projets des utilisateurs. On y retrouve notamment le WiFi et/ou le Bluetooth intégré, cependant les cartes Arduino doivent recevoir un module complémentaire pour bénéficier de ce type de connexion. L'ajout d'un module augmente le coût et l'encombrement du circuit, tout en monopolisant des ports sur la carte (ressource rare quand on souhaite avoir un encombrement minimal !).

Pour quel type d'utilisation ?

Il est préférable d'utiliser la carte Arduino pour **des projets «IoT» assez volumineux et comportant plusieurs capteurs et actionneurs.** L'Arduino semble plus adaptée pour du prototypage que pour la réalisation d'un objet «IoT» final en tant que tel.

Exemples. *Prototypage général, Robots, Systèmes électroniques relativement complexes*

B.3 Raspberry Pi

La carte Raspberry Pi (RPi) existe elle aussi sous plusieurs déclinaisons et est disponible sous plusieurs générations (nous sommes aujourd'hui à la RPi 4). Qui dit nouvelle génération dit aussi nouvelles technologies, dans cette mesure les spécifications de la RPi sont toujours plus grandes. Cette carte est en elle-même un ordinateur miniature. Elle comporte ainsi plus de 512 Mb de mémoire RAM (pour les moins puissantes), un port HDMI, une entrée pour caméra, des ports USB,... Bref ! Elles sont de réels ordinateurs qui évoluent sur un système d'exploitation Raspbian (dérivé du populaire Debian de la plateforme Linux).

Elle semble ainsi surclasser l'ensemble des cartes électroniques présentées dans ce tutoriel mais elle a tout à fait sa place dans ce modeste comparatif. En effet, la plus petite RPi est la Raspberry Pi Zero et ses dimensions sont proches de la carte Arduino Uno, mais avec une puissance bien supérieure !

La RPi possède une connectivité complète : Bluetooth, WiFi, ports USB, port Webcam et parfois port Ethernet et carte son... Elle est très adaptée au prototypage et à la réalisation de projets nécessitant une puissance de calcul considérable. Elle contient aussi des ports GPIO entrée/sortie permettant de piloter des actionneurs ou de recevoir des données provenant des capteurs. On préférera cependant utiliser une Arduino en parallèle de la RPi pour le pilotage des entrées/sorties et la lecture des capteurs. La RPi est, en général, consacrée à la partie calculs et au déroulement du programme général.

Pour quel type d'utilisation ?

Il est préférable d'utiliser la RPi pour **des projets «IoT» assez volumineux, comportant plusieurs capteurs et actionneurs et nécessitant une puissance de calcul considérable ou une interface directe sur un écran d'ordinateur.**

Exemples. *Camera de surveillance connectée, Reconnaissance et analyse vidéo, Reconnaissance vocale, Pilotage d'autres dispositifs «IoT», Centralisation de données*

B.4 Bilan

	ESP32	ESP8266	Arduino UNO	Raspberry Pi Zero
Fréquence	160 Mhz	80 Mhz	16 Mhz	1 GHz
Ports Digitaux	36	17	14	28
Ports Analogiques	18	1	6	-
Bluetooth 4.0 Intégré	✓	-	-	✓
WiFi Intégré	✓	✓	-	✓
Alimentation USB	✓	-	✓	✓

TABLE 1 – Comparatif des cartes électroniques

C Installation de l'ESP32

Passons maintenant à l'étape la plus importante de ce tutoriel : comment programmer une carte ESP32 avec l'**IDE Arduino** ?

Il existe en effet plusieurs méthodes pour programmer une carte ESP32, mais par soucis d'uniformisation et pour limiter le nombre de logiciels requis dans les tutoriels FabLab, nous utiliserons l'outil IDE Arduino qui est assez simple d'utilisation.

La carte Arduino est très facilement programmable avec ce logiciel, cependant il sera nécessaire de **réaliser quelques préparatifs pour pouvoir programmer l'ESP32, notamment installer la carte ESP32 en elle-même**.

⋆ ⋆

SUIVRE LES ÉTAPES CI-DESSOUS

⋆

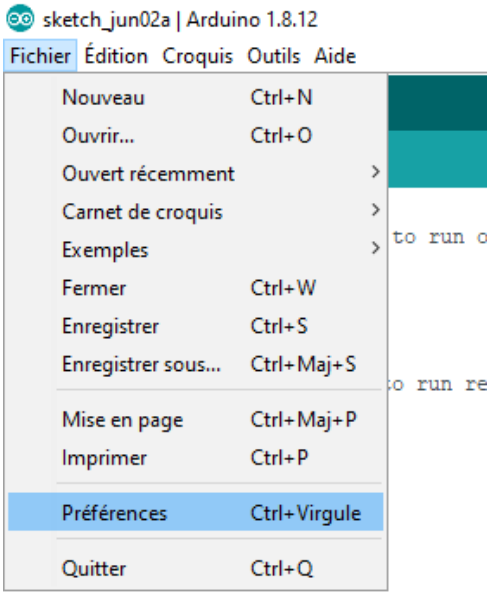
1. **Télécharger et installer l'IDE Arduino** depuis le site officiel Arduino.cc

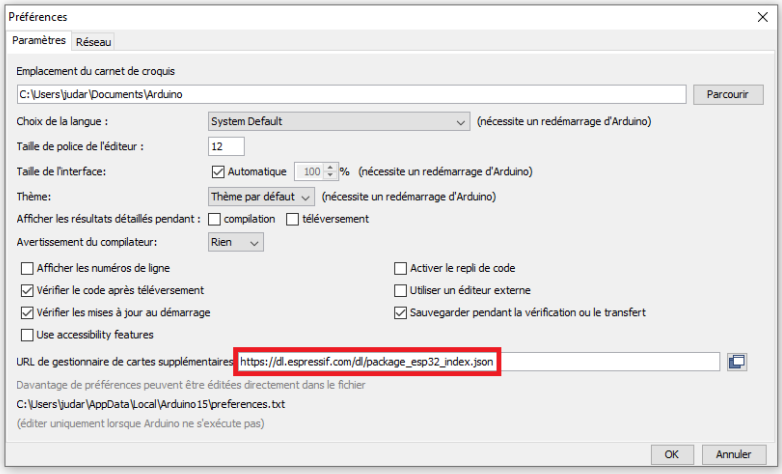
⋆

2. Ajouter la carte ESP32 dans la base des cartes Arduino de l'IDE

Dans l'IDE Arduino :

- Ouvrir l'onglet *Fichier* > *Préférences*
- Entrer l'adresse suivante dans le champ *URL de gestionnaire de cartes supplémentaires* :
  `https://dl.espressif.com/dl/package_esp32_index.json`



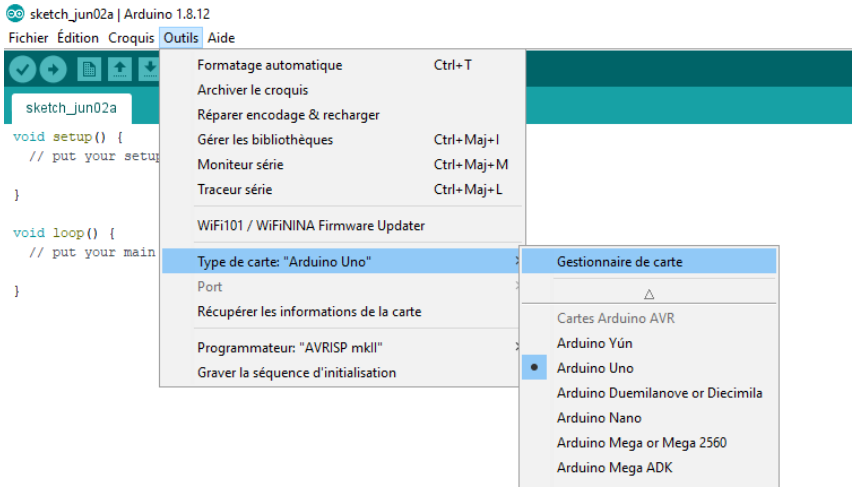


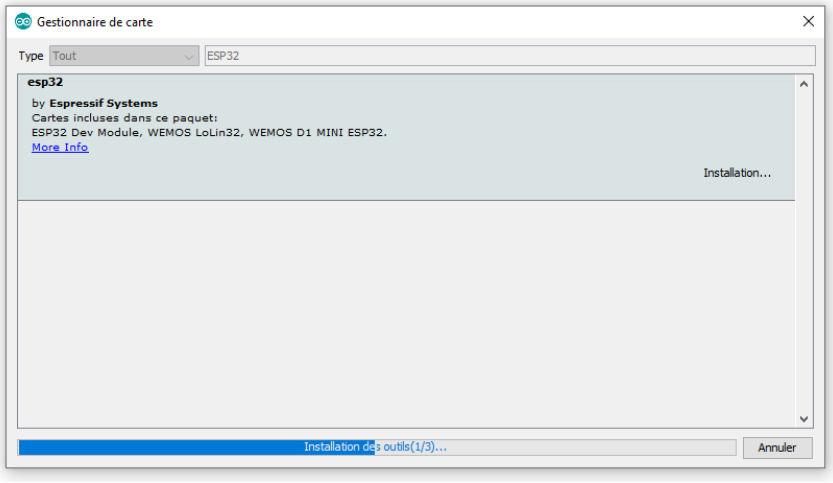
⋆

3. Installer la carte ESP32 dans l'IDE Arduino

Toujours dans l'IDE Arduino :

- Ouvrir l'onglet
  Outil > *Type de carte* > *Gestionnaire de carte*
- Rechercher le paquet *ESP32*
- Installer le paquet : sélectionnez la dernière version disponible et cliquer sur *Installer*





⋆

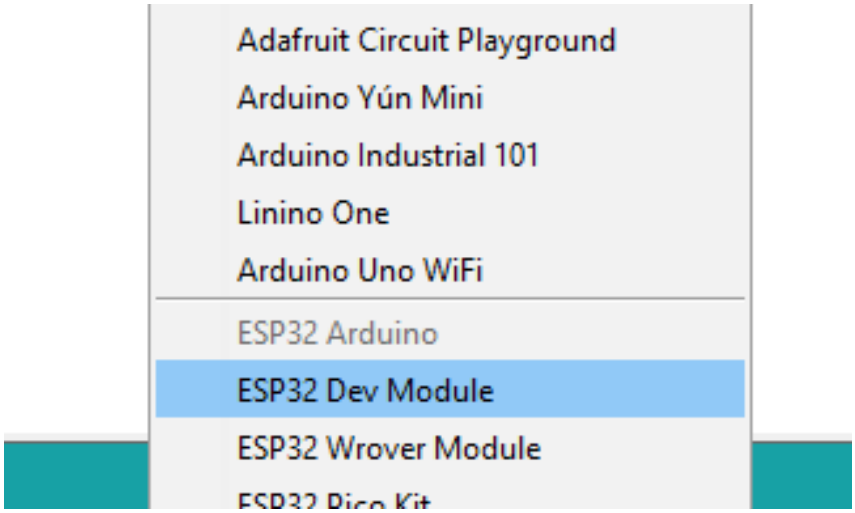
4. Sélectionner la carte ESP32

Il ne reste plus qu'à définir la carte ESP32 pour que le programme soit correctement envoyé sur la carte.
Toujours dans l'IDE Arduino :

- Ouvrir l'onglet

  Outil > *Type de carte*

- Choisissez alors la carte *ESP32* qui correspond à votre version d'ESP32. Le plus souvent ce sera la carte ***ESP32 Dev Module***



⋆

Merci d'avoir suivi ce tutoriel, n'hésitez pas à contacter les membres de l'association FabLabUTC en cas de besoin.

⋆ ⋆ ⋆