ESPhome

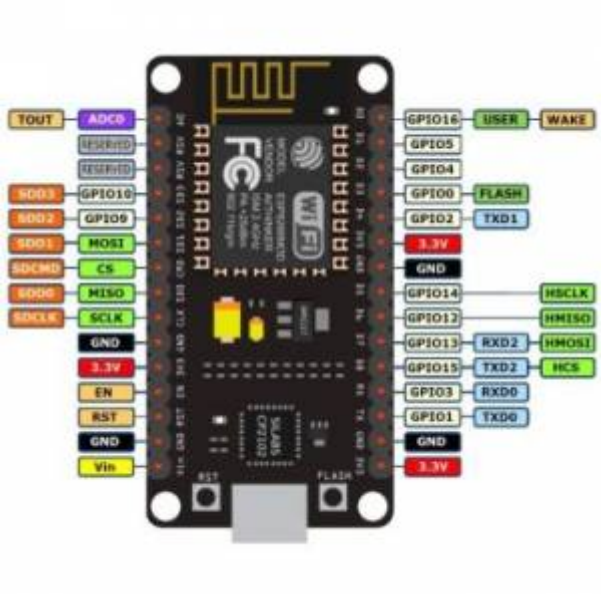
Cette section recense des conseils et des projets utilisant le framework ESPhome.

ESPhome est un outil puissant et convivial pour le développement de dispositifs IoT avec les micro-contrôleurs ESP8266, ESP8285 et ESP32, en simplifiant la création du firmware et en offrant des fonctionnalités d'intégration avec Home Assistant.

Un framework IoT

ESPhome est un framework open source destiné à faciliter le développement de firmware pour les microcontrôleurs ESP8266, ESP8285, ESP32 et d'autres puces similaires. Il est principalement utilisé pour créer des logiciels pour les dispositifs IoT (Internet des objets) utilisant sur des puces ESP.

ESP vs Arduino



On peut considérer les ESP comme les remplaçants ou successeurs des antiques Arduino.

Arduino est reconnu pour sa simplicité d'utilisation, sa vaste communauté et sa grande variété de cartes matérielles, ce qui en a fait longtemps un excellent choix pour les débutants en électronique et en programmation.

En revanche, l'ESP plus moderne se distingue par ses capacités de connectivité sans fil, notamment le Wi-Fi et le Bluetooth, ce qui en fait un choix privilégié pour les projets liés à l'Internet des objets (IoT).

Les microcontrôleurs ESP, comme l'ESP8266 et l'ESP32, sont particulièrement adaptés pour les applications où la communication à distance est essentielle. Il y a quelques années, on considérait encore que le choix entre Arduino et ESP dépendait du type de projet que l'on souhaitait réaliser : Arduino pour des projets de base et l'apprentissage, et l'ESP pour les projets IoT nécessitant une connectivité sans fil avancée.

On peut tout à fait programmer un ESP depuis l'IDE Arduino. Cependant, l'arrivée du framework ESPHome a redéfinit la donne : le recours pour programmer un ESP à un fichier descriptif en Yaml plutôt qu'à un code C++ a permis d'améliorer la qualité des programmes produits et élargir l'assiette des utilisateurs en intégrants les débutant·e·s et les personnes en apprentissage.

Principaux ESP

- ESP8266 : microcontrôleur Wi-Fi largement utilisé pour les projets IoT. Il est connu pour sa faible consommation d'énergie, son faible coût et sa compatibilité avec de nombreuses plates-formes de développement.
- ESP8285 : variante de l'ESP8266 avec un petit espace de stockage Flash intégré, utile pour les projets où l'espace de stockage est limité.
- ESP32 : microcontrôleur plus avancé, doté de capacités de Wi-Fi et de Bluetooth intégrées. Il offre également une puissance de traitement plus élevée, davantage de broches GPIO et une mémoire plus importante par rapport à l'ESP8266.

Développement avec ESPhome

Les ESP peuvent se programmer avec l'IDE Arduino en C++. Comme les Raspberry Pi Pico, les ESP peuvent se programmer également en LUA et en micro python.

ESPhome simplifie considérablement le processus de développement en permettant aux développeurs de configurer les fonctionnalités de leur dispositif IoT à l'aide de fichiers de configuration YAML, plutôt que de devoir écrire manuellement le code source C++ pour le microcontrôleur. Les fichiers de configuration ESPhome définissent les broches matérielles, les capteurs, les actionneurs, les communications réseau, etc.

Une fois que vous avez créé votre configuration ESPhome, vous pouvez compiler et flasher le firmware sur votre microcontrôleur ESP à l'aide de l'outil ESPhome. Cela simplifie grandement le processus de développement, en particulier pour ceux qui ne sont pas familiers avec la programmation embarquée.

ESPhome offre une intégration directe avec Home Assistant, la plateforme domotique open source. C'est l'environnemnt choisi pour la majorité des projets de domotique et de surveillance environnementale développé au BIK'LAB.

- Construction d'un nœud basique de mesures environnementales avec ESPHome
  - ESPhome / BME 280
  - ESPhome / BME 680
  - ESPhome / ds18b20
  - ESPhome / SHT31
  - ESPhome / tsl2561

From:
https://wiki.lebiklab.fr/ - **Wiki Le BIK'LAB**

Permanent link:
https://wiki.lebiklab.fr/doku.php?id=projets:home-assistant:esphome:start&rev=1698163774

Last update: **04/04/2024 15:35**