



PRESENTATION DE L'ESP32

Réalisé par Bruno PIQUEMAL

24 septembre 2020

**Ce guide vous apportera des informations concernant le microcontrôleur
ESP32.**



Introduction

L'ESP32 est un microcontrôleur fabriqué par la multinationale chinoise Expressif Systems, créée en 2008. L'ESP32 est le successeur de l'ESP8266 avec beaucoup d'améliorations faites au niveau de l'hardware et du software. En mai 2020, l'entreprise chinoise lance une version encore plus sûre : l'ESP32-S.

Cette présentation n'abordera que brièvement les caractéristiques que possède la puce ESP32 car il existe plusieurs modèles de celle-ci et avec une grande variété de cartes électroniques pour chacun d'entre eux.

L'important est de comprendre les caractéristiques principales de la puce pour ensuite choisir une carte adaptée à ses besoins.



Table des matières

| | | |
|----------|-------------------------------------|----------|
| A | Possibilités de l'ESP32 | 3 |
| A.1 | Très basse consommation | 3 |
| A.2 | Capteurs et périphériques | 4 |
| B | WROOM ou WROVER? | 5 |
| C | Exemples de cartes ESP32 | 6 |
| D | Pour aller plus loin | 8 |
| E | Bibliographie | 8 |
| F | Annexes | 9 |

* * *



L'ESP32 est une puce de très basse consommation cadencée à 2.4 GHz avec la WiFi et le Bluetooth intégrés.

Elle a été conçue pour atteindre les meilleures performances concernant la puissance et l'émission d'ondes radio. Elle permet d'avoir aussi une bonne robustesse, de la polyvalence et de la fiabilité pour une grande variété d'applications.

La série ESP32 possède différents modèles dont l'ESP32-D0WDQ6, l'ESP32-D0WD, l'ESP32-D2WD et l'ESP32-S0WD.

A Possibilités de l'ESP32

A.1 Très basse consommation

L'ESP32 a été réfléchi pour incorporer des caractéristiques de très basse consommation, d'horloge hyper précise et de plusieurs modes de consommation.

Dans le cas d'un capteur IoT endormi, l'ESP32 ne se réveillera que de manière périodique ou si une certaine condition est vérifiée. Le rapport cyclique est contrôlé afin de minimiser les dépenses énergétiques. La sortie de l'amplificateur de puissance est ajustable afin d'avoir le meilleur ratio entre distance de communication, quantité de données et consommation.

Voici les modes de consommation possibles sur l'ESP32 :

- Active mode : La puce radio est allumée et elle peut recevoir, écouter et transmettre.
- Modem-sleep mode : le CPU tourne encore et l'horloge interne est configurable. Le WiFi, le Bluetooth et la radio sont désactivés.
- Light-sleep mode : le CPU est en pause. La mémoire du Real Time Clock (RTC), les périphériques liés au RTC et le co-processeur Ultra Low Power (ULP) tournent encore. Les événements causant des interruptions pourront réveiller l'ESP32.
- Deep-sleep mode : Sont activés que la mémoire du RTC et ses périphériques ainsi que le co-processeur ULP. Les données WiFi et Bluetooth sont stockées dans la mémoire RTC.
- Hibernation mode : L'oscillateur interne de 8 MHz et le co-processeur ULP sont désactivés. Seulement un Timer du RTC et quelques GPIO's tournent. Ces derniers peuvent faire sortir l'ESP32 de son endormissement.



A.2 Capteurs et périphériques

L'ESP32 possède des aspects très intéressants concernant le traitement des données venant de ses GPIO's ainsi que de ses cartes sans fil.

- GPIO's : l'ESP32 a 34 pins qui peuvent être assignés à diverses fonctions en programmant les registres appropriés. La vaste majorité de ces GPIO's peuvent être configurés avec des pull-up ou pull-down internes ou se comporter comme de fortes impédances. En mode très basse consommation, les pins peuvent être programmés pour conserver leur état et ainsi économiser de l'énergie.
- Analog to Digital Converter (ADC) : l'ESP32 intègre un Successive Approximation Register (SAR) ADC et permet de prendre des mesures sur 18 pins différents. Lors des phases de très basse consommation, le co-processeur ULP est capable de faire des mesures analogiques afin d'optimiser la consommation d'énergie.
- Hall sensor : Dès que l'ESP32 est en présence d'un champ magnétique, le capteur à effet Hall crée une petite tension sur la résistance de N-carrier. Si l'on veut utiliser ce capteur intégré, c'est l'ADC qui se chargera de mesurer la différence de potentiels.
- Digital-to-Analog Converter (DAC) : Deux pins permettent qu'un signal numérique de l'extérieur soit converti en un signal analogique de 8 bits.
- Touch Sensor : l'ESP32 possède 10 pins qui détectent une variation causée par le touché ou par le rapprochement d'un objet. Le faible bruit interne et la grande sensibilité des circuits permet l'utilisation de pads tactiles.
- Ultra-Low-Power (ULP) Co-processor : Le co-processeur ULP et la mémoire RTC demeurent activés dans le mode Deep-sleep. C'est une façon d'économiser de l'énergie tout en ayant la capacité d'effectuer des tâches avec les périphériques, les timers et les capteurs internes.
- Ethernet MAC Interface : un IEEE-802.3-2008-compliant Media Access Controller (MAC) est pourvu pour les communications LAN Ethernet.
- SD/SDIO/MMC Host Controller : ce contrôleur permet des sorties cadencées à 80 MHz avec trois types de bus : 1 bit, 4 bits et 8 bits. Il peut aussi lire des cartes SD/SDIO/MMC4.41 et une carte SD.
- SDIO/SPI Slave Controller : l'ESP32 peut dialoguer avec d'autres appareils en utilisant les protocoles SDIO et SPI.
- Universal Asynchronous Receiver Transmitter (UART) : trois interfaces UART de disponibles avec des communications allant jusqu'à 5 Mbps.
- I²C Interface : Possibilité de modifier les vitesses de communication de ce bus.
- I²S Interface : les deux interfaces ont des contrôleurs dédiés DMA et ils peuvent supporter les interfaces PDM et BT PCM.



- Infrared Remote Controller : l'ESP32 a 8 pins pouvant émettre et recevoir de la communication infrarouge tout en supportant plusieurs protocoles.
- Pulse Counter : ceci permet de compter des signaux selon sept modes différents. Il est aussi possible de définir une valeur qui déclenchera une interruption lorsque le compteur égalera la valeur.
- Pulse Width Modulation (PWM) : le PWM facilite l'utilisation de moteurs et de LED en jouant sur les Timers.
- LED PWM : 16 pins sont dédiés à générer des signaux et il est aussi possible de modifier leur rapport cyclique pour le contrôle de l'allumage des LED.
- Serial Peripheral Interface (SPI) : trois protocoles SPI sont disponibles avec la possibilité de les contrôler à l'aide d'une DMA.
- Accelerator : l'ESP32 a un accélérateur d'algorithmes intégré qui permet d'améliorer la vitesse et de réduire la complexité des programmes suivants : le AES, le SHA, RSA et l'ECC.

B WROOM ou WROVER ?

L'ESP32 possède deux familles de modèles : WROOM et WROVER. La différence entre ces modèles se situe au niveau de l'intégration de la PSRAM (ou Ram Pseudo Statique SPI). Les modèles WROVER intègre de 2 à 8MB de PSRAM. Les modèles WROOM ne disposent pas de PSRAM.

Est-ce vraiment utile d'avoir de la PSRAM ?

La PSRAM est une mémoire volatile que l'on peut utiliser pour étendre la mémoire de l'ESP32. Les données stockées ne sont pas persistantes puisqu'elles sont effacées dès que le module ESP32 n'est plus alimenté.

L'ESP32 dispose de quelques centaines de kilo-octets de RAM interne embarquée directement dans le SoC (SRAM). Cette quantité de RAM peut s'avérer insuffisante, par exemple, pour des applications de traitement vidéo ou de traitement du signal (FFT...). Il est donc possible d'exploiter de la RAM externe (la PSRAM) accessible via le bus SPI.

Il est primordial de choisir la bonne quantité de RAM en fonction de l'application désirée, notamment pour des applications qui nécessitent un traitement vidéo.

C Exemples de cartes ESP32



FIGURE 1 – Carte ESP32 BrightDot

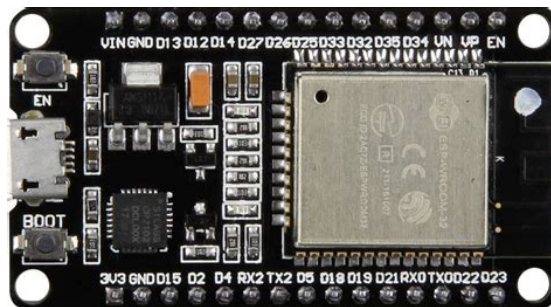


FIGURE 2 – Carte ESP32 NodeMCU

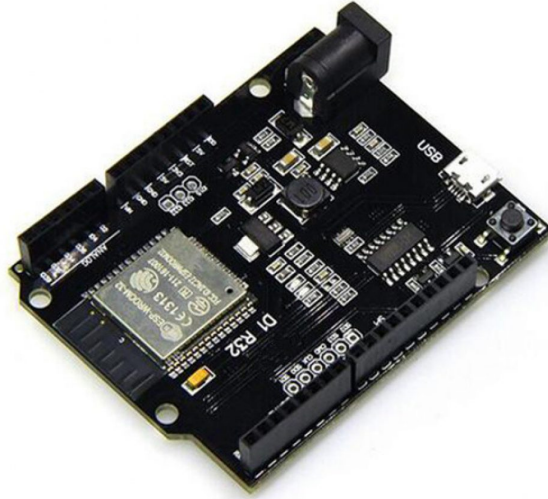


FIGURE 3 – Carte ESP32 format Arduino



FIGURE 4 – Carte ESP32 avec caméra 2MP et carte SD



D Pour aller plus loin

Ce tutoriel est une brève introduction à toutes les fonctionnalités de la carte ESP32. Comme vous pouvez le voir au travers des exemples, il existe beaucoup de modèles qui possèdent des caractéristiques différentes et il est impossible de parler de chacun d'entre eux. Toutefois, il faut savoir que chaque carte a un nombre de GPIO's qui peut varier en fonction du constructeur. Le tout est de prendre la carte qui correspond à l'application souhaitée.

E Bibliographie

<https://projetsdiy.fr/quelle-carte-esp32-choisir-developper-projets-di>

https://docs.ai-thinker.com/_media/esp32/esp32_datasheet_en.pdf



F Annexes

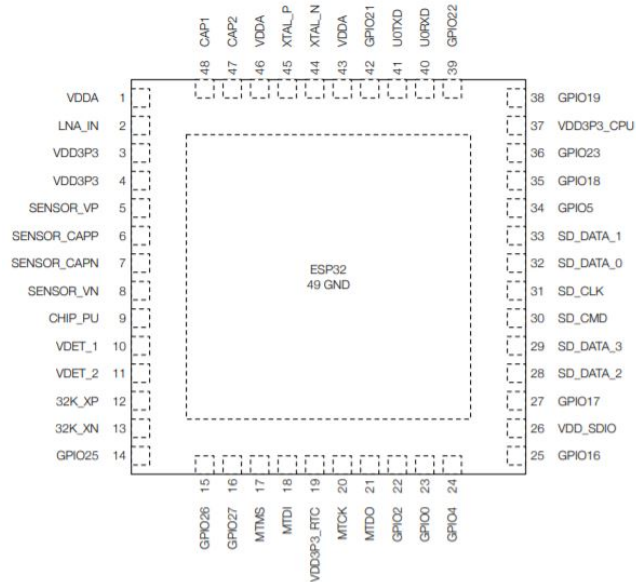


FIGURE 5 – Schéma des connecteurs de l'ESP32

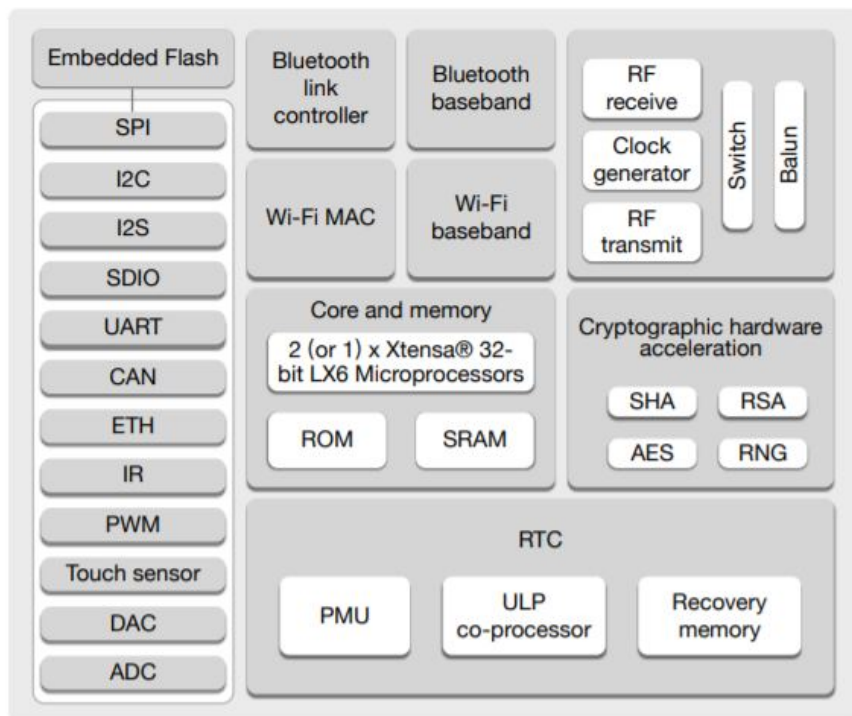


FIGURE 6 – Diagramme de fonctions de l'ESP32



| | Processeur Xtensa | Bluetooth | Core | Broches | Mémoire Flash (MB) | PSRAM (MB) | Deep Sleep* |
|--------------|--------------------|-----------|------|---------|--------------------|------------|-------------|
| ESP32-S2 | LX7 jusqu'à 240MHz | Non | x1 | 42 | | | 20 µA |
| | WROOM ESP32-S2 | | | | 4,8,16 | - | |
| ESP32 | WROVER ESP32-S2 | | | | 4,8,16 | 2 | |
| | Mini ESP32-S2FH4 | | | | | | |
| ESP32 | LX6 80 ~ 240MHz | Oui | x2 | 38 | | | 5 µA |
| | VROOM | | | | 4,8,16 | - | |
| ESP32-SOLO-1 | WROVER | | | | 4,8,16 | 8 | |
| | LX6 @ 160MHz | | x1 | 38 | 4 | - | 5 µA |

FIGURE 7 – Différents modèles d'ESP32 et leurs caractéristiques