



INTRODUCTION À L'ESP32



Sommaire

- ▣ Caractéristiques des ESP32
- ▣ Moniteur Temps-réel
- ▣ Kits de développements
- ▣ Ateliers de développements
- ▣ Références

Caractéristiques des ESP32

- ▣ ESP32 est un SOC (System On Chip)
 - Deux coeurs Xtensa® 32-bit LX6
 - Un co-processeur ULP (Ultra Low Power)
 - Intègre le Wifi 2,4 GHz
 - Intègre le Bluetooth et le BLE
- ▣ Une gamme de chips avec des performances différentes :
 - ESP32, C1, C2, C3, S1, S2, S3
- ▣ 28 ou 32 pins

Possibilités en mémoires

- ▣ Embedded Memory
 - - 448 KB Internal ROM
 - - 520 KB Internal SRAM
 - - 8 KB RTC FAST Memory
 - - 8 KB RTC SLOW Memory
- ▣ • External Memory
 - Off-chip SPI memory can be mapped into the available address space as external memory. Parts of the embedded memory can be used as transparent cache for this external memory.
 - ▣ - Supports up to 16 MB off-Chip SPI Flash.
 - ▣ - Supports up to 8 MB off-Chip SPI SRAM.

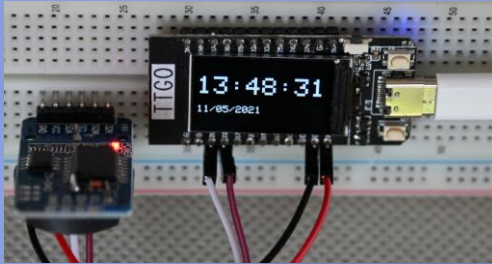
Types de modules Espressif

- ▣ ESP32-WROOM
- ▣ ESP32-WROVER
- ▣ ESP32-MINI
- ▣ Les modules Espressif

Moniteur temps-réel

- ▣ Free RTOS (Real Time Operating System)
 - Multi-tâches
 - Allocation du processeur aux tâches par slots de 1ms
 - Permet de gérer le multi-cores (Wifi, Bluetooth sur le Core 0, appli sur Core 1)

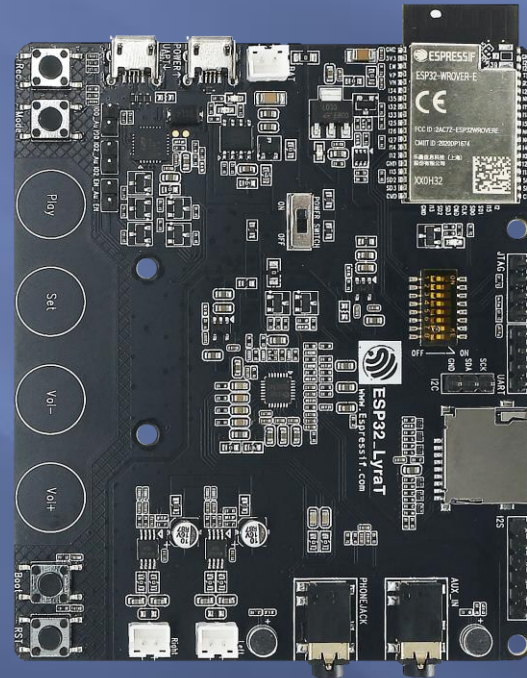
Kits de développement



**TTG0 T-
Display**



WROOM 32



Lyra-t WROVER-E

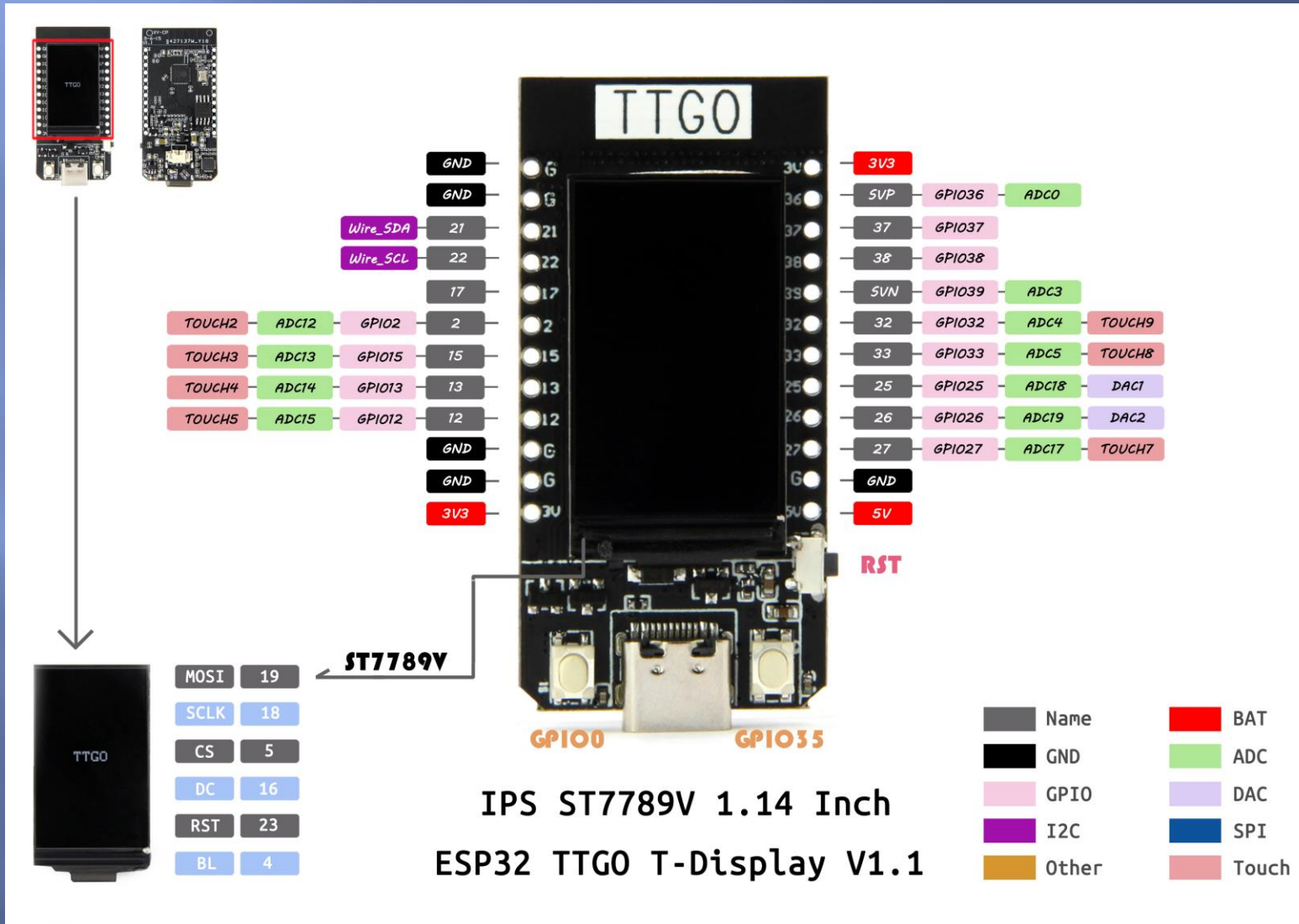


EC Buying

ESP32-Audio-Kit

ESP32 - A1S

TTGO T-Display



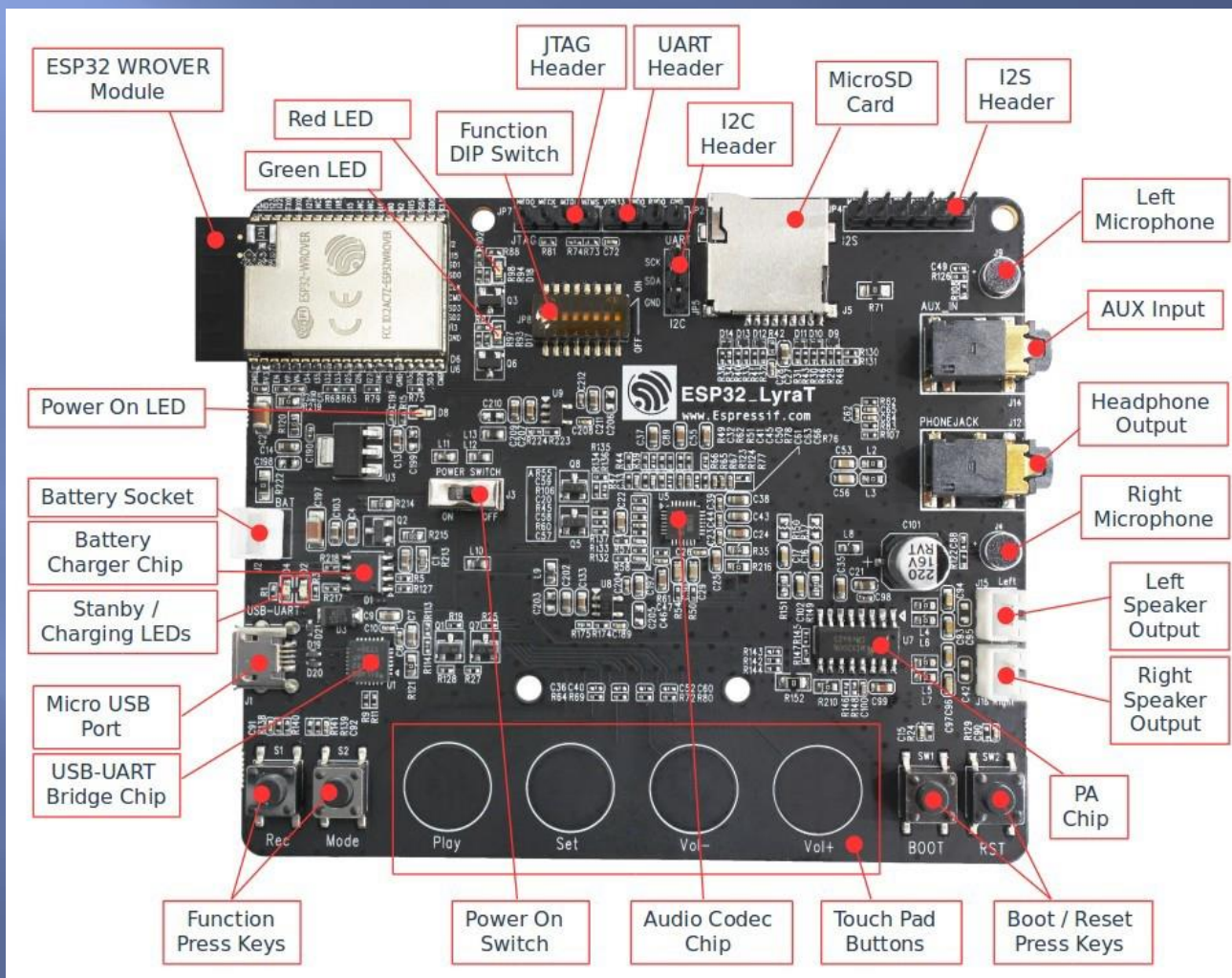
WROOM32

Name	No.	Type	Function
SENSOR_VP	4	I	GPIO36, ADC1_CH0, RTC_GPIO0
SENSOR_VN	5	I	GPIO39, ADC1_CH3, RTC_GPIO3
IO34	6	I	GPIO34, ADC1_CH6, RTC_GPIO4
IO35	7	I	GPIO35, ADC1_CH7, RTC_GPIO5
IO32	8	I/O	GPIO32, XTAL_32K_P (32.768 kHz crystal oscillator input), ADC1_CH4, TOUCH9, RTC_GPIO9
IO33	9	I/O	GPIO33, XTAL_32K_N (32.768 kHz crystal oscillator output), ADC1_CH5, TOUCH8, RTC_GPIO8
IO25	10	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
IO26	11	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
IO27	12	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV
IO14	13	I/O	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
IO12	14	I/O	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
GND	15	P	Ground
IO13	16	I/O	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
SHD/SD2*	17	I/O	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
SWP/SD3*	18	I/O	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
SCS/CMD*	19	I/O	GPIO11, SD_CMD, SPICS0, HS1_CMD, U1RTS

SCK/CLK*	20	I/O	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS
SDO/SD0*	21	I/O	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS
SDI/SD1*	22	I/O	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
IO15	23	I/O	GPIO15, ADC2_CH3, TOUCH3, MTDO, HSPICS0, RTC_GPIO13, HS2_CMD, SD_CMD, EMAC_RXD3
IO2	24	I/O	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPiWP, HS2_DATA0, SD_DATA0
IO0	25	I/O	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
IO4	26	I/O	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPiHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
IO16	27	I/O	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
IO17	28	I/O	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
IO5	29	I/O	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
IO18	30	I/O	GPIO18, VSPICLK, HS1_DATA7
IO19	31	I/O	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
NC	32	-	-
IO21	33	I/O	GPIO21, VSPIHD, EMAC_TX_EN
RXD0	34	I/O	GPIO3, U0RXD, CLK_OUT2
TXD0	35	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
IO22	36	I/O	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
IO23	37	I/O	GPIO23, VSPID, HS1_STROBE
GND	38	P	Ground

4 MB FLASH

Lyra-T



Ateliers de développement

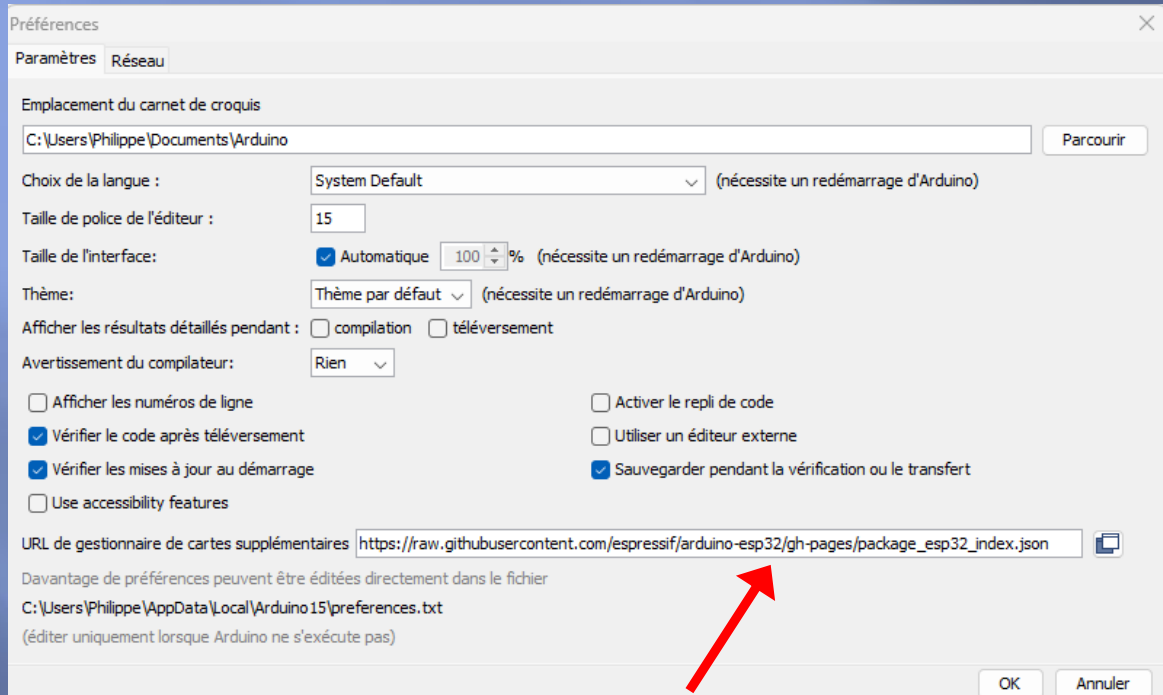
- ▣ IDE Arduino : le plus facile
- ▣ Ateliers espressif :
 - IDF : langage de commande dans shell ou powershell ☹
 - ADF : surcouche de l'IDF pour les développements audio
 - ESP-Skainet : surcouche pour la reconnaissance vocale
 - ...
 - Espressif IDE : dérivation d'Eclipse

Ateliers de développement

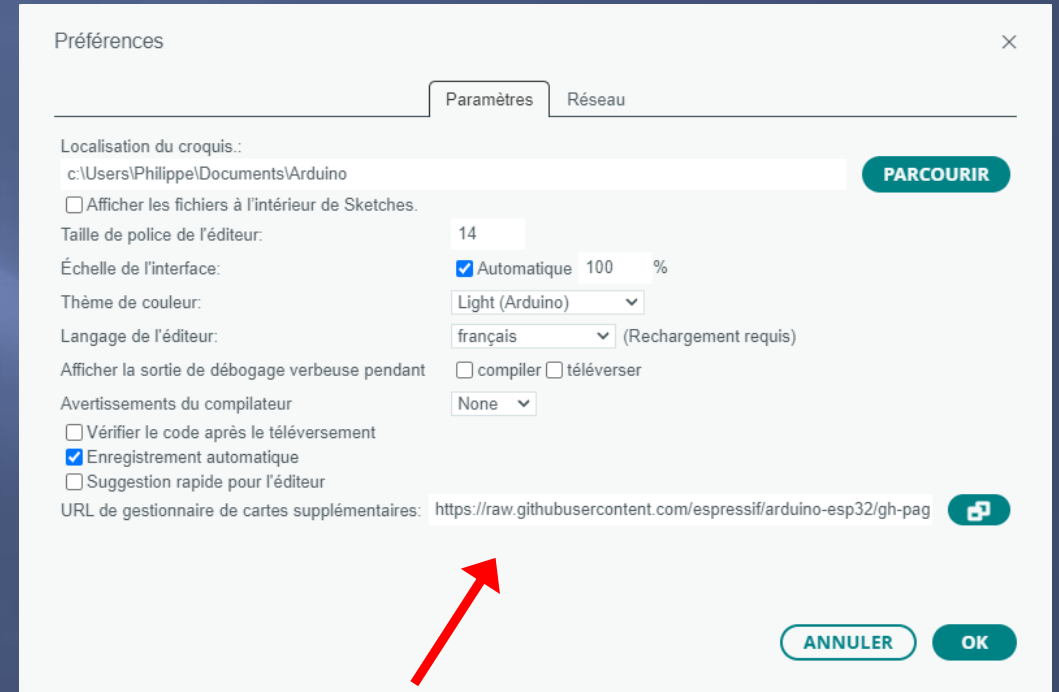
- ▣ Visual Studio Code : encapsulation de l'IDF espressif
- ▣ PlatformIO

IDE Arduino

Installation des cartes ESP32



1.8.19



2.0.3

Quelques retours

- ▣ Attention c'est du 3V !
- ▣ Suivant les modules téléchargement automatique ou par boot/reset
- ▣ Pour une liaison série
 - ▣ SoftwareSerial (Arduino) est remplacée par HardwareSerial
 - ▣ UART 0 = USB, UART 1 = système, UART 2 = libre
 - ▣ Pins souvent à redéfinir dans l'initialization
- ▣ Partitionnement possible de la mémoire FLASH : création d'une mémoire SPIFFS (Serial Peripheral Interface Flash File System) pour stocker le code d'un client web (photos, HTML, CSS, Javascript) par exemple
- ▣ Consulter f6czv.fr ☺

Références

- ▣ Espressif
 - Espressif.com
 - [Sommaire ESP32](#)
 - [Github Espressif](#)
- ▣ Lilygo
 - [Lilygo](#)
 - [Github : Lilygo](#)
- ▣ AI Thinker
 - [AI-Thinker](#)
 - [Github : AI-Thinker A1S Audio Kit](#)
 - [AI Thinker Audiokit](#)
 - [The 'AI Thinker Audio Kit' Arduino Challenge](#)

Références

- ▣ [Random Nerd Tutorials \(ESP32\)](#)
- ▣ [Microcontrolleurs-mcu - esp32-iot](#)
- ▣ [Andreas Spiess \(ESP32\)](#)
 - [RTOS](#)
 - [ESP32 Audio Tutorial with lots of examples](#)
- ▣ Philippe f6czv
 - [Site f6czv](#)
 - [Github f6czv](#)

Références

- ▣ Ateliers :
 - IDE Arduino :
 - ▣ <https://www.arduino.cc/en/software>
 - IDF Espressif :
 - ▣ [Standard Setup of Toolchain for Windows](#)
 - ADF Espressif :
 - ▣ [Espressif ADF](#)
 - ▣ [AI Thinker A1S SDK](#)
 - Microsoft Visual Studio Code
 - ▣ [VS Code](#)
 - ▣ [Installation IDF Espressif](#)
 - PlatformIO :
 - ▣ [PlatformIO.org](#)
 - ▣ [bien-commencer-la-programmation-des-esp32-avec-platformio](#)