

## Utilitário Chronos

**Chronos** é um software utilitário para teste **FTM** (Fine Time Measurement) em dispositivos **Espressif ESP32**. Ele pode ser compilado e instalado em qualquer dispositivo **ESP32** com suporte ao recurso FTM (famílias **S2**, **S3** e **C3**).

O código fonte está disponível no seguinte repositório **github** :

<https://github.com/cezmen/chronos>

## Como Utilizar o Utilitário Chronos

### [1] Defina a CPU Target

```
idf.py set-target esp32s2
```

### [2] Configure o Projeto

```
idf.py menuconfig
```

**[2.1] Defina os parâmetros de configuração nos menus a seguir:**

- **Example Configuration** ► **Soft AP**
- **Example Configuration** ► **TCP Server**
- **Example Configuration** ► **FTM**

PARAMETER	DESCRIPTION	MODULE
ESP_WIFI_SSID	WiFi SSID	<b>SoftAP</b>
ESP_WIFI_PASSWORD	WiFi Password	
ESP_WIFI_CHANNEL	WiFi Channel	
ESP_MAX_STA_CONN	Maximal STA connections	
ESP_INTERFACE_IP	IPv4 Address	
ESP_INTERFACE_GW	Gateway IPv4 Address	
ESP_INTERFACE_NETMASK	Netmask	
ESP_IPV4	IPV4 (y/n)	<b>TCP Server</b>
ESP_IPV6	IPV6 (y/n)	
ESP_PORT	Port	
ESP_KEEPA_LIVE_IDLE	TCP keep-alive idle time(s)	
ESP_KEEPA_LIVE_INTERVAL	TCP keep-alive interval time(s)	
ESP_KEEPA_LIVE_COUNT	TCP keep-alive packet retry send counts	<b>FTM</b>
ESP_FTM_REPORT_LOG_ENABLE	FTM Report logging (y/n)	
ESP_FTM_REPORT_SHOW_DIAG	Show dialog tokens (y/n)	
ESP_FTM_REPORT_SHOW_RTT	Show RTT values (y/n)	
ESP_FTM_REPORT_SHOW_T1T2T3T4	Show T1 to T4 (y/n)	
ESP_FTM_REPORT_SHOW_RSSI	Show RSSI levels (y/n)	

## [2.2] Configuração Adicional de Parâmetros

Component Config ► WiFi ►

- WiFi FTM : y
- FTM Initiator Support : y
- FTM Responder Support : y

Serial Flasher Config ►

- Flash Size : 4MB
- After Flashing : Stay In Bootloader

Component Config ► Common ESP Related ►

- Channel for Console Output :
  - USB CDC ( if using **Franzininho WiFi** )
  - UART0 ( if using **ESP32-S2-Devkit-C** )

## [3] Execute o “Build”

Execute o “build” do projeto :

```
idf.py build
```

## [4] Grave o Firmware ( Execute o Flash )

Colocar a placa em modo DFU ( pressionando as teclas **BOOT** e **RESET** na seguinte sequência : pressione **BOOT**, pressione **RESET**, libere **RESET**, libere **BOOT** ).

Grave o firmware na placa :

```
idf.py -p <device name> flash
```

(Nota : Utilize 'ls /dev/tty\*' para descobrir o <device name> exato em seu ambiente de teste).

## [5] Monitore ( passo opcional )

Reinicie a placa ( pressionando e liberando a tecla **RESET** ).

### [5.1] Monitore a placa Franzininho WiFi (através de USB CDC)

Este Demo não funciona bem com "idf.py monitor", quando a saída Console utiliza a porta **USB CDC**.

Neste caso, utilize um emulador de terminal (como o **screen**).

```
screen <device name> 115200,cs8
```

(Nota : Utilize 'ls /dev/tty\*' para descobrir o <device name>)

(Para sair do **screen**, digite **Ctrl-A with k**, pressionando **y** logo em seguida para matar janela).

# Sistema de Localização baseado em Wi-Fi FTM RTT

## [5.2] Monitore a placa ESP32-S2-Devkit-C (através da UART0)

Execute o monitor ESP-IDF

```
idf.py -p <device name> monitor
```

(Nota : Utilize 'ls /dev/tty\*' para descobrir o <device name> exato em seu ambiente de teste).

(Para sair do monitor, pressione `Ctrl-J`.)

Refira-se ao arquivo **README.md** ( do repositório github ) para instruções complementares.