Wi-Fi FTM RTT Based Positioning System

PART 3

Chronos Utility

Chronos is an utility software for testing **FTM** (Fine Time Measurement) on **Espressif ESP32** devices. It can be built and installed on any **ESP32** device with **FTM** capability (**S2**, **S3** and **C3** families).

The source code is available from the following **github** repository :

https://github.com/cezmen/chronos

How to use the Chronos Utility

[1] Set Target

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

[2] Configure the Project

```
idf.py menuconfig
```

[2.1] Set Configuration Parameters in the following menus :

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

PARAMETER	DESCRIPTION	MODULE
ESP_WIFI_SSID	WiFi SSID	SoftAP
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)	TCP Server
ESP_IPV6	IPV6 (y/n)	
ESP_PORT	Port	
ESP_KEEPALIVE_IDLE	TCP keep-alive idle time(s)	
ESP_KEEPALIVE_INTERVAL	TCP keep-alive interval time(s)	
ESP_KEEPALIVE_COUNT	TCP keep-alive packet retry send counts	
ESP_FTM_REPORT_LOG_ENABLE	FTM Report logging (y/n)	FTM
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)	

CEZAR MENEZES
cezar.menezes@live.com

Wi-Fi FTM RTT Based Positioning System

PART 3

[2.2] Additional Parameters Setup

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] Build

Build the project :

```
idf.py build
```

[4] Flash

Put the board in DFU mode (by pressing **BOOT** and **RESET** keys in the following sequence : press **BOOT,** press **RESET**, release **RESET**, release **BOOT**).

Flash the firmware to the board :

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

(Note : Use 'ls /dev/tty*' to discover the exact **<device name>** in your environment)

[5] Monitor (optional)

Reset the board (by pressing and releasing the **RESET** key).

[5.1] Monitor the Franzininho WiFi board (through USB CDC)

This Demo doesn't work well with **"idf.py monitor"** when the Console Output is using **USB CDC** port.

In this case, use a serial terminal emulator (such as screen) instead.

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

(Note : Use '**ls /dev/tty***' to discover the exact **<device name>**)
(To exit **screen**, type `**Ctrl-A with k**`, pressing `**y**` right after to kill the window).

CEZAR MENEZES
cezar.menezes@live.com

Wi-Fi FTM RTT Based Positioning System

PART 3

[5.2] Monitor the ESP32-S2-Devkit-C board (through UART0)

Run the ESP-IDF monitor

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

(Note : Use '**ls /dev/tty***' to discover the exact **<device name>** in your environment)

(To exit the serial monitor, type `**Ctrl-]**`.)

Refer to the **README.md** file (from the github repository) for further instructions.

CEZAR MENEZES
cezar.menezes@live.com