

DESCRIPTION

1. Full integration on one board:

- ❖ STM32F103C8T6 - 20Kb RAM and 64Kb of flash with a frequency of 72Hz.
- ❖ WiFi ESP8285 with memory 2Mb (megabit) of RAM and 75 Kb RAM.
- ❖ External I2C EEPROM 64KB connected to STM32.
- ❖ SIM800C.

2. All of the modules can work together or each separately.

3. Each module has its own pinout headers.

4. The convenient solution for the development of new projects requiring larger memory space and WiFi for IOT.

5. Via USB you can update sketches and firmware for STM32F103C8T6 and for ESP8285. This on board has the USB-serial converter **CP2102**.

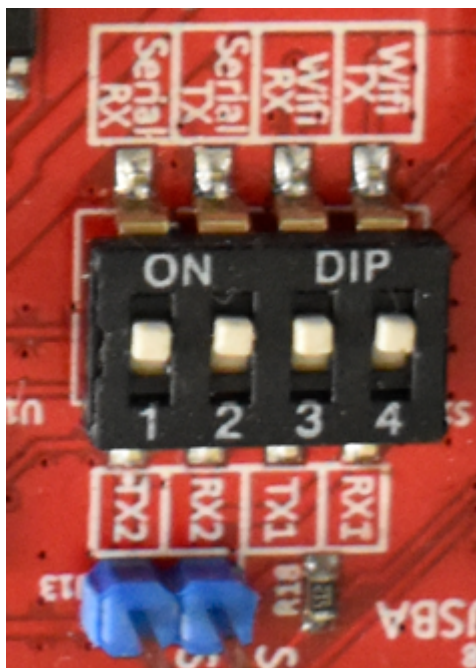
Using this board is very simple.

The board has a DIP-switch, to connect the modules.

For example, to USB and stm32f10c8t6, USB and ESP8285, and USB to external Serial.

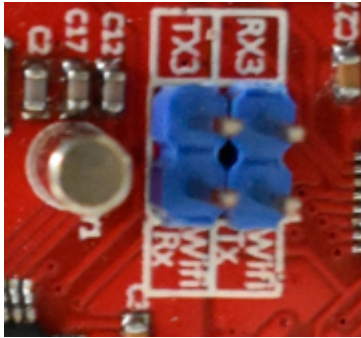
Table for DIP switch

CONNECTION	DIP				
	1	2	3	4	
USB <-> STM32	ON	ON	OFF	OFF	
USB <- > ESP8285	OFF	OFF	ON	ON	
USB <- > EXT SERIAL	ALL OFF Use J13 male header near DIP				Use J13



Special solution:

For ESP8285 <-> STM32 Connect **J4** the header is close to ESP.



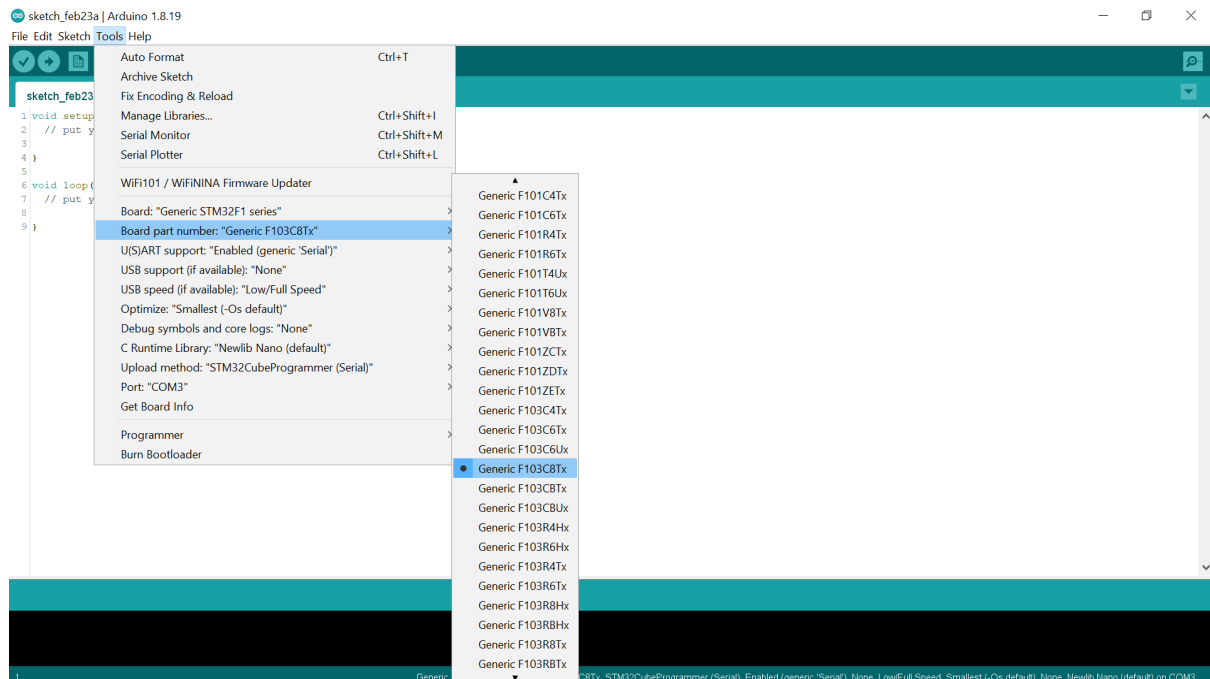
HOW TO FLASH CODE

Once you are done setting up the STM32 Core [github Eago Board readme](#)

Follow the below steps

1. Go to **Tools**.
2. Under **boards** select **STM32 board group** and under it select **Generic STM32F1 series**
3. Now under tools it should give the STM32 board , go to **board part number** and select **STM32F103C8Tx** or **blue pill F103C8**. Both serve libraries for the same chip.
4. Select **upload method** as **STM32Cube programmer (Serial)**
5. Choose a **port**.

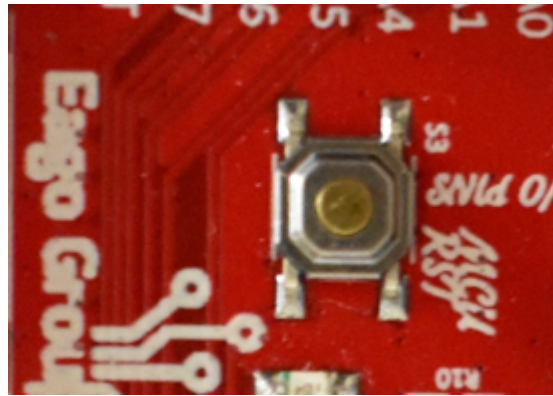
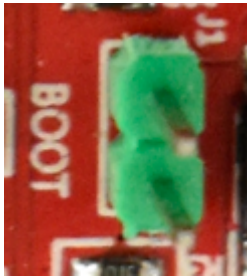
Image below should be your final setting.



nB you will need to download and install [STM32CubeProgrammer](#) to install drivers.

6. When you are done setting up this select programme DIP SWITCH 1 and 2.

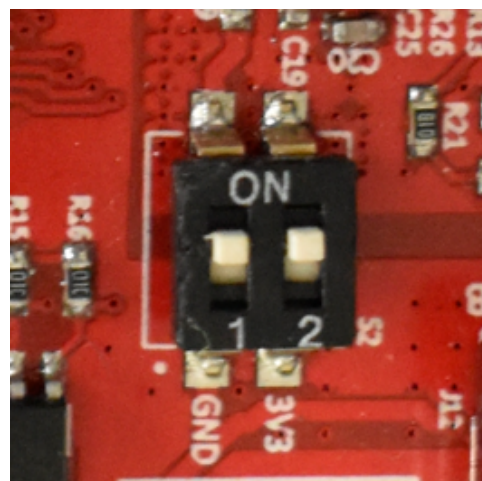
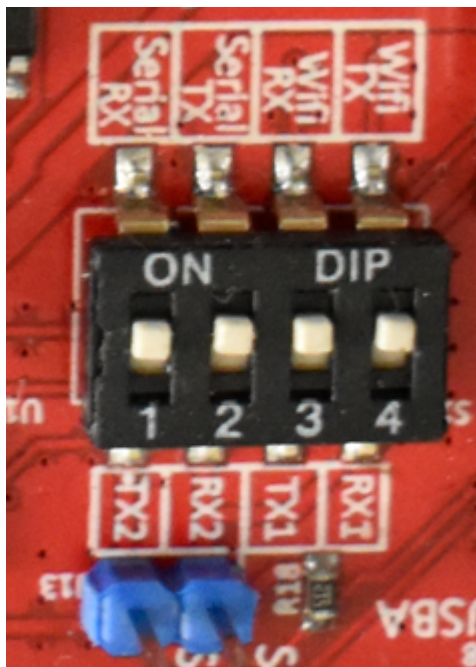
7. Put a jumper cap or short the **BOOT** header pin and press **MCU reset button**.



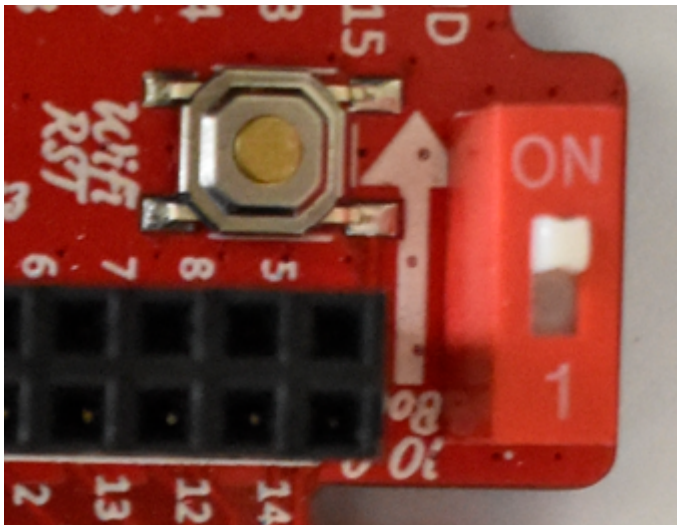
8. Now upload your code and once you are done uploading remove the jumper cap and reset.

HOW TO PROGRAMME ESP8285

1. You will need to setup ESP core from the [ESP github](#).
2. After you are done downloading the arduino cores go to **Tools** and select **ESP8285**.
3. Enable **DIP switch 3 and 4** to enable ESP Serial to USB and enable **DIP switch for ESP power S2**.



4. Enable the **red DIP SWITCH** to where the arrow point in order to enable boot mode by UART.



5. Press the **WIFI reset button**.
6. Select port to programme to.
7. Upload your code.
8. Once done switch back the Red DIP and reset the Wifi button as from the above image.

SIM800C

Sim 800C has a **power key** connected to **PA8** of the STM32.

This pin need to be turned high for 3 seconds anytime you are using SIM800 and then turned off, at the setup();

Use this at the setup

```
pinMode(PA8 , OUTPUT);  
digitalWrite(PA8, HIGH);  
delay(3000);  
digitalWrite(PA8, LOW);
```

The SIM800C is connected to Serial2 (**PA3,PA2**) of the STM32.

To communicate with SIM800 serial, create an instance of the serial then use

```
Your instance.setRx(PA2);  
Your instance.setTx(PA3);
```

Or simply create an instance of HardwareSerial.

E.g **HardwareSerial GSMSerial(PA3, PA2);**

Simple hacks and assignment

Change the Serial pinout to PA2 and PA3 initialization on the Generic variant.h of the Stm32 library to solve the Serial print issue.

