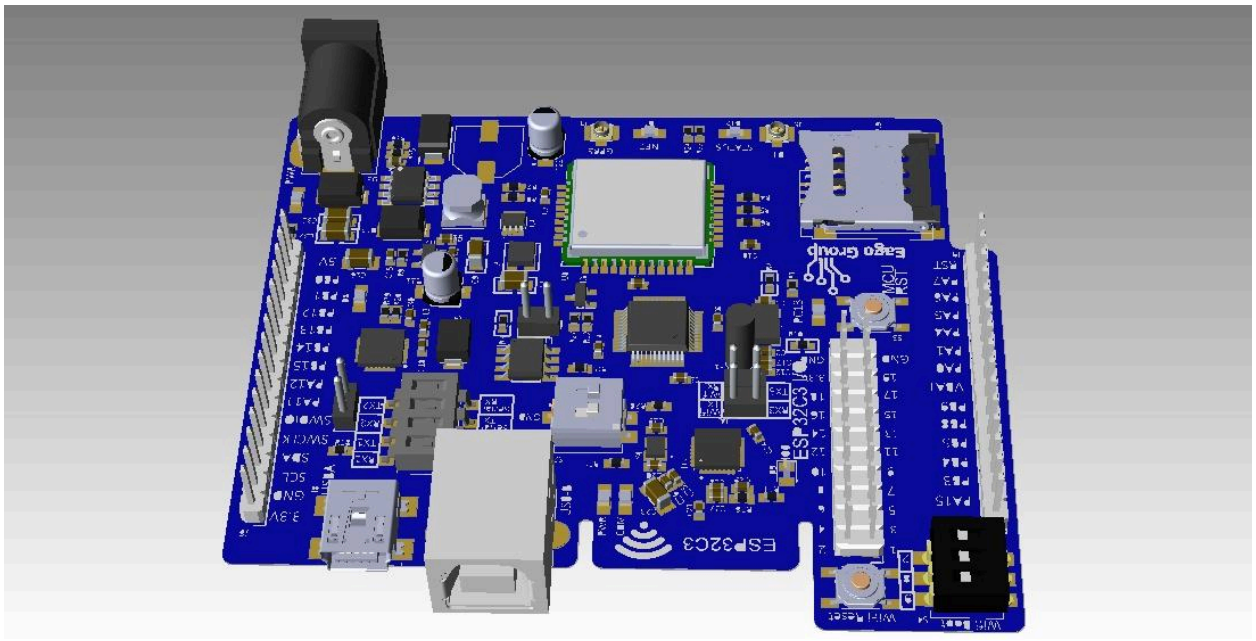
BLUE-EAGO Development Board Documentation

Overview

The **BLUE-EAGO Development Board** is an advanced IoT development platform designed for learners, developers, and hobbyists to quickly prototype and test IoT applications. Equipped with powerful microcontrollers, wireless modules, and robust connectivity options, it offers flexibility for a wide range of projects.



Key Features

- **GSM Connectivity** with SIM800C (with Bluetooth functionality)
- **Wi-Fi and BLE Support** using ESP32-C3FN4
- **STM32F103C8** for robust microcontroller performance
- **Flexible USB-to-Serial Communication** via CP2102
- **External EEPROM** for data storage

- **Wide Power Input Range** (4.5V to 50V) with 1A max draw

Technical Specifications

1. STM32F103C8

- **Core:** ARM Cortex-M3
- **Frequency:** 72 MHz
- **Flash Memory:** 64 KB
- **SRAM:** 20 KB
- **GPIO:** 37 I/O pins
- **Communication Interfaces:** SPI, I2C, USART, USB, CAN, ADC, PWM
- Datasheet Link

2. ESP32-C3FN4

- **Core:** RISC-V Single-core 32-bit
- **Frequency:** 160 MHz
- **Flash Memory:** 4 MB
- **RAM:** 400 KB SRAM
- **Connectivity:** Wi-Fi (802.11 b/g/n), Bluetooth 5 (LE)
- Datasheet Link

3. SIM800C (with Bluetooth)

- **Frequency Bands:** GSM 850/900/1800/1900 MHz
- **GPRS Class:** 12/10
- **Bluetooth Version:** 3.0
- **AT Command Set:** 3GPP TS 27.007, 27.005
- Datasheet Link

4. CP2102 USB-to-Serial Converter

- Used for programming and debugging STM32F103 and ESP32-C3
- Supports USB 2.0 full-speed

5. External EEPROM (24LC64)

- **Capacity:** 64 Kbit
- **Interface:** I2C

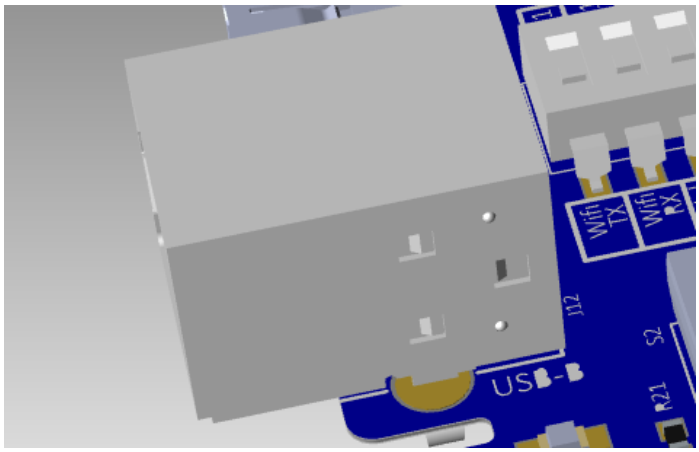
6. Power Specifications

- **Input Voltage:** 4.5V to 50V
- **Maximum Current Draw:** 1A

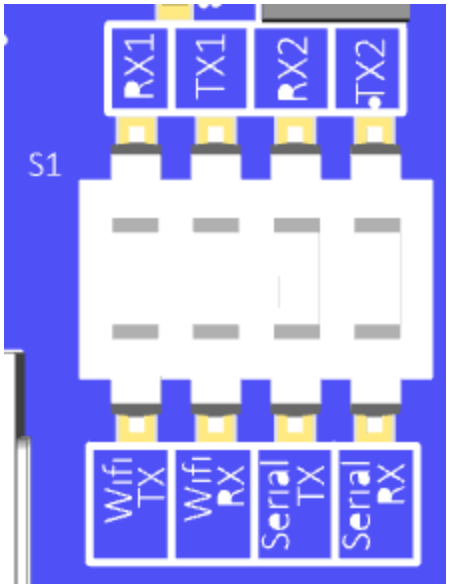
How to Use

Programming the STM32F103C8

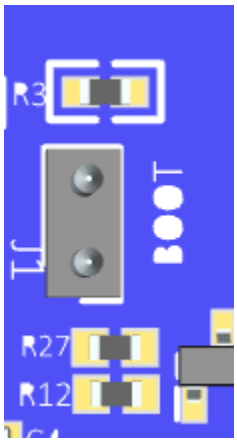
1. Connect the USB Type-B cable to the **J12** port. The **RED PWR LED** will light up.



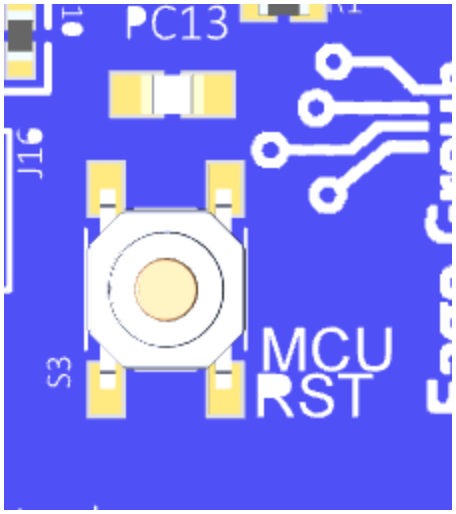
2. Configure the **DIP Switch S1**:
    - Set **Selector 1 (Serial RX)** and **Selector 2 (Serial TX)** to **ON**.
    - **Switches 3 and 4** should be **OFF** (opposite of 1 and 2).



3. Place a **jumper cap** on the **BOOT J**.



4. Press the **MCU RST (Reset)** button. The **White LED** will light during programming.



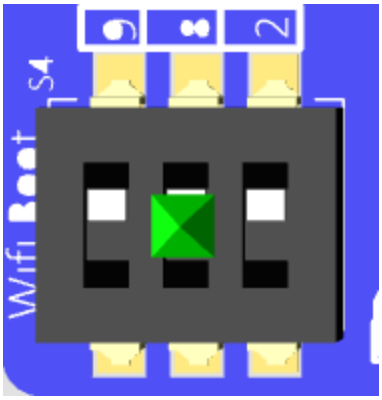
5. After programming:
    - Remove the **jumper cap** from the boot.
    - Press **Reset** to exit boot mode.

Breakout Pins

- **J9 and J6** are breakout pins for STM32.
- **SIM800C** is connected to **Serial2 (PA3, PA2)** of the STM32.

Programming the ESP32-C3

1. Connect the USB Type-B cable to **J12**. The **RED PWR LED** will light up.
2. Set the **DIP Switch S1 (4-Way DIP)**:
    - Turn **Selector 3 (Wi-Fi RX)** and **Selector 4 (Wi-Fi TX) ON**.
    - **Selectors 1 and 2** should be **OFF**.
3. Switch **ESP Power (S2)** to **ON**. The ESP power LED will light up.
4. Configure the **BOOT DIP (S4, 3-Way DIP)**:
    - Set **Switches 1, 2, and 3** to **ON**.



5. Press the **Wi-Fi Reset** button.

6. Proceed to program the ESP32.
7. After programming:
    - When Arduino IDE shows "**Hard resetting via RTS pin**," toggle **Switches 1 and 2** on **S4** to return ESP32 to normal operation.

Breakout Pins

- **J16** serves as breakout pins for the ESP32.

Using the GSM (SIM800C)

Power On:

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

1. digitalWrite(PA8, LOW);

- The **Status** and **GPRS (NET)** LEDs will begin blinking.

Serial Communication:
HardwareSerial GSMSerial(PA3, PA2);

2. GSMSerial.begin(9600);
3. **SIM Card:**
   - Insert the SIM card as shown in the image on the board's bottom side.
   - Connect the **GSM Antenna** for GSM functionality and **BT Antenna** for Bluetooth.

DIP Switch Configuration Table

Function	Selector 1	Selector 2	Selector 3	Selector 4
USB <-> STM32	**ON**	**ON**	**OFF**	**OFF**
USB <->ESP32-C3	**OFF**	**OFF**	**ON**	**ON**
USB <-> External Serial	**OFF**	**OFF**	**OFF**	**OFF**

Use **J13 male header** near DIP for status monitoring.

- **J13** also serves as an external USB-to-serial debugging/programming interface.

Additional Features

- **Inter-MCU Communication:**
    - Use **J4** jumper to connect **STM32 TX3** to **ESP RX** and **STM32 RX3** to **ESP TX**.
    - These pins can be repurposed as I/O if not used for serial communication.
- **External Debugging:**
    - Use **J13** near the DIP switch **S1** as an external USB-to-serial debugging interface for other devices.

Notes

- Always connect antennas when using GSM or Bluetooth functionalities to ensure optimal performance.
- Verify DIP switch configurations before programming to avoid connection issues.