

# GL-S10 二次开发指南 V1.0 -- GL-iNet

版本	更改项	作者	日期
1.0	第一版	何丰	2020.04.15
1.1	添加修改 efuse	何丰	2021.03.20

## 1. 搭建开发环境

GL-S10 主模块采用乐鑫的 ESP32-WROOM-32U，开发环境可基于 window、linux 和 macOS。用户可按自己的编程喜好选择基于 C 语音的 ESP-IDF 平台（乐鑫官方开源 SDK）、AWS freeRTOS 平台（AWS 官方开源 SDK）或者基于 C++ 的 arduino IDE 平台（乐鑫官方提供组件）。

请按照您选择的开发平台搭建环境：

ESP-IDF: <https://docs.espressif.com/projects/esp-idf/en/v3.3/get-started-cmake/index.html>

AWS freeRTOS :

[https://docs.aws.amazon.com/freertos/latest/userguide/getting\\_started\\_espressif.html](https://docs.aws.amazon.com/freertos/latest/userguide/getting_started_espressif.html)

Arduino IDE: <https://github.com/espressif/arduino-esp32>

注：本指南以下内容皆以 ESP-IDF（linux）为开发环境编写，部分步骤在其他平台不能通用。

## 2. 样例源码下载及编译

样例源码仓库: <https://github.com/gl-inet/s10>

编译步骤：

注：编译样例前请先确认您的开发环境（ESP-IDF）是否已搭建成功。

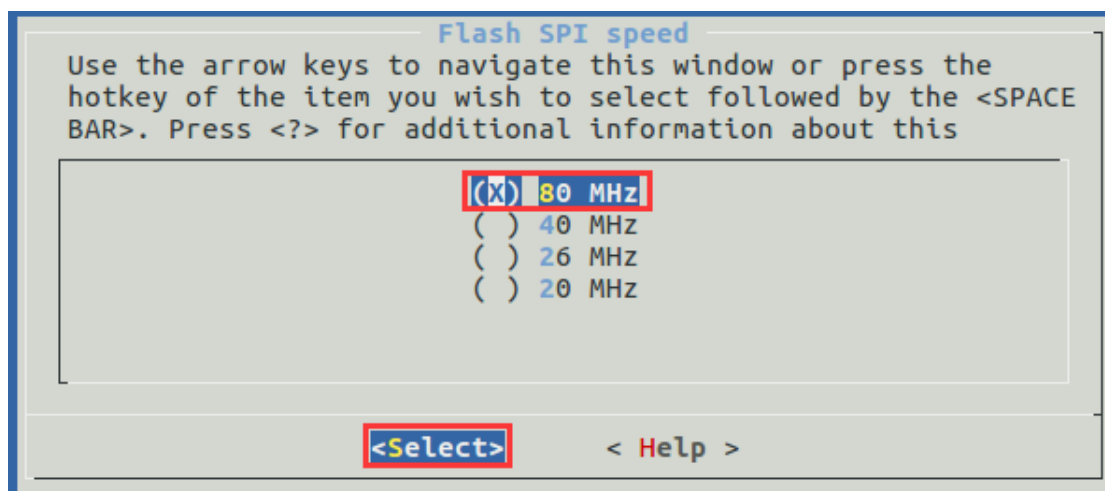
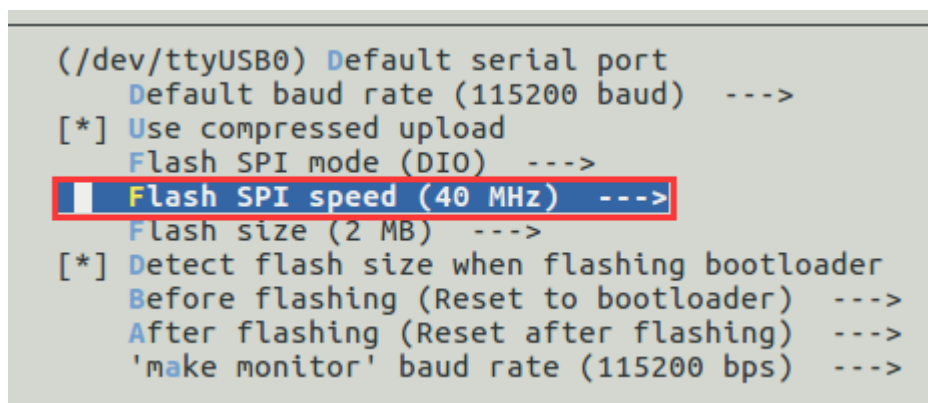
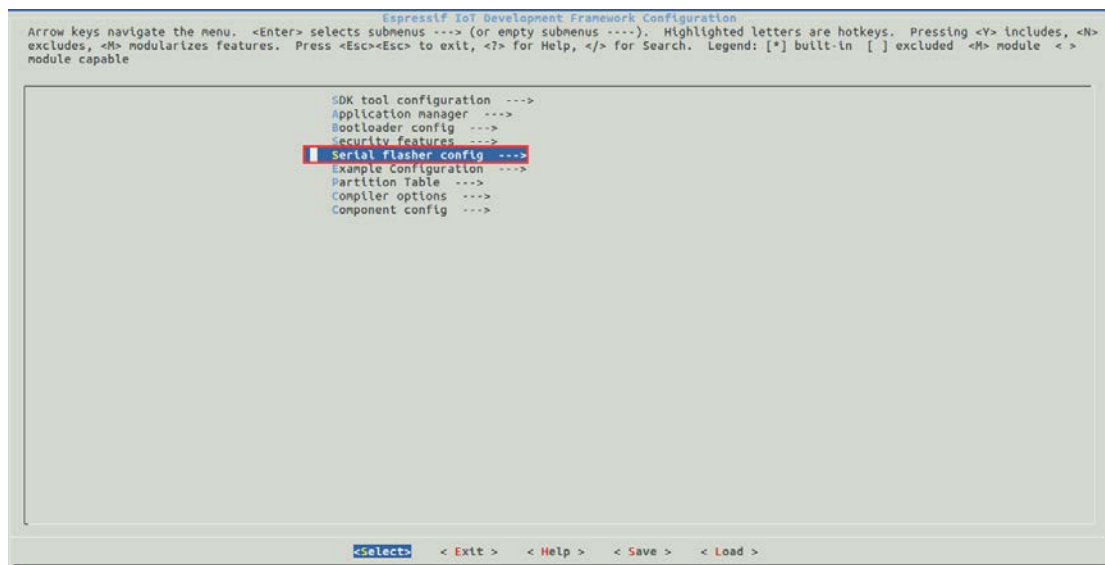
### 1. 样例源码下载：

```
mkdir esp32
cd esp32
git clone
cd demo/BLE-Gateway-Demo
```

### 2. 配置：

```
make menuconfig
```

（1）修改 flash SPI speed 为 80MHz 和 flash size 为 4MB



```

(/dev/ttyUSB0) Default serial port
Default baud rate (115200 baud) --->
[*] Use compressed upload
Flash SPI mode (DIO) --->
Flash SPI speed (80 MHz) --->
Flash size (2 MB) --->
[*] Detect flash size when flashing bootloader
Before flashing (Reset to bootloader) --->
After flashing (Reset after flashing) --->
'make monitor' baud rate (115200 bps) --->

```

**Flash size**

Use the arrow keys to navigate this window or press the hotkey of the item you wish to select followed by the <SPACE BAR>. Press <?> for additional information about this

( ) 1 MB  
 ( ) 2 MB  
 (X) 4 MB  
 ( ) 8 MB  
 ( ) 16 MB

<Select>      < Help >

(2) 修改 partition table (分区表)

```

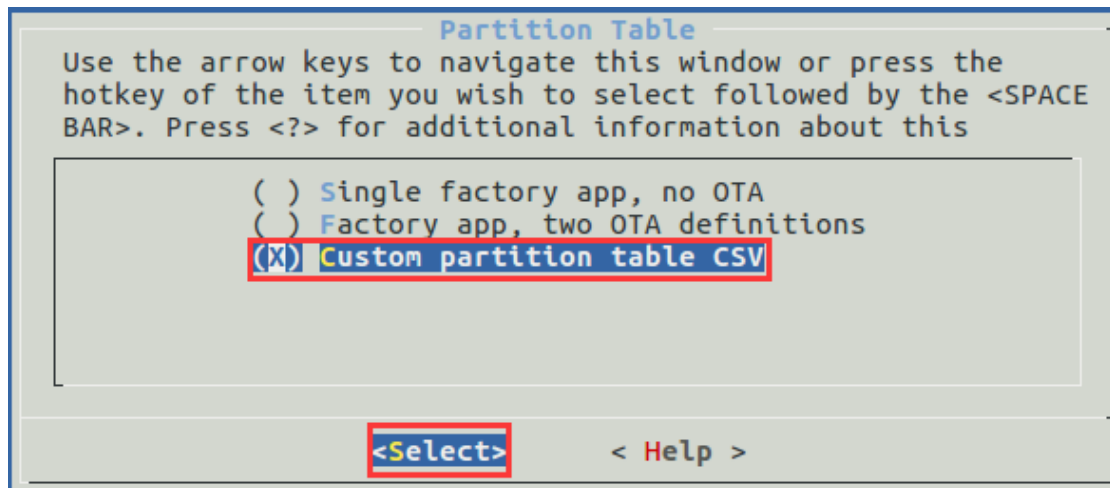
SDK tool configuration --->
Application manager --->
Bootloader config --->
Security features --->
Serial flasher config --->
Example Configuration --->
Partition Table --->
Compiler options --->
Component config --->

```

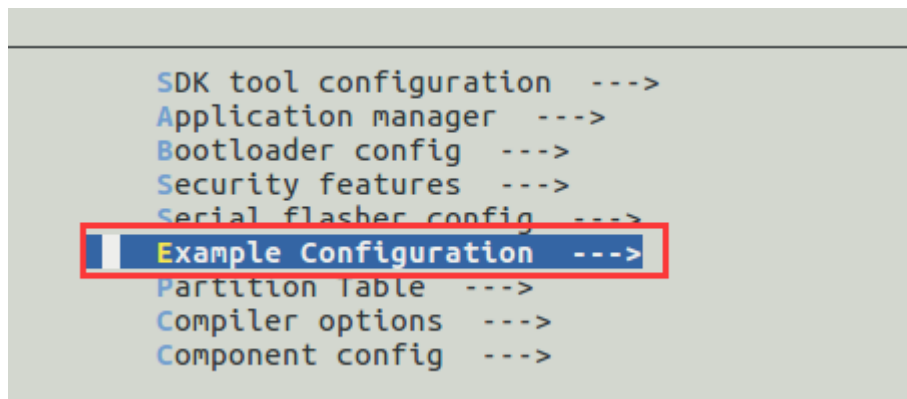
**Partition Table (Single factory app, no OTA) --->**

(0x8000) Offset of partition table

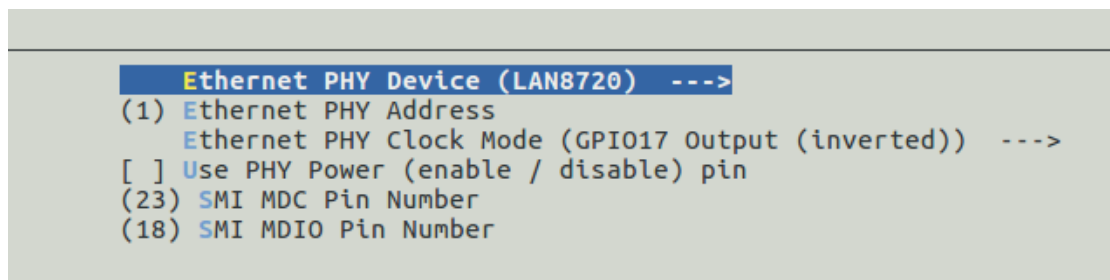
[\*] Generate an MD5 checksum for the partition table



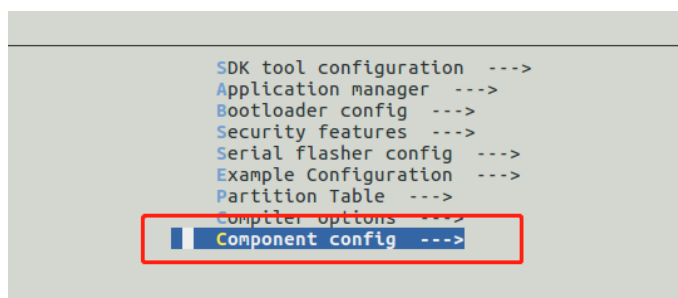
### (3) 配置网口



修改成如下图所示:



### (4) 配置 efuse



```
Application Level Tracing --->
[ ] Amazon Web Services IoT Platform ----
Bluetooth --->
Driver Configurations --->
  eFuse Bit Manager ---->
  ESP32-specific --->
Wi-Fi --->
PHY --->
Power Management --->
ADC-Calibration --->
Event Loop Library --->
ESP HTTP client --->
```

```
[*] Use custom eFuse table
(main/esp_efuse_custom_table.csv) Custom eFuse CSV file
Simulate eFuse operations in RAM
Coding Scheme Compatibility (None Only) --->
```

### 3. 编译:

make all -j4

初次编译用时会较长，当打印如下信息时即为编译完成。

```
cc build/bt/bootloader/ota/ota/esp32_ble_bridge_ota.o
AR build/bt/libbt.a
Generating libbt.a.sections_info
Generating esp32.project.ld
LD build/gl-s10.elf
esptool.py v2.8-dev
To flash all build output, run "make flash" or:
python /mnt/hgfs/workspace/esp32/IDF/esp-idf/v3.3/components/esptool_py/esptool/esptool.py --chip esp32 --port /dev/ttyUSB0 --baud 115200 --before default_reset --after hard_reset write_flash -z --flash mode dio --flash freq 80m --flash size detect 0xd000 /mnt/hgfs/workspace/esp32/project/esp32_ble_bridge/build/ota_data_initial.bin 0x1000 /mnt/hgfs/workspace/esp32/project/esp32_ble_bridge/build/bootloader/bootloader.bin 0x10000 /mnt/hgfs/workspace/esp32/project/esp32_ble_bridge/build/gl-s10.bin 0x8000 /mnt/hgfs/workspace/esp32/project/esp32_ble_bridge/build/partitions.bin
root@ubuntu:/mnt/hgfs/workspace/esp32/project/esp32_ble_bridge#
```

## 3. 固件烧录、debug 调试

### 3.1 串口连接 GL-S10 与 PC

GL-S10 引出了两路 uart，uart0 用于烧录和 debug 打印；uart1 目前并未使用，留给客户按需接入外设。



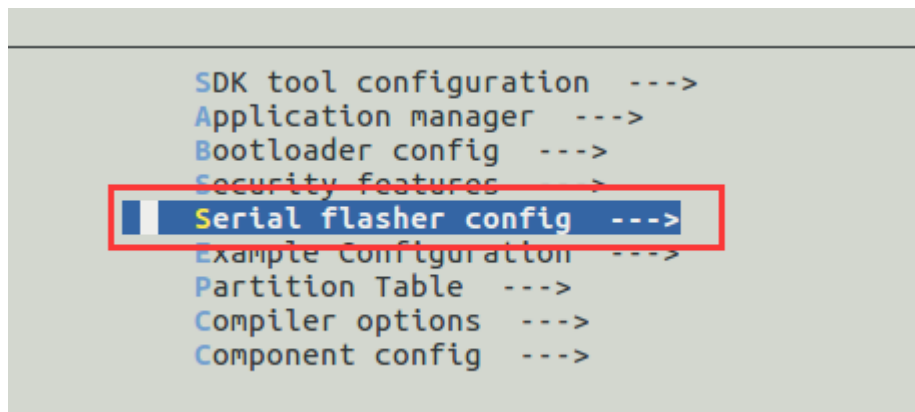
如图所示，uart0 从右往左分别是：TX/RX/GND。用户可使用 USB 转 uart 将 GL-S10 连接到 PC 上。Uart0 默认的波特率为 115200，数据位 8，停止位 1，校验位 0。



## 3.2 固件烧录



开始烧录前，请先将 GL-S10 连接至 PC，按住图中所示的 **BUTTON1**，然后上电。请注意此时的 GL-S10 连接的是否是 `/dev/ttyUSB0`。如果不是，请修改配置表中的串口默认配置，如下图所示



```
(//dev/ttyUSB0) Default serial port
Default baud rate (115200 baud) --->
[*] Use compressed upload
Flash SPI mode (DIO) --->
Flash SPI speed (80 MHz) --->
Flash size (4 MB) --->
[*] Detect flash size when flashing bootloader
Before flashing (Reset to bootloader) --->
After flashing (Reset after flashing) --->
'make monitor' baud rate (115200 bps) --->
```

如果您使用 linux 的 esp-idf 环境，可以直接使用“**make flash**”指令烧录。当打印出如图所示的信息，表示当前已烧录完成。

```
Toolchain version: crosstool-ng-1.22.0-80-g6c4433a
Compiler version: 5.2.0
Python requirements from /mnt/hgfs/workspace/esp32/IDF/esp-idf-v3.3/requirements.txt are satisfied.

App "gl-s10" version: 1.10
Flashing binaries to serial port /dev/ttyUSB0 (app at offset 0x10000)...
esptool.py v2.8-dev
Serial port /dev/ttyUSB0
Connecting.....
Chip is ESP32D0WDQ5 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: 98:f4:ab:0a:8f:04
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Auto-detected Flash size: 4MB
Compressed 8192 bytes to 31...
Wrote 8192 bytes (31 compressed) at 0x0000d000 in 0.0 seconds (effective 7225.6 kbit/s)...
Hash of data verified.
Compressed 24784 bytes to 14744...
Wrote 24784 bytes (14744 compressed) at 0x00001000 in 1.3 seconds (effective 152.2 kbit/s)...
Hash of data verified.
Compressed 1284560 bytes to 771285...
Wrote 1284560 bytes (771285 compressed) at 0x00010000 in 69.5 seconds (effective 147.8 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 128...
Wrote 3072 bytes (128 compressed) at 0x00008000 in 0.0 seconds (effective 1444.5 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
```

如果您使用的是 window 的环境，并且从样例仓库的 release 目录下下载了固件，可以直接使用 flash 下载工具烧录固件（不需要 IDF 环境）。

Flash 工具下载地址：  
[https://www.espressif.com/sites/default/files/tools/flash\\_download\\_tools\\_v3.6.8\\_0.zip](https://www.espressif.com/sites/default/files/tools/flash_download_tools_v3.6.8_0.zip)

### 3.3 debug 调试

debug 调试也是通过烧录的串口进行。在 GL-S10 烧录完成后，输入“**make monitor**”。当打印出如图所示的信息后，将 GL-S10 重新上电，即可看到 debug 打印。



```
root@ubuntu:~# cd /mnt/hgfs/workspace/esp32/IDF/esp-idf-v3.3/
Toolchain path: /home/keidy/esp32/crosstool-ng/xtensa-esp32-elf/bin/xtensa-esp32-elf-gcc
Toolchain version: crosstool-ng-1.22.0-80-g6c4433a
Compiler version: 5.2.0
Python requirements from /mnt/hgfs/workspace/esp32/IDF/esp-idf-v3.3/requirements.txt are satisfied.
MONITOR
--- idf_monitor on /dev/ttyUSB0 115200 ---
--- Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
```

## 4. OTA 升级固件

出厂的 GL-S10 固件都带有 OTA 升级的功能。如果您不希望通过拆壳连接串口烧录，也可以使用 OTA 更换固件。

GL-S10 的 OTA 是通过 http 访问服务器获取固件，相关操作步骤见 GL-S10 BLE-Gateway User Guide