

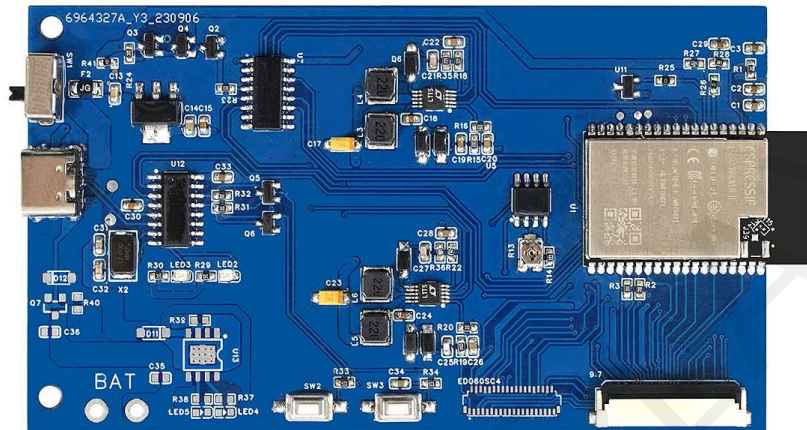


# PARALLEL EPAPER DISPLAY EVALUATION KIT-epdiy



Dalian Good Display Co., Ltd.

# Product Specifications



<b>Customer</b>	<b>Standard</b>
<b>Description</b>	<b>Evaluation Kit For E-paper Display</b>
<b>Model Name</b>	<b>epdiy</b>
<b>Date</b>	<b>2023/10/20</b>
<b>Revision</b>	<b>1.0</b>

	Design Engineering		
	Approval	Check	Design
			

Zhongnan Building, No.18, Zhonghua West ST,Ganjingzi DST,Dalian,CHINA

Tel: +86-411-84619565

Email: info@good-display.com

Website: www.good-display.com

## Contents

1. Overview .....	4
2. Mechanical Specifications .....	4
3. Diagram .....	5
4. Flashing Firmware .....	7

GOODDISPLAY

## 1. Overview

epdiy is a controller that supports a variety of affordable parallel ePaper screens (screens often sold as replacement for e-book readers)

(※ Author by vroland/epdiy on GitHub <https://github.com/vroland/epdiy>)

epdiy targets multiple ePaper displays.

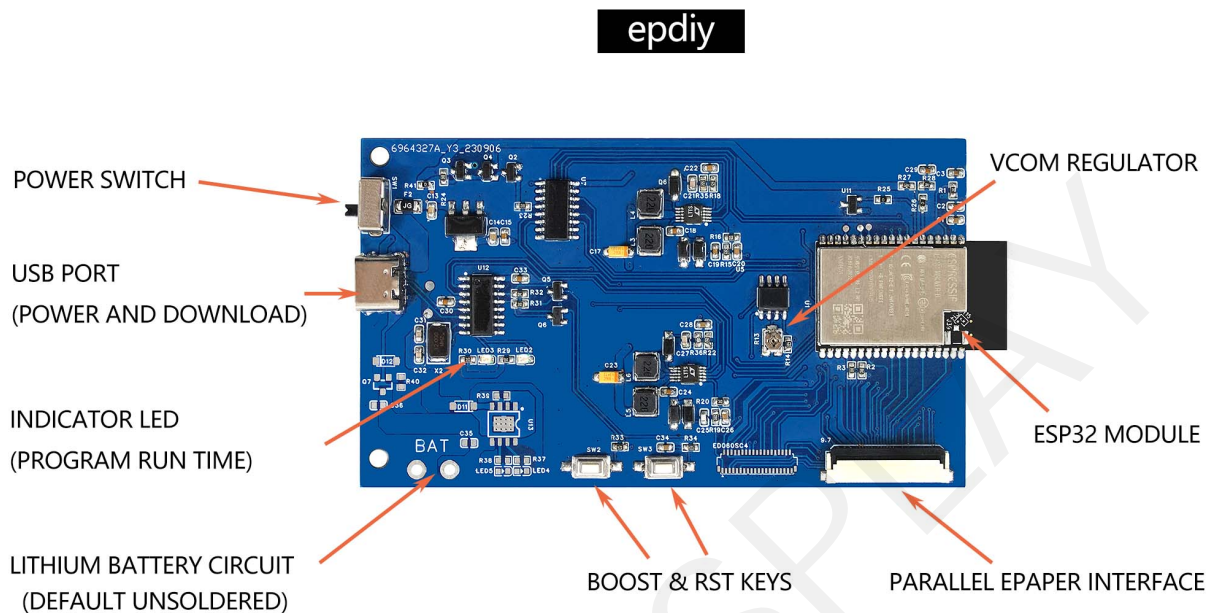
epdiy PCB v5 has a 33-pin connector and can drive 9.7-inch, 13.3-inch and other parallel ePaper displays.

## 2. Structure Specification

Parameter	Specification
Model	epdiy
Platform	ESP-IDF
Dimension	99.8mm x 57.8mm
Power Interface	USB Power supply
Example Code	Available
Operating Temp.	-20℃~70℃(-4℉~158℉ )
Main Function	Learn to drive E-paper display; Test and evaluate e-paper display; Support secondary development
Additional Function	USB to serial port, indicator lights, buttons, current detection, etc.



### 3. Diagram



#### 3.1. Font, Image, Waveform

The ESP32 is quite powerful, but it's still a microcontroller. Therefore, due to limited memory and computing resources, it should perform as much processing as possible. epdiy comes with scripts to convert fonts, images, and waveforms into C headers, and users can simply #include them into the required projects.

#### 3.2. Generate font files

Fonts can only be used by the driver in a special header format that needs to be generated from TTF fonts. If the generated font file with default characters exceeds the size range, it can be modified in intervals fontconvert.py.

#### 3.3. Generate images

The process of converting images is very similar to converting fonts. Run the script using the input image, image name, and output image. The image will be converted to reduced grayscale to fit within and (default is 1200x825). In order to obtain accurate grayscale, it is recommended to use specialized tools to color grade and scale the image before converting it.

### 3.4. Convert waveform

The display is driven by information in so-called waveform files. These specify how pulses are applied to pixels transitioning from one gray tone to another.

### 3.5. Deep sleep current

The I2S peripherals used to drive display will be deinitialized before entering deep sleep, and the pins used by epdiy will be put into a low power state. Normally the deep sleep current should be less than 13 $\mu$ A. If your deep sleep current is much higher, check your connected peripherals. For some modules, you must isolate GPIO 12 before entering deep sleep.

### 3.6. Regulating VCOM value

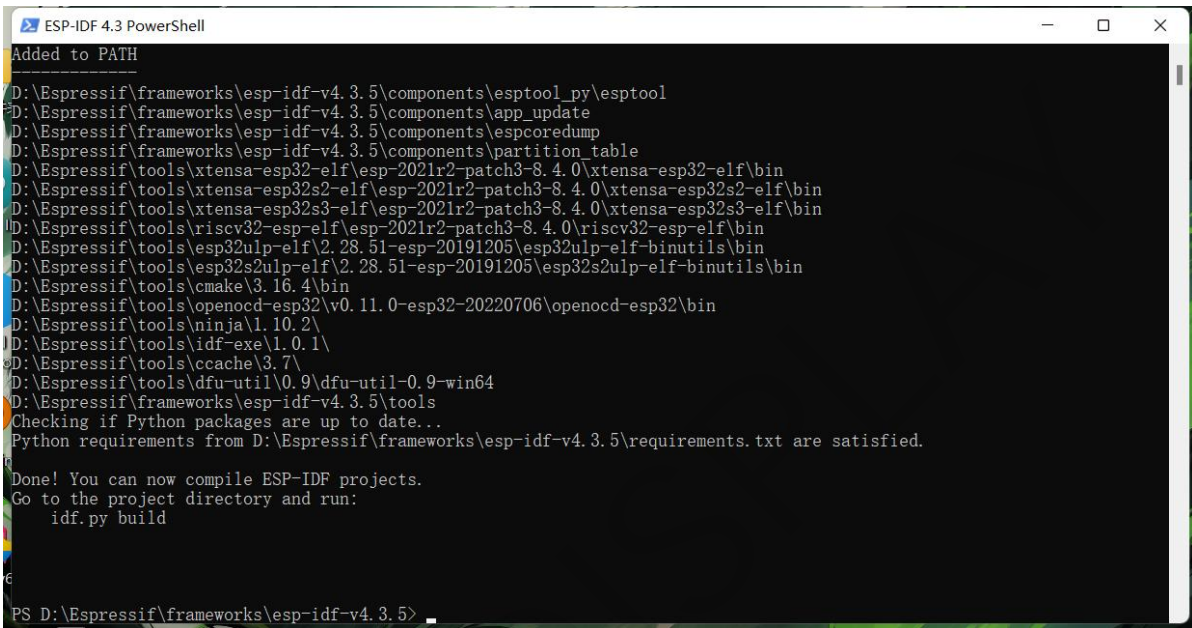
Adjust the VCOM value by adjusting R13 on the board. The VCOM value must be adjusted strictly according to the value on the screen.



## 4. Flashing Firmware

4.1 Download the software (ESP-IDF 4.3 PowerShell) (provided by Espressif's official website) <https://dl.espressif.com/dl/esp-idf/>

The following operations are based on `epdiy-master123\examples\dragon` as an example.



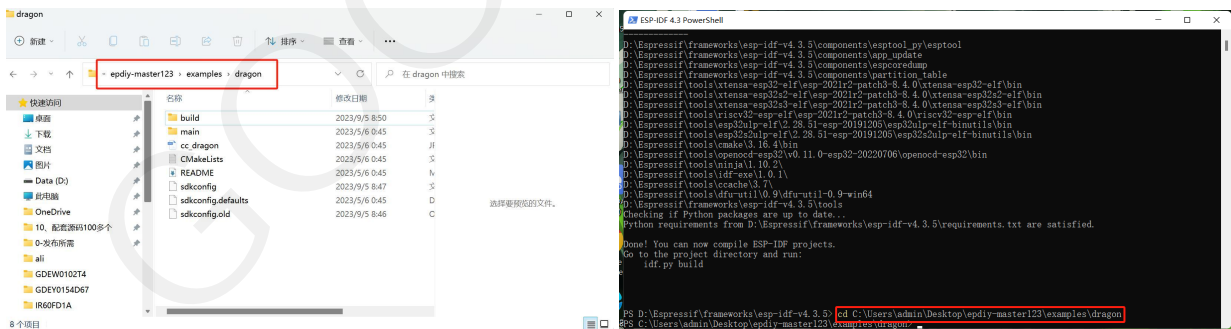
```

ESP-IDF 4.3 PowerShell
Added to PATH
D:\Espressif\frameworks\esp-idf-v4.3.5\components\esptool_py\esptool
D:\Espressif\frameworks\esp-idf-v4.3.5\components\app_update
D:\Espressif\frameworks\esp-idf-v4.3.5\components\espcoredump
D:\Espressif\frameworks\esp-idf-v4.3.5\components\partition_table
D:\Espressif\tools\xtensa-esp32-elf\esp-2021r2-patch3-8.4.0\xtensa-esp32-elf\bin
D:\Espressif\tools\xtensa-esp32s2-elf\esp-2021r2-patch3-8.4.0\xtensa-esp32s2-elf\bin
D:\Espressif\tools\xtensa-esp32s3-elf\esp-2021r2-patch3-8.4.0\xtensa-esp32s3-elf\bin
D:\Espressif\tools\riscv32-esp-elf\esp-2021r2-patch3-8.4.0\riscv32-esp-elf\bin
D:\Espressif\tools\esp32ulp-elf\2.28.51-esp-20191205\esp32ulp-elf-binutils\bin
D:\Espressif\tools\esp32s2ulp-elf\2.28.51-esp-20191205\esp32s2ulp-elf-binutils\bin
D:\Espressif\tools\cmake\3.16.4\bin
D:\Espressif\tools\openocd-esp32\v0.11.0-esp32-20220706\openocd-esp32\bin
D:\Espressif\tools\ninja\1.10.2\
D:\Espressif\tools\idf-exe\1.0.1\
D:\Espressif\tools\ccache\3.7\
D:\Espressif\tools\dfu-util\0.9\dfu-util-0.9-win64
D:\Espressif\frameworks\esp-idf-v4.3.5\tools
Checking if Python packages are up to date...
Python requirements from D:\Espressif\frameworks\esp-idf-v4.3.5\requirements.txt are satisfied.

Done! You can now compile ESP-IDF projects.
Go to the project directory and run:
idf.py build

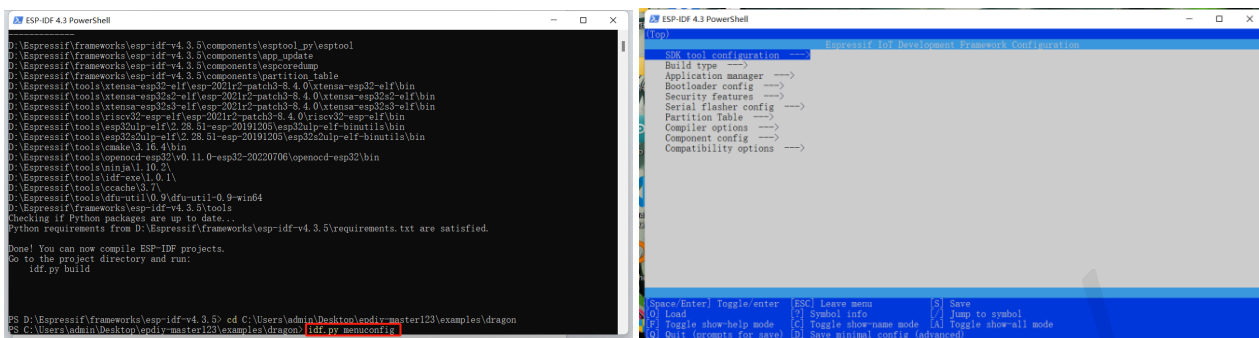
PS D:\Espressif\frameworks\esp-idf-v4.3.5>
  
```

4.2 First enter the `cd` program address and press Enter. Enter the program address to make the compiler jump to the program address.

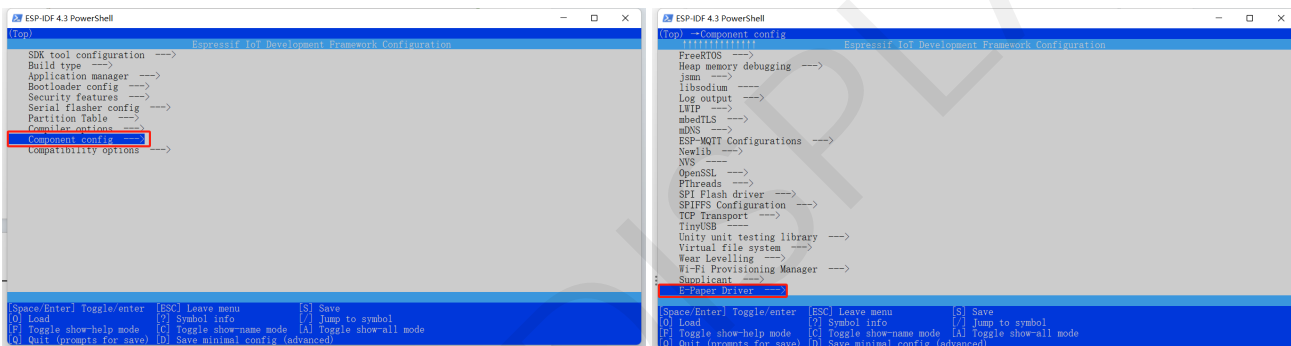


The image shows two windows. On the left is a Windows File Explorer window showing the directory `epdiy-master123\examples\dragon`. The files listed include `build`, `main`, `cc_dragon`, `CMakeLists`, `README`, `idf.py`, `sdccconfig.defaults`, and `sdccconfig.old`. On the right is the ESP-IDF 4.3 PowerShell terminal window. The command `cd C:\Users\admin\Desktop\epdiy-master123\examples\dragon` has been entered, and the prompt has changed to `PS C:\Users\admin\Desktop\epdiy-master123\examples\dragon>`.

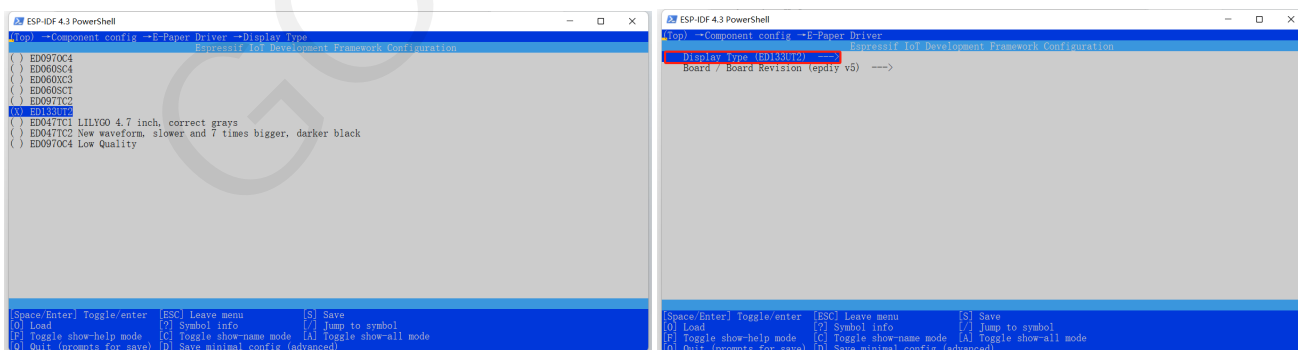
4.3 Set the screen model, enter idf.py menuconfig and press Enter to jump to the program system.



#### 4.3.1 Select Component config and find ePaper driver

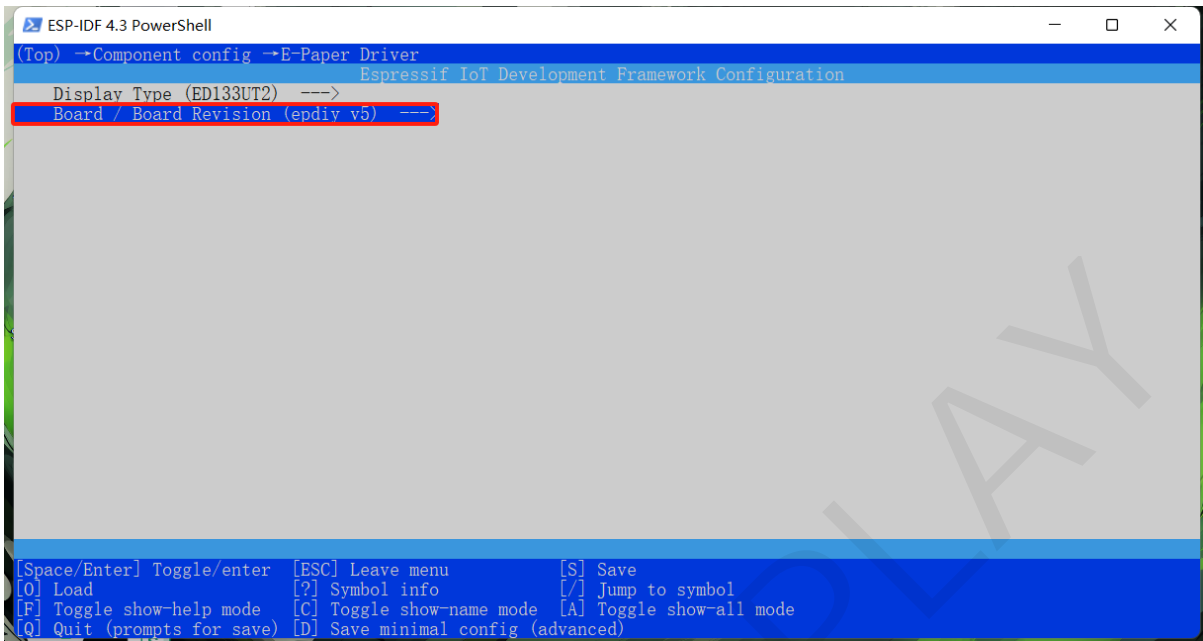


#### 4.3.2 Select Component config and find ePaper driver



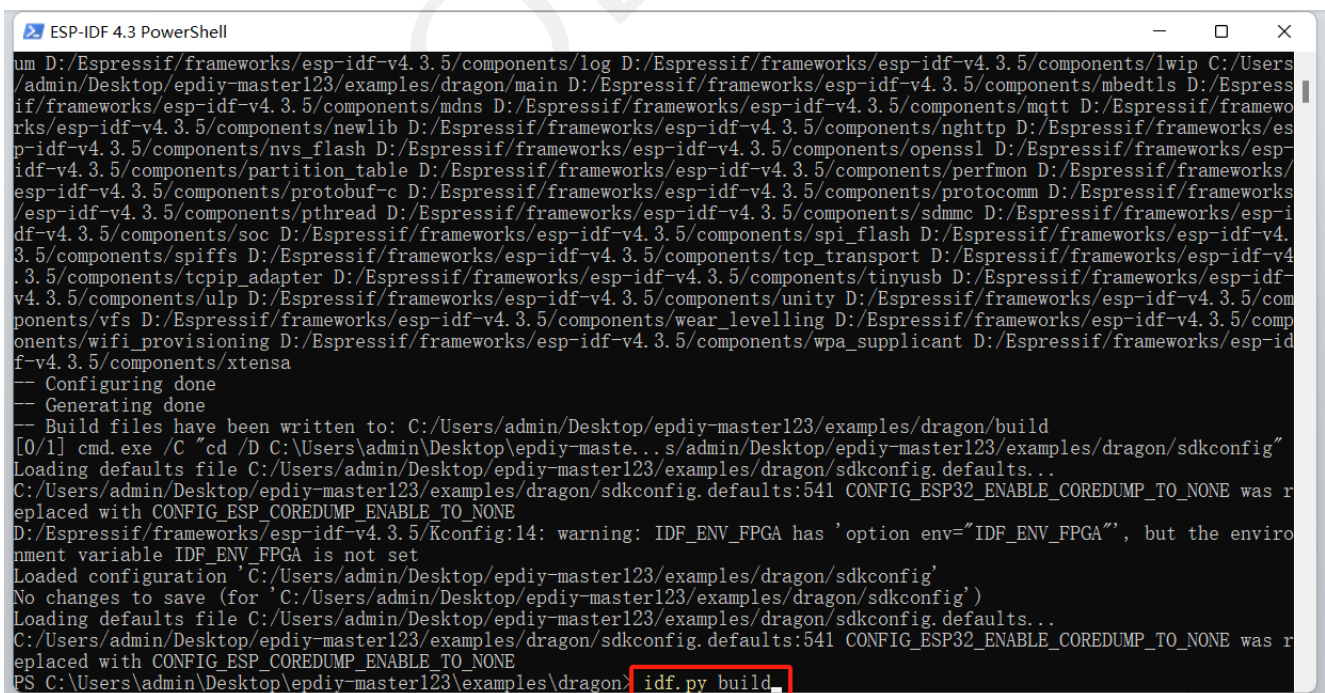


#### 4.3.3 Select Component config and find ePaper driver



#### 4.3.4. After saving the configuration, press the shortcut key Q to exit the system

#### 4.4. Compile the code, enter idf.py build to start compiling the program



#### 4.5. Flashing firmware, enter idf.py flash monitor to start

```
ESP-IDF 4.3 PowerShell
n/Desktop/epdiy-master123/examples/dragon/build/ldgen_libraries --objdump D:/Espressif/tools/xtensa-esp32-elf/esp-2021r2-patch3-8.4.0/xtensa-esp32-elf/bin/xtensa-esp32-elf-objdump.exe"
D:/Espressif/tools/xtensa-esp32-elf/esp-2021r2-patch3-8.4.0/xtensa-esp32-elf/bin/xtensa-esp32-elf-objdump.exe: cpu_util.c.obj: file format not recognized
D:/Espressif/tools/xtensa-esp32-elf/esp-2021r2-patch3-8.4.0/xtensa-esp32-elf/bin/xtensa-esp32-elf-objdump.exe: regi2c_ctrl.c.obj: file format not recognized
D:/Espressif/tools/xtensa-esp32-elf/esp-2021r2-patch3-8.4.0/xtensa-esp32-elf/bin/xtensa-esp32-elf-objdump.exe: rtc_clk.c.obj: file format not recognized
D:/Espressif/tools/xtensa-esp32-elf/esp-2021r2-patch3-8.4.0/xtensa-esp32-elf/bin/xtensa-esp32-elf-objdump.exe: rtc_clk_init.c.obj: file format not recognized
D:/Espressif/tools/xtensa-esp32-elf/esp-2021r2-patch3-8.4.0/xtensa-esp32-elf/bin/xtensa-esp32-elf-objdump.exe: rtc_init.c.obj: file format not recognized
D:/Espressif/tools/xtensa-esp32-elf/esp-2021r2-patch3-8.4.0/xtensa-esp32-elf/bin/xtensa-esp32-elf-objdump.exe: rtc_pm.c.obj: file format not recognized
D:/Espressif/tools/xtensa-esp32-elf/esp-2021r2-patch3-8.4.0/xtensa-esp32-elf/bin/xtensa-esp32-elf-objdump.exe: rtc_sleep.c.obj: file format not recognized
Traceback (most recent call last):
  File "D:\Espressif\frameworks\esp-idf-v4.3.5\tools\ldgen\ldgen.py", line 178, in <module>
    main()
  File "D:\Espressif\frameworks\esp-idf-v4.3.5\tools\ldgen\ldgen.py", line 134, in main
    dump = StringIO(subprocess.check_output([objdump, '-h', library]).decode())
  File "subprocess.py", line 466, in check_output
  File "subprocess.py", line 571, in run
subprocess.CalledProcessError: Command '['D:/Espressif/tools/xtensa-esp32-elf/esp-2021r2-patch3-8.4.0/xtensa-esp32-elf/bin/xtensa-esp32-elf-objdump.exe', '-h', 'C:/Users/admin/Desktop/epdiy-master123/examples/dragon/build/esp-idf/esp_hw_support/libesp_hw_support.a']' returned non-zero exit status 1.
ninja: build stopped: subcommand failed.
ninja failed with exit code 1
PS C:\Users\admin\Desktop\epdiy-master123\examples\dragon> idf.py flash monitor.
```

#### 4.6. Final result

