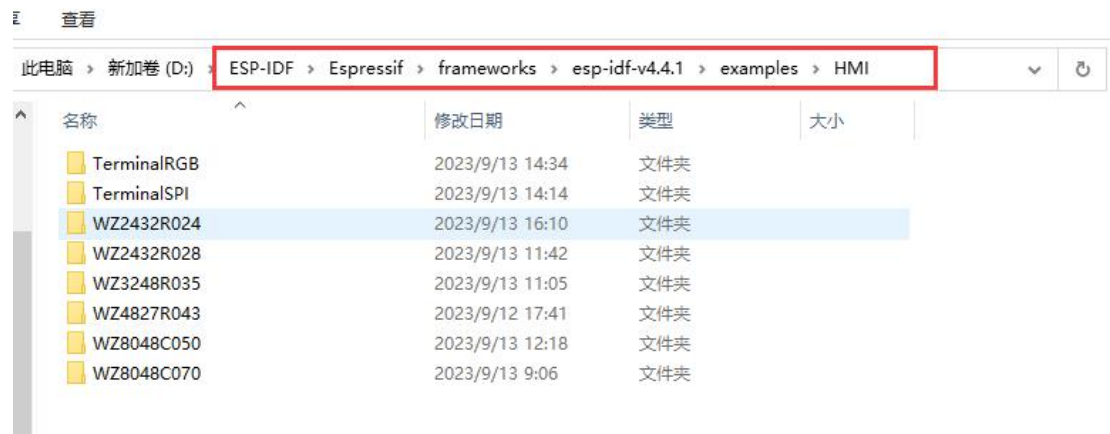


WZ2432R028 Use a tutorial

Place the downloaded project under the IDF directory (as shown below):



Let's first learn at the use of commands :

cd xxx---Moving to the xxx directory, xxx represents the name of the directory, for example: cd example

idf.py set-target esp32s3---Set the target chip for example: esp32s3

idf.py fullclean---Delete the entire build directory, including all the CMake configuration output files.

idf.py clean---It removes the building output files from the building directory and cleans up the entire project..

idf.py menuconfig---Configure the target chip

idf.py build---Compile a private code base

idf.py -p com3 flash---Download the program to the target chip

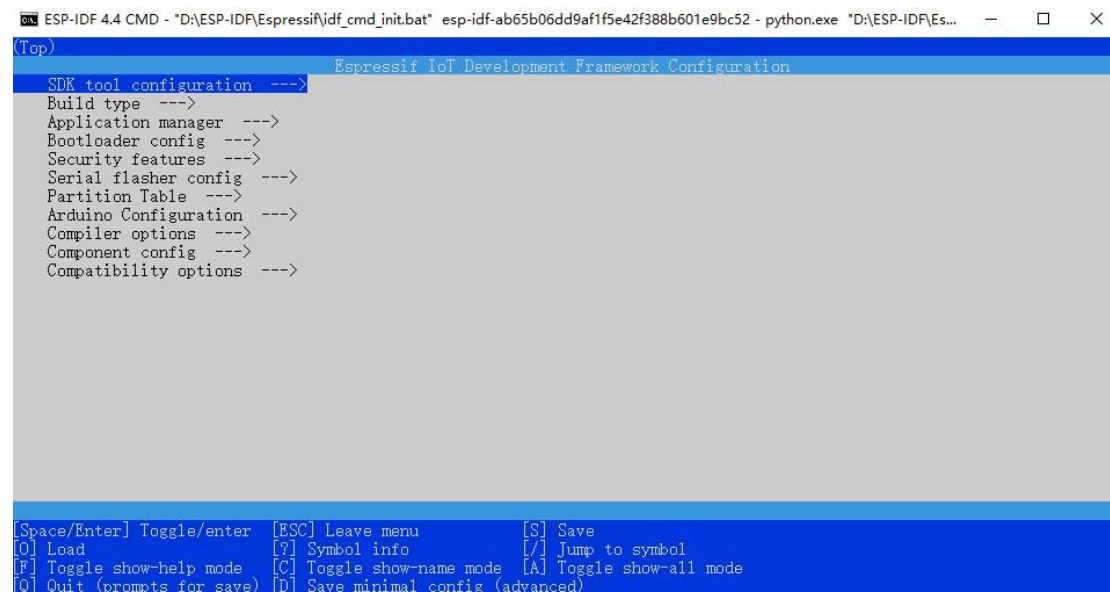
idf.py -p com3 flash monitor---Once compile burn and open monitoring

Now we open the terminal and go to the WZ2432R028 project catalog

```
D:\ESP-IDF\Esspressif\frameworks\esp-idf-v4.4.1\examples\HMI>cd WZ2432R028
D:\ESP-IDF\Esspressif\frameworks\esp-idf-v4.4.1\examples\HMI\WZ2432R028>
```

Now we have to empty the project `idf.py fullclean` once first, and then go into the configuration

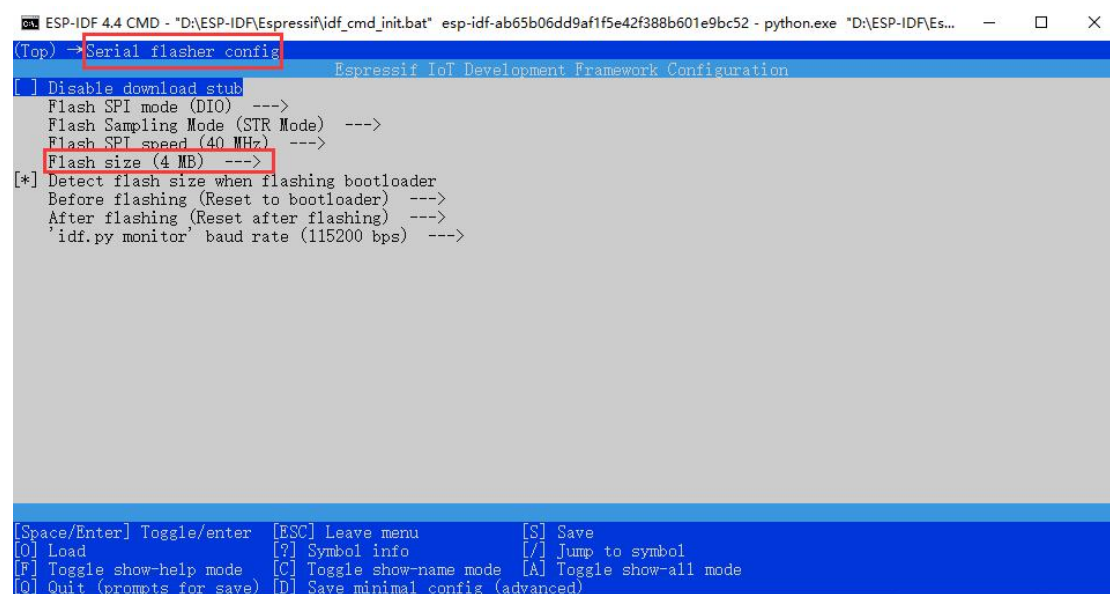
```
D:\ESP-IDF\Esspressif\frameworks\esp-idf-v4.4.1\examples\HMI\WZ2432R028>idf.py fullclean
Executing action: fullclean
Done
```



```
ESP-IDF 4.4 CMD - "D:\ESP-IDF\Esspressif\idf_cmd_init.bat" esp-idf-ab65b06dd9af1f5e42f388b601e9bc52 - python.exe "D:\ESP-IDF\Es...
(Top)
Espressif IoT Development Framework Configuration
SDK tool configuration --->
Build type --->
Application manager --->
Bootloader config --->
Security features --->
Serial flasher config --->
Partition Table --->
Arduino Configuration --->
Compiler options --->
Component config --->
Compatibility options --->

[Space/Enter] Toggle/enter  [ESC] Leave menu          [S] Save
[O] Load                   [?] Symbol info          [/] Jump to symbol
[F] Toggle show-help mode  [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)
```

Now modify the options by following the following steps:



```
ESP-IDF 4.4 CMD - "D:\ESP-IDF\Esspressif\idf_cmd_init.bat" esp-idf-ab65b06dd9af1f5e42f388b601e9bc52 - python.exe "D:\ESP-IDF\Es...
(Top) -> Serial flasher config
Espressif IoT Development Framework Configuration
[ ] Disable download stub
Flash SPI mode (DIO) --->
Flash Sampling Mode (STR Mode) --->
Flash SPI speed (40 MHz) --->
Flash size (4 MB) --->
[*] Detect flash size when flashing bootloader
Before flashing (Reset to bootloader) --->
After flashing (Reset after flashing) --->
'idf.py monitor' baud rate (115200 bps) --->

[Space/Enter] Toggle/enter  [ESC] Leave menu          [S] Save
[O] Load                   [?] Symbol info          [/] Jump to symbol
[F] Toggle show-help mode  [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)
```

```
ESP-IDF 4.4 CMD - "D:\ESP-IDF\Espressif\idf_cmd_init.bat" esp-idf-ab65b06dd9af1f5e42f388b601e9bc52 - python.exe "D:\ESP-IDF\Es...
(Top) → Partition Table
Espressif IoT Development Framework Configuration
Partition Table (Single factory app (large), no OTA) --->
(0x8000) Offset of partition table
[*] Generate an MD5 checksum for the partition table

[Space/Enter] Toggle/enter [ESC] Leave menu [S] Save
[Q] Load [?] Symbol info [/] Jump to symbol
[F] Toggle show-help mode [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)
```

```
ESP-IDF 4.4 CMD - "D:\ESP-IDF\Espressif\idf_cmd_init.bat" esp-idf-ab65b06dd9af1f5e42f388b601e9bc52 - python.exe "D:\ESP-IDF\Es...
(Top) → Component config → LVGL configuration
Espressif IoT Development Framework Configuration
[ ] Uncheck this to use custom lv_conf.h
[ ] LVGL minimal configuration.
Color settings --->
Memory settings --->
HAL Settings --->
Feature configuration --->
Font usage --->
Text Settings --->
Widget usage --->
Extra Widgets --->
Themes --->
Layouts --->
3rd Party Libraries --->
Others --->
Examples --->
Demos --->

[Space/Enter] Toggle/enter [ESC] Leave menu [S] Save
[Q] Load [?] Symbol info [/] Jump to symbol
[F] Toggle show-help mode [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)
```

```
ESP-IDF 4.4 CMD - "D:\ESP-IDF\Espressif\idf_cmd_init.bat" esp-idf-ab65b06dd9af1f5e42f388b601e9bc52 - python.exe "D:\ESP-IDF\Es...
(Top) → Component config → TFT_eSPI
Espressif IoT Development Framework Configuration
Select TFT driver (ILI9341 - 1) --->
Define the colour order (BGR) --->
[ ] M5Stack
Color inversion correction (None) --->
[ ] Enable 8-bit parallel mode (otherwise SPI is assumed)
Display SPI config --->
Control Pin configuration --->
Fonts --->
Touch screen configuration --->

[Space/Enter] Toggle/enter [ESC] Leave menu [S] Save
[Q] Load [?] Symbol info [/] Jump to symbol
[F] Toggle show-help mode [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)
```

```
ESP-IDF 4.4 CMD - "D:\ESP-IDF\Espressif\idf_cmd_init.bat" esp-idf-ab65b06dd9af1f5e42f388b601e9bc52 - python.exe "D:\ESP-IDF\Es...
(Top) -> Component config -> TFT_eSPI -> Display SPI config
Espressif IoT Development Framework Configuration

SPI port (VSPI (SPI2)) --->
(12) TFT MISO pin
(13) TFT MOSI pin
(14) TFT Clock pin
[ ] Use SDA line for reading
(16000000) SPI Frequency (Hz)
(20000000) SPI Read Frequency (Hz)

[Space/Enter] Toggle/enter  [ESC] Leave menu          [S] Save
[O] Load                   [?] Symbol info          [/] Jump to symbol
[F] Toggle show-help mode  [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)
```

```
ESP-IDF 4.4 CMD - "D:\ESP-IDF\Espressif\idf_cmd_init.bat" esp-idf-ab65b06dd9af1f5e42f388b601e9bc52 - python.exe "D:\ESP-IDF\Es...
(Top) -> Component config -> TFT_eSPI -> Control Pin configuration
Espressif IoT Development Framework Configuration

(15) TFT Chip Select pin
(2) TFT Data/Command pin
(-1) TFT Reset pin
[*] Enable backlight control
(27) TFT Backlight pin
    Pin state to activate backlight (LOW) --->

[Space/Enter] Toggle/enter  [ESC] Leave menu          [S] Save
[O] Load                   [?] Symbol info          [/] Jump to symbol
[F] Toggle show-help mode  [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)
```

```
ESP-IDF 4.4 CMD - "D:\ESP-IDF\Espressif\idf_cmd_init.bat" esp-idf-ab65b06dd9af1f5e42f388b601e9bc52 - python.exe "D:\ESP-IDF\Es...
(Top) -> Component config -> TFT_eSPI -> Touch screen configuration
Espressif IoT Development Framework Configuration

[*] Enable Touch
(33) Touch chip select pin
(6000000) SPI frequency for XPT2046 chip (Hz)

[Space/Enter] Toggle/enter  [ESC] Leave menu          [S] Save
[O] Load                   [?] Symbol info          [/] Jump to symbol
[F] Toggle show-help mode  [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)
```

Save the exit after setup, and then execute the **idf.py build**

```
ESP-IDF 4.4 CMD - "D:\ESP-IDF\Espresif\idf_cmd_init.bat" esp-idf-ab65b06dd9af1f5e42f388b601e9bc52 - python.exe "D:\ESP-IDF\Espresif\idf_cmd_init.bat"
D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/lwip D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/examples/HMI/WZ2432R024/main D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/mbdltls D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/mdns D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/mqtt D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/newlib D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/nghttp D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/nvs_flash D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/openthread D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/perfmon D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/protobuf-c D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/protocomm D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/pthread D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/sdmmc D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/soc D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/spi_flash D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/spiffs D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/tcp_transport D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/tcpip_adapter D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/tinyusb D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/ulp D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/unity D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/usb D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/vfs D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/wear_levelling D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/wifi_provisioning D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/wpa_supplicant D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/components/xtensa
-- Configuring done
-- Generating done
-- Build files have been written to: D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/examples/HMI/WZ2432R024/build
[7/1614] Generating ../partition_table/partition-table.bin
Partition table binary generated. Contents:
*****
# ESP-IDF Partition Table
# Name, Type, SubType, Offset, Size, Flags
nvs,data,nvs,0x9000,24K,
phy_init,data,phy,0xf000,4K,
factory,app,factory,0x10000,1500K,
*****
[107/1614] Building C object esp-idf/spi_flash/CMakeFiles/ idf_spi_flash.dir/spi_flash_chip_th.c.o
```

Waiting for the compilation to complete, the following figure interface appears:

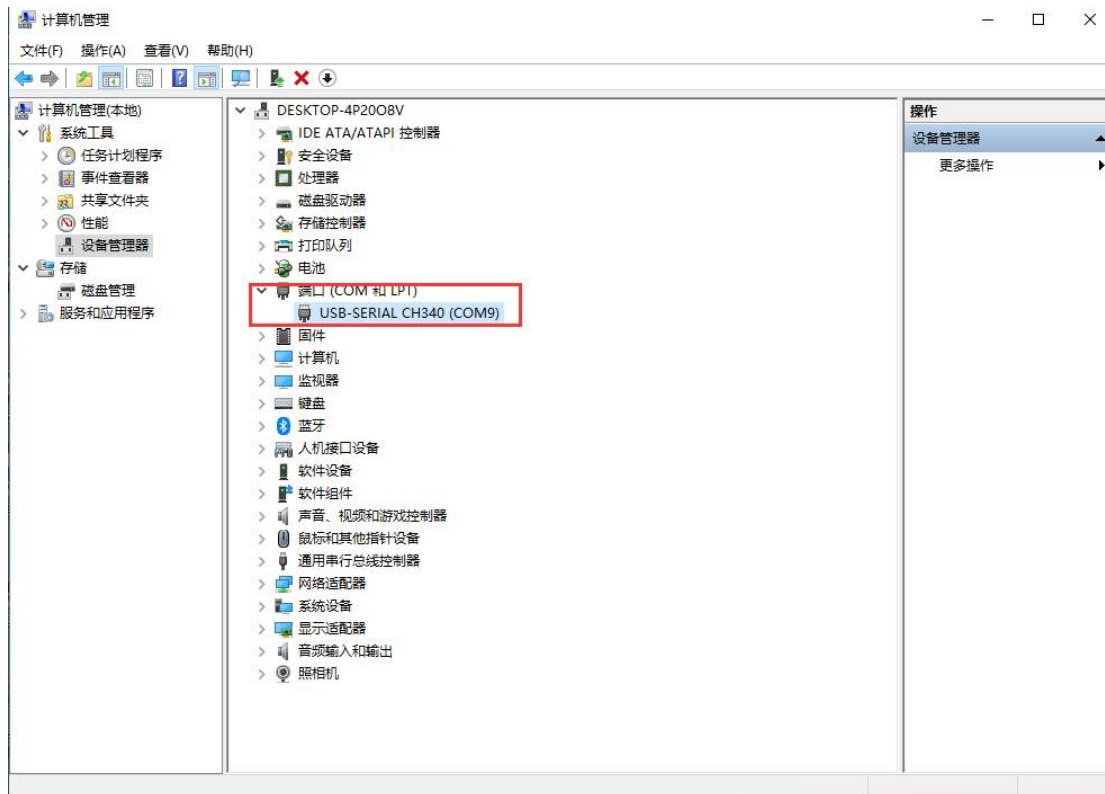
```
ESP-IDF 4.4 CMD - "D:\ESP-IDF\Espresif\idf_cmd_init.bat" esp-idf-ab65b06dd9af1f5e42f388b601e9bc52
void ui_image_set_property(lv_obj_t * target, int id, uint32_t * val);
../components/UI/ui.c:124:16: warning: unused variable 'target' [-Wunused-variable]
    lv_obj_t * target = lv_event_get_target(e);

At top level:
../components/UI/ui.c:44:13: warning: 'anim_x_cb' defined but not used [-Wunused-function]
static void anim_x_cb(void * var, int32_t v)

[1610/1614] Generating ld/sections.ld
warning: the default selection TFT_SPI_2_HOST (undefined) of <choice TFT_SPI_PORT> (defined at D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/examples/HMI/WZ2432R024/components/TFT_eSPI/Kconfig:216) is not contained in the choice
[1613/1614] Generating binary image from built executable
esptool.py v3.3-dev
Creating esp32 image...
Merged 2 ELF sections
Successfully created esp32 image.
Generated D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/examples/HMI/WZ2432R024/build/WZ2432R024.bin
[1614/1614] cmd.exe /C "cd /D D:\ESP-IDF\Espresif\frameworks\esp-idf-v4.4.1/examples/HMI/WZ2432R024/build/WZ2432R024"
WZ2432R024.bin binary size 0xf04a0 bytes. Smallest app partition is 0x177000 bytes. 0x36b60 bytes (36%) free.

Project build complete. To flash, run this command:
D:\ESP-IDF\Espresif\python_env\idf4.4_py3.8_env\Scripts\python.exe ..\..\components\esptool_py\esptool\esptool.py -p (PORT) -b 460800 --before default_reset --after hard_reset --chip esp32 write_flash --flash_mode dio --flash_size det ct --flash_freq 40m 0x1000 build\bootloader\bootloader.bin 0x8000 build\partition_table\partition-table.bin 0x10000 bui a\WZ2432R024 bin
or run 'idf.py -p (PORT) flash'
```


Perform the `idf.py -p com9 flash`



success!

```
ESP-IDF 4.4 CMD - "D:\ESP-IDF\Espressif\idf_cmd_init.bat" esp-idf-ab65b06dd9af1f5e42f388b601e9bc52
Writing at 0x00072d8e... (34 %)
Writing at 0x00098b7d... (38 %)
Writing at 0x000a28c1... (42 %)
Writing at 0x000a8111... (46 %)
Writing at 0x000adf3a... (50 %)
Writing at 0x000b3d13... (53 %)
Writing at 0x000b9990... (57 %)
Writing at 0x000c0d9c... (61 %)
Writing at 0x000c6345... (65 %)
Writing at 0x000cbcf9... (69 %)
Writing at 0x000d13da... (73 %)
Writing at 0x000d7285... (76 %)
Writing at 0x000dd8a3... (80 %)
Writing at 0x000e605f... (84 %)
Writing at 0x000ee710... (88 %)
Writing at 0x000f46b0... (92 %)
Writing at 0x000fa236... (96 %)
Writing at 0x00100051... (100 %)
Wrote 984224 bytes (410199 compressed) at 0x00010000 in 11.7 seconds (effective 673.4 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 105...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (105 compressed) at 0x00008000 in 0.1 seconds (effective 314.3 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
Done
```