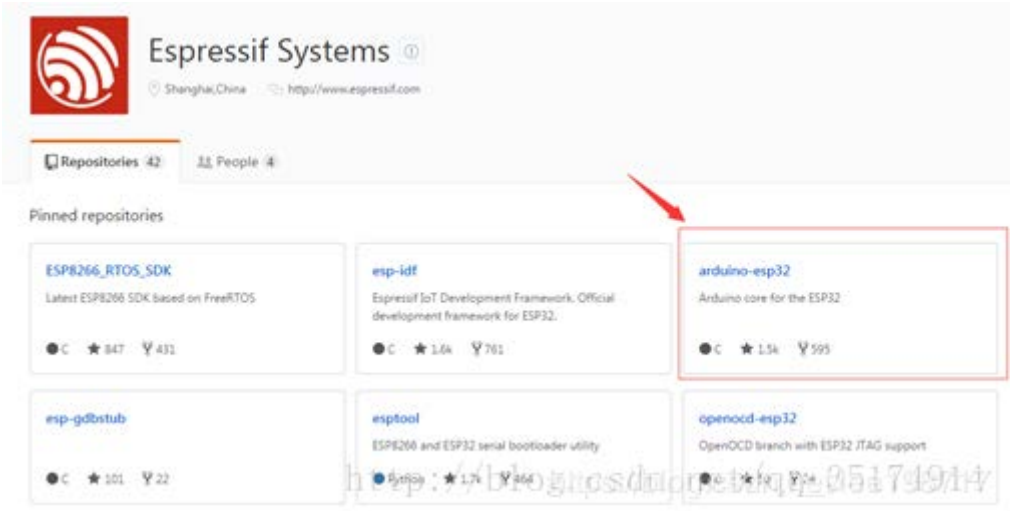
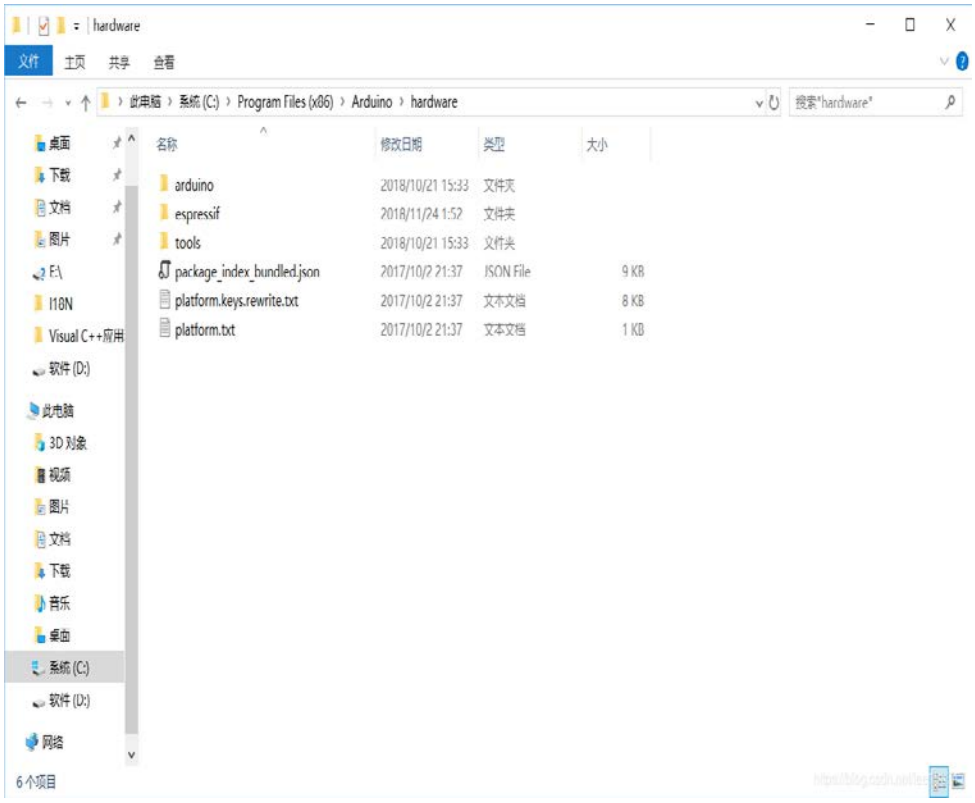
ESP32 Arduino 开发环境搭建

一、安装 arduino 的 ESP32 开发工具包

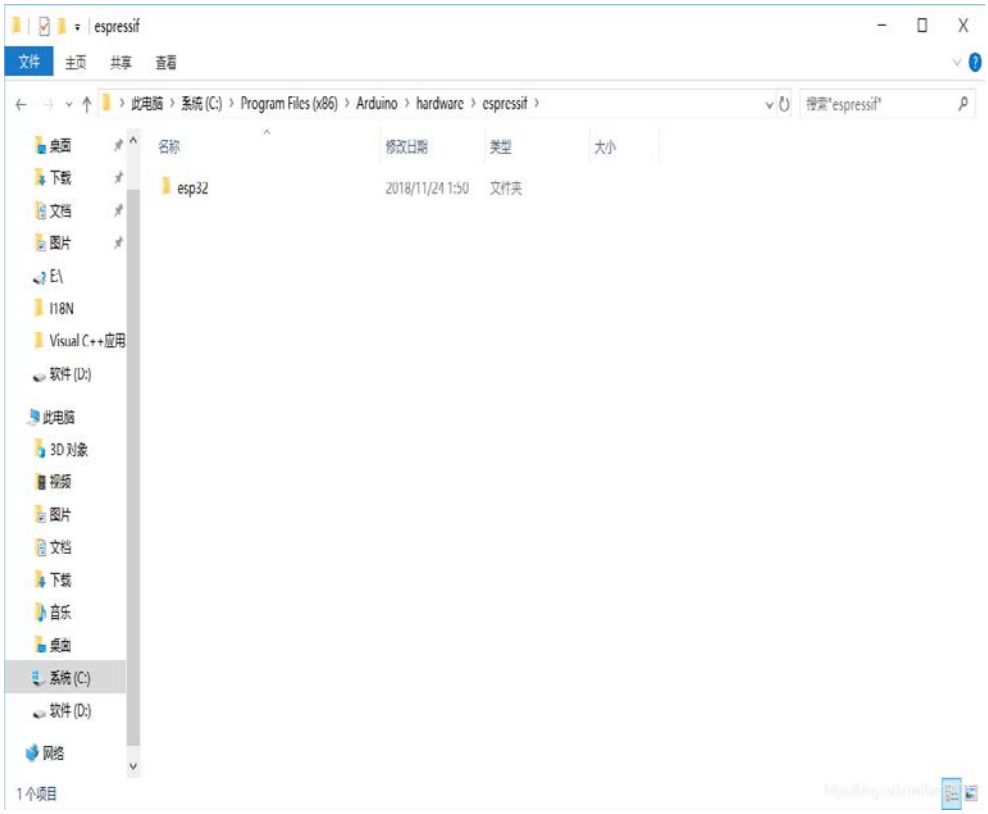
1. 进入 espressif 的仓库 https://github.com/espressif 选择 arduino-esp32，将所有文件打包下载。



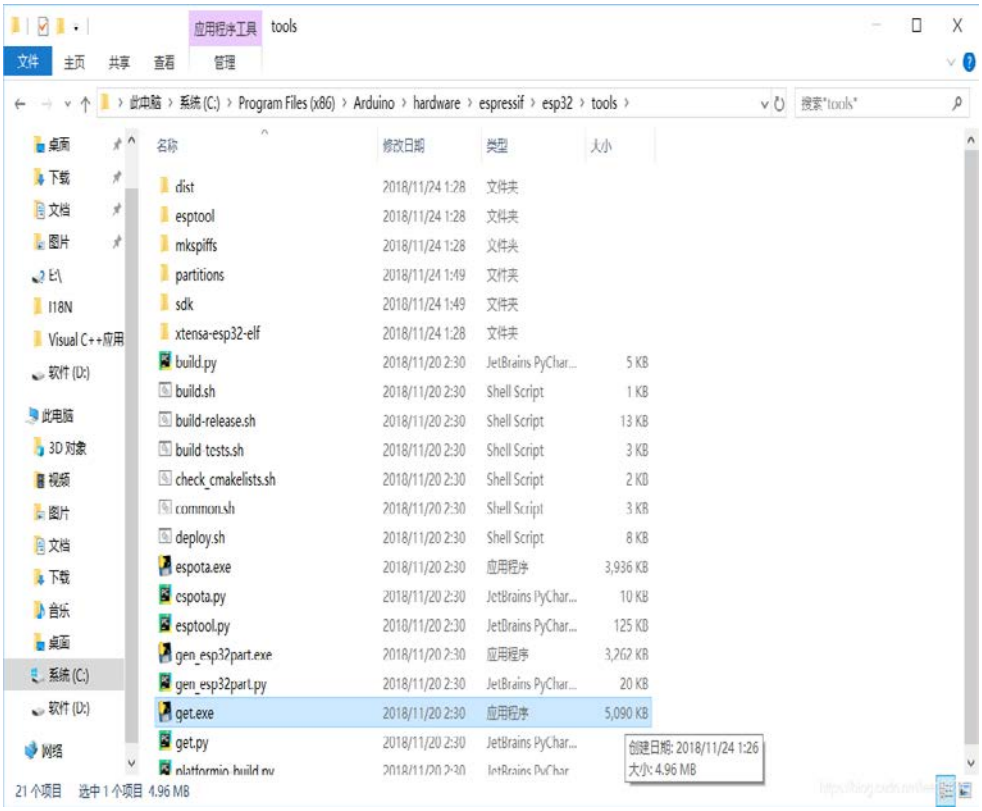
2. 找到 Arduino 软件的安装路径，进入 hardware 文件夹，新建一个名叫 espressif 的空白文件夹。



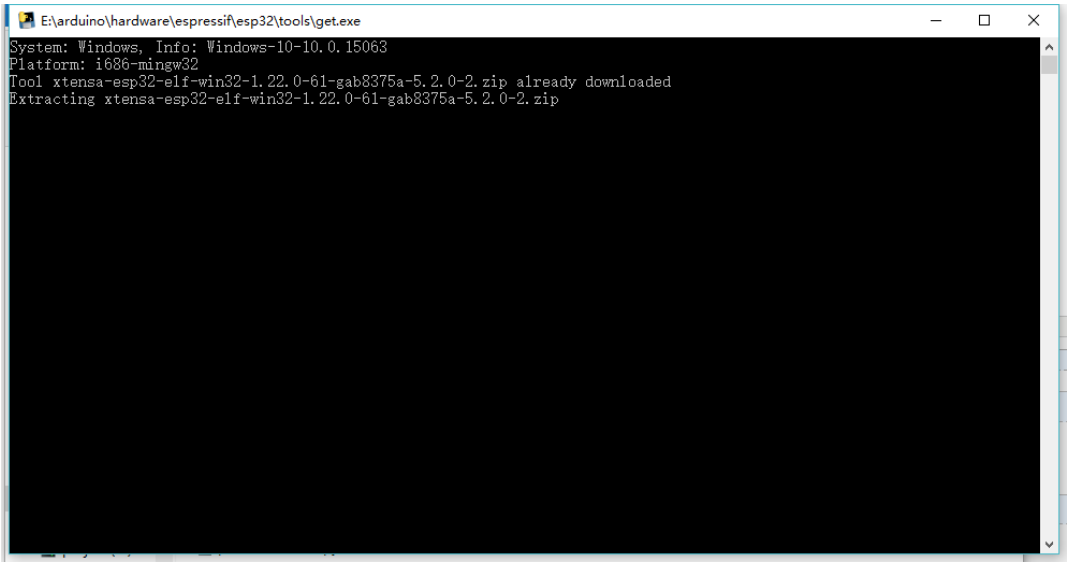
3.然后在 espressif 文件夹下新建一个名叫 esp32 的空白文件夹，将下载下来的压缩包解压，内容复制到 esp32 文件夹中。



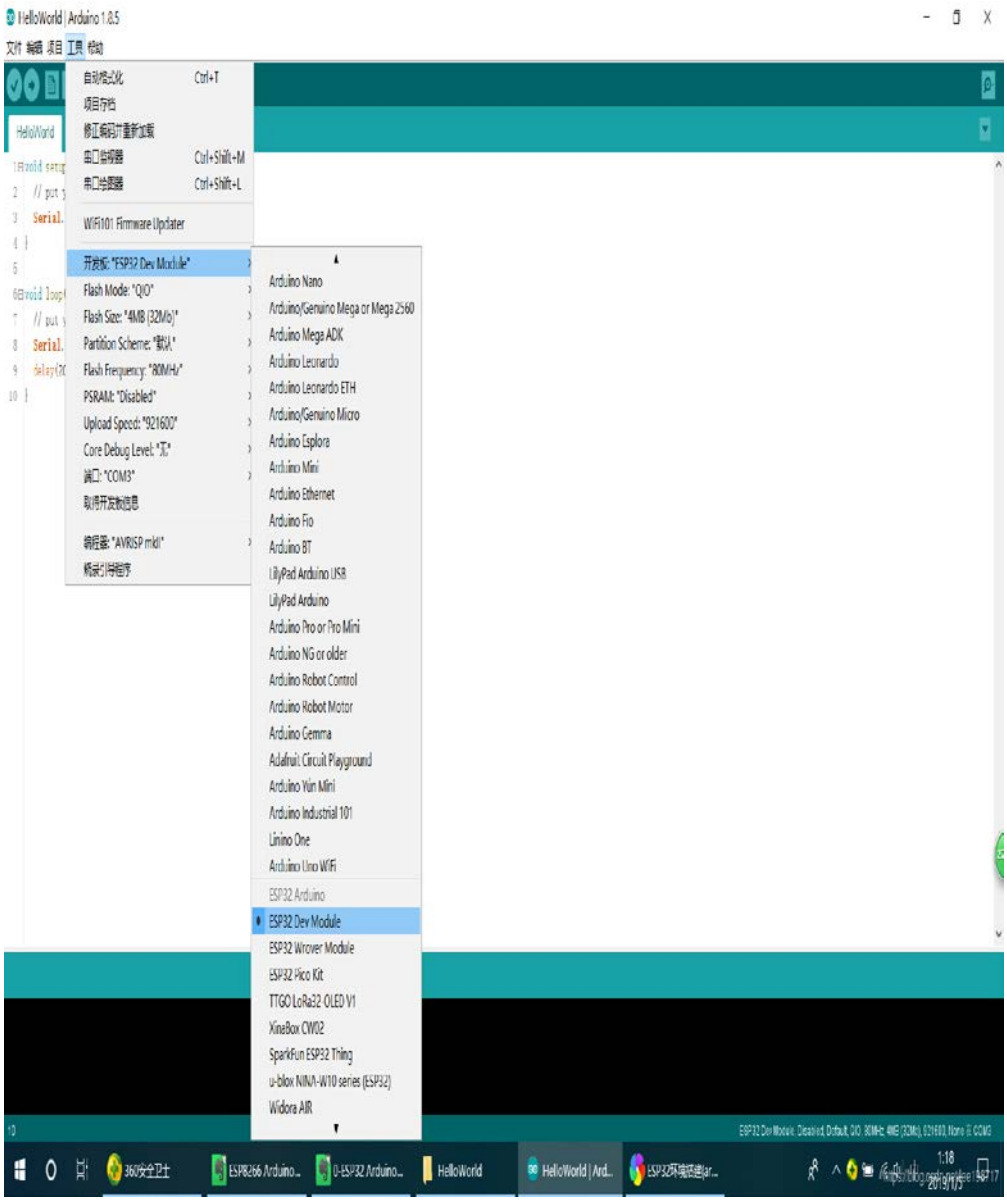
4. 进入 tools 文件夹，点击 get.exe 运行程序。（前提是你的电脑已经安装了 Python）



然后确保网络畅通，等待程序自动运行完毕，黑框自动关闭。



重启 Arduino 软件，打开工具，开发板，选择 ESP Dev Module 作为开发板表示安装成功。

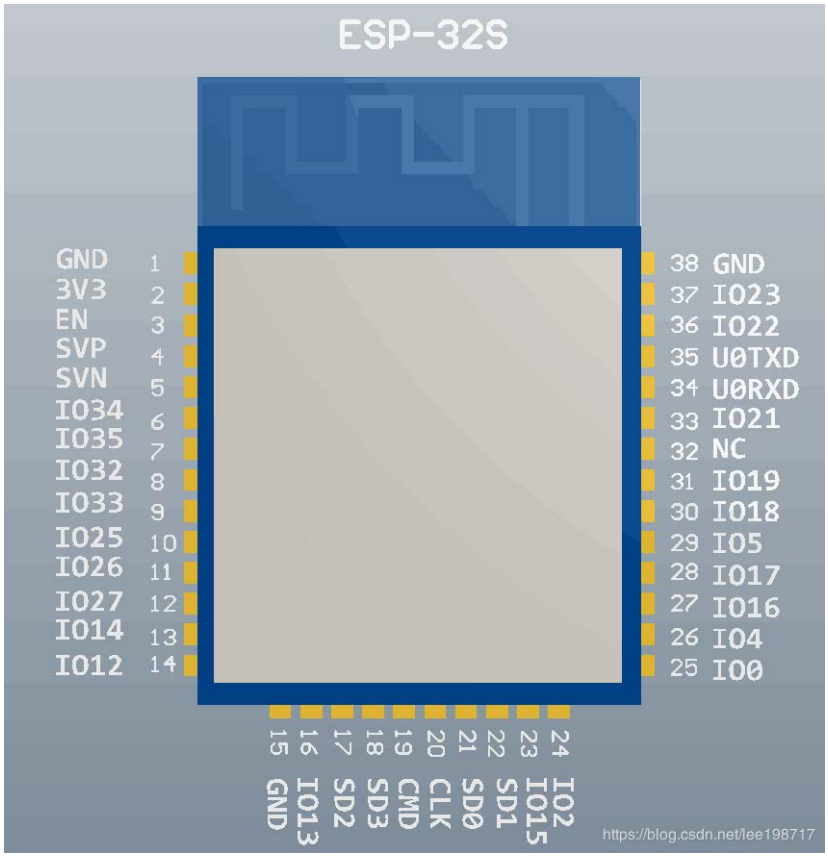


二、下载程序

点击 IDE 中的上传按钮，等到提示：”Connecting.....“时按住 BOOT 键，点击 EN 按钮并放开后进入下载模式，下载完成放开 BOOT 后芯片自动重启。

三、ESP32 模块资料

ESP-32S 管脚图：



ESP-32S 管脚功能定义：

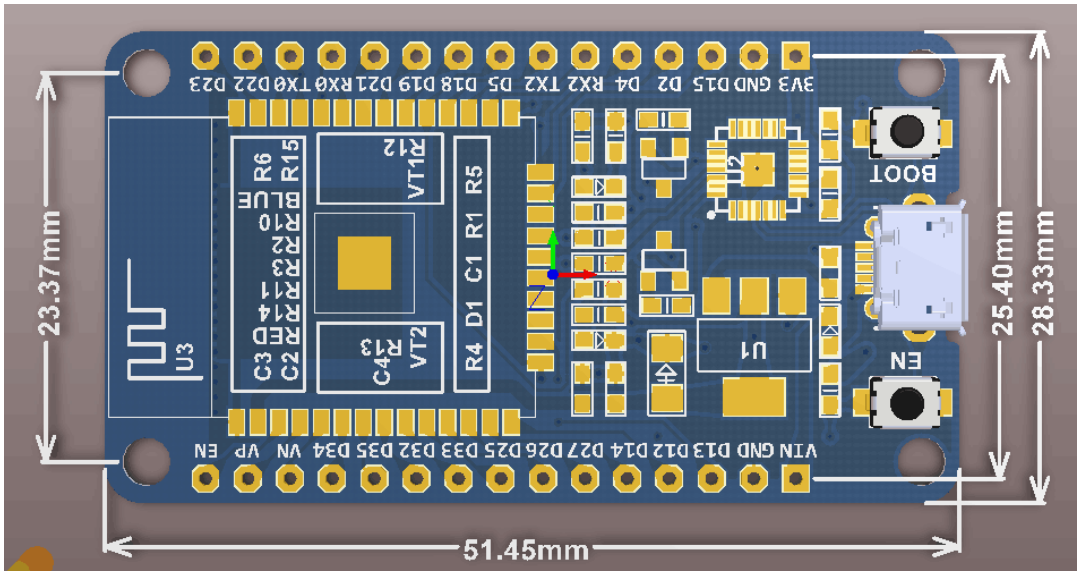
名称	序号	功能
GND	1	接地
3V3	2	供电
EN	3	使能芯片，高电平有效。
SENSOR_VP	4	GPI36, SENSOR_VP, ADC_H, ADC1_CH0, RTC_GPIO0
SENSOR_VN	5	GPI39, SENSOR_VN, ADC1_CH3, ADC_H, RTC_GPIO3
IO34	6	GPI34, ADC1_CH6, RTC_GPIO4
IO35	7	GPI35, ADC1_CH7, RTC_GPIO5
IO32	8	GPIO32, XTAL_32K_P (32.768 kHz crystal oscillator input), ADC1_CH4, TOUCH9, RTC_GPIO9
IO33	9	GPIO33, XTAL_32K_N (32.768 kHz crystal oscillator output), ADC1_CH5, TOUCH8, RTC_GPIO8
IO25	10	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
IO26	11	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
IO27	12	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV

IO14	13	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
IO12	14	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
GND	15	接地
IO13	16	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
SHD/SD2	17	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
SWP/SD3	18	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
SCS/CMD	19	GPIO11, SD_CMD, SPICS0, HS1_CMD, U1RTS
SCK/CLK	20	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS
SDO/SD0	21	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS
SDI/SD1	22	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
IO15	23	GPIO15, ADC2_CH3, TOUCH3, MTDO, HSPICS0, RTC_GPIO13, HS2_CMD, SD_CMD, EMAC_RXD3
IO2	24	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPIWP, HS2_DATA0, SD_DATA0
IO0	25	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
IO4	26	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPIHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
IO16	27	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
IO17	28	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
IO5	29	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
IO18	30	GPIO18, VSPICLK, HS1_DATA7
IO19	31	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
NC	32	-
IO21	33	GPIO21, VSPIHD, EMAC_TX_EN
RXD0	34	GPIO3, U0RXD, CLK_OUT2
TXD0	35	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
IO22	36	GPIO22, VSPIWP, U0RTS, EMAC_TXD1
IO23	37	GPIO23, VSPID, HS1_STROBE
GND	38	接地

ESP-32S 工作模式：

管脚	默认	SPI 启动模式	下载启动模式
GPIO0	上拉	1	0
GPIO2	下拉	无关项	0

四、ESP32 开发板资料

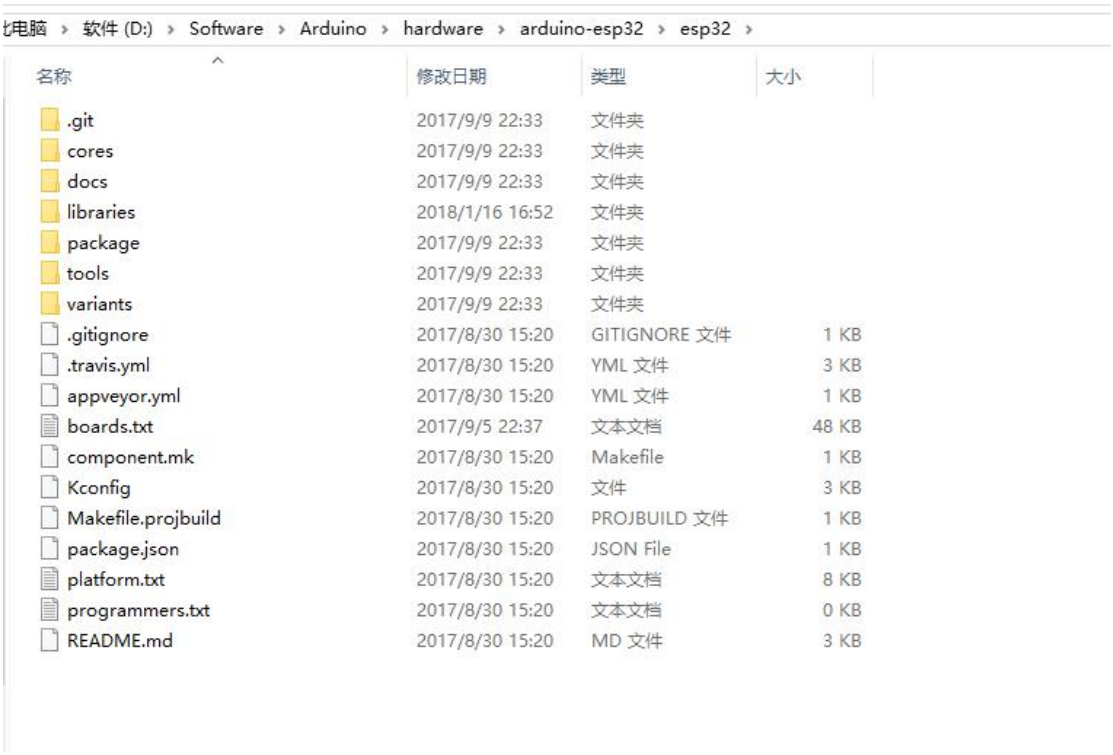


五、ESP32 开发板 Arduino 管脚定义

```
#ifndef Pins_Arduino_h
#define Pins_Arduino_h
#include <stdint.h>
#define EXTERNAL_NUM_INTERRUPTS 16
#define NUM_DIGITAL_PINS        40
#define NUM_ANALOG_INPUTS       16

#define analogInputToDigitalPin(p)  (((p)<20)?(esp32_adc2gpio[(p)]):-1)
#define digitalPinToInterrupt(p)    (((p)<40)?(p):-1)
#define digitalPinHasPWM(p)         (p < 34)

static const uint8_t LED_BUILTIN = 2;
#define BUILTIN_LED  LED_BUILTIN // backward compatibility

static const uint8_t TX = 1;
static const uint8_t RX = 3;

static const uint8_t SDA = 21;
static const uint8_t SCL = 22;
```

```
static const uint8_t SS    = 5;
static const uint8_t MOSI  = 23;
static const uint8_t MISO  = 19;
static const uint8_t SCK   = 18;

static const uint8_t A0 = 36;
static const uint8_t A3 = 39;
static const uint8_t A4 = 32;
static const uint8_t A5 = 33;
static const uint8_t A6 = 34;
static const uint8_t A7 = 35;
static const uint8_t A10 = 4;
static const uint8_t A11 = 0;
static const uint8_t A12 = 2;
static const uint8_t A13 = 15;
static const uint8_t A14 = 13;
static const uint8_t A15 = 12;
static const uint8_t A16 = 14;
static const uint8_t A17 = 27;
static const uint8_t A18 = 25;
static const uint8_t A19 = 26;

static const uint8_t T0 = 4;
static const uint8_t T1 = 0;
static const uint8_t T2 = 2;
static const uint8_t T3 = 15;
static const uint8_t T4 = 13;
static const uint8_t T5 = 12;
static const uint8_t T6 = 14;
static const uint8_t T7 = 27;
static const uint8_t T8 = 33;
static const uint8_t T9 = 32;

static const uint8_t DAC1 = 25;
static const uint8_t DAC2 = 26;

#endif /* Pins_Arduino_h */
```

六、Arduino ESP32 开发环境的文件结构



Core 文件夹：乐鑫提供的 ESP32 底层内核文件，包括 gpio timer iic spi touch uarts 等外设驱动，还有二次封装好的 TCP UDP Server 等常用 arduino 库文件。

libraries 文件夹：包含各厂家的 arduino 例程，我们的例程文件夹是 ESP-32F

variants 文件夹：各种 esp32 开发板的引脚定义头文件 （如果需要添加的自己开发板，则需要在该文件夹内加入自己开发板的引脚定义头文件，添加方法可以参考其他开发板的头文件，依样画葫芦即可）

Tools 文件夹：esp32 sdk 和编译工具链

Boards.txt：定义开发板的各项参数，这些参数将会在 arduino ide 选择开发板型号和参数的面板中体现出来。

```
##############################################################

esp-32f.name=ESP-32F                    //定义开发板名称

esp-32f.upload.tool=esptool             //定义开发板下载工具
esp-32f.upload.maximum_size=1310720                 //定义Flash空间
esp-32f.upload.maximum_data_size=294912             //定义data区空间
esp-32f.upload.wait_for_upload_port=true

esp-32f.serial.disableDTR=true
esp-32f.serial.disableRTS=true

esp-32f.build.mcu=esp32
esp-32f.build.core=esp32
esp-32f.build.variant=esp-32f
esp-32f.build.board=ESP-32F

esp-32f.build.f_cpu=240000000L    //定义cpu频率
esp-32f.build.flash_mode=dio      //定义flash类型
esp-32f.build.flash_size=4MB
esp-32f.build.boot=bootloader
esp-32f.build.partitions=default

esp-32f.menu.FlashFreq.80=80MHz                 //定义可选flash频率
esp-32f.menu.FlashFreq.80.build.flash_freq=80m
esp-32f.menu.FlashFreq.40=40MHz
esp-32f.menu.FlashFreq.40.build.flash_freq=40m

esp-32f.menu.UploadSpeed.921600=921600          //定义可选波特率
esp-32f.menu.UploadSpeed.921600.upload.speed=921600
esp-32f.menu.UploadSpeed.115200=115200
esp-32f.menu.UploadSpeed.115200.upload.speed=115200
esp-32f.menu.UploadSpeed.256000.windows=256000
esp-32f.menu.UploadSpeed.256000.upload.speed=256000
esp-32f.menu.UploadSpeed.230400.windows.upload.speed=256000
esp-32f.menu.UploadSpeed.230400=230400
esp-32f.menu.UploadSpeed.230400.upload.speed=230400
esp-32f.menu.UploadSpeed.460800.linux=460800
esp-32f.menu.UploadSpeed.460800.macosx=460800
esp-32f.menu.UploadSpeed.460800.upload.speed=460800
esp-32f.menu.UploadSpeed.512000.windows=512000
esp-32f.menu.UploadSpeed.512000.upload.speed=512000
```