



移植准备工作

软件准备:

- 1、Keil5(ARM)
- 2、涂鸦涂鸦 MCU SDK
- 3、STM32 Demo 程序

硬件准备:

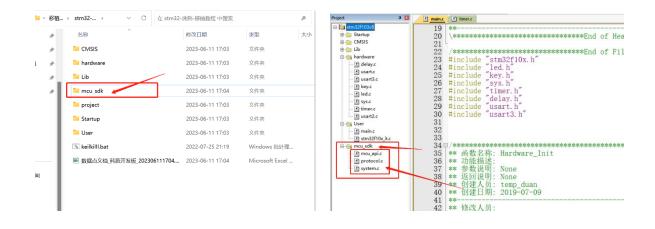
- 1、科派物联网开发板
- 2、数据线
- 3、ST-Link 下载器

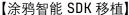
涂鸦 MCU SDK 移植

1、首先在涂鸦工作台中下载所需要的 MCU SDK, 如下图所示:



2、然后将里面所有的 .c 文件添加进工程







3、进入 wifi.h, 添加 #include "stm32f10x.h" eversion vz. o. o 2020. 12. 16 用户无需关心该文件实现F + Startup * @date 11 m (msis * @brief 12 E ib 13 */ hardware 14 # @ User □ ← mcu sdk 由于MCU类型和编译环境多种多样,所以 16 mcu api.c 涂鸦不对MCU功能结果负责。 wifi.h 18 *********** 19 string.h 20 □#ifndef __WIFI_H_ 21 | #define __WIFI_H_ protocol.h system.h 22 23 mcu_api.h #include "stm32f10x.h" protocol.c 24 system.c #include stdio.n #include string.h #include protocol.h #include system.h #include mcu_api.h stdio.h 25 26 27 28 29 30 31

3、在 protocol.c 中,找到 uart_transmit_output 函数 调用串口单字节发送函数。

/定义常量

//如果编译发生错误: #40: expected a

32

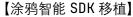
```
protocol.h usarts.n usarts.c* is system.c in protocol.c in witt.h. in main.c in timer.c
97 □ /**
                          * @brief 串口发送数据
* @param[in] {value} 串口要发送的1字节数据
* @return Null
 + Startup
 m (msis
                      99
 H 🗀 Lib
                     100
 hardware
                     101
 User
                         void uart_transmit_output(unsigned char value)
                      102
 mcu_sdk
                     103 ₽ {
 mcu_ap
                               #error "请将MCU串口发送函数填入该函数,并删除该行"
                     104
                             UART3_SendByte(value) 😹
                     105
   ⊕ ± sy
                     106
                     107
                             //Example:
                             extern void Uart_PutChar(unsigned char value);
Uart_PutChar(value);
                     108
                     109
                                                                               //串口发送函数
                     110
                     111
                     112
                     第二步:实现具体用户函数
                         1:APP下发数据处理
                     115
```

4、usart3.c中,在串口接收中断中调用 uart_receive_input(value),参数 value 是中断接收到的数据。记得先打开串口中断,程序运行过程中也尽量不要关闭所用到的串口中断。

```
protocol.h usart3.h usart3.c system.c protocol.c wifi.h main.c timer.c

☐ istm32f103c8

                          23 #include "usart3.h"
24 #include "wifi.h"
 E Startup
 E CMSIS
 ⊞ 🍅 Lib
 🖹 🦳 hardware
                                       **********************
   + delay.c
                          void USART3_IRQHandler(void)
                       39
   usart.c
                       40 ₽ {
  ⊕ 🖭 usart3.c
                       41
   H key.c
                              if(USART_GetITStatus(USART3, USART_IT_RXNE) != RESET)//接收到数据
                       42
   ⊕ ∄ led.c
                       43 🖹
   ⊕ ∰ sys.c
                                  res = USART_ReceiveData(USART3);
                       44
   timer.c
                       45
                                  uart_receive_input(res) 🛪
   usart2.c
 User |
                       47
 mcu_sdk
                          void UART3 SendByte (unsigned char data)
   mcu_api.c
                       48
                       49 ₽ {
   # protocol.c
   ⊕ 🔛 system.c
                            while((USART3->SR & USART_FLAG_TXE) != USART_FLAG_TXE);
                       51
                              USART3->DR = data;
                       52
```





5、mcu_api.c 中,进入函数 my_strcpy(char *dest, const char *src) ,将里面指针 p 的定义放到函数最前面。

```
protocol.h iii usart3.h iii usart3.c iii system.c iii protocol.c iii wifi.h iii main.c iii timer.c
∃ atm32f103c8
                                    * @param[in] {src} 源地址
* @return 数据处理完后的源地址
 ⊕ 🍵 Startup
                             131
 E CMSIS
                             132
                                    */
 E ib
                             133 char *my_strcpy(char *dest, const char *src)
 hardware
                             134 ₽ {
 ⊕ 🍅 User
                                        char *p = NULL;
if((NULL == dest) || (NULL == src)) {
                             135
 ⊨ mcu_sdk
                             136
  ⊕- 🔛 mcu_ap
                                             return NULL;
                             137
   ⊕ ∄ protocol.c
                             138
   system.c
                             139
                                        p = dest;
while(*src!='\0')
                             140
                             141
                                             *dest++ = *src++;
                             142
                             143
                                        *dest = '\0';
                             144
                             145
                                        return p:
                             146
                             147
                             148 = /**
                                           · c = + + + 11++
```

6、在主函数中初始化串口 3,波特率设置为 115200,添加 wifi_protocol_init()完成 wifi 协议初始化。

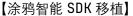
```
oject
              ф×
                   🔝 protocol.h 🔛 usart3.h 📩 usart3.c 🖆 system.c 🕍 protocol.c 📄 wifi.h 🕍 mai
stm32f103c8
+ Startup
                        E CMSIS
                      48 void Hardware Init(void)
± 🛅 Lib
                      49 □ {
i hardware
                      50
                             NVIC_PriorityGroupConfig(NVIC_PriorityGr
 🖹 🦳 User
                             Delay_Init();
                      51
 ⊞ 🕍 main.c 🗢
                      52
                             usart_init(115200);
  stm32f10x_it.c
                      53
                             usart3_init(115200);
 Ė ⊕ mcu_sdk
                             wifi_protocol_init();
                      54
  mcu_api.c
                      55
  ⊕ protocol.c
                      56
  ± system.c
```

7、主函数 while 循环中调用函数: wifi_uart_service()

```
Project
                 T 🔣
                        protocol.h usart3.h usart3.c system.c protocol.c wifi.h main.c*

☐ image: □ stm32f103c8

 69 *********************
 E CMSIS
                           70 int main(void)
 ⊕ 🍅 Lib
                           71 ₽ {
 ⊕ 🍵 hardware
                           72
 🖹 🦳 User
                                   Hardware_Init();
                           73
   ⊕ 🛣 main.c
                           74
   stm32f10x_it.c
                           75
                                   while(1)
 ⊟ <u>⊜</u> mcu_sdk
                           76
   mcu_api.c
                           77
                                     wifi uart service():
   protocol.c
                           78
                                     DelayMs(10);
   system.c
                           79
                           80 }
```





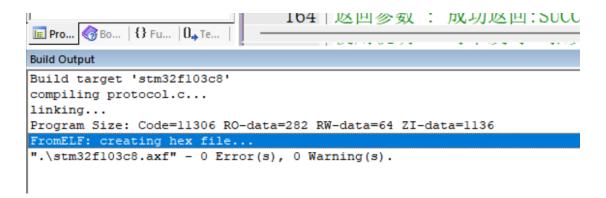
8、根据自己的需求处理自动化生成数据上报函数 all_data_update()

```
protocolh usarl3.h usarl3.c j system.c protocole wifth main.c itimer.c mcu_api.c mcu_
= € stm32f103c8
                                                                                                                                                                            136
           E CMSIS
                                                                                                                                                                                                              void all_data_update(void)
           138
                                                                                                                                                                                                                                               //#error "请在此处理可下发可上报数据及只上报数据示例,处理完成后删除该行"
                                                                                                                                                                              139
           User

main.c

stm32f10x_it.c
                                                                                                                                                                            140
                                                                                                                                                                                                                                        /*
//此代码为平台自动生成,请按照实际数据修改每个可下发可上报函数和只上报函数和只上报函数mcu_dp_bool_update(DPID_SWITCH_1, 当前LED1); //BOOL型数据上报;
mcu_dp_bool_update(DPID_SWITCH_2, 当前LED2); //BOOL型数据上报;
mcu_dp_bool_update(DPID_SWITCH_3, 当前LED3); //BOOL型数据上报;
mcu_dp_value_update(DPID_TEMP, 当前温度); //VALUE型数据上报;
mcu_dp_value_update(DPID_HUM, 当前湿度); //VALUE型数据上报;
mcu_dp_value_update(DPID_VOLTAGE, 当前电压); //VALUE型数据上报;
mcu_dp_value_update(DPID_CURRENT, 当前电流); //VALUE型数据上报;
                                                                                                                                                                            141
                                                                                                                                                                           142
143
              mcu_sdk
                                                                                                                                                                              144
             protocol.c
                                                                                                                                                                            145
                                                                                                                                                                            146
                                                                                                                                                                            147
                                                                                                                                                                               148
                                                                                                                                                                            149
                                                                                                                                                                               150
                                                                                                                                                                              151
```

9、其他报错直接注释掉引起报错的那行即可



10、按键配网

(1) 在 protocol.c,模块工作方式选择打开防误触模式

```
Protocol. DHT11.c It timer.c I main.c stm32f10c/wdg.h It iwdg.c oledfont.h system.h mcu_api.h It mcu_api.c It oled.c pro
 29
                   用户相关信息配置
 30
   *******************************
   31
                1:修改产品信息
 32
   *******************
 33
   #define PRODUCT_KEY "wthm8ygwenyr6o6m" //开发平台创建产品后生成的16位字符产品唯一标证
 34
 35
   #define MCU_VER "1.0.0"
                     //用户的软件版本,用于MCU固件升级,MCU升级版本需修改
 36
 37
   38
                                    //默认工作模式
//安全模式(低功耗配网方式)
//防误触模式(特殊配网方式)
 39
 40
 41 #define CONFIG_MODE
 42
   /* 设置低功耗配网方式和特殊配网方式的配网模式打开时间,该宏处于注释状态将按三分钟处理,可//#define CONFIG_MODE_DELAY_TIME 10 //配网模式打开时间 单位:分钟
 43
 44
 45
     46
   //#define CONFIG_MODE_CHOOSE 0
 47
   //#define CONFIG_MODE_CHOOSE
```



(2)通过一个按键实现配网操作

```
83 = 84
85
86 = 87
88
89
90
91 - 92
93 - 94
95 - 96
                   DelayMs(20);
if(KEY1 == RESET)
                        mcu_reset_wifi();
DelayMs(50);
mcu_set_wifi_mode(SMART_CONFIG);
printf("配网模式 SMART_CONFIG\r\n");
                   while(!KEY1);
              DelayMs(10);
```

- 11、手机 APP 绑定设备
 - (1) 点击又上角加号添加设备

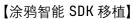


(2) 设备会发现新设备,点击添加



插座

电工





(3) 输入 WiFi 密码, 点击下一步。(WiFi 的频段必须是 2.4GHz)



(4)等待配网完成。

