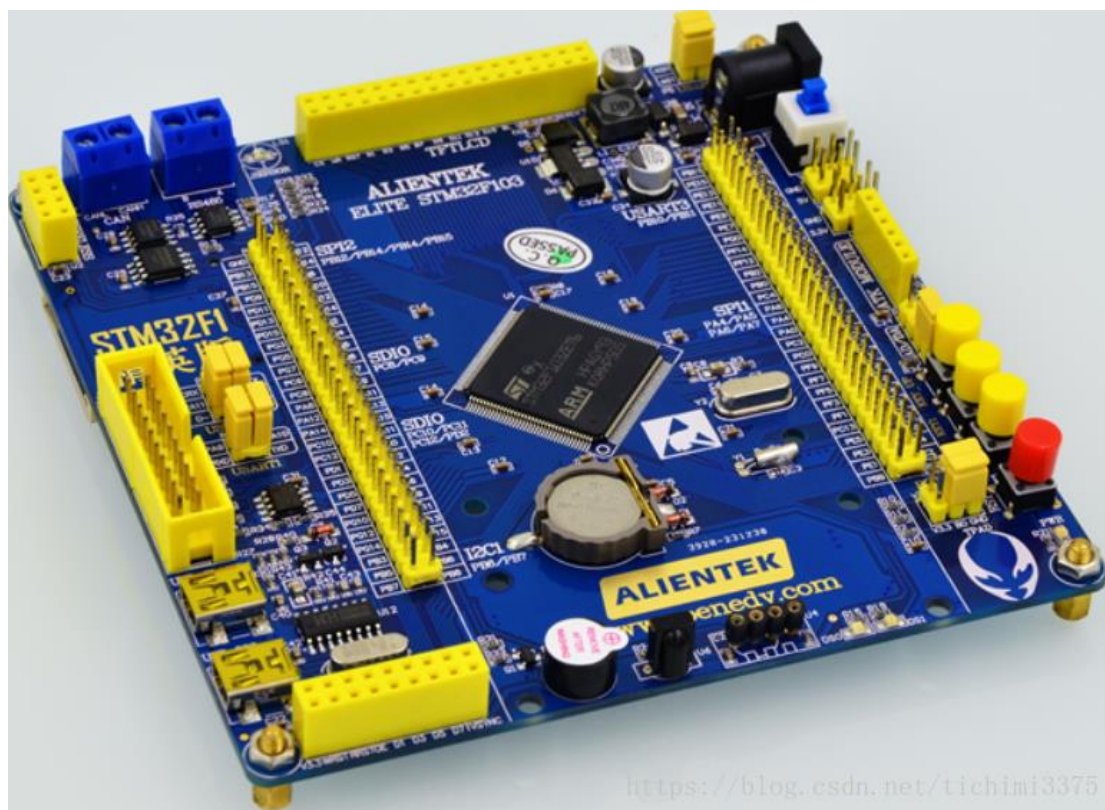


赵国承—STM32 接入机智云教程

硬件准备：

开发板使用正点原子精英板，大家使用其他板也可以，STM32 就行，即使是 STM32F103c8t6 也都可以：



WIFI 模块使用 ESP8266，我用的是正点原子的，大家用其他的 8266 也可以：

ESP8266串口WIFI模块

正点原子 串口控制



体积小 功耗低

提供STM32驱动资料

传感器我直接用精英板上带着的光敏传感器和两个 LED 灯吧。

第一步：ESP8266 烧写机智云的官方固件，这样 ESP8266 才能与机智云进行通信

固件可以去机智云官网下载，也可以直接用我提供的压缩包里的固件，我的固件也是从官网下载的，后面我以我的固件进行讲解

机智云
OzWits

硬件接入 文档中心 NEW 下载中心 社区

2459227990...

客户端开发资源

- 开源框架
- 设备接入SDK
- 统计分析SDK

开发与调试工具

- 机智云Wi-Fi移动通信产品调试APP
- 产测工具
- 机智云串口调试助手

开放源码

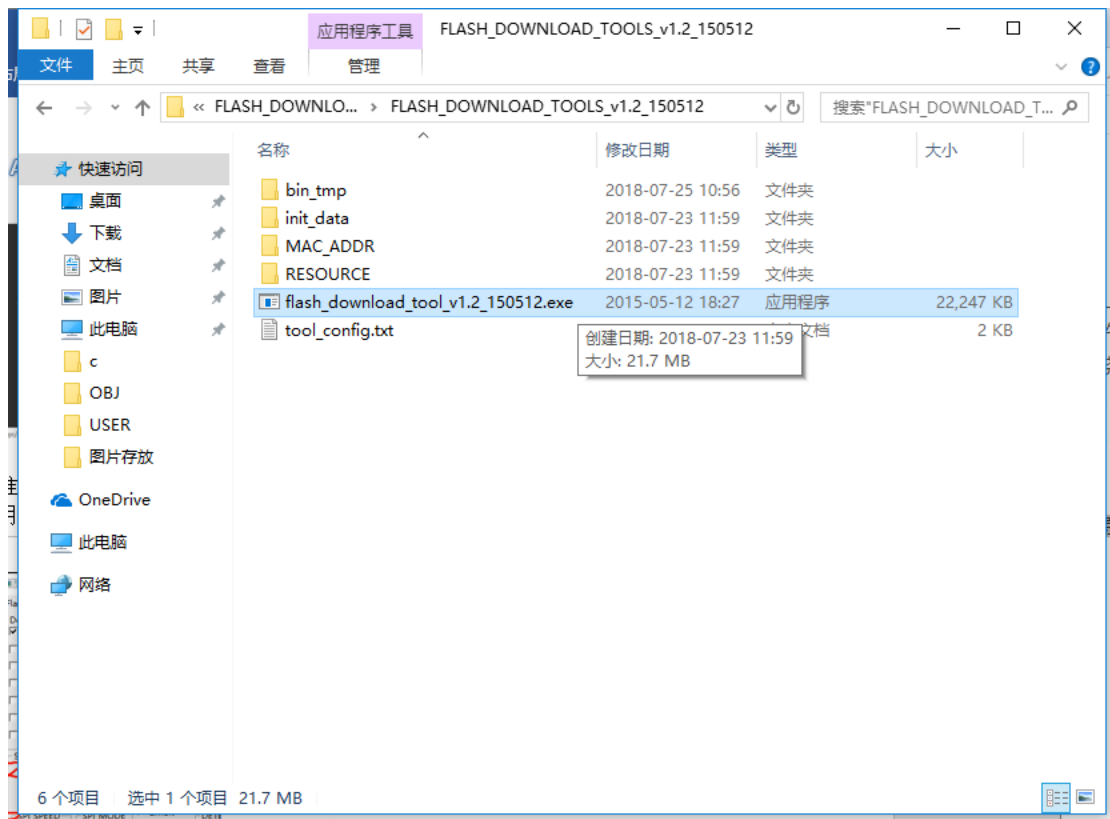
- 开源App
- 开源硬件
- 开源范例

产品开发者使用GAgent后，只需要关心产品的业务逻辑开发，不用关心数据的通讯功能开发，大大降低了开发的难度。

该固件遵循以下协议《机智云平台标准接入协议之MCU与WiFi模组通讯》、《机智云平台标准接入协议之设备与云端通讯》、《机智云平台标准接入协议之App与设备通讯》。

ECE鑫计算esp8266固件包 试用版	资源下载
发布时间：2017-07-20 03:50 更新信息 旧版本下载	
GAgent for ESP8266 04020034	资源下载
发布时间：2018-06-16 07:14 更新信息 旧版本下载	
GAgent for HFLPT120/LPB120/LPB125/LPT220 04020034	资源下载
发布时间：2018-06-16 07:18 更新信息 旧版本下载	
GAgent for HF LPB100 04020022	资源下载
发布时间：2017-09-01 10:14 更新信息 旧版本下载	
GAgent for MXCHIP 04020011	资源下载
发布时间：2016-03-04 06:11 更新信息 旧版本下载	

准备好 GAgent 固件后，那么就需要一个固件的下载工具。固件下载工具，我们用的是机智云官方推荐的乐鑫原厂固件烧写工具，在我提供的压缩包里也有

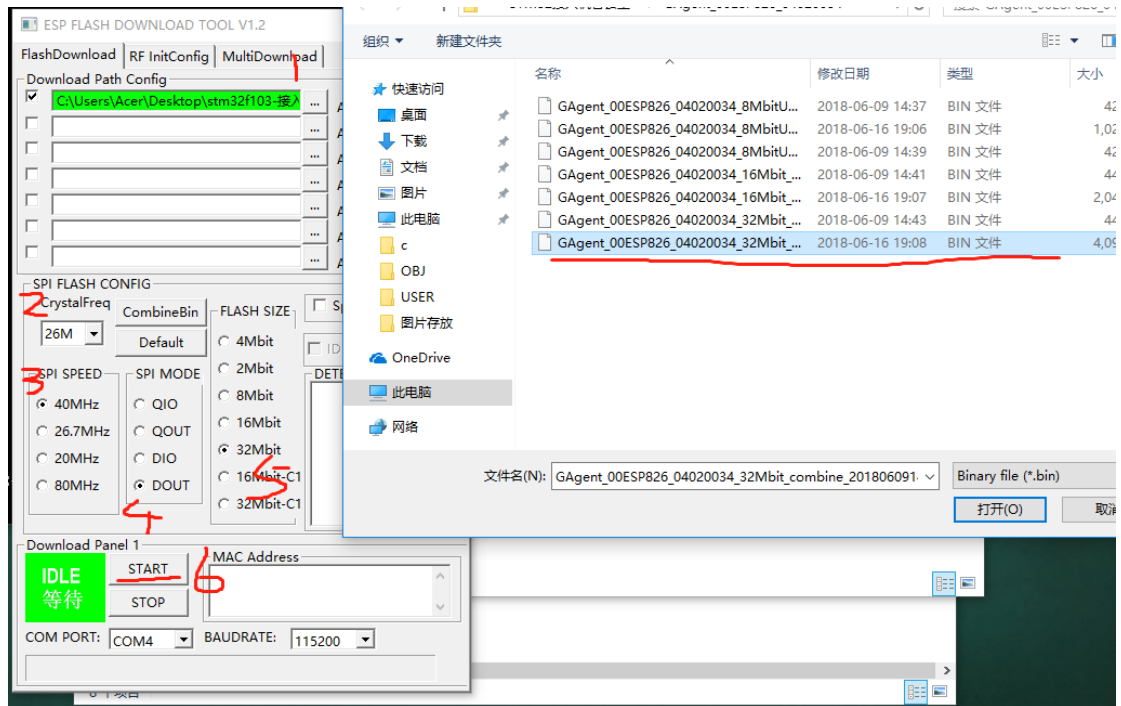


固件的下载需要其硬件的支持，我们可以用精英板上的 CH340 模块，就是这一部分



精英板	ATK-ESP8266模块
5V	VCC
GND	GND
TXD	RXD
RXD	TXD
GND	IO_0

我们也可以通过 USB to TTL 模块烧写



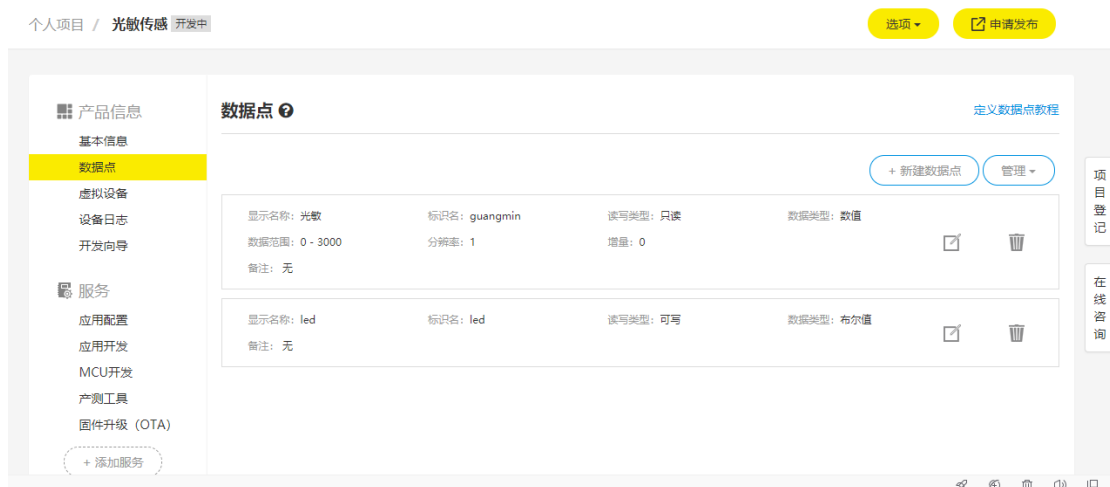
按下 START 键后 等待完成

硬件部分就完成了

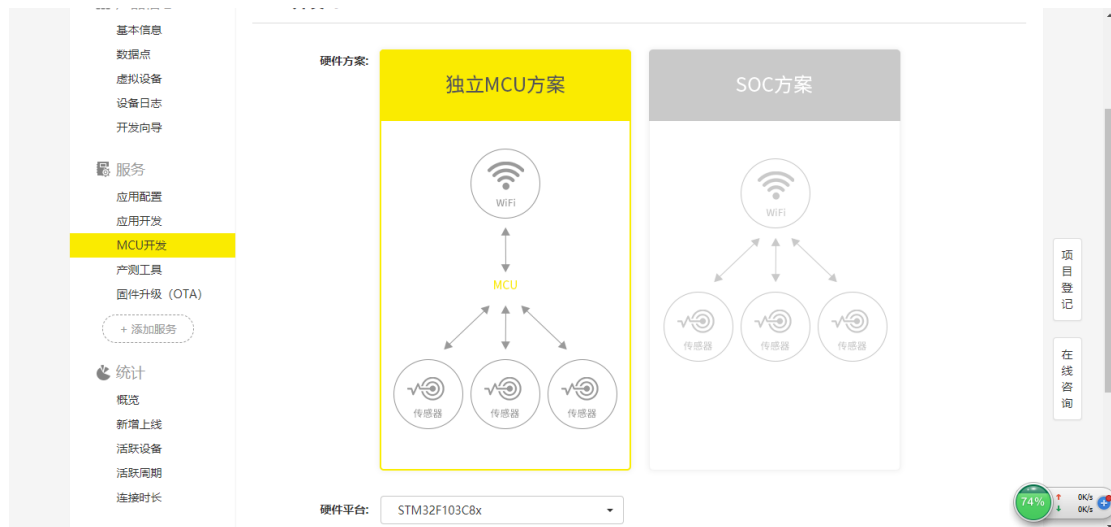
软件部分：

去机智云创建数据点：（具体怎样创建我在这里不做详细说明，不懂的朋友去机智云官网，有教程）

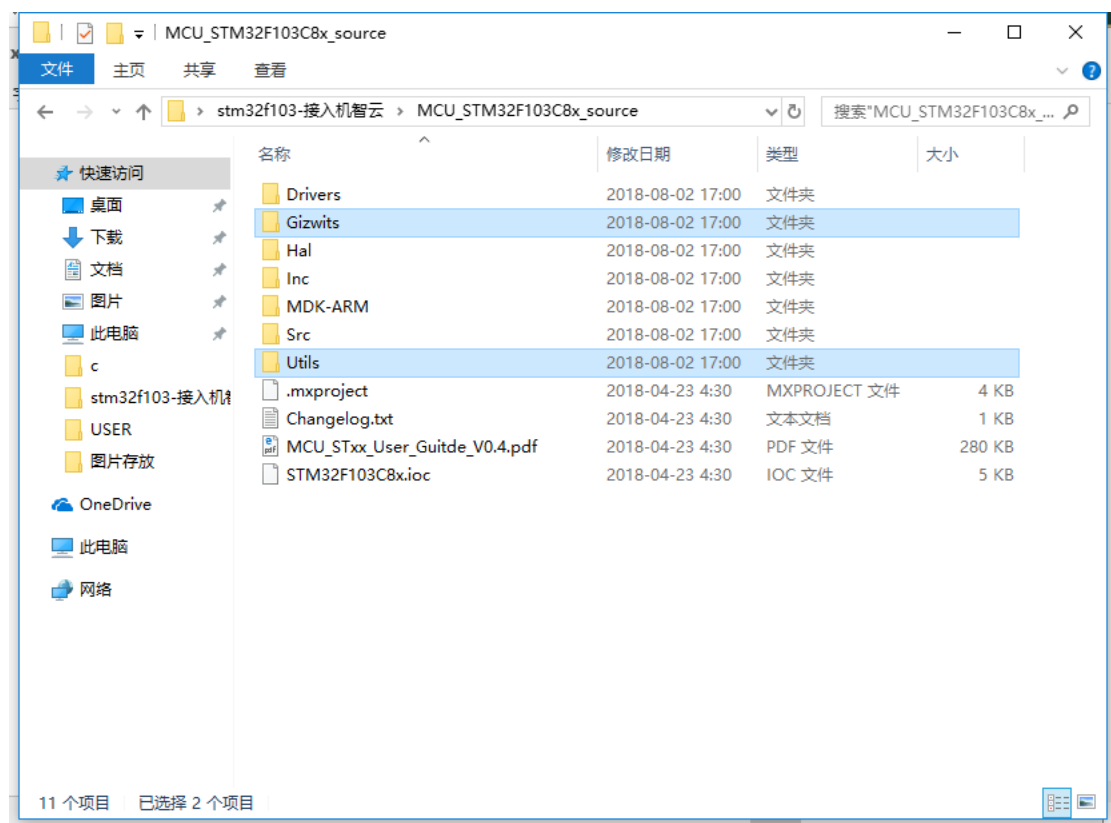
创建两个数据点：如图

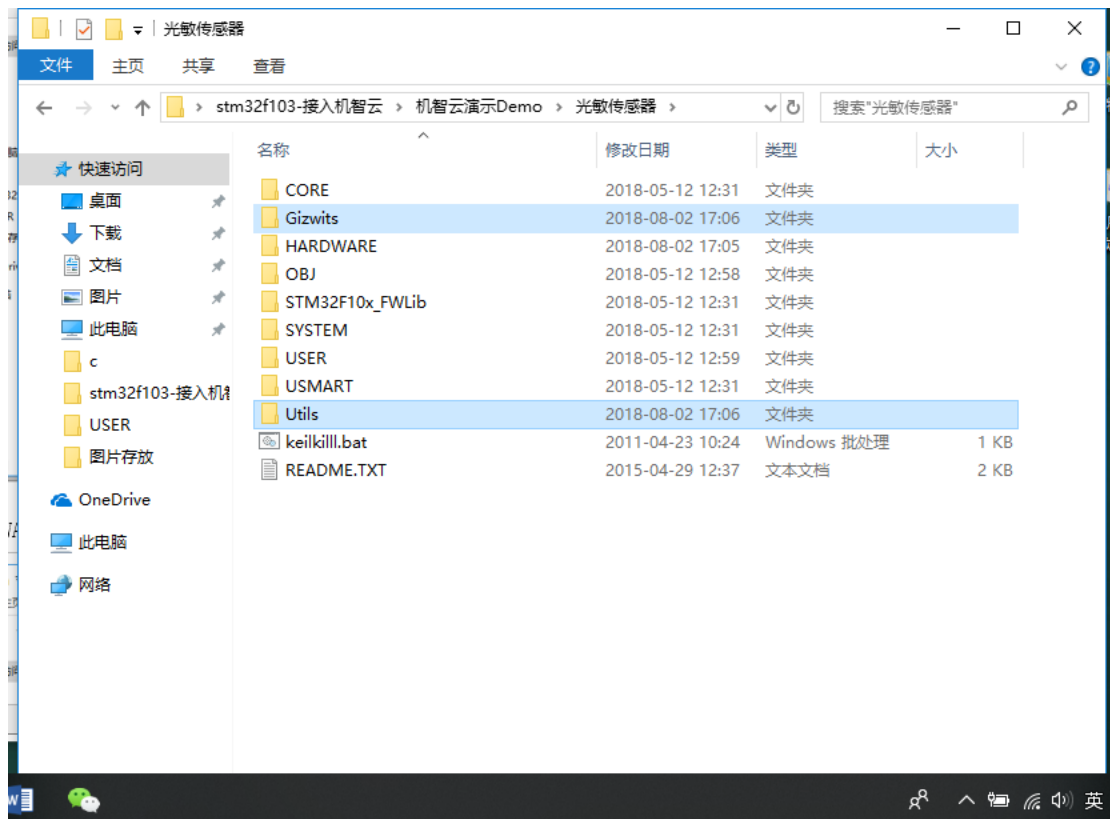


建好数据点后 下载机智云自动生成的 MCU 方案

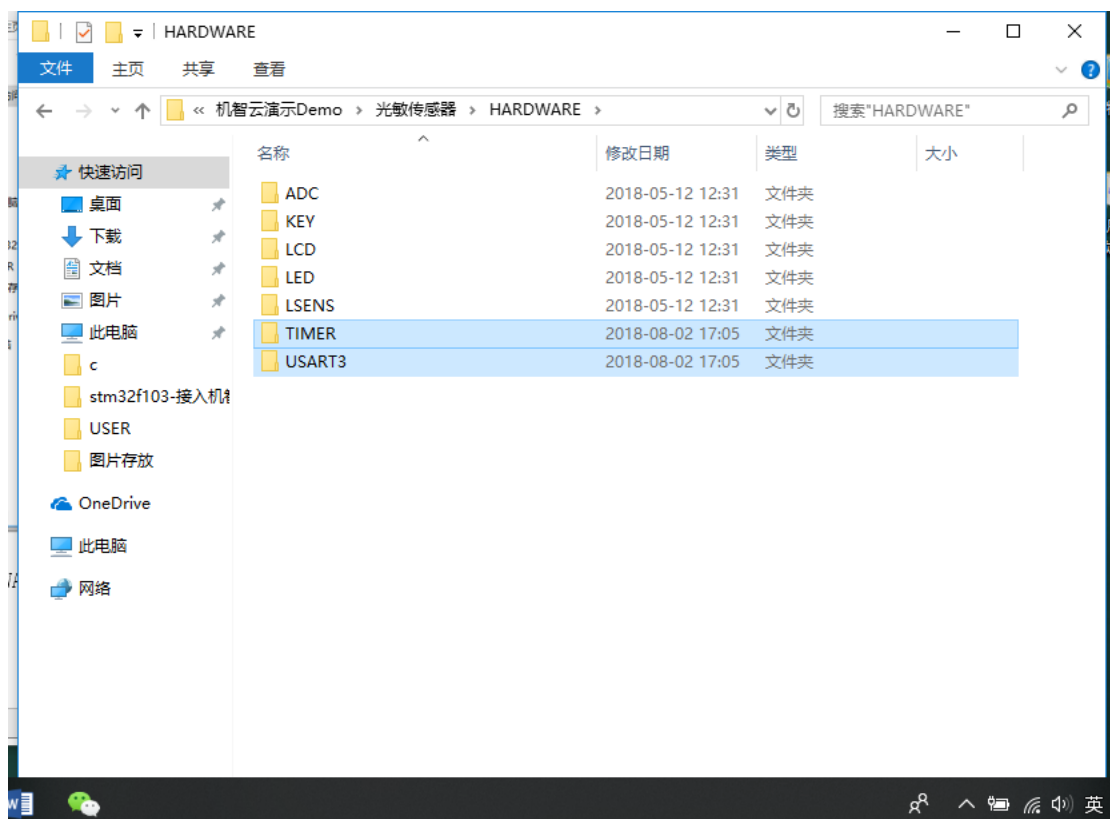


下载后解压，把下面的 Gizwits 和 Utils 文件夹复制到自己的工程里

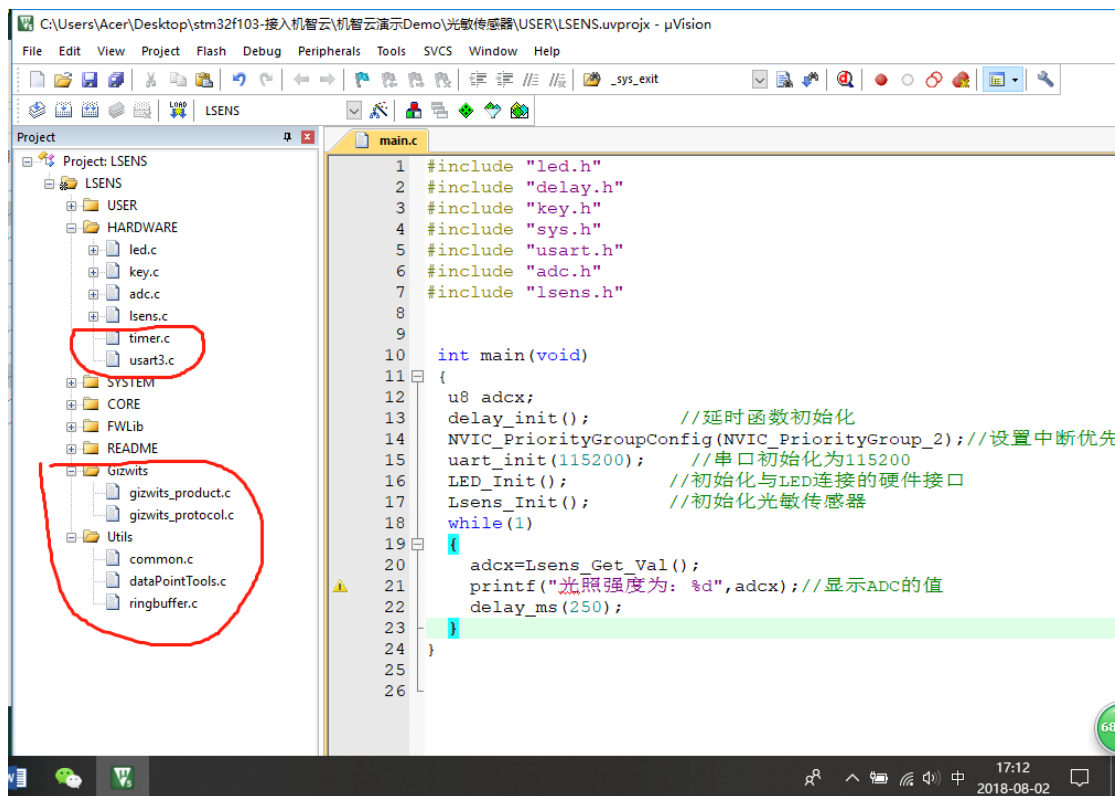




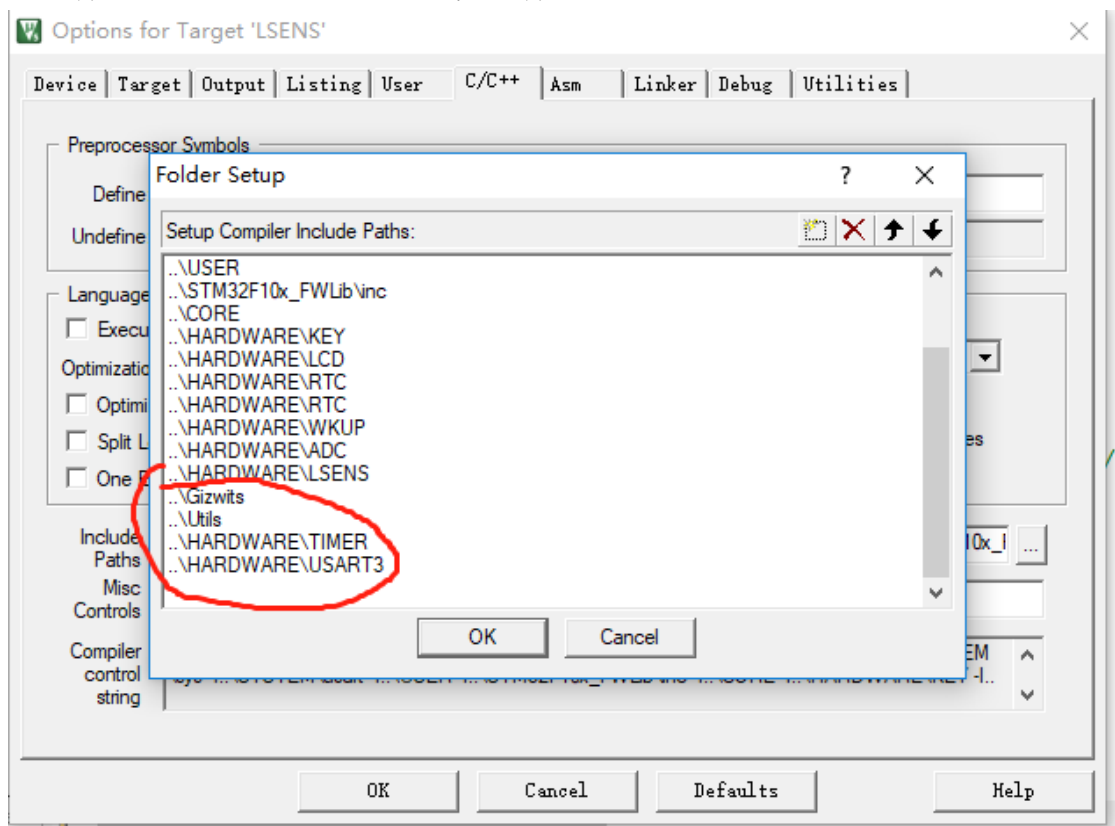
在 HARDWARE 文件夹下粘贴我给的压缩包里的 TIMER 和 USART3



把刚才添加的文件添加到工程里

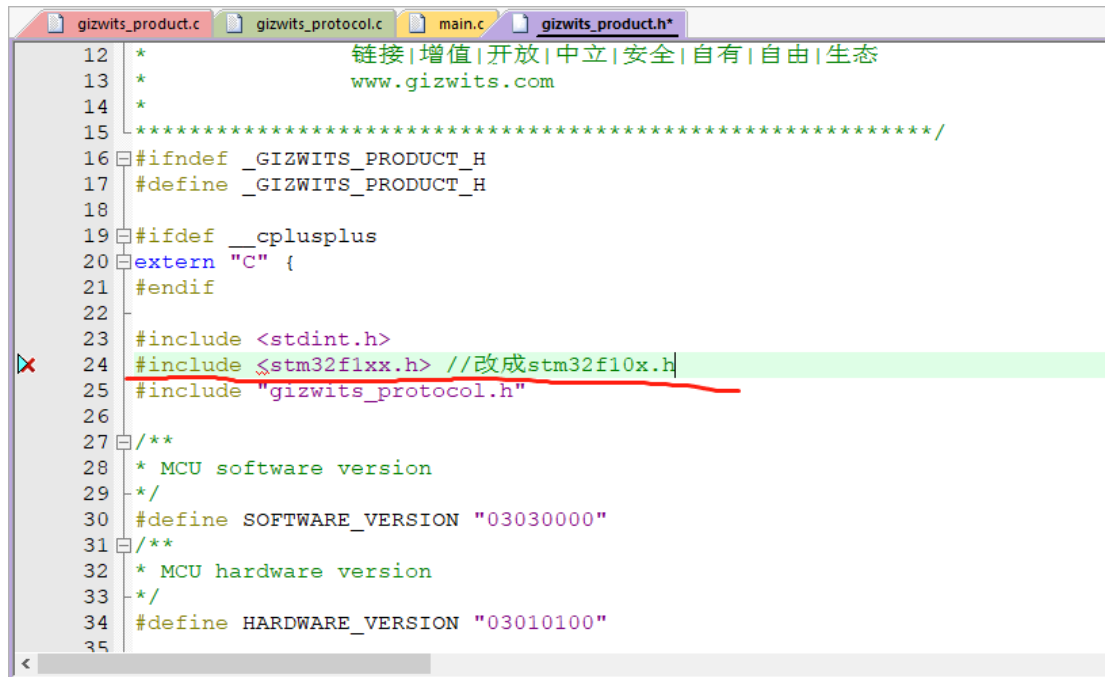


把文件加入工程后，别忘了添加头文件



编译后 有四个错误

打开 gizwits_product.h



```
12 * 链接|增值|开放|中立|安全|自有|自由|生态
13 * www.gizwits.com
14 *
15 *****/
16 #ifndef _GIZWITS_PRODUCT_H
17 #define _GIZWITS_PRODUCT_H
18
19 #ifdef __cplusplus
20 extern "C" {
21 #endif
22
23 #include <stdint.h>
24 #include <stm32f1xx.h> //改成stm32f10x.h
25 #include "gizwits_protocol.h"
26
27 /**
28  * MCU software version
29  */
30 #define SOFTWARE_VERSION "03030000"
31 /**
32  * MCU hardware version
33  */
34 #define HARDWARE_VERSION "03010100"
35
```

打开 gizwits_product.c



```
15 *****/
16
17 #include <stdio.h>
18 #include <string.h>
19 //删除 #include "hal_key.h"
20 #include "gizwits_product.h"
21 #include "common.h"
22 #include "usart3.h" //添加
23 #include "led.h" //添加
24
25 static uint32_t timerMsCount;
26 uint8_t aRxBuffer;
27
28 /** User area the current device state structure*/
29 extern dataPoint_t currentDataPoint;
30 extern keysTypeDef_t keys; //删除
31 extern u8 wifi_sta; //添加
32 extern TIM_HandleTypeDef htim2; //删除
33 extern UART_HandleTypeDef huart1; //删除
34 extern UART_HandleTypeDef huart2; //删除
35
36 /** @ */

```

```
51 for(i=0; i<info->num; i++)
52 {
53     switch(info->event[i])
54     {
55         case EVENT_led:
56             currentDataPoint.valueled = dataPointPtr->valueled;
57             GIZWITS_LOG("Evt: EVENT_led %d \n", currentDataPoint.valueled);
58             if(0x01 == currentDataPoint.valueled)
59             {
60                 //user handle
61                 LED0=0;
62             }
63             else
64             {
65                 //user handle
66                 LED0=1;
67             }
68             break;
69
70
71
72
174 uint32_t gizGetTimerCount(void)
175 {
176     return timerMsCount;
177 }
178
179 /**
180  * @brief MCU reset function
181  * @param none
182  * @return none
183  */
184 void mcuRestart(void)
185 {
186     __set_FAULTMASK(1);
187     HAL_NVIC_SystemReset(); // 删除, 添加 NVIC_SystemReset();
188 }
189
190 /**@} */
191
192
193 #ifdef __GNUC__
194 /* With GCC/RAISONANCE, small printf (option LD Linker->Libraries->Small
195    set to 'Yes') calls _io_putchar() */
```

下面是需要删除的 函数代码

```
void userHandle(void)
PUTCHAR_PROTOTYPE ( )
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
void timerInit(void)
void HAL_UART_RxCpltCallback(UART_HandleTypeDef*UartHandle)
void uartInit(void)
```

删除下面代码

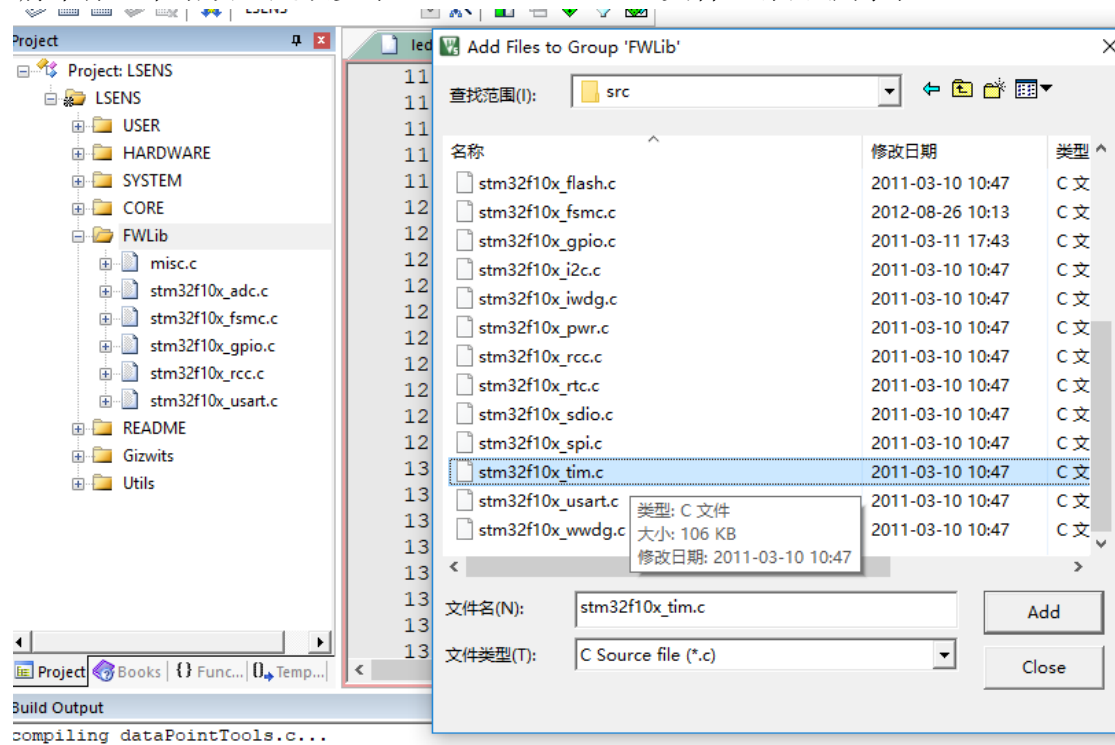
```
led.h gizwits_product.c* gizwits_protocol.c main.c gizwits_product.h
267 * @param buf : buf address
268 * @param len : buf length
269 *
270 * @return : Return effective data length;-1, return failure
271 */
272 int32_t uartWrite(uint8_t *buf, uint32_t len)
273 {
274     uint8_t crc[1] = {0x55};
275     uint32_t i = 0;
276
277     if(NULL == buf)
278     {
279         return -1;
280     }
281
282     for(i=0; i<len; i++)
283     {
284         HAL_UART_Transmit_IT(&huart2, (uint8_t *)&buf[i], 1);
285         while (huart2.gState != HAL_UART_STATE_READY); //Loop until the end of
286
287         if(i >=2 && buf[i] == 0xFF)
288         {
289             HAL_UART_Transmit_IT(&huart2, (uint8_t *)&crc, 1);
290             while (huart2.gState != HAL_UART_STATE_READY); //Loop until the e
291         }
292     }
293 }
```

在添加以下代码

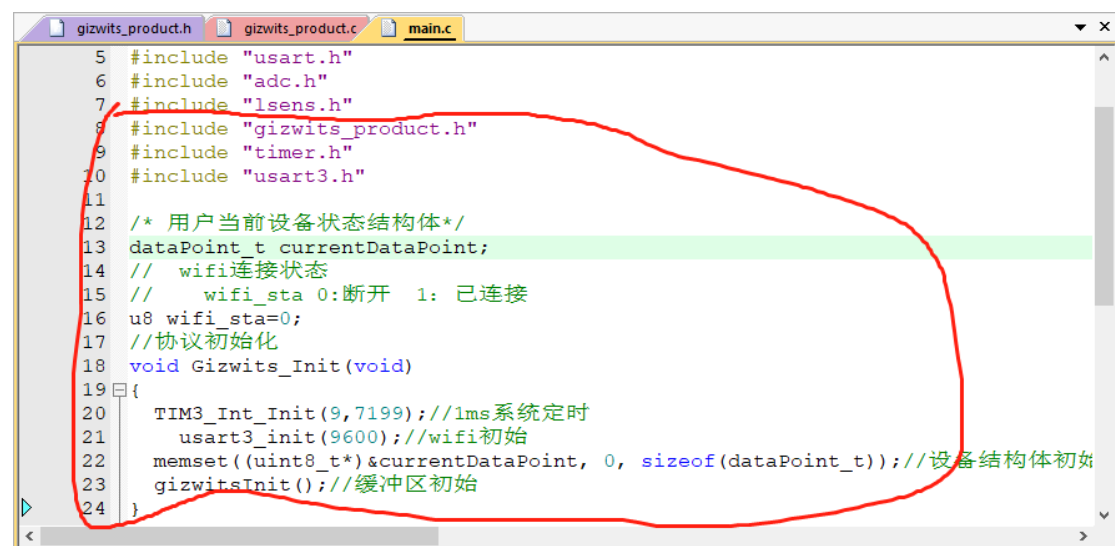
```
USART_SendData(USART3, buf[i]);
while(USART_GetFlagStatus(USART3, USART_FLAG_TC)==RESET); //循环
环发送,直到发送完毕
if(i >=2 && buf[i] == 0xFF)
{
    USART_SendData(USART3, 0x55);
    while(USART_GetFlagStatus(USART3, USART_FLAG_TC)==RESET); //
循环发送,直到发送完毕
}
```

```
led.h gizwits_product.c* gizwits_protocol.c main.c gizwits_product.h
273 {
274     uint8_t crc[1] = {0x55};
275     uint32_t i = 0;
276
277     if(NULL == buf)
278     {
279         return -1;
280     }
281
282     for(i=0; i<len; i++)
283     {
284         USART_SendData(USART3, buf[i]);
285         while(USART_GetFlagStatus(USART3, USART_FLAG_TC)==RESET); //循环发送
286         if(i >=2 && buf[i] == 0xFF)
287         {
288             USART_SendData(USART3, 0x55);
289             while(USART_GetFlagStatus(USART3, USART_FLAG_TC)==RESET); //循环发送
290         }
291     }
292
293 #ifdef PROTOCOL_DEBUG
294     GIZWITS_LOG("MCU2WiFi[%4d:%4d]: ", gizGetTimerCount(), len);
295     for(i=0; i<len; i++)
296     {
297     }
```

编译有 6 个错误，是缺少了 stm32f10x.tim.c 文件，添加就好了



这样就修改完 gizwits_product 文件，下面修改 main



```
gizwits_product.h  gizwits_product.c  main.c
22  memset((uint8_t*)&currentDataPoint, 0, sizeof(dataPoint_t)); //设备结构体初始
23  gizwitsInit(); //缓冲区初始
24  }
25  void userHandle(void)
26  {
27      u8 adcx;
28      adcx=Lsens_Get_Val();
29      //printf("光照强度为: %d",adcx); //显示adc
30      currentDataPoint.valueguangmin = adcx ; //Add Sensor Data Collection
31
32      //判断当前LED1开关量
33      if(LED0==0)
34          currentDataPoint.valueled = 1;
35      else
36          currentDataPoint.valueled = 0;
37  }
38
39
40  int main(void)
41  {
```

```
key.h  key.c  led.h  led.c  gizwits_product.h  gizwits_product.c  main.c
38
39
40  int main(void)
41  {
42      int key;
43      u8 wifi_con=0; //记录wifi连接状态 1: 连接0: 断开
44      delay_init(); //延时函数初始化
45      NVIC_PriorityGroupConfig(NVIC_PriorityGroup_2); //设置中断优先级分组为组2: 2位抢
46      uart_init(115200); //串口初始化为115200
47      LED_Init(); //初始化与LED连接的硬件接口
48      Lsens_Init(); //初始化光敏传感器
49
50      KEY_Init(); //按键初始化
51      Gizwits_Init();
52      printf("-----机智云IOT-光敏传感器-----\r\n");
53      printf("KEY1:AirLink连接模式\t KEY_UP:复位\r\n\r\n");
54      while(1)
55      {
56          if(wifi_con!=wifi_sta)
57          {
58              wifi_con=wifi_sta;
59              wifi_con?printf("connect"):printf("close");
60          }
61
62
63      userHandle(); //用户时间过程
64      gizwitsHandle((dataPoint_t *)&currentDataPoint); //协议处理
65  }
```

```
56     if(wifi_con!=wifi_sta)
57     {
58         wifi_con=wifi_sta;
59         wifi_con?printf("connect"):printf("close");
60     }
61
62
63     userHandle();//用户时间过程
64     gizwitsHandle((dataPoint_t *)&currentDataPoint);//协议处理
65
66     key = KEY_Scan(0);
67     if(key==KEY1_PRES)//KEY1按键
68     {
69         printf("WIFI进入AirLink连接模式\r\n");
70         gizwitsSetMode(WIFI_AIRLINK_MODE);//Air-link模式接入
71     }
72     if(key==WKUP_PRES)//KEY_UP按键
73     {
74         printf("WIFI复位, 请重新配置连接\r\n");
75         gizwitsSetMode(WIFI_RESET_MODE);//WIFI复位
76         wifi_sta=0;//标志wifi已断开
77     }
78     delay_ms(250);
79 }
80 }
81
82
```

修改完编译，0 错误 1 警告

```
compiling common.c...
compiling dataPointTools.c...
compiling ringbuffer.c...
linking...
Program Size: Code=25532 RO-data=716 RW-data=98 ZI-data=2160
FromELF: creating hex file...
"..\\OBJ\\LENS.axf" - 0 Error(s), 1 Warning(s).
```

这样就移植完了。

编译烧写

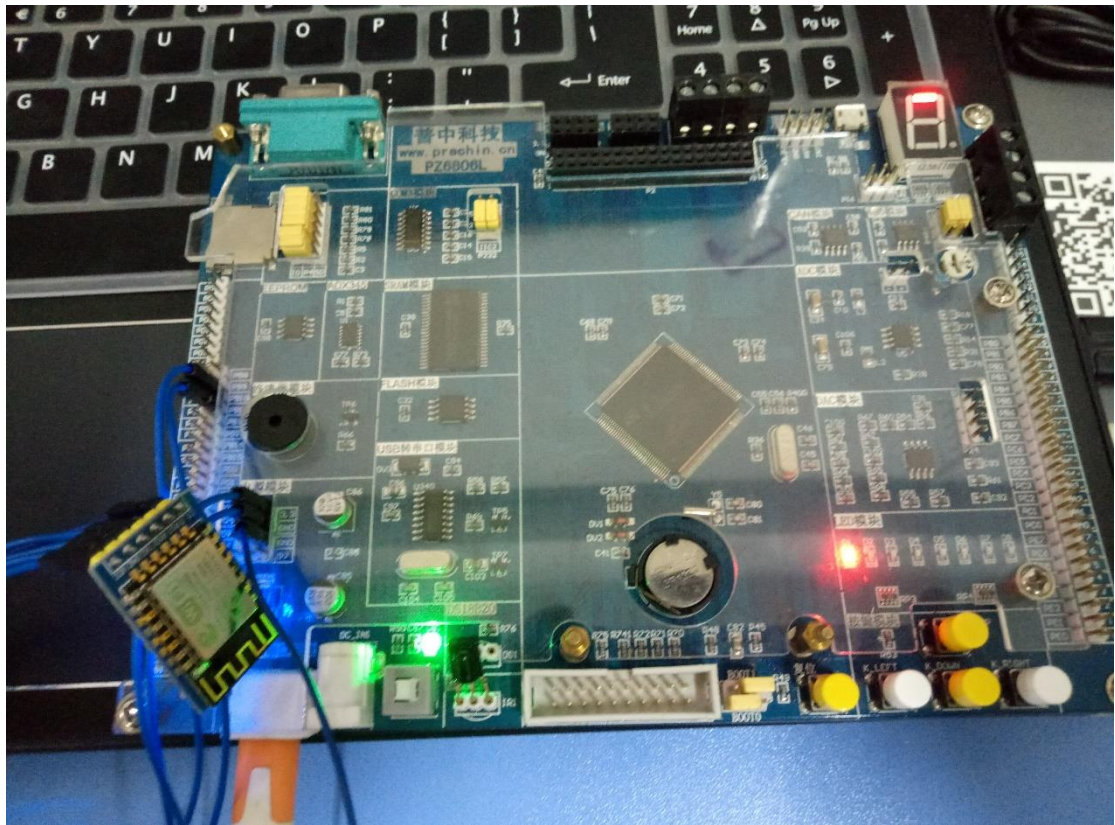
让 wifi 与 stm32 连接好，然后 按 k_up 按键 重置 wifi 配置

添加配置，搜索设备，完成后如下图





下图我用的是普中的 103 开发板调试，原理一样，就是 led.c 和 led.h，io 口改一下，如果是正点战舰或精英，直接使用 不用修改



最后就完成了，祝你好运!!!