

# Thermostat HMI with Tuya IoT Template

# Agenda

---

- Introduction
- Integrate Tuya IoT
- Integrate Thermostat HMI
- Q&A

# Introduction

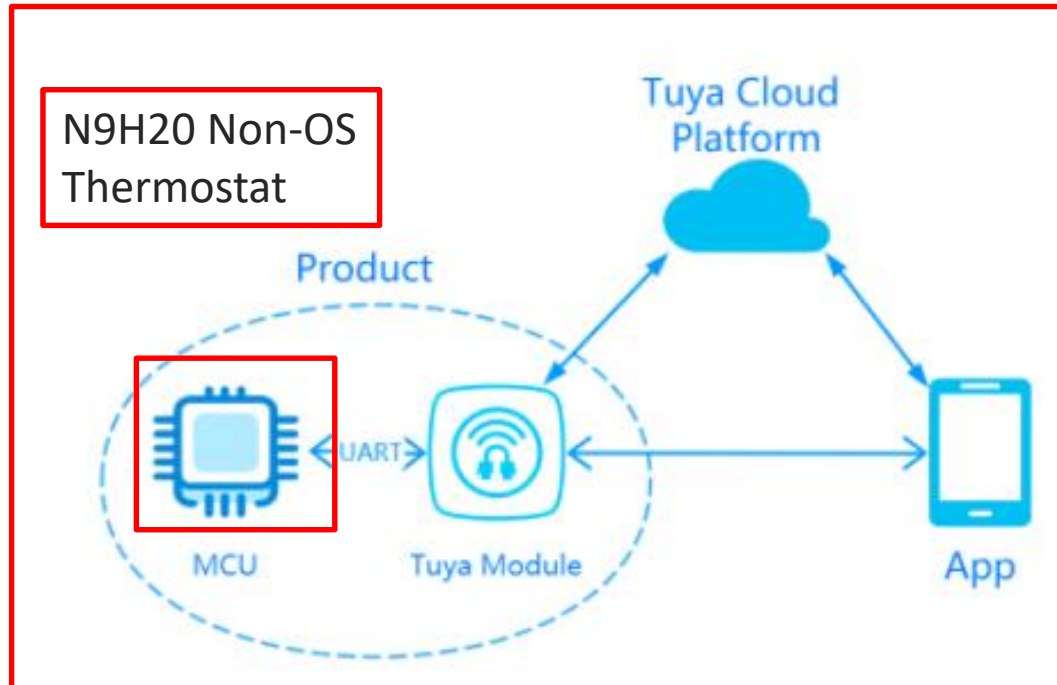
## • 一站式物联网解决方案

- <https://developer.tuya.com/cn/docs/iot/guidelines-for-platform/introduction-of-tuya-iot-platform/introduction-of-tuya-iot-platform?id=K960olyp3qas9>



## CONT.

- Utilize Tuya IoT Wi-Fi feature to apply N9H20 Non-OS Thermostat to cloud feature



# Integrate Tuya IoT

- Start to develop

The screenshot displays the Tuya IoT platform interface. On the left, a sidebar menu has '产品' (Product) highlighted, with a sub-menu '产品开发' (Product Development) also highlighted. The main area shows the 'CS Thermostat' product page. A red box highlights the '自定义开发' (Custom Development) button, and another red box highlights the '硬件开发' (Hardware Development) button. A red arrow points from the '硬件开发' button to a separate window on the right. This window shows the '发布产品' (Publish Product) button, which is also highlighted with a red box. A red arrow points from the '发布产品' button back to the '自定义开发' button in the main window.

You can find your PID here

Also please fill to

\N9H20\_emWin\_NonOS-master

\BSP\Driver\Source\TUYA\_IoT\protocol.h

Please fill your PID to protocol.h and build the TUYA\_IoT library for Thermostat HMI

# CONT.

- Utilize on-line official document

## 产品开发引导

初次使用平台，建议按照以下流程完成智能产品的开发

### ① 功能定义

具体智能设备功能的抽象，用于描述产品功能及参数 [查看更多](#)

### ② 硬件开发

开发智能硬件中的软件功能，开发者在完成功能定义后，即可开始进行嵌入式程序的开发与调试 [查看更多](#)

文档中心 > 产品智能化 > 硬件开发 > 选择和管理固件版本

## 选择和管理固件版本

### 相关文档

- 选品类创建产品
- MCU SDK 移植
- SoC 零代码接入
- IoT 平台固件升级及下单指导

## CONT.

- Add “DP ID” to cloud

功能定义

购买模组 管理产品

进入开发

自定义开发

导出功能

如何定义产品功能?

标准功能

+ 添加功能

DP ID	功能点	标识符	数据传输类型	功能点类型	功能点属性	备注	操作
1	开关	switch	可下发可上报	布尔型			编辑 删除

选择温控器功能

全部选择

已选功能(0)

删除可选功能

传感器选择

DP ID: 43 标识符: sensor\_choose



背光亮度

DP ID: 44 标识符: backlight



背光亮度

DP ID: 44 标识符: backlight



## CONT.

- Utilize “\N9H20\_emWin\_NonOS-master\BSP\SampleCode\emWin\Thermostat\_N9H20\_NonOS\Doc\DP\_ID\_List.xlsx” to add DP ID

自定义开发							
<div>标准功能 ?</div> <div><a href="#">导出功能</a>   <a href="#">如何定义产品功能</a></div> <div><a href="#">+ 添加功能</a></div>							
DP ID	功能点	标识符	数据传输类型	功能点类型	功能点属性	备注	操作
1	开关	switch	可下发可上报	布尔型			<a href="#">编辑</a> <a href="#">删除</a>
44	背光亮度	backlight	可下发可上报	数值型	数值范围: 0-100, 间距: 1, 倍数: 0, 单位: %		<a href="#">编辑</a> <a href="#">删除</a>

When update “DP ID”  
Please reset Tuya IoT module  
And update TUYA\_IoT library



## CONT.

- Utilize self-defined "DP ID"
  - E. g., switching menu

When update "DP ID"  
Please reset Tuya IoT module  
And update TUYA\_IoT library

## 新建自定义功能

① 标准功能点信息会影响固件开发、设备事件、控制界面，如果修改功能点信息，请确定相关信息是否受影响。

说明：自定义功能不支持第三方语音和第三方云对接。建议[提交工单](#)，涂鸦尽快帮您创建标准功能点。[如何自定义功能？](#)

\* DP ID: 101

\* 功能点名称: 功能点名称

\* 标识名: 支持字母、数字、下划线，以字母开头

\* 数据类型: ☐ 布尔型 ☐ 数值型 ☐ 枚举型 ☐ 故障型  
☐ 字符型 ☐ RAW型 ☐ ?

\* 数据传输类型: ☐ 可下发可上报 ☐ 只上报 ☐ 只下发 ☐ ?

1	DP ID	功能点	标识符	数据传输类型	功能点类型	功能点属性	备注
12	107	Menu	Menu	可下发可上报	value	数值范围：1-3, 间距：1, 倍数：0,	单位：

## CONT.

- Download Tuya IoT SDK

硬件开发

## 下载资料



## 开发资料

以下资料根据产品功能数据点自动生成开发资料，如果修改数据点、固件信息，请重新下载开发资料

下载全部

快速入门指南 ⓘ

下载

MCU SDK ⓘ

下载

通用模组串口协议 ⓘ

下载

涂鸦模组调试助手 ⓘ

下载

功能点调试文件 ⓘ

下载

- mcu\_sdk\_CS\_Thermostat\_20210118
- Debugfile\_CS\_Thermostat\_20210118.json
- protocol\_CS\_Thermostat\_20210118.pdf
- readme.txt
- TuYaCloudSerialPortHelper\_CS\_Thermostat\_20210118.zip
- 串口协议\_涂鸦智能\_文档中心.pdf

- mcu\_api.c
- mcu\_api.h
- protocol.c
- protocol.h
- readme.txt
- system.c
- system.h
- VERSION.md
- wifi.h

Please update "DP ID" and SDK file in TUYA\_IoT library

## CONT.

---

- **Tuya IoT SDK memory consumption**
  - ROM: 4KB
  - RAM: 100B

# CONT.

- SDK

- mcu\_api.c (top level API: mcu\_get\_wifi\_connect\_status)
- system.c (middle level API: wifi\_uart\_write\_frame)
- protocol.c (low level API: uart\_transmit\_output)

```
void mcu_get_wifi_connect_status(void)
{
    wifi_uart_write_frame(GET_WIFI_STATUS_CMD, MCU_TX_VER, 0);
}
```

```
void wifi_uart_write_frame(unsigned char fr_type, unsigned char fr_ver, unsigned short len)
{
    unsigned char check_sum = 0;

    wifi_uart_tx_buf[HEAD_FIRST] = 0x55;
    wifi_uart_tx_buf[HEAD_SECOND] = 0xaa;
    wifi_uart_tx_buf[PROTOCOL_VERSION] = fr_ver;
    wifi_uart_tx_buf[FRAME_TYPE] = fr_type;
    wifi_uart_tx_buf[LENGTH_HIGH] = len >> 8;
    wifi_uart_tx_buf[LENGTH_LOW] = len & 0xff;

    len += PROTOCOL_HEAD;
    check_sum = get_check_sum((unsigned char *)wifi_uart_tx_buf, len - 1);
    wifi_uart_tx_buf[len - 1] = check_sum;
```

```
void uart_transmit_output(unsigned char value)
{
    #error "请将MCU串口发送函数填入该函数,并删除该行"
```

# CONT.

- UART Rx

```
93 void UartDataValid_Handler(UINT8* buf, UINT32 u32Len)
94 {
95     //  UINT32 u32Idx;
96     INT32 i;
97
98     if(u32Len == 0)
99     {
100         sysprintf("uart rx Len = %d\n", u32Len);
101         return;
102     }
103     // for(u32Idx = 0; u32Idx < u32Len; u32Idx++)
104     // {
105     //     sysprintf("uart rx buf[%d] = 0x%x ",u32Idx, buf[u32Idx]);
106     //     sysprintf("\n");
107     // }
108
109     for(i = 0; i < u32Len; i++)
110     {
111         uart_receive_input(buf[i]);
112     }
113     memset(buf, 0, u32Len);
114 }
```

## CONT.

- UART Tx

```

85  /**
86      | | | | | 2: 串口单字节发送函数
87      | | | | | 请将MCU串口发送函数填入该函数内,并将接收到的数据作为参数传入串口发送函数
88      | | | | | ****
89
90  /**
91   * @brief  串口发送数据
92   * @param[in] {value} 串口要发送的1字节数据
93   * @return Null
94   */
95 void uart_transmit_output(unsigned char value)
96 {
97     #error "请将MCU串口发送函数填入该函数,并删除该行"
98
99     /*
100     //Example:
101     extern void Uart_PutChar(unsigned char value);
102     Uart_PutChar(value); //串口发送函数
103     */
104 }

```

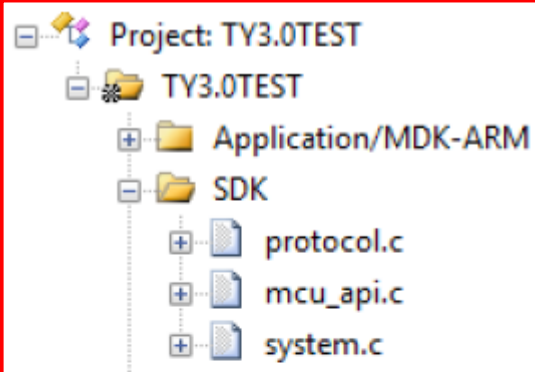
```

103 void uart_transmit_output(unsigned char value)
104 {
105     // #error "Please fill in the MCU serial port send function and delete the line"
106     Uart_PutChar(value);
107 }

```

## CONT.

## • SDK



名称

- mcu\_api.c
- protocol.c
- system.c
- mcu\_api.h
- protocol.h
- system.h
- wifi.h

```
1  #include "wifi.h"
2  ...
3  void main(void)
4  {
5      wifi_protocol_init();
6      ...
7      while(1)
8      {
9          wifi_uart_service();
10         ...
11     }
12     ...
13 }
```

## CONT.

- DPID (Data Pointer ID) upload

```
1 mcu_dp_bool_update(DPID_SWITCH,1); //BOOL 型数据上报
2 mcu_dp_value_update(DPID_TEMPER_SET,25); //VALUE 型数据上报
3 mcu_dp_string_update(DPID_DAY,"1234",4); //STRING 型数据上报
```

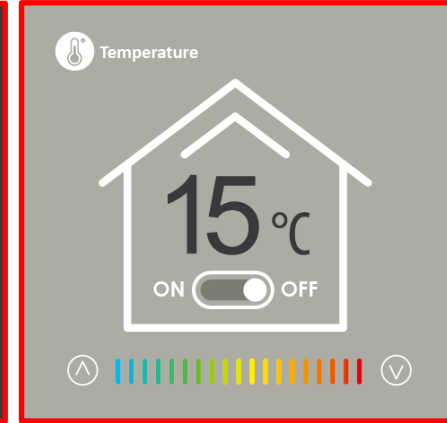
- DPID download

```
11 static unsigned char dp_download_switch_handle(const unsigned char v
12 alue[], unsigned short length)
13 {
14     //示例:当前 DP 类型为 BOOL
15     unsigned char ret;
16     //0:关状态/1:开状态
17     unsigned char switch1;
18
19     switch1 = mcu_get_dp_download_bool(value,length);
```



# Integrate Thermostat HMI

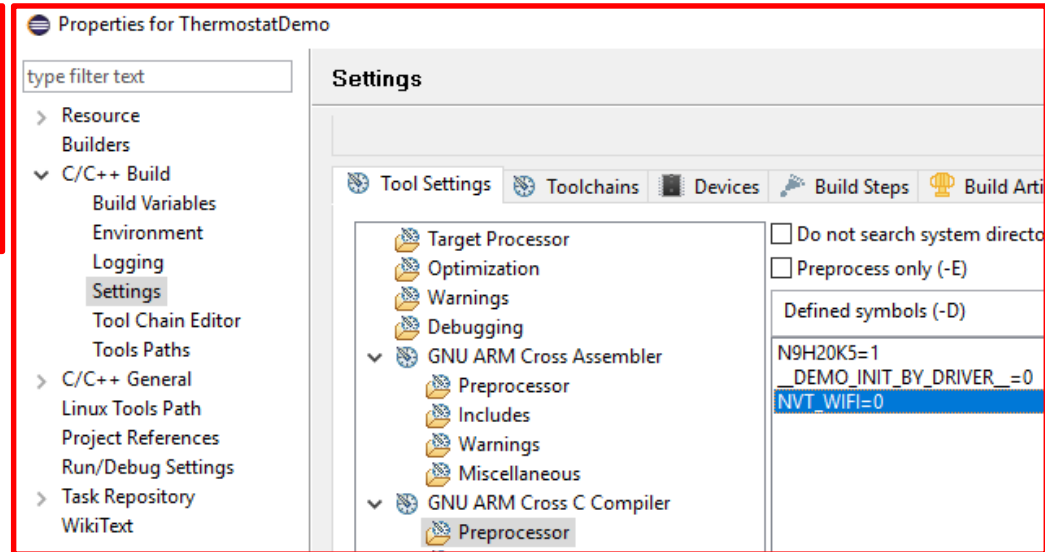
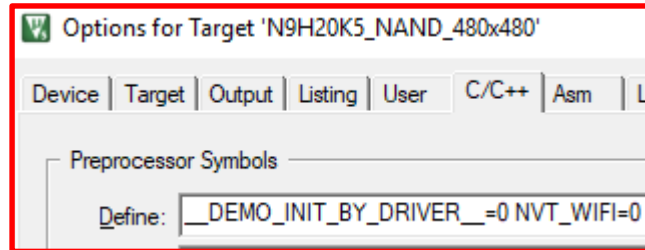
- **MENU1 - Fan, Cool, Heat & Dehumidify**
  - On / Off and level control
- **MENU2**
  - Current temp.
  - Temp. control
  - Backlight control
  - Modbus control
- **MENU3**
  - Calendar date-scheduler



N9H20 Thermostat

## CONT.

- Change UART1 from print debug message to connect with Tuya IoT module device
- Baud-rate is 9600 bps
- Default Thermostat HMI is disabled Tuya IoT feature “**NVT\_WIFI=0**”



## CONT.

- If “**NVT\_WIFI=1**”
  - Please make sure your have Tuya IoT module device (WBR3 / WR3)
  - Build TUYA\_IoT library
  - Copy TUYA\_IoT.lib or libTUYA\_IoT.a to  
    \N9H\_emWin\_Template\Thermostat\_N9H20\_NonOS\Tuya\_IoT
  - Re-build Thermostat

## CONT.

- **Build TUYA\_IoT library**

- In Thermostat template v2.1, there is a new added folder called "Tuya\_IoT"  
(\N9H\_emWin\_Template\Thermostat\_N9H20\_NonOS\Tuya\_IoT)
- Into "Tuya\_IoT" then copy the sub-folder called "TUYA\_IoT" to  
(\N9H20\_emWin\_NonOS-master\BSP\Driver\Source)
- Fill correct PID to "protocol.h" then re-build "TUYA\_IoT"

```
46 #define PRODUCT_KEY "please fill your PID" //开发平台创建产品后生成的16位字符产品唯一标识
```

- Your PID should be in downloaded SDK "protocol.h"

# Online Resource



Nuvoton Website

[www.nuvoton.com](http://www.nuvoton.com)

## Forum

**NuForum**

- <http://forum.nuvoton.com>



**牛卧堂**

- <http://www.nuvoton-MCU.com>



**21ic** 中国电子网

- <http://bbs.21ic.com/iclist-187-1.html>



## Media

**facebook**

- <https://www.facebook.com/NuvotonNuMicro/>



**WeChat**

- ID: nuvoton\_mcu



**Twitter**

- @NuvotonMCU



## BSP

**GitHub**

- <https://github.com/OpenNuvoton>



**GitLab**

- <https://gitlab.com/OpenNuvoton>



**G 码云**

- <https://gitee.com/OpenNuvoton>



## eShop

**nuvoton DIRECT**

- <https://direct.nuvoton.com>



**天猫 TMALL.COM**

- <http://nuvoton.tmall.com/>



**TECH DESIGN**

- [www.techdesign.com/market/nuvoton](http://www.techdesign.com/market/nuvoton)



# Q & A

## CONT.

- How to register Tuya IoT account?
  - <https://auth.tuya.com/?from=https%3A%2F%2Fiot.tuya.com%2F>

使用涂鸦账号登录

手机号或邮箱

密码

☒ 阅读并同意《服务条款》《法律声明》《隐私政策》

登录

还没有账号? 免费注册

忘记密码

## CONT.

- How to purchase Tuya IoT device (WBR3 / WR3)?





## CONT.

---

- **How to pair Tuya IoT device?**
  - Download Android APP “TuyaSmart” and follow the pairing flow

## CONT.

---

- **May I use Tuya IoT cloud service directly without another charge fee?**
  - Yes.

# CONT.

- Block diagram between Tuya IoT module and N9H20



## 1 概述

### 1 概述

涂鸦 Wi-Fi 串口通用协议为涂鸦定制的 Wi-Fi 模组串口通用协议，主要用于涂鸦 Wi-Fi 模组与其它 MCU 串口直连做串口通信，其架构框图如下图所示。

