

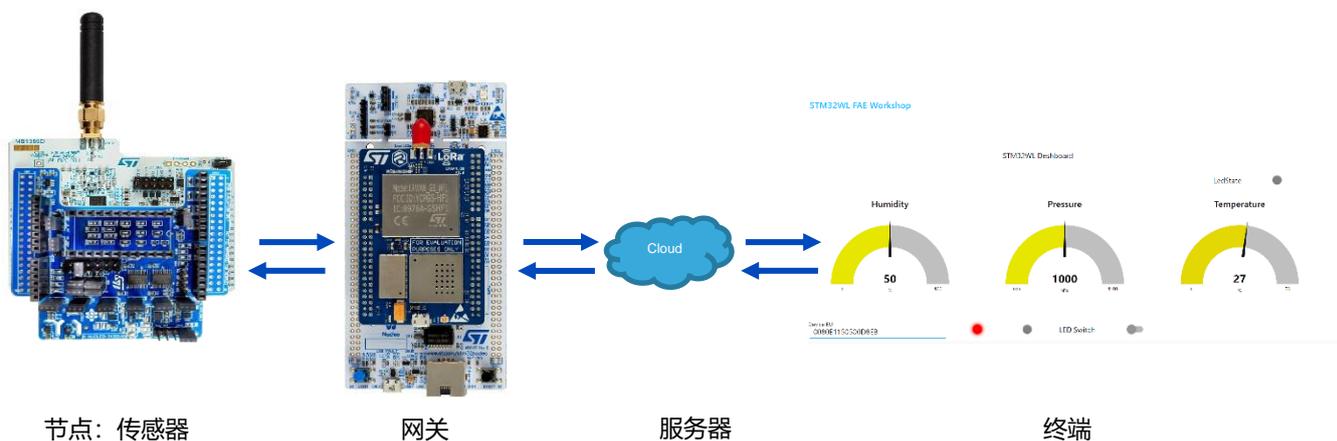
如何用 STM32CubeMonitor 图形化显示 Lora 节点信息

关键字: STM32CubeMonitor, Lora, 节点, 图形化

1. 目标

通过本 LAT 实现一个从 LoRa 节点、LoRa 服务器、终端应用之间的数据或者命令的相互传输的过程。

图 1 功能拓扑结构



2. 实验准备

为了实现上面的目标，需要准备一些必要的软件和硬件。软件的安装、硬件的介绍和 STM32CUBEMX 工具的使用，在本 LAT 中不做介绍。

2.1. 软件

- STM32CUBEMX
- STM32CubeMonitor
- STM32CUBEIDE(or 其它编译器), 本实验使用的 IAR 编译器。
- X-CUBE-MEMS1

2.2. 硬件

- NUCLEO-WL55JC2
- X-NUCLEO-IKS01A3

- STM NUCLEO Pack (P-NUCLEO-LRWAN2) ----- (LoRa LF Band sensor and gateway)

3. 服务器端账号和网关注册

这里只是简单描述，详情请参考 UM2587，可以到 WWW.ST.COM 网站下载

3.1. 账号注册

首先，要在远程服务器上注册自己的账户，本实验以 www.loriot.io 提供的免费服务器为例，创建自己的账户，见图 2。请注意，在注册时选择正确的服务器网站。在本实验中，我们选择了 Singapore 的服务器，网址是 ap1.loriot.io

图 2 注册账户

3.2. 配置网关

在成功正确的注册完账户之后，我们需要在这个账户下注册我们需要的网关。其中需要注意几个方面：

- 配置网关使用不同的频段，在本实验中我们使用的是 CN470,如果网关地址不是的 CN470 的话，需要使用下面 AT 指令对网关设备进行修改：

```
AT+CH=CN470PREQUEL
AT+CH=0,471.5,A
AT+CH=1,471.7,A
AT+CH=2,472.9,A
AT+CH=3,472.1,A
AT+CH=4,472.3,B
AT+CH=5,472.5,B
AT+CH=6,472.7,B
AT+CH=7,472.9,B
AT+CH=8,OFF
AT+CH=9,OFF
```

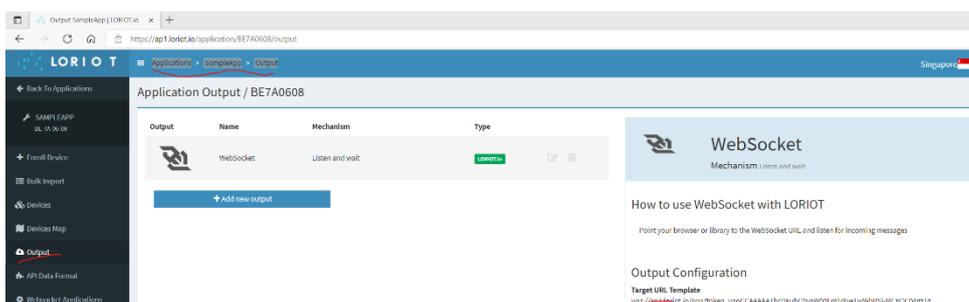
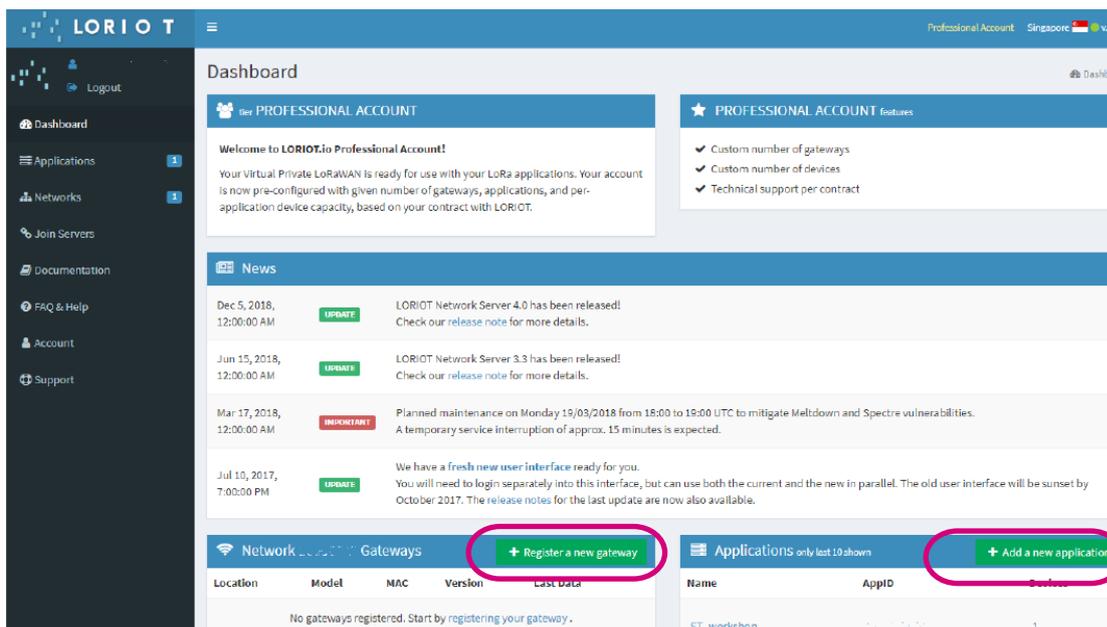
- 调整 LoRaWAN® 服务器, MAC 地址, 网关的 EUI。在做实验之前需要确认网关链接的网络服务器和你注册这个网关的服务器一致。本实验中使用的是 Singapore, 而缺省的是 EU1, 所以要使用下面 AT 指令进行修改。

```
AT+PKTFWD=ap1.loriot.io,1780,1780
```

3.3. 注册网关和设备

- 注册网关, 登录注册的账户后, 可以看到下图的 LORIENT 的 Dashboard。选择 “Register a new gateway” 进行网关注册, 一般我们选择基础平台为 “Packet Forwarder STM”, 然后从我们网关设备的串口 log 拷贝 MAC 地址。
- 注册设备, 请注意这点需要用到后面的一些参数 (device EUI, Join EUI, Application key), 所以刚开始实验可以先跳过这里。请选择 [Loriot Dashboard]>[Application]>[SampleApp]>[Enroll Device]。然后按照步骤注册。
- 在注册完成后, [Loriot Dashboard]>[Application]>[SampleApp]>[output]拷贝外部应用访问服务器的连接, 后面的 cubemx 需要用到。

图 3 注册网关和设备



4. 设备搭建

- 在创建工程之前，下载最新的 STM32CUBEWL 软件包、X-CUBE-MEMS1 和 STM32CUBEMX 并安装。

4.1. 创建工程

- 首先，打开 STM32CUBEMX，选择从 “Start my project from Example” 生成工程，然后搜索 Keyword “LoRaWAN_End_Node”。
- 然后，双击选中的工程。并且选择一个保存实验工程的路径。这个时候，我们就可以看到一个基本的 STM32CUBEMX 工程。
- 也可以直接从 STM32WL 的 Library 路径下面，在下面路径下，复制并粘贴工程文件 STM32Cube_FW_WL_V x.x.x\Projects\NUCLEO-WL55JC\Applications\LoRaWAN\LoRaWAN_End_Node，然后重命名为 LoRaWAN_End_Node_Sensor,最后打开工程文件里面的 STM32CUEMX 工程文件。
- 接着对工程做一些简单的修改：
 1. 配置频段，缺省的频段是欧洲频段，本实验使用的中国频段 CN470.见图 4.
 2. 增加 sensor 的配置。
 - a) 选择正确的 sensor 型号驱动软件，使用温度、湿度和压力。见图 5。
 - b) 使能访问 sensor 板的 I2C2 外设，根据 NUCLEO-WL55JC2 原理图，配置对应的 I2C 总线 GPIO 为 PA11 和 PA12,见图 6。
 - c) 使能 sensor 板，并且选择正确的 i2c 驱动见图 7。
 - d) 生成代码。见图 8

图 4. 配置频段

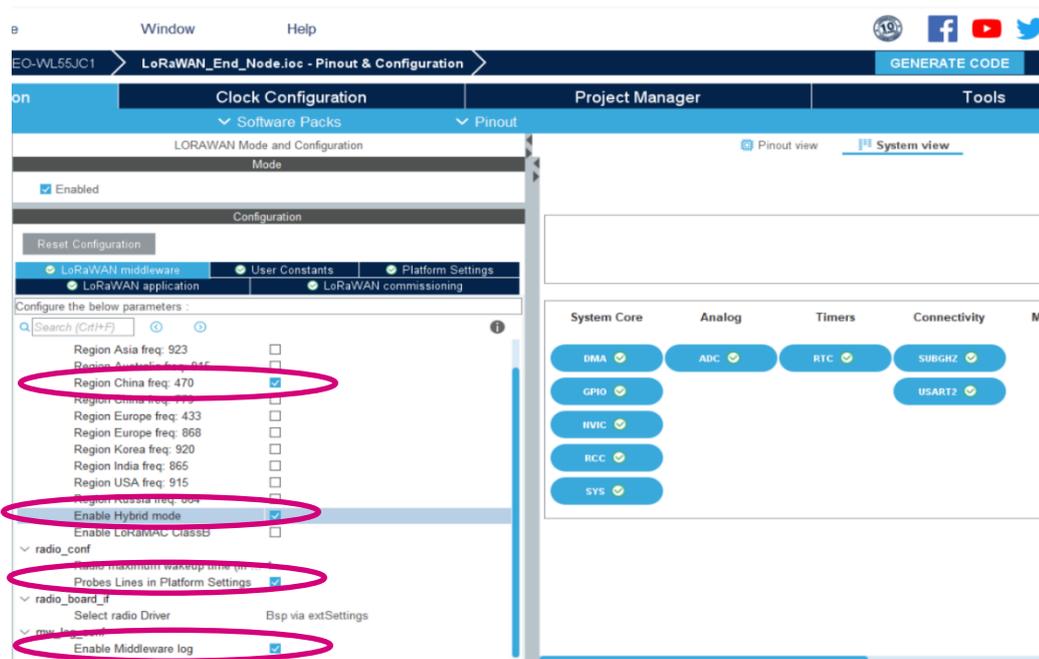


图 5: 选择 sensor 组件

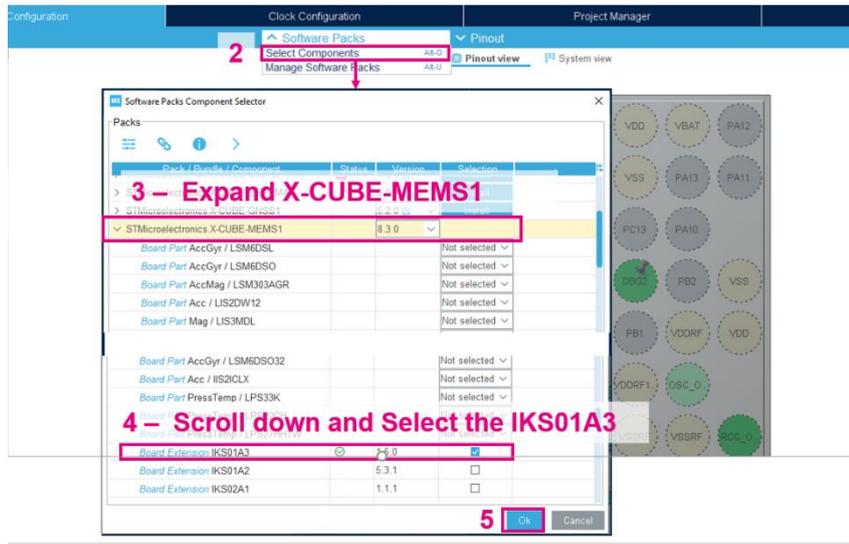


图 6. 使能 I2C2



图 7. 使能 sensor 板的 I2C

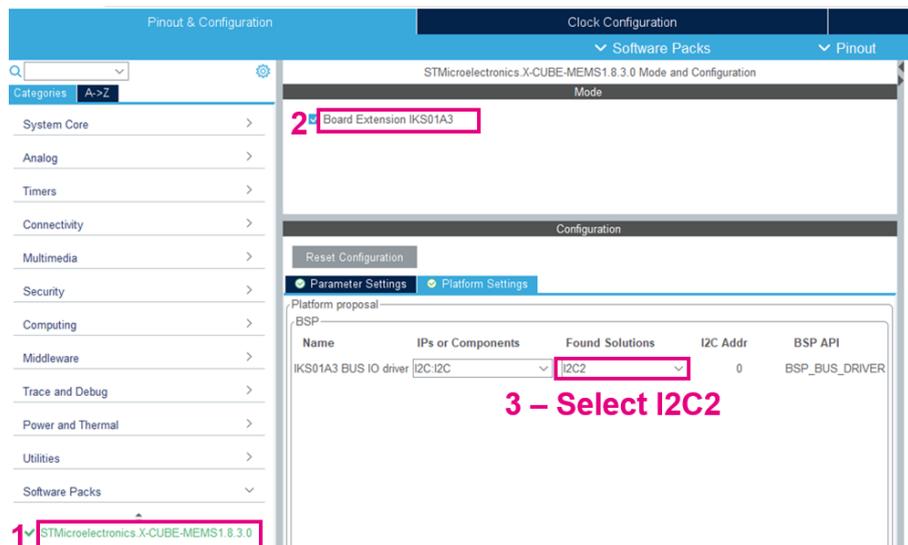
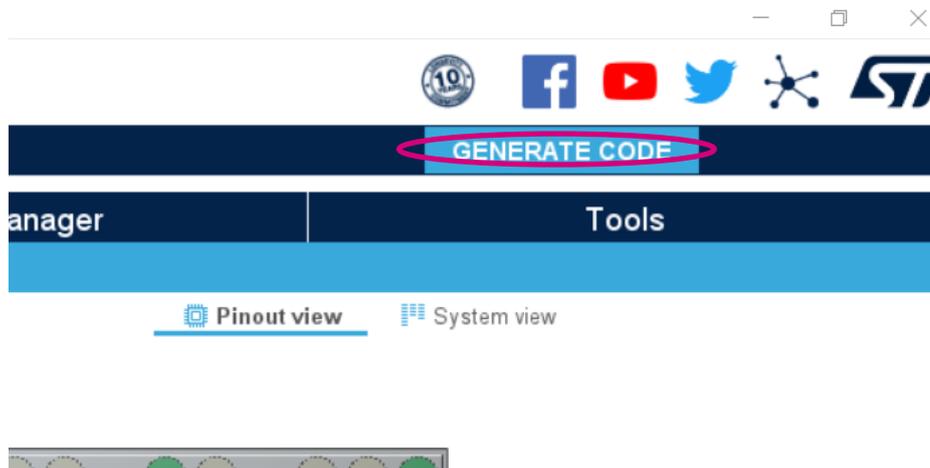


图 8. 生成代码



3. 代码修改

- (i) 添加宏定义，在 IAR 的 “Option 设置->C/C++ Compiler->Preprocessor” 的 “Define symbols” 中添加 X_NUCLEO_IKS01A3” ，见图 9。
- (ii) 找到 sys_sensors.h 使能 Sensor, 见图 10.

```
#define SENSOR_ENABLED      1
```
- (iii) 在 “lora_app.c” 的 “SendTxData” 函数中，修改 Lora 提取的上传数据，见图 11。
- (iv) 为了更加容易加入网络，将 CN470_FIRST_RX1_CHANNEL 改为 “(uint32_t) 470300000)” 。
- (v) 将 BSP_BUTTON_SWx_IT_PRIORITY 改为 15。
- (vi) 将由于有 sensor 读取，所以关掉低功耗模式。

```
#define LOW_POWER_DISABLE      1
```

图 9. 宏添加

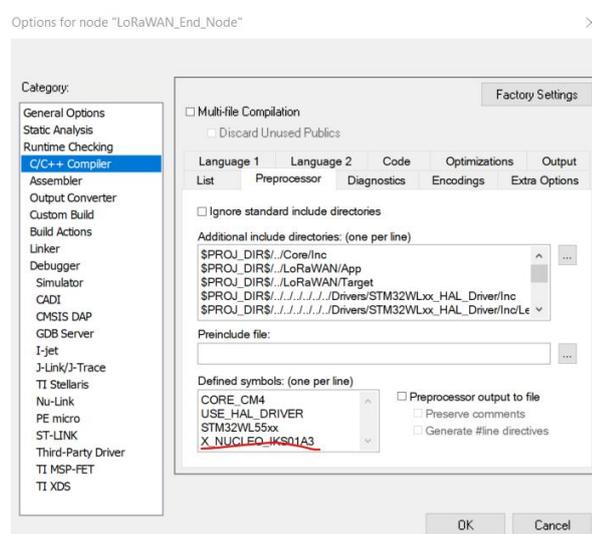


图 10. 使能 sensor

```

    /**
     * USER CODE BEGIN Header
     */
    /* USER CODE END Header */

    /* Define to prevent recursive inclusion -----*/
    #ifndef SENSORS_H
    #define SENSORS_H

    #ifdef __cplusplus
    extern "C" {
    #endif
    /* Includes -----*/

    /* USER CODE BEGIN Includes */

    #define SENSOR_ENABLED 1
    /* USER CODE END Includes */

    /* Exported types -----*/
    /**

```

图 11. 上传数据

<pre> EnvSensors_Read(&sensor_data); // temperature = (SYS_GetTemperatureLevel() >> 8); temperature = (uint16_t)(sensor_data.temperature); pressure = (uint16_t)(sensor_data.pressure); /* in hPa / 10 */ AppData.Port = LORAWAN_USER_APP_PORT; #ifdef CAYENNE_LPP CayenneLppReset(); CayenneLppAddBarometricPressure(channel++, pressure); CayenneLppAddTemperature(channel++, temperature); CayenneLppAddRelativeHumidity(channel++, (uint16_t)(sensor_data.humidi if ((LmHandlerParams.ActiveRegion != LORAMAC_REGION_US915) && (LmHandl && (LmHandlerParams.ActiveRegion != LORAMAC_REGION_AS923)) { CayenneLppAddDigitalInput(channel++, GetBatteryLevel()); CayenneLppAddDigitalOutput(channel++, AppLedStateOn); } CayenneLppCopy(AppData.Buffer); AppData.BufferSize = CayenneLppGetSetSize(); #else /* not CAYENNE_LPP */ humidity = (uint16_t)(sensor_data.humidity); /* in %*10 </pre> <p style="text-align: center; color: red; font-weight: bold;">修改后</p>	<pre> EnvSensors_Read(&sensor_data); temperature = (SYS_GetTemperatureLevel() >> 8); pressure = (uint16_t)(sensor_data.pressure * 100 / 10); AppData.Port = LORAWAN_USER_APP_PORT; #ifdef CAYENNE_LPP CayenneLppReset(); CayenneLppAddBarometricPressure(channel++, pressure); CayenneLppAddTemperature(channel++, temperature); CayenneLppAddRelativeHumidity(channel++, (uint16_t)(sensor_data if ((LmHandlerParams.ActiveRegion != LORAMAC_REGION_US915) && (&& (LmHandlerParams.ActiveRegion != LORAMAC_REGION_AS923)) { CayenneLppAddDigitalInput(channel++, GetBatteryLevel()); CayenneLppAddDigitalOutput(channel++, AppLedStateOn); } CayenneLppCopy(AppData.Buffer); AppData.BufferSize = CayenneLppGetSetSize(); #else /* not CAYENNE_LPP */ humidity = (uint16_t)(sensor_data.humidity * 10); </pre> <p style="text-align: center; color: blue; font-weight: bold;">修改前</p>
---	---

(vii)最后编译运行,可以从串口 log 看到下面的运行信息,如图 12,然后拷贝设备信息(device EUI, Join EUI, Application key)继续完成 3.3 对设备的注册。

图 12. 运行结果

5. 创建 STM32CubeMonitor 工程

通过创建一个 STM32CubeMonitor 工程,来显示 sensor 的数据,从而达到终端数据的利用展示。

5.1. 创建一个新的 STM32CubeMonitor 工程。

1. 打开 STM32CubeMonitor, 单击图 12 的 “+” 图标, 创建一个 flow。
2. 双击默认 basic flow, 然后单击 “Delete” 按钮, 见图 13。
3. 修改新 flow 的名字为自己想要的名字, 这里改为 STM32WL_Sensor_Dashboard, 见图 14。
4. 创建与 flow 关联的 Dashboard, 见图 15。
5. 创建关联的 group 并且修改, 将鼠标放置在新建 tab 上, 选择 “+tab” 添加 group, 并且通过选择 “edit” 编辑需要的名字, 这里设为 “STM32WL_Sensor_Dashboard”, 见图 16。
6. 用与步骤 5 相同的方式, 修改 group 为 “STM32WL_Sensor”

图 12. 创建新 flow

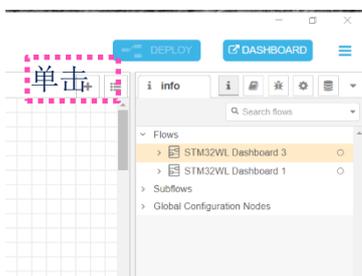


图 15. 创建 Dashboard



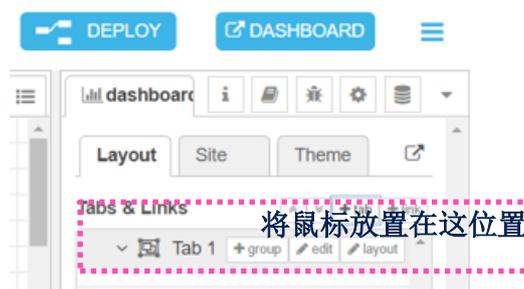
图 13. 删除默认 flow



图 14. 修改 flow 名字



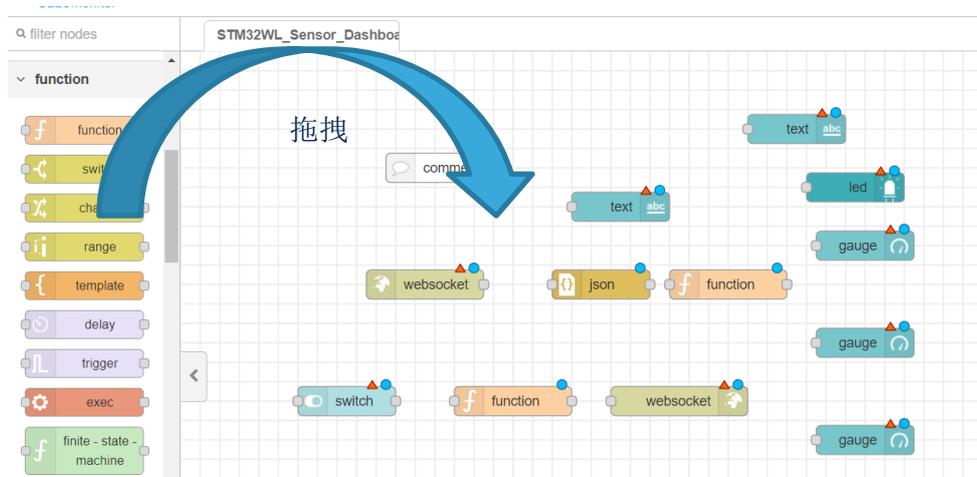
图 16. 修改 flow 名称



5.2. 添加需要的控件。

1. 从控件面板中拖拽需要的控件到 flow 中，从而实现需要显示的功能。这里需要用到 2 个 TEXT 控件来显示文本信息；3 个 gauge 来显示 sensor 数据；两个 function 控件来，转换 json 格式，一个 comment 来提示输入；一个 websocket 输入和一个 websocket 输出；一个 led 来显示板子 LED 灯状态，一个 switch 来控制板子的 LED 灯的显示。见图 17。

图 17. 修改 flow 名字



2. 逐个双击 flow 上的每个控件打开其属性窗口，接着对逐个的参数进行修改，并将它关联到“STM32WL_Sensor” group。
 - 修改“comment”控件，修改名字为“Remind”在文本框内添加一些提醒内容，如“Edit the "Loriot Uplink" node with your LORIoT WebSocket Token; you will find your LORIoT WebSocket Token under Application -> SampleApp -> Output.”
 - 修改其中一个“TEXT”控件，修改 Label 为“STM32WL_Sensor_Dashboard”；修改 Name 为“Title”；选择 group 为“STM32WL_Sensor”；size 设置 6x2，最后我们还再统一调整。在 layout 根据需要选择显示时的字体排列。
 - 修改另外一个“TEXT”控件，修改 Label 为“LED”；修改 Name 为“LED Status”；选择 group 为“STM32WL_Sensor”；size 设置 1x1，在 layout 根据需要选择显示时的字体排列。
 - 修改“Websocket in”控件，选择 Type 为“connect on”；修改 Name 为“Loriot uplink”；添加 path 为前面提到的从服务器获取的 link（见 3.3），这个例程中用到为
 “wss://ap1.loriot.io/app?token=vnoGCAAAA1hcDEubG9yaW90LmlvXyeT wWbRSI-MCBQC04gzTg==”
 - 修改“Websocket out”控件，选择 Type 为“Connect to”；修改 name 为“Loriot DownLink”， path 它会自动填充 websocket in 的路径。

- 修改“LED”控件，修改 Name 为“LED”；选择 group 为“STM32WL_Sensor”；size 设置 1x1;在“Colors for value of msg.payload”下，单击“X”删掉一个选项，并且点击下拉箭头将类型选为“number”，同时将 number 设为“1”，然后选择一个你喜欢的颜色，意思是当收到数字为“1”时，显示红色。如图 18.

图 18. LED 设置



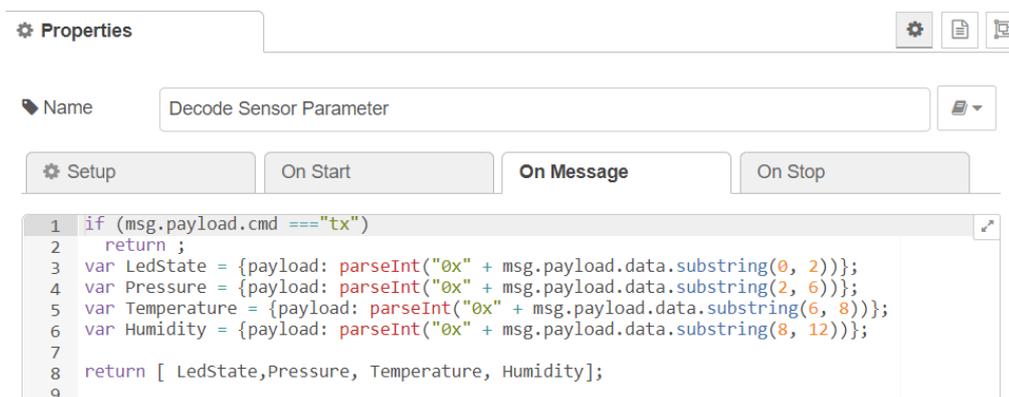
- 修改“json”控件，修改 Name 为“json”，其它的默认就可以了。
- 修改一个 function 控件，修改名字为“Decode Sensor Parameter”，这里我们需要写一个函数来提取 sensor 数据，在“ON message”选项添加函数如图 19:

```

if (msg.payload.cmd === "tx") return ;
var LedState = {payload: parseInt("0x" + msg.payload.data.substring(0, 2))};
var Pressure = {payload: parseInt("0x" + msg.payload.data.substring(2, 6))};
var Temperature = {payload: parseInt("0x" + msg.payload.data.substring(6, 8))};
var Humidity = {payload: parseInt("0x" + msg.payload.data.substring(8, 12))};
return [ LedState,Pressure, Temperature, Humidity];
    
```

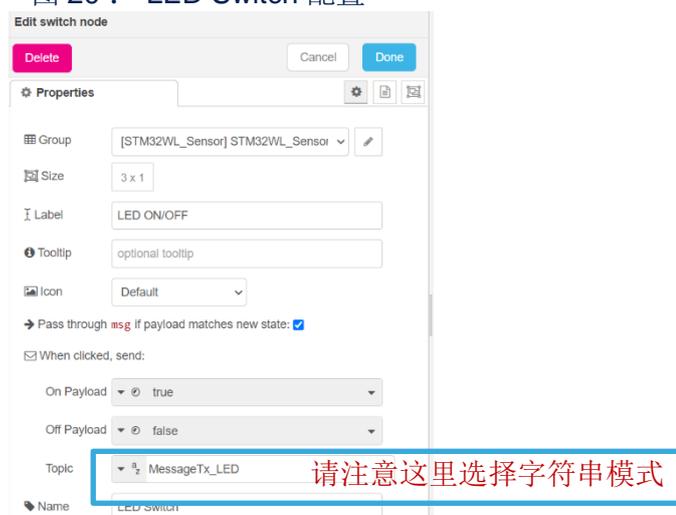
同时在“Setup”选项中设置输出个数为 4;

图 19. Decode 函数



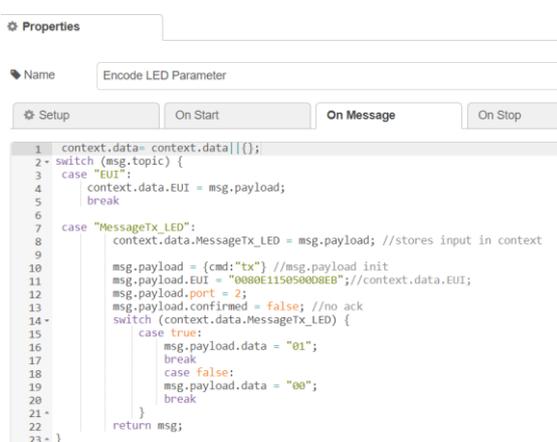
- 修改“Switch”控件，修改 Label 为“LED ON/OFF”；修改 Name 为“LED Switch”；选择 group 为“STM32WL_Sensor”；size 设置为 3x1; 添加一个 topic 域为“MessageTx_LED”，如图 20.

图 20：LED Switch 配置



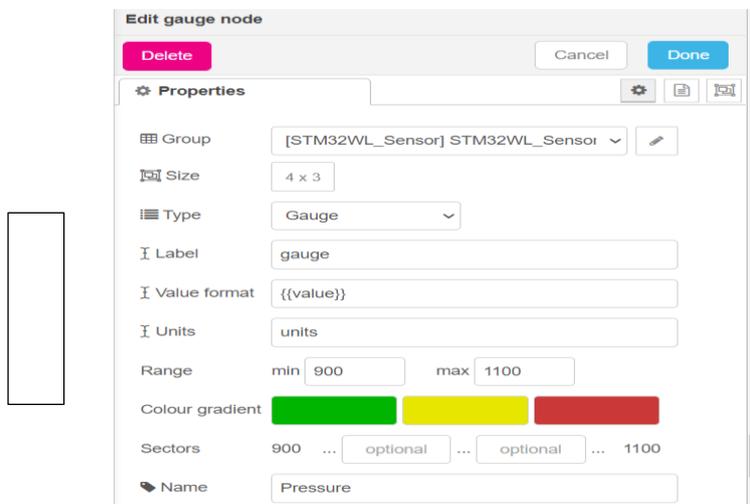
- 修改另一个“function”控件，修改名字为“Encode LED Parameter”，这里需要一个函数来提取 sensor 数据，在“ON message”选项添加函数如图 21，请注意这里用了 Device EUI，这里需要从节点设备的信息 log 中获取。

图 21.Encode sensor data 函数。



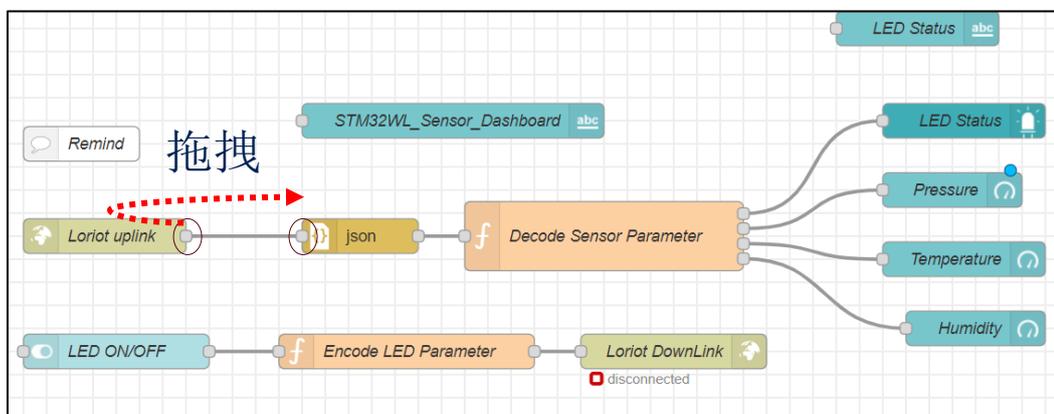
- 修改“Gauge”控件，现在任意一个“Gauge”控件，双击设置属性，选择 group 为“STM32WL_Sensor”，size 设置为 4x3，Label 为“Pressure”，range 的 min 为“900”，max 为“1100”，如图 22。
- 同理修改其它“Gauge”控件，选择一个“Gauge”控件，双击设置属性，选择 group 为“STM32WL_Sensor”，size 设置为 4x3，Label 为“Temperature”，range 的 min 为“0”，max 为“50”。
 - 修改“Gauge”控件，将最后一个“Gauge”控件，双击设置属性，选择 group 为“STM32WL_Sensor”，size 设置为 4x3，Label 为“Humidity”，range 的 min 为“0”，max 为“100”。

图 22. 修改 LED Switch 配置



- 修改完控件之后，我们需要将其所有的控件关联在一起。将一个控件的节点拖拽到另一个需要关联的控件即可，如图 23。这里要特别注意的是 decoder 的输出函数的输出值的顺序必须跟你需要显示的值一致，这里是按照“LedState,Pressure, Temperature, Humidity”顺序，所以必须是 led 控件连接到最上面一个输出口。

图 23. 关联控件



- 按照之前 5.1.5 的方式再次选择 Dashboard-》 layout 对显示界面进行布局。如图 24。图中右侧宽度是总的显示宽度，长度自动根据显示而设定的。单击控件上的锁定图标，从而锁定显示，然后拖拽而改变大小，按住鼠标左键移动位置。在在适当调整之后，点击 STM32CubeMonitor 右上角的“DEPLOY”部署工程,然后点击“DASHBOARD”,显示效果，如果不满意，可以再次用前面的方式对界面多次布局，直到满意为止。如图 25,。

图 24 ， 布局显示界面

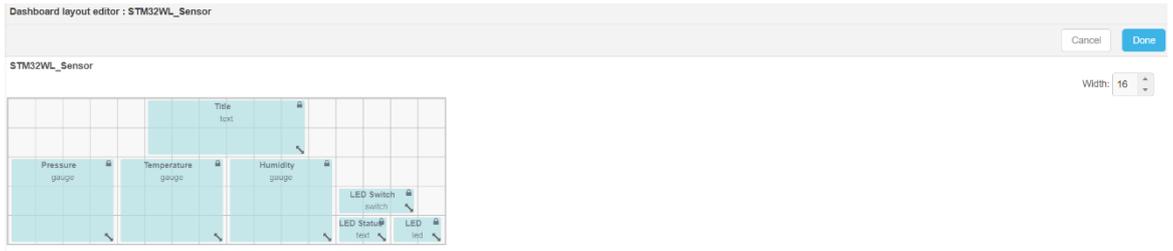
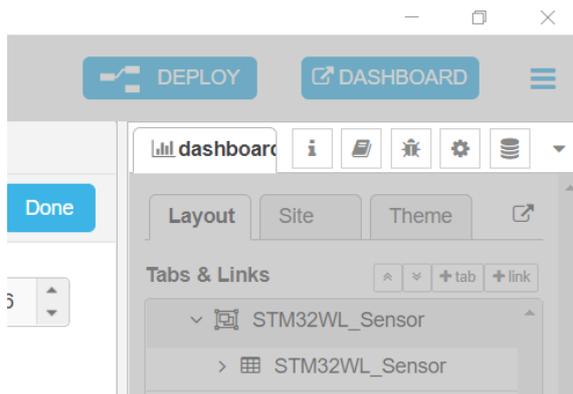
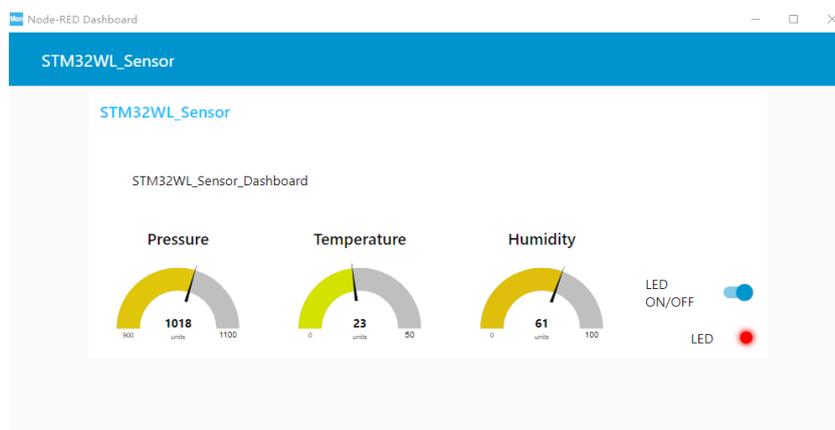


图 25 部署或显示



- 将网关通电确保接入外部网络（Internet），然后 sensor 节点通电，确保节点加入 LoRa 网络，可以通过节点和网关的串口数据确认是否正确，在网关和节点工作正常后，点击“DASHBOARD”按钮，我们就可以看到图 26 的效果。

图 26.效果



参考文献

【如有，请注明；否则，请注明：无】

文件编号	文件标题	版本号	发布日期
UM2587	Getting started with the P-NUCLEO-LRWAN2 and P-NUCLEO-LRWAN3 starter packs	Rev 2	April 2021
UM1718	STM32CubeMX for STM32 configuration and initialization C code generation	Rev 35	June 2021
Wiki	https://wiki.stmicroelectronics.cn/stm32mcu/wiki/STM32CubeMonitor:STM32CubeMonitor_overview		

文档中所用到的工具及版本

- STM32CUBEMX 版本 6.3
- STM32CUBEMonitor 版本 1.3
- IAR 编译器 版本 9.0
- X-CUBE-MEMS1 版本 9.0

LAT 中的附件

【如有，请注明；否则，请注明：无】

版本历史（内部参考）

（仅供内部参考，评审完成后 Auditor 需要删除这部分内容）

日期	版本	变更	撰写/变更者
2021年10月22日	0.1	原始版本	Min DU
2021年11月27日	0.2	Reviewed	Willee gong
2022年1月17日	0.3	个别字词调整、格式版本维护	Miler Shao

版本历史

日期	版本	变更
YYYY年MM月DD日	1.0	详细变更信息

重要通知 - 请仔细阅读

意法半导体公司及其子公司 (“ST”) 保留随时对 ST 产品和 / 或本文档进行变更的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。若需 ST 商标的更多信息，请参考 www.st.com/trademarks。所有其他产品或服务名称均为其各自所有者的财产。

本文档是 ST 中国本地团队的技术性文章，旨在交流与分享，并期望借此给予客户产品应用上足够的帮助或提醒。若文中内容存有局限或与 ST 官网资料不一致，请以实际应用验证结果和 ST 官网最新发布的内容为准。您拥有完全自主权是否采纳本文档（包括代码，电路图等信息，我们也不承担因使用或采纳本文档内容而导致的任何风险。

本文档中的信息取代本文档所有早期版本中提供的信息。