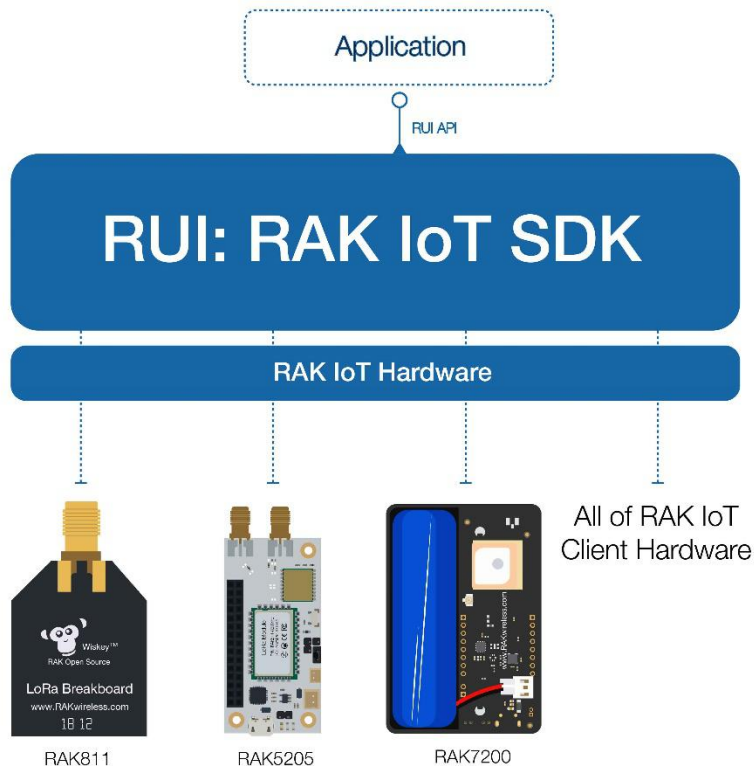


RAK LoRaWAN 节点模块 openMCU 开发指导手册

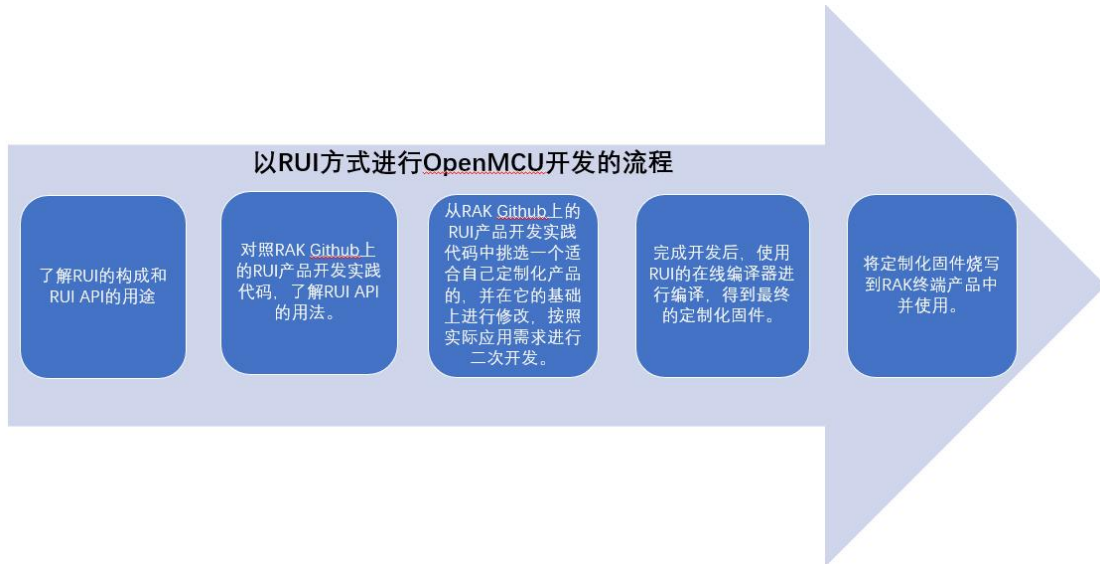


背景： 如上图所示，RAK LoRaWAN 节点模块以及相关的传感器产品（内嵌模块）除了支持传统的 AT 命令使用方式外，还允许客户利用内部 MCU 做应用开发。RAK 通过提供了一种命名为 RUI 的软件框架以及一组 RUI API，供用户根据自己的应用需求进行二次开发，形成自己的定制化软件。

范围： 本文档主要针对初次使用 RAK LoRaWAN 模块产品以 OpenMCU 方式做二次开发的开发者们，引导大家如何逐步了解 RUI API 和 RUI 生态，进而完成基于 RAK LoRaWAN 终端模块的二次开发。当前支持的 LoRaWAN 模块型号包括

- RAK811 LoRaWAN 模块
- RAK4200 LoRaWAN 模块
- RAK4600. LoRaWAN+BLE 模块

流程介绍：



总体来说，在 RAK 终端模块上，围绕 RUI 进行二次开发的基础和核心是熟悉 RUI 的构成以及 RUI API 的用法。具体的切入步骤，如上图所示，分为 5 个步骤，具体为：

1、了解 RUI 的构成和 RUI API 的用途。

基于 RUI 开发的固件，由两部分组成：RUI SDK，以及 Application，如本文图 1 所示。

其中，RUI SDK 部分为 RUI 的内核，已经适配了 RAK 的 LoRaWAN 模块产品，同时向上提供一组统一的 RUI API，供 Application 部分调用。

RUI API 的用途和用法介绍，请参考 RUI API 手册：

<https://doc.rakwireless.com/developer-tools/developer-tools/>

注：1) Application 是开放源码的，用户可以对其进行二次开发，按照自己的应用逻辑进行修改，需要的底层功能支撑都有相应的 RUI API 可以调用。

2) RUI SDK 部分是非开放源码的，它被放置于 RUI 在线编译器中，当用户完成 Application 部分的二次开发并使用 RUI 在线编译器进行编译时，RUI SDK 会被自动加载到定制化固件中。

2、对照 RAK Github 上的 RUI 产品开发实践代码，了解 RUI API 的用法。

RAK 的 LoRaWAN 模块及衍生产品，其官方发布的固件所对应的 Application 代码，均可以在 RAK 的 Github 中找到，地址为：

https://github.com/RAKWireless/Products_practice_based_on_RUI

■ RUI online compiler	Create How to use RUI online compiler.pdf
■ based on RAK4200	Update lora api.
■ based on RAK4400/4400	delete rui.h.
■ based on RAK4600	Update join processing.
■ based on RAK5010	hologram debug
■ based on RAK811	modify stm based RAK811 adr RUI
■ based on RAK8212	hologram debug
■ common header	Change SNR type.
📄 LICENSE	Initial commit
📄 README.md	Update README.md

可以看到，这个代码仓是按照核心模块的型号来分目录的。我们只需要找到对应的核心模块目录，就可以看到基于该模块的一些产品实践代码了。

以 RAK811 模块为例，当我们进入到“based on RAK811”目录后，可以看到一些基于 RAK811 模块的产品实践示例，这些代码展示了如何基于 RAK811 模块增加一些外设和传感器，通过这些示例代码可以让开发者更快了解如何通过 RUI 加载传感器：

app_5205	modify stm based RAK811 adr RUI
app_7204	modify stm based RAK811 adr RUI
app_PT100_Digital_Pressure_Sensor	update
app_RAK811	modify stm based RAK811 adr RUI
README.md	The first version

其中，

1) “app_5205” 是 RAK5205 的官方固件所对应的 Application 源码。













BME680-board.c	update based on RAK811
BME680-board.h	update based on RAK811
BME680.c	update based on RAK811
BME680.h	update based on RAK811
BME680_defs.h	update based on RAK811
app_5205.c	optimize flash store,fix 5205 sleep sensor bug
at_cmd.c	modify RAK811:add 1 channel PWM function at RAK811 pin_num 2(For RUI ...
at_cmd.h	update RAK5205&RAK7204
bsp.c	update app_5205
bsp.h	updata RAK811_20191104_1
gps.c	update RAK811
gps.h	update RAK5205&RAK7204
lis3dh_reg.c	update RAK811
lis3dh_reg.h	update based on RAK811
lora_config.c	modify stm based RAK811 adr RUI
lora_config.h	updata RAK811_20191104_1

RAK5205 是基于 RAK811 模块，增加了环境监测传感器 (BME680)、三轴加速度传感器 (LIS3DH)、以及 GPS 模块 (Ublox MAX 7Q)。该目录中的代码示例展示了如何基于 RAK811 模块，通过 I2C 挂接其它 sensor 以及通过 UART 挂接 GPS 模块，并通过 RUI API 调用的方式完成定制化固件的开发。

2) “app_7204” 是 RAK7204 的官方固件所对应的 Application 源码，包括 RAK7204 的传感器驱动、AT 命令实现、应用逻辑处理等，都在这里，这里面使用了很多 RUI API，用于实现相关功能。。

BME680-board.c	update RAK5205&RAK7204
BME680-board.h	update based on RAK811
BME680.c	update based on RAK811
BME680.h	update based on RAK811
BME680_defs.h	update based on RAK811
app_7204.c	optimize flash store,fix 5205 sleep sensor bug
at_cmd.c	modify RAK811:add 1 channel PWM function at RAK811 pin_num 2(For RUI ...
at_cmd.h	update RAK5205&RAK7204
lora_config.c	modify stm based RAK811 adr RUI
lora_config.h	update RAK5205&RAK7204
sensor.c	update RAK811
sensor.h	update RAK7204

RAK7204 是基于 RAK811 模块，增加了环境监测传感器（BME680）。因此，可以通过该产品实践代码了解到如何基于 RAK811 模块增加一款 I2C 的传感器，并通过 RUI API 的调用，快速完成定制化固件的开发。例如，“app_7204.c”是应用逻辑处理的实现，如果要做应用逻辑的定制开发，可以重点对照这里的代码来理解 RUI API 的实际用法。

 BME680-board.c	update RAK5205&RAK7204
 BME680-board.h	update based on RAK811
 BME680.c	update based on RAK811
 BME680.h	update based on RAK811
 BME680_defs.h	update based on RAK811
 app_7204.c	optimize flash store,fix 5205 sleep sensor bug
 at_cmd.c	modify RAK811:add 1 channel PWM function at RAK811 pin_num 2(For RUI ...
 at_cmd.h	update RAK5205&RAK7204
 lora_config.c	modify stm based RAK811 adr RUI
 lora_config.h	update RAK5205&RAK7204
 sensor.c	update RAK811
 sensor.h	update RAK7204



3) “app_PT100_Digital_Pressure_Sensor” 是基于 RAK811 模块，增加一款液
压传感器 PT100.

4) “app_RAK811” 是 RAK811 模块的官方固件所对应的 application 代码，如果
想要修改 RAK811 模块的工作逻辑和流程，可以基于该实例代码进行修改。

**3、从 RAK Github 上的 RUI 产品开发实践代码中挑选一个合适的实例代码，并在它的
基础上进行修改，按照实际应用需求进行二次开发。**

RAK Github 上的这些产品开发实践代码已经实现了 Application 的主体部分，基于
它进行代码修改和二次开发将事半功倍。

例如，前文所述的 RAK7204 示例代码：

 app_5205	modify stm based RAK811 adr RUI
 app_7204	modify stm based RAK811 adr RUI
 app_PT100_Digital_Pressure_Sensor	update
 app_RAK811	modify stm based RAK811 adr RUI
 README.md	The first version

里面对于节点收到的 LoRa 下行数据，只是将其通过串口打印出来，并未做处理，这一点可以从“app_7204.c”文件中的代码看出：

```
void LoRaReceive_callback(RUI_RECEIVE_T* Receive_datapackage)
{
    char hex_str[3] = {0};
    RUI_LOG_PRINTF("at+recv=%d,%d,%d,%d", Receive_datapackage->Port, Receive_datapackage->Rssi, Receive_datapackage->Snr, Receive_datapacka

    if ((Receive_datapackage->Buffer != NULL) && Receive_datapackage->BufferSize) {
        RUI_LOG_PRINTF(":");
        for (int i = 0; i < Receive_datapackage->BufferSize; i++) {
            sprintf(hex_str, "%02x", Receive_datapackage->Buffer[i]);
            RUI_LOG_PRINTF("%s", hex_str);
        }
    }
    RUI_LOG_PRINTF("\r\n");
}
```

如果你希望自己的定制化固件中对 LoRa 的下行数据进行处理，可以通过修改该函数的代码实现来达成目标。

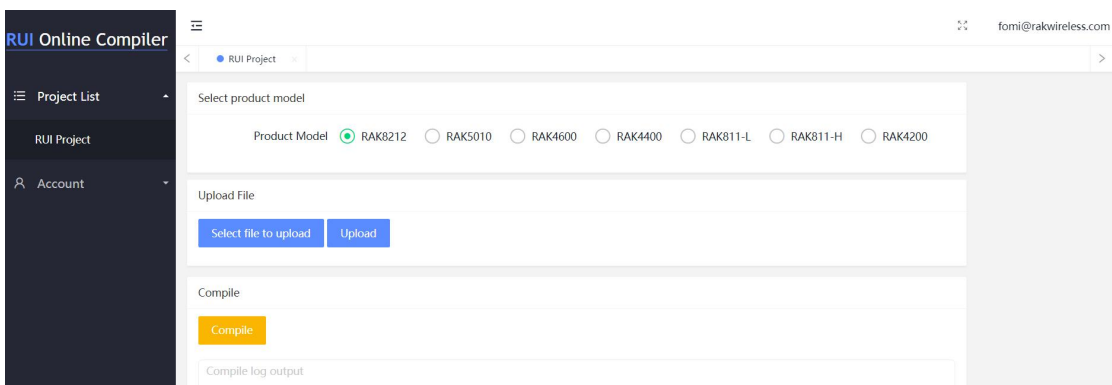
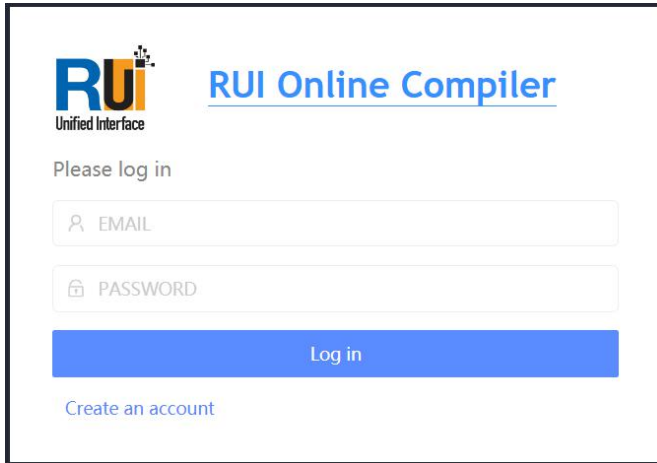
4、完成二次开发后，使用 RUI 的在线编译器进行编译，得到最终的定制化固件。

完成 Application 部分的定制化开发后，就可以使用 RUI 的在线编译器进行编译了，如前问所述，编译时，编译器会自动将 Application 和 RUI SDK 进行加载并编译都一起，输出最终的定制化固件。

RUI 在线编译器的地址为：<http://47.112.137.11:12090/#/user/login>

可以用邮箱自行注册并使用，详细的使用方法请参见 RUI 在线编译器的使用说明文档：

<https://downloads.rakwireless.com.cn/cn/RUI/%E5%A6%82%E4%BD%95%E4%BD%BF%E7%94%A8RUI%E5%9C%A8%E7%BA%BF%E7%BC%96%E8%AF%91%E5%99%A8.pdf>



5、将编译出的定制化固件烧写到 RAK 物联网终端产品中并使用

OK，经过前面的步骤，你已经编译得到了最终的定制化固件二进制文件了，现在可以将它烧写到产品中进行使用了。烧写固件的方法，在对应的产品使用文档中都有详述，此处不再复述，可到 <https://doc.rakwireless.com/> 查找对应的产品使用文档。

以 RAK7204 为例：

<https://doc.rakwireless.com/rak7204-lora---environmental-sensor/burning-the-firmware>

按照文档中的操作步骤执行即可完成固件烧写。