

S76G/S78G SDK1 and SDK2 Q&A



Document Name	S76G/S78G SDK1 and SDK2 Q&A
Version	V1.0.0.zh-CN
Doc No	
Date	Jul. 25, 2019

Index

1. S76G/S78G SDK1

1.1 LoRaWAN Q&A

- 1.1.1 [支持的区域](#)
- 1.1.2 [Device EUI, Application EUI, Application KEY for OTAA join.](#)
- 1.1.3 [Network Security KEY, Application Security KEY, Device Address for ABP join.](#)
- 1.1.4 [S7xG SDK1 当中的 LoRaMac mac 版本为多少?](#)
- 1.1.5 [S7xG SDK1 中的 LoRaWAN stack 功能实现依据为?](#)
- 1.1.6 [在 LoRaWAN 终端装置上如何编译成 Class C?](#)
- 1.1.7 [入网方法是 OTAA 或 ABP?](#)

1.2 The other Q&A

- 1.2.1 [如何除错?](#)
- 1.2.2 [Run-time variables monitoring tool](#)

2. S76G/S78G SDK2

2.1 LoRaWAN Q&A

- 2.1.1 [支持的区域](#)
- 2.1.2 [Device EUI, Application EUI, Application KEY for OTAA join.](#)
- 2.1.3 [Network Security KEY, Application Security KEY, Device Address for ABP join.](#)
- 2.1.4 [S7xG SDK2 当中的 LoRaMac mac 版本为多少?](#)
- 2.1.5 [S7xG SDK2 中的 LoRaWAN stack 功能实现依据为?](#)
- 2.1.6 [在 LoRaWAN 终端装置上如何编译成 Class C?](#)
- 2.1.7 [入网方法是 OTAA 或 ABP?](#)
- 2.1.8 [Lib 的路径?](#)
- 2.1.9 [Lib 有哪些函式?](#)
- 2.1.10 [如何在范例里更换至其他区域的 Lib?](#)
- 2.1.11 [执行 mac Tx\(\) 函式后, 打印显示 ">> busy"](#)
- 2.1.12 [传送一个封包后, 为何之后无法立刻再传送? 须等多久才可再传送?](#)
- 2.1.13 [为何 LoRaWAN 自动重传机制下, 当自动重传时, 每次频点\(频率/信道\)都不一样?](#)
- 2.1.14 [SDK2 里, 可否增加 RX1 频点\(频率/信道\)的设定功能?](#)
- 2.1.15 [为何某些区域的 LoRaWAN 协议下, 在 OTAA 入网时, 都只使用某些频点\(频率/信道\)?](#)

2.2 MCU Q&A

- 2.2.1 [在预设状况下, Data2Flash\(\) 已先占用哪些数据存储器区块?](#)
- 2.2.2 [如何去清除整个数据存储器?](#)

2.3 The other Q&A

2.3.1 [适合在 OS 下执行吗? 如移植到 freeRTOS 或其他 OS/RTOS?](#)

2.3.2 [如何除错?](#)

2.3.3 [Run-time variables monitoring tool](#)

3. GPS of S7xG SDK1 and S7xG SDK2

3.1 GPS Q&A

3.1.1 [状态描述](#)

3.1.2 [GPS 1PPS output](#)

3.1.3 [Level Shifter OE pin control](#)

3.1.4 [GPS Active Low Power Mode](#)

3.1.5 [GPS Sleep](#)

3.1.6 [New GPS NMEA Parser](#)

3.1.7 [GPS NMEA Log Output to UART1 \(只支援在 S7xG SDK1\)](#)

1. S76G/S78G SDK1

1.1 LoRaWAN Q&A

1.1.1 支持的区域

REGION_AS923 : 亚洲 923MHz(关于更细部细节请根据各区域/国家法规)

REGION_AU915 : 澳洲 915~928MHz

REGION_CN470 : 中国 470~510MHz

REGION_CN779 : 中国 779~787MHz

REGION_EU433 : 欧洲 433~434MHz

REGION_EU868 : 欧洲 863~870MHz

REGION_IN865 : 印度 865~867MHz

REGION_KR920 : 韩国 920~923MHz

REGION_US915 : 美洲 902~928MHz

REGION_US915_HYBRID : 美洲 902~928MHz HYBRID

1.1.2 Device EUI, Application EUI, Application KEY for OTAA join.

在 "Commissioning.h" 档案, 设定以下宏定义 LORAWAN_DEVICE_EUI、
LORAWAN_APPLICATION_EUI、LORAWAN_APPLICATION_KEY。

1.1.3 Network Security KEY, Application Security KEY, Device Address for ABP join.

在 "Commissioning.h" 档案, 设定以下宏定义 LORAWAN_DEVICE_ADDRESS、
LORAWAN_NWKSKEY、LORAWAN_APPSKEY。

1.1.4 S7xG SDK1当中的 LoRaMac mac 版本为多少?

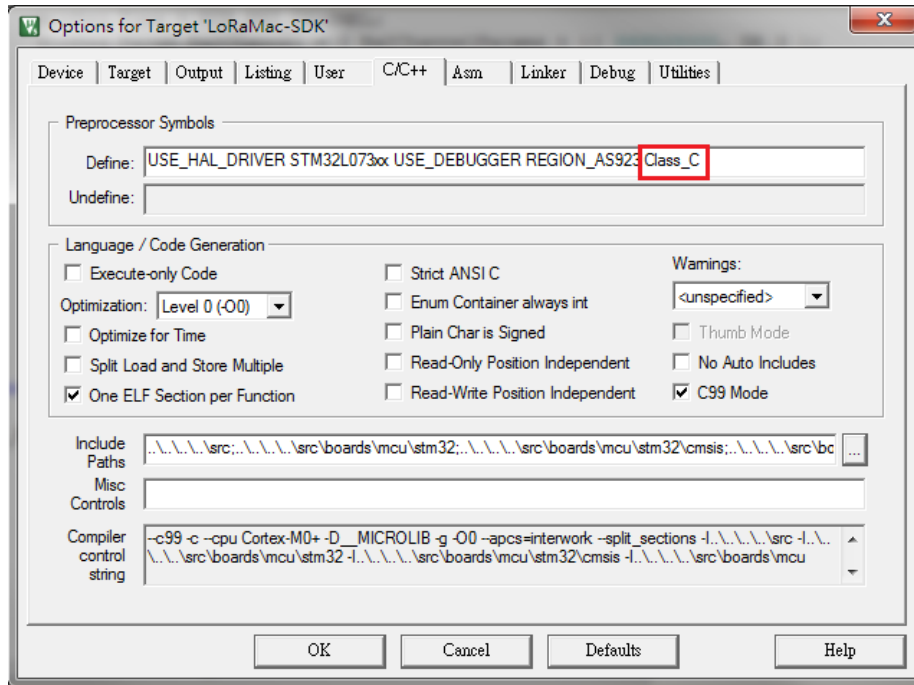
LoRaMac mac 版本为 GitHub Lora-net/LoRaMac-node v4.4.1。

1.1.5 S7xG SDK1 中的 LoRaWAN stack 功能实现依据为?

LoRaWAN stack 功能实现完全依照 "LoRaWAN Regional Parameters v1.0.2rB" 文件当中各个区域的参数定义。Class A 与 Class C 的终端功能实现也完整符合 "LoRaWAN specification 1.0.2" 文件要求。

1.1.6 在 LoRaWAN 终端装置上如何编译成 Class C?

预设为 Class A，若需定义成 Class C，请在 Define 字段输入关键词 “Class_C”，如下图所示。



1.1.7 入网方法是 OTAA 或 ABP?

假如入网方法是 OTAA，将宏定义 “OVER_THE_AIR_ACTIVATION” 设定为 1。

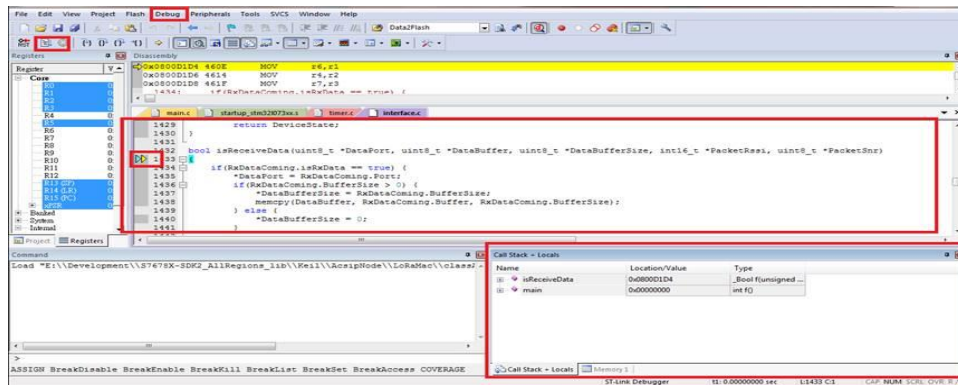
假如入网方法是 ABP，将宏定义 “OVER_THE_AIR_ACTIVATION” 设定为 0。

1.2 The other Q&A

1.2.1 如何除错?

在 Keil 软件开发工具之下，进入 Debug 模式去执行程序，当死机发生时，按下” STOP” 键，停止程序执行，从程序代码区块中的箭头，可了解目前MCU的执行是停在什么地方；另外从” Call Stack - Locals” 区块可知，是之前从哪些函式(或程序)层层呼叫(跳转)执行到此(既是可回溯到上层)。

如此可以知道是在哪个程序点位造成死机，并从此处推导可能发生的细节。如下范例图所示。



1.2.2 Run-time variables monitoring tool

Tool Name : STMicroelectronics STM Studio.

Demo : <https://www.youtube.com/watch?v=eTp-oUD7YwA>

Tutorial : <https://www.youtube.com/watch?v=UzvIXeRCZw0>

2. S76G/S78G SDK2

2.1 LoRaWAN Q&A

2.1.1 支持的区域

AS923-Lib：亚洲 923MHz(关于更细部细节请根据各区域/国家法规)

AU915-Lib：澳洲 915~928MHz

CN470-Lib：中国 470~510MHz

CN779-Lib：中国 779~787MHz

EU433-Lib：欧洲 433~434MHz

EU868-Lib：欧洲 863~870MHz

IN865-Lib：印度 865~867MHz

KR920-Lib：韩国 920~923MHz

US915-Lib：美洲 902~928MHz

US915_HYBRID-Lib：美洲 902~928MHz HYBRID

2.1.2 Device EUI, Application EUI, Application KEY for OTAA join.

在“main.c”档案，“LoRaWAN_t LoRaWAN_Set”结构里，设定以下变量 OTAA.DevEUI、OTAA.AppEUI、OTAA.AppKey。

2.1.3 Network Security KEY, Application Security KEY, Device Address for ABP join.

在“main.c”档案，“LoRaWAN_t LoRaWAN_Set”结构里，设定以下变量 Session.NwkSKey、Session.AppSKey、Session.DevAdd。

2.1.4 S7xG SDK2当中的 LoRaMac mac 版本为多少?

LoRaMac mac 版本为 GitHub Lora-net/LoRaMac-node v4.4.0.

2.1.5 S7xG SDK2 中的 LoRaWAN stack 功能实现依据为?

LoRaWAN stack 功能实现完全依照 "LoRaWAN Regional Parameters v1.0.2rB" 文件当中各个区域的参数定义。Class A 与 Class C 的终端功能实现也完整符合 "LoRaWAN specification 1.0.2" 文件要求。

2.1.6 在 LoRaWAN 终端装置上如何编译成 Class C?

在 “main.c” 档案, “LoRaWAN_t LoRaWAN_Set” 结构里, 设定变量 Node_Class 为 “CLASS_C” 即可。

2.1.7 入网方法是 OTAA 或 ABP?

入网方式为 OTAA, 则在 “main.c” 档案, “LoRaWAN_t LoRaWAN_Set” 结构里, 设定变量 Join_Method 为 “otaa”。

并在之后的入网函数调用以下函式格式: `mac_Join(LoRaWAN_Set.Join_Method, LoRaWAN_Set.OTAA.DevEUI, LoRaWAN_Set.OTAA.AppEUI, LoRaWAN_Set.OTAA.AppKey, NULL, LoRaWAN_Set.Session.Network_ID)`。

入网方式为 ABP, 则在 “main.c” 档案, “LoRaWAN_t LoRaWAN_Set” 结构里, 设定变量 Join_Method 为 “abp”

并在之后的入网函数调用以下函式格式: `mac_Join(LoRaWAN_Set.Join_Method, LoRaWAN_Set.Session.NwkSKey, LoRaWAN_Set.Session.AppSKey, NULL, LoRaWAN_Set.Session.DevAddr, LoRaWAN_Set.Session.Network_ID)`;

2.1.8 Lib 的路径?

“S7678X-SDK2_AllRegions_lib\src\lib”。

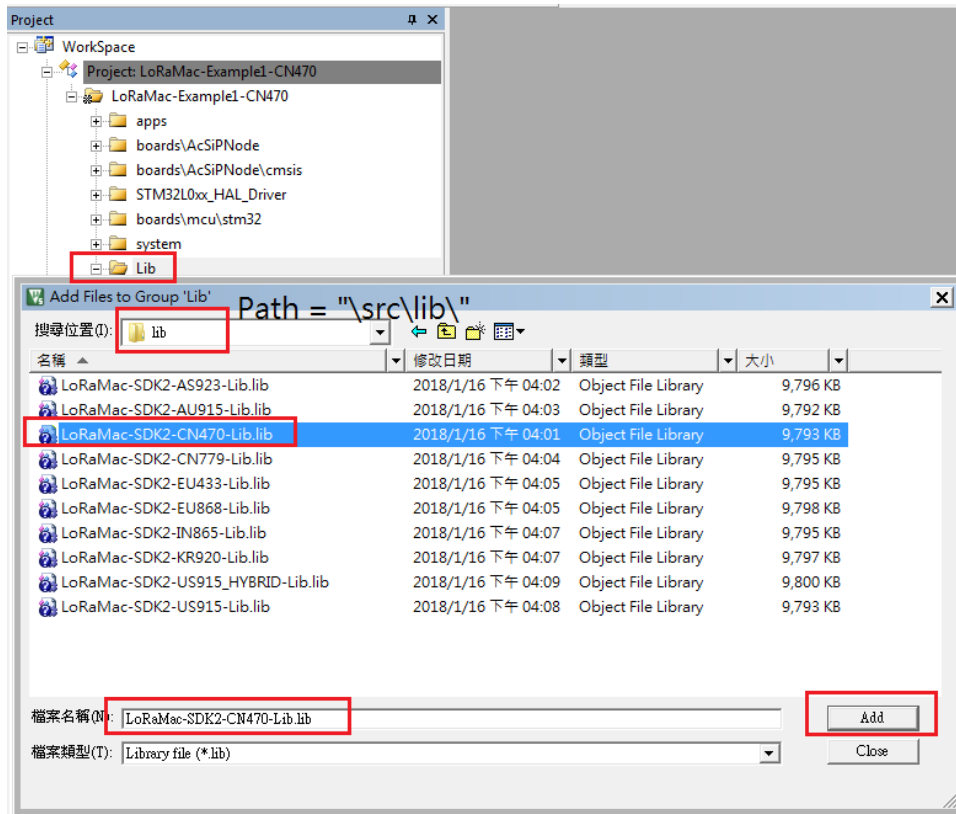
2.1.9 Lib 有哪些函式?

请参考以下档案 ‘lorawan_lib.h’。

2.1.10 如何在范例里更换至其他区域的 Lib?

- 在项目中的 ‘Lib’ 里, 将目前的 lib 移除。
- 点选右键, 选择加入已存在的档案。
- 之后到此路径 “\src\lib\”, 选择欲加入的 lib。
- 如下图所示。

欲编译成什么区域的 LoRaWAN, 就选择那区域的 lib。之后再细部修改主程序的行为, 以符合各别区域的 LoRaWAN 行为差异。



2.1.11 执行 mac_Tx() 函式后，打印显示 “>> busy”

当显示打印 “>> busy” 字样，表示 LoRaMac 在忙碌状态，可能是在接收模式或是传送模式，请等待直到其处理完成。

2.1.12 传送一个封包后，为何之后无法立刻再传送?须等多久才可再传送?

关于这部分需视使用的 LoRaWAN 区域而定，之后再考虑如 duty cycle, dwell, LBT, data length, DR 等设定值，此都会影响到等待时间的长短。

2.1.13 为何 LoRaWAN 自动重传机制下，当自动重传时，每次频点(频率/信道)都不一样?

针对各区域或各国的法规，如 CE、FCC，以及本身 LoRaWAN 的要求，其发射时都要符合随机精神，故就算是重传也必须遵从此，所以重传时的频点(信道频率)不一样是很正常。

2.1.14 SDK2 里，可否增加 RX1 频点(频率/信道)的设定功能？

LoRaWAN 里面针对 RX1 的信道频率，有二种做法，收发同频与收发异频，这是视 REGION 而定。收发同频，指的是 RX1 信道频率等于 TX 的信道频率。收发异频则如此名，就是 RX1 信道频率不等于 TX 的信道频率。在收发异频里，中间有一个算法，TX 的通道值会去对映到其 RX1 的信道频率。

TX 通道的启用，其对映到的 RX1 才会被使用到，所以当未启用的 TX 通道，其对映的 RX1 也不会被执行到。

故添加此一 RX1 设定指令是无意义的，因为容易造成违反 LoRaWAN 协议，而无法与 LoRaWAN Gateway 沟通，且未启用的 TX 通道，就算去设定其 RX1 通道，也不会被执行到。故若客户有此强烈需求，建议使用 S7xG SDK1 进行开发，因此为全开源。

2.1.15 为何某些区域的 LoRaWAN 协议下，在 OTAA 入网时，都只使用某些频点(频率/信道)？

在 LoRaWAN 某些 Region 里，OTAA join 预设只有某些频点(频率/信道)才可以用来入网，例如 EU868，在 JOIN 过程时，LoRaWAN 的预设就只会用到通道 0、1、2，其他额外设定加进去的通道，只会在后续一般的 UpLinks 与 DownLinks 使用。

2.2 MCU Q&A

2.2.1 在预设状况下，Data2Flash() 已先占用哪些数据存储器区块？

占用 0x08080000 ~ 0x08080BFF。

开放给客户使用的数据存储器区块：0x08080C00 ~ 0x080817FF。

2.2.2 如何去清除整个数据存储器？

在档案 “eeprom-board.c”，EraseAllDataEEPROM() 函式用来清除整个数据存储器区块。

2.3 The other Q&A

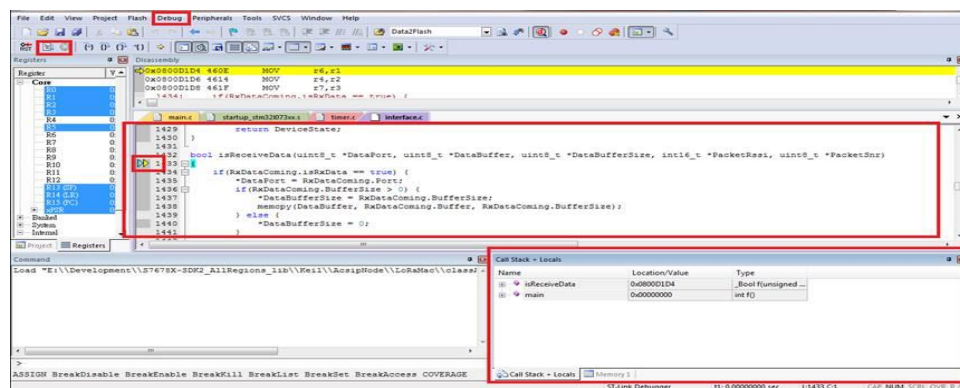
2.3.1 适合在 OS 下执行吗？如移植到 freeRTOS 或其他 OS/RTOS？

在此所有已开发的 lib 都不适合在 OS/RTOS 下执行，因在开发当下，都未作 OS 端的考虑与优化，故不建议在 OS/RTOS 里执行或移植，避免无法预期的状况发生。

2.3.2 如何除错？

在 Keil 软件开发工具之下，进入 Debug 模式去执行程序，当死机发生时，按下” STOP”键，停止程序执行，从程序代码区块中的箭头，可了解目前MCU的执行是停在什么地方；另外从” Call Stack - Locals” 区块可知，是之前从哪些函式(或程序)层层呼叫(跳转)执行到此(既是可回溯到上层)。

如此可以知道是在哪个程序点位造成死机，并从此处推导可能发生的细节。如下范例图所示。



2.3.3 Run-time variables monitoring tool

Tool Name : STMicroelectronics STM Studio.

Demo : <https://www.youtube.com/watch?v=eTp-oUD7YwA>

Tutorial : <https://www.youtube.com/watch?v=UzviXERCZw0>

3. GPS of S7xG SDK1 and S7xG SDK2

3.1 GPS Q&A

3.1.1 状态描述

State	CXD5603GF				
	GNSS	CPU	Always-on block	Backup RAM	Main RAM
S0: Exec	Operation	Operation	Operation	Hold	Hold
S1: Idle	Standby	Operation	Operation	Hold	Hold
S2: Sleep0	Power-off	Power-off	Operation	Hold	Hold
S3: Sleep1	Power-off	Power-off	Operation	Hold	Power-off
S4: Sleep2	Power-off	Power-off	Operation	Power-off	Power-off

S0: Exec(执行状态)

GNSS 定位执行中。

S1: Idle(空置状态)

此为指令等待阶段，系统等待指令的到来，此状态的功率消耗小于执行状态。

S2: Sleep0(睡眠状态 0)

CXD5603GF 会将程序代码、日期时间、卫星数据等保存，但关闭其它逻辑电路的供电。

CXD5603GF 从这状态醒来时，不需要从外部内存或 MCU 加载数据。

S3: Sleep1(睡眠状态 1)

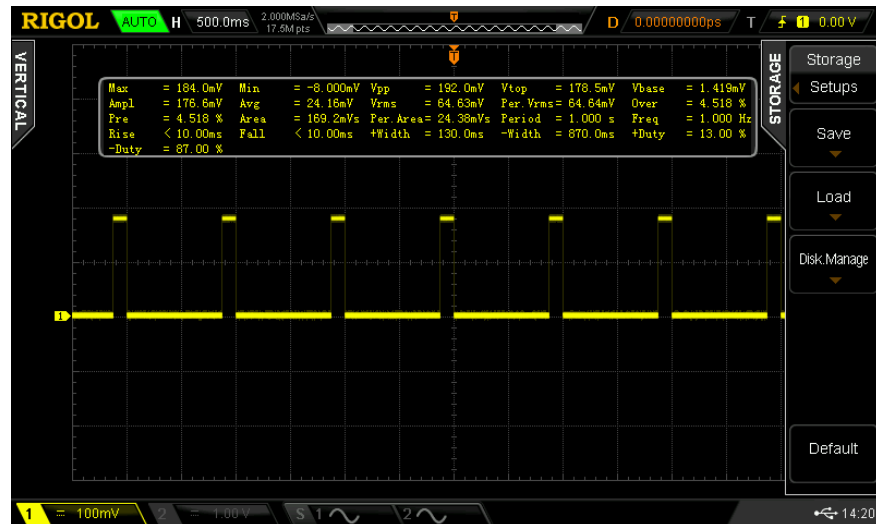
CXD5603GF 在这状态只会保存卫星数据，所以从这状态醒来时，必须另外从外部内存或 MCU 加载程序数据等，但因有保存卫星数据，所以醒来时可以利用 hot start 定位。

S4: Sleep2(睡眠状态 2)

在此状态下，CXD5603GF 只会将内部的 PMU 与 always-on block 保持在上电状况，其余的部分都关闭(不上电)。

3.1.2 GPS 1PPS output

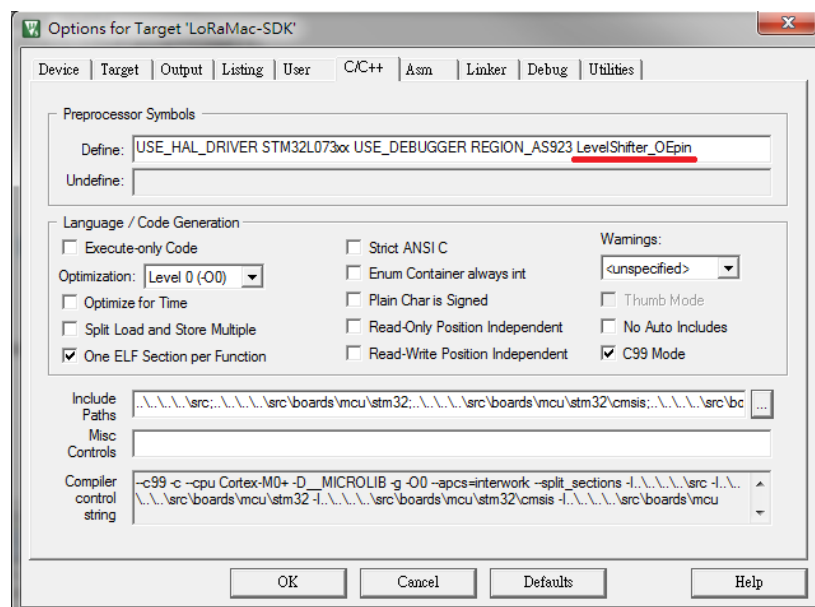
GPS 1PPS 输出功能是致能状态时，当 GNSS 从卫星接收到有效的 UTC 信息后，1PPS 输出接脚会输出频率为 1Hz 方波。当 GPS 1PPS 输出功能是去能状态时，1PPS 输出接脚不会输出任何讯号。



3.1.3 Level Shifter OE pin control

假如 S76G 或 S78G 的电路板本为 v2，则必须去控制 Level Shifter 致能/去能接脚，电路版本若为 v1，不需要做此控制。

所以当 S76G 或 S78G 的电路版本为 v2，必须在“Define”字段，输入字符串“LevelShifter_OEpin”，之后按下“OK”按钮储存设定，并重新编译整个项目，设定如下图所示。



3.1.4 GPS Active Low Power Mode

此功能的默认状态是关闭，若欲启用此功能，请开启文件“gps_driver.h”，并且将宏定义“GPS_ActiveLowPowerControlMode”设定为 1，如下图所示。

```

gps_driver.h
16 #include "board.h"
17 #include "uart-board.h"
18
19 /*
20  * Enable GPS active low power control mode or Not
21  */
22 #define GPS_ActiveLowPowerControlMode 1 // 1:Enable, 0:Disable
23 #define GPS_ActiveLowPowerMode_DUTYCYCLE 60000 //value in [ms].

```

宏定义“GPS_ActiveLowPowerMode_DUTYCYCLE”指的是定位循环的周期，单位为 ms。

若欲了解更多细节，因版权与专利关系，请参考文件“SONY CXD5603GF Intro.pdf”中的章节“Operation modes”。

3.1.5 GPS Sleep

此功能的默认状态是关闭，若欲启用此功能，请开启文件“board.h”，并将宏定义“ENABLE_POWER_SAVING”设定为 1，并进一步再将欲启用的睡眠状态设定为 1，如下图所示。

初步可以参考此份文件的“3.1.1 状态描述”，若欲了解更多细节，因版权与专利关系，请参考文件“SONY CXD5603GF Intro.pdf”中的章节“Sleep”。

```

board.h
122 /*
123  * Enable Power Saving Demo or Not
124  */
125 #define ENABLE_POWER_SAVING 1 // 1:Enable, 0:Disable
126 #define POWER_SAVING_GPS_Sleep0 1 // 1:Enable, 0:Disable
127 #define POWER_SAVING_GPS_Sleep1 0 // 1:Enable, 0:Disable
128 #define POWER_SAVING_GPS_Sleep2 0 // 1:Enable, 0:Disable
129 /*
130  * Set Power Saving Interval Time
131  */
132 #define POWER_SAVING_INTERVAL 30 // second

```

3.1.6 New GPS NMEA Parser

在版本 S7xG SDK1 v0.0.5 与 S7xG SDK2 AR.1(或这些版本之后)，增加了新的 GPS NMEA 字符串解析器，默认运作是在新的解析器，若客户是在这些版本之前开发的，因当时是使用旧版本的 GPS NMEA 字符串解析器，若欲执行在旧版本的 GPS NMEA 字符串解析器，则设定如下图所示。

```

gps_driver.h
19 /*
20  * Enable GPS new parser or Not
21  */
22 #define GPS_NEW_PARSER 0
23

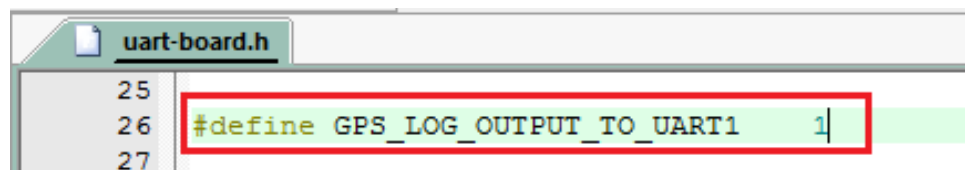
```

3.1.7 GPS NMEA Log Output to UART1 (只支援在 S7xG SDK1)

此功能默认是关闭的，若欲开启此功能，请开启文件“uart-board.h”，并将宏定义“GPS_LOG_OUTPUT_TO_UART1”设定为 1，如下图所示。

此功能开启后，MCU STM32L073 只是单纯的将从 GPS 收到的所有字符串转输出至 UART1，且 MCU STM32L073 不会去做其他工作，例如不会去做 LoRaWAN 的任何功能，这是为了避免漏接或漏发 GPS 的字符串。

此功能开发出来只是为了验证测试 GPS 用，S7xG SDK2 不支持此功能，此功能只支持在 S7xG SDK1。



```
25
26 #define GPS_LOG_OUTPUT_TO_UART1 1
27
```