S76G/S78G

SDK1 and SDK2 Q&A

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

Index

1. S76G/S78G SDK1
   1.1 LoRaWAN Q&A
      1.1.1 Support Regions
      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 What is the LoRaMac code base in S7xG SDK1?
      1.1.5 What the LoRaWAN stack implements in S7xG SDK1?
      1.1.6 How to define the class C in the LoRaWAN Node?
      1.1.7 Join by OTAA or ABP?
   1.2 The other Q&A
      1.2.1 How to debug?
      1.2.2 Run-time variables monitoring tool
2. S76G/S78G SDK2
   2.1 LoRaWAN Q&A
      2.1.1 Support Regions
      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 What is the LoRaMac code base in S7xG SDK2?
      2.1.5 What the LoRaWAN stack implements in S7xG SDK2?
      2.1.6 How to define the class C in the LoRaWAN Node?
      2.1.7 Join by OTAA or ABP?
      2.1.8 Where the lib(path)?
      2.1.9 Lib functions?
      2.1.10 How to change the region lib in example?
      2.1.11 Call function mac_Tx() after print the Tx status ">> busy"
      2.1.12 We can send the data packets by mac_Tx(), but after cannot, then how long we have to wait till we can?
   2.2 MCU Q&A
      2.2.1 In the default, the function Data2Flash(), what the area occupied in data memory?
      2.2.2 How to erase the data memory?
   2.3 The other Q&A
      2.3.1 Is the Lib for OS, example porting the lib in freeRTOS or the other OS/RTOS?
      2.3.2 How to debug?
      2.3.3 Run-time variables monitoring tool

3. GPS of S7xG SDK1 and S7xG SDK2
3.1 GPS Q&A
3.1.1 State Description
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(Just support in S7xG SDK1)

1. S76G/S78G SDK1

1.1 LoRaWAN Q&A

1.1.1 Support Regions

REGION_AS923 : For Asia 923MHz(about the detail to reference the country rule)

REGION_AU915 : For Australia 915~928MHz

REGION_CN470 : For China 470~510MHz

REGION_CN779 : For China 779~787MHz

REGION_EU433 : For Europe 433~434MHz

REGION_EU868 : For Europe 863~870MHz

REGION_IN865 : For India 865~867MHz

REGION_KR920 : For South Korea 920~923MHz

REGION_US915 : For United States 902~928MHz

REGION_US915_HYBRID : For United States 902~928MHz HYBRID

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

In “Commissioning.h” file, setting the define LORAWAN_DEVICE_EUI, LORAWAN_APPLICATION_EUI, LORAWAN_APPLICATION_KEY.

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

In “Commissioning.h” file, setting the define LORAWAN_DEVICE_ADDRESS, LORAWAN_NWKSKEY, LORAWAN_APPSKEY.

1.1.4 What is the LoRaMac code base in S7xG SDK1?

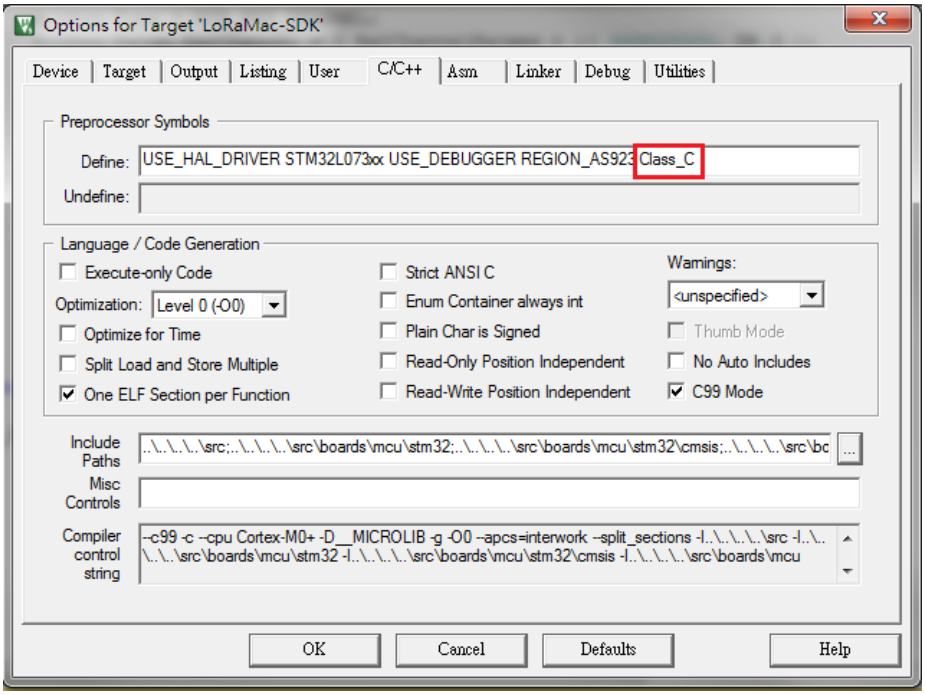
The LoRaMac code base is GitHub Lora-net/LoRaMac-node v4.4.1.

1.1.5 What the LoRaWAN stack implements in S7xG SDK1?

LoRaWAN stack implements all regions defined in "LoRaWAN Regional Parameters v1.0.2rB" document. Class A and Class C endpoint implementation is fully compatible with "LoRaWAN specification 1.0.2".

1.1.6 How to define the class C in the LoRaWAN Node?

Default is Class A, if need setting in class C, input the key word “Class_C” in Define field. As shown in the following figure.



1.1.7 Join by OTAA or ABP?

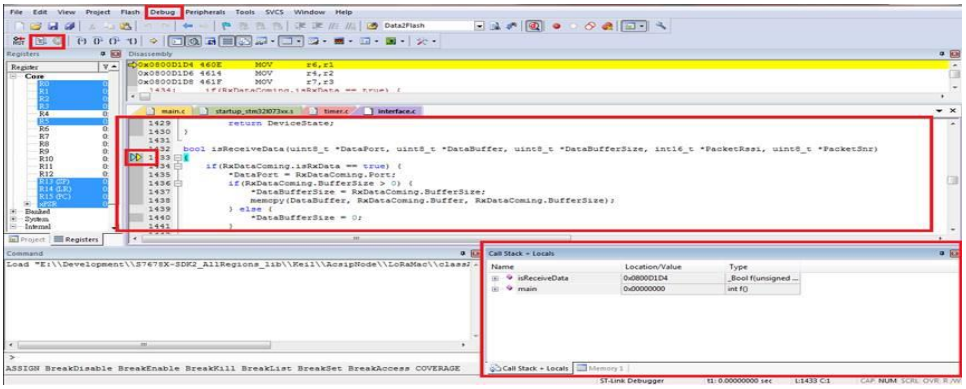
If join by OTAA, setting the define “OVER_THE_AIR_ACTIVATION” is 1.

If join by ABP, setting the define “OVER_THE_AIR_ACTIVATION” is 0.

1.2 The other Q&A

1.2.1 How to debug?

Running the MDK Keil IDE tool, build and download, into “Debug” mode, and run the application, if halt after, stop the application, see the “Call Stack - Locals” window, analyze the functions and flow. After can understand that why to halt or crash. As shown in the following figure.



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 Support Regions

AS923-Lib : For Asia 923MHz(about the detail to reference the country rule)

AU915-Lib : For Australia 915~928MHz

CN470-Lib : For China 470~510MHz

CN779-Lib : For China 779~787MHz

EU433-Lib : For Europe 433~434MHz

EU868-Lib : For Europe 863~870MHz

IN865-Lib : For India 865~867MHz

KR920-Lib : For South Korea 920~923MHz

US915-Lib : For United States 902~928MHz

US915_HYBRID-Lib : For United States 902~928MHz HYBRID

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

In “main.c” file, in the “LoRaWAN_t LoRaWAN_Set” structure, setting the OTAA.DevEUI, OTAA.AppEUI, OTAA.AppKey.

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

In “main.c” file, in the “LoRaWAN_t LoRaWAN_Set” structure, setting the Session.NwkSKey, Session.AppSKey, Session.DevAddr.

2.1.4 What is the LoRaMac code base in S7xG SDK2?

The LoRaMac code base is GitHub Lora-net/LoRaMac-node v4.4.0.

2.1.5 What the LoRaWAN stack implements in S7xG SDK2?

This LoRaWAN stack implements all regions defined in "LoRaWAN Regional Parameters v1.0.2rB" document. Class A and Class C endpoint implementation is fully compatible with "LoRaWAN specification 1.0.2".

2.1.6 How to define the class C in the LoRaWAN Node?

In “main.c” file, in the “LoRaWAN_t LoRaWAN_Set” structure, setting the Node_Class is "CLASS_C".

2.1.7 Join by OTAA or ABP?

Join by OTAA, in “main.c” file, in the “LoRaWAN_t LoRaWAN_Set” structure, setting the Join_Method is "otaa".

And after to call the function: mac_Join(LoRaWAN_Set.Join_Method, LoRaWAN_Set.OTAA.DevEUI, LoRaWAN_Set.OTAA.AppEUI, LoRaWAN_Set.OTAA.AppKey, NULL, LoRaWAN_Set.Session.Network_ID).

Join by ABP, in “main.c” file, in the “LoRaWAN_t LoRaWAN_Set” structure, setting the Join_Method is "abp".

And after to call the function: 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 Where the lib(path)?

in “S7678X-SDK2_AllRegions_lib\src\lib”.

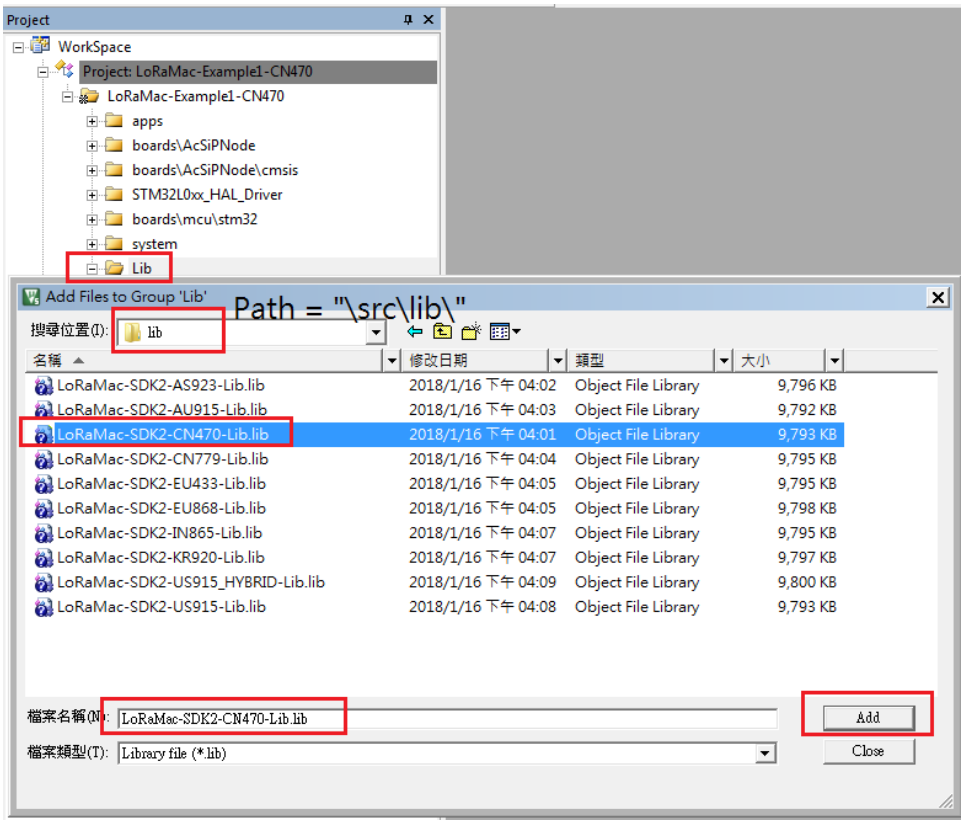
2.1.9 Lib functions?

Please reference the file ‘lorawan_lib.h’.

2.1.10 How to change the region lib in example?

- Remove existing lib in group ‘Lib’ in project.
- Add existing file to group ‘Lib’ in project.
- Go to the path “\src\lib\” to select and add right region lib.
- As shown in the following figure.

What region are you want or build? A : Select the region lib.



2.1.11 Call function mac_Tx() after print the Tx status “>> busy”

When print the status “>> busy”, meaning the LoRaMac in busy status.

Maybe the LoRaMac in Rx mode or Tx mode, please waiting until completely dealt with.

2.1.12 We can send the data packets by mac_Tx(), but after cannot, then how long we have to wait till we can?

About this, need to see what the region, what the setting. Like duty cycle, dwell, LBT, data length, DR etc. How long time need to wait, All about the setting.

2.2 MCU Q&A

2.2.1 In the default, the function Data2Flash(), what the area occupied in data memory?

Occupied the 0x08080000 ~ 0x08080BFF.

0x08080C00 ~ 0x080817FF for free to use.

2.2.2 How to erase the data memory?

In the file “eeprom-board.c”, have the function EraseAllDataEEPROM().

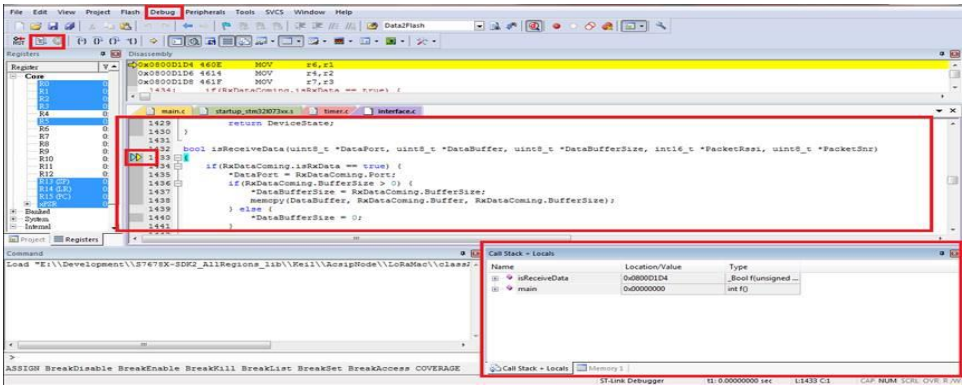
2.3 The other Q&A

2.3.1 Is the Lib for OS, example porting the lib in freeRTOS or the other OS/RTOS?

All the developed libraries have not researched the OS/RTOS so that not support the OS/RTOS.

2.3.2 How to debug?

Running the MDK Keil IDE tool, build and download, into “Debug” mode, and run the application, if halt after, stop the application, see the “Call Stack - Locals” window, analyze the functions and flow. After can understand that why to halt or crash. As shown in the following figure.



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 State Description

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

GNSS positioning can be performed.

S1: Idle

This is a command waiting state. The system can accept commands but power consumption is managed to be low.

S2: Sleep0

The CXD5603GF holds program code, data and satellite date but other logic circuit is powered off. The CXD5603GF can wake up from this state without loading the data from an external FLASH memory or the system MCU.

S3: Sleep1

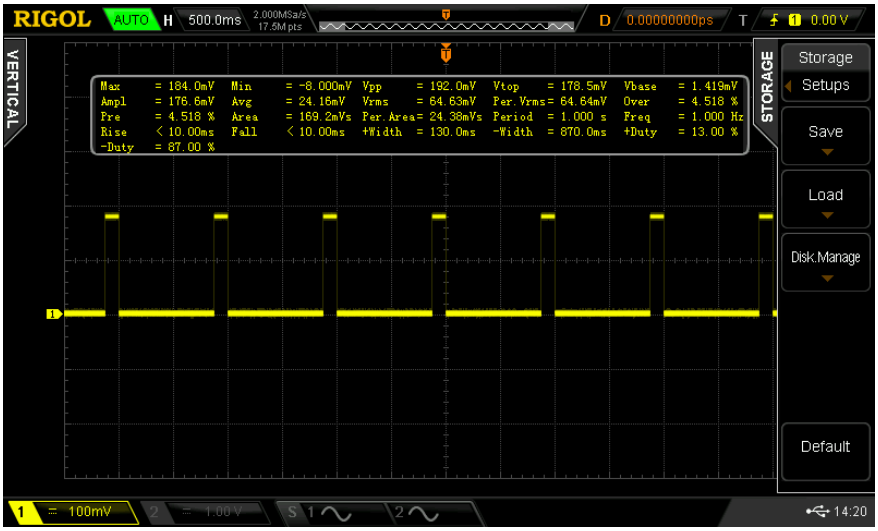
Because the CXD5603GF holds satellite data only in this state, it must load program data from an external FLASH memory or the system MCU for wake-up but it can get a position with hot start.

S4: Sleep2

In this state, the CXD5603GF is powered off except an internal PMU and always-on block.

3.1.2 GPS 1PPS output

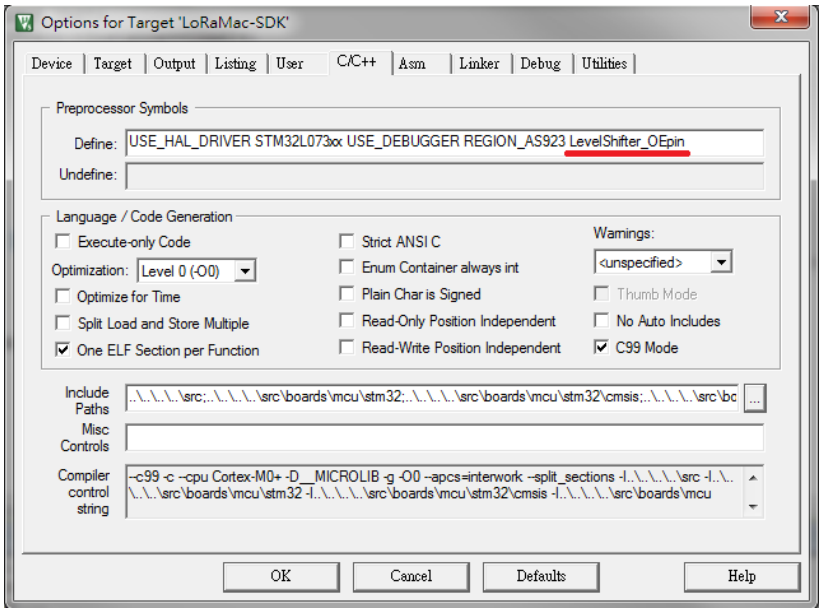
When GPS 1PPS output is enabled, timing pulse is output in 1 sec period from 1PPS output pin after clock information being received from GNSS. When 1PPS output is disabled, timing pulse is not output from 1PPS output pin.



3.1.3 Level Shifter OE pin control

If the S76G or S78G the circuit version is v2. Need to control the Level Shifter OE pin. Circuit version is v1 : don't need to control the Level Shifter OE pin.

So if S76G or S78G the circuit version is v2, need to do this: input the define string "LevelShifter_OEpin", then setting like this, After press "OK" button, and build the Project.



3.1.4 GPS Active Low Power Mode

Disable by default. If want to enable, please to open the file "gps_driver.h", and setting the key word "GPS_ActiveLowPowerControlMode" is "1". Then setting like this,

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].
```

"GPS_ActiveLowPowerMode_DUTYCYCLE" is positioning cycle.

Please refer to the attached file : "SONY CXD5603GF Intro.pdf", chapter "Operation modes".

3.1.5 GPS Sleep

Disable by default. If want to enable, please to open the file "board.h", and setting the key word "ENABLE_POWER_SAVING" is "1".And what the GPS sleep level are you want to choice, please to setting to "1", default is "level 0".

Please refer to the attached file : "SONY CXD5603GF Intro.pdf", chapter "Sleep". Then setting like this.

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

This time (=S7xG SDK1 v0.0.5, S7xG SDK2 AR.1 and after) added new GPS NMEA parser. Default is new parser. If want try in old parser, please to open the file "gps_driver.h", setting the "GPS_NEW_PARSER" is "0".

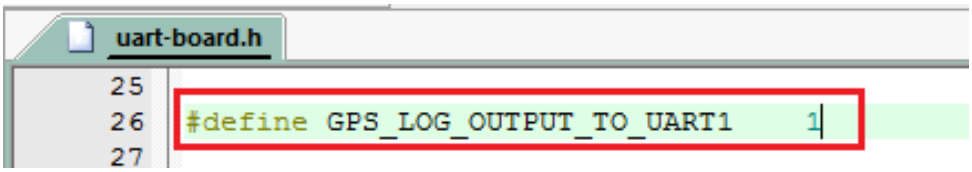
gps_driver.h

```
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 (Just support in S7xG SDK1)

Disable by default. If want to enable, please to open the file "uart-board.h", and setting the key word "GPS_LOG_OUTPUT_TO_UART1" is "1".

Not support in S7xG SDK2, just support in S7xG SDK1.


uart-board.h
25
26
27
#define GPS_LOG_OUTPUT_TO_UART1 1