Differences with v4.4.0 of the Semtech Stack

RM1xx

Application Note *v1.1*

INTRODUCTION

This document describes the differences between v4.4.0 of the Semtech stack and previous releases and how these differences could affect the performance of the RM1xx modules with the following version numbers: 101.6.1.0, 111.6.1.0, 102.6.1.0, 112.6.1.0, and later. The previous releases are those with the old numbering system of 17.x.y.z or 18.x.y.z.

These changes generally affect only modules with 72 enabled channels which can be randomly selected to transmit a JoinRequest or a packet of data to the gateway. Currently, this only includes the US and AU regions. The changes fall under two closely-related areas, Channel enabling and JoinRequests.

The EU modules behave in a completely different way; this document does not apply to them.

JOINREQUESTS

With previous firmware versions, a JoinRequest was continually transmitted by the module on one of the 500 kHz channels until the gateway/server responded with a JoinAccept. Because the 500 kHz channels have less range than most of the possible 125 kHz channel configurations, transmitting on these channels could have affected performance; the module would have to be closed to initially Join.

Other potential down-sides were:

- It could potentially drain the battery because the transmit process wouldn't stop even if there was no chance of a response.
- Because the stack kept sending the same packet, the DevNonce never changed. In an extreme case, if the module didn't receive the JoinAccept from the gateway, it would only re-send the JoinRequest. When the network server received the module's re-transmitted JoinRequest with the same DevNonce, it would reject it. The module would then continue to send JoinRequests. This process would repeat indefinitely (or until the module was rebooted, forcing a new JoinRequest to be transmitted).

This issue is fixed in the new version of the stack. The stack now switches between random 125 kHz and 500 kHz channels as defined in the LoRaWAN spec. Also, the stack transmits the same JoinRequest a configured amount of times before sending an RxTimeout event. At that point, the *smart*BASIC application determines the next step.

The default number of attempts in the US and AU modules is two after which the stack throws an RxTimeout. Figure 1 shows what to expect if you send a new JoinRequest on receipt of the RxTimeout event.

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

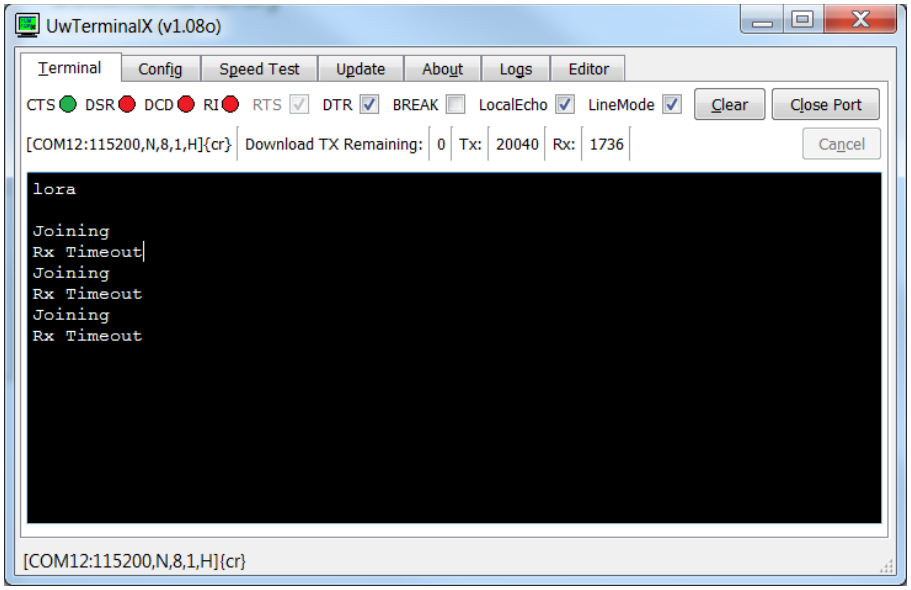

Figure 1: New JoinRequest sent after a RxTimeout

This solution has the advantage of getting around the two problems previously discussed; however, if all 72 channels are enabled, it has a major disadvantage of reducing the chances of the stack selecting a supported channel on which to transmit.

On the first attempt to send a JoinRequest, on the 500 kHz channel, there is a 1 in 8 chance of randomly selecting the 500 kHz channel supported by a gateway. The stack logs the attempted channel and does not retry on that channel. On the second attempt, on a 125 kHz channel, there is again a 1 in 8 chance of selecting a successful channel (the odds slowly improve). If the module continues to resend the JoinRequest, the odds gradually improve.

However, when the module receives the RxTimeout and sends a JoinRequest, the stack clears this history, returning the probability to 1 in 8. Because of this, it could take longer than usual to join a network.

To avoid this, you can configure the number of JoinRequest attempts by calling the following function in their *smart*BASIC app (where ***x*** is the number of attempts prior to an RxTimeout):

```
LORAMACSetOption(27,x)
```

This option is not persisted. It resets to the default value of two after a power cycle. Also, the command must be called *before* the module transmits the first JoinRequest, otherwise it has no effect.

Note: The previously-mentioned issue can only occur if all 72 channels are enabled. If you take advantage of the options described in the *ChannelMap Selection* section of this application note, you should have no problem connecting.

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

CHANNELMAP SELECTION

On boot-up, the new stack originally defaults to all 72 channels enabled, which likely results in reduced performance. Despite this reduced performance, the stack eventually selects an enabled channel to transmit a JoinRequest on.

Once in the Joined state, the network server can utilize the ADR command to modify the ChannelMap. However, with most of the network providers, although started, this hasn't been fully implemented. You will still have enabled channels that are not supported by a gateway. Because most network providers only support one sub-band, there may be a high probability of a packet being transmitted on a non-supported channel, again affecting the performance of the module.

In the previous versions of firmware, you had the option of defining your own ChannelMap through the following commands:

```
at+cfgex 1009
```

or

```
LoramacSetOption(LORAMAC_OPT_CHANNELMASK,"")
```

Although supported by Laird's RG1xx Sentrius gateway, the versatility offered by these commands is not generally supported by most gateways. In addition, it is a bit complicated to use. Because of this, the following functionality has been added:

```
at+cfg 1001 X and
LoramacSetOption(LORAMAC_OPT_SUBBAND,"X")
```

These commands select a specific sub-band, **X**, which has a value of between 1 and 8. As with other commands, **at+cfg** is persisted and outside the scope of the *smart*BASIC application. **LoramacSetOption** is not persisted and is dynamically called from within the app.

Important: Before using these commands, contact your network provider to verify the correct sub-band.

As both **at+cfg** and **at+cfgex** commands are valid, use the following command to define which one the code uses:

```
at+cfg 1002
```

Valid values are shown in Table

Remember, **at+cfg** or **at+cfgex** commands only take effect after a module reboot.

Table 1: Valid at+cfg 1002 values

at+cfg 1002	Action
0 (default)	Stack default – all channels enabled
1	Use at+cfg 1001
2	Use at+cfgex 1009

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610

Values set using the **LoramacSetOption** command are dynamic and take precedence over the configured values for the duration of that power cycle. You do not have to set **at+cfg 1002** for the **LoramacSetOption** to take effect.

Use caution with the ChannelsMap. When a JoinRequest is sent to the stack, the 500-kHz section of the ChannelMap tends to be reset. Even if you enabled sub-band 1, the stack may still enable all 500 kHz channels so the first JoinRequest may fail (because it will be transmitted on a randomly-selected 500 kHz channel) although the second request should be accepted. This occurs after the JoinRequest is loaded so any changes you've made are overwritten.

All network providers should eventually support the ADR message and dynamically modify the ChannelMap to the one that it supports after the JoinAccept message. However, to get to this stage, the JoinRequest must be accepted. For the JoinRequest therefore, it is still a case of waiting for the random selection of the supported channel.

Note that this is how the system is supposed to work and it should only be the JoinRequest that has problems getting through. Once the module has joined and providing the server has transmitted the correct configuration, the module should only transmit on a supported channel. This may be annoying during testing but in real life applications it is something that may have to be accepted.

Revision History

Version	Date	Notes	Document/Changes Initiated By	Approved
1.0	03 July 2017	Initial Release	Colin Anderson	Jonathan Kaye
1.1	25 October 2017	Updated for production release	Colin Anderson	Jonathan Kaye

Americas: +1-800-492-2320
Europe: +44-1628-858-940
Hong Kong: +852 2923 0610