Laird
Smart Technology. Delivered.™

RM1xx LoRa *smart*BASIC Extensions

RM1xx Series

Document version 1.0

REVISION HISTORY

Version	Date	Notes	Approver
1.0	27 Jan 2017	Initial version	Jonathan Kaye

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

CONTENTS

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

1 INTRODUCTION

1.1 Documentation Overview

This RM1xx LoRa Extension Functionality user guide provides detailed information on LoRa-specific *smart*BASIC extensions which provide a high level managed interface to the underlying LoRa device and Bluetooth stack in order to manage the following:

- Joining a LoRa gateway and transmitting/receiving data payload
- Link checking on LoRa connection
- Managing LoRa sleep intervals and reading chipset registers
- Events related to the above

This document deals specifically with the *smart*BASIC APIs relating to the LoRa functionality in the RM1xx series of modules. For other details of programming the RM1xx, see any of the following documents:

- RM1xx LoRaMAC BLE Central Extensions Guide (central BLE functions)
- RM1xx BLE Peripheral Extensions Guide (peripheral BLE functions for the RM186_PE or RM191_PE modules)
- *smart*BASIC Core Reference Guide (common functions across all *smart*BASIC modules)

All the above documents are found in the documentation tab of the RM1xx product page: http://www.lairdtech.com/products/rm1xx-lora-modules

1.2 What Does a LoRa/BLE Module Contain?

Laird's *smart* BASIC-based LoRa/BLE modules are designed to provide a complete wireless processing solution and contain the following:

- A highly integrated radio with an integrated antenna (external antenna options are also available)
- BLE Physical and Link Layer
- Higher level stack
- Multiple GPIO and ADC
- Wired communication interfaces such as UART, I2C, and SPI
- A *smart* BASIC run-time engine
- Program-accessible flash memory which contains a robust flash file system exposing a conventional file system and a database for storing user configuration data
- Voltage regulators and brown-out detectors

For simple end devices, these modules can completely replace an embedded processing system.

The following block diagram (Figure 1) illustrates the structure of the BLE + LoRa *smart* BASIC module from a hardware perspective on the left and a firmware/software perspective on the right.

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

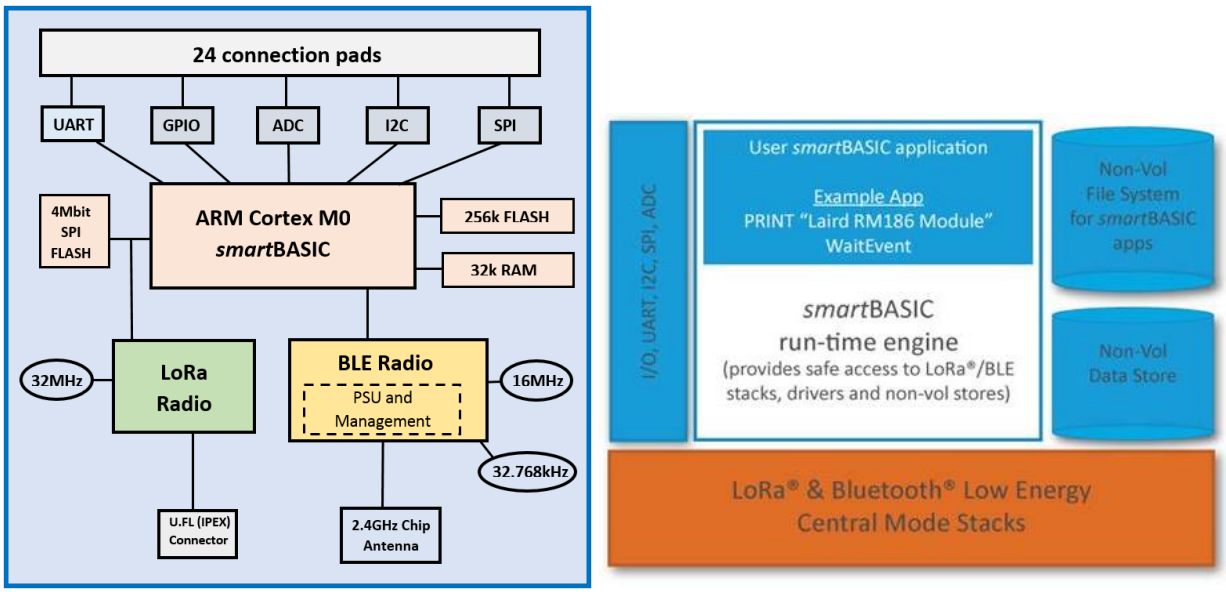

Figure 1: RM1xx smartBASIC module block diagram

2 Interactive Mode Commands

Interactive mode commands allow a host processor or terminal emulator to interrogate and control the operation of a *smart* BASIC-based module. Many of these emulate the functionality of AT commands. Others add extra functionality for controlling the filing system and compilation process.

Syntax Unlike commands for AT modems, a space character must be inserted between AT, the command, and subsequent parameters. This allows the *smart* BASIC tokeniser to efficiently distinguish between AT commands and other tokens or variables starting with the letters **AT**.

```
'Example:
AT I 3
```

The response to every Interactive mode command has the following form:

<linefeed character> response text <carriage return>

This format simplifies the parsing within the host processor. The response may be one or multiple lines. Where more than one line is returned, the last line has one of the following formats:

<lf>00<cr> for a successful outcome, or

<lf>01<tab> hex number <tab> optional verbose explanation <cr> for failure.

Note: In the case of the 01 response, the **<tab>optional_verbose_explanation** is missing in resource constrained platforms like the RM1xx modules. The *verbose explanation* is a constant string and since there are over 1000 error codes, these verbose strings can occupy more than 10 kilobytes of flash memory.

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

The hex number in the response is the error result code consisting of two digits which can be used to help investigate the problem causing the failure. Rather than provide a list of all the error codes in this manual, you can use UWTerminal to obtain a verbose description of an error when it is not provided on a platform.

To get the verbose description, click the BASIC tab (in UWTerminal) and, if the error value is hhhh, enter the command ER 0xhhhh and note the 0x prefix to hhhh. This is illustrated in Figure 2.

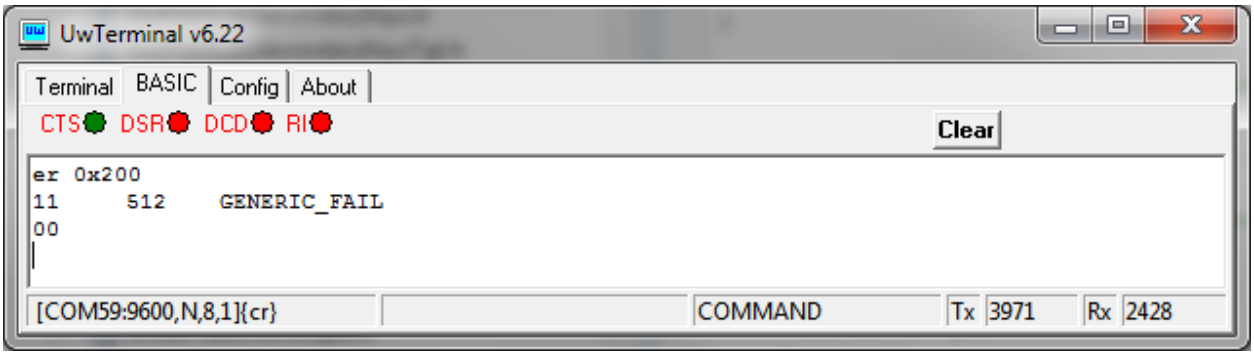

Figure 2: Optional verbose explanation

You can also obtain a verbose description of an error by highlighting the error value, right-clicking, and selecting **Lookup Selected ErrorCode** in the Terminal window.

If you get the text UNKNOWN RESULT CODE 0xHHHH, please contact Laird for the latest version of UWterminal.

2.1.1 AT+CFGEX

COMMAND

AT+CFGEX is used to set a non-volatile configuration key with a string. The syntax of this command is defined in the smartBASIC Core Functionality Manual.

The following configuration key IDs are specific to the RM1xx module.

Key ID	Definition	Notes
1009	ChannelsMask	Sets the ChannelsMask. Only valid for the RM191. See the Setting RM191 ChannelsMask section below
1010	AppEui	Application Identifier – 8 Bytes/16 Hex Characters
1011	DevEui	End Device Identifier – 8 Bytes/16 Hex Characters
1012	AppKey	Application Key – 16 Bytes/32 Hex Characters
1013	NwkSKey	Network Session Key – 16 Bytes/32 Hex Characters
1014	AppSKey	Application Session Key – 16 Bytes/32 Hex Characters
1015	DevAddr	End device Address – 8 character Hex string

The at+cfgex command returns an invalid key error (7312) if an invalid Key id is entered or the length of the entered string is incorrect for that specific Key id.

The new config value is only available for use after a system reset.

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

Note: The NwkSKey, AppSKey and AppKey values are write only. These values cannot be read back using the **at+cfgex xxxx?** command.

Prior to firmware versions 17/18.4.1.0 the Key Ids in the table above were in the range of 1000 – 1005 instead of the new range of 1010-1015.

3 LoRa Extensions Built-in Routines

The following commands are specific to the LoRa functionality of the RM1xx.

3.1.1 LoRaMACSleepMode

FUNCTION

This command places the LoRa chipset in ultra-low power sleep mode. This closes the SPI driver and places the chipset in an unusable state. LORAMACReset must be called prior to re-using the LoRa device.

LORAMACSleepMode ()

Returns	None
Arguments: None	
Interactive Command	No

```
rc = LORAMACSleepMode()
```

LORAMACSleepMode is an extension function.

3.1.2 LORAMACJoin

FUNCTION

This command begins the Join process to connect to a LoRa gateway device. Before actioning the Join the module will revert back to its default state, specifically with respect to the datarata, txpower and, in the case of the RM186, the available frequency channels. All the appropriate network parameters must be configured using the AT+CFGEX commands prior to making this call.

There are two possible Join options:

- OTA – where the RM1xx sends IDs to the server and both ends of the link use these IDs to calculate the NwkSKey and the AppSKey that are used in the encryption and decryption of the subsequent data packets.
- Personalization – where both the RM1xx and server are preconfigured with the keys NwkSKey, AppSKey and the DevAddr

For more information on the Join options, refer to the *Interfacing with LoRaWAN* application notes which are available from the Laird RM1xx product page.

If the OTA option is selected, once the LORAMACJoin command is sent an EVLORAMACJOINING event is thrown.

Then on successful completion of the process the EVLORAMACJOINED event will be thrown.

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

3.1.3 LORAMACJoin (nFlags)

Returns	0x0000 = Successfully started Join process 0xnnnn = resultCode
Arguments:	
nFlags	**ByVal nFlags INTEGER** LORAMAC_JOIN_BY_REQUEST LORAMAC_JOIN_BY_PERSONALIZATION
Interactive Command	No

```
rc = LORAMACJoin(LORAMAC_JOIN_BY_REQUEST);
#define LORAMAC_JOIN_BY_REQUEST          1 // Used with LORAMACJoin
#define LORAMAC_JOIN_BY_PERSONALIZATION     0 // Used with LORAMACJoin

DIM rc

FUNCTION LoramacJoining() As Integer
 print "\nJoining"
endfunc 1

FUNCTION LoramacJoined() As Integer
 print "\nJoined"
endfunc 1


ONEVENT EVLORAMACJOINING CALL LoramacJoining
ONEVENT EVLORAMACJOINED  CALL LoramacJoined


rc = LORAMACJoin(LORAMAC_JOIN_BY_REQUEST)

WAITEVENT
```

LORAMACJoin is an extension function.

3.1.4 LORAMACLinkCheck

FUNCTION

This command sends a link check request upstream to the gateway. When a link check response is received from the gateway device, a EVLORAMACLINKCHECKRESPMSG message is sent to the application. This message contains two 8-bit values. The first represents the gateway's link margin, in dB, of the last successfully received link check request. The second represents the number of gateways that successfully received the last link check request.

LORAMACLinkCheck ()

Returns	0x0000 = Successfully sent a LoRa Link Check request message upstream 0xnnnn = resultCode
Arguments: None	
Interactive Command	No

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

```
//=============================================================================
// This handler is called when a Link Check Response is received
//=============================================================================
FUNCTION HandlerLoRaLinkCheckResponse(BYVAL nMargin AS INTEGER, BYVAL nGwCnt AS
INTEGER) AS INTEGER
      PRINT "Link Check Response: Margin = ";nMargin;"dB    Gateway Count =
";nGwCnt;"\n"
endfunc 1

OnEvent EVLORAMACLINKCHECKRESPMSG call HandlerLoRaLinkCheckResponse
…
LORAMACLinkCheck()
```

LORAMACLinkCheck is an extension function.

3.1.5 LORAMACTxData

FUNCTION

This command sends data upstream to the LoRa gateway on the specified port.

Note: Port 0 is reserved for MAC commands between the gateway and the end node.

LORAMACTxData (nPort, Data$, nFlags)

Returns	0x0000 : Successfully queued data for upstream transmission 0xnnnn : resultCode
Arguments:	
nPort	**ByVal nPort INTEGER** The port to be used by the transmission.
Data$	**ByRef data$ STRING** The data to be sent upstream.
nFlags	**ByVal nFlags INTEGER** Bit mask for options. ▪ 0 – Do not request confirmation ▪ 1 – Request confirmation
Interactive Command	No

```
DIM data$

FUNCTION LoramacRxComplete() As Integer
      print "\nRx sequence completed "
endfunc 1

FUNCTION LoramacTxComplete() As Integer
      print "\nTx sequence completed "
endfunc 1

ONEVENT EVLORAMACTXCOMPLETE      CALL LoramacTxComplete
ONEVENT EVLORAMACRXCOMPLETE      CALL LoramacRxComplete

data$ = "foo"
//Send a confirmed packet upstream to port 1
```


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

```
LORAMACTxData(1, data$, 1)
```

LORAMACTxData is an extension function.

3.1.6 LORAMACRxData

FUNCTION

This function returns downlink data received from the LoRa gateway. It should only be called from the EVLORAMACRXDATA event handler.

LORAMACRxData (stData$, pRxRSSI, pRxPort, pRxSNR)

Returns	0x0000 : Success
Arguments:	
stData$	**ByRef stData$ STRING** The data read from the received packet.
pRxRSSI	**ByRef pRxRSSI INTEGER** The RSSI value of the received packet.
pRxPort	**ByRef pRxPort INTEGER** The port on which the packet was received.
pRxSNR	**ByRef pRxSNR INTEGER** The SNR value of the received packet.
Interactive Command	No

```
//================================================================================
// This handler is called when downlink data is received from the gateway
//================================================================================
function HandlerLoRaRxData() as integer
  dim data$
      dim nRSSI,nPort,nSNR

      rc = LORAMACRxData(data$, nRSSI, nPort, nSNR)
      print "LoRa Received downstream data on port ";nPort;"\nRSSI: ";nRSSI;"  SNR:
";nSNR;"\n";data$;"\n"
endfunc 1

OnEvent EVLORAMACRXDATA     call HandlerLoRaRxData
```

LORAMACRxData is an extension function.

3.1.7 LORAMACGetOption

FUNCTION

This function retrieves the value for a specified option.

Note: A list of options and their descriptions can be found in RM1xx-defs.h as well as in Table 1.

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

LORAMACGetOption (nOptID, optValue$)

Returns	0x0000 : Successfully retrieved option value 0xnnnn : Result code
Arguments:	
nOptID	**ByVal nOptID INTEGER** The Option ID as defined in TargetRM1xx-defs.h
optValue$	**ByRef optValue$ STRING** The string in which to return the option value.
Interactive Command	No

```
//Example Code
dim reg
dim stringVal$
dim rc

// Retrieve the current data rate and print it
rc = LORAMACGetOption(LORAMAC_OPT_DATA_RATE, stringVal$)
print stringVal$
```

LORAMACGetOption is an extension function.

3.1.8 LORAMACSetOption

FUNCTION

This function sets the value for a specified option.

Note: A list of options and their descriptions can be found in RM1xx-defs.h as well as in Table 1.

LORAMACSetOption (nOptID, optValue$)

Returns	0x0000 : Successfully set option value 0xnnnn : Result code
Arguments:	
nOptID	**ByVal nOptID INTEGER** The Option ID as defined in TargetRM1xx-defs.h
optValue$	**ByRef optValue$ STRING** The value to set.
Interactive Command	No

```
//Set the data rate to 5
temp$ = "5"
rc = LORAMACSetOption(LORAMAC_OPT_DATA_RATE, temp$)
```

LORAMACSetOption is an extension function.

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

3.1.9 LORAMAC Option List

These options are available for use with the LORAMACGetOption and LORAMACSetOption (as specified in the *Get* and *Set* columns. Enumerations of these options can be found in RM1xx-defs.h.

Table 1: LORAMAC Option List

Option Name	Get	Set	Description
LORAMAC_OPT_TX_POWER	YES	YES	The output power in dB of the LoRa radio. The value entered must be one of the values specified in the LoRaWAN specification. For the RM186 valid values are 2, 5, 8, 11, 14 and 20dBm and for the RM191 10 to 30dBm in 2dBm steps. If an invalid value is entered then the code will set the power to the nearest value lower than the one entered. Note that the RM1xx is limited to a maximum output power of 13dBm. If a value higher than that is entered the code will automatically limit the power.
LORAMAC_OPT_DATA_RATE	YES	YES	The data rate of the LoRa radio. Valid values are 0 to 5 for the RM186 and 0 to 4 for the RM191.
LORAMAC_OPT_JOIN_STATE	YES	NO	LoRaWAN network joined attribute
LORAMAC_OPT_DEV_EUI	YES	NO	The device EUI assigned by Laird
LORAMAC_OPT_CUSTOM_DEV_EUI	YES	YES	An optional custom device EUI *not* provided by Laird
LORAMAC_OPT_DEV_ADDR	YES	YES	The end-device address
LORAMAC_OPT_APP_EUI	YES	YES	The application EUI
LORAMAC_OPT_APP_KEY	NO	YES	The application key
LORAMAC_OPT_VERSION	YES	NO	The version number of the device
LORAMAC_OPT_RSSI	YES	NO	The Received Signal Strength Indicator of the last recevied packet available after an RX Complete event
LORAMAC_OPT_SNR	YES	NO	The Signal-to-Noise Ratio of the last received packet available after an RX Complete event.
LORAMAC_OPT_DOWNLINK_COUNTER	YES	NO	Number of down-link packets received
LORAMAC_OPT_UPLINK_COUNTER	YES	NO	Number of up-link packets sent
LORAMAG_OPT_SOURCE_VOLTAGE	YES	NO	The supply voltage of the device in millivolts
LORAMAC_OPT_915_HYBRID_MODE	-	-	Deprecated in version 17/18.4.1.0 in RM1xx and never supported in RM1xx_PE.
LORAMAC_OPT_BIRTHDAY	YES	NO	The date the device was created
LORAMAC_OPT_ADR_ENABLE	YES	YES	Enable (1) and disable (0) automatic data rate adaptation.
LORAMAC_OPT_CHANNELLIST	YES	NO	Lists the channel number, frequency and maximum datarate of all the enabled channels. Only valid in EU mode

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

Option Name	Get	Set	Description
LORAMAC_OPT_CHANNELMASK	YES	YES/ NO	The channel mask designating the enabled channelsOnly valid in EU mode. Set option can only be used on the 915(US) MHz radio. The set option will override the current value, but will not be persisted. After a reboot it will revert back to the default or configured value. See the Setting RM191 ChannelsMask section below.
LORAMAC_OPT_NEXT_TX	YES	NO	Returns the time, in seconds, until the packet just loaded will be transmitted to the gateway.
LORAMAC_OPT_TEMPERATURE	YES	NO	Temperature in degrees Celcius.
LORAMAC_OPT_TEMP_COMP_FACTOR	YES	NO	Temperature compensation factor – device specific.
LORAMAC_OPT_FREQ_ERROR	YES	NO	Frequency error (Hz) of the SX1272.
LORAMAC_OPT_FREQ_OFFSET	YES	NO	SX1272 frequency offset of the SX1272.
LORAMAC_OPT_MAX_RETRIES	YES	YES	Sets the number of times the module will attempt to resend a confirmed packet if an acknowledgment is not received. The maximum and default values are 8.

3.1.10 LORAMACSetDebug

This command should only be used during development. It is not recommended that this command is left active in any production version of a *smart*BASIC application.

FUNCTION

This function sets the module up to output certain debug information, either as a text string or as a waveform.

LORAMACSetDebug (nDebug, nTxSio, nRxSio)

Returns	0x0000 : Successfully actioned command 0xnnnn : Result code
Arguments:	
nDebug	**ByVal** nDebug **INTEGER** 0 – Disables debug mode 1 – Enables debug mode
nTxSio	**ByVal** nTxSio **INTEGER** The sio pin that the Tx waveform is output on.
nRxSio	**ByVal** nRxSio **INTEGER** The sio pin that the Rx waveform is output on.
Interactive Command	No

If debug mode is enabled, the module will output the frequency that a packet will be transmitted on, the data rate of that packet when transmitting, and the spreading factor/symbols timeout when receiving.

It will also output 2 waveforms, on the selected SIOs, that mark when the module is in transmit or receive mode.

TxSio and RxSio cannot be the same value unless debug mode is being disabled.

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

Debug mode is not persisted. If debug mode was enabled and the module was then reset, it would boot up with debug mode disabled.

3.2 Setting RM191 ChannelsMask

Note that this section is only relevant to the RM191 modules. The setting of the ChannlesMask in the RM186 is handled completely differently and requires no input by the user.

The RM191 has 64 available 125kHz upstream channels (channels 0-63) starting at 902.3MHz and incrementing linearly in 200kHz steps up to 914.9MHz. There are also 8 500kHz upstream channels (channels 64-71) starting at 903MHz and incrementing linearly in 1.6MHz steps up to 914.2MHz.

Channels are enabled by setting the bit that corresponds to their channel number in a hex string that is either loaded during bootup from a default value or from a value stored in Flash using the at+cfgex 1009 command.

The value that needs to be set is governed by what frequency channels are allowed by the gateway or gateways you wish to communicate with, and for that you will have to consult your network administrator.

For example, the Multitech gateways are configured by selecting one of a series of 8 channel sub-bands, each containing 8x125kHz channels and 1x500kHz channel. . Sub-band 1 consists of channels 0-7 and channel 64, sub-band 2 of channels 8-15 and channel 65, up to sub-band 8 which consists of channels 56-63 and channel 71.

The RM191 will default to sub-band 1, however if a different configuration is required then you can set the required channels as show below.

Table 2 : ChannelMask commands

Sub-Band	Frequency Range (MHz)	Channels	Command
1	902.3–903.7	0-7	at+cfgex 1009 "00010000000000000ff"
2	903.9–905.3	8-15	at+cfgex 1009 "0002000000000000ff00"
3	905.5–906.9	16-23	at+cfgex 1009 "00040000000000ff0000"
4	907.1–908.5	24-31	at+cfgex 1009 "00080000000ff000000"
5	908.7–910.1	32-39	at+cfgex 1009 "0010000000ff00000000"
6	910.3–911.7	40-47	at+cfgex 1009 "00200000ff0000000000"
7	911.9–913.3	48-55	at+cfgex 1009 "004000ff000000000000"
8	915.5–914.9	56-63	at+cfgex 1009 "0080ff00000000000000"
All bands	902.3–914.9	0-63	at+cfgex 1009 "00ffffffffffffffffff"

In the table above, channel 0 would be the least significant bit on the extreme right hand side of the string.

The “All Bands” option may be used if you don’t know the configuration of a gateway. With this ChannelsMask the RM191 continues to transmit a packet on random channels until it finds one that worked. This would be repeated every packet. So there could be many failed transmissions for each packet.

Most gateways at present only support the “1 side-band” option as this is determined by the configuration of the SX1301 module. However, if a gateway contains more than one of these modules then it is simply a case of enabling the extra bits. For example, to enable sub-bands 1 and 4 the following string would be entered:

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

```
at+cfgex 1009 "000900000000ff0000ff"
```

While it is possible to set the hex string to any channel configuration, the present method of configuring the gateways means that the ChannelsMask will most likely be grouped in bands of 3, 4 or 5 channels. For example if a gateway supported channels 4-7, 28-31 and 67 then the following string would be required:

```
at+cfgex 1009 "000800000000f00000f0"
```

Note that the 500kHz channel must fit in one of the 125kHz channel sub-bands.

However, again, it is important to stress that you must contact your network administrator to determine which settings you require.

As has previously been mentioned, the at+cfgex 1009 "xxx…" command is stored in flash and the module must be configured before running a smartBASIC application. However it is also possible to modify the channelsmask from a smartBASIC application using the LoramacSetOption(LORAMAC_OPT_CHANNELMASK,xxx…).

So if you wanted to set the RM191 to sub-band 2 then you could enter:

```
LoramacSetOption(LORAMAC_OPT_CHANNELMASK, 00020000000000ff00)
```

Note with this method you do not require the "" around the hex string. ChannelsMasks entered this way are not persisted. When the module reboots it will revert back to the stored or default value.

3.3 Events and Messages

3.3.1 EVLORAMACJOINING

The device has started the Over-the-Air (OTA) Join procedure. A Join Request message will be sent and the device will await the Join Response message from the server.

3.3.2 EVLORAMACJOINED

The device successfully completed the Join procedure. If using the OTA Join procedure, this indicates that a successful Join Response message was received from the server. This event will be thrown immediately when joining the network using Activation by Personalization (ABP).

3.3.3 EVLORAMACJOINFAILED

The Join procedure failed.

3.3.4 EVLORAMACREJOINFAILED

The device is no longer trying to send data upstream.

3.3.5 EVLORAMACRESET

The SX1272 device has been reset.

3.3.6 EVLORAMACRXCOMPLETE

The device successfully received a LoRa packet with payload data.

3.3.7 EVLORAMACTXTIMEOUT

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

An ACK was not received for a confirmed uplink message

3.3.8 EVLORAMACRXTIMEOUT

A timeout occurred waiting for a downlink message.

3.3.9 EVLORAMACRXERROR

An error occurred in the receive path.

3.3.10 EVLORAMACRXDATA

Downstream data received from the gateway. The data can be read by the application using LORAMACRxData().

3.3.11 EVLORAMACLINKCHECKRESPMSG

This message is returned from the LoRa stack when a Link Check response is received from a LoRa gateway. The demodulation margin is reported as an 8-bit unsigned integer named **Margin** which ranges from 0 to 254. Margin indicates the link margin in dB of the last successfully reported LinkCheckReq command. The gateway count is is returned as a variable named **GwCnt** and represents the number of gateways that successfully received the last LinkCheckReq command. See section 5.1 of the LoRaWAN specification for more details.

3.3.12 EVLORAMACTXDONE

This event indicates that a packet has successfully been transmitted from the radio. It is this event that all subsequent receive windows are timed with respect to.

3.3.13 EVLORAMACNOSYNC

This event indicates that the receiver has failed to detect a sync message from the gateway during a receive window and so has exited the receive state. As there can be 2 potential receive windows for every transmitted packet, in worse case scenarios there can be two of these events for every uplink packet.

3.3.14 EVLORAMACADR

This event indicates that the RM1xx has received an ADR command from the gateway/server and that a parameter setting on the module could have changed. The parameters that this command can change are the TxPower, Datarate or ChannelsMask.

3.3.15 EVLORAMACNEXTTX

This event indicates that there as available duty cycle on one of the sub-bands. If a LORAMACTxData command is sent on receipt of this event, the data will be sent without any delays.

4 ACKNOWLEDGEMENTS

The following are required acknowledgements to address our use of open source code on the RM1xx to implement AES encryption.

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

Laird's implementation includes the following files: **aes.c** and **aes.h**.

4.1.1.1 License Terms

The redistribution and use of this software (with or without changes) is allowed without the payment of fees or royalties providing the following:

- Source code distributions include the above copyright notice, this list of conditions and the following disclaimer;
- Binary distributions include the above copyright notice, this list of conditions and the following disclaimer in their documentation;
- The name of the copyright holder is not used to endorse products built using this software without specific written permission.

4.1.1.2 Disclaimer

This software is provided 'as is' with no explicit or implied warranties in respect of its properties, including, but not limited to, correctness and/or fitness for purpose.

Issue 09/09/2006

This is an AES implementation that uses only 8-bit byte operations on the cipher state (there are options to use 32-bit types if available).

The combination of mix columns and byte substitution used here is based on that developed by Karl Malbrain. His contribution is acknowledged.

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

5 INDEX OF *SMART*BASIC COMMANDS