LoRa mPCIe Smart Gateway Card Command Reference Guide

1.0 INTRODUCTION

This document describes the communication interface to the OccamSmart LoRa mPCIe Smart Gateway Card. The card contains a PIC24FJ256GB206 microcontroller which provides the interface for the customer's host micro controller. This interface can be used to configure and control the LoRa Gateway Radio Module. The interface uses the USB signals on the mPCIe connector.

1.1 mPCIe PIN ASSIGNMENTS

TABLE 1: PIN Function Table

Name	Number	Function
Vcc	2,24,39,41,52	Supply Voltage
Vss	4,9,15,18,21,26, 27,29,34,35,37, 40,43,50,	Ground
PPS	5	GPS PPS Input (Future)
HOST_URX	17	Host UART Receive Input (SG9002 only)
HOST_UTX	19	Host UART Transmit Output. (SG9002 only)
RF_ENABLE	20	RF Enable Input.
RESET	22	Reset Module Input.
USB_D-	36	Host USB D-.
USB_D+	38	Host USB D+.
LED1	42	LED1 Output.
LED2	44	LED2 Output.
LED3	46	LED3 Output.
PCI_DETECT	48	PCI bus detect input. (Future)

Note: All other pins are unused.

1.2 PIN DESCRIPTIONS [FIX NUMBERING BELOW]

1.1 GPS PPS Input: (FUTURE)
The GPS synchronizing clock (PPS) can be input to this pin.

1.2 UART Receive (HOST_URX): (SG9002 only)
The HOST_URX input is the asynchronous UART receive to the module.

1.3 UART Transmit (HOST_UTX): (SG9002 only)
The HOST_UTX output is the asynchronous UART transmit from the module.

1.4 Enable Module RF (RF_ENABLE):
This input is used to enable the RF radio on the LoRa Gateway Module. A low level on RF_ENABLE will disable all RF transmission and reception.

1.5 Reset Module (RESET):
This input is used to reset the microcontroller on the LoRa mPCIe Smart Gateway Card. A low level pulse on RESET will instruct the module perform a software reset**.** This pin is sampled by the microcontroller.

1.6 LED Outputs (LED1, LED2, LED3):
The module provides three output pins capable of driving off-board LEDs. These signals can be controlled by the host microcontroller using software commands to the module.

1.7 PCI Bus Detect (PCI_DETECT): (FUTURE)
The PCI_DETECT input is used to select the LoRa mPCIe Smart Gateway Card communication interface. It is intended to be connected to the 1.5VDC pin on the mPCIe connector. Presence of 1.5V on this pin informs the module that it is connected to a standard mPCIe interface.

1.3 Hardware Interface

The module communicates with the host system using the USB (SG9001) or UART (SG9002) signals on the mPCIe connector. When powered up, the SG9001 will perform the required USB handshake with the host's USB controller and enumerate as a CDC (Communications Device Class). This will make the card appear to the host as a serial communication device. The SG9002 communicated using a dedicated UART and needs no handshaking at startup.

2.0 CONFIGURATION

When the module boots, it loads the stored configuration settings form NVM which can be changed and stored for specific network configurations. After loading the configuration settings, the module is then ready to send and receive LoRa packets. To configure parameters, send packets and/or view current configuration, you must send commands over the module interface. If the module's RF ENABLE pin is High, it will asynchronously send any received packets to the host.

Radio Configuration

The SX1301 digital baseband chip contains 10 programmable reception paths. Those paths have differentiated levels of programmability and allow different use cases. It is important to understand the differences between those demodulation paths to make the best possible use from the system.

IF0 to IF7 LORA channels

These channels can be connected individually to Radio 0 or 1. The channel bandwidth is 125kHz and cannot be modified or configured. Each channel's IF frequency can be individually configured relative to the selected radio's center frequency. On each of these channels any data rate can be received without prior configuration. Several packets using different data rates may be demodulated simultaneously even on the same channel. These channels are intended to be used for a massive asynchronous star network of 10000's of sensor nodes. Each sensor may use a random channel (amongst IF0 to IF7) and a different data rate for any transmission.

Typically, sensors located near the gateway will use the highest possible data rate in the fixed 125kHz channel bandwidth (i.e. 6kbit/sec) while sensors located far away will use a lower data rate down to 300bit/sec (minimum LORA data rate in a 125kHz channel) .

The SX1301 digital Baseband chip scans the 8 (IF0 to IF7) channels for preambles of all data rates at all times.

IF8 LORA channel

This receive channel can be connected to Radio 0 or 1 and programmed for any intermediate frequency within the allowed range. This channel is LORA only. The demodulation bandwidth can be configured to be 125 , 250 or 500kHz. The data rate can be configured to any of the LORA available data rates (SF7 to SF12) but, as opposed to IF0 to IF7 , ONLY the configured data rate will be demodulated. This channel is intended to serve as a high speed backhaul link to other gateways or infrastructure equipment.

IF9 (G)FSK channel

Same as IF8 except that this channel is connected to a (G)FSK demodulator. The channel bandwidth and bitrate can be adjusted. This demodulator offers a very high level of configurability, going well beyond the scope of this document. The demodulator characteristics are essentially the same than the GFSK demodulator implemented on the SX1232 and SX1272 Semtech chips. This demodulation path can demodulate any legacy FSK or GFSK formatted signal.

3.0 COMMAND REFERENCE

The module comes from the factory with default settings (TABLE 3A/3B FACTORY DEFAULT SETTING) which can be restored by performing a factory reset of the module.

TABLE 3A: FACTORY DEFAULT SETTING (868 MHz Module)

PARAMETER	DEFAULT VALUE
UART Baud Rate	115200
RF Chain 0	Center Freq = 868200000
RF Chain 1	Center Freq = 869200000
IF Chain 0	Enable= True, RF Chain = 0, Offset Freq = -100000, BW= 125 kHz, SF= Multi
IF Chain 1	Enable= True, RF Chain = 0, Offset Freq = 100000, BW= 125 kHz, SF= Multi
IF Chain 2	Enable= True, RF Chain = 0, Offset Freq = 300000, BW= 125 kHz, SF= Multi
IF Chain 3	Enable= True, RF Chain = 1, Offset Freq = -350000, BW= 125 kHz, SF= Multi
IF Chain 4	Enable= True, RF Chain = 1, Offset Freq = -150000, BW= 125 kHz, SF= Multi
IF Chain 5	Enable= True, RF Chain = 1, Offset Freq = 325000, BW= 125 kHz, SF= Multi
IF Chain 6	Enable= False, RF Chain = x, Offset Freq = xx, BW= 125 kHz, SF= Multi
IF Chain 7	Enable= False, RF Chain = x, Offset Freq = xx, BW= 125 kHz, SF= Multi
IF Chain 8	Enable= True, RF Chain = A, Offset Freq = 100000, BW= 250 kHz, SF= SF7
IF Chain 9	Enable= True, RF Chain = A, Offset Freq = 100000, BW= 250 kHz, DR= 50000
RX Polling Window	0x02 (2* 50ms)
LoRa Sync Byte	0x34

TABLE 3B: FACTORY DEFAULT SETTING (915 MHz Module)

PARAMETER	DEFAULT VALUE
UART Baud Rate	115200
RF Chain 0	Center Freq = 902600000
RF Chain 1	Center Freq = 903400000
IF Chain 0	Enable= True, RF Chain = 0, Offset Freq = -300000, BW= 125 kHz, SF= Multi
IF Chain 1	Enable= True, RF Chain = 0, Offset Freq = -100000, BW= 125 kHz, SF= Multi
IF Chain 2	Enable= True, RF Chain = 0, Offset Freq = 100000, BW= 125 kHz, SF= Multi
IF Chain 3	Enable= True, RF Chain = 0, Offset Freq = 300000, BW= 125 kHz, SF= Multi
IF Chain 4	Enable= True, RF Chain = 1, Offset Freq = -300000, BW= 125 kHz, SF= Multi
IF Chain 5	Enable= True, RF Chain = 1, Offset Freq = -100000, BW= 125 kHz, SF= Multi
IF Chain 6	Enable= True, RF Chain = 1, Offset Freq = 100000, BW= 125 kHz, SF= Multi
IF Chain 7	Enable= True, RF Chain = 1, Offset Freq = 300000, BW= 125 kHz, SF= Multi
IF Chain 8	Enable= True, RF Chain = 0, Offset Freq = 0, BW= 250 kHz, SF= SF10
IF Chain 9	Enable= True, RF Chain = 0, Offset Freq = 0, BW= 250 kHz, DR= 64000
RX Polling Window	0x02 (2* 50ms)
LoRa Sync Byte	0x34

3.1 Operation Mode:

Commands to the LoRa mPCIe Smart Gateway Card and data returned from the Module use the following communication format:

FIGURE 1: COMMUNICATION PROTOCOL FORMAT

Communication Protocol					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**

TABLE 4: COMMUNICATION PROTOCOL DESCRIPTION

Name	Description	Size (Bytes)
Frame Start	Frame Start -> 0x23	1
Command	Command (See **TABLE 5**)	1
Length	Length of Data Payload	2
Data Payload	Data	0 ~ 300
Checksum	Checksum = (Frame Start + Length + Command + Data Payload) MOD 0x100	1
CR	Carriage Return -> 0x0D	1

TABLE 5 COMMAND LIST contains a list of the commands supported by the module.

TABLE 5: COMMAND LIST

Command	Value	Direction	Command Description
FACTORY	0x6E	Host -> Module	Restores the factory default settings.
SAVE	0x21	Host -> Module	Saves the current settings to EEPROM.
SETUART	0x42	Host -> Module	Sets the UART baud rate to the Host.
GETUART	0x62	Host -> Module	Returns the Host UART baud rate setting.
START	0x30	Host -> Module	Start RF Module.
STOP	0x31	Host -> Module	Stop RF Module.
SEND	0x32	Host -> Module	Host Send LoRa Packet.
RECEIVE	0x33	Module -> Host	Host Receive LoRa Packet.
RFCONFIG	0x34	Host -> Module	Configure RF chain center frequency.
IFCONFIG	0x35	Host -> Module	Configure IF chain offset frequency.
IF8CONFIG	0x36	Host -> Module	Configure IF8 (LoRa) offset frequency, bandwidth and data rate.
IF9CONFIG	0x37	Host -> Module	Configure IF9 (FSK) offset frequency, bandwidth and data rate.
TXABORT	0x38	Host -> Module	Aborts any current or scheduled transmission.
TXSTATUS	0x39	Host -> Module	Returns radio transmitter status.
VERSION	0x3A	Host -> Module	Returns Module version information.

TABLE 5: COMMAND LIST (CONTINUED)

Command	Value	Direction	Command Description
RFCHAIN	0x3B	Host -> Module	Returns RF (0-1) chain settings.
IFCHAIN	0x3C	Host -> Module	Returns IF chain (0-7) settings.
IF8CHAIN	0x3D	Host -> Module	Returns IF8 chain settings.
IF9CHAIN	0x3E	Host -> Module	Returns IF9 chain settings.
SETLEDS	0x3F	Host -> Module	Sets the output LED's state.
SETSYNC	0x40	Host -> Module	Sets the LoRa IF channels Sync Word.
GETSYNC	0x41	Host -> Module	Returns the LoRa IF channels Sync Word.
RXSTATUS	0x43	Host -> Module	Returns the LoRa Receive status.
BOOTLOADER MODE	0x50	Host -> Module	Invalidates the checksum of the application. Need to followed by a RESET command to switch to the Bootloader*
RESET	0x51	Host -> Module	Resets the module
SENDCW	0x25	Host -> Module	Enable CW Mode (Continuous Transmit)
INVALID	0xFF	Module -> Host	Previous command received from Host does not exist
MFGDATA	0x07	Host -> Module	Program Manufacturing data (band, version, serial number)

* Note: For more information on the Bootloader, please refer to chapter 6.

The module will respond to ALL commands using the same communication format (FIGURE 1: UART COMMUNICATION PROTOCOL). Each response will have the same command byte followed by a payload. Response payloads are detailed in the next section. Some responses contain only a status byte. The status byte codes are described in TABLE 6 STATUS BYTE FORMAT.

TABLE 6: STATUS BYTE FORMAT

STATUS Byte	Name	Description
Bit 0	ACK/NAK	ACK=0, NAK=1
Bit 1	-	Unused
Bit 2	-	Unused
Bit 3	-	Unused
Bit 4	Err:0	Error Code: 0000: No Error 0001: Command Fail
Bit 5	Err:1	
Bit 6	Err:2	

Bit 7	Err:3	

3.2 Command Details
The following sections provide the specific details of the module commands.

3.3 Start RF Module Command (START):
Use to start up the LoRa Module. Command will reset the RF chain and configure it according to the currently set RF and IF chain parameters. This command MUST be issues before sending or receiving RF frames.

Example():

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<30>	<00><00>	Empty	<53>	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<30>	<01><00>	<01>	<55>	<0D>

3.4 Stop RF Module Command (STOP):
Use to stop the LoRa concentrator and disconnect. The Module will respond the current STATUS register result of the command.

Example():

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<31>	<00><00>	Empty	<54>	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<31>	<01><00>	<01>	<56>	<0D>

3.5 Send LoRa packet Command (SEND):
The SEND command is used to send packets. The following parameters can be dynamically programmed with each packet:

- Transmit frequency
- RF Chain. Only Chain 0 is active on the board

- FSK or LORA modulation
- Bandwidth, data rate, and coding rate (LORA mode)
- Bandwidth, bit rate and frequency deviation (FSK mode)
- RF output power
- Time of departure (immediate or differed based on the gateway hardware clock with 1uS accuracy)

Dynamic parameter fields are sent alongside the payload in the same data buffer. The data buffer on the module can only hold a single packet at a time (next packet to be sent). The scheduling and ordering task is left to the host micro-processor. The user payload starts at byte 26. This is the first byte of the RF frame. The metadata in bytes 0 to 25 are not transmitted and are used to dynamically configure the radio before each transmission.

Example (Send LoRa Packet):

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<32>	<15><0>	<header[0:23]><payload[0:4]>	Checksum	<0D>
***Note: See TABLE 7 LoRa Send Packet Format**					
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<32>	<01><00>	Ack (0x00)	<56>	<0D>

TABLE 7: LoRa SEND PACKET FORMAT

Byte	Subfield (bits)	Description	Comment
0	0 – 7	Transmit Frequency	
1	8 – 15		
2	16 – 23		
3	24 - 31		
4		Transmit Mode	0 = Immediate, 1 = Time
5	0 – 7	Transmit Start Time (when Transmit Mode = 1)	microseconds
6	8 – 15		
7	16 – 23		
8	24 - 31		
9		RF Chain	0 or 1; Only Ch. 0 is active
10		Tx Power	-6 to 27 (dBm)
11		Modulation	0x10 = LoRa, 0x20 = FSK
12		Modulation Bandwidth	LoRa: 1 = 500 kHz, 2 = 250, 3 = 125 FSK: 1 = 500 kHz, 2 = 250, 3 = 125, 4 =

			62.5, 5 = 31.25, 6= 15.625, 7 = 7.8125
13	0 – 7	Data Rate (LoRa) / Baudrate (FSK)	LoRa: 2 = SF7, 4 = SF8, 8 = SF9, 16 = SF10, 32 = SF11, 64 = SF12. FSK: 500 - 250000
14	8 – 15		
15	16 – 23		
16	24 - 31		
17		Coding Rate (LoRa only)	1 = 4/5, 2 = 4/6, 3 = 4/7, 4 = 4/8
18		Invert Polarity (LoRa only)	0 = Normal Polarity, 1 = Invert Polarity
19		Frequency Deviation (FSK Only)	FSK: 1 – 200 (kHz)
20	0 – 7	Preamble Length	0 = Use default (LoRa: 6 symbols, FSK: 5 bytes)
21	8 - 15		
22		CRC Disable	0 = CRC Enable, 1 = CRC Disable
23		Implicit Header Enable (LoRa Only)	0 = Explicit Header 1 = Implicit Header
24	0 - 7	Payload Length	Size in bytes (max = 255)
25	8 - 16		
26 – n		Payload	

*Note: Unused bytes (depending on protocol) should be set to 0.

3.6 Receive LoRa packet Command (RECEIVE):

After receiving a START command, the module will begin asynchronously sending received packets. it will send the packets to the host using the RECEIVE command. After successfully receiving a packet, the host must acknowledge the packet. Once the host has acknowledged the packet it is removed from the modules receive buffer. If the host does not successfully receive the packet it can request the packet be sent again by NAK'ing the packet.

Example (Receive Packet):

Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<33>	<16><00>	<LoRa Receive Packet>	Checksum	<0D>
***Note: See TABLE 8 LoRa Receive Packet Format**					
Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<33>	<01><00>	<00>	<57>	<0D>

TABLE 8: RECEIVE PACKET FORMAT

Byte	Sub Field	Parmeter	Description
0		RX Status	Receive status (See TABLE 19: RXSTATUS RETURN CODES)
1	31:24	Channel Frequency	IF Channel Frequency
2	23:16		
3	15:8		
4	7:0		
5		IF Chain	IF Chain packet received
6		Status	Status of Received Packet (0: No Status Rcv'd, 0x01:NO CRC, 0x11: CRC_BAD, 0x10: CRC_OK)
7	31:24	Time Stamp	32bit Time stamp, 1uSec step
8	23:16		
9	15:8		
10	7:0		
11		RF Chain	RF Chain packet received
12		Modulation	<See Item 3>
13		Bandwidth	<See Item 4>
14	31:24	Data Rate	RX Data rate of the packet (<See Item 5>)
15	23:16		
16	15:8		
17	7:0		
18		Coderate	Error-correcting code of the packet (LoRa Only)
19	15:8	RSSI	Average signal strength in dB
20	7:0		
21	15:8		
22	7:0		
23	15:8		

24	7:0	SNR Average	Packet Average SNR in dB (LoRa Only)
25	15:8		
26	7:0		
27	15:8	SNR Minimum	Packet Minimum SNR in dB (LoRa Only)
28	7:0		
29	15:8		
30	7:0		
31	15:8	SNR Max	Packet Maximum SNR in dB (LoRa Only)
32	7:0		
33	15:8		
34	7:0		
35	15:8	CRC	Value of computed CRC16
36	7:0		
37	15:8	Payload Size	Number of Bytes in Payload
38	7:0		
Payload:			

Byte	Sub Field	Parmeter	Description
39		Payload First Byte	Up to 256 bytes

3.7 Configure RF Radio Chain Command (RFCONFIG):

The RFCONFIG command is used to configure the RF center frequency for each of the SX1257 RF transceivers. The LoRa mPCIe Smart Gateway Card contains 2 SX1257 radios labeled Chain 0 and Chain 1. The center frequency is a 4-byte value in Hz.

TABLE 9: CONFIGURE RF CHAIN FORMAT

Byte	Parmeter	Description
0	RF Chain	RF Chain
1	Enable	Enable Chain
2	Frequency[7:0]	

3	Frequency[15:8]	Chain Center Frequency
4	Frequency[23:16]	
5	Frequency[31:24]	

Example (Chain 0, Enable= True, Cf = 868200000 Hz):

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<34>	<06><00>	<00><01><40><AE><BF><33>	Checksum	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<34>	<01><00>	<00>	<58>	<0D>

3.8 Configure IF chain Command (IFCONFIG):

The IFCONFIG command is used set the center frequency of the IF chains 0 through 7 relative to the assigned RF chain frequency in 2's compliment. The IFCONFIG data payload is sent in the following format:

TABLE 10: CONFIGURE IF CHAIN FORMAT

Byte	Parameter	Description
0	IF Channel	IF Channel : 0 - 7
1	Enable	Enable IF Channel: Enable = 1, Disable = 0
2	RF Chain	IF Channel's RF Chain: RF Chain 0 = 0, RF Chain 1 = 1
3	Freq[0:7]	Offset Frequency (signed integer)
4	Freq[15:8]	
5	Freq[23:16]	
6	Freq[31:24]	

Example (Channel = 2, Enable = True, RF Chain = 0, Cf = -300000 Hz):

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<35>	<07><00>	<02><01><00><20><6C><FB><FF>	Checksum	<0D>
Module -> Host:					

Frame Start	Command	Length	Data Payload	Checksum	CR
<23>	<35>	<01><00>	<00>	<59>	<0D>

3.9 Configure IF8 Channel Command (IF8CONFIG):

The IF8CONFIG command is used to set the IF8 channel receive frequency, bandwidth and data rate.

TABLE 11: IF8 CHANNEL FORMAT

Byte	Parameter	Description	
0	Pad Byte	Padding byte (0x00)	
1	Enable	Enable = 1, Disable = 0	
2	RF Chain	RF Chain 0 = 0, RF Chain 1 = 1	
3	Freq[0:7]	Offset Frequency (signed integer)	
4	Freq[15:8]		
5	Freq[23:16]		
6	Freq[31:24]		
7	BW	**Bandwidth**	**Value**
		500 kHz	0x01
		250 kHz	0x02
		125 kHz	0x03
8	Data Rate	**Data Rate**	**Value**
		SF7	0x02
		SF8	0x04
		SF9	0x08
		SF10	0x10
		SF11	0x20
		SF12	0x40

Example (Enable = True, RF Chain = 0, Cf = -300000 Hz, BW = 500 kHz, Data Rate = SF7):

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<36>	<08><00>	<01><00><20><6C><FB><FF><01><02>	Checksum	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<36>	<01><00>	<00>	<59>	<0D>

3.10 Configure IF9 Channel Command (IF9CONFIG):

The IF9CONFIG command is used to set the IF9 channel RX bandwidth and data rate.

TABLE 12: IF9 CHANNEL FORMAT

Byte	Parameter	Description
0	Pad Byte	Padding Byte (0x00)
1	Enable	Enable = 1, Disable = 0
2	RF Chain	RF Chain A = 0, RF Chain B = 1
3	Freq[0:7]	Offset Frequency (signed integer)
4	Freq[15:8]	
5	Freq[23:16]	
6	Freq[31:24]	
7	BW	1 = 500 kHz, 2 = 250, 3 = 125, 4 = 62.5, 5 = 31.25, 6 = 15.625, 7 = 7.8125
8	Rate[0:7]	Baudrate
9	Rate[15:8]	
10	Rate[23:16]	
11	Rate[31:24]	

Example (Enable = True, RF Chain = 0, Cf = -300000 Hz, BW = 500 kHz, Rate =):

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<37>	<0B><00>	<01><00><20><6C><FB><FF> <01><><><><>	Checksum	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<37>	<01><00>	<00>	<59>	<0D>

IF9 (G)FSK channel

3.11 About Transmission Command (TXABORT):

The TXABOUT command is used to abort any schedules or active radio transmission.

Example ():

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<38>	<00><00>	Empty	Checksum	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<38>	<01><00>	<00>	<59>	<0D>

3.12 Transmitter Status Command (TXSTATUS):

The TXSTATUS command is used to get radio's current transmitter status. The Module will respond the current TXSTATUS return code (see TABLE 13).

TABLE 13: TXSTATUS RETURN CODES

Value	Description
0x00	TX state Unknown.
0x01	TX modem is disabled, it will ignore commands.
0x02	TX modem is free, ready to receive a command.
0x03	TX modem is loaded, ready to send after an event and/or delay.
0x04	TX modem is emitting.

Example (TXSTATUS = 0x02):

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<39>	<00><00>	Empty	Checksum	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<39>	<01><00>	<02>	<5A>	<0D>
***Note: See TABLE 13 TXSTATUS Return Codes**					

3.13 Module Version Command (VERSION):

The VERSION command is used to read the module's serial number, type, and version information.

TABLE 14: VERSION INFORMATION FORMAT

Byte	Parameter	Description
0	Band	Module Frequency Band (868 MHz = 1, 915 MHz = 2)
1	HWREV	Hardware Revision
[2:13]	SERNUM	Module Serial Number
14	Minor	Minor Revision
15	Major	Major Revision

Example (Band = 868 MHz, HW REV = 1, Serial Num = ENG152000001, Version = 1.0):

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<3A>	<00><00>	Empty	Checksum	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<3A>	<10><00>	<01><01><45><4E><47><33><32> <00><00><00><00><00><00><31> <00><01>	<xx>	<0D>

3.14 Read RF Radio Chain Settings Command (RFCHAIN):

The RFCHAIN command is used to read the RF center frequency for each of the SX1257 RF transceivers. The LoRa mPCIe Smart Gateway Card contains 2 SX1257 radios, numbered

starting from 0. The center frequency is a 4-byte value in Hz.

TABLE 15: READ RF CHAIN SETTINGS FORMAT

Byte	Parameter	Description
0	Enable	Enable = 1, Disable = 0
1	RF Chain Freq[7:0]	RF Chain 0 Center Frequency
2	RF Chain Freq[15:8]	
3	RF Chain Freq[23:16]	
4	RF Chain Freq[31:24]	

Example (Chain 0: Enable = True, Center Frequency = 868200000):

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<3B>	<01><00>	<00>	Checksum	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<3B>	<05><00>	<01><40><AE><BF><33>	<xx>	<0D>

3.15 Read IF chain Settings Command (IFCHAIN):

The IFCHAIN command is used read the offset frequency of the IF chains 0 through 7 relative to the assigned RF chain frequency in 2's compliment.

Example (Channel 2: Enable = True, RF Chain = 0, Cf =):

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<3C>	<01><00>	<02>	Checksum	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<3C>	<07><00>	<01><00><20><6C><FB><FF>	<xx>	<0D>

TABLE 16: READ IF CHANNEL SETTINGS FORMAT

Byte	Parameter	Description
0	Enable	1=Enable, 0=Disable
1	RFChain	RF Chain 0=Chain 0, 1= Chain 1
2	Frequency[7:0]	Frequency Offset (signed integer)
3	Frequency[15:8]	
4	Frequency[23:16]	
5	Frequency[31:24]	

3.16 Read IF8 chain Settings Command (IF8CHAIN):

The IF8CHAIN command is used read the center frequency of the IF8 chain relative to the assigned RF chain frequency in 2's compliment.

Example (Enable = True, RF Chain = 0, Cf = -300000, BW = 500 kHz, SF = SF7):

Host -> Module:

Frame Start	Command	Length	Data Payload	Checksum	CR
<23>	<3D>	<00><00>	Empty	Checksum	<0D>

Module -> Host:

Frame Start	Command	Length	Data Payload	Checksum	CR
<23>	<3D>	<08><00>	<01><00><20><6C><FB><FF><01><02>	<xx>	<0D>

TABLE 17: READ IF8 SETTINGS FORMAT

Byte	Parameter	Description	
0	Enable	Enable = 1, Disable = 0	
1	RF Chain	RF Chain 0 = 0, RF Chain 1 = 1	
2	Freq[0:7]	Offset Frequency (signed integer)	
3	Freq[15:8]		
4	Freq[23:16]		
5	Freq[31:24]		
		Bandwidth	**Value**
		500 kHz	0x01

6	BW	250 kHz	0x02
		125 kHz	0x03
7	Data Rate	**Data Rate**	**Value**
		SF7	0x02
		SF8	0x04
		SF9	0x08
		SF10	0x10
		SF11	0x20
		SF12	0x40

3.17 Read IF9 chain Settings Command (IF9CHAIN):

The IF8CHAIN command is used read the center frequency of the IF9 chain relative to the assigned RF chain frequency in 2's compliment. See Table 12 IF9 CHANNEL FORMAT for return data format.

Example (Enable = True, RF Chain = 0, Cf = -300000, BW = 500 kHz, Data Rate = xxxx):

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<3E>	<00><00>	Empty	Checksum	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<3E>	<0B><00>	<01><00><20><6C><FB><FF><01><><><><>	<xx>	<0D>

TABLE 18: READ IF9 SETTINGS FORMAT

Byte	Parameter	Description
0	Enable	Enable = 1, Disable = 0
1	RF Chain	RF Chain A = 0, RF Chain B = 1
2	Freq[0:7]	Offset Frequency (signed integer)
3	Freq[15:8]	
4	Freq[23:16]	
5	Freq[31:24]	

6	BW	1 = 500 kHz, 2 = 250, 3 = 125, 4 = 62.5, 5 = 31.25, 6 = 15.625, 7 = 7.8125
7	Rate[0:7]	Baudrate
8	Rate[15:8]	
9	Rate[23:16]	
10	Rate[31:24]	

3.18 Set LED Command (SETLEDS):

The SETLEDS command is used to turn ON or OFF any of the three LED0-2 output pins. LED0 = 0x01, LED1 = 0x02, LED2 = 0x04.

Example (LED0 = ON, LED1 = OFF, LED2 = ON):

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<3F>	<01><00>	<05>	Checksum	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<3F>	<01><00>	<01>	<xx>	<0D>

3.19 Set LoRa Sync Word Command (SETSYNC):

The SETSYNC command is used to set the LoRa channel's sync word. The sync word is used for all LoRa channel's 0-8.

Example (Sync = 0x34):

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<40>	<01><00>	<34>	Checksum	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<40>	<01><00>	<00>	<xx>	<0D>

3.20 Get LoRa Sync Word Command (GETSYNC):

The GETSYNC command is used to read the LoRa channel's sync word.

Example (Sync = 0x34):

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<41>	<00><00>	None	Checksum	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<41>	<01><00>	<34>	<xx>	<0D>

3.21 RX Status Command (RXSTATUS):

The RXSTATUS command is used to read the modules current RX Status. The RX status codes are described in TABLE 19: RXSTATUS RETURN CODES.

Example (RX Status = 0x32, Wait for ACK with 3 packets in RX Buffer):

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<43>	<00><00>	None	Checksum	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<43>	<01><00>	<32>	<xx>	<0D>

TABLE 19: RXSTATUS RETURN CODES (bit encoded)

Value	Description
[0x03:00]	Number of RX Packets in Receive Buffer
[0x04]	RX Buffer Overflow.
[0x05]	Waiting for ACK from Host.
[0x06]	Unused
[0x07]	Unused

3.22 Bootloader Mode Command (BOOTLOADER MODE):

The Bootloader MODE command is used to switch from the application mode to the bootloader mode. Once in the bootloader mode, a new application firmware can be programmed. Refer to chapter 6 for more details about bootloader.

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<50>	<00><00>	None	Checksum	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<50>	<01><00>	<00>	<xx>	<0D>

3.23 RESET Command (RESET):

The RESET command is used to RESET the module. If this command is followed after a BOOTLOADER_MODE, then module resets in bootloader mode.

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<51>	<00><00>	None	Checksum	<0D>

3.24 Continuous Wave Mode (SENDCW):

The SENDCW command is used to transmit Continuous Wave for testing purposes.

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<25>	<0B><00>	<CW Mode Parameters>	Checksum	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<25>	<01><00>	<00>	<xx>	<0D>

Table 20: CW Mode Parameters

Byte	Parameter	Description	
0	Enable	0 = Disable, 1 = Enable	
1	Freq[0:7]	Center Frequency	
2	Freq[15:8]		
3	Freq[23:16]		
4	Freq[31:24]		
5	RF Chain	RF Chain (Must be 0)	
6	Bandwidth	**Bandwidth**	**Value**
		500 kHz	0x01
		250 kHz	0x02
		125 kHz	0x03

Byte	Parameter	Description
7	Rate[0:7]	Data Rate (SF7 = 0x02, SF8 = 0x04, SF9 = 0x08, SF10 = 0x10, SF11 = 0x20, SF12 = 0x40, Multi = 0x7E) FSK Minimum Baud = 500; FSK Maximum Baud = 250000
8	Rate[15:8]	
9	Rate[23:16]	
10	Rate[31:24]	
11	Txpower	TX Power

3.25 INVALID Command (INVALID):

This module to host frame indicates host that the previous command sent by the Host is not a valid command

Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<FF>	<01><00>	<00>	Checksum	<0D>

3.26 Program Manufacturing Data (MFGDATA):

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<07>	<0E><00>	<Manufacturing Data>	Checksum	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<07>	<01><00>	<00>	<xx>	<0D>

Table 19: Manufacturing Data

Byte	Parameter	Description
1	Band	Frequency Band (868 = 0x01; 915 = 0x02)
2	Version	Hardware version of the module
3	Serial Number[0:11]	Serial Number (12 Characters)

4.0 FEATURES AND OPERATION

This section describes some of the commons features and operations of the LoRa mPCIe Smart Gateway Card.

4.1 Storing Setting to NVM

When the module boots it will load the current settings that are stored in NVM. Setting can then be changed and stored back to NVM using the SAVE command for use again after a Power On Reset. Settings can also be changed and used without storing them to NVM by changing the desired setting and performing a STOP command followed by a START command. This will cause the radio to load settings currently stored in its volatile memory.

If desired, the factory defaults can be reloaded by issuing the FACTORY command. To save the factory loaded defaults to NVM, a SAVE command must be issued.

4.2 Sending LoRa Packets

To send packets, first send a TXSTATUS command to ensure the Radio is ready to handle the new packet to send. Next, send the packet to the Module using the SEND command.

4.3 Receiving LoRa Packets

Receiving packets from the module is an asynchronous event. Following a START command, the radio is continuously listening for and storing pacts in its buffer. The module services this buffer by polling the radio every and loading any packets into its buffer. This buffer is capable of holding 6 LoRa packets. The module then asynchronous sends the first packet to the host and waits for the host to acknowledge the packet by sending an ACK status. If the host wishes for the module to resend the packet it can do so by sending an NAK status. The host

can always see if the module has additional packets to send by either reading the RX Status byte of the currently sent packet or by sending an RXSTATUS command. Once the module receives the ACK status from the host it will send the next packet in it buffer. If the module begins to lose packets, it will set the Overflow flag in the RX Status byte (TABLE 19: RXSTATUS RETURN CODES).

5.0 POWER-ON STATE

The LoRa mPCIe Smart Gateway Card powers on in one of the following modes based on the model.

- mPCIe Host USB Mode (SG9001)
- Embedded Host Mode with dedicated UART (SG9002)

5.1 mPCIe Host USB Mode (SG9001)

The module communicates with the host system using the USB signals on the mPCIe connector. When powered up, the card will perform the required USB handshake with the host's USB controller and enumerate as a CDC (Communications Device Class). This will make the card appear to the host as a serial communication device.

5.2 Embedded Host Mode (SG9002)

When the Module boots up in this mode, it will configure the module for Embedded Host Mode. In this mode, the Host UART interface will be initialized and configured for sending and receiving commands from the embedded host micro controller. For a list of commands refer to Section 2.0.

5.3 Enable Module RF (RF_ENABLE):

During operation, the module will check the RF_ENABLE pins state. If the pin is Low it will disable the RF Transceiver on the module. If the Pin is High it will Enable the RF Transceiver.

5.4 Reset Module (RESET):

The RESET pin is used to reset the LoRa mPCIe Smart Gateway Card. A low level pulse on RESET will instruct the module perform a software reset. After reset the module will perform a full system initialize the system with the current settings stored in the EEPROM.

6.0 Using BOOTLOADER

Bootloader uses the same communication protocol as the main application. The following commands support application firmware programming.

TABLE 20: COMMAND LIST (BOOTLOADER ONLY)

Command	Value	Direction	Command Description
ERASE_APP	0x10	Host -> Module	Erases the entire application program memory (Bootloader program is protected)
WRITE_WORD	0x11	Host -> Module	Programs a memory instruction word (3 bytes)
WRITE_ROW	0x12	Host -> Module	Programs a memory row (64 Instructions == 192 bytes)

READ_ROW	0x14	Host -> Module	Returns 192 bytes of row data
VALID_APP	0x15	Host -> Module	Programs unique identifier (0xA5) at memory address 0x14FFE indicating valid firmware signal to the bootloader.
RESET	0x16	Host -> Module	Resets the module (Needs to follow after a VALID_APP command to switch to the main application)
INVALID	0xFF	Module -> Host	Previous command received from Host does not exist

More information about the commands are mentioned in the following sections.

6.1 Erase Application Program Memory (ERASE_APP):

ERASE_APP command is used to erase the entire application memory. After the module is placed in bootloader mode(as described in 3.22), the user needs to erase the current application firmware and reprogram the memory with new application firmware.

Note: The user cannot just send a VALID_APP command to use the previous application; application firmware needs to be programmed prior to sending VALID_APP command

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<10>	<00><00>	None	Checksum	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<10>	<01><00>	<00>	<xx>	<0D>

6.2 Program an instruction (WRITE_WORD):

WRITE_WORD is used to program 3 bytes of instruction.

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<11>	<07><00>	<addr0><addr1><addr2><addr3><data0><data1><data2>	Checksum	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**

<23>	<11>	<01><00>	<00>	<xx>	<0D>

Data Payload = Program Memory address (4 bytes) + Program Memory Instruction (3 bytes)
Example: For address 0x0000C000 use <0x00> <0xC0> <0x00> <0x00>

6.3 Program a Row of memory(WRITE_ROW):

WRITE_ROW is used to program 64 instructions (192 bytes).

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<12>	<C4><00>	<addr0><addr1><addr2><addr3><DATA[0:192]>	Checksum	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<12>	<01><00>	<00>	<xx>	<0D>

Data Payload = Program Memory address (4 bytes) + Program Memory Instructions (192 bytes)
Example: For address 0x0000C000 use <0x00> <0xC0> <0x00> <0x00>

6.4 Read a row of memory (READ_ROW):

READ ROW returns 192 bytes of program memory.

Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**

<23>	<14>	<04><00>	<addr0><addr1><addr2><addr3>	Checksum	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<14>	<C0><00>	Data Payload	<xx>	<0D>

6.5 RESET Command (RESET):

The RESET command is used to RESET the module. After sending VALID_APP command, use RESET to switch to the main application.

Host -> Module:

Frame Start	Command	Length	Data Payload	Checksum	CR
<23>	<16>	<00><00>	None	Checksum	<0D>

6.6 Valid application (VALID_APP):

The VALID_APP command is used to indicate to the bootloader that application programming is complete and then after a RESET command the control can switch to the main application

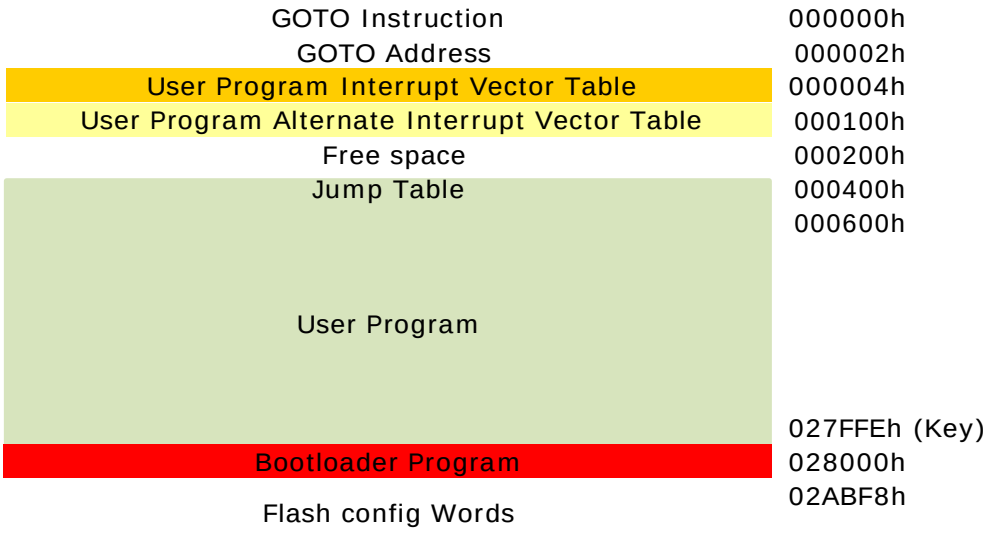
Host -> Module:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<15>	<00><00>	NA	Checksum	<0D>
Module -> Host:					
Frame Start	**Command**	**Length**	**Data Payload**	**Checksum**	**CR**
<23>	<15>	<01><00>	<00>	<xx>	<0D>

7.0 PIC24FJ256GB206 Application Memory Map

Table 21: Memory Map

Page size: 512 Instructions = 768 Bytes

Memory Map:



02AC00h