1.3. Pin Diagram

The following diagram shows the pin arrangement , top view.

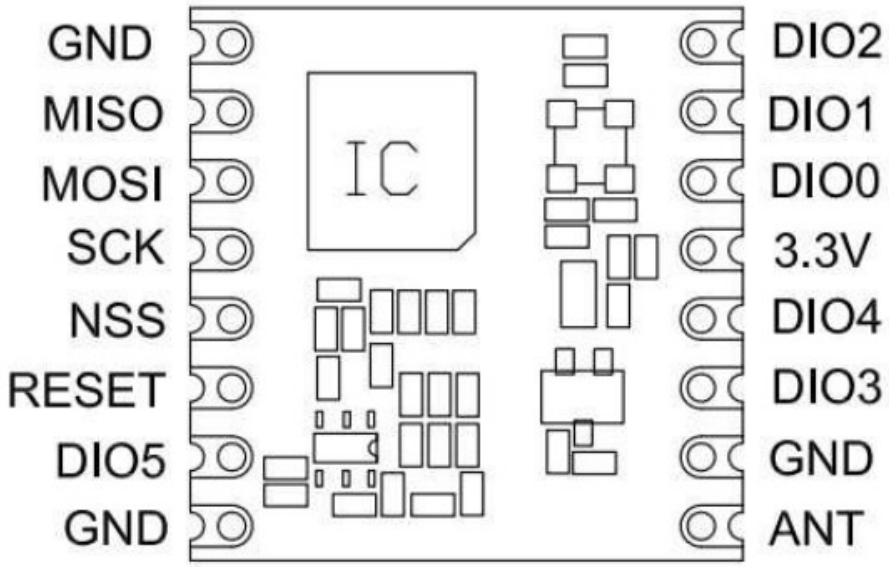

Figure 2. Pin Diagrams

1.4. Pin Description

Number	Name	Type	Description Description Stand Alone Mode
1	GND	-	Ground
2	MISO	I	SPI Data output
3	MOSI	O	SPI Data input
4	SCK	I	SPI Clock input
5	NSS	I	SPI Chip select input
6	RESET	I/O	Reset trigger input
7	DIO5	I/O	Digital I/O, software configured
8	GND	-	Ground
9	ANT	-	RF signal output/input.
10	GND	-	Ground
11	DIO3	I/O	Digital I/O, software configured
12	DIO4	I/O	Digital I/O, software configured
13	3.3V	-	Supply voltage
14	DIO0	I/O	Digital I/O, software configured
15	DIO1	I/O	Digital I/O, software configured
16	DIO2	I/O	Digital I/O, software configured

3. RFM95/96/97/98(W) Features

This section gives a high-level overview of the functionality of the RFM95/96/97/98(W) low-power, highly integrated transceiver. The following figure shows a simplified block diagram of the RFM95/96/97/98(W).

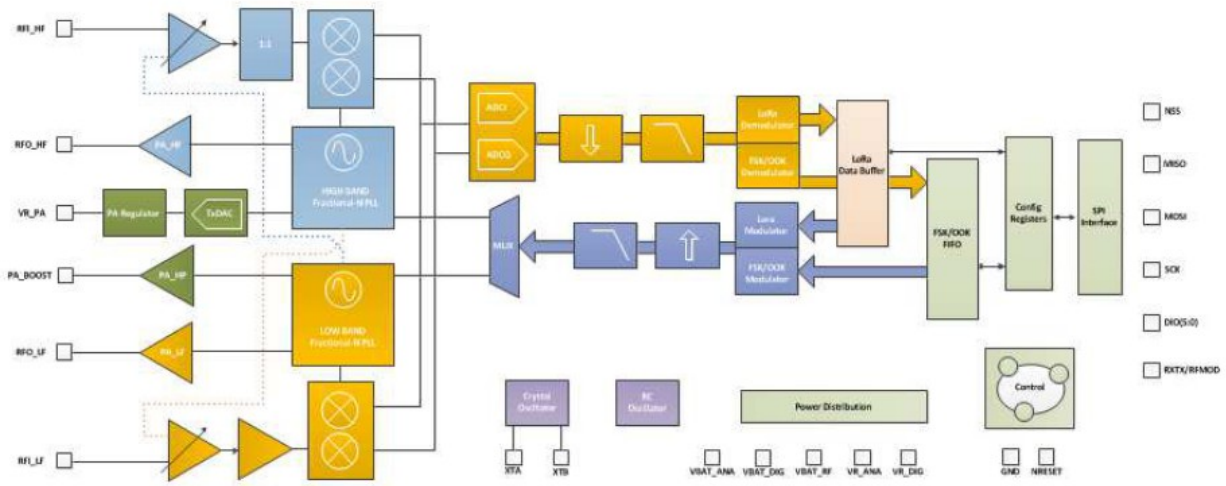

Figure 3. RFM95/96/97/98(W) Block Schematic Diagram

4.1.1.6. LoRa™ Packet Structure

The LoRa™ modem employs two types of packet format, explicit and implicit. The explicit packet includes a short header that contains information about the number of bytes, coding rate and whether a CRC is used in the packet. The packet format is shown in the following figure.

The LoRa™ packet comprises three elements:

- A preamble.
- An optional header.
- The data payload.

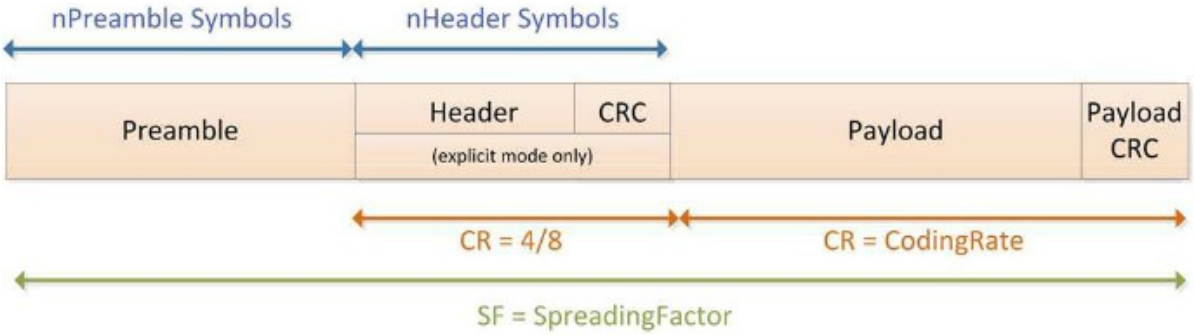

Figure 5. LoRa™ *Packet Structure*

Preamble

The preamble is used to synchronize receiver with the incoming data flow. By default the packet is configured with a 12 symbol long sequence. This is a programmable variable so the preamble length may be extended, for example in the interest of reducing to receiver duty cycle in receive intensive applications. However, the minimum length suffices for all communication. The transmitted preamble length may be changed by setting the register *PreambleLength* from 6 to 65535, yielding total preamble lengths of 6+4 to 65535+4 symbols, once the fixed overhead of the preamble data is considered. This permits the transmission of a near arbitrarily long preamble sequence.

The receiver undertakes a preamble detection process that periodically restarts. For this reason the preamble length should be configured identical to the transmitter preamble length. Where the preamble length is not known, or can vary, the maximum preamble length should be programmed on the receiver side.

Header

Depending upon the chosen mode of operation two types of header are available. The header type is selected by the *ImplictHeaderMode* bit found within the *RegSymbTimeoutMsb* register.

Explicit Header Mode

This is the default mode of operation. Here the header provides information on the payload, namely:

- The payload length in bytes.
- The forward error correction code rate
- The presence of an optional 16-bits CRC for the payload.

The header is transmitted with maximum error correction code (4/8). It also has its own CRC to allow the receiver to discard invalid headers.

Implicit Header Mode

In certain scenarios, where the payload, coding rate and CRC presence are fixed or known in advance, it may be advantageous to reduce transmission time by invoking implicit header mode. In this mode the header is removed from the packet. In this case the payload length, error coding rate and presence of the payload CRC must be manually configured on both sides of the radio link.

Note *With SF = 6 selected, implicit header mode is the only mode of operation possible.*

Payload

The packet payload is a variable-length field that contains the actual data coded at the error rate either as specified in the header in explicit mode or in the register settings in implicit mode. An optional CRC may be appended. For more information on the payload and how it is loaded from the data buffer FIFO please see Section 4.1.2.3.

4.1.2. LoRa™ Digital Interface

The LoRa™ modem comprises three types of digital interface, static configuration registers, status registers and a FIFO data buffer. All are accessed through the RFM95/96/97/98(W)'s SPI interface - full details of each type of register are given below. Full listings of the register addresses used for SPI access are given in Section 6.4.

4.1.2.1. LoRa™ Configuration Registers

Configuration registers are accessed through the SPI interface. Registers are readable in all device mode including Sleep. However, they should be **written only in Sleep and Stand-by modes**. Please note that **the automatic top level sequencer (TLS modes) are not available in** LoRa™ **mode** and **the configuration register mapping changes as shown in Table** 85**.** The content of the LoRa™ configuration registers is retained in FSK/OOK mode. For the functionality of mode registers common to both FSK/OOK and LoRa™ mode, please consult the Analog and RF Front End section of this document (Section 5).

4.1.2.2. Status Registers

Status registers provide status information during receiver operation.

4.1.2.3. LoRa™ Mode FIFO Data Buffer

Overview

The RFM95/96/97/98(W) is equipped with a 256 byte RAM data buffer which is uniquely accessible in LoRa mode. This RAM area, thereafter reffered to as the FIFO Data buffer, is fully customizable by the user and allows access to the received, or to be transmitted, data. All access to the LoRa™ FIFO data buffer is done via the SPI interface. A diagram of the user defined memory mapping of the FIFO data buffer is shown below. These FIFO data buffer can be read in all operating modes except sleep and store data related to the last receive operation performed. It is automatically cleared of old content upon each new transition to receive mode.

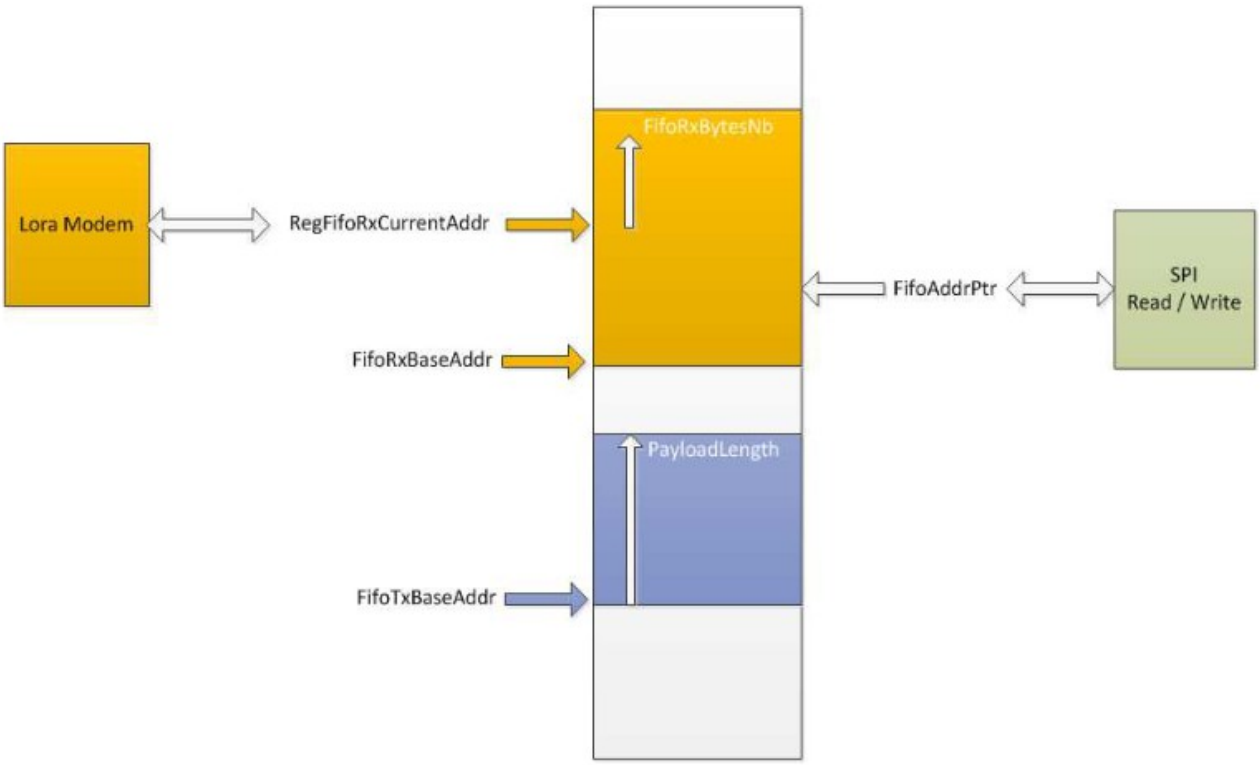

Figure 7. LoRa™ *data buffer*

Principle of Operation

Thanks to its dual port configuration, it is possible to simultaneously store both transmit and receive information in the FIFO data buffer. The register *FifoTxBaseAddr* specifies the point in memory where the transmit information is stored. Similarly, for receiver operation, the register *FifoRxBaseAddr* indicates the point in the data buffer where information will be written to in event of a receive operation.

By default, the device is configured at power-up so that half of the available memory is dedicated to Rx (*FifoRxBaseAddr* initialized at address 0x00) and the other half is dedicated for Tx (*FifoTxBaseAddr* initialized at address 0x80).

However, due to the contiguous nature of the FIFO data buffer, the base addresses for Tx and Rx are fully configurable across the 256 byte memory area. Each pointer can be set independently anywhere within the FIFO. To exploit the maximum FIFO data buffer size in transmit or receive mode, the whole FIFO data buffer can be used in each mode by setting the base addresses *FifoTxBaseAddr* and *FifoRxBaseAddr* at the bottom of the memory (0x00).

The FIFO data buffer is cleared when the device is put in SLEEP mode, consequently no access to the FIFO data buffer is possible in sleep mode. However, the data in the FIFO data buffer are retained when switching across the other LoRa modes of operation, so that a received packet can be retransmitted with minimum data handling on the controller side. The FIFO data buffer is not self-clearing (unless if the device is put in sleep mode) and the data will only be "erased" when a new set of data is written into the occupied memory location.

The actual location to be read from, or written to, over the SPI interface is defined by the address pointer *FifoAddrPtr*. Before any read or write operation it is hence necessary to initialise this pointer to the corresponding base value. Upon reading or writing to the FIFO data buffer (*RegFifo*) the address pointer will then increment automatically.

The register *FifoRxBytesNb* defines the size of the memory location to be written in the event of a successful receive operation. On the other hand *PayloadLength* indicates the size of the memory location to be transmitted. In implicit header mode, the *FifoRxBytesNb* is not used as the number of payload bytes is known. Otherwise, in explicit header mode, the initial size of the receive buffer is set to the packet length in the received header. The variable *FifoRxCurrentAddr* indicates the location of the last packet received in the FIFO so that the last packet received can be easily read by pointing the *FifoAddrPtr* to this register.

It is important to notice that all the received data will be written to the FIFO data buffer even if the CRC is invalid. This allows for post-processing of received data for debug purposes for instance. It is also imporant to note that when receiving, if the packet size exceeds the buffer memory allocated for the Rx it will overwrite the transmit portion of the data buffer.

4.1.3. Operation of the LoRa™ Modem

4.1.3.1. Operating Mode Control

The operating modes of the LoRa™ modem are accessed by enabling LoRa™ mode (setting the *LongRangeMode* bit of *RegOpMode*). Depending upon the operating mode selected the range of functionality and register access is given by the following table:

Table 61 LoRa™ Operating Mode Functionality

Operating Mode	Description
SLEEP	Low-power mode. In this mode only SPI and configuration registers are accessible. Lora FIFO is not accessible. Note that this is the only mode permissible to switch between FSK/OOK mode and LoRa mode.
STAND-BY	both Crystal oscillator and Lora baseband blocks are turned on.RF part and PLLs are disabled
FSTX	This is a frequency synthesis mode for transmission. The PLL selected for transmission is locked and active at the transmit frequency. The RF part is off.
FSRX	This is a frequency synthesis mode for reception. The PLL selected for reception is locked and active at the receive frequency. The RF part is off.
TX	When activated the RFM95/96/97/98(W) powers all remaining blocks required for transmit, ramps the PA, transmits the packet and returns to Stand-by mode.
RXCONTINUOUS	When activated the RFM95/96/97/98(W) powers all remaining blocks required for reception, processing all received data until a new user request is made to change operating mode.
RXSINGLE	When activated the RFM95/96/97/98(W) powers all remaining blocks required for reception, remains in this state until a valid packet has been received and then returns to Stand-by mode.
CAD	When in CAD mode, the device will check a given channel to detect LoRa preamble signal

It is possible to access any mode from any other mode by changing the value in the *RegOpMode* register.

4.1.5. LoRa™ Modem State Machine Sequences

The sequence for transmission and reception of data to and from the LoRa™ modem, together with flow charts of typical sequences of operation, are detailed below.

Data Transmission Sequence

In transmit mode power consumption is optimized by enabling RF, PLL and PA blocks only when packet data needs to be transmitted. Figure 8 shows a typical LoRa™ transmit sequence.

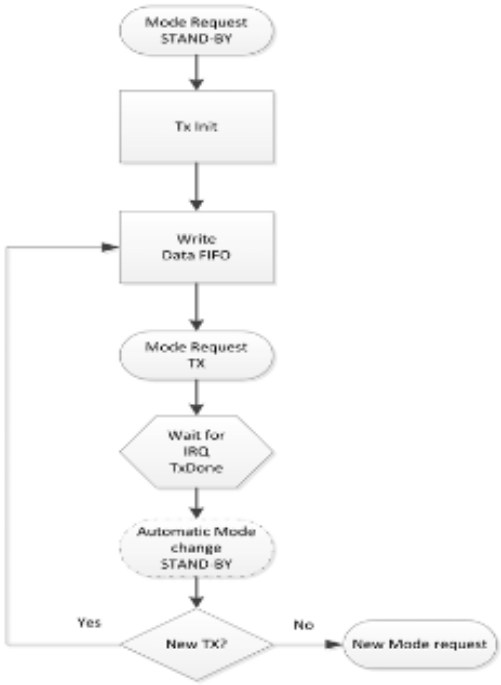

Figure 8. LoRa™ *modulation transmission sequence.*

- Static configuration registers can only be accessed in Sleep mode, Stand-by mode or FSTX mode.
- The LoRa™ FIFO can only be filled in Stand-by mode.
- Data transmission is initiated by sending TX mode request.
- Upon completion the *TxDone* interrupt is issued and the radio returns to Stand-by mode.
- Following transmission the radio can be manually placed in Sleep mode or the FIFO refilled for a subsequent Tx operation.

LoRa™ Transmit Data FIFO Filling

In order to write packet data into FIFO user should:

1. Set *FifoPtrAddr* to *FifoTxPtrBase*.
2. Write *PayloadLength* bytes to the FIFO (*RegFifo*)

Data Reception Sequence

Figure 9 shows typical LoRa™ receive sequences for both single and continuous receiver modes of operation.

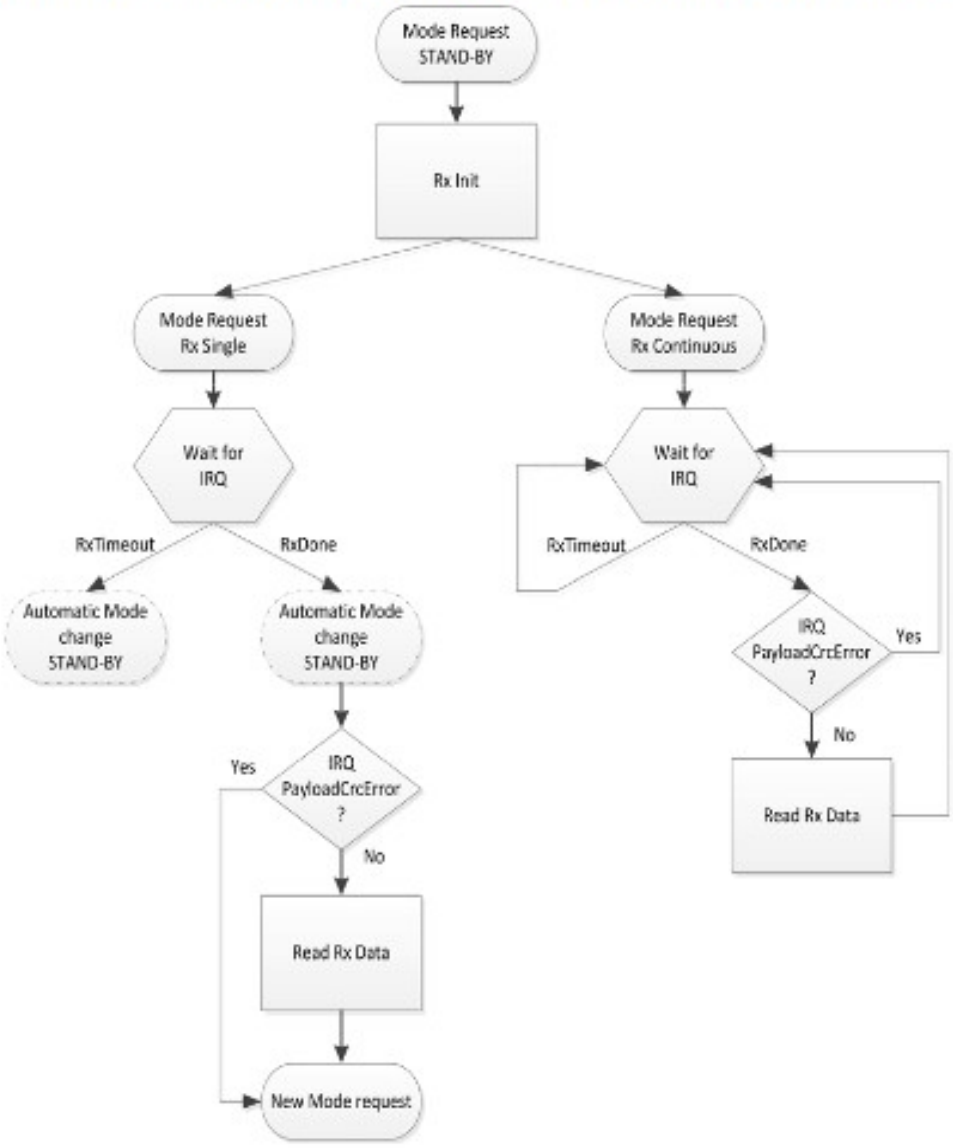

Figure 9. LoRa™ receive sequence.

The LORA receive modem can work in two distinct mode

1. Single receive mode
2. Continuous receive mode

Those two modes correspond to different use cases and it is important to understand the subtle differences between them.

Single Reception Operating Mode

In this mode, the modem searches for a preamble during a given time window. If a preamble hasn't been found at the end of the time window, the chip generates the RxTimeout interrupt and goes back to stand-by mode . The length of the window (in symbols) is defined by the RegSymbTimeout register and should be in the range of 4 (minimum time for the modem to acquire lock on a preamble) up to 1023 symbols. (The default value being 5). If no preamble is detected during this window the RxTimeout interrupt is generated and the radio goes back to stand-by mode.

At the end of the payload, the RxDone interrupt is generated together with the interrupt PayloadCrcError if the payload CRC is not valid. However, even when the CRC is not valid, the data are written in the FIFO data buffer for post processing. Following the RxDone interrupt the radio goes to stand-by mode.

The modem will also automatically return in stand-by mode when the interrupts RxDone or RxTimeout are generated. Therefore, this mode should only be used when the time window of arrival of the packet is known . In other cases, the RX continuous mode should be used.

In Rx single mode low-power is achieved by turning off PLL and RF blocks as soon as a packet has been received. The flow is as follows:

1 *Set FifoPtrAddr* to *FifoRxPtrBase*.

2 Static configuration register device can be written in either Sleep mode, Stand-by mode or FSRX mode.

3 A single packet receive operation is initiated by selecting the operating mode RXSINGLE.

4 The receiver will then await the reception of a valid preamble. Once received, the gain of the receive chain is set. Following the ensuing reception of a valid header, indicated by the ValidHeader interrupt in explicit mode. The packet reception process commences. Once the reception process is complete the RxDone interrupt is set. The radio then returns automatically to Stand-by mode to reduce power consumption.

5 The receiver status register PayloadCrcError should be checked for packet payload integrity.

6 If a valid packet payload has been received then the FIFO should be read (See Payload Data Extraction below). Should a subsequent single packet reception need to be triggered, then the RXSINGLE operating mode must be re-selected to launch the receive process again - taking care to reset the SPI pointer (FifoPtrAddr) to the base location in memory (FifoRxPtrBase).

Continuous Reception Operating Mode

In continuous receive mode the modem scans the channel continuously for a preamble. Each time a preamble is detected the modem detects and tracks it until the packet is received and then carries on waiting for the next preamble.

If the preamble length exceeds the anticipated value set by the registers RegPreambleMsb and RegPreambleLsb (measured in symbol unit), the preamble will be dropped and the search for a preamble restarted. However, this scenario will not be flagged by an interrupt. In continous RX mode, opposite to the single RX mode, when a timeout interrupt is generated, the device will not go in standby mode. In this case, the user must simply clear the interrupt while the device carry on waiting for a valid preamble.

It is also important to note that the demodulated bytes are written in the data buffer memory in the order received. Meaning, the first byte of a new packet is written just after the last byte of the preceding packet. The RX modem address pointer is never reseted as long as this mode is enabled. It is therefore necessary for the controller to handle the address pointer to make sure the FIFO data buffer is never full.

In continuous mode the received packet processing sequence is given below.

1 Whilst in Sleep or Stand-by mode select RXCONT mode.
2 Upon reception of a valid header CRC the *RxDone* interrupt is set. The radio remains in RXCONT mode waiting for the next RX LoRa™ packet.
3 The *PayloadCrcError* flag should be checked for packet integrity.
4 If packet has been correctly received the FIFO data buffer can be read (see below).
5 The reception process (steps 2 - 4) can be repeated or receiver operating mode exited as desired.

In continuous mode status information are available only for the last packet received, i.e. the corresponding registers should be read before the next RxDone arrives.

Payload Data Extraction from FIFO

In order to retrieve received data from FIFO the user must ensure that *ValidHeader*, *PayloadCrcError*, *RxDone* and *RxTimeout* interrupts in the status register RegIrqFlags are not asserted to ensure that packet reception has terminated successfully (i.e. no flags should be set).

In case of errors the steps below should be skipped and the packet discarded. In order to retrieve valid received data from the FIFO the user must:

- *FifoNbRxBytes* Indicates the number of bytes that have been received thus far.
- *RegRxDataAddr* Is a dynamic pointer that indicates precisely where the Lora modem received data has been written up to.
- Set *FifoPtrAddr* to *FifoRxCurrentAddr*. This sets the FIFO pointer to the the location of the last packet received in the FIFO. The payload can then be extracted by reading the *RegFifo* address *RegNbRxBytes* times. Alternatively, it is possible to manually point to the location of the last packet received from the start of the current packet by setting *FifoPtrAddr* to *RegRxDataAddr - FifoNbRxBytes*. In the same way, packet bytes can then be extracted from FIFO by reading the *RegFifo* address *RegNbRxBytes* times.

4.3. SPI Interface

The SPI interface gives access to the configuration register via a synchronous full-duplex protocol corresponding to CPOL = 0 and CPHA = 0 in Motorola/Freescale nomenclature. Only the slave side is implemented.

Three access modes to the registers are provided:

- SINGLE access: an address byte followed by a data byte is sent for a write access whereas an address byte is sent and a read byte is received for the read access. The NSS pin goes low at the beginning of the frame and goes high after the data byte.
- BURST access: the address byte is followed by several data bytes. The address is automatically incremented internally between each data byte. This mode is available for both read and write accesses. The NSS pin goes low at the beginning of the frame and stay low between each byte. It goes high only after the last byte transfer.
- FIFO access: if the address byte corresponds to the address of the FIFO, then succeeding data byte will address the FIFO. The address is not automatically incremented but is memorized and does not need to be sent between each data byte. The NSS pin goes low at the beginning of the frame and stay low between each byte. It goes high only after the last byte transfer.

The figure below shows a typical SPI single access to a register.

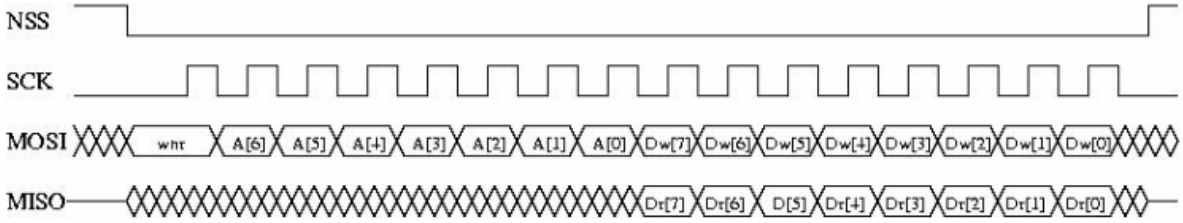

Figure 38. SPI Timing Diagram (single access)

MOSI is generated by the master on the falling edge of SCK and is sampled by the slave (i.e. this SPI interface) on the rising edge of SCK. MISO is generated by the slave on the falling edge of SCK.

A transfer is always started by the NSS pin going low. MISO is high impedance when NSS is high.

The second byte is a data byte, either sent on MOSI by the master in case of a write access or received by the master on MISO in case of read access. The data byte is transmitted MSB first.

Proceeding bytes may be sent on MOSI (for write access) or received on MISO (for read access) without a rising NSS edge and re-sending the address. In FIFO mode, if the address was the FIFO address then the bytes will be written / read at the FIFO address. In Burst mode, if the address was not the FIFO address, then it is automatically incremented for each new byte received.

The frame ends when NSS goes high. The next frame must start with an address byte. The SINGLE access mode is therefore a special case of FIFO / BURST mode with only 1 data byte transferred.

During the write access, the byte transferred from the slave to the master on the MISO line is the value of the written register before the write operation.

6. Description of the Registers

The register mapping depends upon whether FSK/OOK or LoRa™ mode has been selected. The following table summarises the location and function of each register and gives an overview of the changes in register mapping between both modes of operation.

6.1. Register Table Summary

Table 85 Registers Summary

Address	Register Name		Reset (POR)	Default (FSK)	Description	
	FSK/OOK Mode	LoRa™ Mode			FSK Mode	LoRa™ Mode
0x00	RegFifo		0x00		FIFO read/write access	
0x01	RegOpMode		0x01		Operating mode & LoRa™ / FSK selection	
0x02	RegBitrateMsb	Unused	0x1A		Bit Rate setting, Most Significant Bits	
0x03	RegBitrateLsb		0x0B		Bit Rate setting, Least Significant Bits	
0x04	RegFdevMsb		0x00		Frequency Deviation setting, Most Significant Bits	
0x05	RegFdevLsb		0x52		Frequency Deviation setting, Least Significant Bits	
0x06	RegFrfMsb		0xE4		RF Carrier Frequency, Most Significant Bits	
0x07	RegFrfMid		0xC0		RF Carrier Frequency, Intermediate Bits	
0x08	RegFrfLsb		0x00		RF Carrier Frequency, Least Significant Bits	
0x09	RegPaConfig		0x0F		PA selection and Output Power control	
0x0A	RegPaRamp		0x19		Control of PA ramp time, low phase noise PLL	
0x0B	RegOcp		0x2B		Over Current Protection control	
0x0C	RegLna		0x20		LNA settings	
0x0D	RegRxConfig	RegFifoAddrPtr	0x08	0x0E	AFC, AGC, ctrl	FIFO SPI pointer
0x0E	RegRssiConfig	RegFifoTxBaseAddr	0x02		RSSI	Start Tx data
0x0F	RegRssiCollision	RegFifoRxBaseAddr	0x0A		RSSI Collision detector	Start Rx data
0x10	RegRssiThresh	RegIrqFlags	0xFF		RSSI Threshold control	LoRa™ state flags
0x11	RegRssiValue	RegIrqFlagsMask	-		RSSI value in dBm	Optional flag mask
0x12	RegRxBw	RegFreqIfMsb	0x15		Channel Filter BW Control	IF Frequency
0x13	RegAfcBw	RegFreqIFLsb	0x0B		AFC Channel Filter BW	
0x14	RegOokPeak	RegSymbTimeoutMsb	0x28		OOK demodulator	Receiver timeout value
0x15	RegOokFix	RegSymbTimeoutLsb	0x0C		Threshold of the OOK demod	
0x16	RegOokAvg	RegTxCfg	0x12		Average of the OOK demod	LoRa™ transmit parameters
0x17	Reserved17	RegPayloadLength	0x47		-	
0x18	Reserved18	RegPreambleMsb	0x32		-	Size of preamble
0x19	Reserved19	RegPreambleLsb	0x3E		-	
0x1A	RegAfcFei	RegModulationCfg	0x00		AFC and FEI control	Modem PHY config
0x1B	RegAfcMsb	RegRfMode	0x00		Frequency correction value of the AFC	Test register
0x1C	RegAfcLsb	RegHopPeriod	0x00			FHSS Hop period

Address	FSK/OOK Mode	LoRa™ Mode	(POR)	(FSK)	FSK Mode	LoRa™ Mode
0x1D	RegFeiMsb	RegNbRxBytes	0x00		Value of the calculated frequency error	Number of received bytes
0x1E	RegFeiLsb	RegRxHeaderInfo	0x00			Info from last header
0x1F	RegPreambleDetect	RegRxHeaderCntValue	0x40	0xAA	Settings of the Preamble Detector	Number of valid headers received
0x20	RegRxTimeout1	RegRxPacketCntValue	0x00		Timeout Rx request and RSSI	Number of valid packets received
0x21	RegRxTimeout2	RegModemStat	0x00		Timeout RSSI and *PayloadReady*	Live LoRa™ modem status
0x22	RegRxTimeout3	RegPktSnrValue	0x00		Timeout RSSI and *SyncAddress*	Espimation of last packet SNR
0x23	RegRxDelay	RegRssiValue	0x00		Delay between Rx cycles	Current RSSI
0x24	RegOsc	RegPktRssiValue	0x05	0x07	RC Oscillators Settings, CLKOUT frequency	RSSI of last packet
0x25	RegPreambleMsb	RegHopChannel	0x00		Preamble length, MSB	FHSS start channel
0x26	RegPreambleLsb	RegRxDataAddr	0x03		Preamble length, LSB	LoRa™ rx data pointer
0x27	RegSyncConfig	RESERVED	0x93		Sync Word Recognition control	RESERVED
0x28-0x2F	RegSyncValue1-8		0x55	0x01	Sync Word bytes, 1 through 8	
0x30	RegPacketConfig1		0x90		Packet mode settings	
0x31	RegPacketConfig2		0x40		Packet mode settings	
0x32	RegPayloadLength		0x40		Payload length setting	
0x33	RegNodeAdrs	RESERVED	0x00		Node address	RESERVED
0x34	RegBroadcastAdrs		0x00		Broadcast address	
0x35	RegFifoThresh		0x0F	0x8F	Fifo threshold, Tx start condition	
0x36	RegSeqConfig1		0x00		Top level Sequencer settings	
0x37	RegSeqConfig2		0x00		Top level Sequencer settings	
0x38	RegTimerResol		0x00		Timer 1 and 2 resolution control	
0x39	RegTimer1Coef		0xF5		Timer 1 setting	
0x3A	RegTimer2Coef		0x20		Timer 2 setting	
0x3B	RegImageCal		0x82	0x02	Image calibration engine control	
0x3C	RegTemp		-		Temperature Sensor value	
0x3D	RegLowBat		0x02		Low Battery Indicator Settings	
0x3E	RegIrqFlags1		0x80		Status register: PLL Lock state, Timeout, RSSI	
0x3F	RegIrqFlags2		0x40		Status register: FIFO handling flags, Low Battery	
0x40	RegDioMapping1		0x00		Mapping of pins DIO0 to DIO3	
0x41	RegDioMapping2		0x00		Mapping of pins DIO4 and DIO5, ClkOut frequency	
0x42	RegVersion		0x11		Hope RF ID relating the silicon revision	
0x44	RegPllHop	Unused	0x2D		Control the fast frequency hopping mode	Unused
0x4B	RegTcxo		0x09		TCXO or XTAL input setting	

Address	Register Name		Reset (POR)	Default (FSK)	Description	
	FSK/OOK Mode	LoRa™ Mode			FSK Mode	LoRa™ Mode
0x4D	RegPaDac		0x84		Higher power settings of the PA	
0x5B	RegFormerTemp		-		Stored temperature during the former IQ Calibration	
0x5D	RegBitRateFrac	Unused	0x00		Fractional part in the Bit Rate division ratio	Unused
0x61	RegAgcRef		0x13		Adjustment of the AGC thresholds	
0x62	RegAgcThresh1		0x0E			
0x63	RegAgcThresh2		0x5B			
0x64	RegAgcThresh3		0xDB			
others	RegTest		-		Internal test registers. Do not overwrite	

Note - Reset values are automatically refreshed in the chip at Power On Reset

- Default values are the Hope RF recommended register values, optimizing the device operation

*- Registers for which the Default value differs from the Reset value are denoted by a * in the tables of section 6.2*

6.4. LoRa™ Mode Register Map

This section details the RFM95/96/97/98(W) register mapping and the precise contents of each register in LoRa™ mode.

It is essential to understand that the LoRa modem is controlled independently of the FSK modem. Therefore, care should be taken when accessing the registers, especially as some register may have the same name in LoRa or FSK mode.

The LoRa registers are only accessible when the device is set in Lora mode (and, in the same way, the FSK register are only accessible in FSK mode). However, in some cases, it may be necessary to access some of the FSK register while in LoRa mode. To this aim, the *AccesSharedReg* bit was created in the *RegOpMode* register. This bit, when set to '1', will grant access to the FSK register 0x0D up to the register 0x3F. Once the setup has been done, it is strongly recommended to clear this bit so that LoRa register can be accessed normally.

Convention: r: read, w: write, c : set to clear and t: trigger.

Name (Address)	Bits	Variable Name	Mode	Reset	LoRa™ Description
RegFifo (0x00)	7-0	Fifo	rw	0x00	LoRa™ base-band FIFO data input/output. FIFO is cleared an not accessible when device is in SLEEP mode
Common Register Settings					
RegOpMode (0x01)	7	LongRangeMode	rw	0x0	0 → FSK/OOK Mode 1 → LoRa™ Mode This bit can be modified only in Sleep mode. A write operation on other device modes is ignored.
	6	AccessSharedReg	rw	0x0	This bit operates when device is in Lora mode; if set it allows access to FSK registers page located in address space (0x0D:0x3F) while in LoRa mode 0 → Access LoRa registers page 0x0D: 0x3F 1 → Access FSK registers page (in mode LoRa) 0x0D: 0x3F
	5-4	reserved	r	0x00	reserved
	3	LowFrequencyModeOn	rw	0x01	Access Low Frequency Mode registers 0 → High Frequency Mode (access to HF test registers) 1 → Low Frequency Mode (access to LF test registers)
	2-0	Mode	rwt	0x01	Device modes 000 → SLEEP 001 → STDBY 010 → Frequency synthesis TX (FSTX) 011 → Transmit (TX) 100 → Frequency synthesis RX (FSRX) 101 → Receive continuous (RXCONTINUOUS) 110 → receive single (RXSINGLE) 111 → Channel activity detection (CAD)
(0x02)	7-0	reserved	r	0x00	-
(0x03)	7-0	reserved	r	0x00	-
(0x04)	7-0	reserved	rw	0x00	-
(0x05)	7-0	reserved	r	0x00	-
RegFrMsb (0x06)	7-0	Frf(23:16)	rw	0x6c	MSB of RF carrier frequency
RegFrMid (0x07)	7-0	Frf(15:8)	rw	0x80	MSB of RF carrier frequency

Name (Address)	Bits	Variable Name	Mode	Reset	LoRa™ Description
RegFrLsb (0x08)	7-0	Frf(7:0)	rwt	0x00	LSB of RF carrier frequency $f_{RF} = \frac{(XOSC) \cdot Frf}{2^{19}}$ Resolution is 61.035 Hz if F(XOSC) = 32 MHz. Default value is 0x6c8000 = 434 MHz. Register values must be modified only when device is in SLEEP or STAND-BY mode.
Registers for RF blocks					
RegPaConfig (0x09)	7	PaSelect	rw	0x00	Selects PA output pin 0 → RFO pin. Output power is limited to +14 dBm. 1 → PA_BOOST pin. Output power is limited to +20 dBm
	6-4	MaxPower	rw	0x04	Select max output power: Pmax=10.8+0.6*MaxPower [dBm]
	3-0	OutputPower	rw	0x0f	Pout=Pmax-(15-OutputPower) if PaSelect = 0 (RFO pin) Pout=17-(15-OutputPower) if PaSelect = 1 (PA_BOOST pin)
RegPaRamp (0x0A)	7-5	unused	r	-	unused
	4	reserved	rw	0x00	reserved
	3-0	PaRamp(3:0)	rw	0x09	Rise/Fall time of ramp up/down in FSK 0000 → 3.4 ms 0001 → 2 ms 0010 → 1 ms 0011 → 500 us 0100 → 250 us 0101 → 125 us 0110 → 100 us 0111 → 62 us 1000 → 50 us 1001 → 40 us 1010 → 31 us 1011 → 25 us 1100 → 20 us 1101 → 15 us 1110 → 12 us 1111 → 10 us
RegOcp (0x0B	7-6	unused	r	0x00	unused
	5	OcpOn	rw	0x01	Enables overload current protection (OCP) for PA: 0 → OCP disabled 1 → OCP enabled
	4-0	OcpTrim	rw	0x0b	Trimming of OCP current: Imax = 45+5*OcpTrim [mA] if OcpTrim <= 15 (120 mA) / Imax = -30+10*OcpTrim [mA] if 15 < OcpTrim <= 27 (130 to 240 mA) Imax = 240mA for higher settings Default Imax = 100mA

Name (Address)	Bits	Variable Name	Mode	Reset	LoRa™ Description
RegLna (0x0C)	7-5	LnaGain	rwx	0x01	LNA gain setting: 000 → not used 001 → G1 = maximum gain 010 → G2 011 → G3 100 → G4 101 → G5 110 → G6 = minimum gain 111 → not used
	4-3	LnaBoostLf	rw	0x00	Low Frequency (RFI_LF) LNA current adjustment 00 → Default LNA current Other → Reserved
	2	reserved	rw	0x00	reserved
	1-0	LnaBoostHf	rw	0x00	High Frequency (RFI_HF) LNA current adjustment 00 → Default LNA current 11 → Boost on, 150% LNA current
Lora page registers					
RegFifoAddrPtr (0x0D)	7-0	FifoAddrPtr	rw	0x00	SPI interface address pointer in FIFO data buffer.
RegFifoTxBaseAddr (0x0E)	7-0	FifoTxBaseAddr	rw	0x80	write base address in FIFO data buffer for TX modulator
RegFifoRxBaseAddr (0x0F)	7-0	FifoRxBaseAddr	rw	0x00	read base address in FIFO data buffer for RX demodulator
RegFifoRxCurrentAddr (0x10)	7-0	FifoRxCurrentAddr	r	n/a	Start address (in data buffer) of last packet received
RegIrqFlagsMask (0x11)	7	RxTimeoutMask	rw	0x00	Timeout interrupt mask: setting this bit masks the corresponding IRQ in RegIrqFlags
	6	RxDoneMask	rw	0x00	Packet reception complete interrupt mask: setting this bit masks the corresponding IRQ in RegIrqFlags
	5	PayloadCrcErrorMask	rw	0x00	Payload CRC error interrupt mask: setting this bit masks the corresponding IRQ in RegIrqFlags
	4	ValidHeaderMask	rw	0x00	Valid header received in Rx mask: setting this bit masks the corresponding IRQ in RegIrqFlags
	3	TxDoneMask	rw	0x00	FIFO Payload transmission complete interrupt mask: setting this bit masks the corresponding IRQ in RegIrqFlags
	2	CadDoneMask	rw	0x00	CAD complete interrupt mask: setting this bit masks the corresponding IRQ in RegIrqFlags
	1	FhssChangeChannelMask	rw	0x00	FHSS change channel interrupt mask: setting this bit masks the corresponding IRQ in RegIrqFlags
	0	CadDetectedMask	rw	0x00	Cad Detected Interrupt Mask: setting this bit masks the corresponding IRQ in RegIrqFlags

Name (Address)	Bits	Variable Name	Mode	Reset	LoRa™ Description
RegIrqFlags (0x12)	7	RxTimeout	rc	0x00	Timeout interrupt: a write operation clears IRQ
	6	RxDone	rc	0x00	Packet reception complete interrupt: a write operation clears IRQ
	5	PayloadCrcError	rc	0x00	Payload CRC error interrupt: a write operation clears IRQ
	4	ValidHeader	rc	0x00	Valid header received in Rx: a write operation clears IRQ
	3	TxDone	rc	0x00	FIFO Payload transmission complete interrupt: a write operation clears IRQ
	2	CadDone	rc	0x00	CAD complete: write to clear: a write operation clears IRQ
	1	FhssChangeChannel	rc	0x00	FHSS change channel interrupt: a write operation clears IRQ
	0	CadDetected	rc	0x00	Valid Lora signal detected during CAD operation: a write operation clears IRQ
RegRxNbBytes (0x13)	7-0	FifoRxBytesNb	r	n/a	Number of payload bytes of latest packet received
RegRxHeaderCntValueMsb (0x14)	7-0	ValidHeaderCntMsb(15:8)	r	n/a	Number of valid headers received since last transition into Rx mode, MSB(15:8). Header and packet counters are reseted in Sleep mode.
RegRxHeaderCntValueLsb (0x15)	7-0	ValidHeaderCntLsb(7:0)	r	n/a	Number of valid headers received since last transition into Rx mode, LSB(7:0). Header and packet counters are reseted in Sleep mode.
RegRxPacketCntValueMsb (0x16)	7-0	ValidPacketCntMsb(15:8)	rc	n/a	Number of valid packets received since last transition into Rx mode, MSB(15:8). Header and packet counters are reseted in Sleep mode.
RegRxPacketCntValueLsb (0x17)	7-0	ValidPacketCntLsb(7:0)	r	n/a	Number of valid packets received since last transition into Rx mode, LSB(7:0). Header and packet counters are reseted in Sleep mode.
RegModemStat (0x18)	7-5	RxCodingRate	r	n/a	Coding rate of last header received
	4	ModemStatus	r	'1'	Modem clear
	3		r	'0'	Header info valid
	2		r	'0'	RX on-going
	1		r	'0'	Signal synchronized
	0		r	'0'	Signal detected
RegPktSnrValue (0x19)	7-0	PacketSnr	r	n/a	Estimation of SNR on last packet received.In two's compliment format mutiplied by 4. $SNR[dB] = \frac{Packet\ Snr[twos\ compliment]}{4}$

Name (Address)	Bits	Variable Name	Mode	Reset	LoRa™ Description
RegPktRssiValue (0x1A)	7-0	PacketRssi	r	n/a	RSSI of the latest packet received (dBm) $RSSI[dBm] = -137 + PacketRssi$
RegRssiValue (0x1B)	7-0	Rssi	r	n/a	Current RSSI value (dBm) $RSSI[dBm] = -137 + Rssi$
RegHopChannel (0x1C)	7	PllTimeout	r	n/a	PLL failed to lock while attempting a TX/RX/CAD operation 1 → PLL did not lock 0 → PLL did lock
	6	RxPayloadCrcOn	r	n/a	CRC information extracted from the received packet header 0 → Header indicates CRC off 1 → Header indicates CRC on
	5-0	FhssPresentChannel	r	n/a	Current value of frequency hopping channel in use.
RegModemConfig 1 (0x1D)	7-4	Bw	rw	0x07	Signal bandwidth: 0000 → 7.8 kHz 0001 → 10.4 kHz 0010 → 15.6 kHz 0011 → 20.8kHz 0100 → 31.25 kHz 0101 → 41.7 kHz 0110 → 62.5 kHz 0111 → 125 kHz 1000 → 250 kHz 1001 → 500 kHz other values → reserved In the lower band (169MHz), signal bandwidths 8&9 are not supported)
	3-1	CodingRate	rw	'001'	Error coding rate 001 → 4/5 010 → 4/6 011 → 4/7 100 → 4/8 All other values → reserved In implicit header mode should be set on receiver to determine expected coding rate. See Section 4.1.1.3
	0	ImplicitHeaderModeOn	rw	0x0	0 → Explicit Header mode 1 → Implicit Header mode

Name (Address)	Bits	Variable Name	Mode	Reset	LoRa™ Description
RegModemConfig 2 (0x1E)	7-4	SpreadingFactor	rw	0x07	SF rate (expressed as a base-2 logarithm) 6 → 64 chips / symbol 7 → 128 chips / symbol 8 → 256 chips / symbol 9 → 512 chips / symbol 10 → 1024 chips / symbol 11 → 2048 chips / symbol 12 → 4096 chips / symbol other values reserved.
	3	TxContinuousMode	rw	0	0 → normal mode, a single packet is sent 1 → continuous mode, send multiple packets across the FIFO (used for spectral analysis)
	2	RxPayloadCrcOn	rw	0x00	CRC information extracted from the received packet header 0 → Header indicates CRC off 1 → Header indicates CRC on
	1-0	SymbTimeout(9:8)	rw	0x00	RX Time-Out MSB
RegSymbTimeoutLsb (0x1F)	7-0	SymbTimeout(7:0)	rw	0x64	RX Time-Out LSB RX operation time-out value expressed as number of symbols: $TimeOut = SymbTimeout \cdot Ts$
RegPreambleMsb (0x20)	7-0	PreambleLength(15:8)	rw	0x0	Preamble length MSB, = PreambleLength + 4.25 Symbols See Section XX for more details.
RegPreambleLsb (0x21)	7-0	PreambleLength(7:0)	rw	0x8	Preamble Length LSB
RegPayloadLength (0x22)	7-0	PayloadLength(7:0)	rw	0x1	Payload length in bytes. The register needs to be set in implicit header mode for the expected packet length. A 0 value is not permitted
RegMaxPayloadLength (0x23)	7-0	PayloadMaxLength(7:0)	rw	0xff	Maximum payload length; if header payload length exceeds value a header CRC error is generated. Allows filtering of packet with a bad size.
RegHopPeriod (0x24)	7-0	FreqHoppingPeriod(7:0)	rw	0x0	Symbol periods between frequency hops. (0 = disabled). 1st hop always happen after the 1st header symbol
RegFifoRxByteAddr (0x25)	7-0	FifoRxByteAddrPtr	r	n/a	Current value of RX databuffer pointer (address of last byte written by Lora receiver)
RegModemConfig 3 (0x26)	7-4	Unused	r	0x00	
	3	MobileNode	rw	0x00	0 → Use for static node 1 → Use for mobile node
	2	AgcAutoOn	rw	0x00	0 → LNA gain set by register LnaGain 1 → LNA gain set by the internal AGC loop
	1-0	Reserved	rw	0x00	Reserved
(0x27) - (0x3F)	-	Reserved	r	n/a	Reserved