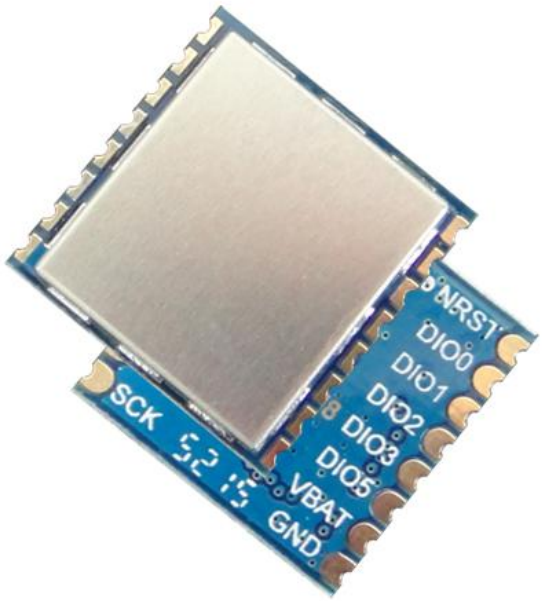
APC1278 射频前端技术说明书

VER2.1.2



深圳市安美通科技有限公司

2015.6.6

一．应用场合

应用：

- 无线传感器
- 远距离数据通讯
- 工业过程自动化控制
- 自动化数据采集
- 野外远程数据遥控、遥测
- 智能电力，智能交通
- 无人机，机器人控制
- 矿山石油设备控制通讯
- 车辆监控管理
- 环境、节能、温度监测

二. 尺寸及引脚定义

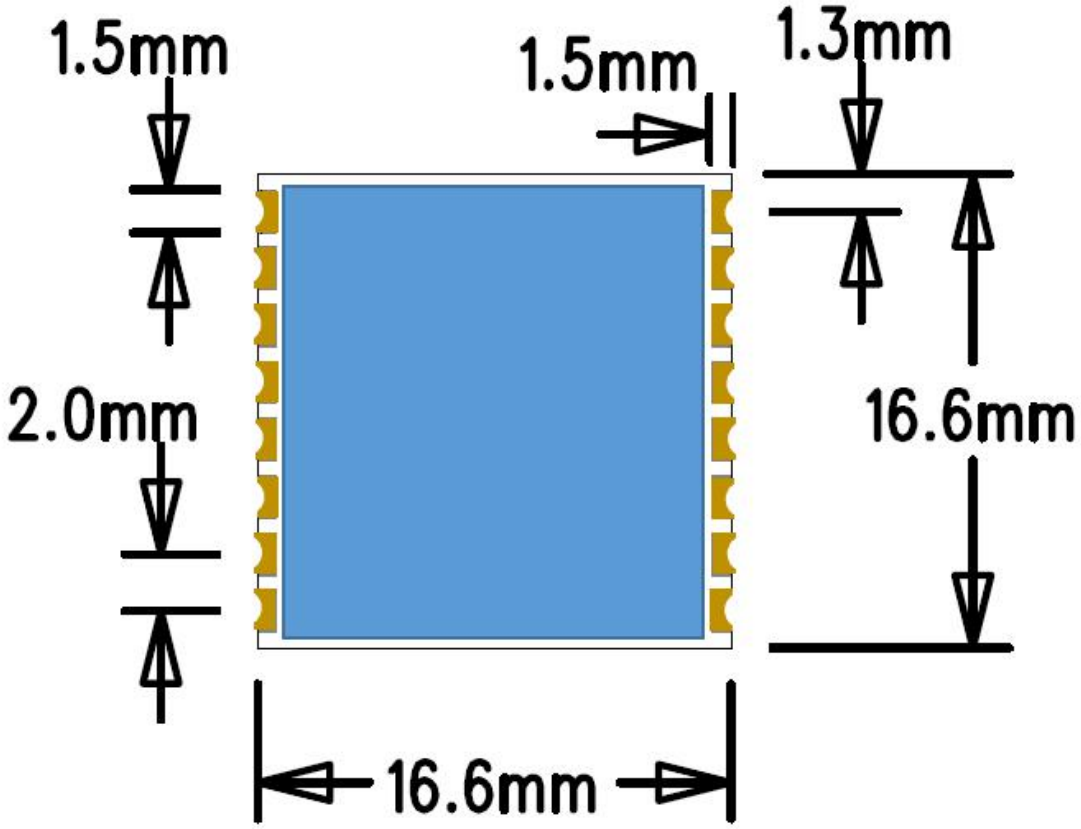


图 1. 外框尺寸

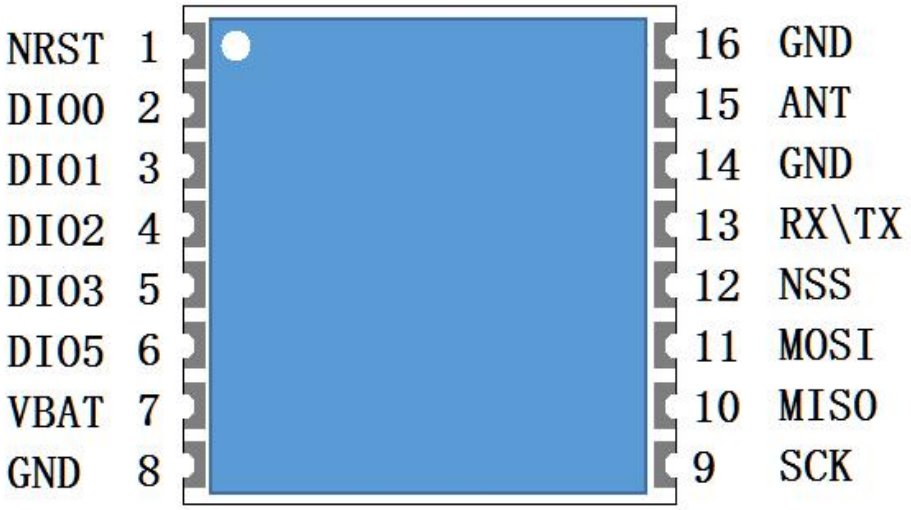


图 2. 管脚标识(仰视图)

2.1 引脚定义：

序号	引脚名字	引脚功能	引脚描述
1	NRST	RESET	Reset trigger input
2	DIO0	I/O 口	Software configured
3	DIO1	I/O 口	Software configured
4	DIO2	I/O 口	Software configured
5	DIO3	I/O 口	Software configured
6	DIO5	I/O 口	Software configured
7	VBAT	电源	1.8V-3.6V
8	GND	电源地	电源地(0V)
9	SCK	SPI 口	SPI clock input
10	MISO	SPI 口	SPI data output
11	MOSI	SPI 口	SPI data input
12	NSS	SPI 口	SPI chip select input
13	RX\TX	RTX Control	切换射频开关收发模式; 发送:H; 休眠: L; 接收:L
14	GND	电源地	电源地(0V)
15	ANT	天线馈线口	射频信号输出口
16	GND	电源地	电源地(0V)

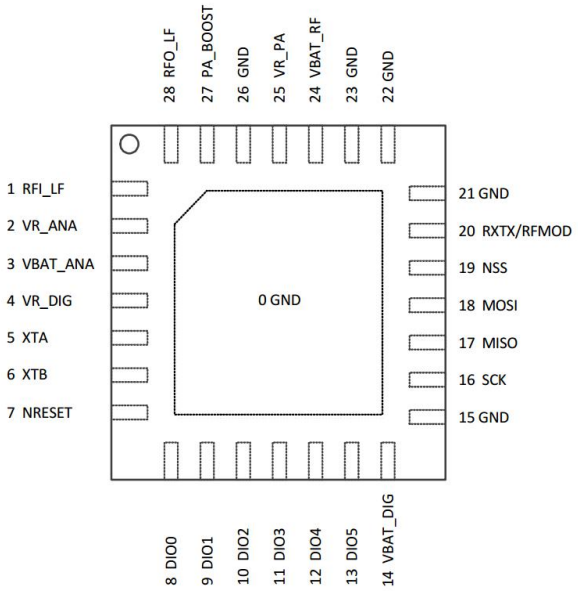
表 1. 引脚定义

三. 技术指标

APC1278 技术指标（测试条件：2.1-3.6V,25℃±5℃）	
工作频率	410-455MHz，可定制470-510MHz
调制方式	LORA
发射功率	Max +20dBm @High power mode
接收灵敏度	-148dBm@SF=12, 7.8 kHz bandwidth
工作湿度	10%～90% (无冷凝)
工作温度	-40℃ - 85℃
电源	2.1 – 3.6V
发射电流(典型值)	85mA@50mW
持续接收电流(典型值)	10.5mA@5Kbps
频率误差	≤ ±20ppm

四.

参考设计原理图



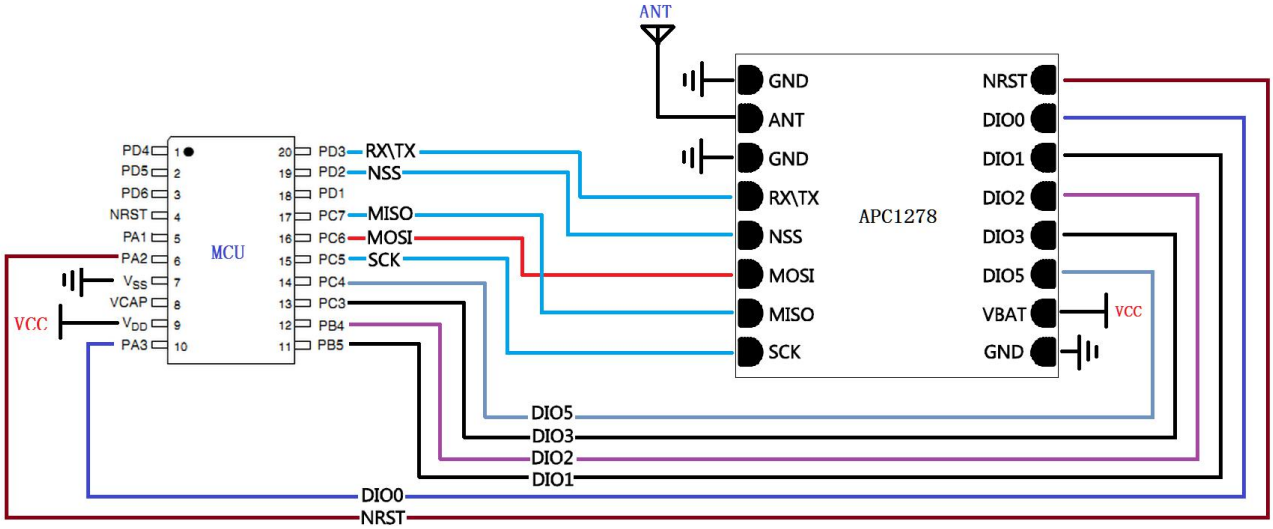


图 3. 参考原理图

五. 应用说明

5.1.1 APC1278复位时序

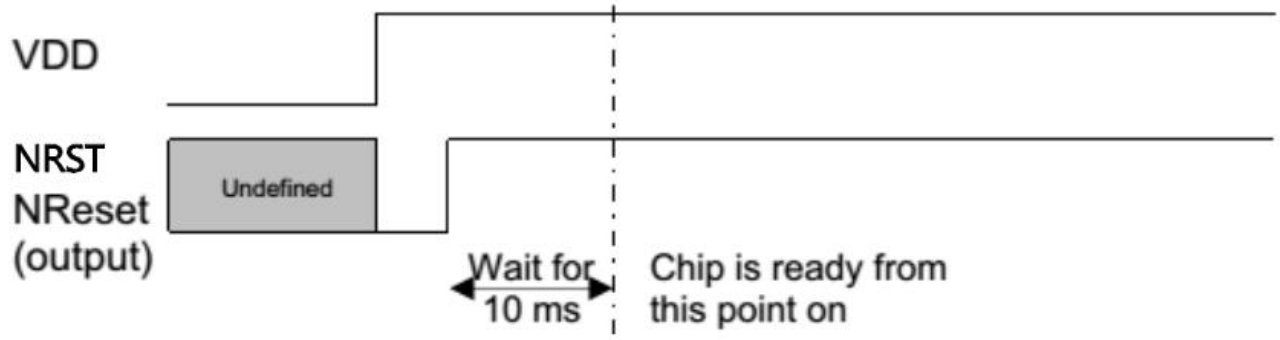


图 4. APC1278 上电复位时序图

5.1.2 APC1278 手动复位时序

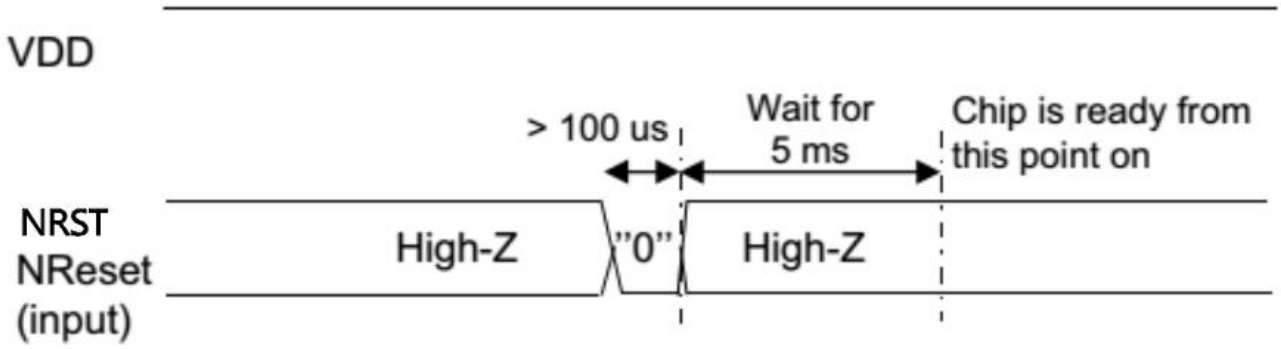


图 5. APC1278 手动复位时序图

5.1.2.1 APC1278复位步骤：

NRST = Low 低电平大于100uS，然后NRST = High 高电平大于5mS 即可完成手动复位。

5.2 SPI接口说明

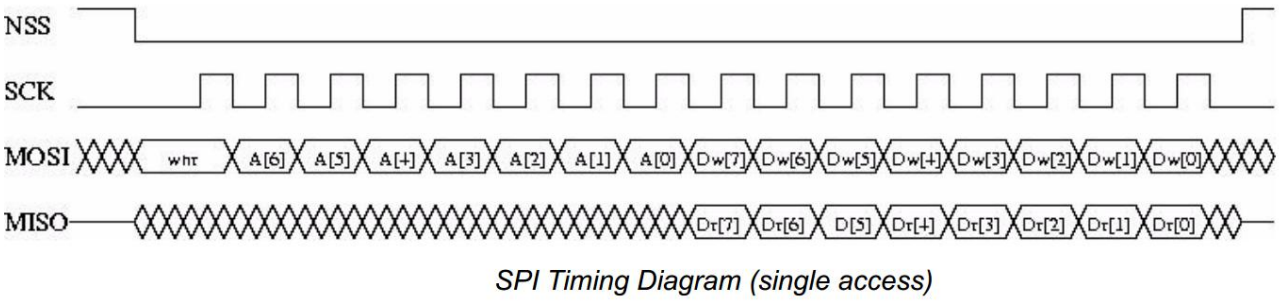


图 6. APC1278 SPI 接口时序图

CPOL = 0, CPHA = 0; MSB；支持单字节、多字节连续读写。

MOSI 在SCK下降沿或者低电平时输出数据，APC1278在SCK上升沿时采样数据。

MISO 在SCK下降沿或者低电平时输出数据， MCU在SCK上升沿时采样数据。

SPI第一个数据字节为地址域，bit7 为读写控制位，“1”表示写，“0”表示读；bit（6-0）对应当前操作的寄存器地址。在连续读写操作模式时，寄存器会自动加“1”，直到NSS脚被拉高； 特别注意：FIFO操作时，寄存器地址不会自动增加，而是FIFO内的缓存地址。 寄存器的详细说明请参考芯片数据手册的“寄存器说明章节”。

5.2.1 SPI接口C程序

```
// SPI 接口底层子程序
uint8_t SpiInOut( uint8_t outData )
{
    /* Send SPIy data */
    SPI_I2S_SendData( SPI_INTERFACE, outData );
    while( SPI_I2S_GetFlagStatus( SPI_INTERFACE, SPI_I2S_FLAG_RXNE ) == RESET );
    return SPI_I2S_ReceiveData( SPI_INTERFACE );
}
// 写寄存器子程序  单字节操作
void SX1276Write( uint8_t addr, uint8_t data )
{
    SX1276WriteBuffer( addr, &data, 1 );
}
// 读寄存器子程序  单字节操作
void SX1276Read( uint8_t addr, uint8_t *data )
{
    SX1276ReadBuffer( addr, data, 1 );
}

// 写寄存器子程序  连续多字节操作
void SX1276WriteBuffer( uint8_t addr, uint8_t *buffer, uint8_t size )
{
    uint8_t i;

    //NSS = 0;
    GPIO_WriteBit( NSS_IOPORT, NSS_PIN, Bit_RESET );

    SpiInOut( addr | 0x80 );
    for( i = 0; i < size; i++ )
    {
        SpiInOut( buffer[i] );
    }

    //NSS = 1;
    GPIO_WriteBit( NSS_IOPORT, NSS_PIN, Bit_SET );
}
```

```
// 读寄存器子程序   连续多字节操作
void SX1276ReadBuffer( uint8_t addr, uint8_t *buffer, uint8_t size )
{
    uint8_t i;

    //NSS = 0;
    GPIO_WriteBit( NSS_IOPORT, NSS_PIN, Bit_RESET );

    SpiInOut( addr & 0x7F );

    for( i = 0; i < size; i++ )
    {
        buffer[i] = SpiInOut( 0 );
    }

    //NSS = 1;
    GPIO_WriteBit( NSS_IOPORT, NSS_PIN, Bit_SET );
}

// 写FIFO寄存器子程序   连续多字节操作
void SX1276WriteFifo( uint8_t *buffer, uint8_t size )
{
    SX1276WriteBuffer( 0, buffer, size );
}

// 读FIFO寄存器子程序   连续多字节操作
void SX1276ReadFifo( uint8_t *buffer, uint8_t size )
{
    SX1276ReadBuffer( 0, buffer, size );
}

//   SX127x 复位子程序
void SX1276SetReset( uint8_t state )
{
    GPIO_InitTypeDef GPIO_InitStructure;

    if( state == RADIO_RESET_ON )
    {
        // Set RESET pin to 0
        GPIO_WriteBit( RESET_IOPORT, RESET_PIN, Bit_RESET );
        // Configure RESET as output
        GPIO_InitStructure.GPIO_Mode  = GPIO_Mode_Out_PP;
        GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
```

```
            GPIO_InitStructure.GPIO_Pin = RESET_PIN;
            GPIO_Init( RESET_IOPORT, &GPIO_InitStructure );
        }
        else
        {
            // Configure RESET as input
            GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
            GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
            GPIO_InitStructure.GPIO_Pin =    RESET_PIN;
            GPIO_Init( RESET_IOPORT, &GPIO_InitStructure );

        }
}

// SX127x  复位程序
void SX1276LoRaReset( void )
{
        SX1276SetReset( RADIO_RESET_ON );

        // Wait 1ms
        uint32_t startTick = GET_TICK_COUNT( );
        while( ( GET_TICK_COUNT( ) - startTick ) < TICK_RATE_MS( 1 ) );

        SX1276SetReset( RADIO_RESET_OFF );

        // Wait 6ms
        startTick = GET_TICK_COUNT( );
        while( ( GET_TICK_COUNT( ) - startTick ) < TICK_RATE_MS( 6 ) );
}

// TX天线开关切换，  进入发送模式前调用
void Rfsw_tx(void)
{


SX1276WriteReg( REG_LR_DIOMAPPING1,(RFLR_DIOMAPPING1_DIO0_01 |
RFLR_DIOMAPPING1_DIO1_00 | RFLR_DIOMAPPING1_DIO2_00 | RFLR_DIOMAPPING1_DIO3_00));
        SX1276WriteReg( REG_LR_DIOMAPPING2,(RFLR_DIOMAPPING2_DIO4_00 |
RFLR_DIOMAPPING2_DIO5_00 ));



}
```

注： 为了兼容后续产品，发送模式DIO4=00，参见下图Table 18

```
// RX天线开关切换, 进入接收/休眠模式前调用
void Rfsw_rx(void)
{
    SX1276WriteReg( REG_LR_DIOMAPPING1,(RFLR_DIOMAPPING1_DIO0_00 |
RFLR_DIOMAPPING1_DIO1_00 | RFLR_DIOMAPPING1_DIO2_00 | RFLR_DIOMAPPING1_DIO3_00));
    SX1276WriteReg( REG_LR_DIOMAPPING2,(RFLR_DIOMAPPING2_DIO4_01 |
RFLR_DIOMAPPING2_DIO5_00 ));
}
```

5.3 常用寄存器说明

5.3.1 频段说明：

Frequency Bands

Name	Frequency Limits	Products
Band 1 (HF)	862 (*779)-1020 (*960) MHz	SX1276/77/79
Band 2 (LF)	410-525 (*480) MHz	SX1276/77/78/79
Band 3 (LF)	137-175 (*160)MHz	SX1276/77/78/79

** For SX1279*

5.3.2 频率的计算公式：

$$F_{STEP} = \frac{F_{XOSC}}{2^{19}}$$

$$F_{RF} = F_{STEP} \times Frf(23,0)$$

F_{RF} ：工作频率

$F_{rf(23,0)}$ ：寄存器0x06,0x07,0x08

F_{setp} = 32Mhz/2^19 = 61Hz

举例说明： 工作频率=434Mhz

根据公式 Frf = Fstep x Frf(23,0)

= > Frf(23,0) = Frf / Fstep

= Frd/(32Mhz/2^19)

Frf(23,0) = 434,000,000Hz / (32,000,000Hz/2^19)

= 7110656

= 0x6C8000

5.3.3 RSSI的计算公式：

当前信道RSSI：

RSSI[dBm] = -157 + Rssi (using HF output port)

or

RSSI[dBm] = -164 + Rssi (using LF output port)

数据包RSSI：

RSSI[dBm] = -157 + Rssi (using HF output port, SNR >= 0)

or

RSSI[dBm] = -164 + Rssi (using LF output port, SNR >= 0)

5.3.3 工作模式：

Lora模式：

RegOpMode.Mode

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)

FSK/OOK模式：

RegOpMode.Mode

000 → Sleep mode

001 → Stdby mode

010 → FS mode TX (FSTx)

011 → Transmitter mode (Tx)

100 → FS mode RX (FSRx)

101 → Receiver mode (Rx)

110 → reserved

111 → reserved

详细寄存器配置请参见 SX1276 Starter Kit 软件：

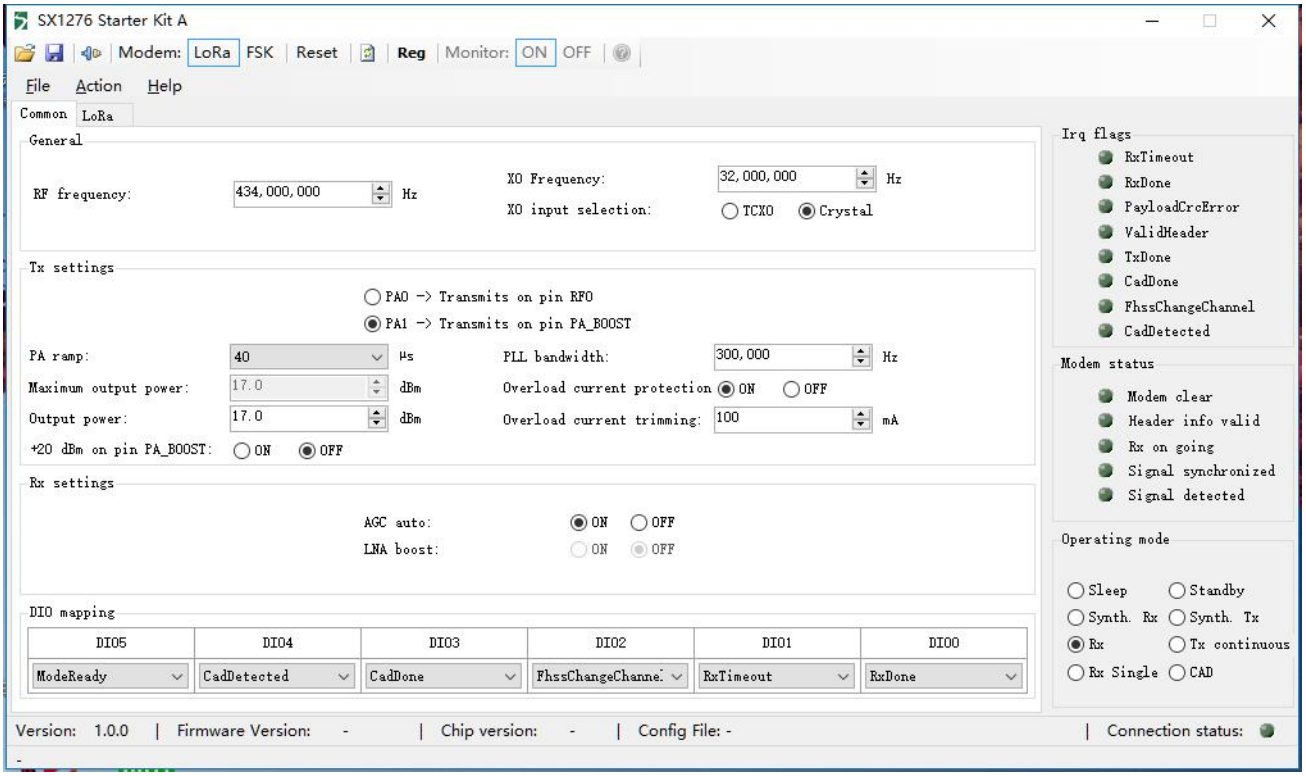


Table 18 DIO Mapping LoRa™ Mode

Operating Mode	DIOx Mapping	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
ALL	00	ModeReady	CadDetected	CadDone	FhssChangeChannel	RxTimeout	RxDone
	01	ClkOut	PllLock	ValidHeader	FhssChangeChannel	FhssChangeChannel	TxDone
	10	ClkOut	PllLock	PayloadCrcError	FhssChangeChannel	CadDetected	CadDone
	11	-	-	-	-	-	-

深圳市安美通科技有限公司

深圳市福田区侨城东路与侨香路交界君子广场904-905

Tel: 86-0755-82776762

Fax: 86-0755-82787392

Email:appcon@126.com

http://www.appcon.com.cn