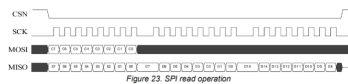


- 究极debug中的bug。记得每次改代码后要下电，不然代码不生效，烧进去了。需要reset,IC而reset单片机没用
- <https://www.arduino.cn/thread-86275-4-1.html>神贴
- TX_ADDR：目标地址，用以填充发送，发送时候被读取到数据包的地址段。
- RX_ADDR：用以检测接收，读取的时候检测空中的信号中地址段是否和本机RX_ADDR一致来确定是否继续读取接下来的信号。
- CE在PTX时只需要维持最少10us。而PRX时维持高电平则一致启动PRX状态(包括自动切换应答发送)
- W_REGISTER、ACTIVATE**只有在powerdown 和standby状态下才能使用**
- **Status值在CSN从高到低时(MOSI发送第一个字节时(也就是发送命令字时))同时发送到MISO，在IQR由高到低时更新STATUS寄存器，写对应bit 1复位。**
 - 有三种方式，可以使用NOP指令，STATUS指令，或在发送其他指令的同时。
- SPI传输完1次指令和数据后就要拉高表示次传输完毕。下一次传输拉低。



- PTX使用自动重发和接收应答时，用数据通道0接收应答信号。故TX_ADDR与RX_ADDR_P0 (PTX) 配置成一样。
- MAX_RT IRQ 不清除不能发送信息。
- **2Mbps情况下，信道间隔必须大于2MHz，否则信道的频带重叠。**
- **自动重发延ARD时需要考虑自动应答数据长度在空中时间。具体看下面介绍**
- 之所以规定有效数据长度是因为 —— 时分复用。保证双向实时性，以解决半双工问题。
- 使能自动应答后只有收到ACK才有TX_DS IRQ。
- 想用带回复的应答，必须开启带回复的应答，开启动态数据长度，使能对应通道，使能对应通道自动应答，使能对应通道动态数据长度。
- 静态数据长度时，接收方数据长度由寄存器配置的决定，可以通过指令读取RXFIO信息的长度。发送方的长度由SPI发送数据时候的长度来决定，且两者必须一致。
- 上电到power down 10ms，power down -> standby1 (设置PWR_UP位) 1.5ms。standby1->TXorRX模式 130us。
- 1次传输中PRX的RX_IRQ 比PTX的TX_IRQ早。

传输时序讲解

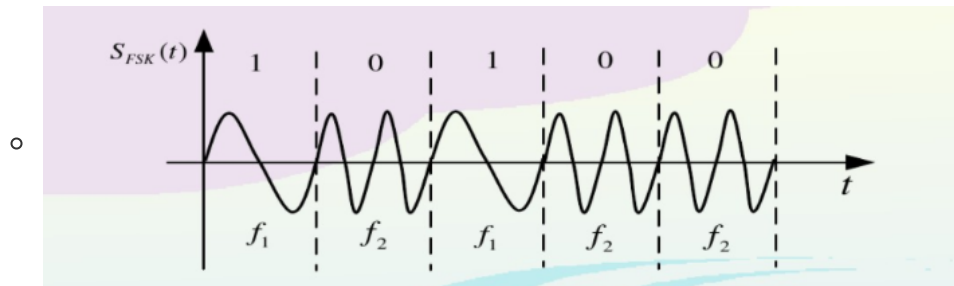
看框图是看文档的核心，否则你都不知道他在讲啥。各个部分又是啥。

- XC1和XC2是外部晶振，为ic提供精确的时基(而不是用内部的rc，因为那个东西不准，在射频这种高速场合不能用)。接在收发处，解调用的还是啥，反转就是提供一个时钟把？？
- 电源管理引脚好说

射频前端部分

- 左边ANT1,ANT2 是天线引脚，射频信号的收发处，在这里进行电平变化。
- PA接在发送处，PA是功率放大器，所以是向外的。
- LNA是低噪声放大器，接在接受处，用以在噪声中接收信息。
- TXfilter和Rxfilter就是滤波器了。

- GFSK是高斯频移键控，调制器和解调器



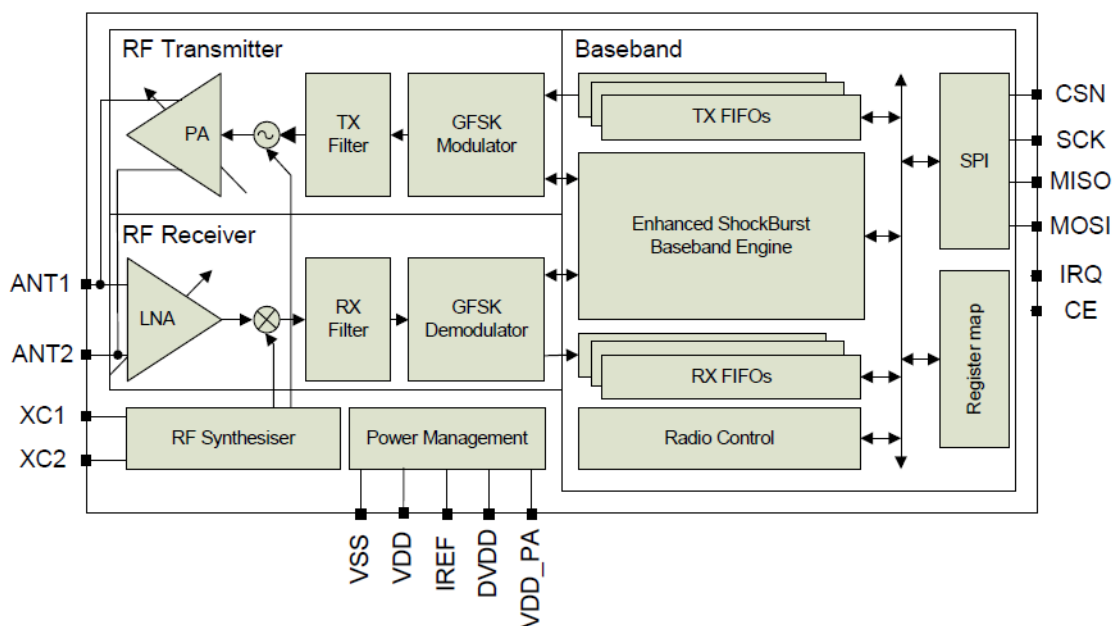
- 键控频移就是0的时候在对应频率(载波频率2.4GHz~2.525GHz,记得避开WIFI信号频段)上高一点，1低一点。通过检测频率来检波。而每个bit维持一定时间的0高频率和1低频率才被检测为合格bit。这个维持时间就是控制速率(所说的1Mbps和2Mbps)的倒数。
- 在信号调制后，会占用载波附近一个频率带叫频带的信号。而两个不同信道的频带不能重叠。对于1Mbps来说可调的间距1MHz够用了，但是对于2Mbps来说，频带间隔必须设置相差2MHz。

基带

- TX FIFOs用以储存要发送的数据。这也是软件层抽象的最大层度，认为要发送的东西都存在这里了
- Rx FIFOs用以存储接收到的数据。也是抽象最高层
- 增强型ShockBurst控制块
- 射频控制块
- 内部硬件SPI
- 寄存器映射组

##

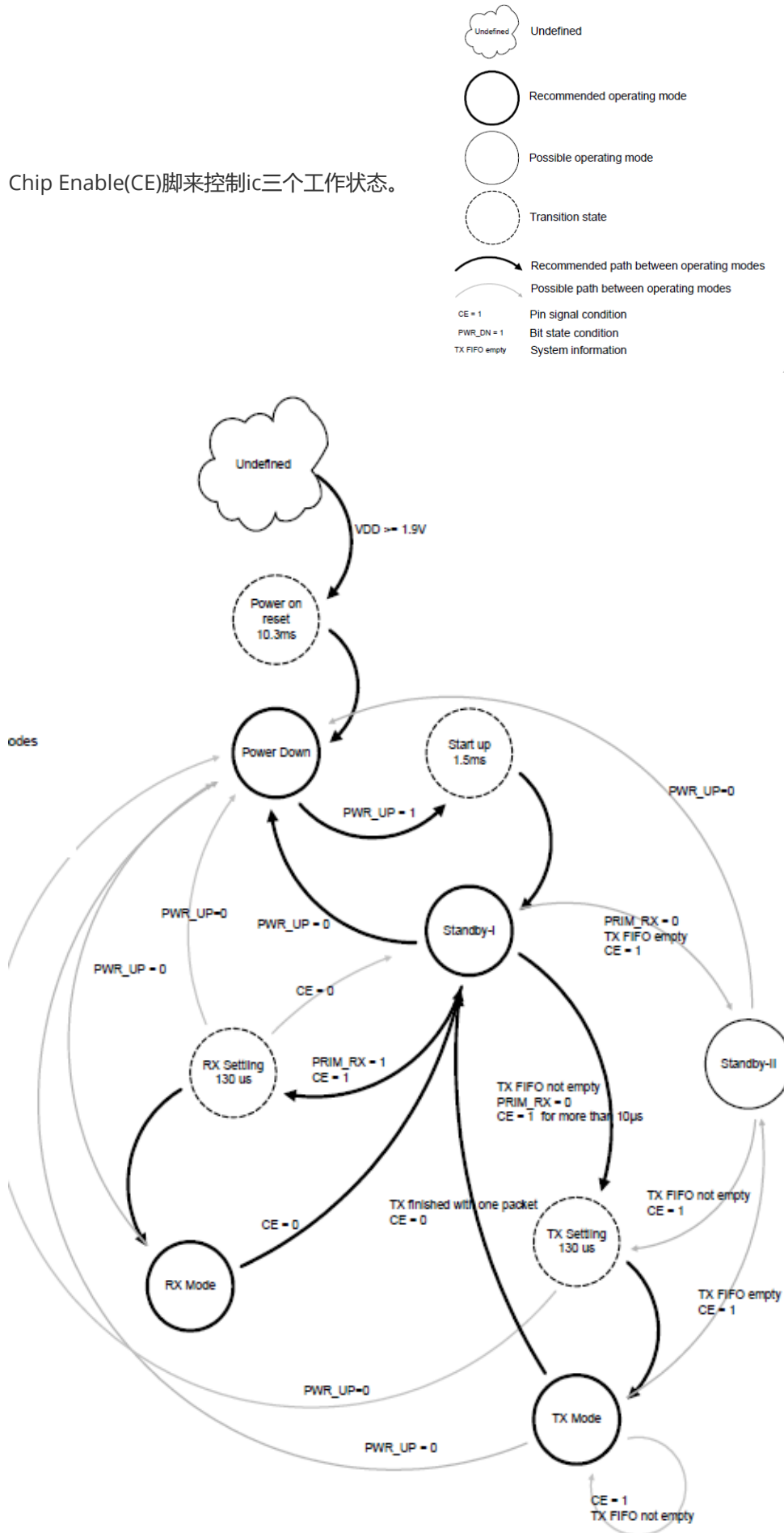
电流消耗：下电待机状态只要0.9uA，非发送接收时候的上电状态大概300uA。接收发送的时候大概10mA。



状态流程图：

当VDD超过1.9V时候，nrf24l01进入上电复位状态直到进入下电模式，模式下，MCU可以通过SPI和

Chip Enable(CE)脚来控制ic三个工作状态。



- power down: 设置PWR_UP bit in the CONFIG register low可以待机。

- 标准状态1: pwr_up 设1, **经过很短的启动时间(150us)**, 从power down 唤醒。当从Rx或Tx模式下(令CE=0后)返回后的状态
- 标准状态2: 不是很清楚, 好像差不多。就是多了什么始终设置
- RX Mode: pwr_up 1, PRIM_RX =1, 且CE脚=1(经过130us后设置完毕)。解调器不断从射频通道**解调**电平信息。然后传输到基带的协议引擎。基带协议引擎不断**判断**合法数据包(地址匹配和通过CRC校验)。合法数据包被放入RX FIFOs中的空位, 如果满了就被丢弃。如果不被配置成power down和标准1(CE=0), 则一直保持RX Mode。如果内部使能了增强ShockBurst, 则会切换成别的状态来实现协议。
- Tx Mode: PWR_UP =1, PRIM_RX =0, FIFO非空, **且CE 高电平不少于10us**, 进入TXMODE模式(经过130us后设置完毕)切换状态完成切换至TXmode。发送完当前数据包后, 如果CE还=1且TX FIFOs还有数据包, 则继续发送, 直到发完后进入标准状态2。如果CE=0, 则回到标准状态1。不能在标准状态2超过4ms。
 - 在此工作模式下, 如果CE保持高电平, 则清空TX FIFO, 并执行所有必要的ACK和可能的重传。只要重新填充TX FIFO, 传输就会继续。如果CE仍然为高电平时, 如果TX FIFO为空, 则nRF24L01进入待机II模式。在此模式下, 在将数据包上载 (UL) 到TX FIFO之后, 只要将CSN设置为高电平, 就开始传输数据包。
 - 如果只是维持10us那么当发送完1包数据后, TXFIFO还有为发送完毕的。但是CE=0了, 所以切换到standby1模式
 - 如果CE一维持高, 则回把TXFIFO的全发完, 然后进入standby2。(比1功耗高1个数量级), 此时可以调用W_REGISER命令写寄存器, 然后继续发送数据到TXFIFO, 如果TXFIFO在standby2下非空, 则又转换到TXMode发送数据。

当VDD下电, 寄存器数据就会丢失。

空中速率: 是解调器使用解调信令的速率(载波肯定是2.4GHz啊..)。空中数据速率可以是1 Mbps或2 Mbps。与2 Mbps相比, **1 Mbps的数据速率提供了更好的3dB接收器灵敏度**。高空中数据速率意味着更低的平均电流消耗和更低的空中碰撞概率。

RF_DR bit in the RF_SETUP register.设置

RF channel frequency:

射频通道的频率: 从2.4G到2.525GHz可以设置共126个通道。 $F_0 = 2400 + RF_CH$ [MHz]

但是: 2MHz频带下的信道宽度更大。为了在2MHz时候使用, 各个频率间隔要大点。而1MHz就按配置来就好。

因为频带越宽越容易识别出响应的bit, 而高速率响应提高频带宽度可以保证可靠性。故2MHz频带更宽。

PA控制:

功放的设置

SPI RF-SETUP (RF_PWR)	RF output power	DC current consumption
11	0dBm	11.3mA
10	-6dBm	9.0mA
01	-12dBm	7.5mA
00	-18dBm	7.0mA

增强型sockburst 是通过用户软件设置发送，(然后硬件自动切换成接收尝试得到应答，并尝试重新发送)。来减小MCU的负担。具有自动应答，重新发送，自动打包，自动解包，自动验证数据的功能。

可变的数据长度。

6 data pipe MultiCeiver™ for 1:6 star networks-> 可实现1对6传输。

发送时，Enhanced ShockBurst™ uses ShockBurst™来打包数据1位1位发送到发送端。接收的时候，ShockBurst™不断搜索解调信号的地址是否有效。发现有效地址的信号后，判断CRC是否正确。如果正确，就是有效信号。然后放入RXfifo。

Enhanced ShockBurst™可以实现可靠的双向数据链。一个增强Enhanced ShockBurst的发送是数据包在收发机之间的交换。当有1个是主接收机，1个是主发送机时。增强数据包的发送是从PTX的发送开始，从PRX**确认接收**时完成

自动数据包发送流程如下：

- 用户通过PTX想PRX发送数据包，增强型会自动将PTX设置成接收模式来 等待ACK应答数据包。
- 如果数据包被PRx接收，则增强型在返回接收模式之前自动组装确认数据包(ACK分组)并向PTx**发送确认数据包**
- 如果在一定时间内PTX没收到应答数据，则会自动重发，并切换模式等待。

在PRX的应答数据包中，用户也可以添加数据到数据包payload中，来实现数据双向链路。增强型有很多可以配置的地方。**可以配置做大重发次数和两次发送间隔延迟。**

Preamble 1 byte	Address 3-5 byte	Packet Control Field 9 bit	Payload 0 - 32 byte	CRC 1-2 byte
-----------------	------------------	----------------------------	---------------------	--------------

增强型的数据帧格式：1个1byte的起使。起使只有2种，根据地址的起使是0还是1来决定。

Payload length 6bit	PID 2bit	NO_ACK 1bit
---------------------	----------	-------------

控制位分成3部分。PID是Packet identification。用来判断数据包时新的还是重发的数据包。

应答位：NO_ACK 可以选择是否自动应答。在自动应答的时候可以用，设置了flag为1，则接收者被告知数据包不用应答。W_TX_PAYLOAD_NOACK用这个指令来配置

CRC：由address, Packet Control Field, and Payload计算。如果计算失败则被抛弃

地址：如果地址只有1次比特翻转，则可能收到很多噪声。比如地址为00001111111111。尽量用类似0001100110101这样多次翻转的。

The address is fetched from the TX_ADDR register. The address field can be configured to be 3, 4 or 5 bytes long with the AW register.

地址是从寄存器TX_ADDR读取并自动组装的。地址长度可以在AW寄存器配置

Preamble 1 byte	Address 3-5 byte	Packet Control Field 9 bit	Payload 0 - 32 byte	CRC 1-2 byte
-----------------	------------------	----------------------------	---------------------	--------------

Figure 4. An Enhanced ShockBurst™ packet with payload (0-32 bytes)

7.3.1 Preamble

The preamble is a bit sequence used to detect 0 and 1 levels in the receiver. The preamble is one byte long and is either 01010101 or 10101010. If the first bit in the address is 1 the preamble is automatically set to 10101010 and if the first bit is 0 the preamble is automatically set to 01010101. This is done to ensure there are enough transitions in the preamble to stabilize the receiver.

7.3.2 Address

This is the address for the receiver. An address ensures that the correct packet are detected by the receiver. The address field can be configured to be 3, 4 or, 5 bytes long with the AW register.

Note: Addresses where the level shifts only one time (that is, 000FFFFFFF) can often be detected in noise and can give a false detection, which may give a raised Packet-Error-Rate. Addresses as a continuation of the preamble (hi-low toggling) raises the Packet-Error-Rate.

发出的信息从 The payload is fetched from the TX FIFO。即数据包中的信息是从TXFIO取出的。

自动数据包处理:

增强型ShockBurst™使用ShockBurst™进行自动数据包处理

这些功能是静态和动态有效载荷长度，自动数据包组装，自动数据包验证和自动数据包拆卸。

数据包长度可以设置成静态和动态。

静态：默认静态，此时接收和发送的长度是必须一样的。静态有效负载长度由接收器侧的RX_PW_Px寄存器设置。发送器端的有效负载长度由时钟输入TX_FIFO的字节数设置，并且必须等于接收器端RX_PW_Px寄存器中的值

动态：EN_DPL设定FEATURE寄存器的这位，接收者的数据长度由解码出来的长度确定而不是RX_PW_Px决定。MCU可以通过R_RX_PL_WID来读取接收长度。为了启用DPL，必须将FEATURE寄存器中的EN_DPL位置1。在RX模式下，必须设置DYNPD寄存器。发送到启用了DPL的PRX的PTX必须将DYNPD中的DPL P0位置1以是能自动应答接收通道的DRL模式。

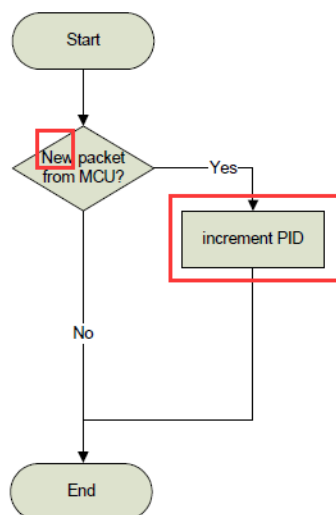
- d. If ACK packet payload is activated, ACK packets have dynamic payload lengths and the Dynamic Payload Length feature should be enabled for pipe 0 on the PTX and PRX. This is to ensure that they receive the ACK packets with payloads. If the payload in ACK is more than 15 byte in 2Mbps mode the ARD must be 500µs or more, and if the payload is more than 5byte in 1Mbps mode the ARD must be 500µs or more.

自动数据包组件将前同步码，地址，数据包控制字段，有效负载和CRC组装在一起，以在发送之前形成完整的数据包

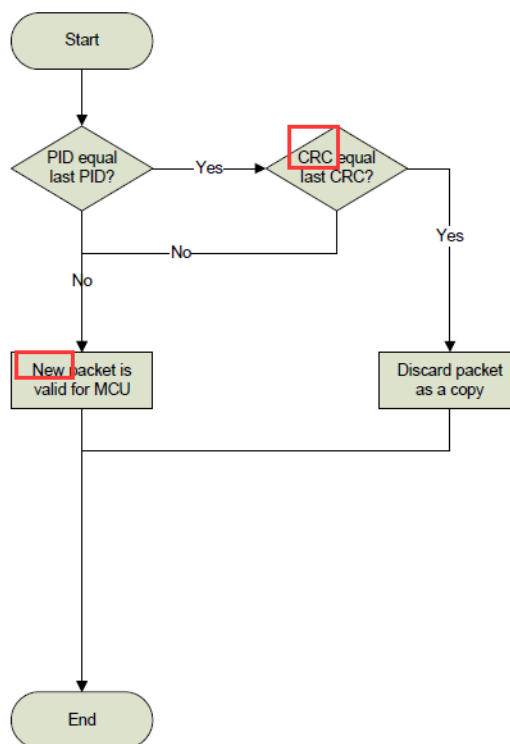
自动合法验证:

解调器不断从射频通道解调电平信息。然后传输到基带的协议引擎。基带协议引擎不断判断合法数据包(地址匹配和通过CRC校验，如果是静态长度，会通过寄存器中的PX_RW读取数据长度来获取数据，动态的从数据中解码得到数据长度，如果捕获数据后就进行CRC，CRC合法会检测PID位，如果这个PID和以前的不一样，认为是新的数据包不是重发的，如果相同就CRC，CRC也相同就认为是同一个数据，然后抛弃)。合法数据包被放入RX FIFOs中的空位，and assert the RX DR IRQ。

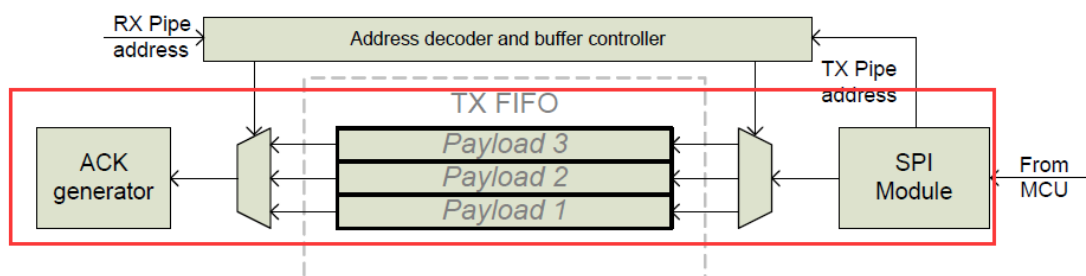
PTX side functionality



PRX side functionality



自动应答：当接收到并合法验证的消息后，自动发送ACK 包到PTX。可以减少MCU的操作和SPI的负担。可以通过设定 EN_AA register使能。如果收到的数据包中NO_ACK是高，自动回复无用。ACK包可以加入可选的数据内容从PRX->PTX，使用此功能需要使用动态数据长度，PRX侧得MCU需要通过 **W ACK PAYLOAD**指令来更新数据到TXFIFO，数据会挂号在TXFIFO，直到新数据包收到，NRF24L01可以同时TXFIFO中挂号3个数据。



如果链路消失了，且txfifo(RTX的) 还是满的，则TXFIFO会被阻塞。MCU 可以刷新TX FIFO (PRX) 使用 FLUSH_TX指令。

为了使用带负载的自动应答包。 **EN ACK PAY 位在 FEATURE寄存器中必须被值位**

而PTX的TX ADDR与RX ADDR P0 (PTX) 配置成一样以能够收到应答信号。

- d. If ACK packet payload is activated, ACK packets have dynamic payload lengths and the Dynamic Payload Length feature should be enabled for pipe 0 on the PTX and PRX. This is to ensure that they receive the ACK packets with payloads. If the payload in ACK is more than 15 byte in 2Mbps mode the ARD must be 500µs or more, and if the payload is more than 5byte in 1Mbps mode the ARD must be 500µs or more.

如果想用带数据的应答，应该在PRX切换到TX模式前到IRQ中断产生的这130us内处理完中断并通过SPI将数据通过指令发送到TXFIFO中。

In order to enable DPL the EN_DPL bit in the FEATURE register must be set. In RX mode the DYNPD register has to be set. **A PTX** that transmits to a PRX with DPL enabled must have the **DPL P0 bit in DYNPD** set.

带数据的自动应答，应答数据包发送完毕中断需要PTX再发送1次信息后PRX接收成功同时产生TXIRQ

自动重发：当PTX侧没收到PRX发来的应答时，可以通过设置ARC bits in the SETUP_RETR register来设置重发次数。然后PTX进入RX模式。

每当发送1个数据包后等待一个时间周期来获取应答。等待时间由以下决定：

ARD：发送完毕后到开始重发的间隔。可以设置SETUP_RETR寄存器来设置，步进值250us。

- Auto Retransmit Delay (ARD) or
- No address match within 250μs or
- After received packet (CRC correct or not) if address match within 250μs

当收到ACK数据包的时候TX_DS IRQ会触发。然后如果CE=0，且没有数据要发送了，则切换到状态StandBy I。

如果没收到ACK，则经过一个**由ARD决定的延时**后重新发送数据。这个过程一直发生，直到接收到信息或者超出最大重发次数了。每次丢包，ARC_CNT and PLOS_CNT in the OBSERVE_TX register都会增加。前者增加当本个数据包的重发送次数。后者记录自从上次改变频道后的丢失次数，每到达一次MAX_RT增加1，增加到最大值15进行溢出保护不变。

如果ARC_CNT设置成0，则每次传输失败都不会重发，直接产生MAX_RT IRQ

ARC CNT可以通过发送新的传输复位。PLOS CNT 可以通过 (dummy write also) 写RF_CH寄存器复位。可以用OBSERVE_TX寄存器数据对通道质量进行评估。

ARD时间在带数据的ACK是有限制的，要大于启动和传播之和。

For 1Mbps data rate and 5 byte address; 5 byte is maximum ACK packet payload length for ARD=250μs (reset value).

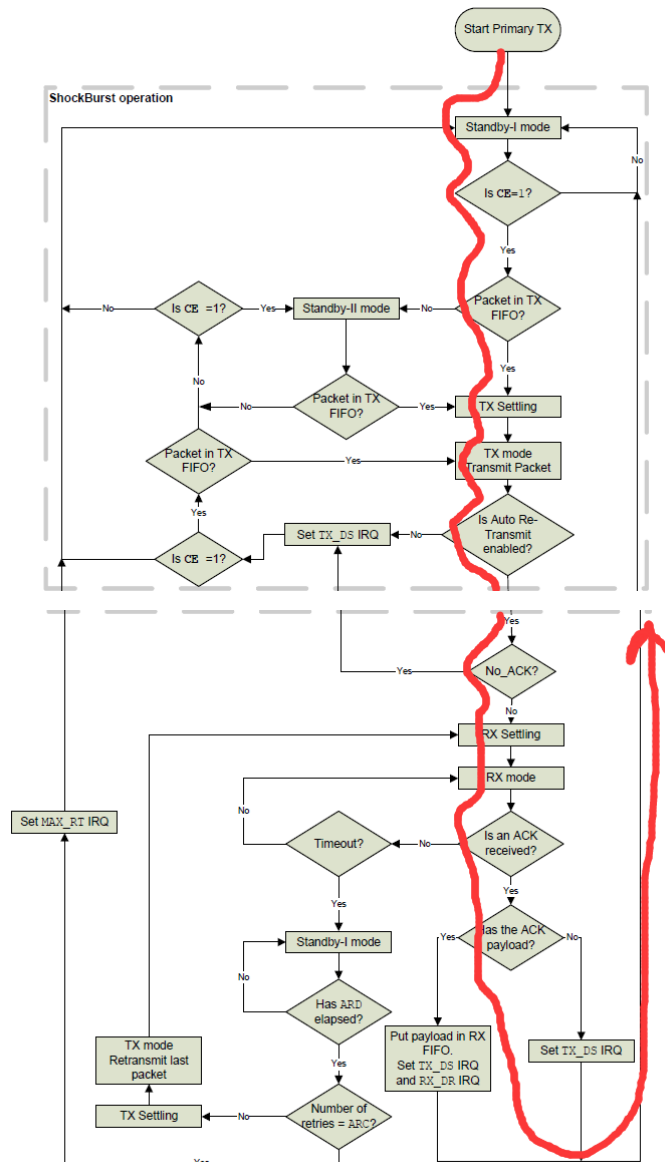
For 2Mbps data rate and 5 byte address; 15 byte is maximum ACK packet payload length for ARD=250μs (reset value).

ARD=500μs will be long enough for any payload length.

作为自动重发的替代方法，可以手动将nRF24L01设置为多次重发数据包。这是通过REUSE_TX_PL命令完成的。使用该命令后，MCU必须通过CE引脚启动数据包的每次传输。

如果重发收到应答，则TX_DS被置高，且TXFIFO中数据被清除。如果超出最大重发次数，MAX_RT被置高，TXFIFO中的payload不会被清除。当中断标记为高时，IRQ管脚激活为低。重写中断位为1清除对应中断，**MAX_RT中断不清除，则TXFIFO被阻塞，不会发送任何数据包，直到用指令清除**

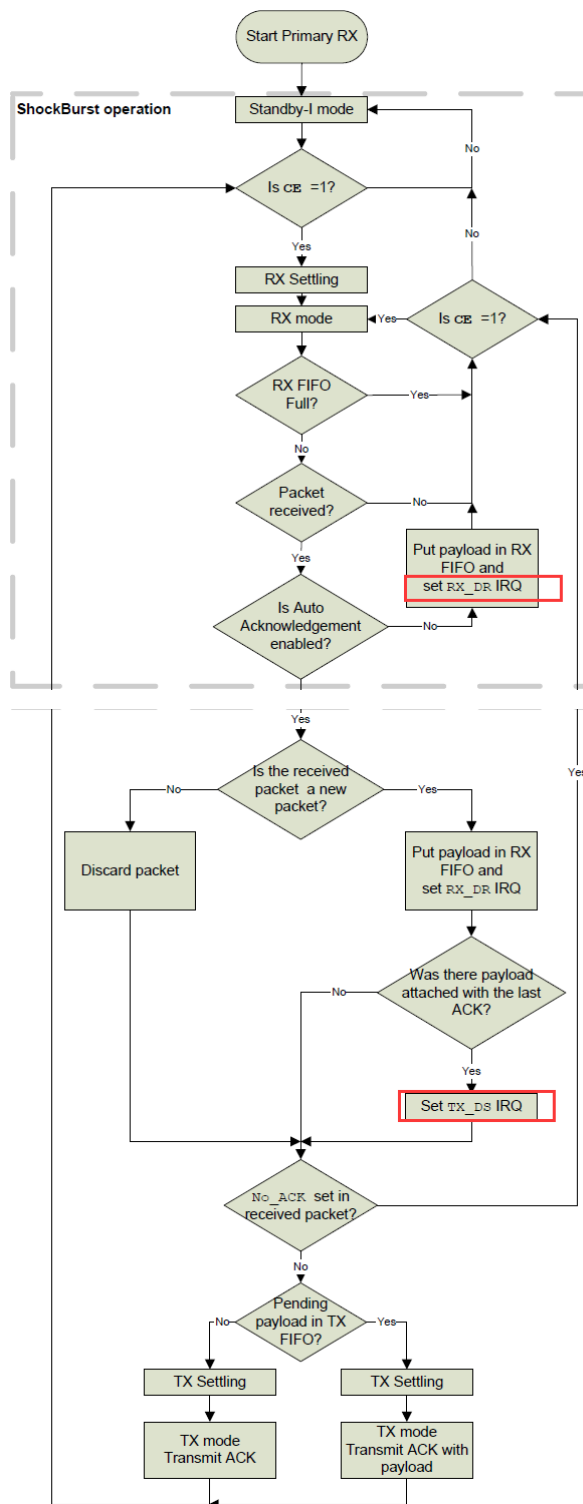
增强型发送流程图



流程可知：上电后设置成standby-1模式，CE=1且FIFO非空进入发送设置，发送数据包，如果没设置自动重发且机器检测NO_ACK flag被设置，则设置TX_DS IRQ。如果要求重发，转换成接受模式。如果收到ACK发送TX_DS IRQ,如果是带数据的ACK则RX_DR_IRQ也有。如果超出最高重发次数，则MAX_RT IRQ。如果CE=1而TXfifo空则进入sb2

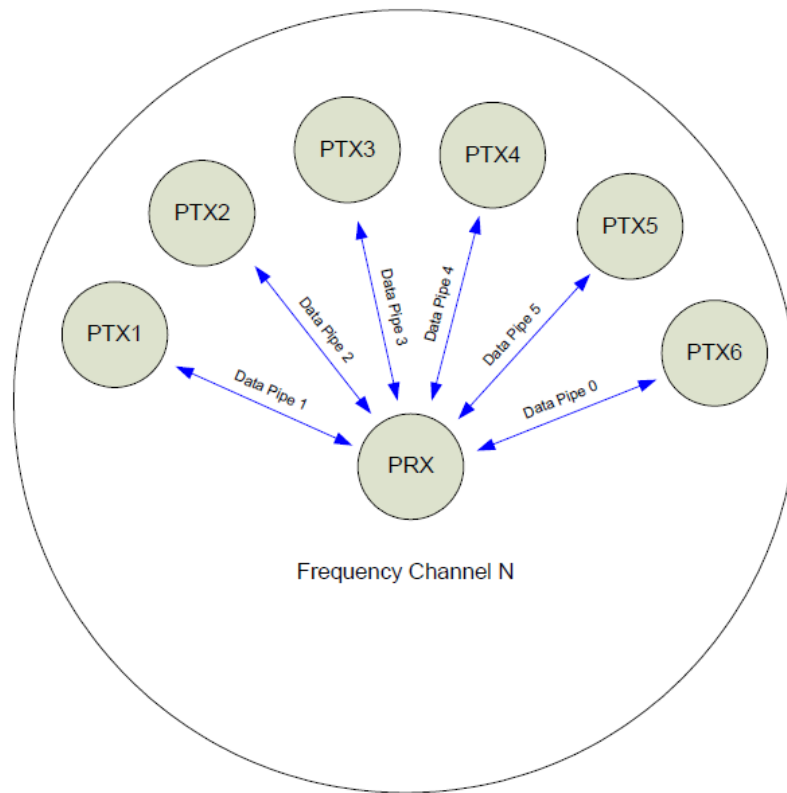
PRX流程图：

注：RX模式只要不CE=0，完成一次接收操作后(包括转换成TX应答)总会回到接收模式。



多个接收者

多接收器模式是在RX模式下使用的功能，它包含一组具有唯一地址的6个并行数据管道。每个数据通道都是物理通道的逻辑通道。每个数据通道有自己的物理地址。



设置成PRX的NRF可以同时在一个频道中监听来自6个不同数据通道的数据，每个数据通道都有自己的地址。多达六个PTX可以与一个PRX通信。**同时搜索所有数据通道地址，但是一次只有1个数据通道可以接收数据。**

数据通道可以通过设置EN_RXADDR寄存器。但默认只有数据通道0和1使能。

- 每个数据通道地址可以被配置在RX_ADDR_PX 寄存器。每个寄存器多大5字节地址。
- 数据通道0的地址可以单独配置。
- 数据通道1-5的地址共享四字节。最后字节

	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Data pipe 0 (RX_ADDR_P0)	0xE7	0xD3	0xF0	0x35	0x77
Data pipe 1 (RX_ADDR_P1)	0xC2	0xC2	0xC2	0xC2	0xC2
Data pipe 2 (RX_ADDR_P2)	0xC2	0xC2	0xC2	0xC2	0xC3
Data pipe 3 (RX_ADDR_P3)	0xC2	0xC2	0xC2	0xC2	0xC4
Data pipe 4 (RX_ADDR_P4)	0xC2	0xC2	0xC2	0xC2	0xC5
Data pipe 5 (RX_ADDR_P5)	0xC2	0xC2	0xC2	0xC2	0xC6

每个数据通道的如下数据是一致的。

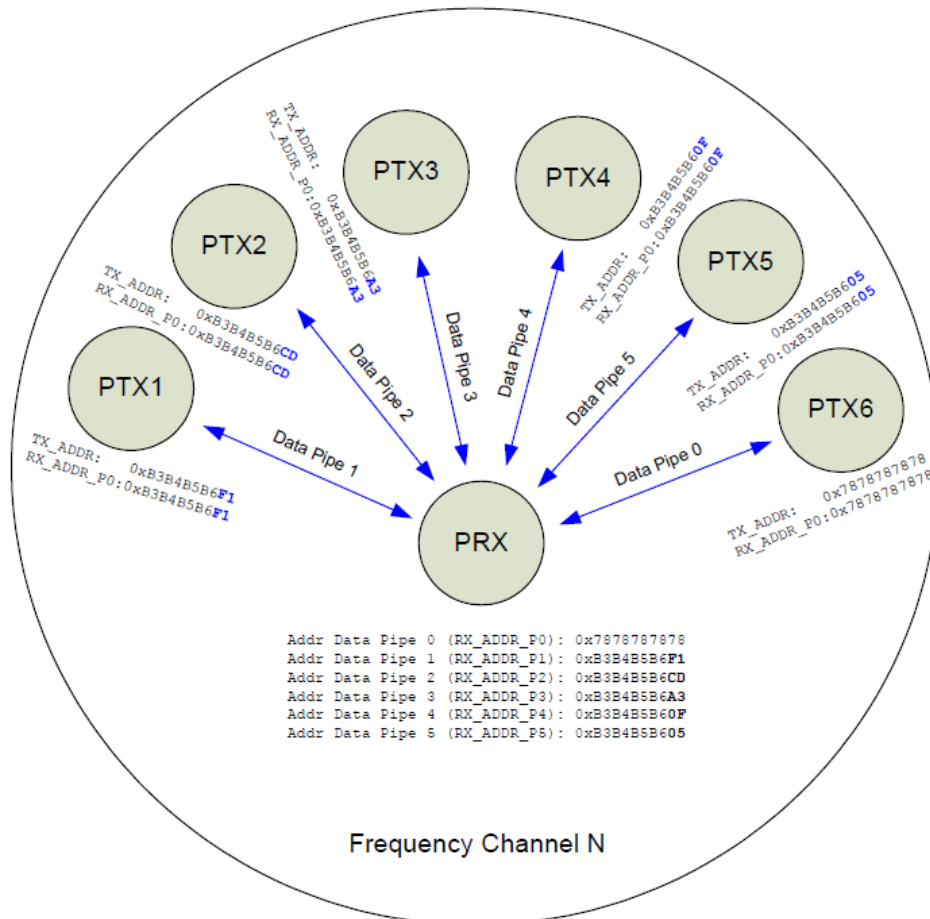
- CRC enabled/disabled (CRC always enabled when Enhanced ShockBurst™ is enabled)
- CRC encoding scheme
- RX address width
- Frequency channel
- Air data rate
- LNA gain

使用了多接收器和增强型Shockburst的PRX，当接受了1个数据来自某个PTX后，为了能够确保准确发送应答信号到响应的PTX，PRX将接收数据包的数据管道地址用作发送ACK数据包时的TX地址。

例子：PRX和PTX的地址配置，

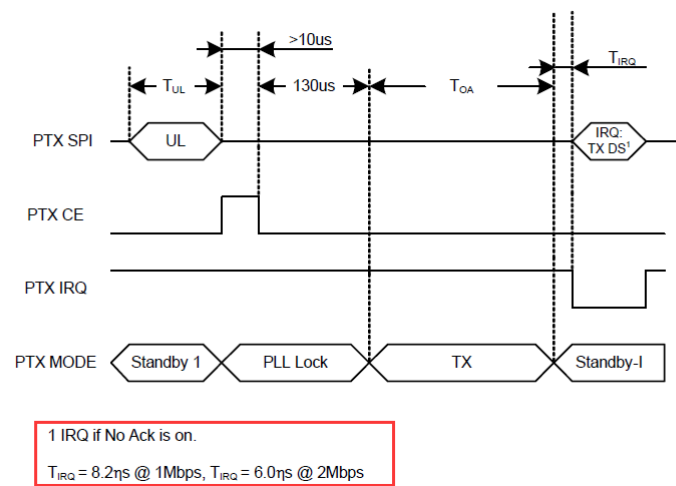
- 在PTX上，TX_ADDR (PTX) 必须与RX_ADDR_P0 (PTX) 相同(这样自动应答才能收到)，并且必须与指定管道的管道地址【RX_ADDR_P0 (PRX)】相同。
- 在PRX上，管道配置RX_ADDR_P0 (PRX) 与TX_ADDR (PTX) 一致。而TX_ADDR (PRX) 自动填充为响应接收管道的地址

注：当PRX搜索的某个通道地址接收到完整数据包之前，没有其他数据管道可以接收数据。当多个PTX发送的时候，使用自动重传可以让他们之间只阻塞一次。

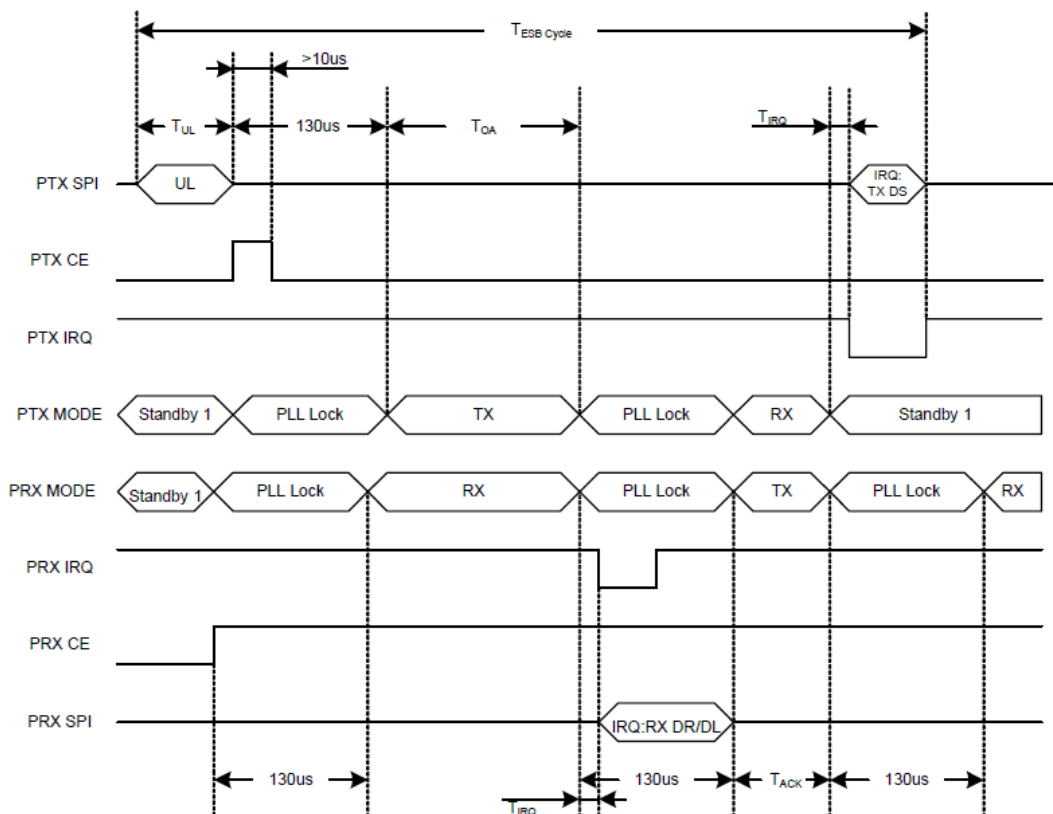


timing

在CE=0时才能用SPI写入？

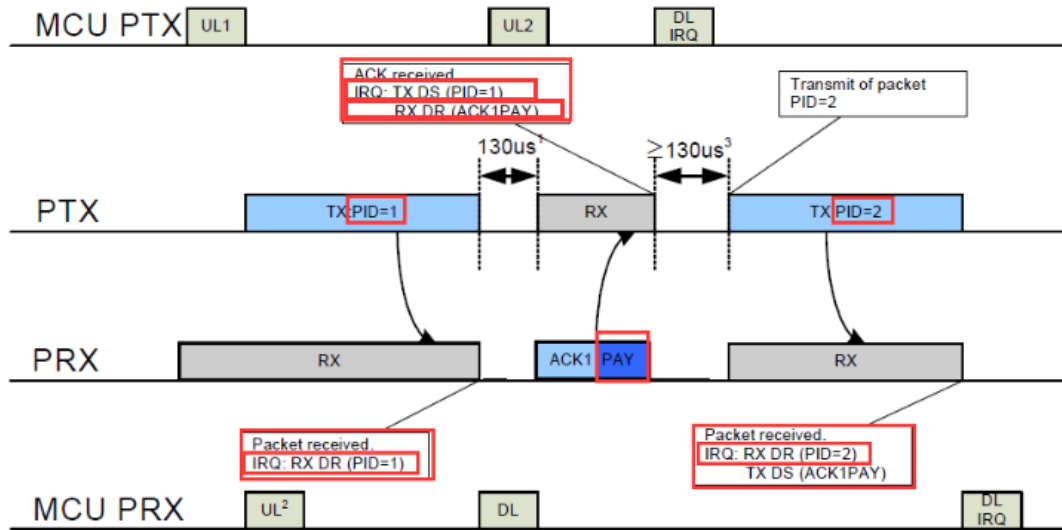


- TUL用SPI传输长度和SPI速率计算
- TOA 空中传播时间: $(8 \times (5 \text{至} 8 + N(\text{数据长度})) + 9) / \text{空中比特速率} (1Mbps \text{或} 2Mbps)$ 按最大长度和最大速率算。大概160us。最短长度和最大速度率算大概60us。
- TESB(1次ENHANCE shockburst循环时间): $TUL + 2T_{stby2a} + T_{ack} + T_{irq} = TUL + 260\mu s + 160\mu s$ 最大 + ...。大概不到500us ms。



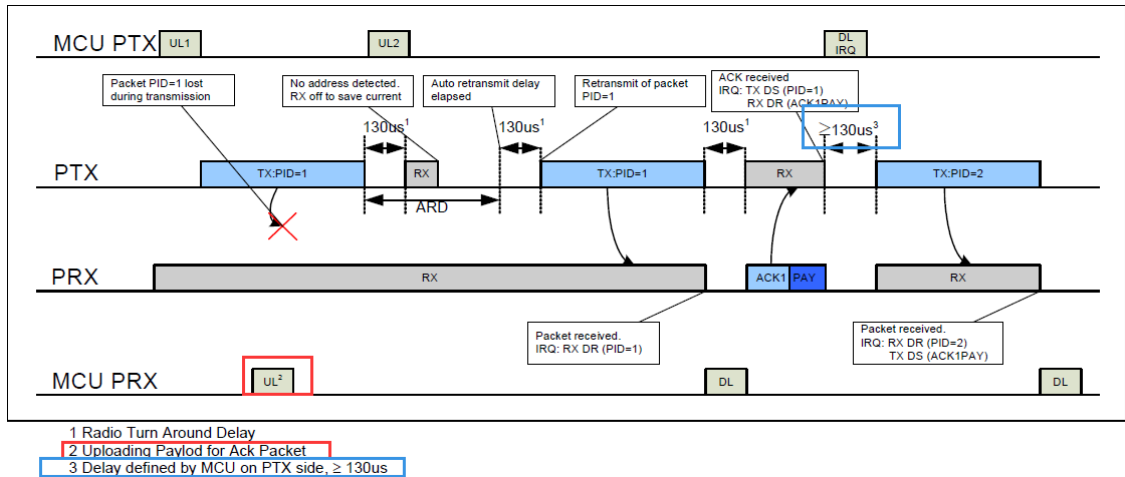
传输图:

1. 单次传输, 有ACK, 中断

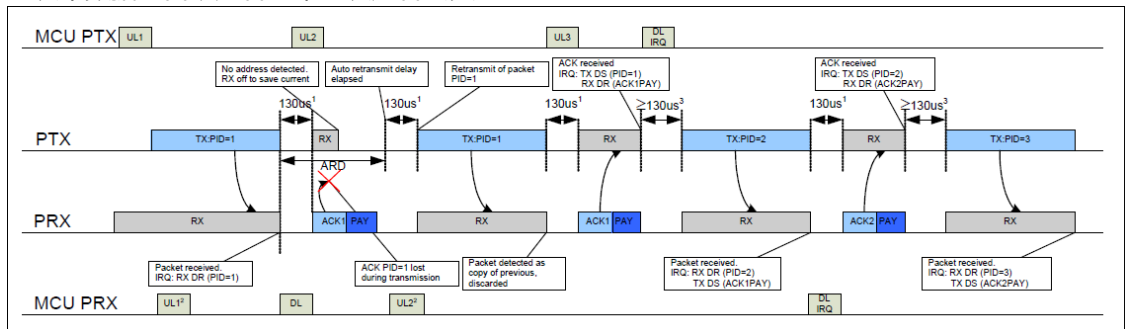


可见，每次新的TXFIFO数据发送完后，PID++。接收到带数据包的应答后IRQ两个中断都产生。PRX可以接收到数据后产生中断，然后发送带数据包应答。但在下次接收成功才产生应答完成中断。

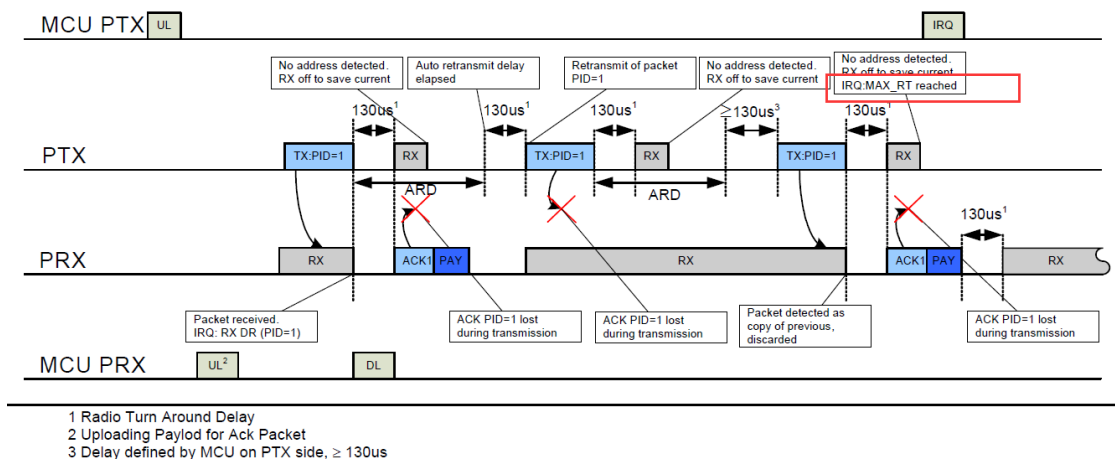
5. 单次带数据包应答丢失数据包



6. 两次带数据包自动应答且第一次应答丢失



7. 两次传输且最大重发次数达到



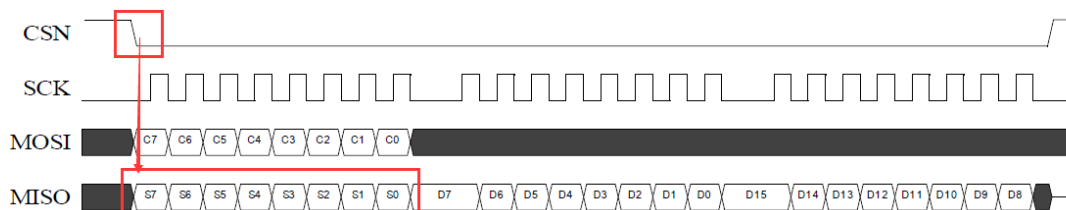
控制接口

注：先发低字节的高位

- 0-8M spi接口
- 8bit指令集
- 简易的寄存器映射
- 三级FIFOs

只有在standby和powerdown状态下可以写寄存器指令

- W_REGISTER和R_REGISTER命令可以对单字节或多字节寄存器进行操作。
- 访问多字节寄存器时，您首先要读取或写入LSByte的MSBit。
- 您可以在写入多字节寄存器中的所有字节之前终止写入，而使未写入的MSByte保持不变。
- 例如，可以通过仅向RX_ADDR_P0寄存器写入一个字节来修改RX_ADDR_P0的LSByte。
- 在CSN上从高到低的跳变后，**状态寄存器的内容总是读到MISO**。（SPI的CSN高到低跳变时）



注意：STATUS寄存器中的3位管道信息在IRQ引脚从高到低转换期间更新。如果在IRQ引脚从高电平到低电平过渡期间读取STATUS寄存器，则管道信息不可靠。

指令：

- REUSE_TX_PL：因为TX发送1次数据就会把对应数据从TXFIFO中删除，需要再次填词，如果使用这个命令就可以不需要填写，而一直自动重发上次数据包，直到重新写入数据。
- R_RX_PL_WID：读取命令之前，先通过 R_RX_PL_WID 获取有效数据长度，（再读取STATUS中的RX_P_NO来获取是哪个通道接收到的数据，其实在第一步R_RX_PL_WID的时候就自动收到了MISO来的STATUS数据），再用R_RX_PAYLOAD获取数据。（PTX没这个烦恼。PTX只能发给1个人，所以收到的也只能是这个人发来的，所以规定是PTX的pipe0接收）
- 读取STATUS有三种发放
 - SPI CSN由高->低时候MISO收到的

- 使用STATUS指令
 - 使用NOP空指令(方法1的一种快捷方式)
-

FIFO

- RXFIFO 3级32字节
- TXFIFO 3级32字节

两者都可以通过SPI的特殊指令来操作。

- 读写类操作，包括如何给TX FIFO写入数据，如何从RX FIFO读出数据，如果数据不想要了如何清空FIFO中的数据等等
- 查询类操作，包括FIFO中是否存在有效数据，FIFO是否已经写满，是否还有空余位置，RX FIFO某单元中有效数据来自哪个pipe，32字节里有多少是有效数据等等

TXFIFO可以被三个指令写入：且当MAX_RT IRQ发生时，TXFIFO不动

- W_TX_PAYLOAD 发送模式下
- W_TX_PAYLOAD_NO_ACK 发送模式下
- W_ACK_PAYLOAD 接收模式下
 - 指令中有位写是写给哪个pipe的应答数据，要指明。
 - TXFIFO最多挂3个带数据的应答。同pipe的先进先出原则。
 - 如果3个FIFO都被挂号满了，且应答发送链路丢失。则FIFO被阻塞，需要使用FLUSH_TX清除FIFO

RXFIFO可被读出

- R_RX_PAYLOAD 在发送和接收模式下都可以

FIFO寄存器可以读出TX/RX FIFO是否满还是空。且TX_REUSE位可以设置重发上个数据包，直到W_TX_PAYLOAD or FLUSH TX 被发送后停止重发。

中断引脚

- TX_DS IRQ
- RX_DR IRQ
- MAX_RT IRQ

三个都可以激活引脚为低电平。当对STATUS寄存器写IRQ bit 为1时复位高。且CONFIG寄存器可以配置中断屏蔽

发现：PRX的中断一定是有RX_DR的。

模块身份	中断组合	说明
PTX	MAX_RT	MAX_RT 只会单独出现
	TX_DS	当收到 PRX 发来的 auto-ack 时，如果 PRX 的回复中 不包含 有效数据，则 TX_DS 单独出现
	TX_DS + RX_DR	当收到 PRX 发来的 auto-ack 时，如果 PRX 的回复中 包含 有效数据，则 TX_DS 和 RX_DR 同时出现
PRX	RX_DR	当收到 PTX 发来的数据时，如果自己之前 没给 PTX 回复过有效数据，则 RX_DR 单独出现
	RX_DR + TX_DS	当收到 PTX 发来的 新数据(重发包不算) 时，如果自己之前 有给 PTX 回复过有效数据，则 RX_DR 和 TX_DS 同时出现，其中： RX_DR 表示“收到一份新数据”； TX_DS 表示“ 上次 回复的那份数据 PTX 已经收到了”。

寄存器映射

注：未定义的位读出来都是0

- CONFIG 配置寄存器 —— 中断屏蔽、上电、收发状态设置。
- EN_AA —— 使能6个数据通道的增强型shockburst 自动应答
- EN_RXADDR —— 使能6个数据通道地址
- SETUP_AW —— 设置地址宽度，少于5byte时低位生效。
- SETUP_RETR —— 设置自动重发：重发间隔和最大重发次数
- RF_CH —— 射频 频率通道
- RF_SETUP —— 射频设置：空中速率、功放增益、LNA使能
- STATUS —— 状态寄存器，当SPI发送指令时并行的自动从MISO串行输出(CSN 1->0触发)，对某bit写1将复位：三个中断标记位、RX_FIFO是从哪个数据通道PIPE得到的、TX_FIFO满了
- OBSERVE_TX —— 丢包次数(最大15，写RF_CH复位)、重发次数(发送新数据时复位)
- CD —— 载波检测,高的时候表示信道拥挤，低的时候表示通道可用。

载波检测—CD：

当接收端检测到射频范围内的信号时将 CD 置高，否则 CD 为低。内部的 CD 信号在写入寄存器之前是经过滤波的，内部 CD 高电平状态至少保持 128us 以上。

○

在增强型ShockBurst™模式中只有当发送模块没有成功发送数据时，推荐使用CD检测功能。如果发送端PLOS_CNT显示数据包丢失率太高时，可将其设置位接收模式检测CD值，如果CD为高（说明通道出现了拥挤现象），需要更改通信频道；如果CD为低电平状态（距离超出通信范围），可保持原有通信频道，但需作其它调整。

In RX mode a carrier detect signal is available. The carrier detect is a signal that is set high when a RF signal is detected inside the receiving frequency channel. The signal must be FSK modulated for a secure detection. Other signals can also be detected. The Carrier Detect (CD) is set high when an RF signal is detected in RX mode, otherwise CD is low. The internal CD signal is filtered before presented to CD register. The RF signal must be present for at least 128μs before the CD is set high. How to use the CD is described

○

- RX_ADDR_P0-P5 —— 多接收器模式时数据通道地址
- TX_ADDR —— 只对PTX有用。使用增强SHOCKBURST时配置成和 RX_ADDR_P0 一样来接收信息。
- RX_PW_P0-P5 —— 数据通道 数据宽度
- FIFO_STATUS —— FIFO状态寄存器：TX重发、TXFIFO满或空、接收FIFO满或空
- ACK_PLD —— 自动应答数据：通过SPI指令写入，只能在RX模式使用。

- TX_PLD —— TX发送的数据内容，3级fifo实现，通过SPI写入，只能在TX模式使用。
- RX_PLD —— RX接收的数据内容，3级fifo实现，通过SPI读出。所有数据通道**公用FIFO**
- DYNPD —— 使能动态数据长度，对应6个数据通道可以设置。
- FEATURE ——

发射流程

增强型的，设置config的接收模式位后CE=1就启动发射。

备注：

1. 2.4G WiFi的频段和nrf2401有很大一块频带是重叠的，如果你家有wifi信号，那很容易会受干扰而导致丢包严重，将nrf2401的频段设置为高于2490MHz可以显著降低丢包率。

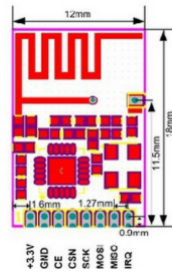
2. 全写在一个数组中这种样子发送也不错。

```
static const uint8_t nrf_init_seq_table[] PROGMEM =
{
    //nrf_rx_flush()
    0x01, NRF_CMD_FLUSH_RX,
    //nrf_tx_flush()
    0x01, NRF_CMD_FLUSH_TX,
    //nrf_clear_irq(0x70); //clear all IRQ
    0x02, NRF_CMD_W_REG(NRF_REG_STATUS), 0x70,
    //nrf_set_rf_channel
    0x02, NRF_CMD_W_REG(NRF_REG_RF_CH), (NRF_RF_CHANNEL-2400),
    //nrf_set_rf_data_rate(NRF_VAL_RF_DR_2Mbps);
    0x02, NRF_CMD_W_REG(NRF_REG_RF_SETUP), (NRF_AIR_DATA_RATE|NRF_VAL_RF_PWR_0dBm),
    // nrf_set_addr_wid(NRF_VAL_AW_5Bytes);
    0x02, NRF_CMD_W_REG(NRF_REG_SETUP_AW), NRF_VAL_AW_5Bytes,
    //set adr for pipe0, LSBytes first
    0x06, NRF_CMD_W_REG(NRF_REG_RX_ADDR_P0), NRF_DEF_ADR_BYTE, NRF_DEF_ADR_BYTE, NRF_DEF_ADR_BYTE, NRF_DEF_ADR_BYTE, NRF_DEF_ADR_BYTE,
    // nrf_set_feature((1<<NRF_BIT_EN_DPL)|(1<<NRF_BIT_EN_ACK_PAY)); //enable [Dynamic Payload Length]
    // and [Payload with ACK] (common for for all data pipes)
    0x02, NRF_CMD_W_REG(NRF_REG_FEATURE), (1<<NRF_BIT_EN_DPL)|(1<<NRF_BIT_EN_ACK_PAY),
    // nrf_enable_pipe((1<<NRF_PIPE0)); enable RX address of pipe0
    0x02, NRF_CMD_W_REG(NRF_REG_EN_RXADDR), (1<<NRF_PIPE0),
    // nrf_enable_pipe_auto_ack((1<<NRF_PIPE0)); enable auto-ACK for pipe0
    0x02, NRF_CMD_W_REG(NRF_REG_EN_AA), (1<<NRF_PIPE0),
    // nrf_enable_pipe_dpl((1<<NRF_PIPE0)); enable Dynamic Payload Length for pipe0
    0x02, NRF_CMD_W_REG(NRF_REG_DYNPD), (1<<NRF_PIPE0),
    //power up as PRX
    0x02, NRF_CMD_W_REG(NRF_REG_CONFIG), (1<<NRF_BIT_EN_CRC)|(1<<NRF_BIT_PRIM_RX)|(1<<NRF_BIT_PWR_UP),
    0x02, NRF_CMD_R_REG(NRF_REG_RX_ADDR_P0), 0x00,
    ////////////////////////////////////////////////////
    //END of config flow
    0x00
};
```





3. 不自己画了，用邮票孔形式的把



为了提高可靠度，可以学习增强型SHOCKBURST，在增强型SHOCKBURST模式下利用增强型SHOCKBURST原理在写1层应用层协议。