

文档编号: AN1000

上海东软载波微电子有限公司

应用笔记

HW2000 User Guide

修订历史

| 版本 | 修订日期 | 修改概要 |
|------|----------|--------|
| V1.0 | 2018-5-8 | 初版公开发布 |
| | | |
| | | |

地 址：中国上海市龙漕路 299 号天华信息科技园 2A 楼 5 层

邮 编：200235

E-mail: support@essemi.com

电 话：+86-21-60910333

传 真：+86-21-60914991

网 址：http://www.essemi.com

版权所有©

上海东软载波微电子有限公司

本资料内容为上海东软载波微电子有限公司在现有数据资料基础上慎重且力求准确无误编制而成，本资料中所记载的实例以正确的使用方法和标准操作为前提，使用方在应用该等实例时请充分考虑外部诸条件，上海东软载波微电子有限公司不担保或确认该等实例在使用方的适用性、适当性或完整性，上海东软载波微电子有限公司亦不对使用方因使用本资料所有内容而可能或已经带来的风险或后果承担任何法律责任。基于使本资料的内容更加完善等原因，上海东软载波微电子有限公司保留未经预告的修改权。使用方如需获得最新的产品信息，请随时用上述联系方式与上海东软载波微电子有限公司联系。

目 录

内容目录

| | | |
|--------------|----------------------|-----------|
| 第 1 章 | 编程说明 | 5 |
| 1.1 | 工作模式 | 5 |
| 1.2 | 初始化 | 5 |
| 1.3 | 数据包收发 | 5 |
| 1.3.1 | 硬件链路控制方式 | 5 |
| 1.3.2 | 软件链路控制方式 | 11 |
| 1.4 | 寄存器初始化设置 | 12 |
| 第 2 章 | RF 测试说明 | 15 |
| 2.1 | PA 功率 | 15 |
| 2.2 | RSSI | 16 |
| 第 3 章 | 芯片故障分析 | 17 |
| 第 4 章 | 常见问题 | 18 |

图目录

| | |
|-------------------------------------|----|
| 图 1-1 ACK PAYLOAD 功能不使能收发控制流程..... | 8 |
| 图 1-2 ACK 带 ACK PAYLOAD 收发控制流程..... | 9 |
| 图 1-3 FIFO 数据包重复发送控制流程..... | 10 |
| 图 1-4 软件链路收发控制流程..... | 11 |
| 图 2-1 发送功率与配置..... | 15 |
| 图 2-2 RSSI 寄存器与输入能量对应关系..... | 16 |

表目录

| | |
|-----------------------------|----|
| 表 1-1 1Mbps 寄存器初始化设置..... | 12 |
| 表 1-2 250Kbps 寄存器初始化设置..... | 13 |
| 表 1-3 16M/20M 晶振参数配置..... | 13 |
| 表 2-1 不同发送功率下寄存器配置值..... | 15 |

第 1 章 编程说明

1.1 工作模式

HW2000 芯片主要有 POWER DOWN、SLEEP、IDLE、TX、RX 五个工作模式，各个工作模式之间的切换见芯片数据手册的“芯片工作模式控制”章节内容。

1.2 初始化

- ◆ 芯片上电
- ◆ 寄存器初始化配置（以下配置不分先后）
 - ✧ 射频前端寄存器初始化，详见“寄存器初始化设置”章节。
 - ✧ 发送速率 RATE 设置(0x2A)，默认 1Mbps。
 - ✧ 外部晶振参考频率 REF_FQ，频点 PLL_CH_NO 设置(0x22)。
 - ✧ PREAMBLE 长度、SYNCWORD 长度、TRAILER 长度、编码方式、是否支持 FEC 设置(0x20)。
 - ✧ 同步字允许错误个数阈值 SYNC_THRES 设置(0x28)。
 - ✧ 是否支持扰码功能，CRC8/16 选择设置(0x29)。
 - ✧ 使能通讯使用的 PIPE、设置 PIPE 地址、以及该 PIPE 是否支持 ACK(0x3C、0x40~0x47)。
 - ✧ 如果支持 ACK 模式，设置重发次数 RE_TX_TIMES (0x23，设置值需大于‘1’)。

注：以上寄存器设置针对硬件链路控制方式，软件链路控制方式不支持 CRC、FEC、ACK 功能，可忽略相应寄存器配置步骤。

1.3 数据包收发

HW2000 收发支持硬件链路控制（默认方式）与软件链路控制两种方式，可通过寄存器 PACK_LENGTH_EN (0x29[12])配置。

1.3.1 硬件链路控制方式

硬件链路控制方式需设置 PACK_LENGTH_EN 为‘1’。此模式支持两级收发 FIFO 与两级 ACKFIFO，每级 FIFO 最大支持 63bytes PAYLOAD，每级 ACKFIFO 最大支持 32bytes ACK PAYLOAD。

PTX 在此控制方式下所填 FIFO/ACKFIFO 的第一个 byte 代表该级 FIFO/ACKFIFO 的 PAYLOAD 长度（填写值需大于‘0’）。

PRX 可依据收取 PAYLOAD 的 FIFO 第一个 byte 值确定所需读取 PAYLOAD 的长度，类似的，PTX 可依据收取 ACK PAYLOAD 的 ACKFIFO 第一个 byte 值确定所需读取 ACK PAYLOAD 的长度。各级 FIFO 需满足条件才能进行收发。

1.3.1.1 PTX发送PAYLOAD流程

FIFO0 发送有效状态

- ✧ FIFO0_EN 为‘1’
- ✧ PTX_FIFO0_OCPY 为‘1’
- ✧ FIFO0 内 PAYLOAD 所属 PIPE 使能（例如使用 PIPE0 通信需使能 P0_EN 且设置 PTX_FIFO0_PIPE 为‘3'b000’）
- ✧ INT0 为‘0’

FIFO1 发送有效状态

- ✧ FIFO0 不处于发送有效状态（发送 FIFO0 优先级高于 FIFO1）
- ✧ FIFO1_EN 为‘1’
- ✧ PTX_FIFO1_OCPY 为‘1’
- ✧ FIFO1 内 PAYLOAD 所属 PIPE 使能（例如使用 PIPE0 通信，需使能 P0_EN 且设置 PTX_FIFO1_PIPE 为‘3'b000’）
- ✧ INT1 为‘0’

对于 PTX，通过 SPI 接口以不同的 FIFO 入口地址向 FIFO0(0x32)、FIFO1(0x33)填写发送的 PAYLOAD 数据。填写 PAYLOAD 完成之后，需软件填写 0x36/0x37 寄存器的 PTX_FIFO_n_PIPE（默认为 PIPE0）与 PTX_FIFO_n_OCPY 位（置‘1’），以指示所填的 FIFO 被占用并说明 FIFO 中 PAYLOAD 所属的 PIPE。

硬件状态机依次依据 FIFO0 与 FIFO1 的 PTX_FIFO_n_OCPY 位并自动匹配 PTX_FIFO_n_PIPE 位指示的 PIPE 地址，向 FIFO 中取数发送。

发送完成后状态机将检查是否有其它 FIFO 处于发送有效状态以继续发送。

待 PTX 发送完成中断 IRQ 置起后，需软件依次清相应 PTX_FIFO_n_OCPY 位与中断标志位 INT_n。（详见图 1-1）

1.3.1.2 PRX接收PAYLOAD流程

FIFO0 接收有效状态

- ✧ FIFO0_EN 为‘1’
- ✧ INT0 为‘0’

FIFO1 接收有效状态

- ✧ FIFO0 不处于接收有效状态（接收 FIFO0 优先级高于 FIFO1）
- ✧ FIFO1_EN 为‘1’
- ✧ INT1 为‘0’

对于 PRX，硬件在匹配接收 PIPE 地址之后，依次依据 FIFO 的有效状态将接收 PAYLOAD 填入处于接收有效状态下的 FIFO，填写完成之后自动将相应 PRX_FIFO_n_OCPY 位置‘1’，并填写 PRX_FIFO_n_PIPE 寄存器以指示该 FIFO 接收 PAYLOAD 所对应的 PIPE。

待接收 FIFO 中断 IRQ 置起后，软件需查询 INT_n 指示位（0x3D）以确定收取 PAYLOAD 的 FIFO。读取 PAYLOAD 之后需将 INT_n 标志清‘0’以保证后续 PAYLOAD 接收。

1.3.1.3 PRX发送ACK PAYLOAD流程

ACKFIFO0/1 满足发送条件

- ✧ ACKFIFO 配置寄存器 0x38/0x39 中所填 PIPE (PRX_ACKFIFO_n_PIPE)与当前收取帧的 PIPE 地址匹配
- ✧ ACKFIFO_n_EN 为‘1’
- ✧ PRX_ACKFIFO_n_OCPY 为‘1’

当 PRX 使能 ACK 带 ACK PAYLOAD 功能时 (Pn_EN = ‘1’, Pn_ACK_EN = ‘1’并且 Pn_ACKPAYLOAD_EN = ‘1’), PRX 在向 PTX 返回 ACK 的过程中自动从满足发送条件的 ACKFIFO 中取出 ACK PAYLOAD 进行发送。

1.3.1.4 PTX接收ACK PAYLOAD流程

ACKFIFO0 接收有效状态

- ✧ ACKFIFO0_EN 为‘1’
- ✧ ACKINT0 为‘0’

ACKFIFO1 接收有效状态

- ✧ ACKFIFO0 不处于接收有效状态 (接收 ACKFIFO0 优先级高于 ACKFIFO1)
- ✧ ACKFIFO1_EN 为‘1’
- ✧ ACKINT1 为‘0’

当 PTX 使能 ACK 带 ACK PAYLOAD 功能时 (Pn_EN = ‘1’, Pn_ACK_EN = ‘1’并且 Pn_ACKPAYLOAD_EN = ‘1’), PTX 在接收到 ACK PAYLOAD 后将数据填入满足条件的 ACKFIFO 中, 填写完成之后自动将相应 PTX_ACKFIFO_n_OCPY 位置‘1’, 并填写 PTX_ACKFIFO_n_PIPE 寄存器以指示该 ACKFIFO 接收 ACK PAYLOAD 所对应的 PIPE。

待 PTX 的中断置起后, 软件需查询 ACKINT_n 指示位 (0x3D) 以确定收取 ACK PAYLOAD 的 ACKFIFO。读取 ACK PAYLOAD 之后软件需将 ACKINT_n 标志清‘0’以保证后续 ACK PAYLOAD 接收。

1.3.1.5 操作流程

- ◆ FIFO0 收发 ACK PAYLOAD 功能不使能

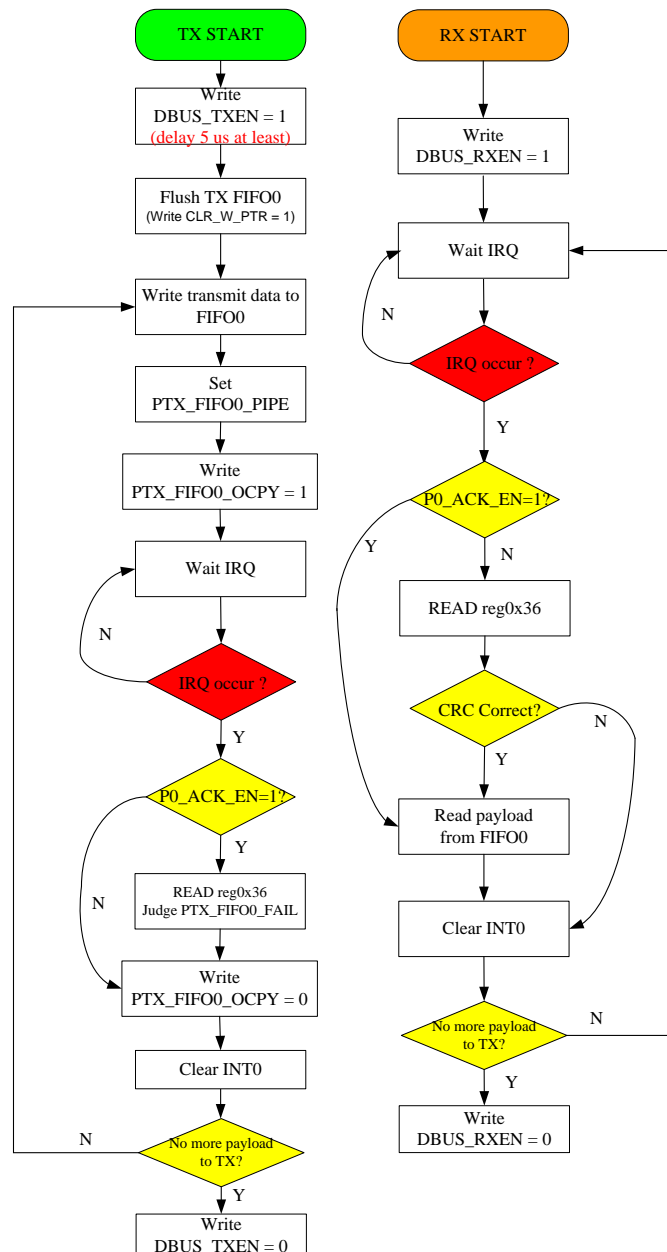


图 1-1 ACK PAYLOAD 功能不使能收发控制流程

上图所示为硬件链路收发方式下只使用 FIFO0 进行收发的操作流程，使用 FIFO1 或两级 FIFO 的流程基本类似，即需发送与接收 FIFO 满足上述条件才可进行正常收发流程。

- ✧ 若使能 ACK 功能（P0_ACK_EN = '1'），PTX 在读取中断 INT0 之后需检查 PTX_FIFO0_FAIL 标志位，以判断中断源为 PTX 接收 ACK 成功或是重传超时。
- ✧ PRX 在 ACK 不使能情况下，响应中断之后可以读取相应 FIFO 的 CRC 检测标志位判断接收 PAYLOAD 是否正确。
- ✧ PRX 在 ACK 使能情况下接收 PAYLOAD CRC 出错将自动重收，所以不会出现正常中断置起接收 PAYLOAD 出错的情况（无需检测 CRC）。

注 1：芯片上电默认为 SLEEP 状态，该状态下可进行寄存器初始化操作。

注 2：PTX 有发送需求使能 DBUS_TXEN 后，需等待 5us，系统时钟稳定后方可对 FIFO 进行操作。

◆ FIFO0 收发 ACK PAYLOAD 功能使能

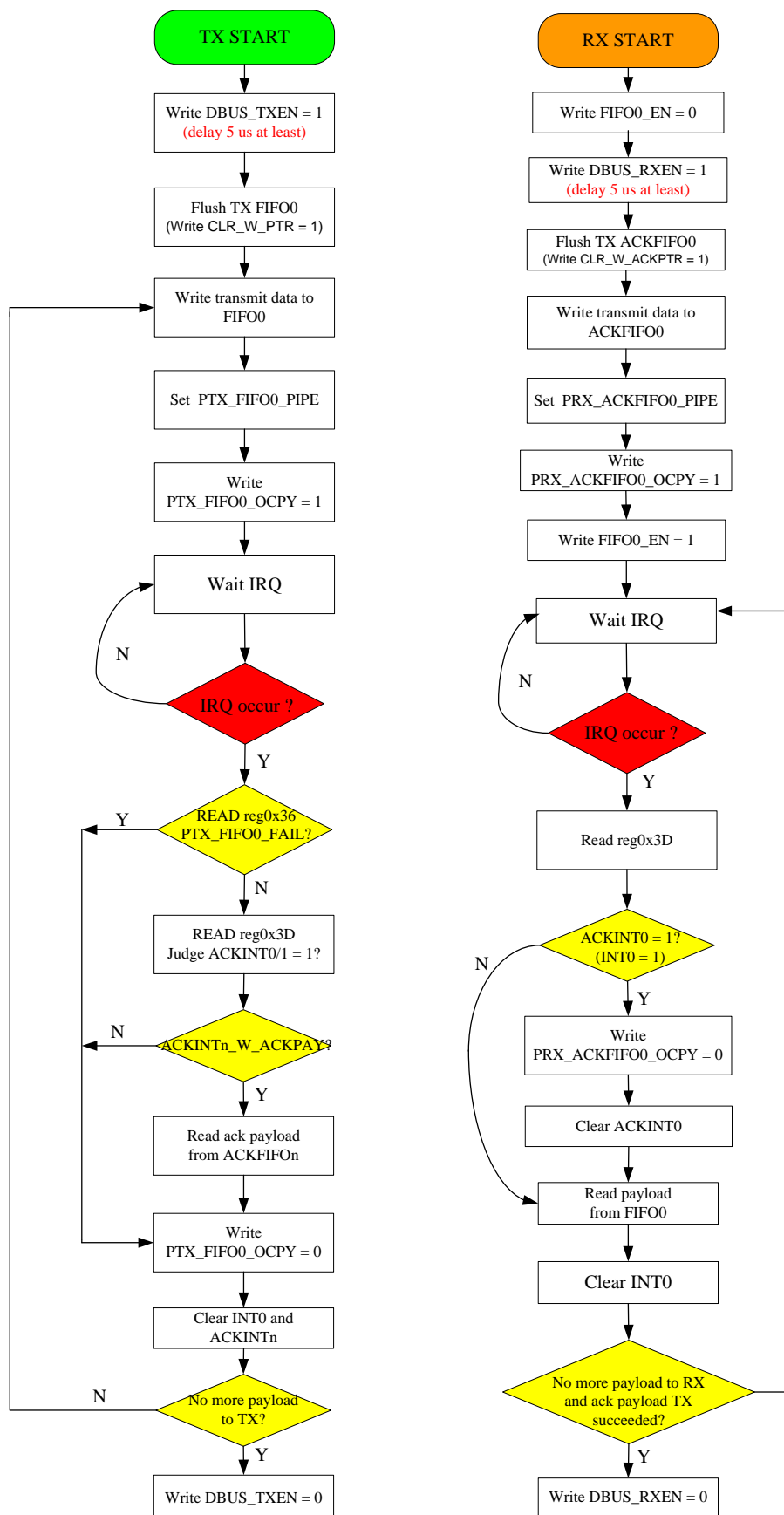


图 1-2 ACK 带 ACK PAYLOAD 收发控制流程

上图所示为只使用 FIFO0 收发 PAYLOAD、ACKFIFO0 发送 ACK PAYLOAD 的收发流程。

- ✧ ACKFIFO 的配置方式、流程与 FIFO 基本一致, PTX 需对发送 FIFO 与接收 ACKFIFO 进行配置, PRX 需对接收 FIFO 与发送 ACKFIFO 进行配置。
- ✧ PTX 在响应中断时需读取 0x3D 寄存器以确认哪个 ACKFIFO 成功接收 ACK PAYLOAD (优先级 ACKFIFO0 高于 ACKFIFO1)。
例如 ACKINT0 = '1', ACKINT0_W_ACKPAY = '1' 表示 ACKFIFO0 成功接收 ACK PAYLOAD; ACKINT0 = '1', ACKINT0_W_ACKPAY = '0' 表示此次接收 ACK 过程中未发现 ACK PAYLOAD。
- ✧ PRX 在返回 ACK 过程中若无满足发送条件的 ACKFIFO, 将只返回 ACK 信号, PTX 可依据 ACKINTn_W_ACKPAY 指示信号确认。
- ✧ PRX 在发送 ACK PAYLOAD 之后需在下次收到同一 PIPE 新包之后才将 ACKINTn 标志置起。

◆ FIFO 数据包重复发送功能

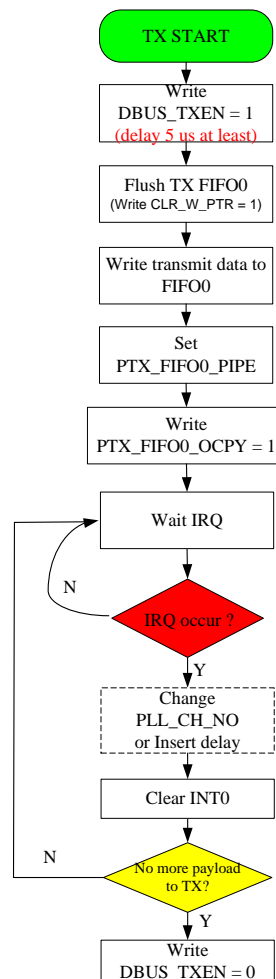


图 1-3 FIFO 数据包重复发送控制流程

芯片支持 FIFO 数据包重复发送功能, 以满足某些数据包重复发送或快速跳频等应用场合, 可以有效减少主控 MCU 的软件开销 (如图 1-3 所示)。芯片在一次发送完成后只通过清中断标志 INTO 操作即可实现 FIFO 内数据包的重发流程, 用户可在清中断标志 INTO 之前插入延时控制重发时间间隔或更改发送频点。

注: 频点设置需在芯片发送或接收状态有效之前完成。

1.3.2 软件链路控制方式

软件链路控制方式需设置 `PACK_LENGTH_EN` 为'0'。此模式只支持 `FIFO0` 一级 `FIFO`，不支持 `ACK` 与硬件 `CRC` 校验功能。该模式下需要软件不断查询 `FIFO0` 的半空或半满标志位，完成对 `FIFO0` 的写入与读取操作，配合物理层硬件对 `PAYLOAD` 数据的发送与接收。具体操作流程可参考下图。

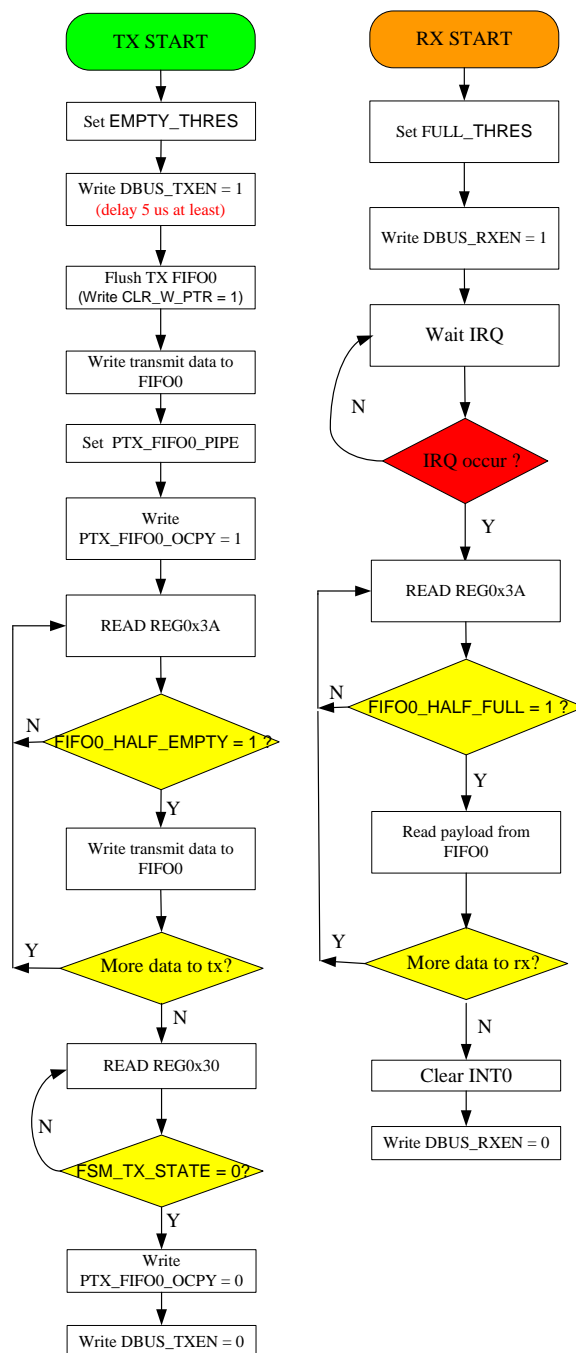


图 1-4 软件链路收发控制流程

注：可通过 `EMPTY_THRES[4:0](0x28[15:11])`，`FULL_THRES[4:0](0x28[10:6])` 寄存器设置半空半满阈值，设置值需考虑 `SPI` 接口的访问速度。

1.4 寄存器初始化设置

表 1-1 与表 1-2 分别为使用 12MHz 晶振时，1Mbps 与 250Kbps 的寄存器初始化配置(收发双方)，请按以下给定顺序进行初始化设置。

| Reg address (Hex) | Default Value (Hex) | Recommend Value (12MHz crystal frequency) (Hex) |
|----------------------|------------------------|---|
| 4C | 0000 | 55AA（开启内部寄存器写使能） |
| 12~15 | 0000 | 见表 2-1 |
| 50 | 0000 | 0012 |
| 51 | 0000 | 0012 |
| 52 | 0000 | 0012 |
| 53 | 0000 | 0012 |
| 54 | 0000 | 0012 |
| 55 | 0000 | 0012 |
| 56 | 0000 | 0052 |
| 57 | 0000 | 0249 |
| 58 | 0000 | 0251 |
| 59 | 0000 | 0252 |
| 5A | 0000 | 0451 |
| 5B | 0000 | 0491 |
| 5C | 0000 | 04D1 |
| 5D | 0000 | 04D9 |
| 5E | 0000 | 06D8 |
| 5F | 0000 | 06E0 |
| 60 | 0000 | 06E1 |
| 61 | 0000 | 06E2 |
| 01 | 0000 | 4D55 |
| 02 | 0000 | 44CC |
| 06 | 0000 | B000 |
| 07 | 0000 | 54E0 |
| 08 | 0000 | 7BC8 |
| 09 | 0000 | CC00 |
| 19 | 0000 | 2084 |
| 1B | 0000 | E754 |
| 1C | 0000 | 51A0 |
| 2A | C07D | C0E4 |
| 2C | 8883 | 918B |
| 4C | 0000 | FFFF（关闭内部寄存器写使能） |

表 1-1 1Mbps 寄存器初始化设置

注：ACK 模式下 0x2A 寄存器配置请参考 HW2000_Datasheet_C 寄存器说明。

| Reg address (Hex) | Default Value (Hex) | Recommend Value (12MHz crystal frequency) (Hex) |
|----------------------|------------------------|---|
| 4C | 0000 | 55AA(开启内部寄存器写使能) |
| 12~15 | 0000 | 见表 2-1 |
| 50 | 0000 | 0012 |
| 51 | 0000 | 0012 |
| 52 | 0000 | 0012 |
| 53 | 0000 | 0012 |
| 54 | 0000 | 0012 |
| 55 | 0000 | 0012 |
| 56 | 0000 | 0052 |
| 57 | 0000 | 0249 |
| 58 | 0000 | 0251 |
| 59 | 0000 | 0252 |
| 5A | 0000 | 0451 |
| 5B | 0000 | 0491 |
| 5C | 0000 | 04D1 |
| 5D | 0000 | 04D9 |
| 5E | 0000 | 06D8 |
| 5F | 0000 | 06E0 |
| 60 | 0000 | 06E1 |
| 61 | 0000 | 06E2 |
| 01 | 0000 | 4D55 |
| 02 | 0000 | 44CC |
| 08 | 0000 | 7BC8 |
| 09 | 0000 | CC00 |
| 19 | 0000 | 0884 |
| 1A | 0000 | 0D31 |
| 2A | C07D | 40B4 |
| 2C | 8883 | 918B |
| 4C | 0000 | FFFF(关闭内部寄存器写使能) |

表 1-2 250Kbps 寄存器初始化设置

注：ACK 模式下 0x2A 寄存器配置请参考 HW2000_Datasheet_C 寄存器说明。

若使用 16M 或 20M 晶振，需另外添加如下寄存器初始化：

| 晶振 (MHz) | 寄存器配置(Hex) | | |
|-------------|------------|------|------|
| | 0x04 | 0x1C | 0x22 |
| 16 | 4800 | 5198 | 2030 |
| 20 | 4A00 | 5193 | 2830 |

表 1-3 16M/20M 晶振参数配置

注：使用 16M/20M 晶振，以上寄存器初始化需在表 1-1 或表 1-2 首尾寄存器初始化之间完成。

芯片提供低功耗接收模式，在该模式下灵敏度下降约 6dB，接收功耗约为 16mA。请将表 1-1 与表 1-2 中 0x08 寄存器初始化值由 0x7BC8 改为 0x0BC8。

第 2 章 RF测试说明

2.1 PA功率

图 2-1 为 HW2000 芯片在 VDD = 3.0V 室温下的 PA 输出测试曲线图（横坐标为 0x15[15:10] 寄存器设置值，纵坐标为输出功率），表 2-1 为常用输出功率下的寄存器配置。

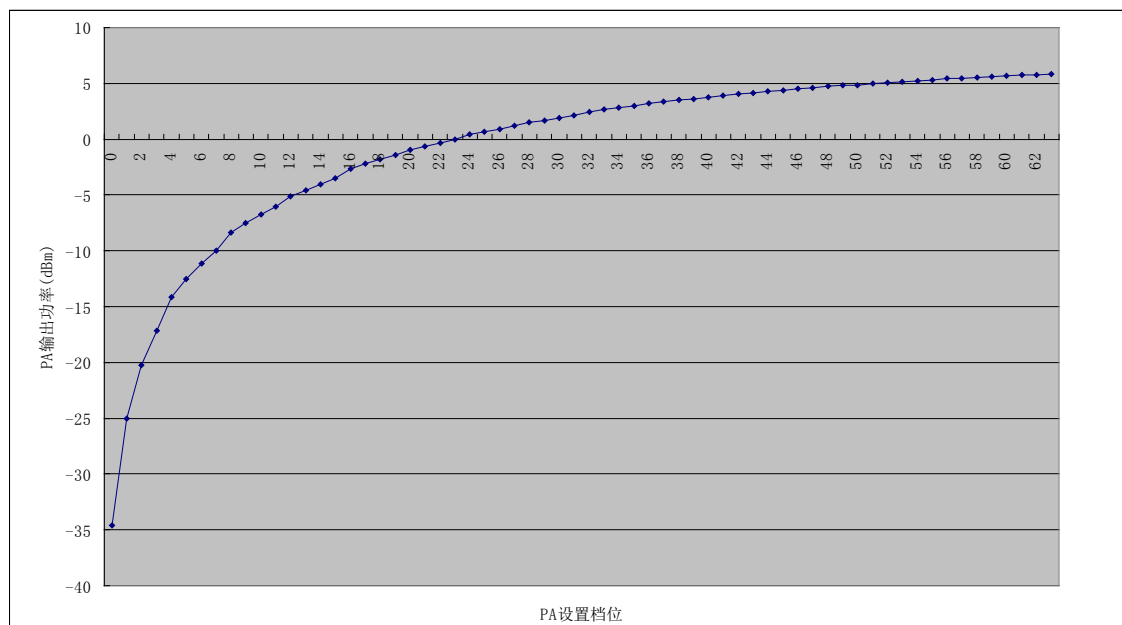


图 2-1 发送功率与配置

| 发送功率 (dBm) | 寄存器配置(Hex) | | | |
|---------------|------------|------|------|------|
| | 0x12 | 0x13 | 0x14 | 0x15 |
| +8 | 3043 | 4000 | 6032 | FC0C |
| +5 | 2043 | 3C00 | 7C32 | 840C |
| +2 | 1043 | 5000 | 5C32 | 600C |
| 0 | 1043 | 5000 | 5C32 | 440C |
| -5 | 1043 | 2000 | 3032 | 240C |
| -10 | 0843 | 1000 | 1832 | 140C |
| -15 | 0443 | 0800 | 0C32 | 0C0C |
| -20 | 0443 | 0800 | 0832 | 080C |
| -35 | 0043 | 0000 | 0032 | 000C |

表 2-1 不同发送功率下寄存器配置值

注：以上配置仅作参考，芯片实际输出功率受 PCB 外围影响较大，芯片最终输出功率由 0x15[15:10] 设置值决定（设置值与输出功率趋势如图 2-1 所示），可在以上参考功率档位设置基础上调整 0x15[15:10] 设置值（保持 0x12[9:0]、0x13[9:0]、0x14[9:0]、0x15[9:0] 寄存器设置不变）以得到实际需要的发送功率。

2.2 RSSI

测试方法：

1. 芯片上电，初始化 HW2000 寄存器，见“寄存器初始化设置”章节。
2. 将芯片设为 RX 状态。
3. 改变输入能量，当读取 PIPE_ADDR_MATCH 寄存器（0x30）为‘1’时，读取 RSSI(0x2D，补码形式)。若需检测环境能量，建议在 RX 接收使能后延时 350us 左右读取 RSSI。

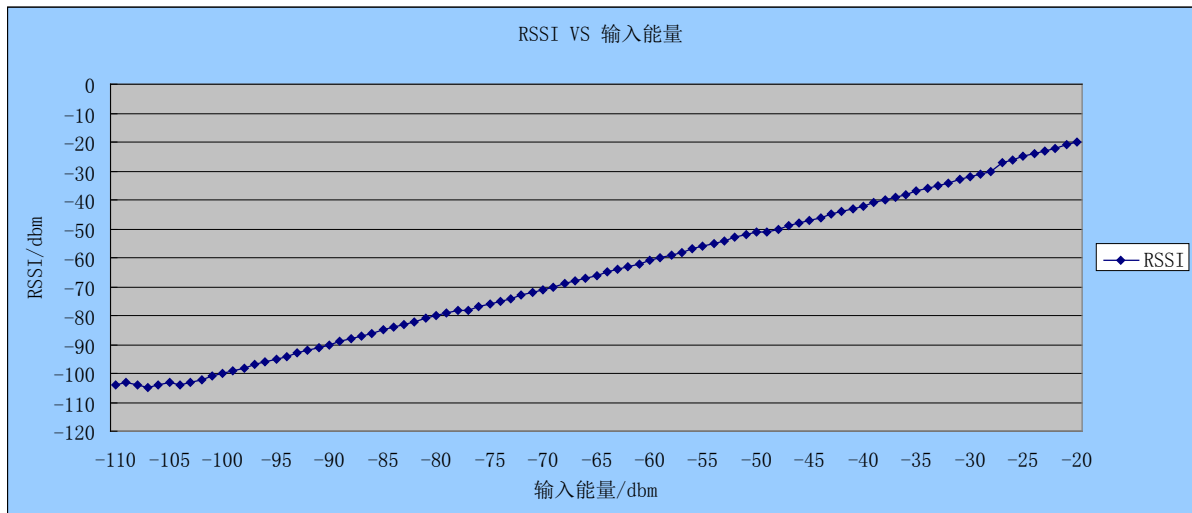


图 2-2 RSSI 寄存器与输入能量对应关系

注：推荐在数据包地址相关完成后(PIPE_ADDR_MATCH 为‘1’)读取 RSSI 数值，数据包接收完成后 RSSI 寄存器将被清‘0’。

第 3 章 芯片故障分析

如果两颗芯片不能进行正常收发通讯，首先应该将这两颗芯片分别与已测试过的正常芯片进行收发通讯，确定发射芯片还是接收芯片出现问题。在此前提下，可参照以下内容分析芯片发送或接收故障的原因。

✧ 检查 SPI 读写是否正常

对寄存器进行读写操作检查 SPI 驱动程序是否正确。如果寄存器写入与读取值不一致，利用示波器抓取 SPI 读写波形，检查 SPI 四根线的电平是否正确，波形时序是否和产品手册一致；检查 SPI 通讯速率是否小于给定的最大通讯速率。

✧ 软件检查

检查软件对 HW2000 芯片的寄存器初始化配置是否与参考代码有差异，软件收发流程是否与操作例程有差异。建议先使用最简单的 NOACK 模式进行调试。

✧ 检查 ISM 频段有无干扰

测试 ISM 频段(2400MHz~2483MHz)有无干扰的最直接方法是将 2.4G 天线连接到频谱仪，直接测空间辐射的信号，设置频谱仪 Span=2300MHz~2500MHz，Ref Amplitude<-50dBm。如有较大空间干扰，建议避开此频点通讯。

✧ 连续发送模式

向 FIFO0 内填写 0x55 或 0xAA(64 bytes)，将 0x29 设置为 0x0000，使能发送。观察发射频点是否锁定，与设置值是否一致。

✧ 检查发送频偏

检查连续发送模式下中心频点频偏值是否小于 100KHz(测量值频率与理论值频率之间的偏差)。

✧ 检查接收频偏

检查接收频偏值是否小于 100KHz，频偏补偿方法详见 HW2000 芯片数据手册的“自动频偏校正”章节。

✧ 检查接收本振是否锁定

判断接收本振是否锁定需要使用频谱仪观察，设置频谱仪 Span=250KHz，Ref Amplitude=-50dBm。使能接收，如果接收频点一直停留在设定值不跳动就说明已锁定。

✧ 检查晶振是否正常工作

上电后测试晶振 XTALP 或 XTALN 引脚，如果有信号说明晶振正常，否则可能是晶振损坏。

✧ 晶振频偏调整

调整晶振外接的两个电容 C11 和 C12（详见应用原理图），如果调整以后的晶振频率值一致性仍很差，则需要更换晶振，提高晶振精度 ppm。

第 4 章 常见问题

✧ 芯片发送或接收频点为什么与设置值不一致？

频点设置需在芯片发送或接收状态有效之前完成，否则芯片内部 PLL 将无法正确锁定。

✧ 写指针在什么时候需要软件清‘0’？

芯片内状态机写 FIFO（只出现在 PRX 端）的指针与 SPI 写 FIFO（只出现在 PTX 端）的指针复用同一个指针。PTX 与 PRX 在正常收发不切换时，硬件在合适情况下自动清写指针无需软件参与，但在收发角色切换（PRX 切换为 PTX）写指针主控权发生变化时，需要软件参与在 SPI 写 FIFO 前将写指针清‘0’。

✧ 为什么 PTX 发送完成后中断无法清‘0’？

发送中断置起后需要先将 PTX_FIFOn_OCPY 控制信号清‘0’再清中断，若只清中断 INTn，FIFOn 将又处于有效状态进入发送流程，发送完成后中断又将置起。

✧ 为什么同步字长度设为 16bits 很容易误相关？

同步字长度设为 16bits 时，长度较短出现误相关的概率较大，推荐同步字错误阈值 SYNC_THRES（0x28）设置为 1 或 0。

✧ 收发 ACK 使能时为什么 PTX 发送中断正常置起而 PRX 接收无正常中断？

ACK 使能时，PTX 发送方中断置起分两种情况：

1. PTX 正常收到 ACK 信号，通讯成功。
2. PTX 重传超时，通讯不成功。

PTX 中断置起后可以通过 PTX_FIFOn_FAIL 标志位区分两种情况。

ACK 使能时，PRX 在接收 PID 与 CRC 较上一次相同时将自动弃包而不置起中断。若 PTX 出现重传超时，PTX 在下一次发送帧时 PID 不累加。

✧ ACK PAYLOAD 功能使能时为什么 PTX 的 ACK_INTn 中断标志正常置起但读取 ACKFIFO 的值不正确或一直保留上一次收到的值？

PTX 在收到零长度的 ACK PAYLOAD（即只有 ACK 信号）时不对 ACKFIFO 进行写操作，但标志位 ACKINTn_W_ACKPAY 将会置起，上层软件在读 0x3D 寄存器时，可依据此标志位确认此次收发流程有无 ACK PAYLOAD。

✧ 为什么收发双方 PIPE Address 配置一致时，会出现 PRX 的 PIPE 指示位（PRX_FIFOn_PIPE）所指示的 PIPE 与发送使用的 PIPE 不一致的现象？

各 PIPE Address（0x40~0x47）设置值之间的码间距需大于接收同步字允许错误个数阈值 SYNC_THRES，否则接收各 PIPE 容易出现误同步。

✧ 为什么 PRX 接收 CRC 正确但接收 FIFO 的读取值与 PTX 发送 FIFO 内填写值不一致？

1. PTX 在写 FIFO 时有可能 SPI 受到干扰而误写，按写入 FIFO 的值硬件自动生成 CRC，PRX 收到误写的值将出现上述现象。
2. PRX 在读 FIFO 时 SPI 受到干扰而可能造成误读。

✧ ACK PAYLOAD 功能使能时为什么 PRX 的 INTn 中断标志正常但无 ACK_INTn 中断标志？

1. PRX 在返回 ACK PAYLOAD 之后，只有再收到新的帧时（PID 变化），确认 PTX 成功收到上一帧的 ACK PAYLOAD，才会置起中断标志位 ACK_INTn。
2. PRX 可能无可匹配的 ACKFIFO 只向 PTX 返回了 ACK。
(详见 HW2000 芯片数据手册的“ACK 带 ACK PAYLOAD”章节)。

✧ **CE、PD_CTRL、SFT_RST 区别？**

1. 正常工作时，请将 CE 引脚拉高，使能 HW2000 芯片。若拉低 CE 引脚将全局复位 HW2000 芯片，包括复位芯片内部各状态信号与寄存器。
2. PD_CTRL(0x23[15])为芯片进入 POWER DOWN 模式使能信号，仅控制芯片进入掉电模式，寄存器状态保持并可读写。
3. SFT_RST(0x23[14])为软件复位使能信号，仅复位芯片内部各状态信号，并不复位内部寄存器。也即 INT 中断信号等状态变量将会被复位，同时 SFT_RST 使能后会将写保护关闭（0x4C 寄存器的值变为 0xFFFF），导致功率寄存器（0x12~0x15）等未开放寄存器不能修改。

✧ **接收/发射模式下，无法更换频点？**

若需要动态更换频点，请关闭接收或发射功能，再进行频点更换。

✧ **发射使能后填写 FIFO，写入值出现错误？**

1. 发射端 TX 使能(DBUS_TX='1')后，需延时 5us 等待芯片从 SLEEP 状态切换为 IDLE 状态后才能写 FIFO，否则填入 FIFO 的数据可能会出错。
2. SPI 速率为 1MHz 或小于 1MHz 的情况下，可以不需要以上等待延时操作，在 SPI 速率大于 1M 的情况下，必须有延时操作。
3. 在程序代码中，TX 使能后，将 FIFO 指针清零后再填 FIFO，相当于在 TX 使能后加了一定的延时时间，可省去延时操作。

✧ **内部寄存器 0x06 读写不一致问题**

内部寄存器 0x06 只有高 4 位是可写位，其余为内部 ADC 采样只读寄存器，其值会动态变化。

✧ **接收端清中断操作**

在连续接收情况下（接收使能一直打开），接收端在接收完数据后，判断 CRC 校验是否成功，但是不管校验是否成功，都必须清接收中断，否则接收将暂停。

✧ **ACK 使能时，不支持 FEC 功能**

FEC 功能只在硬件链路控制和 ACK 不使能的情况下有效

✧ **ACK PAYLOAD 功能使能情况下，PREAMBLE 长度最少为 6 bytes**

在 ACK PAYLOAD 功能使能时，PREAMBLE 长度最少设置为 6 bytes，否则 PRX 会出现丢包或错包现象，PTX 出现重发超时或 ACK 不带 PAYLOAD 现象。

✧ 收发未使能情形下， **IRQ** 中断标志位无法清‘0’

在完成收发流程中断置起后，若关闭发送/接收使能，芯片将进入 **SLEEP** 模式，在此模式下无法进行中断标志清‘0’操作。可使能发送/接收后清‘0’或使用 **SFT_RST** 软复位清‘0’。