RDA5830 编程指南 v2.0

I2C 接口

将 MODE 引脚接低电平，即进入 I2C 接口模式。

RDA5830 的 I2C 接口与 I2C-Bus Specification 2.1 兼容，包含 2 个信号：SCLK 和 SDIO。

I2C 接口是由 START，命令字节，数据字节，及每个字节后的 ACK 或 NACK 比特，和 STOP 组成。命令字节包括一个 7 比特的 chip 地址（0010001b）和一个读写 r/$\overline{w}$ 命令比特。ACK（或 NACK）由接收器发出。

RDA5830 的 I2C 接口格式如下：

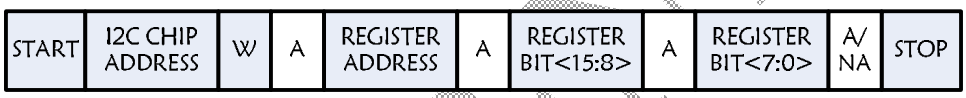
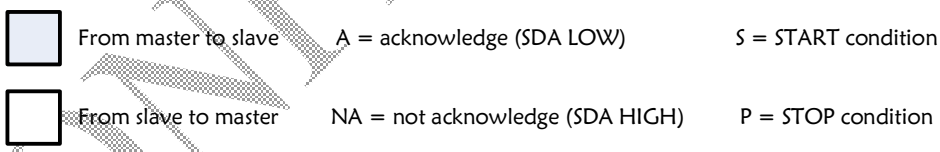


图 10 复合格式 i2c 写格式

图 11 复合格式 i2c 读格式



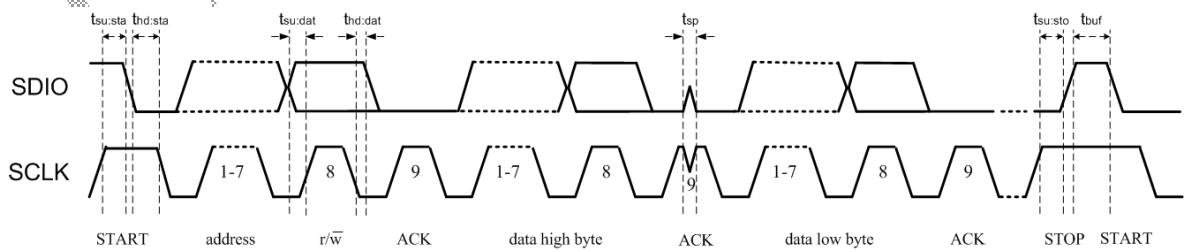


图 1 I2C 接口写数据时序

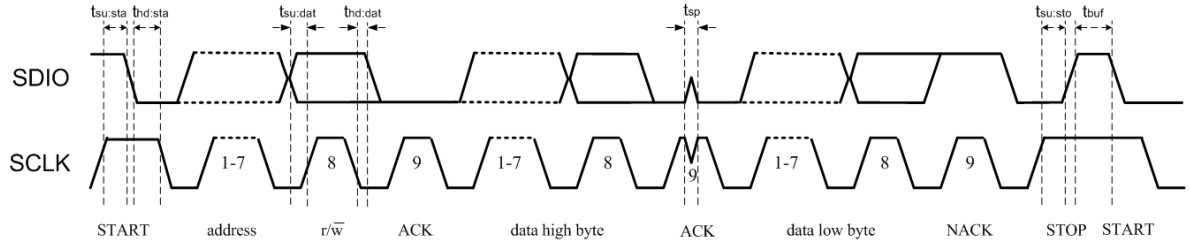


图 2 I2C 接口读数据时序

I2C Timing Characteristics

Parameter	Symbol	Test Condition	Min	Typ	Max	Unit
SCLK Frequency	fscl		0	–	400	KHz
SCLK High Time	thigh		0.6	–	–	us
SCLK Low Time	tlow		1.3	–	–	us
Setup Time for START Condition	tsu:sta		0.6	–	–	us
Hold Time for START Condition	thd:sta		0.6	–	–	us
Setup Time for STOP condition	tsu:sto		0.6	–	–	us
SDIO Input to SCLK ↑ Setup	tsu:dat		100	–	–	ns
SDIO Input to SCLK ↓ Hold	thd:dat		0	–	900	ns
STOP to START Time	tbuf		1.3	–	–	us
SDIO Output Fall Time	tf:out		20+0.1Cb	–	250	ns
SDIO Input, SCLK Rise/Fall Time	tr:in tf:in		20+0.1Cb	–	300	ns
Input Spike Suppression	tsp		–	–	50	ns
SCLK, SDIO Capacitive Loading	Cb		–	–	50	pF

状态转换

RDA5830 有 5 种状态：复位初始化（Reset&Initial），设置频点（Tune），搜台（Seek），工作（Working），休眠（Sleep）。

在芯片上电和复位后，软件通过编写 ENABLE（02H，bit 0）寄存器，将其置为 1，即可使 RDA5830 进入上电状态。软件通过编程相应寄存器，即可使 RDA5830 进入 Tune（包括

FmTune 和 AmTune）或 Seek（包括 FmSeek 和 AmSeek）状态，这些操作之后，RDA5830 进入 Working 状态（包括 Fm 和 Am）。软件通过将 ENABLE 置为 0，可使 RDA5830 进入睡眠状态，此时所有寄存器值保持不变（与未睡眠之前相同）。在睡眠状态时，软件可通过编写 ENABLE 为 1，即可将 RDA5830 回到正常工作（Working）状态。

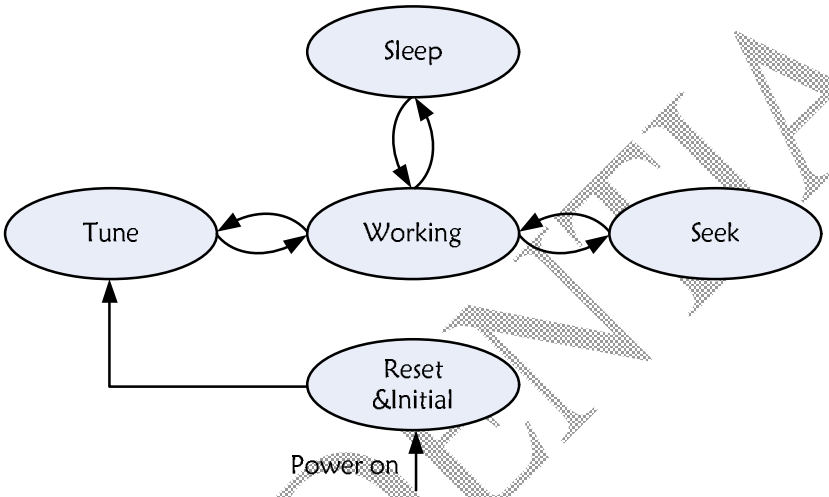


图 3 RDA5830 状态转移图

复位初始化（Reset&Initial）

上电过程中，RDA5830 需要正确的 Reset 和初始化过程来进行上电。

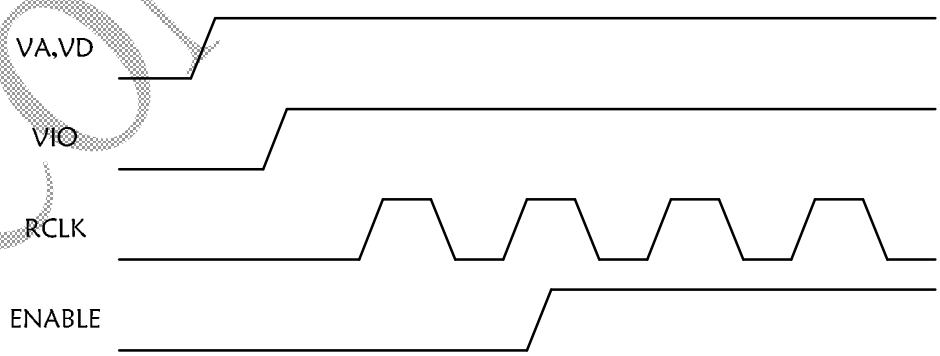


图 4 Reset&Initial 时序图

编程伪程序：

Supply VA and VD.

Supply VIO

Provide 32.768KHz crystal clock. (optional, if use TCXO)

(or 12MHz/24MHz/13MHz/26MHz/19.2MHz/38.4MHz clk)

Wait 1ms

Mov 0XC003, 02H //write enable=1

Wait 0.5s //optional, for wait RCLK stable if use DCXO

//通过 I2C 或 3 线接口进行以下写操作（初始化芯片内部寄存器，如必要，则由 RDA 提供）

Mov 0XC001, 02H
Mov 0x00F1, 14H
Mov 0x10A0, 15H
Mov 0x0542, 1AH
Mov 0x7F80, 21H
Mov 0x50A4, 23H
Mov 0x47C0, 75H
Mov 0xd893, 79H

休眠（Sleep）

在空闲时，软件可以通过编程 ENABLE（置 0）使 RDA5830 进入睡眠模式，以便减小功耗。

在睡眠模式，RDA5830 模拟和数字模块电源都被关掉，但各寄存器值保持不变，SPI 和 I2C 接口依然可以工作。

软件可以通过编程 ENABLE（置 1）使 RDA5830 进入工作模式。进入工作模式后，软件需要重新设置所需要的频点，即重新进行一次 Tune 操作。

编程伪程序：

Enter Sleep Mode：

Mov 0XC003, 02H //clear ENABLE bit low to bring RDA5830 into sleep mode

Exit Sleep Mode：

Mov 0XC001, 02h //set ENABLE bit high to bring RDA5830 into working mode

```
Wait 0.5s              //optional, wait RCLK stable, if in DCXO mode
Mov 0x0150, 03h        //Set channel number to 87.5MHz, space to 100KHz, band to 87~108MHz
*Wait for GPIO2=0      //optional, wait for tune complete, if use interrupt
*Wait for STC=1        //optional, wait for tune complete, if use polling method
Read 0A, 0BH           //read stauts
Stop Tune
```

FM 接收

设置 40H 寄存器的 CHIP_FUNC[3:0]=0 即可定义当前工作模式为 FM 接收模式。

设置频点（FmTune）

软件可以通过配置 03H 寄存器来选择 FM 频道。搜台(Seek)的步进长度(100KHz, 200KHz, 50KHz 和 25KHz）由 SPACE[1:0]来选择，频道由 CHAN[9:0]来选择，频率范围（76MHz~91MHz, 87MHz~108MHz, 76MHz~108MHz 或用户自定义 65MHz~115MHz 范围内频段）由 BAND[1:0]来选择。自定义的频段由寄存器 53H（chan_bottom）和 54H（chan_top）来设置，单位为 100KHz，即定义 65MHz~76MHz，可设置 BAND[1:0]=3，并且设置 chan_bottom=0x028A，chan_top=0x02f8。

当软件写 03H 寄存器的 TUNE 位为 1 时，RDA5830 会自动开始 Tune。在 Tune 结束时(如果 STCIEN 设为 1，会产生一个中断信号 INT 由 GPIO2 送出），STC 会被置 1，真假台判断结束后 FM_READY 会被置 1，软件可以通过读 0AH 和 0BH 寄存器来得到当前频点的状态值（ST, FM_TRUE, FM_READY, RSSI, READCHAN 等）。接收时整个 Tune 过程要持续 10ms，如需判断真台与否则需要 20ms。

频点计算方法见寄存器 CHAN 和 READCHAN 的换算公式。

编程伪程序：

```
Mov 0x0000, 40H      //set FM mode
Mov 0x1A10, 03H      //Set channel number to 97.4MHz, space to 100KHz, band to 87~108MHz
*Wait for GPIO2=0    //optional, wait for tune complete, if use interrupt
*Wait for STC=1      //optional, wait for tune complete, if use polling method
Read 0A, 0BH         //read stauts
Stop Tune
```

搜台（FMSeek）

1、半自动搜台

软件可以通过设定 02H 寄存器的 SEEK 和 SEEKUP 来设置 RDA5830 进行向上（向下）搜台。同样，SEEKTH[6:0]（seek 门限，对应信噪比）也可通过写 05H 寄存器来设定。RDA5830 会跳到下一个（向上或下由 SEEKUP 确定）频道来判断其是否是真台，步进由 SPACE 确定。在 Seek 时，如果 SKMODE 设为 0，在 Seek 时，当 RDA5830 内部触到所选频段的边界时，会自动从另一边界绕回，继续搜台。当 RDA5830 找到一个台（RDA5830 会工作在当前所在频道上，STC 会被置 1，SF 会被置 0，FM_READY 和 FM_TRUE 都会被置 1），或者在整个频道都没有找到台，Seek 操作会停止（RDA5830 会工作在 Seek 操作前所在频道上，STC 会被置 1，SF 会被置 1，FM_READY 会被置 1，而 FM_TRUE 则为 0）；如果 SKMODE 设为 1，在 Seek 时，当 RDA5830 内部触到所选频段的边界时会停止 Seek 并停留在边界处（STC 会被置 1，SF 会被置 1，FM_READY 会被置 1，而 FM_TRUE 则为 0）。Seek 结束后，软件可以通过读取 0AH 和 0BH 寄存器来得到当前频道号，RSSI 值和其他一些状态信息。RDA5830 内部 Seek 操作是由一连串的 Tune 操作组成，每个频点的 Tune 和搜台判断需要

20ms，所以 Seek 操作的时间取决于被搜频点号的数量。在搜台过程中，写 02H 寄存器的 SEEK 位为 0，则 RDA5830 会停止搜台，并停留在当前搜索的频点上，同时 STC 会被置 1。RDA5830 内部半自动搜台模式编程流程如下图所示。（此模式兼容 5830/3/4 搜台软件，新的软件可根据需要采用软件搜台或者全自动搜台模式）

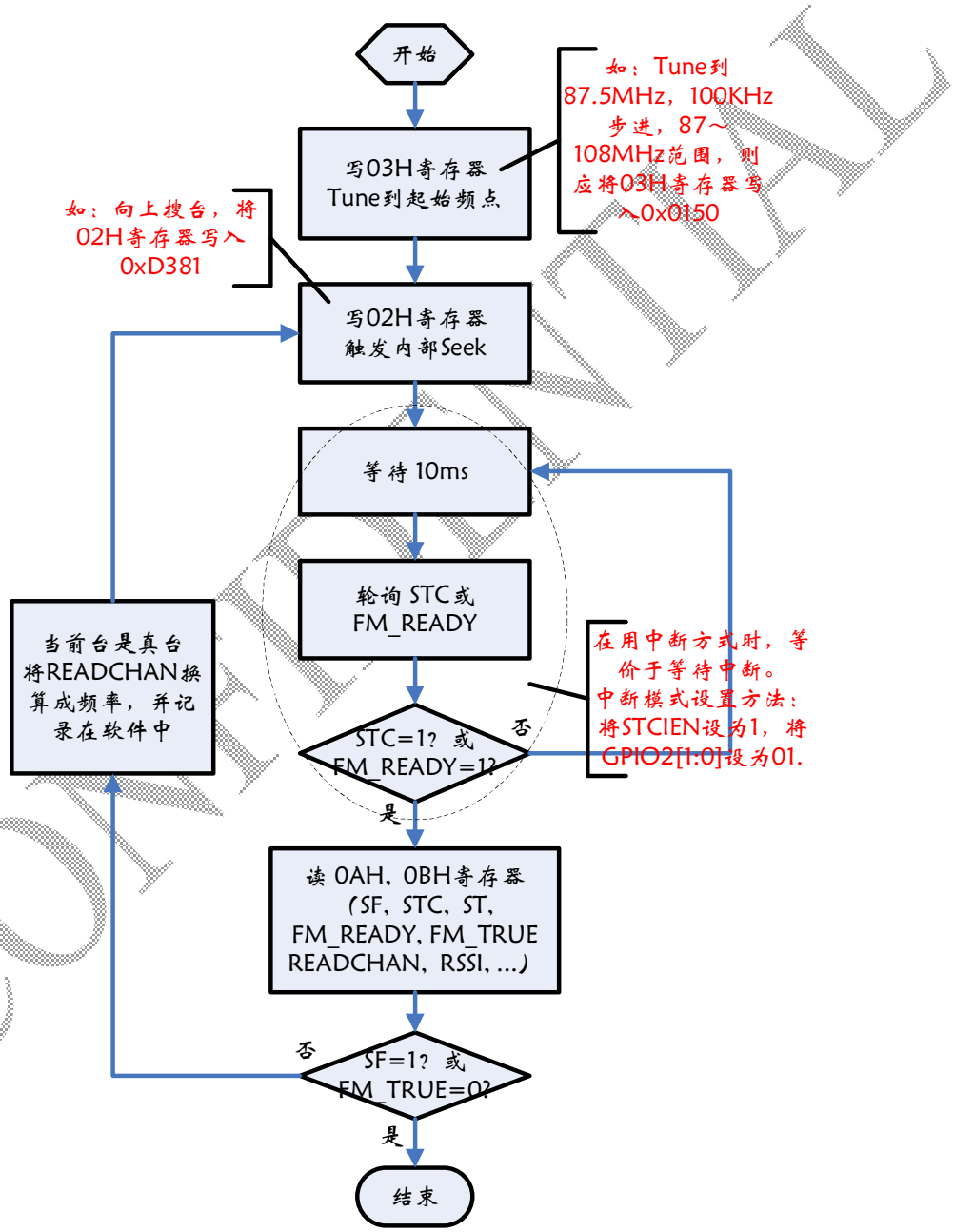


图 5 RDA5830 硬件搜台模式编程流程图

编程伪程序：

Step1：

Mov 0x0000, 40H //set Rx mode

Mov 0x0150, 03H //Set channel number to 87.5MHz, space to 100KHz, band to 87~108MHz

*Wait for GPIO2=0 //optional, wait for tune complete, if use interrupt

*Wait for STC=1 //optional, wait for tune complete, if use polling method

Read 0A, 0BH //read stauts

Go to step2

Step2：

Mov 0xD381, 02H // set SEEK and SEEKUP for seek operation

*Wait for GPIO2=0 //optional, wait for seek complete, if use interrupt

*Wait for STC=1 or FM_READY=1

//optional, wait for seek complete, if use polling method

Read 0A, 0BH //read stauts

If SF=1 or FM_TRUE=0, go to step3;

else memorize READCHAN and go to step2.

Step3：

Stop Seek

2、软件手动搜台

RDA5830 软件搜台模式编程流程如下图所示。（这种搜台方式，软件控制起来比较灵活，并且适用于带搜台频点显示功能的软件）

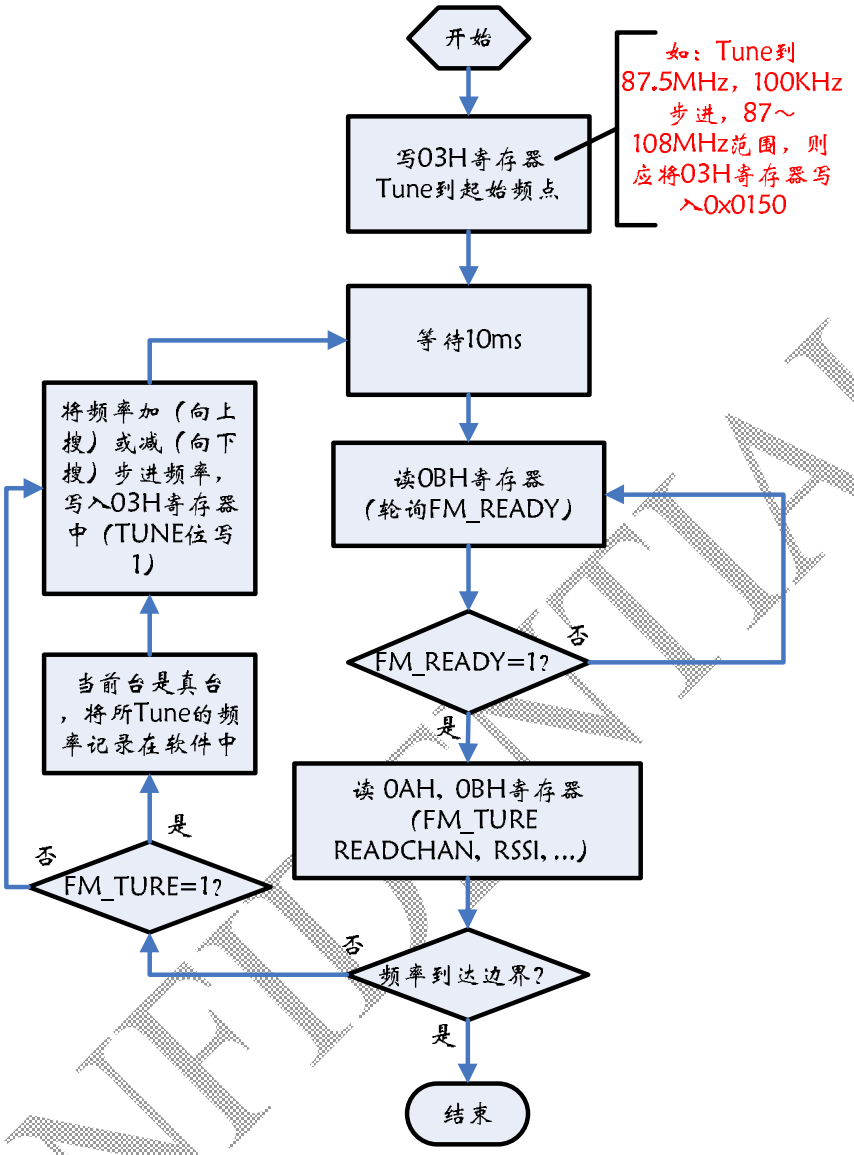


图 6 RDA5830 软件搜台模式编程流程图

编程伪程序：

Step1：

```
Mov 0x0000, 40H       //set Rx mode
CHAN=0x0005;
VALUE=(CHAN<<6)+0x0010;
Mov VALUE, 03H        //Set channel number to 87.5MHz, space to 100KHz, band to 87~108MHz
Go to step2
```

Step2：

```
Wait for FM_READY=1
```

Read 0A, 0BH //read stauts

If freq beyond band limit, go to Step4. Else go to Step3.

Step3：

If FM_TRUE=1, memorize READCHAN.

CHAN=CHAN+1;

VALUE=(CHAN<<6)+0x0010;

Mov VALUE, 03H

Go to Step2.

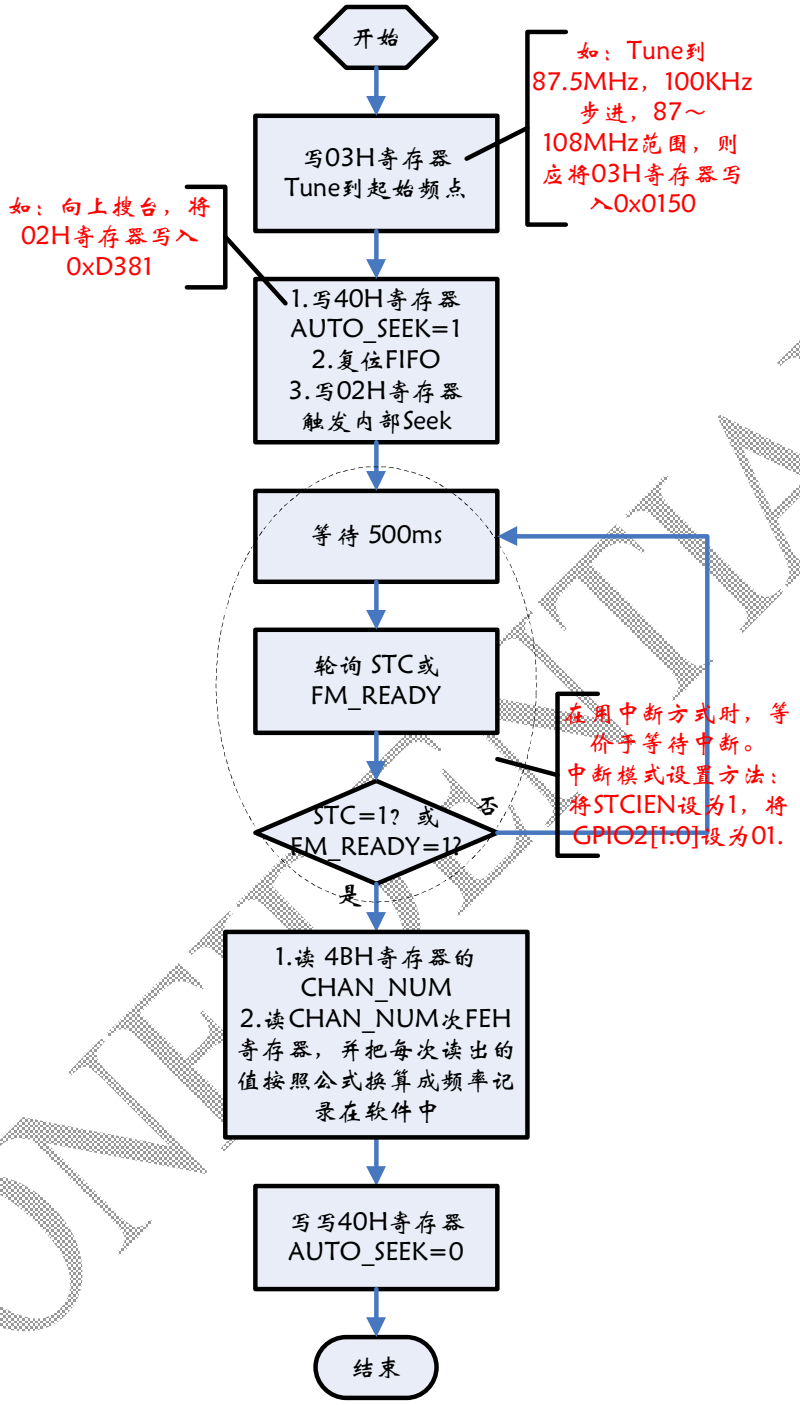
Step4：

Stop Seek.

3、全自动搜台

RDA5830 提供了全自动搜台模式，即软件只需设置搜台频段（BAND[0:1]），搜台步进（SPACE[1:0]），搜台方向（SEEKUP），搜台起始频点（CHANNEL[9:0]），搜台停止模式（SKMODE，到边停止或到边绕回），并且设置 40H 寄存器的 AUTO_SEEK=1，同时触发 02H 的 SEEK，芯片内部就会在频段内搜台，并把搜到的所有台号，存在 RDA5830 的 FIFO（读 FIFO 的入口地址为 FEH）里面。搜台结束后，STC 会被置 1，所搜到的台的数目会写在 4BH 寄存器的 CHAN_NUM[7:0]里。此时软件可以通过 i2c 或 3 线接口，从 FIFO 里把相应的电台号取出（为保证 FIFO 里面的指针不影响搜台结果，全自动搜台前应先将 FIFO 里的指针复位，方法即先写 41H 的 MEM_CLR=1，再写 MEM_CLR=0）。在搜台过程中，写 02H 寄存器的 SEEK 位为 0，则 RDA5830 会停止搜台，并停留在当前搜索的频点上，同时 STC 会被置 1。

RDA5830 全自动搜台模式编程流程如下图所示。（这种方法优点是搜台快速，减少了主控和 RDA5830 的交互）



编程伪程序：

Step1：

```
Mov 0x0000, 40H        //set Rx mode
Mov 0x0150, 03H        //Set channel number to 87.5MHz, space to 100KHz, band to 87~108MHz
*Wait for GPIO2=0      //optional, wait for tune complete, if use interrupt
```

*Wait for STC=1 //optional, wait for tune complete, if use polling method

Read 0A, 0BH //read stauts

Go to step2

Step2：

Mov 0x8000, 40H //set Rx mode & AUTO SEEK MODE=1

Mov 0x1300, 41H //reset FIFO address pointer

Mov 0x0300, 41H //reset FIFO address pointer

Mov 0xD381, 02H // set SEEK and SEEKUP for seek operation

*Wait for GPIO2=0 //optional, wait for seek complete, if use interrupt

*Wait for STC=1 or FM_READY=1

//optional, wait for seek complete, if use polling method

Read 4BH CHAN_NUM //read CHAN_NUM

For (i=1; i<=CHAN_NUM; i=i+1) Read FEH, and memorize the value just has been read

go to step3.

Step3：

Mov 0x0000, 40H //set Rx mode & AUTO SEEK MODE=0

Stop Seek

内部小天线模式（Antenna）

RDA5830 支持内部小天线，在接收状态时设置 02H 寄存器的 ANTENNA=1，然后进行 Tune 或 Seek（在使用内部小天线进行搜台时，搜台速度较慢）操作即可。

不采用内部小天线，则设置 ANTENNA=0 即可。

RBDS/RDS 接收

RDA5830 中设计了 RBDS/RDS 处理器，该处理器包括 BLOCK 的同步，误码检测及纠错功能。软件可以通过将寄存器 02H 中的 RDS_EN 置 1 来开启 RBDS/RDS 接收功能。

RDA5830 支持寄存器读取和 FIFO 读取模式。流程如下：

1、寄存器读取模式（兼容 5802/3）

设置 64H 寄存器的 RDS_RXMOD[1:0]=3，即为寄存器读取模式。接收到的 RDS/RBDS 数据会通过寄存器 0AH~10H 送出。

当寄存器 0AH 中的 RDSS（RBDS/RDS 同步信息）为 1 时，表明 RBDS/RDS 的 GROUP 已经同步，为 0 时，则表示同步丢失。当同步完成时，每当一个 GROUP 数据被接收后，RDSR 位会置 1。RDA5830 有 2 种 RDSR 模式，当 RDSR_MODE 设为 0 时，在每一个 GROUP 数据被接收后，RDSR 会置 1，并且会保持至少 40ms，之后会被自动置 0；当 RDSR_MODE 设为 1 时，在每一个 GROUP 数据被接收后，RDSR 会置 1，并且会一直保持，直到软件对寄存器 0CH 进行读操作，RDSR 才会被置 0。当每一个 GROUP 数据被接收时，同样可以送出中断信号，设置 RDSIEN 为 1，并且 GPIO2[1:0]为 01 即可在 GPIO2 引脚送出 RDSR 的中断信号。RDA5830 同样会通过寄存器 BLERA, BLERB, BLERC, BLERD 送出每个 BLOCK 的误码信息。其含义间寄存器说明。

RBDS/RDS 模式可以通过配置寄存器 04H 的 RBDS 来选择，RBDS 为置 1 则进入 RBDS 模式，置 0 则进入 RDS 模式。在 RBDS 接收模式下，接收到的 BLOCK 除了 RDS 的 BLOCK（A,B,C/c',D）外，还会有 BLOCK E（4 个 BLOCK E 组成 1 个 GROUP）。软件可以通过读寄存器 ABCD_E 来区分 BLOCK ID 信息。当 ABCD_E 为 0 时，则寄存器 0CH,0DH,0EH 和 0FH 当前的数据分别对应 A,B,C/c',D4 个 BLOCK。当 ABCD_E 为 1 时，则寄存器 0CH,0DH,0EH 和 0FH 当前的数据都为 BLOCK E。

软件接收 RDS 数据流程如下图。

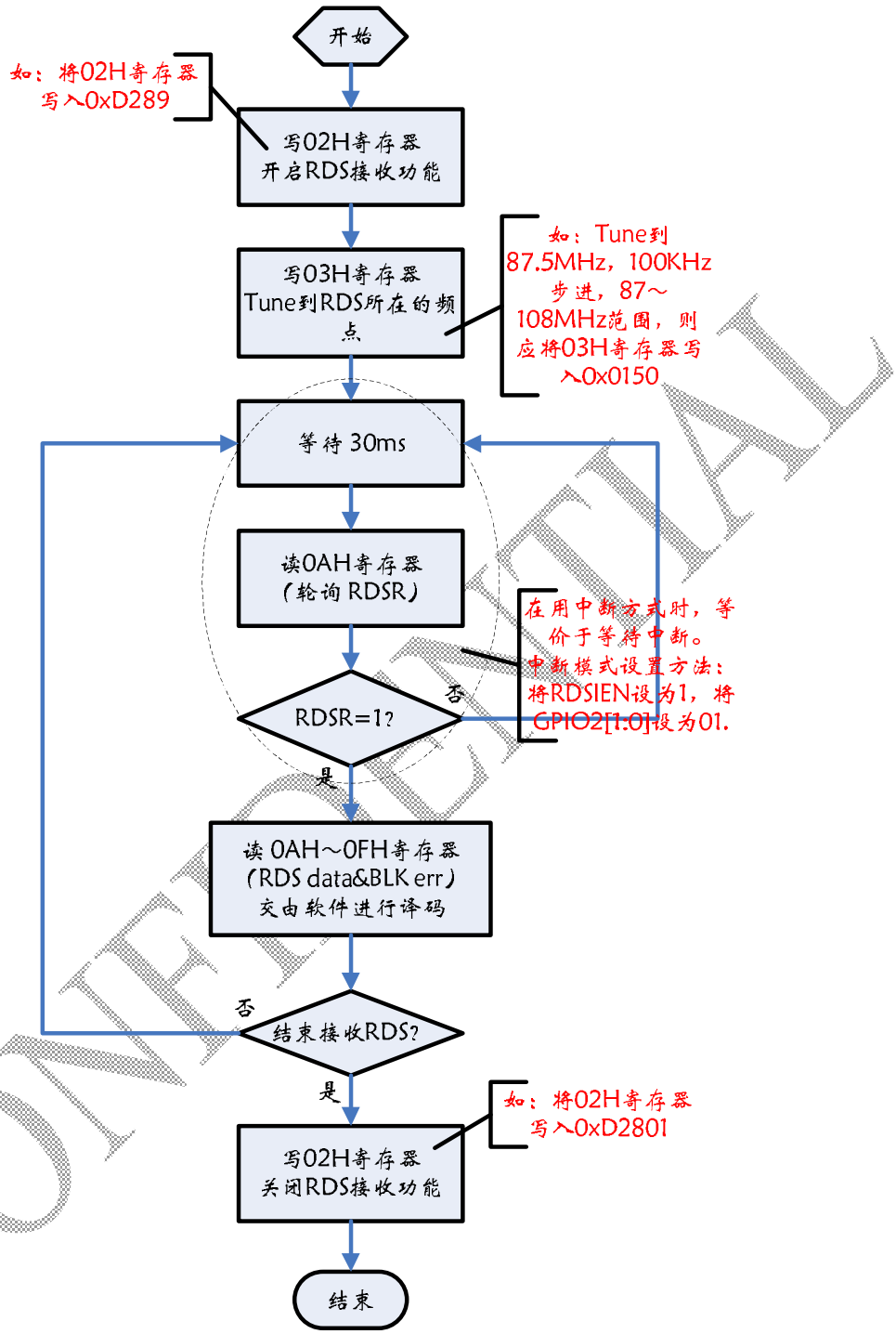


图 7　软件接收 RDS 数据流程图

编程伪程序：

Step1：

```
Mov 0x0000, 40H      //set Rx mode
Mov 0x0018, 64H      //set RDS_RXMOD[1:0]=3
```

Mov 0xD289, 02H //set RBDS/RDS enable.

Mov 0x0150, 03h //Set channel number to 87.5MHz, space to 100KHz, band to 87~108MHz

Step2：

*Wait for GPIO2=0 //optional, wait for RBDS/RDS group ready, if use interrupt

*Wait for RDSR=1 //optional, wait for RBDS/RDS group ready, if use polling method

Read 0A~0FH //read RDS data

Software decode RDS information

If end RDS receive, go to Step3; else go to Step 2.

Step3：

Mov 0xD281, 02H //clear RBDS/RDS enable.

Stop RDS receive

2、FIFO 读取模式

RDA5830 内部有个 FIFO，供存储 RDS 等信息用。设置 64H 寄存器的 RDS_RXMOD[1:0]=2，即为 FIFO 读取模式。当接收 RBDS/RDS 同步后，RBDS/RDS 的信息会以 GROUP 为单位依次写入 FIFO 中。

写入 FIFO 中的 RBDS/RDS 信息依次为：{第 n-1 组：……，BLOCK C（或 c'/E），BLOCK D（或 E），BLOCK INFO（包括各 block 的误码信息，和 block E 的指示信息），第 n 组：BLOCK A（或 E），BLOCK B（或 E），BLOCK C（或 c'/E），BLOCK D（或 E），BLOCK INFO（包括各 block 的误码信息，和 block E 的指示信息），第 n+1 组：BLOCK A（或 E），BLOCK B（或 E），……}。

由于 RBDS/RDS 存入 FIFO 是以 5 个 16bit 为单位，所以此时设置 FIFO 的长度（41H 寄存器 MEM_DEPTH[8:0]）必须为 5 的倍数。

当 FIFO 中存储的数据达到中断要求时，同样可以送出中断信号，设置相应的 MEM_INTMOD[4:0]（4AH 寄存器），并且 GPIO2[1:0]为 01 即可在 GPIO2 引脚送出 FIFO

的中断信号。

设置 FIFO 长度为 5*n（MEM_DEPTH[8:0]=5n），FIFO 中断源为 FULL 模式（MEM_INTMOD[4:0]=16）时，当接收到的 RBDS/RDS 信息为 n 个 GROUP 时，会产生中断，软件需要通过读 5*n 次 FEH 寄存器从 FIFO 里读取数据（从中断产生到读取结束，要控制在 80ms 以内）。

编程伪程序：

Step1：

```
Mov 0x0000, 40H      //set Rx mode
Mov 0x0010, 64H      //set RDS_RXMOD[1:0]=2
Mov 0x8300, 41H      //reset FIFO address pointer
Mov 0x02FF, 41H      //set MEM_DEPTH[8:0] = 255
Mov 0x0010, 4AH      //set BUF_INT_MOD[4:0] = 16
Mov 0x0640, 04H      //set GPIO2[1:0]=1
Mov 0xD289, 02H      //set RBDS/RDS enable.
Mov 0x0150, 03h      //Set channel number to 87.5MHz, space to 100KHz, band to 87~108MHz
```

Step2：

```
Wait for GPIO2=0     //optional, wait for FIFO full interrupt, if use interrupt mode
Read 0A~0CH          //read RDS information, and clear interrupt
For (i=0;i<255;i=i+1) read FEH, store RDS information just has been read
Software decode RDS information
If end RDS receive, go to Step3; else go to Step 2.
```

Step3：

```
Mov 0xD281, 02H      //clear RBDS/RDS enable.
Stop RDS receive
```

AM/LW/SW 接收

设置 40H 寄存器的 CHIP_FUNC[3:0]=2 可定义当前工作模式为 AM/LW 接收模式，设置 CHIP_FUNC[3:0]=3，则定义当前工作模式为 SW 接收模式。

设置频点（AmTune）

软件可以通过配置 72H 寄存器的 AM_FREQ[15:0]（单位 KHz）来选择 AM/LW/SW 频率。当软件写 72H 寄存器时，RDA5830 会自动开始 Tune。在 Tune 结束时（如果 STCIEN 设为 1，会产生一个中断信号 INT 由 GPIO2 送出），STC 会被置 1，真假台判断结束后 AM_READY 会被置 1，软件可以通过读 0AH，0BH 和 7AH 寄存器来得到当前频点的状态值（AM_TRUE, AM_READY, RSSI, READAMFREQ 等）。发射时整个 Tune 过程要持续 100ms。

编程伪程序：

```
Mov 0x0002, 40H       //set AM/LW mode
Mov 0x01F4, 72H       //Set channel number to 500KHz
*Wait for GPIO2=0     //optional, wait for tune complete, if use interrupt
*Wait for STC=1       //optional, wait for tune complete, if use polling method
Read 0AH, 0BH, 7A H   //read status
Stop Tune
```

搜台（AMSeek）

1、半自动搜台

软件可以通过设定 02H 寄存器的 SEEK 和 SEEKUP 来设置 RDA5830 进行向上（向下）搜台，搜索的范围可通过 73H 和 74H 寄存器的 AM_BOTTOM（单位 KHz）和 AM_TOP（单

位 KHz）进行设置，搜索的步进频率可通过 71H 寄存器的 AM_SPACE（单位 KHz）进行设置（如设置 AM_BOTTOM=0x01F4，AM_TOP=0x06A4，AM_SPACE=0x000a，则搜索范围为 500KHz~1.7MHz，步进频率为 10KHz）。同样，SEEKTH[6:0]（seek 门限，对应信噪比）也可通过写 05H 寄存器来设定。RDA5830 会跳到下一个（向上或下由 SEEKUP 确定）频道来判断其是否是真台，步进由 AM_SPACE 确定。在 Seek 时，当 RDA5830 找到一个台（RDA5830 会工作在当前所在频道上，STC 会被置 1，SF 会被置 0，AM_READY 和 AM_TRUE 都会被置 1），当 RDA5830 内部触到所选频段的边界时会停止 Seek 并停留在边界处（STC 会被置 1，SF 会被置 1，AM_READY 会被置 1，而 AM_TRUE 则为 0）。Seek 结束后，软件可以通过读取 0AH，0BH 和 7AH 寄存器来得到当前频率，RSSI 值和其他一些状态信息。RDA5830 内部 Seek 操作是由一连串的 Tune 操作组成，每个频点的 Tune 和搜台判断需要 20ms，所以 Seek 操作的时间取决于被搜频点号的数量。在搜台过程中，写 02H 寄存器的 SEEK 位为 0，则 RDA5830 会停止搜台，并停留在当前搜索的频点上，同时 STC 会被置 1。

RDA5830 内部半自动搜台模式编程流程如下图所示。

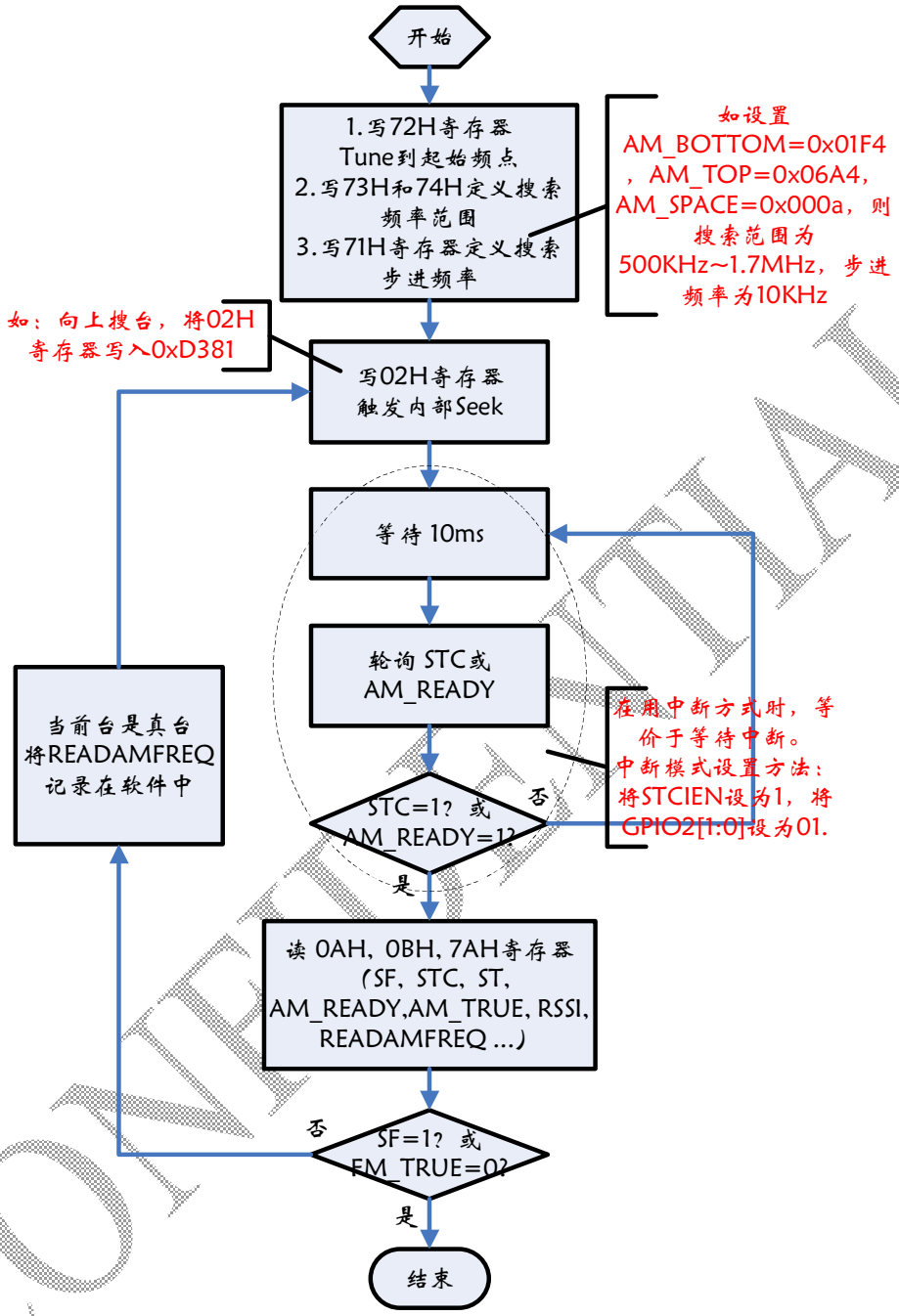


图 8 RDA5830 硬件搜台模式编程流程图

编程伪程序：

Step1：

```
Mov 0x0002, 40H       //set AM/LW mode
Mov 0x01F4, 72H       //Set AM frequency to 500KHz
```

*Wait for GPIO2=0 //optional, wait for tune complete, if use interrupt

*Wait for STC=1 //optional, wait for tune complete, if use polling method

Read 0A, 0BH //read stauts

Go to step2

Step2：

Mov 0x000a, 71H //set AM seek step to 10KHz

Mov 0x01F4, 73H //set AM seek bottom frequency

Mov 0x06A4, 74H //set AM seek top frequency

Mov 0xD381, 02H // set SEEK and SEEKUP for seek operation

*Wait for GPIO2=0 //optional, wait for seek complete, if use interrupt

*Wait for STC=1 or AM_READY=1

//optional, wait for seek complete, if use polling method

Read 0A, 0B, 7A H //read stauts

If SF=1 or AM_TRUE=0, go to step3;

else memorize READAMFREQ and go to step2.

Step3：

Stop Seek

2、软件手动搜台

RDA5830 软件搜台模式编程流程如下图所示。（这种搜台方式，软件控制起来比较灵活，并且适用于带搜台频点显示功能的软件）

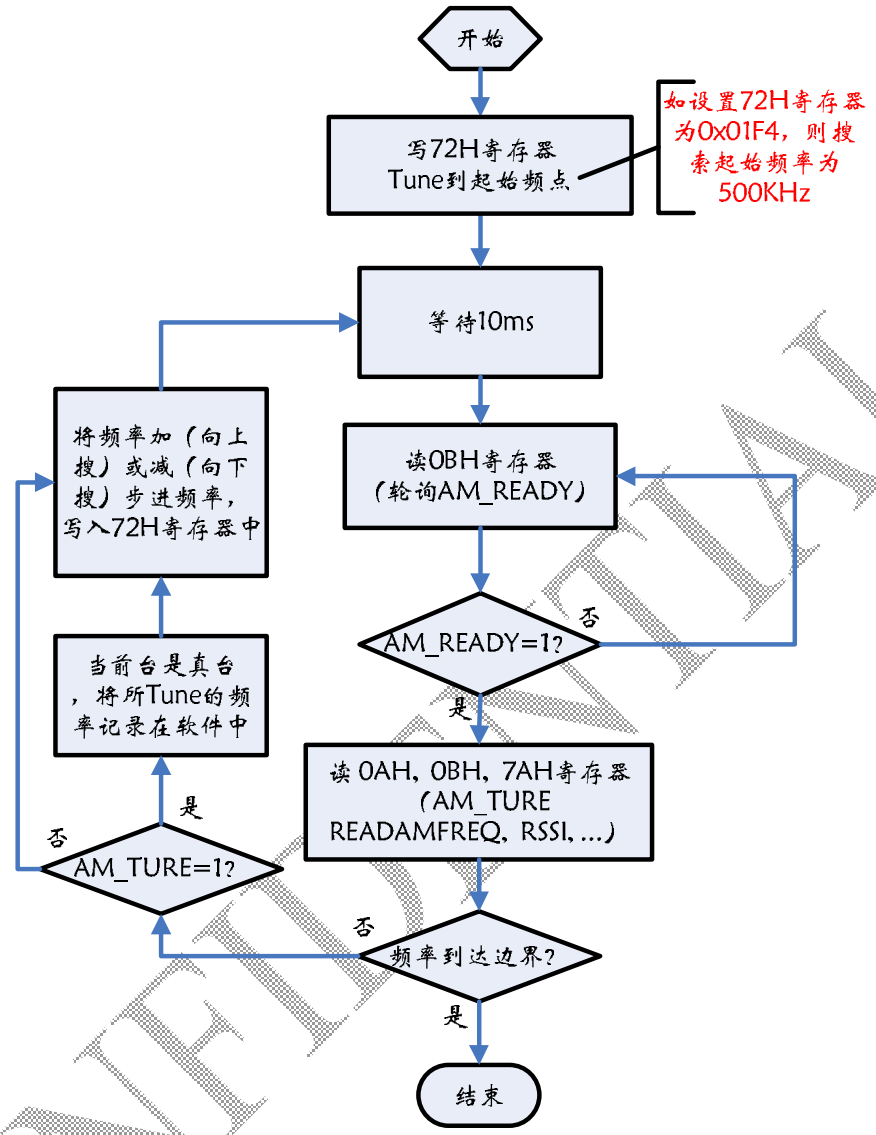


图 9 RDA5830 软件搜台模式编程流程图

编程伪程序：

Step1：

Mov 0x0002, 40H //set AM/LW mode

AM_FREQ=0x01F4

Mov AM_FREQ, 72H //Set AM frequency to 500KHz

Go to step2

Step2：

Wait for AM_READY=1

Read 0A, 0B, 7A H //read stauts

If freq beyond band limit, go to Step4. Else go to Step3.

Step3：

If AM_TRUE=1, memorize READAMFREQ.

AM_FREQ=AM_FREQ+SEEK_STEP;

Mov AM_FREQ, 72H

Go to Step2.

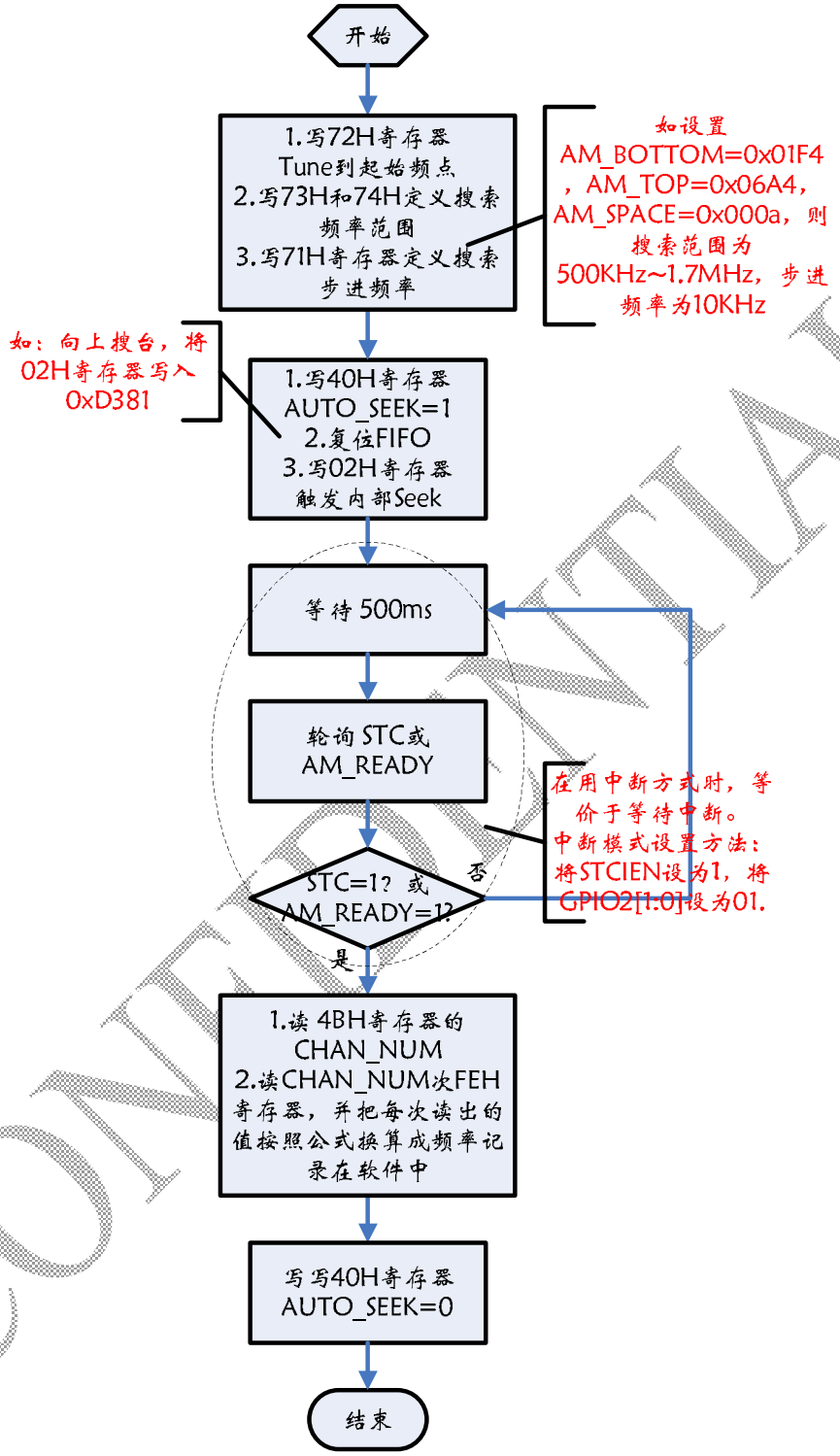
Step4：

Stop Seek.

3、全自动搜台

RDA5830 提供了全自动搜台模式，即软件只需设置搜台频段（AM_BOTTOM 和 AM_TOP），搜台步进（AM_SPACE），搜台方向（SEEKUP），搜台起始频点（AM_FREQ），并且设置 40H 寄存器的 AUTO_SEEK=1，同时触发 02H 的 SEEK，芯片内部就会在频段内搜台，并把搜到的所有频率，存在 RDA5830 的 FIFO（读 FIFO 的入口地址为 FEH）里面。搜台结束后，STC 会被置 1，所搜到的台的数目会写在 4BH 寄存器的 CHAN_NUM[7:0] 里。此时软件可以通过 i2c 或 3 线接口，从 FIFO 里把相应的电台号取出（为保证 FIFO 里面的指针不影响搜台结果，全自动搜台前应先将 FIFO 里的指针复位，方法即先写 41H 的 MEM_CLR=1，再写 MEM_CLR=0）。在搜台过程中，写 02H 寄存器的 SEEK 位为 0，则 RDA5830 会停止搜台，并停留在当前搜索的频点上，同时 STC 会被置 1。

RDA5830 全自动搜台模式编程流程如下图所示。（这种方法优点是搜台快速，减少了主控和 RDA5830 的交互）



编程伪程序：

Step1：

```
Mov 0x0002, 40H        //set AM/LW mode
```

Mov 0x01F4, 72H //Set AM frequency to 500KHz

*Wait for GPIO2=0 //optional, wait for tune complete, if use interrupt

*Wait for STC=1 //optional, wait for tune complete, if use polling method

Read 0A, 0B, 7A H //read stauts

Go to step2

Step2：

Mov 0x000a, 71H //set AM seek step to 10KHz

Mov 0x01F4, 73H //set AM seek bottom frequency

Mov 0x06A4, 74H //set AM seek top frequency

Mov 0x8002, 40H //set AM/LW mode & AUTO SEEK MODE=1

Mov 0x1300, 41H //reset FIFO address pointer

Mov 0x0300, 41H //reset FIFO address pointer

Mov 0xD381, 02H // set SEEK and SEEKUP for seek operation

*Wait for GPIO2=0 //optional, wait for seek complete, if use interrupt

*Wait for STC=1 or AM_READY=1

//optional, wait for seek complete, if use polling method

Read 4BH CHAN_NUM //read CHAN_NUM

For (i=1; i<=CHAN_NUM; i=i+1) Read FEH, and memorize the value just has been read

go to step3.

Step3：

Mov 0x0002, 40H //set AM/LW mode & AUTO SEEK MODE=0

Stop Seek

其他功能

DAC

RDA5830 可作 dac 使用，用 i2s 通过 gpio 口将音频数据写入 RDA5830，而 RDA5830 则

将接收到的音频数据通过其内部的 dac 播放出来。设置 CHIP_FUNC[3:0]=12，并且 ENABLE=1 即可。

I2S

RDA5830 的 AM/LW/SW 和 FM 接收都可以使用 i2s。支持多种采样率。在接收模式时，设置 I2S_EN=1 即可。

FIFO

RDA5830 内部的 FIFO 可以提供给主控使用，以扩展主控的存储空间。FIFO 最大可达到 256x16bit，即最大长度为 256，并且可通过寄存器 MEM_DEPTH[8:0]进行设置。FIFO 可通过 i2c 或 3 线接口访问，写的入口地址为 FDH，读的入口地址为 FEH，读写时序格式与 i2c 和 3 线接口一致。

寄存器说明

REG	BITS	NAME	FUNCTION	DEFAULT
00H	15:0	CHIPID[7:0]	Chip ID.	0x5830
02H	15	DHIZ	Audio Output High-Z Disable. *0 = High impedance; 1 = Normal operation*	0
	14	DMUTE	Mute Disable. *0 = Mute; 1 = Normal operation*	0
	13	MONO	Mono Select. *0 = Stereo; 1 = Force mono*	0
	12	BASS	Bass Boost. *0 = Disabled; 1 = Bass boost enabled*	0
	11	RESEVED	Must be 0	0
	10	? ? ? ?		
	9	SEEKUP	Seek Up. *0 = Seek down; 1 = Seek up*	0
	8	SEEK	Seek.	0

REG	BITS	NAME	FUNCTION	DEFAULT
			0 = Disable&Stop seek; 1 = Enable *Seek begins in the direction specified by* SEEKUP *and ends when a channel is found or the entire band has been searched.* *The* SEEK *bit is set low and the* STC *bit is set high when the seek operation completes.*	
	7	SKMODE	Seek Mode 0 = wrap at the upper or lower band limit and continue seeking 1 = stop seeking at the upper or lower band limit	0
	6:4	CLK_MODE[2:0]	000=32.768kHz 001=12Mhz 101=24Mhz 010=13Mhz 110=26Mhz 011=19.2Mhz 111=38.4Mhz	000
	3	RDS_EN	RDS Enable 0 = Disabled 1 = Enable	0
	2	ANTENNA	1 = inner antenna 0 = normal	0
	1	SOFT_RESET	Soft reset. If 0, not reset; If 1, reset.	0
	0	ENABLE	Power Up Enable. *0 = Disabled; 1 = Enabled*	0
03H	15:6	CHAN[7:0]	Channel Select. BAND *= 0* *Frequency =* *Channel Spacing (kHz) x* CHAN*+ 87MHz* BAND *= 1 or 2* *Frequency =* *Channel Spacing (kHz) x* CHAN *+ 76.0 MHz* CHAN *is updated after a seek operation.* BAND *= 3* *Frequency =* *Channel Spacing (kHz) x* CHAN *+* CHAN_BOTTOM *x 100KHz*	0x00

REG	BITS	NAME	FUNCTION	DEFAULT
			CHAN *is updated after a seek operation.*	
	5	RESERVED	Must be 0	0
	4	TUNE	Tune 0 = Disable 1 = Enable The tune operation begins when the TUNE bit is set high. The STC bit is set high when the tune operation completes. The tune bit is reset to low automatically when the tune operation completes.	0
	3:2	BAND[1:0]	Band Select. 00 = 87~108 MHz (US/Europe) 01 = 76~91 MHz (Japan) 10 = 76~108 MHz (Japan wide) 11 = user defined via 53H~54H	00
	1:0	SPACE[1:0]	Channel Spacing. 00 = 100 kHz 01 = 200 kHz 10 = 50kHz 11 = reserved	00
04H	15	RDSIEN	RDS Interrupt Enable 0 = Disable Interrupt(Default) 1 = Enable Interrupt Setting RDSIEN = 1 and GPIO2[1:0] = 01 will generate a low pulse int on GPIO2 when the RDSR 0Ah[15] bit is set.	0
	14	STCIEN	Seek/Tune Complete Interrupt Enable. 0 = Disable Interrupt 1 = Enable Interrupt Setting STCIEN = 1 will generate a low pulse on GPIO2 when the interrupt occurs.	0
	13	RBDS	1=rbds 0=rds	0
	12	RESERVED	Must be 0	0
	11	DE	De-emphasis. *0 = 75 μs; 1 = 50 μs*	0

REG	BITS	NAME	FUNCTION	DEFAULT
	10	RDSR _MODE	0 = RDSR width fixed 40 ms high 1 = read reg0CH clr RDSR	1
	9:7	RESERVED	Must be 000	000
	6	I2S_ENABLED	I2S bus enable If 0, disabled; If 1, enabled.	0
	5:4	GPIO3[1:0]	General Purpose I/O 3. *00 = High impedance* *01 = Mono/Stereo indicator (ST)* *10 = Low* *11 = High*	00
	3:2	GPIO2[1:0]	General Purpose I/O 2. *00 = High impedance* *01 = Interrupt (INT)* *10 = Low* *11 = High*	00
	1:0	GPIO1[1:0]	General Purpose I/O 1. *00 = High impedance* *01 = Reserved* *10 = Low* *11 = High*	00
05H	15	INT _MODE	If 0, generate 5ms interrupt; If 1, interrupt last until read reg0CH action occurs.	1
	14:8	SEEKTH[6:0]	Seek Threshold. RSSI scale is logarithmic. 0000000 = min RSSI	0001000
	7:6	LNA_PORT_SEL[1:0]	LNA input port selection bit: 00: no input 01: LNAN 10: LNAP 11: dual port input	10
	5:4	LNA_ICSEL_BIT[1:0]	Lna working current bit: 00=1.8mA 01=2.1mA 10=2.5mA 11=3.0mA	10
	3:0	VOLUME[3:0]	DAC Gain Control Bits (Volume). *0000=min; 1111=max* *Volume scale is logarithmic*	1000
0AH	15	RDSR	RDS ready 0 = No RDS/RBDS group ready 1 = New RDS/RBDS group ready	0
	14	STC	Seek/Tune Complete. 0 = Not complete 1 = Complete The seek/tune complete flag is set	0

REG	BITS	NAME	FUNCTION	DEFAULT
			when the seek or tune operation completes.	
	13	SF	Seek Fail. *0 = Seek successful; 1 = Seek failure* *The seek fail flag is set when the seek operation fails to find a channel with an* RSSI *level greater than* SEEKTH[5:0].	0
	12	RDSS	RDS Synchronization 0=RDS decoder not synchronized(default) 1 = RDS decoder synchronized Available only in RDS Verbose mode	0
	11	RESERVED		
	10	ST	Stereo Indicator. *0 = Mono; 1 = Stereo* *Stereo indication is available on* GPIO3 *by setting* GPIO1[1:0] *=01.*	1
	9:0	READCHAN[9:0]	Read Channel. BAND = 0 Frequency = Channel Spacing (kHz) x READCHAN[7:0]+ 87 MHz BAND = 1 or 2 Frequency = Channel Spacing (kHz) x READCHAN[7:0]+ 76.0 MHz BAND = 3 Frequency = Channel Spacing (kHz) x READCHAN[7:0]+ CHAN_BOTTOM x 100KHz READCHAN[9:0] is updated after a tune or seek operation.	8'h00
0BH	15:9	RSSI[6:0]	RSSI. 000000 = min 111111 = max RSSI scale is logarithmic.	0
	8	FM_TRUE/ AM_TURE	1 = the current channel is a station 0 = the current channel is not a station	0
	7	FM_READY/ AM_READY	Used for soft seek 1 = ready 0 = not ready	0
	6:5	RESERVED		
	4	ABCD_E	1 = the block id of register	0

REG	BITS	NAME	FUNCTION	DEFAULT
			0cH,0dH,0eH,0fH is E 0 = he block id of register 0cH,0dH,0eH,0fH is A,B,C,D	
	3:2	BLERA[1:0]	BLK Errors Level of RDS_DATA_0, and is always read as Errors Level of RDS BLOCK A(in RDS mode) or BLOCK E(in RBDS mode when ABCD_E flag is 1). 00 = 0 errors requiring correction 01 = 1~2 errors requiring correction 10 = 3~5 errors requiring correction 11 = 6+ errors or error in checkword, correctionnot possible Available only in RDS Verbose mode	00
	1:0	BLERB[1:0]	BLK Errors Level of RDS_DATA_1, and is always read as Errors Level of RDS BLOCK B(in RDS mode) or E(in RBDS mode when ABCD_E flag is 1). 00 = 0 errors requiring correction 01 = 1~2 errors requiring correction 10 = 3~5 errors requiring correction 11 = 6+ errors or error in checkword, correctionnot possible Available only in RDS Verbose mode	00
0CH	15:0	RDSA[15:0]	BLOCK A (in RDS mode) or BLOCK E(in RBDS mode when ABCD_E flag is 1)	0
0DH	15:0	RDSB[15:0]	BLOCK B (in RDS mode) or BLOCK E(in RBDS mode when ABCD_E flag is 1)	0
0EH	15:0	RDSC[15:0]	BLOCK C/c' (in RDS mode) or BLOCK E(in RBDS mode when ABCD_E flag is 1)	0
0FH	15:0	RDSD[15:0]	BLOCK D (in RDS mode) or BLOCK E(in RBDS mode when ABCD_E flag is 1)	0
10H	15:14	BLERC[1:0]	BLK Errors Level of RDS_DATA_2, and is always read as Errors Level of RDS BLOCK C/c' (in RDS mode) or BLOCK E(in RBDS mdoe when	00

REG	BITS	NAME	FUNCTION	DEFAULT
			ABCD_E flag is 1). 00 = 0 errors requiring correction 01 = 1~2 errors requiring correction 10 = 3~5 errors requiring correction 11 = 6+ errors or error in checkword, correctionnot possible Available only in RDS Verbose mode	
	13:12	BLERD[1:0]	BLK Errors Level of RDS_DATA_3, and is always read as Errors Level of RDS BLOCK D(in RDS mode) or BLOCK E(in RBDS mdoe when ABCD_E flag is 1). 00 = 0 errors requiring correction 01 = 1~2 errors requiring correction 10 = 3~5 errors requiring correction 11 = 6+ errors or error in checkword, correctionnot possible Available only in RDS Verbose mode	00
	11:0	RESERVED		
40H	15	AUTO_SEEK	1=FULL AUTO SEEK MODE 0=HALF AUTO SEEK MODE（the same as 5802）	0
	14:4	reserved		0
	3:0	CHIP_FUNC[1:0]	0000=RX 0010=AM/LW 0011=SW 1100=DAC Others reserved	0000
41H	15	MEM_CLR	1=reset fifo 0=no reset	0
	14:9	reserved		000000
	8:0	MEM_DEPTH[8:0]	The length of FIFO Max=256	1_0000_0000
4AH	15:5	reserved		000_0000_0000
	4:0	MEM_INTMOD[4:0]	10000=FULL 00001=RPT EMPTY 01000=EMPTY Others reserved	00000
4BH	15:8	reserved		
	7:0	CHAN_NUM[7:0]	Valid when auto seek mode, equals	

REG	BITS	NAME	FUNCTION	DEFAULT
			to the number of station which has just been found.	
53H	15:11	reserved		
	10:0	CHAN_BOTTOM[10:0]	Valid when band[1:0]=3 & FM mode, unit 100kHz	
54H	15:11	reserved		
	10:0	CHAN_TOP[10:0]	Valid when band[1:0]=3 & FM mode, unit 100kHz	
64H	15:5	Reserved		000_0000_0110
	4:3	RDS_RXMOD[1:0]	10=FIFO mode 11=register mode Others reserved	11
	2:0	Reserved		0
71H	15:0	AM_SPACE[15:0]	AM/LW/SW seek step frequency, unit KHz	0000_0000_ 0000_1010
72H	15:0	AM_FREQ[15:0]	AM tune frequency, unit KHz	
73H	15:0	AM_BOTTOM[15:0]	AM/LW/SW seek bottom frequency, unit KHz	
74H	15:0	AM_TOP[15:0]	AM/LW/SW seek top frequency, unit KHz	
7AH	15:0	READAMFREQ[15:0]	Current AM/LW/SW frequency, unit KHz	
80H	15:4	reserved		
	3:1	AM_BW[2:0]	AM/LW/SW low pass filter bandwidth 000 = 1KHz 001 = 2KHz 010 = 3KHz 011 = 4KHz 100 = 2KHz 101 = 4KHz 110 = 6KHz 111 = 8KHz	
	0	reserved		

The information provided here is believed to be reliable; RDA Microelectronics assumes no liability for inaccuracies and omissions. RDA Microelectronics assumes no liability for the use of this information and all such information should entirely be at the user's own risk. Specifications described and contained here are subjected to change without notice for the purpose of improving the design and performance. All of the information described herein shall only be used for sole purpose of development work of RDA5830, no right or license is implied or granted except for the above mentioned purpose. RDA Microelectronics does not authorize or warrant any RDA products for use in the life support devices or systems.

For technical questions and additional information about RDA Microelectronics Inc.:

Website: www.rdamicro.com
Mailbox: info@rdamicro.com

RDA Microelectronics (Shanghai), Inc.
Tel: +86-21-50271108
Fax: +86-21-50271099

RDA Microelectronics (Beijing), Inc.
Tel: +86-10-63635360
Fax: +86-10-82612663