

SL2823 LPCD 功能使用说明

北京捷联微芯科技有限公司

1. LPCD 简介

传统读卡器不包含 LPCD，其工作模式为，周期性按照协议发送寻卡命令（REQA、REQB），之后保持电磁场打开等待一段时间，如果场内有卡则读卡器会收到卡片回复而进入通信流程；如果场内没有卡片，读卡器进入 standby 模式，如此周而复始。由于读卡器在寻卡时 RF 部分是打开的，尤其是天线要向外辐射电磁场，而发射电磁场消耗的电流通常比较大。而实际使用中，大部分时间读卡器的天线附近是没有卡片的，因此这种周期性的开场通信会造成功耗的浪费。

LPCD(Low Power Card Detection)是在寻卡之前，设法检测场内是否存在卡片，如果卡片存在再发送协议规定的命令，否则读卡器会进入 standby 模式。相比传统的方法，LPCD 的优势在于，仅检测卡是否存在需要的场开启时间远远小于寻卡所需要的开启时间，可以降低读卡器的平均功耗，另外，相比寻卡，LPCD 的检测过程通常不需要 MCU 参与，对功耗的降低也有好处。

2. LPCD 工作机制

一个 LPCD 的工作周期包括：

1. STANDBY，读卡器不开场，整个芯片仅有一个低频时钟在工作，低频时钟用于计时，等待时间 T1 后切换到 STARTUP 状态。
2. STARTUP，开场准备时间，由于芯片内部分电路及晶振需要一定的稳定时间，因此这个状态下提前将相关电路及电磁场打开，为检测做准备。
3. DETECTION，对场强进行检测，通过与预设的阈值进行比较，判断是否有卡在场内，如果检测到卡存在，则产生中断通知 MCU；如果没有检测到卡，则回到 STANDBY 状态，继续循环。

一个 LPCD 工作周期内，DETECTION 状态的时间由芯片特性决定，不建议调整。STANDBY 的时间 T1 可以根据需要进行调整，调整范围为 64ms~4.096s，调整精度为 64ms。STARTUP 状态的时间 T2 调整的范围为 62.5~4ms，调整精度为 62.5us。

3. LPCD 使用方法

为了方便 LPCD 功能的使用，我们提供了以下 API 函数

void osc_calib();

用于对 16K 时钟进行校准

agc_ctrl= agc_calib();

用于 agc 校准，并将校准值保存下来

void lpcd_init(unsigned char fst_en, unsigned char adc_value);

fst_en: 是否为首次校准

该函数用于正常使用中的 lpcd 初始化

t1：LPCD 过程中两次开场之间的时间间隔，取值范围为 0x0~0x3f，实际时间 $T=(t1+1)*64ms$ 。

sense：期望的 LPCD 检测灵敏度，刻度为 8mv，值越小，灵敏度越高，但是在无卡时误认为有卡的概率也越高。

adc_value：为 lpcd 唤醒时候的量化电压。

void lpcd_entry();

该函数用于进入 LPCD 模式，调用该函数后，芯片进入 LPCD 模式，并开启 LPCD 中断。

`uint8_t lpcd_exit();`

该函数用于退出 LPCD 模式，调用该函数后，芯片退出 LPCD 模式。并返回 lpcd 唤醒时候的量化电压。

3.1. Demo 软件处理流程

3.1.1. Demo 软件说明

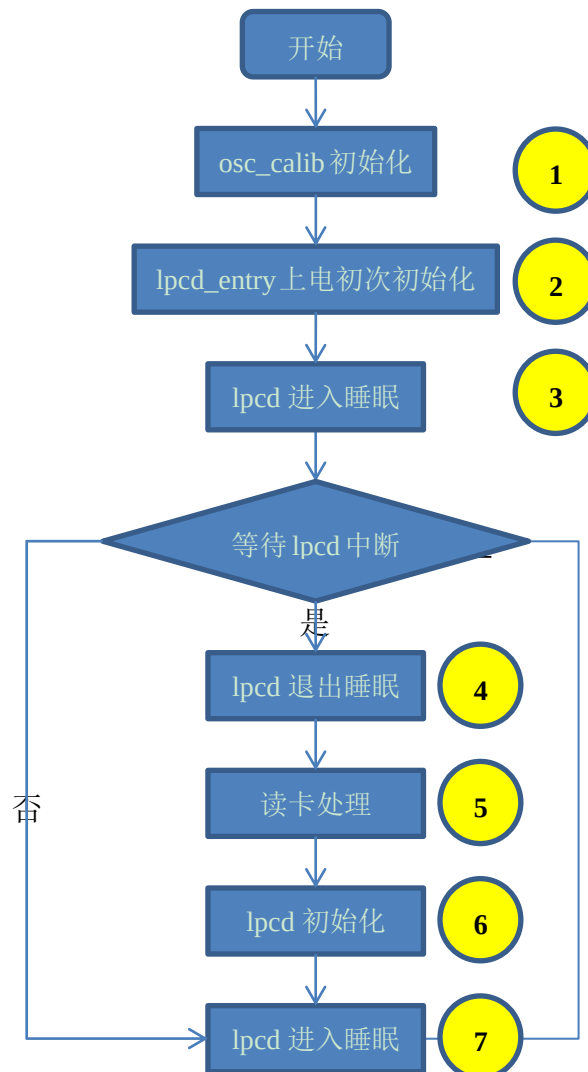
正常使用时，每次唤醒后需要对 LPCD 功能进行初始化，然后再进去睡眠。使用流程如下：

1. 调用 `osc_calib` 进行 16k 时钟初始化；
2. 调用 `lpcd_init(1, 0x40)` 进行初次 LPCD 功能初始化；
3. 调用 `lpcd_entry()` 进入睡眠。初次初始化后，会立即从睡眠中醒来。
4. 等待 LPCD 中断产生，然后调用 `lpcd_exit()` 退出睡眠状态。
5. 进行必要的初始化后，调用 `nfc_demo()` 进行正常读卡操作。
6. 读卡完毕后，调用 `lpcd_init(t1, sense)` 进行 LPCD 功能初始化。
7. 调用 `lpcd_entry()` 再次进入睡眠状态，再次等待 LPCD 中断产生。
8. 正常处理流程就是在 4、5、6、7 一直循环中。（4→5→6→7→4→5→6→7）

3.1.2. Demo 软件实现

```
void lpcd_demo(void) {
    uint8_t card_status;
    uint8_t adc_value;
    uint32_t wait_cnt = 0;
    irq_flag_io = 0;
    lpcd_cfg.t1 = LPCD_DETECT_PERIOD; //0x12 300ms; //0x20 500ms; //0x38 800ms;
    lpcd_cfg.sense = LPCD_DETECT_SENS;
    printf("***** start lpcd_demo() *****\r\n");
    ① osc_calib();
    ② lpcd_init(1, 0x40);
    ③ lpcd_entry();
    while(1) {
        #ifdef NOT_IRQ
        if (! (INT_PIN == 0))
        #else
        if (irq_flag_io)
        #endif
        {
            printf(".....\r\n");
            printf("\r\n***** 1. lpcd_exit() *****\r\n");
            irq_flag_io = 0;
            ④ adc_value = lpcd_exit();
            Rfid_Init();
            printf("***** 2. nfc_demo() *****\r\n");
            ⑤ card_status = nfc_demo();
            printf("***** 3. lpcd_init(t1, sense) *****\r\n");
            ⑥ lpcd_init(card_status, adc_value);
            pcd_antenna_off();
            printf("***** 4. lpcd_entry() *****\r\n");
            ⑦ lpcd_entry();
        } else {
            wait_cnt++;
            if (wait_cnt == 500000) {
                wait_cnt = 0;
                printf(".");
            }
        }
    }
}
```

3.1.3. Demo 软件流程



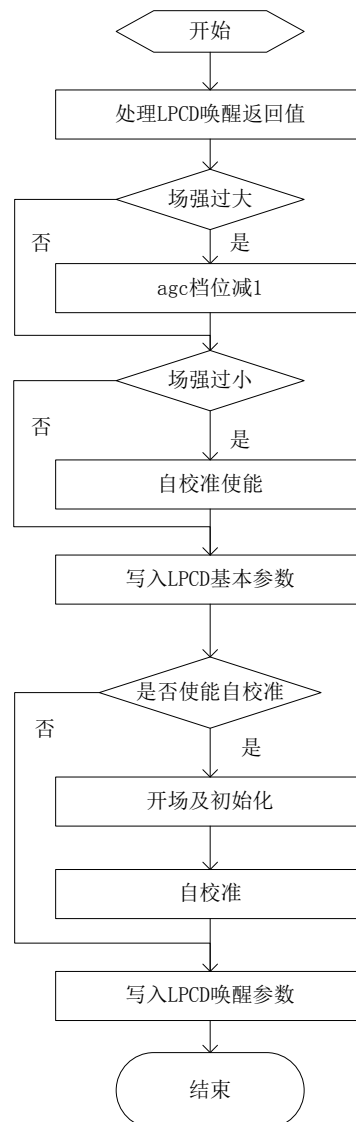
注意：唤醒后需要进行必要的初始化后，才能进行正常的读卡操作。

3.1.4. 常用参数使用说明

- LPCD_DETECT_PERIOD: 探卡周期：每隔多久开一次场
计算方式：探卡周期 $\approx$ LPCD_DETECT_PERIOD*64ms
典型值：0x5 320ms; //0x8 500ms ; //0x10 1s
- LPCD_DETECT_SENS (0x8) //0x2 17mv; //0x4 35mv
探卡灵敏度：场强值变化超过此值产生中断
说明：此值越小探卡越灵敏，但同时也存在误检风险，需根据实际板级测量出的场强值偏差大小设定
计算方式：零敏度 $\approx$ LPCD_DETECT_SENS*8.5mv
- AGC_RX_MAX
场强值上限：场强值超过此值预警，推荐值 0x80，需满足
(AGC_RX_MAX+LPCD_DETECT_SENS)<0xff

- AGC_RX_MIN
场强值下限：场强值小于此值预警，推荐值 0x20，需满足
 $AGC_RX_MIN > LPCD_DETECT_SENS$
- AGC_TARGET
场强目标值：校准阶段将场强调到此值附近，推荐值 0x60，需满足
 $AGC_TARGET < AGC_RX_MAX$
- ADC_TIME0_VAL：
开场时间
典型值：0x50(12us), 0xff(37us)
计算方式：开场时间 = $2 * ((ADC_TIME0 + 1) / 13.56) \mu s$
说明：此参数不能小于 0x50，开场时间越长测量的场强值越稳定。

3.2. LPCD 校准流程



- 处理 LCDP 唤醒返回值：如果此值合理执行正常传输标志，执行相应自校准流程；如果此值不合理强制进行自校准。
- AGC 自校准是将场强调到合适范围内，睡眠后进行卡片检测。
- LPCD 写入的唤醒参数，其实就行 ADC 采集出来的阈值参数。此值如果不合理会存在无法唤醒和一直唤醒的问题。
- LPCD
- 是否进行 LPCD 自校准是根据读卡逻辑进行判断。
 - 1、读卡时能够寻到卡片，读卡完成以后必须执行自校准流程。（进入睡眠以后必定唤醒）。
 - 2、读卡时没有寻到卡片，退出后不用执行自校准流程。睡眠后不会自己唤醒，只有卡片靠近探测到卡片后才会唤醒。

注意：一定要按照上面写的读卡流程传入正确的处理参数。

4. LPCD 相关寄存器

LPCD 相关寄存器如下：

（*由于 LPCD 功能各参数调整较复杂，建议客户尽量直接使用我们提供的 API，不要直接对寄存器进行修改）

4.1. DIVIEN(0x03)

Addr	Name	Bits	Default	R/W	Functions
0x03	irq_push_pull	7	0	RW	IRQ PIN 工作模式： 0: IRQ PIN 工作在 open-drain 方式 1: IRQ PIN 工作在标准 CMOS 输出方式
	lpcd_fst_ien	6	0	RW	首次 LPCD 中断使能 0: LPCD_FST 中断只反映在内部寄存器，不会引起 IRQ PIN 的变化 1: LPCD_FST 中断反映在 IRQ PIN 上
	lpcdien	5	0	RW	LPCD 中断使能 0: LPCD 中断只反映在内部寄存器，不会引起 IRQ PIN 的变化 1: LPCD 中断反映在 IRQ PIN 上
	mfinactien	4	0	RW	MFIN 中断使能 0: MFIN 中断只反映在内部寄存器，不会引起 IRQ PIN 的变化 1: MFIN 中断反映在 IRQ PIN 上
	-	3	0	R	-
	crcien	2	0	RW	CRC 中断使能 0: CRC 中断只反映在内部寄存器，不会引起 IRQ PIN 的变化 1: CRC 中断反映在 IRQ PIN 上
	-	1:0	0	R	-

4.2. DIVIRQ(0x05)

Addr	Name	Bits	Default	R/W	Functions
0x05	set2	7	0	W	表示当前写 DIVIRQ 的操作是要清除中断还是将中断置 1 0: 当前写 DIVIRQ 将中断清 0 1: 当前写 DIVIRQ 将中断置 1
	lpcd_fst_irq	6	0	D	首次 LPCD 中断, 当首次 LPCD 检测完成时产生中断
	lpcdirq	5	0	D	LPCD 中断, 当 LPCD 检测到卡时产生中断
	mfinactirq	4	0	D	MFIN 中断, 当检测到 MFIN 的高电平或低电平时产生中断, 高电平或低电平由 MODREG(0x11) 的 bit[3]决定
	-	3	0	R	-
	crcirq	2	0	D	CRC 中断, CRC 计算完成后给出中断
	-	1:0	0	R	-

4.3. EXT_OSC_CALIB_TARGET1(0x4C)

Addr	Name	Bits	Default	R/W	Functions
0x4C	osc_target[7:0]	7:0	0x9F	RW	OSC 校准时使用 27.12M 对 16K 进行计数时的目标计数值

4.4. EXT_OSC_CALIB_TARGET2(0x4D)

Addr	Name	Bits	Default	R/W	Functions
0x4D	-	7:3	0	R	-
	osc_target[10:8]	2:0	0x06	RW	OSC 校准时使用 27.12M 对 16K 进行计数时的目标计数值

4.5. EXT_OSC_PARAM(0x4E)

Addr	Name	Bits	Default	R/W	Functions
0x4E	osc_calib_done	7	0	R	OSC 频率校准结束标志位 1: 校准结束 0: 校准未结束
	-	6:5	0	R	-

	osc_param	4:0	0x10	RW	OSC 频率控制参数，可以由 MCU 直接写入或者通过校准电路自动更新
--	-----------	-----	------	----	-------------------------------------

4.6. EXT_AGC_CTRL(0x5F)

Addr	Name	Bits	Default	R/W	Functions
0x5F	agc_en	7	1	RW	片内 AGC 使能控制 0: 片内 AGC 不使能 1: 片内 AGC 使能
	Reserved	6:4	0	RW	Reserved
	agc_ctrl	3:0	0x7	RW	AGC 使能时的片内 R1 的阻值控制/ohms 4'b0000: 50 4'b0001: 70 4'b0010: 100 4'b0011: 150 4'b0100: 220 4'b0101: 320 4'b0110: 460 4'b0111: 670 4'b1000: 1k 4'b1001: 1.5k 4'b1010: 2.5k 4'b1011: 4.5k 4'b1100: 9.7k 4'b1101: 51k

4.7. EXT_ADC_CTRL(0x81)

Addr	Name	Bits	Default	R/W	Functions
0x81	adc_value_ok	7	0	RW	ADC 测量值有效指示标志 1: 有效 0: 无效
	Reserved	6	0	RW	Reserved
	adc_meas_mode	5	0	RW	ADC 测量平均模式 1: 滑窗滤波，历史 8 个值平均作为均值 0: alpha 滤波，根据滤波系数计算均值
	adc_meas_ant_off	4	0	RW	ADC 测量时采样完成后是否关掉场 1: 关闭 0: 不关闭
	adc_meas_times	3: 2	0	RW	ADC 测量次数 0: 1 次 1: 2 次

					2: 4 次 3: 8 次
	sof_b_start_adc_en	1	0	RW	测量 TypeB 发送的高低场强, 用于计算调制系数
	adc_start	0	0	RW	1, 启动 ADC 测量, 写完自动清零

4.8. EXT_ADC_TIME0(0x82)

Addr	Name	Bits	Default	R/W	Functions
0x82	adc_sample_time	7:0	0x60	RW	ADC 采样时间, 单位 2/fc

4.9. EXT_ADC_TIME1(0x83)

Addr	Name	Bits	Default	R/W	Functions
0x83	adc_cmp_time	7:0	0x28	RW	ADC 采样比较时间, 单位 1/fc

4.10. EXT_LPCD_CTRL0 (0x86)

Addr	name	Bits	Default	R/W	Description
0x86	lpcd_out_dig	7	1	R	adc 比较结果
	lpcd_env_byass	6	0	RW	LPCD 包络模式下, 包络信号跨过采样开关, 直接连接到源极跟随器的使能控制 1: LPCD 采样支路被旁路 0: LPCD 采样支路正常工作
	lpcd_fst_en	5	0	RW	是否为第一次启动 lpcd 1: 是第一次启动 0: 不是第一次启动
	lpcd_out_inv	4	1	RW	adc 比较结果反相使能 1: 反向 0: 不反向
	lpcd_envsel	3	1	RW	LPCD 采样输入信号选择 1: LPDC 直接采样 RX 0: LPCD 采样 FSD 输出的 RX 包络信号
	lpcd_sample	2	0	RW	LPCD 的采样时钟使能 1: LPCD 电路时钟被使能, 电路采样 0: LPCD 电路时钟被关闭, 电路保持之前的采样结果不变

lpcd_debug	1	0	RW	LPCD debug 功能使能控制 1: LPCD debug 功能使能 0: LPCD debug 功能不使能
lpcd_work	0	0	RW	软掉电模式下是否打开 16K 模块时钟开关 1: 打开 16K 模块时钟 0: 关闭 16K 模块时钟

4.11. EXT_LPCD_CTRL1 (0x87)

Addr	name	Bits	Default	R/W	Description
0x87	antoff_after_det_ph2	7	0	RW	在 Phase2 阶段检测完成后是否关场 1: 关场 0: 不关
	antoff_after_det_ph1	6	1	RW	在 Phase1 阶段检测完成后是否关场 1: 关场 0: 不关
	lpcd_envsel_ph2	5	1	RW	phase2 阶段采用包络还是采样 1: 采样 0: 包络
	lpcd_envsel_ph1	4	0	RW	phase1 阶段采用包络还是采样 1: 采样 0: 包络
	lpcd_nofst_en	3	0	RW	lpcd 非第一次启动使能 1: 使能 0: 关闭
	lpcd_calib_en	2	0	RW	lpcd 校准模式，每做一次 lpcd 上报一次中断 1: 启动 0: 关闭
	need_detect_ph2	1	0	RW	是否需要做 Phase2 检测 1: 检测 0: 不检测
	need_detect_ph1	0	1	RW	是否需要做 Phase1 检测 1: 检测 0: 不检测

4.12. EXT_LPCD_CTRL2 (0x88)

Addr	name	Bits	Default	R/W	Description
0x88	lpcd_sample_keep	7	0	RW	采样完成后是否保持 lpcd_sample 为高 1: 保持

					0: 不保持
	Reserved	6	0	R	Reserved
	lpcd_det_done_dly	5:4	0	RW	检测完成后等待周期: 单位 1/osc_16k
	Reserved	3	0	R	Reserved
	ignore_times	2:0	0	RW	忽略前几次检测结果

4.13. EXT_LPCD_CTRL3 (0x89)

Addr	name	Bits	Default	R/W	Description
0x89	Reserved	7	0	R	Reserved
	detect_ok_thd	6:4	1	RW	检测成功次数阈值
	Reserved	3	0	R	Reserved
	detect_times	2:0	1	RW	连续检测成功次数

4.14. EXT_LPCD_TIME0 (0x8A)

Addr	name	Bits	Default	R/W	Description
0x8A	ant_stable_time	7:0	0x20	RW	开场稳定时间

4.15. EXT_LPCD_TIME1 (0x8B)

Addr	name	Bits	Default	R/W	Description
0x8B	lpcd_period_time	7:0	0xa	RW	LPCD 探测周期

4.16. EXT_LPCD_TIME2 (0x8D)

Addr	name	Bits	Default	R/W	Description
0x8D	lpcd_startup_time	7:0	0x3f	RW	LPCD 启动稳定时间

4.17. EXT_LPCD_PARAM0 (0x8E)

Addr	name	Bits	Default	R/W	Description
0x8E	card_det_abs_thd1_ph1	7:0	0	RW	卡检测 Phase1 绝对值下限阈值

4.18. EXT_LPCD_PARAM1 (0x8F)

Addr	name	Bits	Default	R/W	Description
0x8F	card_det_abs_thd2_ph1	7:0	0	RW	卡检测 Phase1 绝对值上限阈值

4.19. EXT_LPCD_PARAM2 (0x90)

Addr	name	Bits	Default	R/W	Description
0x90	card_det_rel_thd_ph1	7:0	1	RW	卡检测 Phase1 相对值阈值

4.20. EXT_LPCD_PARAM3 (0x91)

Addr	name	Bits	Default	R/W	Description
0x91	Reserved	7:6	0	R	Reserved
	lpcd_rlt_or_en	5	0	RW	phase1 和 phase2 检测结果相或
	lpcd_rlt_and_en	4	0	RW	phase1 和 phase2 检测结果相与
	lpcd_ph1_rel_det_en	3	1	RW	phase1 相对值检测使能
	lpcd_ph1_abs_det_en	2	0	RW	phase1 绝对值检测使能
	filt_coef_ph1	1:0	0	RW	卡检测 Phase1 滤波系数

4.21. EXT_LPCD_PARAM4 (0x92)

Addr	name	Bits	Default	R/W	Description
0x92	card_det_abs_thd1_ph2	7:0	0	RW	卡检测 Phase2 绝对值下限阈值

4.22. EXT_LPCD_PARAM5 (0x93)

Addr	name	Bits	Default	R/W	Description
0x93	card_det_abs_thd2_ph2	7:0	0	RW	卡检测 Phase2 绝对值上限阈值

4.23. EXT_LPCD_PARAM6 (0x94)

Addr	name	Bits	Default	R/W	Description
0x94	card_det_rel_thd_ph2	7:0	0	RW	卡检测 Phase2 相对值阈值

4.24. EXT_LPCD_PARAM7 (0x95)

Addr	name	Bits	Default	R/W	Description
0x95	Reserved	7:4	0	R	Reserved
	lpcd_ph2_rel_det_en	3	0	RW	phase2 相对值检测使能
	lpcd_ph2_abs_det_en	2	0	RW	phase2 绝对值检测使能
	filt_coef_ph2	1:0	0	RW	卡检测 Phase2 滤波系数

4.25. EXT_LPCD_HYS (0x98)

Addr	name	Bits	Default	R/W	Description
0x98	lpcd_hys	7:0	0x80	RW	lpcd adc 比较阈值

5. LPCD 外围参考电路

LPCD 功能的实现需要外围电路的支持，外围电路的参考原理图如下图所示。其中天线及匹配电路为常见结构，推荐的天线在 13.56MHz 的谐振阻抗为 40 Ohms。注意 RX 支路通过电阻 R2 连接到 C2 上，如果 R2 接到 L0 有可能会使 LPCD 的灵敏度下降。Cvmid 推荐使用 1nF。相对于使用 100nF 或者更大的电容，Cvmid 取 1nF 可以使得 LPCD 检测阶段信号的偏置电平和信号的幅度快速达到稳定，可以提高 LPCD 检测的稳定性和灵敏度。匹配电路中部分元器件的参考取值如下表。

元器件	取值
CRx	15pF
Cvmid	1nF
R1	NC 或者保留(如果保留，推荐值为 330 欧)

	姆，须根据实际环境调整)
R2	2.2KOhms (须根据实际环境调整)

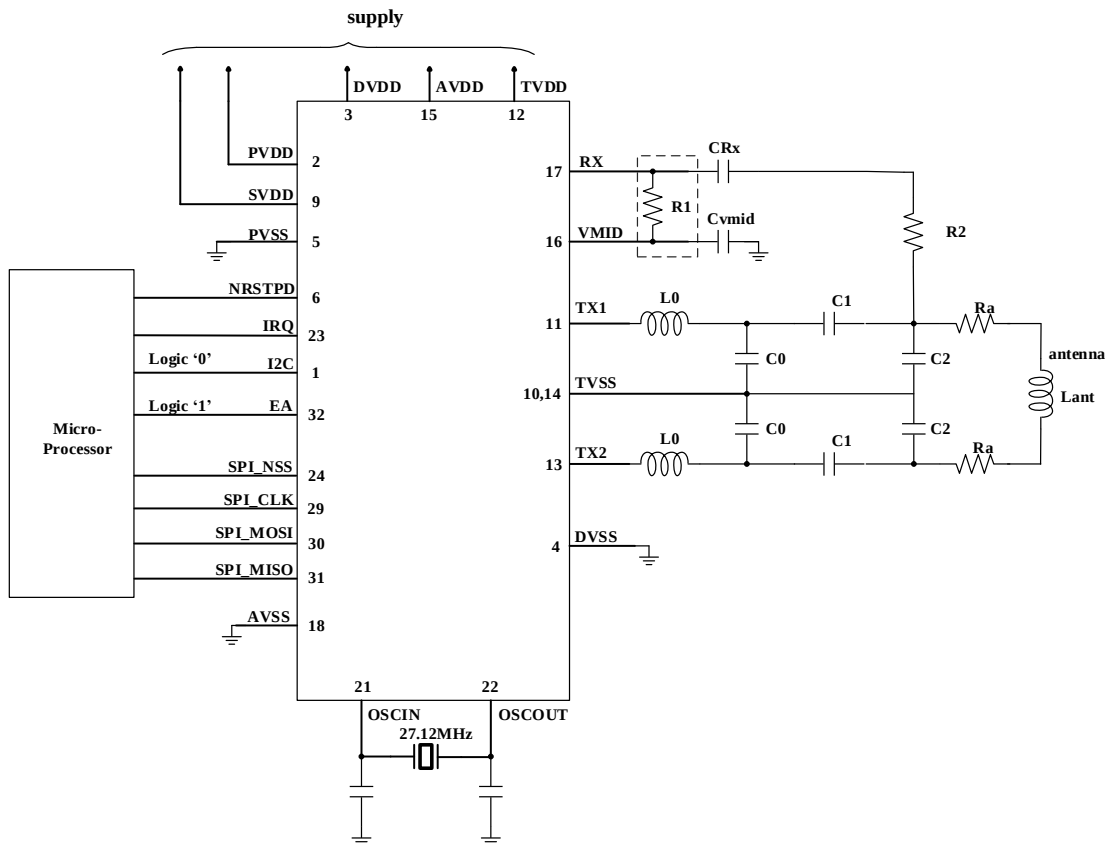
2823 内置了 AGC 功能，可以克服温漂和电源电压波动等因素对 LPCD 的影响。AGC 功能是通过调整片内某个电阻，来调整 RX (pin17) 管脚的信号幅度，电阻的具体值请参考前述下表。

Addr	Name	Bits	Default	R/W	Functions
0x5F	agc_en	7	1	RW	片内 AGC 使能控制 0: 片内 AGC 不使能 1: 片内 AGC 使能
	Reserved	6:4	0	RW	Reserved
	agc_ctrl	3:0	0x7	RW	AGC 使能时的片内 R1 的阻值控制/ohms 4'b0000: 50 4'b0001: 70 4'b0010: 100 4'b0011: 150 4'b0100: 220 4'b0101: 320 4'b0110: 460 4'b0111: 670 4'b1000: 1k 4'b1001: 1.5k 4'b1010: 2.5k 4'b1011: 4.5k 4'b1100: 9.7k 4'b1101: 51k

下图中的电阻 R1 有两种处理方法，可以去掉，也可以保留。

如果去掉该电阻，LPCD 校准时将使用 2823 集成的 AGC 功能来调整 RX 管脚的信号幅度。

如果保留片外 R1，该电阻将与 2823 内部的 AGC 电阻并联后再与片外电阻 R2 形成分压关系来确定信号幅度。保留片外电阻 R1，AGC 调整信号幅度的步长会减小，即 RX 端口信号变化可用的档位将变得更细。



5.1. R1 和 R2 的调试方法

R1 和 R2 电阻的确定，就是要保证 RX 管脚信号的幅度处在合理范围内。当在 R1 和 R2 取值合理的条件下，AGC 锁定时，串口打印的 RX 信号的量化值应在 0x50~0x70 之间，用示波器观测芯片 RX 管脚上的信号最大值 V_{peak} 应为 2.9V~3.1V 之间。

实际操作中，建议通过串口打印，直接观测 RX 信号的量化值随 AGC 档位的变化。类似 demo 程序中的函数 test_agc() 的功能，一一列出 AGC 档及其对应的 RX 信号量化值。

如果去掉片外电阻 R1，那么 R2 电阻的取值要相对大一点，推荐在 3.3K~10K 范围内选择。相对而言，R2 取值越大，则 RX 信号的量化值落在 0x20~0x80 范围内的 AGC 档位将越多。注意如果档位数过多，说明 RX 管脚的电压幅度会相对偏小，甚至可能超过 AGC 工作的电压范围。推荐档位数在 2~5 个。

第一步，建议从 4.7K 做初值，通过打印观测 RX 信号的量化值落在 0x20~0x80 范围内的 AGC 的档位数。

第二步，如果步骤一中的档位数 ≤ 1 ，则须减小 R2 电阻的值。如果档位数 ≥ 5 ，则须增大 R2 的取值。

第三步，重复上述步骤一和二，直到档位数在 2~5 个之间即可。

如果保留片外电阻 R1，那么 R2 电阻的取值要相对小一点，推荐在 1.8K~4.7K 范围内选择。与去掉片外 R1 类似，推荐档位数在 2~5 个。区别在于，保留的片外 R1 将与片内 AGC 的电阻并联，再与片外 R2 电阻分压。

第一步，建议片外 R1 以 330 欧姆做初值，R2 以 2.2K 做初值，通过打印观测 RX 信号的量化值落在 0x20~0x80 范围内的 AGC 的档位数。

第二步，如果步骤一中的档位数 ≤ 1 ，则须减小 R2 电阻的值。如果档位数 ≥ 5 ，则须增大 R2 的取值。

第三步，重复上述步骤一和二，直到档位数在 2~5 个之间。如果发现当前 R1 取值条件下，R2 取常用的电阻得到的 AGC 档位数都偏多，则需要增大片外 R1。否则需要减小 R1。换完 R1 电阻再重复前述步骤一和二。

第四步，重复上述三个步骤，直到档位数在 2~5 个之间即可。

6. 修改记录

修改时间	修改内容
2021.12.3	修改图 2，去掉 16K 校准、场强测量步骤
2021.12.3	修改 4.1DivIen 寄存器描述
2021.12.3	修改 4.8 EXT_ADC_TIME0 寄存器默认值
2021.12.3	删除地址 0x96 寄存器
2021.12.3	修改一些格式不统一的问题
2021.12.11	修改软件处理流程、LPCD 自校准流程、LPCD 外围参考电路说明
2022.8.5	修改 AGC 控制寄存器的描述，增加 LPCD 外围电路中 R1 和 R2 的调试方法 增加 3.1.4 节常用参数使用说明
2022.8.8	修正部分文字错误