PAN163 应用

Rev 1.0

Panchip Microelectronics

www.panchip.com

目录

第1章 PAN163 应用原理图

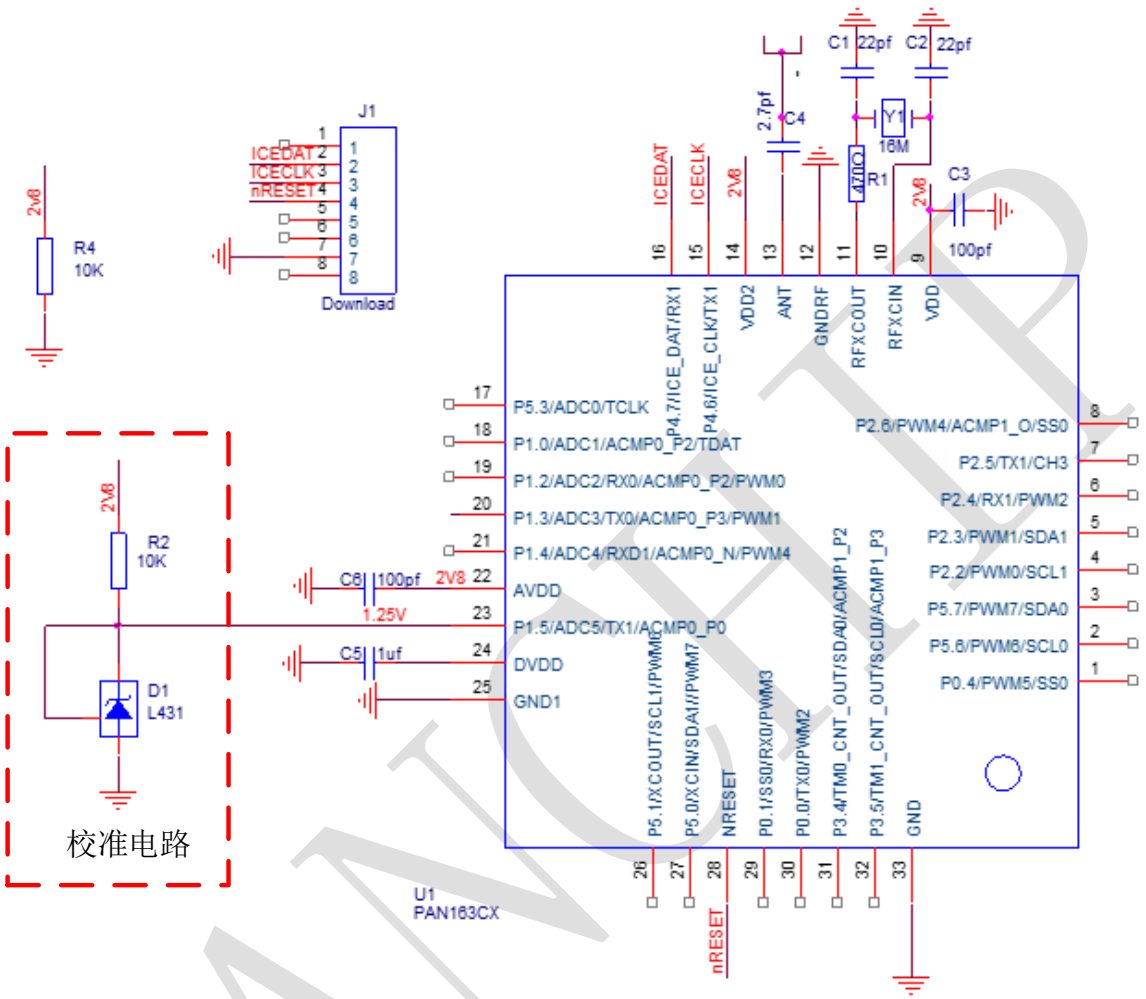


图 1 PAN163 应用原理图

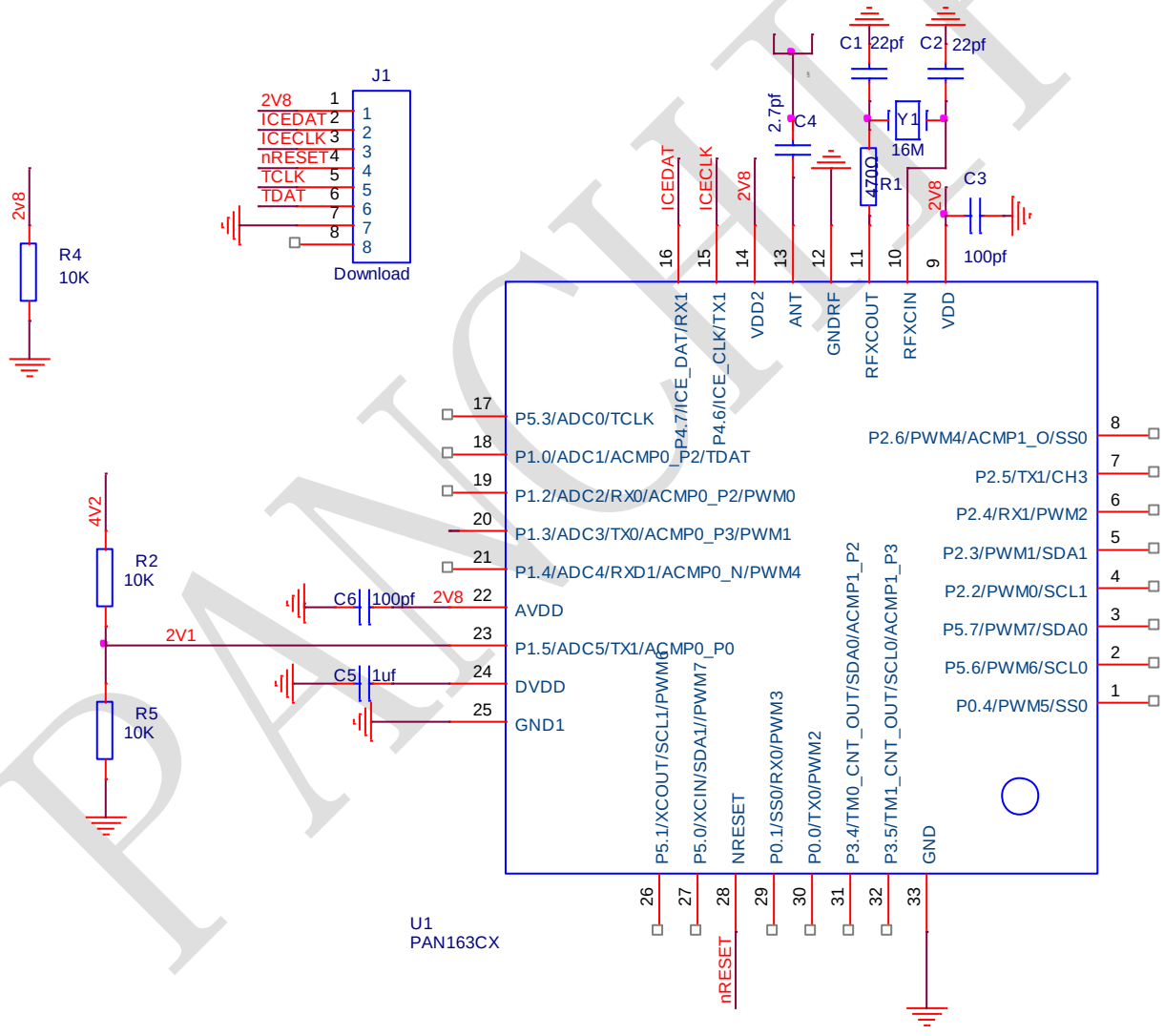
注意事项：

1. **C1/C2 谐振电容，根据不同型号的晶振进行微调，推荐范围 15~36pF;**
2. **C4 推荐 2.7pF，可选择范围在 2~4pF;**
3. **NRESET 脚可以悬空，芯片衬底建议 GND;**
4. **MCU 供电为 2.4--2.8V;**
5. **P1.4 管脚外部输入电压不能超过(AVDD+0.3V);**
6. **使用一个基准电压 1.25V 通道作为 ADC 校准;**
7. **R4 防止 IO 脚漏电;**
8. **校准电路：如果应用场合需要很高的 ADC 精度（1%以内），需要添加这个电路，否则**

可以不添加这部分电路。注意：校准电路的 L431 类的芯片（已测试芯片如 L431LM3B）精度要在 0.5%以内，输出电压在 0.8-2.5V 直接都可以。参考库中工程 ADC_OSC_Calib

第2章 ADC校准测试内容

2.1 使用 PAN-LINK 烧录器进行 ADC 校准



ADC5 通道电压 Vch=2.1V，电池电压为 Vbat=4.2V。在量产 8 线烧录时 Vbat 提供一个稳定高精度的电流源 4.2V，勾选 PAN-LINK ADC 校准功能。在烧录时 PAN163 会开启 ADC5，并将检测到的 ADC_CODE 存储在 UCID2 中。ADC_CODE=UCID2&0XFFF;改变电池电压为 Vbat1，这时 ADC5 检测到的 ADC 值记为 ADC_CODE1。Vbat1=2*2.1V*ADC_CODE1/ADC_CODE。

ADC_CODE 存储在 UCID2 中的格式：烧录的时候 PAN-LINK 会将 12BIT 的 ADC_CODE 存储在 UCID2[0：11],ADC_CODE 取反存储在 UCID2[12:23],UCID2[24:31]=0X50。

2.2 ADC 内部校准

2.2.1 出厂两点式默认校准

出厂时测试两个已知固定电压储存在 ID 区域。ADC 模式为 Y=KX+B。根据这两个 ID 值求出 K,B。参考工程 ADC_Calibrate。**ADC 测试电压范围推荐为 1.1V-2.4V,精度可达 3%。**

2.2.2 内部 BGAP 和外部基准电压校准

推荐使用

A. ADC 模型为 Y=KX+B

B. Y=KX,使用内部基准电压测试出系数 K.

ADC 模型 V=K*CODE ;V 电压值；K:ADC 比例系数 , Xcode 表示 ADC 转换结果（0--4095）。**内部 BGAP 为恒定不变电压**。Vbgap=k*CODE_bgap;芯片在出厂时给芯片供恒定电压 2730mv，使用 ADC 测试出 BGAP 值,记为 BGAP_CODE_UID0;当前供电情况想,测试 BGAP 电压记为 bgap_code，k=2730/4095*BGAP_CODE_UID0/bga_code 。

待测通道电压值 V_x=X_CODE*2730/4095*BGAP_CODE_UID0/bga_code 。**ADC 测试电压范围推荐为 1.1V-2.4V,此时不加基准电压，精度可达 3%。**

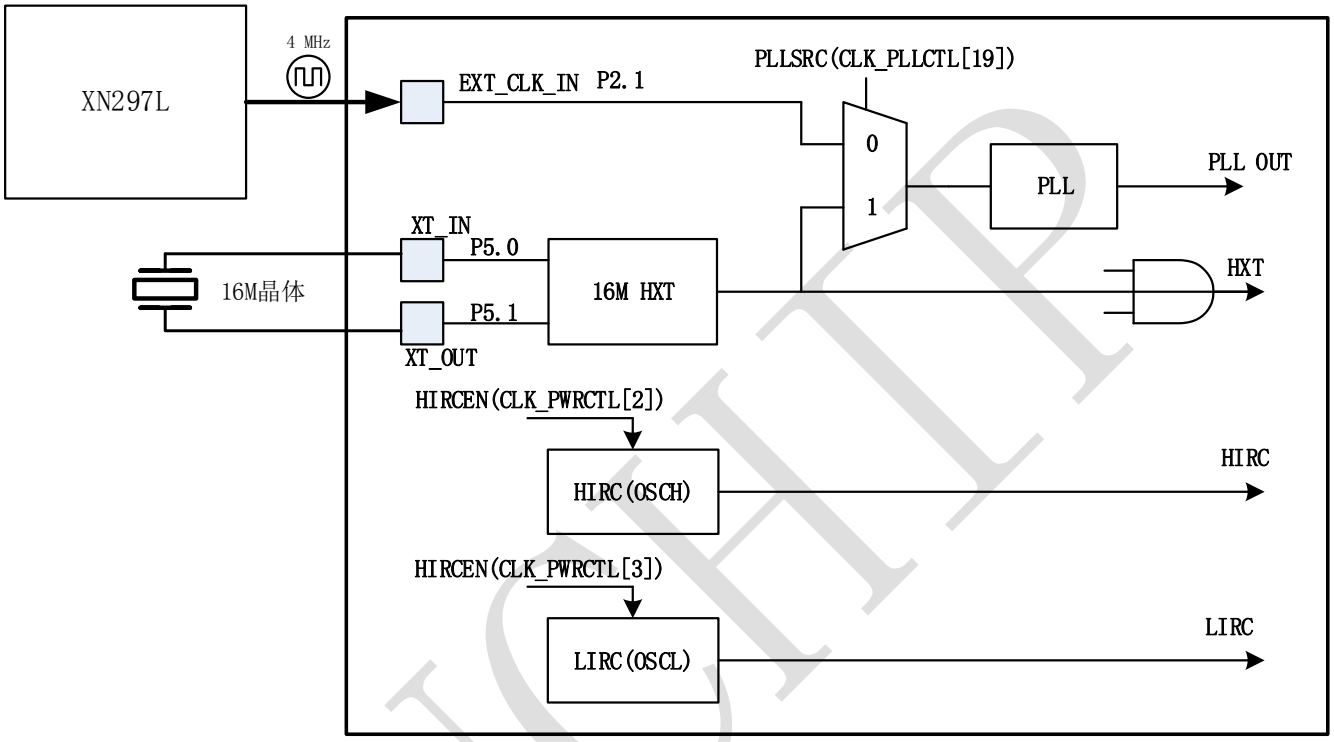
C. 根据外部基准电压算出 B

如果要求 ADC 精度为 1%，添加图 1 中校准电路。使用外部电压来消除 ADC 偏差 B。

参考工程 ADC_OSC_Calib。

第3章 系统时钟

3.1 系统时钟框图



PAN163 系统时钟源框图

系统时钟源选择有三种方式：

1.外接 16Mhz 晶体作为系统时钟

2.使用内部 RC 作为系统时钟（有 48Mhz 和 55Mhz 两档）

3.XN297L 输出 4M 时钟经过 PLL 倍频到 48Mhz 作为系统时钟

3.2 外接 16Mhz 晶体

1、直接将 16M 时钟作为系统时钟，使用库函数 CLK_InitHXT（）

2、经过 PLL 将 16M 时钟倍频到 48Mhz，使用库函数 CLK_InitHXTPLL（）

3.3 使用内部 RC

芯片内部有可调高频 RC 振荡器，范围为 20—60Mhz，出厂时将内部 RC 分别校准到 48Mhz（1.1%精度）和 55Mhz（1%精度）。

1、48Mhz 档：调用库函数 CLK_InitHIRC（）

2、55Mhz 档：定义宏__HIRC_55M__并调用库函数 CLK_InitHIRC（）

3.4 使用 XN297L 时钟

3.4.1 工作原理

XN297L 输出 4M 时钟经过 PLL 倍频到 48Mhz 作为系统时钟。MCU 使用内部 RC 启动芯片，使 XN297L 一直输出 4Mhz 的方波时钟，MCU 系统时钟切换到外部时钟。

XN297L 输出时钟条件：

1、RF_CAL 的 bit 23（EN_CLK_OUT）配置为 1

2、RF_CAL2 的 bit <35:34>（CLK_SEL）配置 10：4MHz

3、XN297L 处于 RX/TX/STB2 状态（即 CE 为高）。发送端 TX FIFO 寄存器为空并且 CE 引脚置 1，进入 STB2(预发射状态)。

3.4.2 使用方法

参考工程 OSC_XN297L

1、初始化内部时钟

SYS_UnlockReg();

CLK_InitHIRC();

2. rf_init

CE_HIGH

BB_cal_data[] = {0x0A,0x6D,0x67,0x9C,0x46}; //1M 下 rf 配置

RF_cal_data[] = {0xF6,0x37,0xDD

RF_cal2_data[] = {0x45,0x21,0xef,0xAC,0x5A,0x50};

Dem_cal_data[] = {0x01};

Dem_cal2_data[] = {0x0b,0xDF,0x02};

RX_MODE()

3.切换到外部时钟

CLK_InitExtOSC4MPLL()

SYS_LockReg();

3.5 通过 XN297L 检测内部 RC 时钟

3.5.1 工作原理

内部 RC 在出厂时已校准到 1%，如果能够知道具体 RC 时钟频率那么将减少软件计算误差。

检测内部 RC 时钟方法：内部 RC 做系统时钟，XN297L 输出 2M 的方波到 IO 脚 P2.1。MCU 开启定时器,同时 IO 脚 P2.1 采集固定的方波个数.通过读取定时器的计数值来判断内部 RC 的大小，更新 SystemCoreClock 变量。

3.5.2 使用方法

参考工程 OSC_XN297L_CALIB

1.启动系统时钟

SYS_UnlockReg();

CLK_InitHIRC();

SystemCoreClockUpdate();

2.校准变量 SystemCoreClock

CLK_Calibrate();