

MASTERS 2009

THE WORLDWIDE CONFERENCE FOR EMBEDDED CONTROL ENGINEERS



RTC

实时时钟的实现

课程目标

学习完本课程后，您将了解：

- 晶振的特性
 - **RTC**的功耗和可靠性
 - 漂移
 - 频率随温度的变化
 - 限制**RTC**精度的因素
- PIC单片机中的硬件**RTC**
 - **RTC**的不同实现方法
- 校正技术的实现

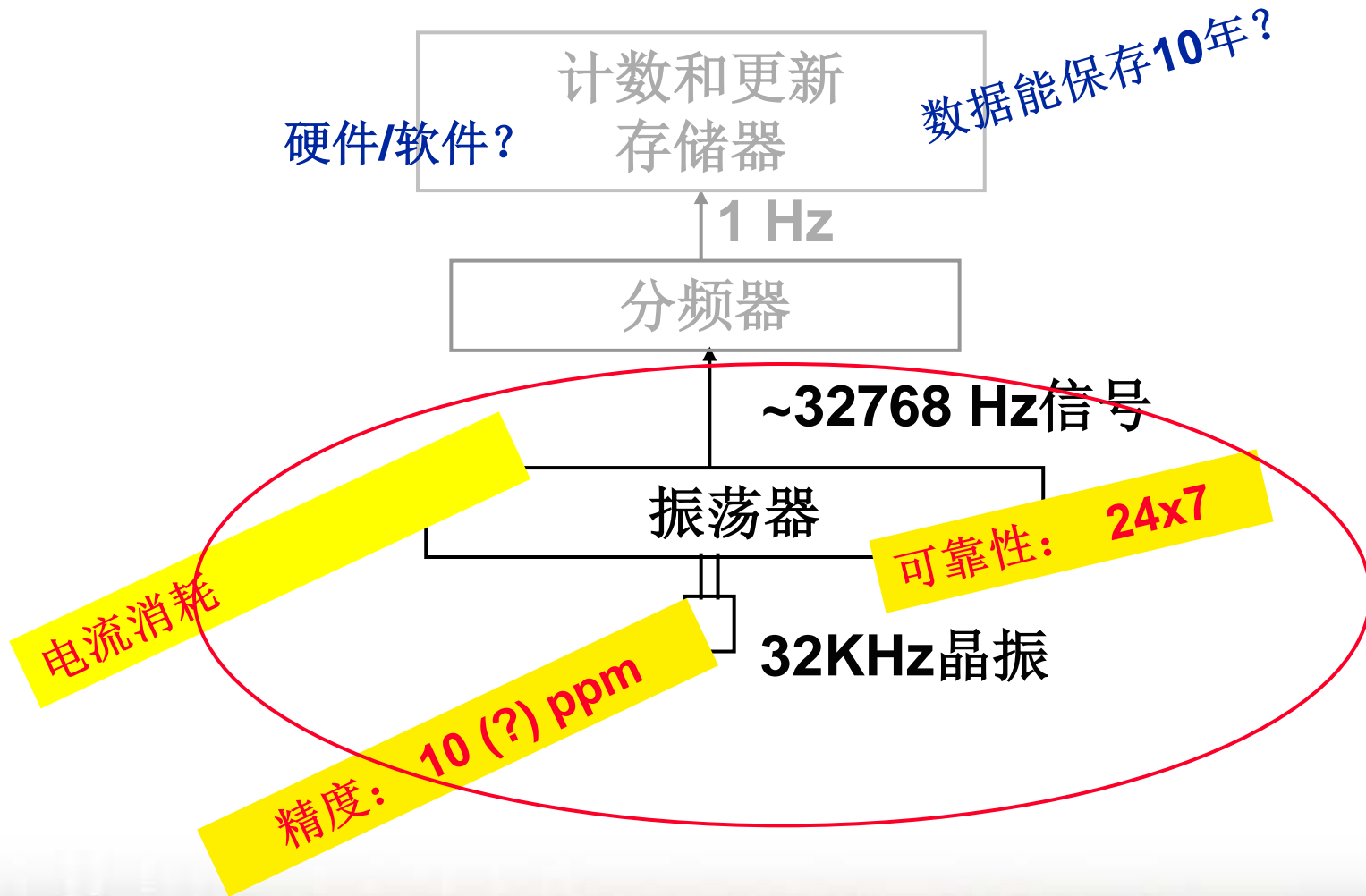
课程安排

- RTC的核心
 - **32768 Hz晶振和振荡器**
 - 驱动电平
 - 可靠性
 - 频率变化和校正
- PIC单片机的RTCC外设
 - 与竞争对手的某些器件作比较
- RTC的实现以及校正示例
 - 漂移
 - 温度补偿

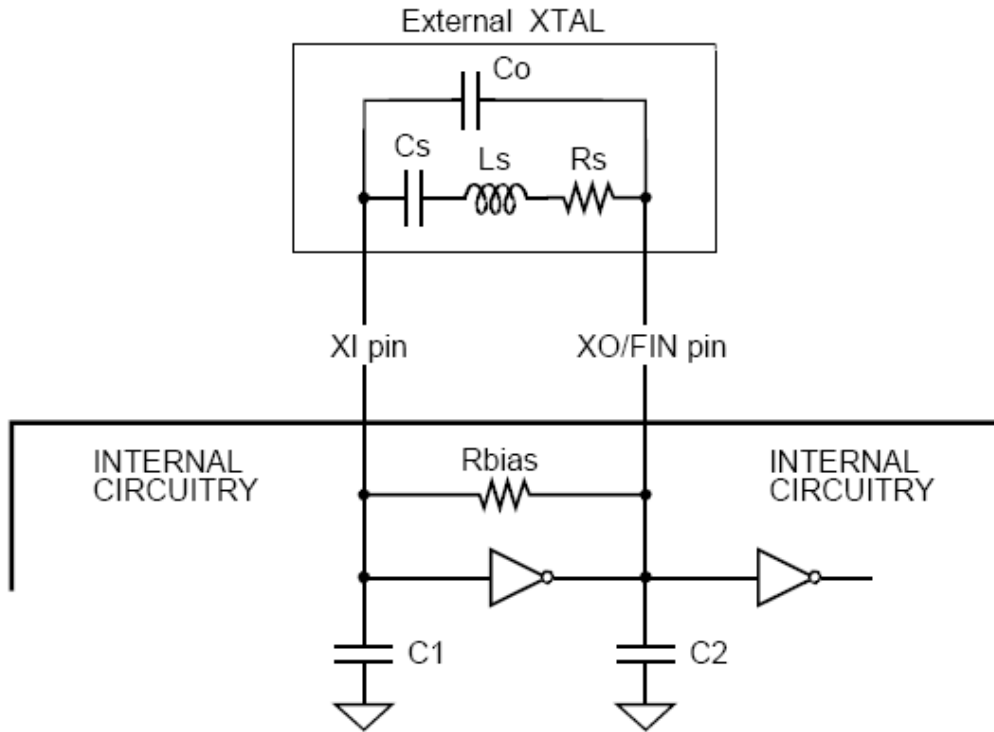
RTC的核心



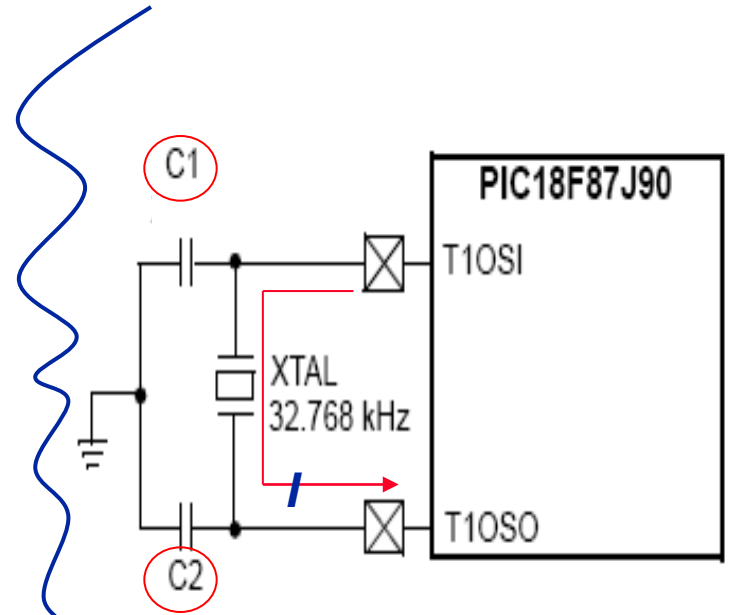
RTC框图



振荡器驱动



典型的晶振和振荡器



PIC单片机使用的典型的
32768 Hz 振荡器

PIC18F87J90振荡器

- PIC18F87J90: 32768 Hz振荡器的电流消耗
 - 低功耗模式
 - 1.0uA (条件: 25 °C 和2.0V)
 - 6uA (条件: 25 °C 和3.3V)
 - 高功耗模式
 - 针对3uA设计
 - 可靠的晶振设计

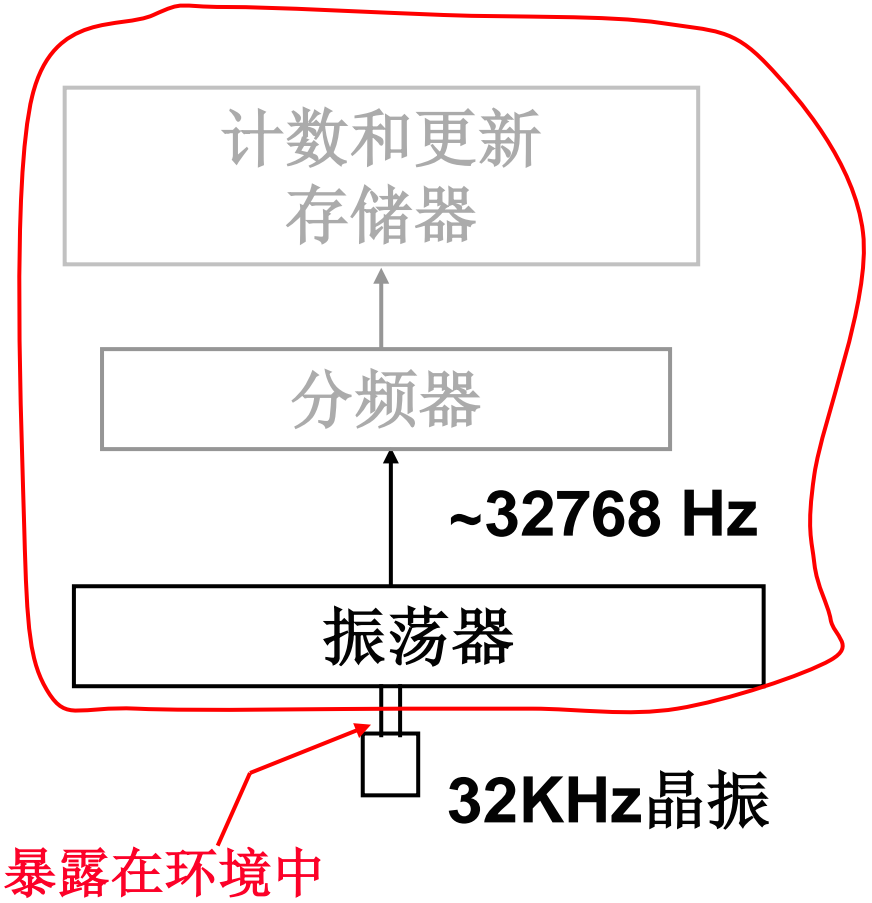
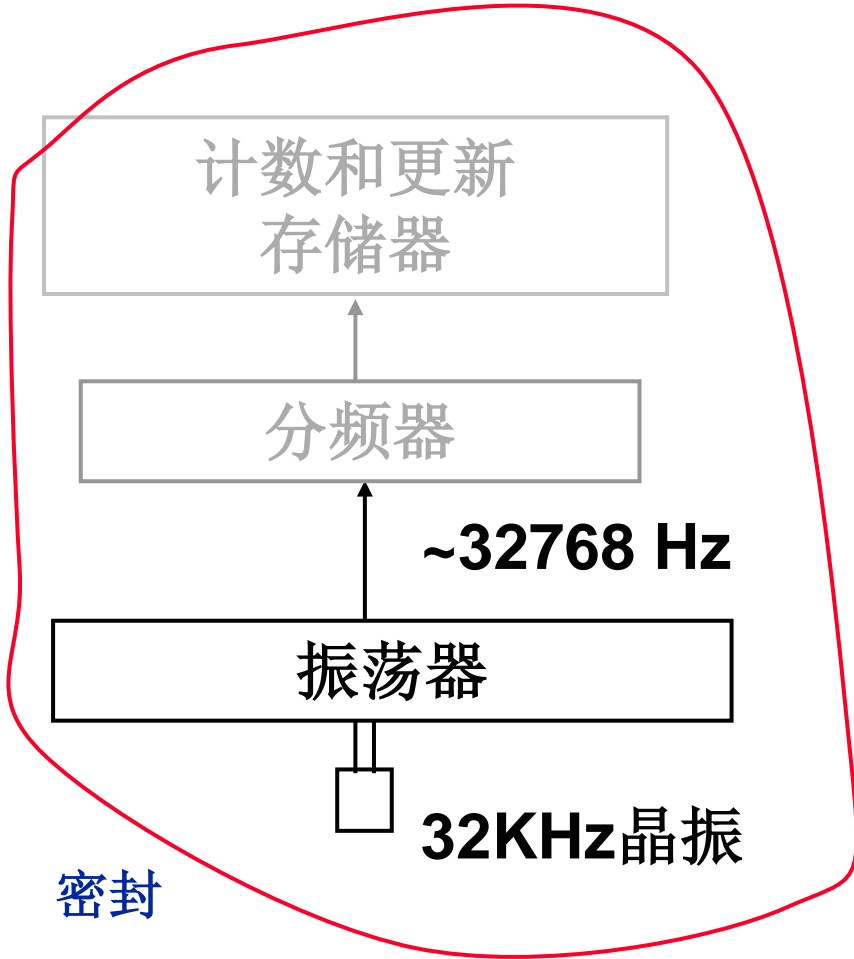
晶振数据

Package Size		DS26	DS15	DS10	
Nominal frequency	F_L	32.768	32.768	32.768	kHz
Load capacitance ¹⁾	C_L	8.2	8.2	8.2	pF
Frequency tolerance ²⁾	$\Delta F/F$	+/-20	+/-20	+/-20	ppm
	$\Delta F/F$	+/-30	+/-30	+/-30	ppm
	$\Delta F/F$	+/-100	+/-100	+/-100	ppm
Series resistance typ./max.	R_S	30 / 42	35 / 50	45 / 60	k Ω
Motional capacitance typ.	C_1	2.1	2.1	2.4	fF
Static capacitance typ.	C_0	0.9	0.9	0.9	pF
Drive level max.	P	1.0	1.0	1.0	μ W
Insulation resistance min.	R_i	500	500	500	M Ω
Aging first year max.	$\Delta F/F$	+/-3	+/-3	+/-3	ppm
Turnover temperature	T_0	25 +/-5	25 +/-5	25 +/-5	$^{\circ}$ C
Frequency vs. temperature	$\Delta F/F_0$	$-0.035 \text{ ppm}/^{\circ}\text{C} \cdot (T - T_0)^2 \text{ +/-} 10\%$			ppm

晶振驱动电平

- $I^2Rs = (1 \times 10^{-6})^2 \times 60 \times 10^3 = 0.06 \mu W$
- $I^2Rs = (3 \times 10^{-6})^2 \times 60 \times 10^3 = 0.54 \mu W$
 - 可靠的设计
 - 在最大驱动电平下工作良好
- **32768 Hz晶振的典型Rs值的数量级在30千欧**

框图



可靠的振荡器

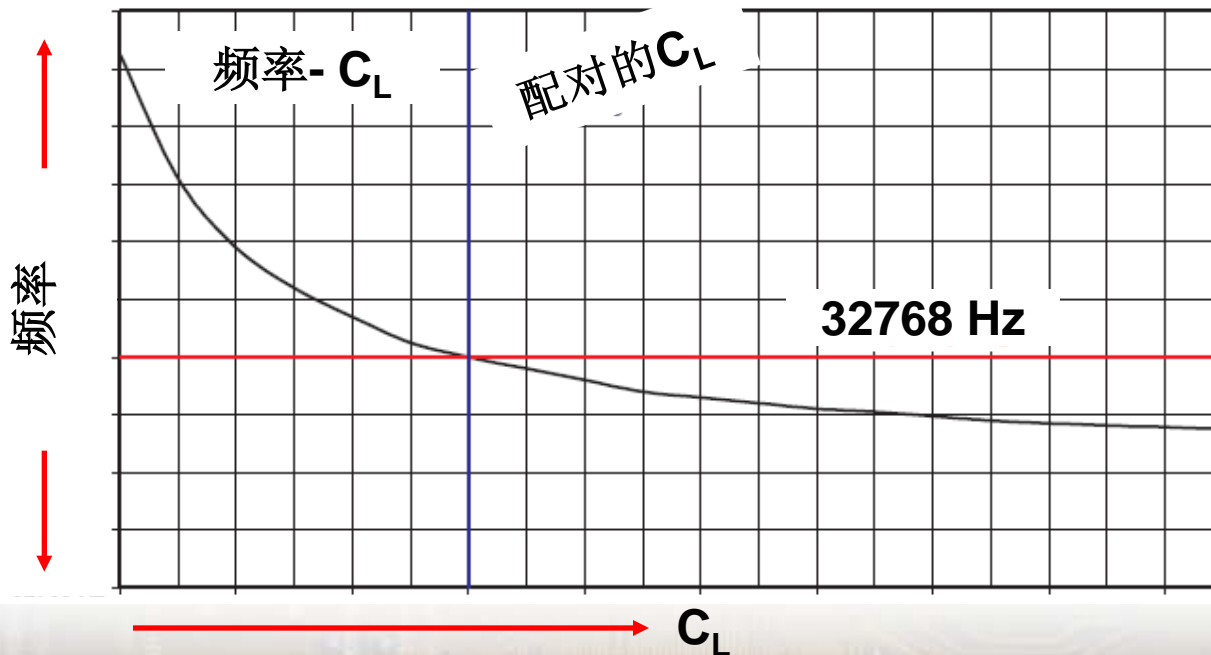
- 可使用可靠的32768 Hz晶振和PIC单片机进行设计
 - 一些**PIC**单片机具有 **32768 Hz**振荡器的高功耗和低功耗模式
 - 诸如**DS3231** 或**8025-T**等分立式**RTC**芯片集成有晶振，但售价昂贵
- 晶振和电容的电路板布线
 - 数据手册中包含有关说明
- 晶振的外壳不应焊接到印刷电路板上
- 清除焊锡的残渣
- 绝缘/防潮涂层

晶振数据手册示例

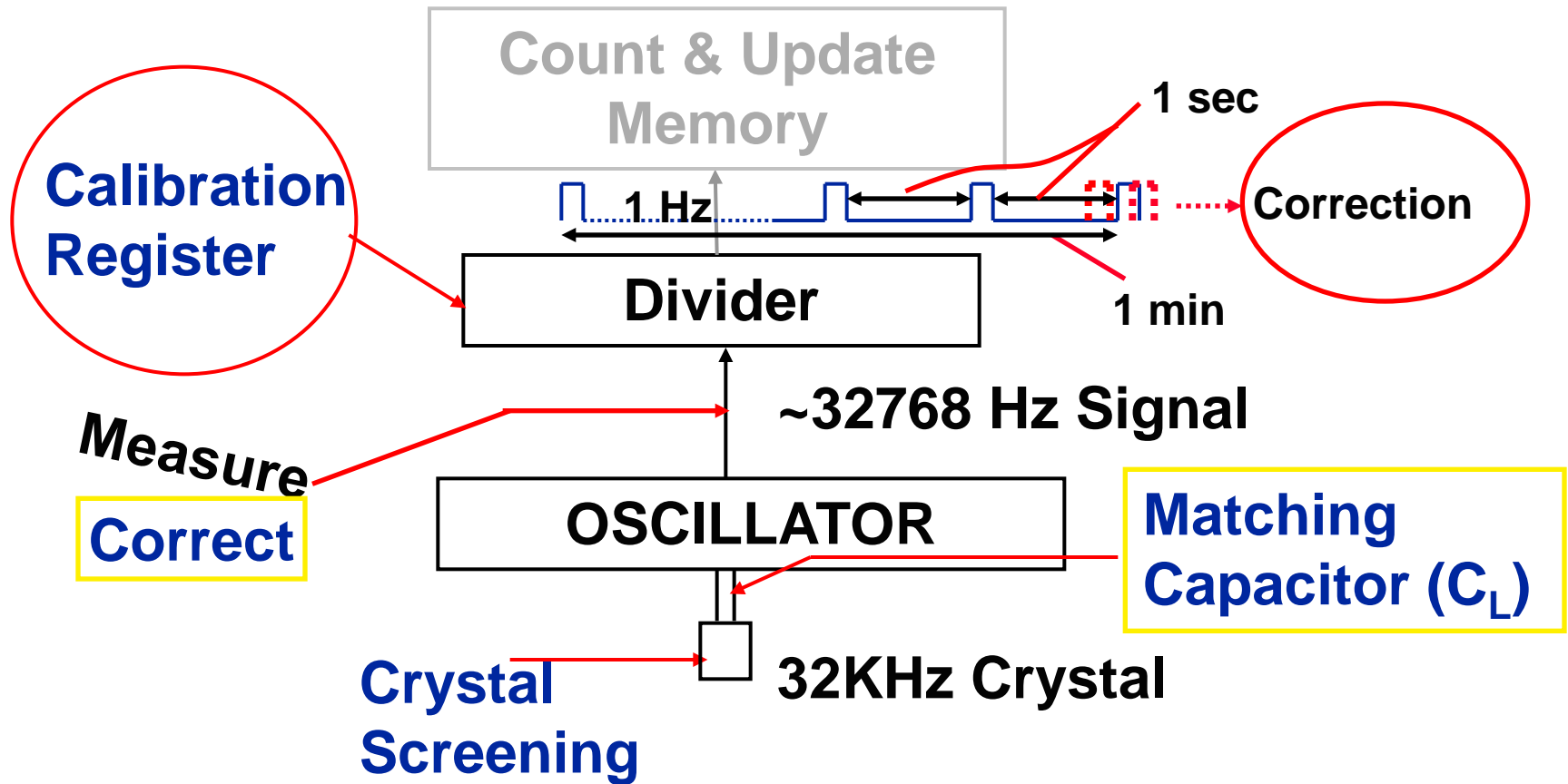
Package Size		DS26	DS15	DS10	
Nominal frequency	F_L	32.768	32.768	32.768	kHz
Load capacitance ¹⁾	C_L	8.2	8.2	8.2	pF
Frequency tolerance ²⁾	$\Delta F/F$	+/-20	+/-20	+/-20	ppm
	$\Delta F/F$	+/-30	+/-30	+/-30	ppm
	$\Delta F/F$	+/-100	+/-100	+/-100	ppm
Series resistance typ./max.	R_S	30 / 42	35 / 50	45 / 60	k Ω
Motional capacitance typ.	C_1	2.1	2.1	2.4	fF
Static capacitance typ.	C_0	0.9	0.9	0.9	pF
Drive level max.	P	1.0	1.0	1.0	μ W
Insulation resistance min.	R_i	500	500	500	M Ω
Aging first year max.	$\Delta F/F$	+/-3	+/-3	+/-3	ppm
Turnover temperature	T_0	25 +/-5	25 +/-5	25 +/-5	$^{\circ}$ C
Frequency vs. temperature	$\Delta F/F_0$	-0.035 ppm/ $^{\circ}$ C $\cdot (T - T_0)^2$ +/-10%			ppm

负载电容

- C_L 影响电流和振荡频率
- 如果 $C_L = 8.2\text{pF}$
 - $C_{L1} = C_{L2} = 2C_L = 16.4\text{pF}$
 - 寄生电容??

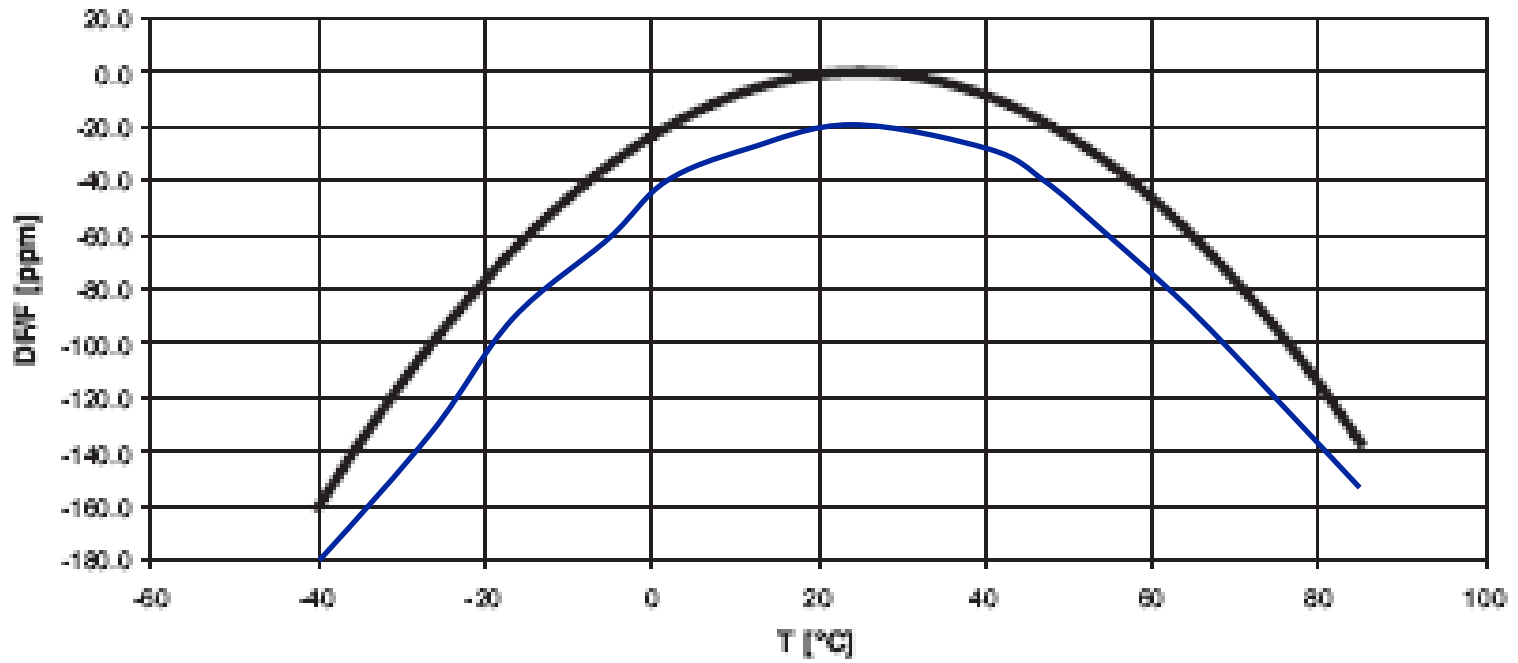


Measure and Correct



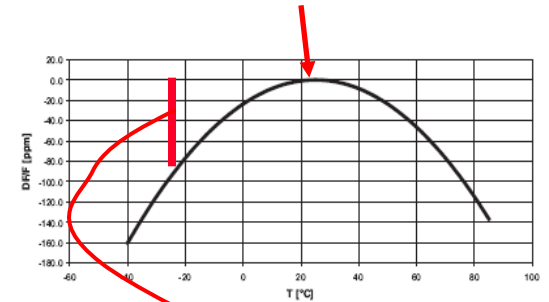
晶振数据手册示例

Package Size		DS26	DS15	DS10	
Nominal frequency	F_L	32.768	32.768	32.768	kHz
Load capacitance ¹⁾	C_L	8.2	8.2	8.2	pF
Frequency tolerance ²⁾	$\Delta F/F$	+/-20	+/-20	+/-20	ppm
	$\Delta F/F$	+/-30	+/-30	+/-30	ppm
	$\Delta F/F$	+/-100	+/-100	+/-100	ppm
Series resistance typ./max.	R_S	30 / 42	35 / 50	45 / 60	k Ω
Motional capacitance typ.	C_1	2.1	2.1	2.4	fF
Static capacitance typ.	C_0	0.9	0.9	0.9	pF
Drive level max.	P	1.0	1.0	1.0	μ W
Insulation resistance min.	R_i	500	500	500	M Ω
Aging first year max.	$\Delta F/F$	+/-3	+/-3	+/-3	ppm
Turnover temperature	T_0	25 +/-5	25 +/-5	25 +/-5	$^{\circ}$ C
Frequency vs. temperature	$\Delta F/F_0$	-0.035 ppm/ $^{\circ}$ C $\cdot (T - T_0)^2$ +/-10%			ppm



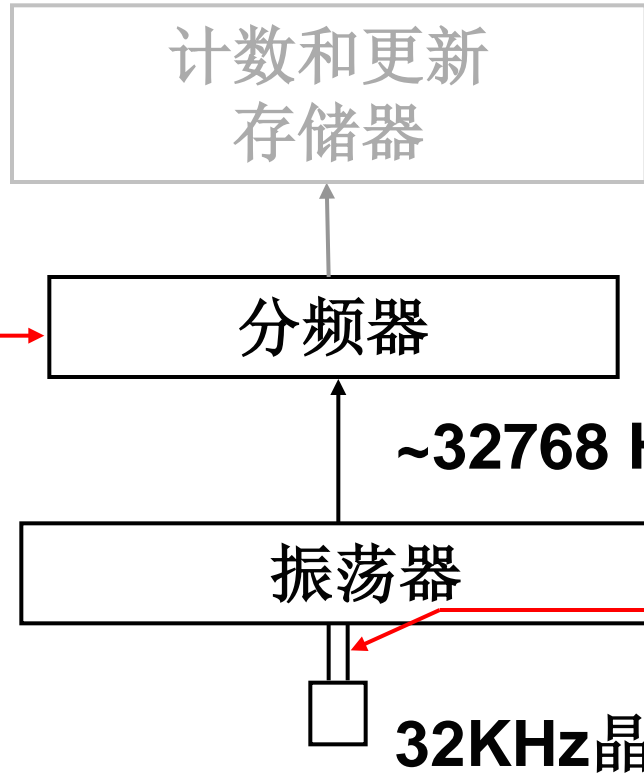
温度补偿

测量和校正



校准寄存器

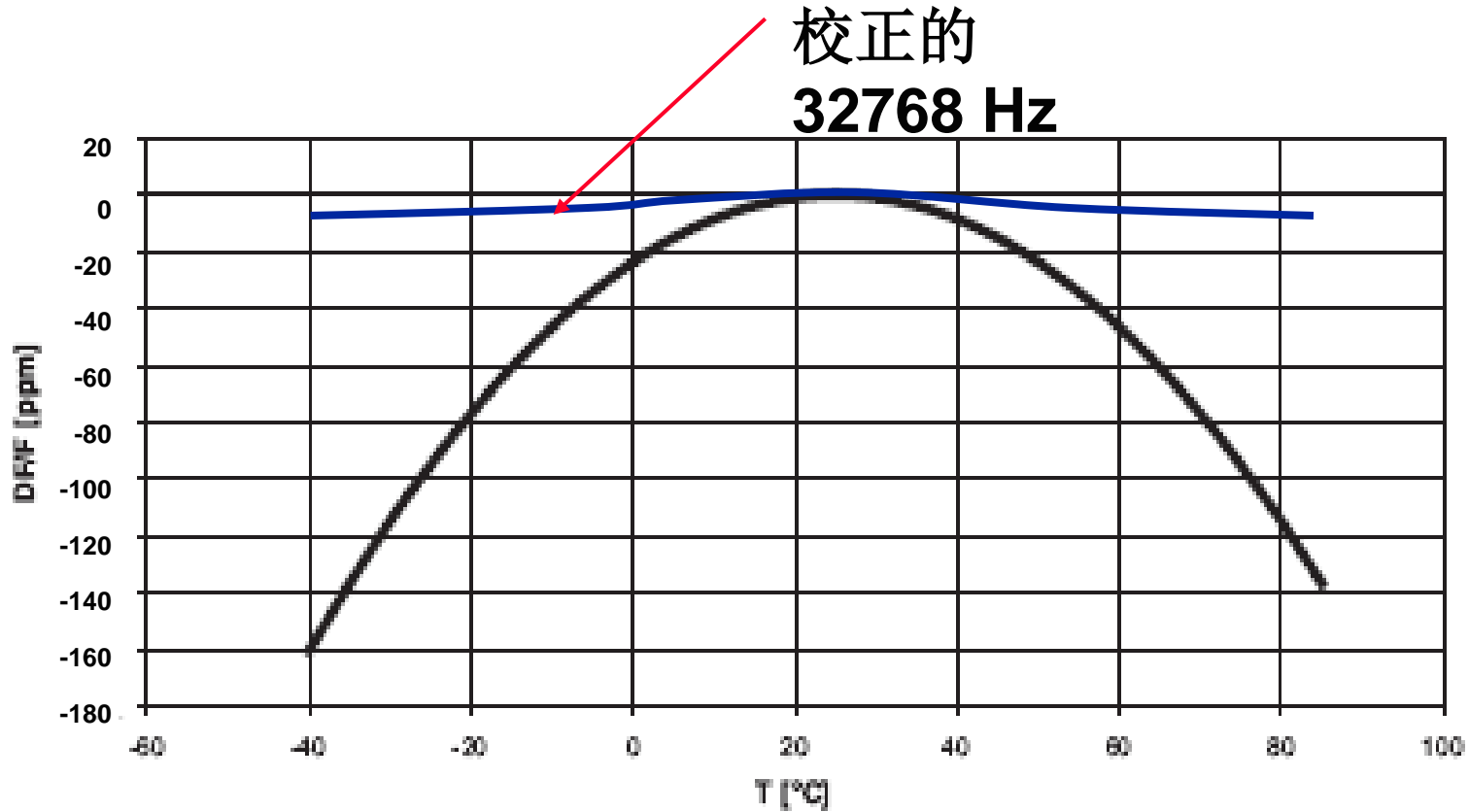
温度检测



匹配电容 (C_L)

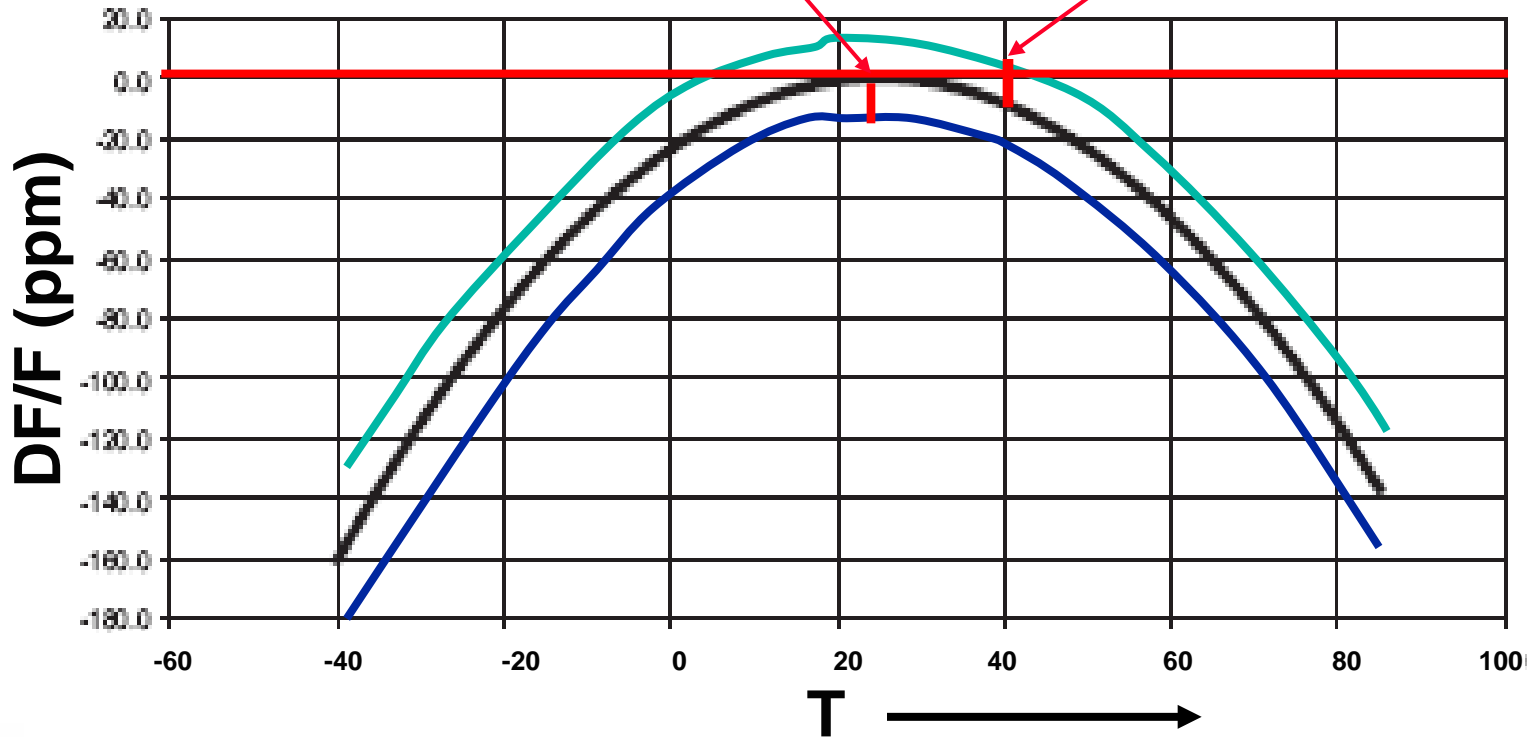
检测温度并根据曲线图改变值

由匹配的C_L校正



校准寄存器校正

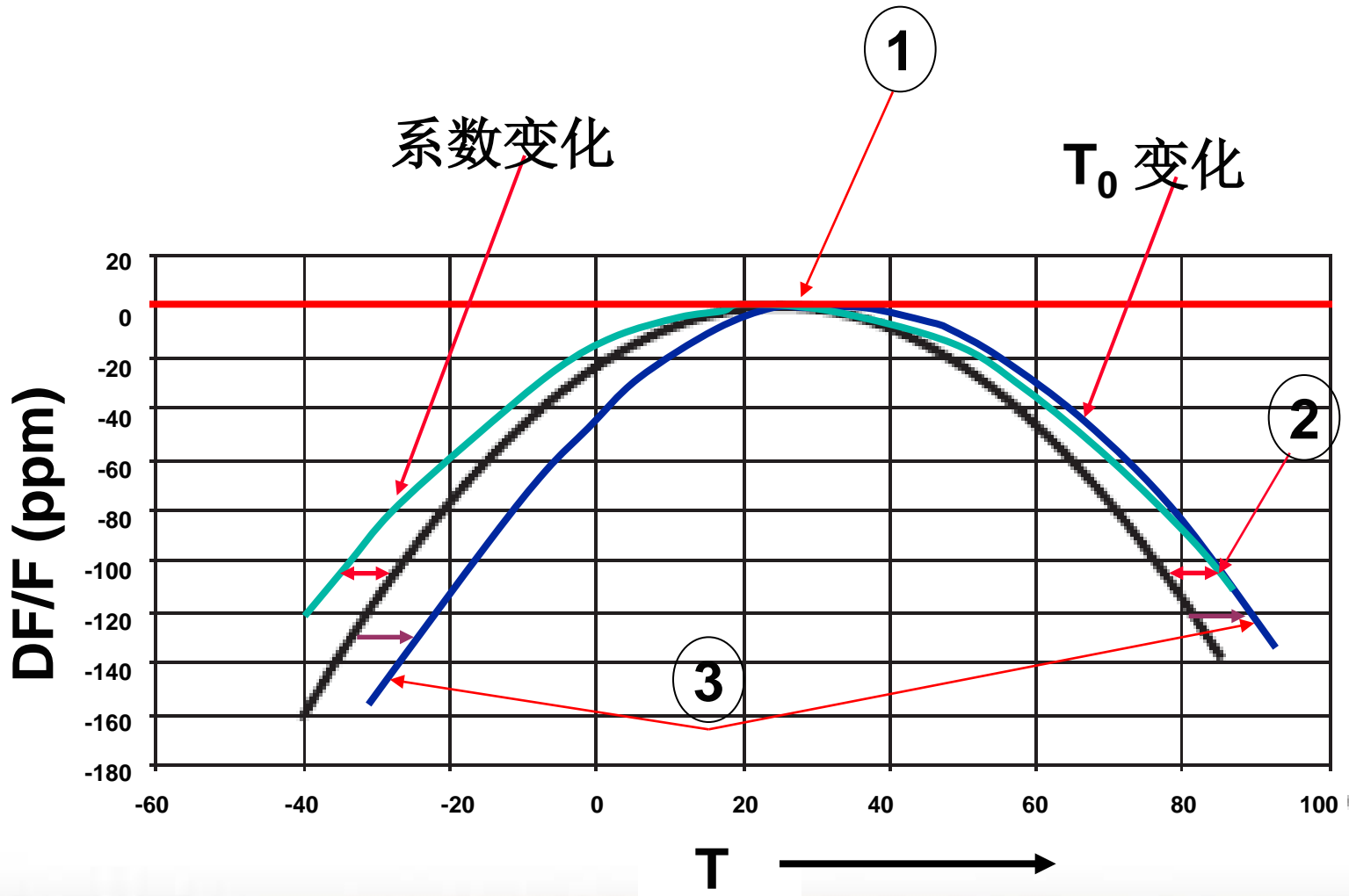
带硬件RTC
的PIC 单片机



晶振数据手册示例

Package Size		DS26	DS15	DS10	
Nominal frequency	F_L	32.768	32.768	32.768	kHz
Load capacitance ¹⁾	C_L	8.2	8.2	8.2	pF
Frequency tolerance ²⁾	$\Delta F/F$	+/-20	+/-20	+/-20	ppm
	$\Delta F/F$	+/-30	+/-30	+/-30	ppm
	$\Delta F/F$	+/-100	+/-100	+/-100	ppm
Series resistance typ./max.	R_S	30 / 42	35 / 50	45 / 60	k Ω
Motional capacitance typ.	C_1	2.1	2.1	2.4	fF
Static capacitance typ.	C_0	0.9	0.9	0.9	pF
Drive level max.	P	1.0	1.0	1.0	μ W
Insulation resistance min.	R_i	500	500	500	M Ω
Aging first year max.	$\Delta F/F$	+/-3	+/-3	+/-3	ppm
Turnover temperature	T_0	25 +/-5	25 +/-5	25 +/-5	$^{\circ}$ C
Frequency vs. temperature	$\Delta F/F_0$	-0.035 ppm/ $^{\circ}$ C $\cdot (T - T_0)^2$ +/-10%			ppm

对温度曲线的变化进行校正



晶振总结

- 振荡器驱动通过晶振的电流
- 振荡器电流是RTC电流的主要来源
- 驱动电流过小：不能产生振荡
- 驱动电流过大：加速老化，损坏晶振
- 刚能满足振动的驱动电流：
 - 环境的变化（温度和湿度）可能会改变寄生电容和泄漏电流
 - 振荡可能会暂时停止

晶振总结

(续)

- 可靠的振荡器设计
 - 布板、装配以及晶振的驱动电流应适当大于刚能产生振荡的电流
- 晶振频率变化
 - 制造工艺
 - 负载电容
 - 温度
- 精度
 - 不同的校正机制

PIC单片机的RTCC和一些比较

RTCC特性

- 时间：小时、分钟和秒
- 24小时格式（军方时间）
- 日历：星期、日期、月和年
- 可配置闹钟
- 年份范围：2000至2099
- 闰年校正
- 采用**BCD**格式可压缩固件
- 针对低功耗工作进行了优化
- 带自动调节的用户校准
- 校准范围：每个月的误差为±2.64秒
- 需求：外部32.768 kHz时钟晶振
- **RTCC**引脚的闹钟脉冲或秒时钟输出

电流比较

VBAT current, VBAT=3.6V	Battery backup, $\leq 25^{\circ}\text{C}$	2	4	μA
	$V3P3A=V3P3D=VLCD=0\text{V}$ $f_{\text{osc}} = 32\text{kHz}$ 85°C	4	12	μA

71M6511/71M6511H

DD25 (ΔI_{oscB})	RTCC + Timer1 Osc. with 32 kHz Crystal ⁽⁸⁾	0.9	4.0	μA	-10°C	$V_{\text{DD}} = 2.0\text{V}$, $V_{\text{DDCORE}} = 2.0\text{V}^{(4)}$	32 kHz on Timer1 ⁽⁸⁾
		1.0	4.5	μA	$+25^{\circ}\text{C}$		
		1.1	4.5	μA	$+85^{\circ}\text{C}$		
		1.1	4.5	μA	-10°C	$V_{\text{DD}} = 2.5\text{V}$, $V_{\text{DDCORE}} = 2.5\text{V}^{(4)}$	32 kHz on Timer1 ⁽⁸⁾
		1.2	5.0	μA	$+25^{\circ}\text{C}$		
		1.2	5.0	μA	$+85^{\circ}\text{C}$		
		1.6	6.5	μA	-10°C	$V_{\text{DD}} = 3.3\text{V}$	32 kHz on Timer1 ⁽⁸⁾
		1.6	6.5	μA	$+25^{\circ}\text{C}$		
2.1	8.0	μA	$+85^{\circ}\text{C}$				

PIC18F87J90

$f_{\text{SUB}} = 32.768\text{ kHz}^{\text{Note 5}}$, $V_{\text{DD}} = 5.0\text{ V}$	Resonator connection	2.4	22	μA
--	----------------------	-----	----	---------------

5. Not including the operating current of the X1 oscillation,

78K0/LF3

使用校准寄存器实现校正的机制

- 根据误差会加上或减去一个完整的Teridian秒（根据软件要求） **Teridian**
- 每分钟加上或减去少量计数值（一秒的极小分支）（根据软件要求由硬件定期完成） **带RTCC的PIC单片机**
- 每20秒或每分钟加上或减去少量计数值（一秒的极小分支）（根据软件要求由硬件定期完成） **NEC**
- 每两分钟加上或减去少量计数值（一秒的极小分支）（根据软件要求由硬件定期完成） **Renesas**

RTCC校准

REGISTER 14-2: RTCCAL: RTCC CALIBRATION REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0
bit 7							bit 0

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

bit 7-0

CAL<7:0>: RTC Drift Calibration bits

01111111 = Maximum positive adjustment; adds 508 RTC clock pulses every minute

·
·
·

00000001 = Minimum positive adjustment; adds four RTC clock pulses every minute

00000000 = No adjustment

11111111 = Minimum negative adjustment; subtracts four RTC clock pulses every minute

·
·
·

10000000 = Maximum negative adjustment; subtracts 512 RTC clock pulses every minute

校准范围（硬件校正）

- $32768 \times 60 = 1966080$ （对于0ppm晶振）
- $32768 \times 59 + 32768 \pm 4$ （最小计数）
 - $(4 / 1966080) \times 1000000 = 2.03 \text{ppm}$ （分辨率）
 - **1ppm** 是校正的最小单位
- $32768 \times 59 + 32768 \pm 508$ （校正的最大计数）
 - $(508 / 1966080) \times 1000000 = 258.38 \text{ppm}$ （最大校准）
- NEC产品的可校正范围/分辨率
 - **189.2ppm/3.05**（20秒间隔）
 - **63.1ppm/1.02ppm**（1分钟间隔）
- Renesas产品的可校正范围/分辨率
 - $(8 / (1966080 \times 2)) \times 1000000 = 2.03 \text{ppm}$
 - $((8 \times 63) / (1966080 \times 2)) \times 1000000 = 128 \text{ppm}$

需要什么

- 温度变化
 - **0.035 ppm / (T-T₀)²**
 - **T₀为25, T=70, 变化在 71ppm左右**
- 漂移在-10至-100ppm
- 漂移和温度变化的增加有上限
- 最小分辨率：越小越好

总结

- PIC单片机的RTCC外设
 - 电流消耗
 - 校准范围/分辨率
- 与其他器件的一些比较

在PIC单片机上实现温度补偿 硬件RTC

为何采用硬件RTC

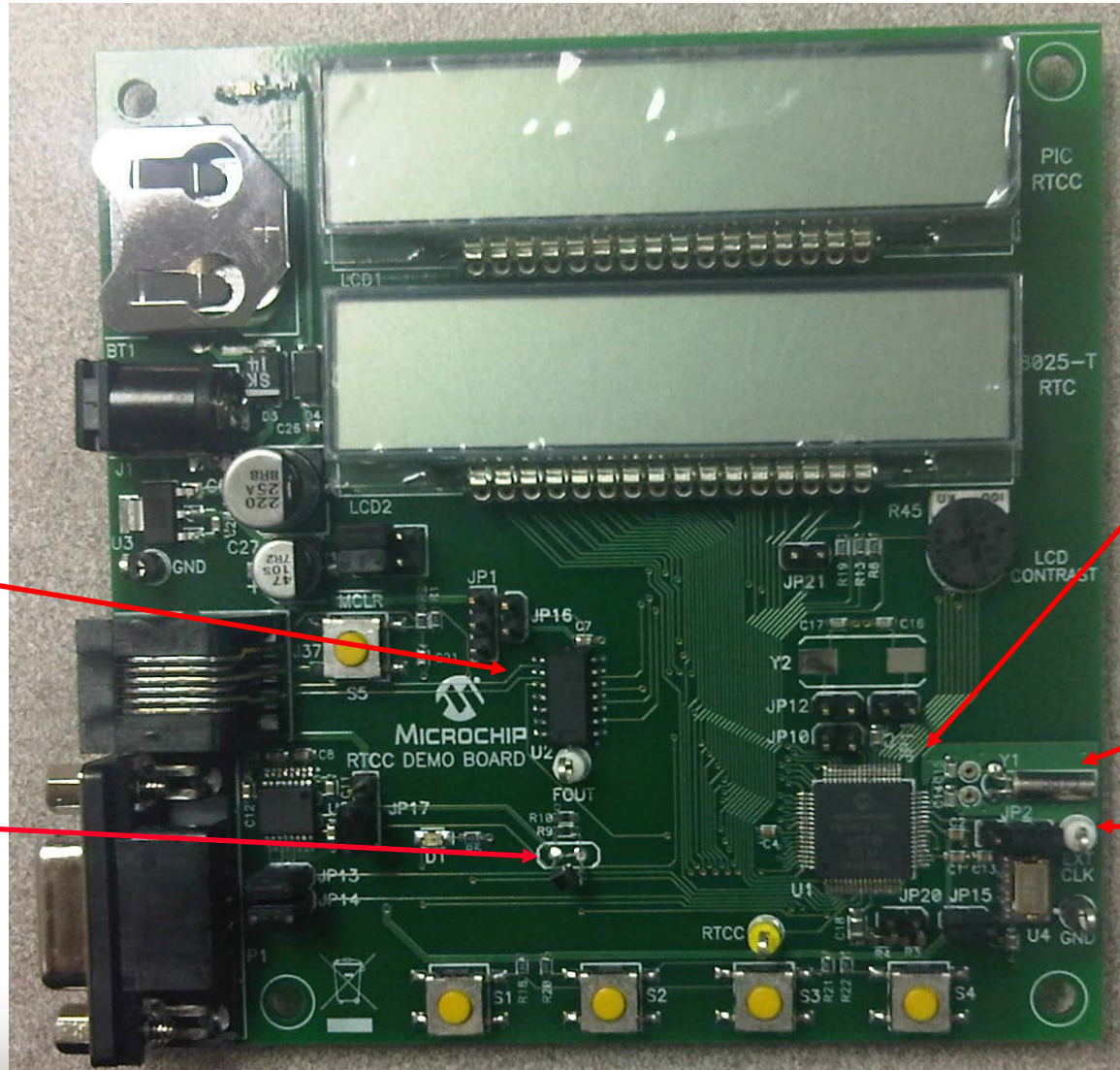
□ 软件RTC

- 1 Hz中断，在in_isr软件中更新时间 and 数据等
- 软件开销
- 复位引起误差

□ 硬件RTC

- 1Hz信号提供给硬件计数器
- 即便我们长时间使用任何种类的复位脉冲，该计数器都将计数
 - 硬件校准非常简单
 - PIC单片机的RTCC外设

RTCC演示板



8025-T
硬件 RTC

NTC
热敏电阻

18F67J90
(硬件RTC)

时钟晶振

参考脉冲

使用的资源

□ RTCC

- RTCC外设和Timer1振荡器

□ 初始校准

- Timer0用于计数参考脉冲

- Timer0仅在RTCC启动时用于初始校准

□ 温度补偿

- ADC通道1

初始校准

□ 确定初始校准值

- Timer0对参考脉冲计数一段特定的时间（在演示软件中为5秒）
 - 启动时校准一次（除了这个最初的要求，**Timer0** 在其他时间空闲）
 - 启动**RTCC**和**Timer0**
 - 累加**Timer0**中的参考脉冲计数值数秒，直到**RTCCIF**置（本演示板为5秒）
- 通过处理计算PPM
 - 累加的参考脉冲计数和精确的参考脉冲计数之间的差值
 - 精确的参考脉冲计数 = 参考脉冲频率 x 秒数（本演示板为5秒）

校准/补偿

- 如果PPM误差大于 $|1| \text{ ppm}$
 - 修改RTCCAL寄存器的值并使用正确的符号位
 - RTCCAL寄存器的数值改变1即可实现约 2ppm的校正

温度补偿

```
void DoRTCTempCal(void)
{
    signed char TempCalValue;

    TempCalValue = GetTemperatureCal(); //returns compensation value as a
                                        //function of temperature
    TempCalValue = TempCalValue + InitialOffset;

    //Ready to use the new RTCCAL value, but write to it during a safe
    //opportunity.
    if(RTCCFGbits.RTCEN == 1)
    {
        while(RTCCFGbits.RTCSYNC); //Want to be careful when updating
                                    //registers while the RTCC is running.
    }
    RTCCAL = TempCalValue; //Update the RTCCAL register.
                           //Next minute uses this new value.
}
```

□ 执行RTC温度校准

– 周期

□ 推荐**10秒**，因为温度变化并不非常快

– 定期测量温度

□ **NTC连接到ADC通道1**

– 如果需要对校准值进行补偿

– 休眠期间不执行RTC温度校准

□ 休眠时**Timer1和RTCC使能，ADC关断**

– 单片机在超过99%的时间内处于休眠模式

□ 温度补偿造成电流消耗增加的程度可以忽略不计

```
signed char GetTemperatureCal(void)
```

```
{
```

```
    unsigned char i;
```

```
    //Enable the NTC and perform a temperature measurement
```

```
    NTCEnablePinTris = 0;
```

```
    //Allows current through the NTC.
```

```
    ADCON0bits.CHS = NTCADCChannel; //Measure the voltage
```

```
    for(i = 0; i < 10; i++);
```

```
    //Small delay. NTC settled.
```

```
    ADCON0bits.GO = 1;
```

```
    //It was right justified before this.
```

```
    while(ADCON0bits.GO);
```

```
    NTCEnablePinTris = 1;
```

```
    //No longer need NTC powered.
```

```
    BusyModifyingADCTempValue = 0xFF; //True, If interrupt occurs
```

```
    ADCTempValue = ADRES;
```

```
    BusyModifyingADCTempValue = 0x00; //False
```

```
    return CalTable[ADCTempValue];
```

```
    //Get the value from the lookup table  
    //in flash memory.
```

```
}
```



```
const rom signed char CalTable[] =
```

```
{
```

```
    0x7F, // Temperature: -4323  ADC: 0x0
```

```
.....  
    0x16, // Temperature: -10.7  ADC: 0x17B
```

```
    0x15, // Temperature: -10.2  ADC: 0x17C
```

```
.....  
    0x0, // Temperature: 25.1  ADC: 0x205
```

```
    0x0, // Temperature: 25.2  ADC: 0x206
```

```
.....  
    0x3, // Temperature: 38.1  ADC: 0x258
```

```
.....  
    0x23, // Temperature: 70.2  ADC: 0x322
```

```
    0x23, // Temperature: 70.4  ADC: 0x323
```

```
    0x24, // Temperature: 70.6  ADC: 0x324
```

```
.....  
    0x7F // Temperature: 461.3  ADC: 0x3FF
```

```
};
```

□ TempCalRTCC.xls

- 该文件是针对特定的 **NTC**（**Panasonic**的**ERT-D2FHL103S**），晶振温度系数为**0.035**，且转换温度为**25 °C**的情况创建的。
- 使用不同制造商的不同晶振进行测试，表明**25 °C**时的精度为**+/-5ppm**。
- **-10至+60°C**，
 - 10个晶振中有8个的精度为**+/-10ppm**
 - 10个晶振中有2个的精度为**+/-20ppm**
- 可在不同温度下对一批晶振的频率进行测试
 - 可找到更精确的系数和转换温度
- 可使用不同的晶振和**NTC**参数修改该**xl**文件
- 温度变化时的精度取决于查找表的精度

商标

- Microchip的名称和徽标组合、Microchip徽标、dsPIC、KeeLoq、KeeLoq徽标、MPLAB、PIC、PICmicro、PICSTART、rfPIC和UNI/O均为Microchip Technology Inc.在美国和其他国家或地区的注册商标。
- FilterLab、Hampshire、HI-TECH C、Linear Active Thermistor、MXDEV、MXLAB、SEEVAL和The Embedded Control Solutions Company均为Microchip Technology Inc.在美国的注册商标。
- Analog-for-the-Digital Age、Application Maestro、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、HI-TIDE、In-Circuit Serial Programming、ICSP、Mindi、MiWi、MPASM、MPLAB Certified徽标、MPLIB、MPLINK、mTouch、Octopus、Omniscient Code Generation、PICC、PICC-18、PICDEM、PICDEM.net、PICkit、PICtail、PIC32徽标、REAL ICE、rfLAB、Select Mode、Total Endurance、TSHARC、UniWinDriver、WiperLock和ZENA均为Microchip Technology Inc.在美国和其他国家或地区的商标。
- SQTP是Microchip Technology Inc.在美国的服务标记。
- 在此提及的所有其他商标均为各持有公司所有。
- © 2009, Microchip Technology Inc.版权所有。