

通过 L-C 传感器和 Kinetis L MCU 实现低功耗旋转感测

Freescale Kinetis L 低功耗 MCU 系列可提供低功耗特性，从而可以让 CPU 处于众多低功耗模式中的一个，同时其他低功耗外设处于异步工作状态中。

LPUART、SPI、I²C、ADC、DAC、LPTIMER 和 DMA 支持低功耗模式操作，并且不需要唤醒内核，从而为设计人员和工程师提供开发各种低功耗感测器件的便利。

1 简介

本应用笔记介绍了如何基于电感(LC)传感器构建低功耗编码器。目标是构建超低功耗 / 低转速设计，可用于浆轮流量计进行浆轮转速测量。

用于旋转运动的 LC 传感器是混合流量计设计中的常见应用。流量计浆轮旋转通过 LC 传感器进行测量，后者可检测与流量计浆轮相连的旋转盘上是否存在导电材料。电子浆轮流量计（混合流量计）由电池供电，低功耗设计是设计的关键部分。

内容

1. 简介	1
2. LC 传感器原理以及如何构建正交编码器	2
3. 设置 Kinetis L 外设，以便进行低功耗编码器测量	3
4. 软件	11
5. 结语	12
附录 A. 电路板示意图	14

2 LC 传感器原理以及如何构建正交编码器

编码器的主要原理是通过 LC 电路对导电材料进行非接触式感应。如果 LC 电路受激发，它将开始振荡。振荡将以某个速率衰减，该速率由寄生电感电阻率值给定的衰减以及附近导电材料中产生的涡电流导致的阻尼系数确定。

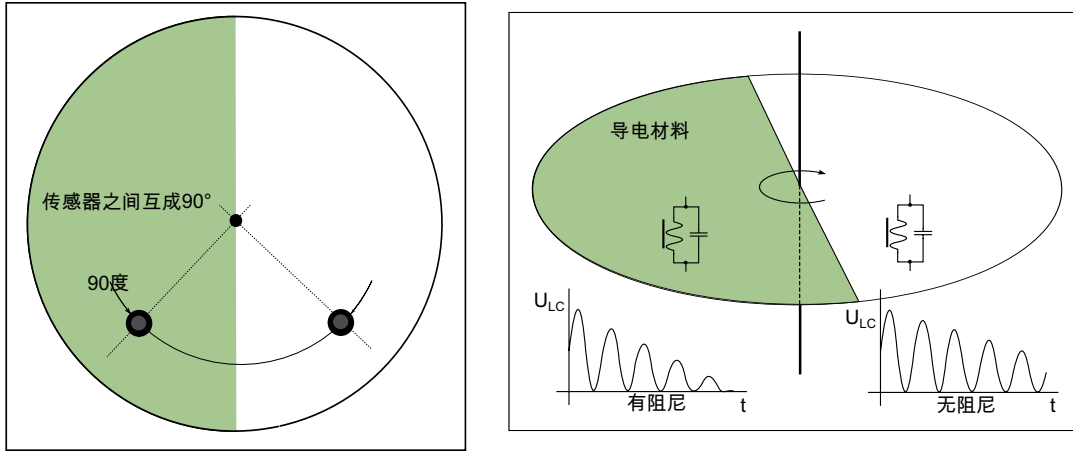


图 1. LC 传感器工作原理

¹ 旋转盘附近有两个线圈。旋转盘被导电材料部分覆盖，可构成正交编码器。

LC 传感器将定期受激发（采样率）并且会测量 LC 电路振荡阻尼系数，以确定传感器附近是否存在导电材料。

两个线圈传感器彼此呈 90 度放置在旋转盘附近，其表面的一半覆盖有导电材料，如图 1 中所示，可构成正交编码器。

正交编码器可产生如图 2 中所示的信号。

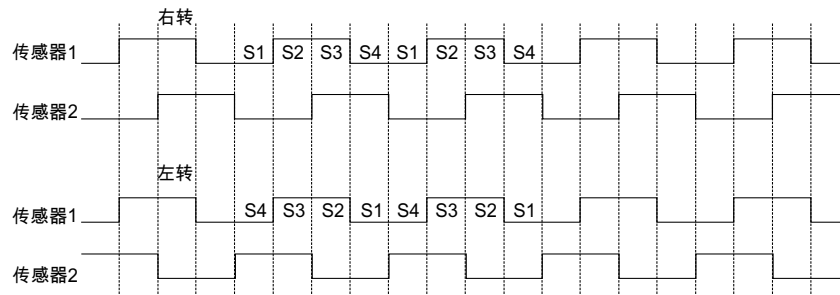


图 2. 在连续模式下右转 / 左转时，来自传感器的正交编码器信号

图 2 是编码器产生的信号图。可确定旋转以及旋转方向。图中的信号为连续信号，而 LC 传感器仅允许以给定的采样率对这些信号进行采样。

为了实现准确、安全的旋转检测，需要选择正确的采样率。如果两个传感器彼此呈 90 度，则使用以下公式计算最小采样率。

$$fs = 2 (sensors) * 2 (disc parts) * 2(nuynquist) * rot_{max}$$

$$rot_{max} \dots rotation \text{ disc maximal rotation } \left[\frac{turns}{sec} \right]$$

随后将处理由线圈产生的信号样本，以计算旋转盘的圈数。

3 设置 Kinetis L 外设，以便进行低功耗编码器测量

Kinetis L 系列微控制器具有多组外设和功能，可以在无需 CPU 干预的情况下感测和处理模拟信号。低功耗外设与 DMA 的独特功能巧妙组合，构成一个用于感测编码器的独立扫描引擎。

本章节介绍 LC 传感器的连接、外设的配置以及测量技巧。

3.1 LC 传感器示意图以及与 Kinetis L 外设连接

如前面所述，混合流量计采用电池供电，显然功耗非常重要，因此传感器的连接方式决定最终的功耗。为了在感测 LC 传感器的同时实现低功耗，仅当 CPU 处于超低功耗模式时外设才工作。图 3 所示为 LC 传感器的连接框图。

- 定时器 TPM0（直接通过 32 kHz 晶振提供时钟）可在 CPU 处于其中一种节电模式下继续运行。定时器继续为时钟计数，并且如果处于 PWM 模式，则定时器的输出可通过引脚输出。
- 定时器 TMP0 和 TMP1 为传感器生成控制信号。
- TMP0CH1 信号用于控制传感器的激发。定时器以采样频率生成 30uS 脉冲。
- 如果使用多个传感器，则 TPM1 CH0 和 TPM1 CH1 信号可使能具体某一个正在工作的传感器。对于每个测量序列，只会测量一个传感器。分别使能某一个传感器有助于节能。
- TPM0CH1 脉冲结束时，比较器 CMP 读取实际传感器上的电压。比较器的多路复用器从实际测得的传感器中选择信号。

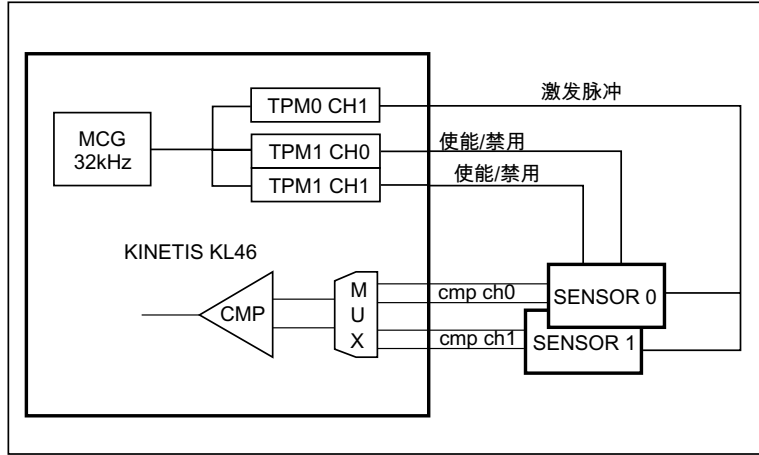


图 3. LC 传感器连接框图

¹ 定时器 TPM0 用于激发传感器并为其供电，而定时器 TPM1 则用于使能 / 禁用实际传感器。

我们很清楚传感器如何连接到控制器，但到底什么是传感器呢？我们将一个电子感应线圈称为“传感器”。

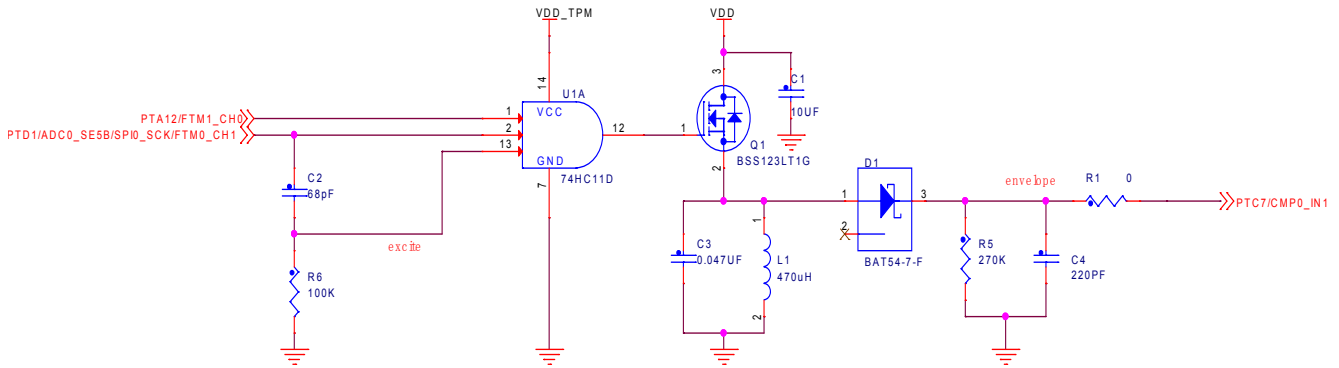


图 4. 单 LC 传感器示意图

L1-C2 谐振电路通过 Q1 NMOS 晶体管供电。晶体管门极由 AND 逻辑门输出引脚 12 控制。AND 逻辑门具有三个输入，分别由以下控制：

- 第一个 AND 门输入由 TPM1CH0 输出控制。TPM1CH0 定时器使能实际传感器。如果传感器已使能，则定时器 TPM1CH0 提供高电平逻辑输出。
- 第二个 AND 门输入由 TPM0CH1 输出控制。定时器 TPM0CH1 以采样频率生成 30us 的激发脉冲（逻辑高电平）。
- 第三个 AND 门“激发”输入由 C1-R2 延迟单元控制。由于时钟源来自 32kHz 低功耗晶振，因此，定时器可生成的最小时间步阶约为 30us，但对于 L-C 元件来说时间过长而无法充电。使用较短的时间来激发 LC 电路进而节省流入线圈的能量，这一方法非常有用。R2-C1 组合的时间常数控制 L-C 回路激发的时间。激发时间约为 $t = R2 * C1$ ，并且已设为

6us。当电容 C1 充电至 $VDD/2$ 时，AND 门输出切换为逻辑零，Q1 晶体管闭合且 L-C 电路开始振荡。

传感器的主要部件为 L1 线圈，它是一个感应元件。线圈 L1 通过 C2 电容谐振。线圈和电容参数对于实现适当的灵敏度非常重要。线圈的等效串联电阻和振荡频率将影响阻尼系数和阻尼时间。此外，线圈的物理尺寸影响流入线圈的电流，并且对总电流消耗有影响。

LC 回路上的电压会通过 D1-R3-C3 形成的包络检测器。比较器将比较来自包络电压检测器的输出电压和比较器 DAC 上设置的阈值电压。

3.2 传感器测量时序图

图 5 所示为详细的传感器控制时序图。TPM1CH0 和 TPM1CH1 信号相互逆转，以生成传感器 0 和传感器 1 的备用使能 / 禁用信号。TPM0CH1 定时器同时生成两个传感器的激发信号。

RC 时序元件通过 TPM0CH1 信号生成短脉冲，并驱动 Q1 晶体管门极为 LC 感应元件充电。只为已使能的传感器充电，然后由比较器进行测量。图 5 中的传感器 0 和传感器 1 信号显示 LC 感应元件上的电压黑线。红色信号线显示包络检测器 D1-R3-C3 上的电压。此信号将与 TPM0 计数器中断溢出事件上的预设电压阈值比较。

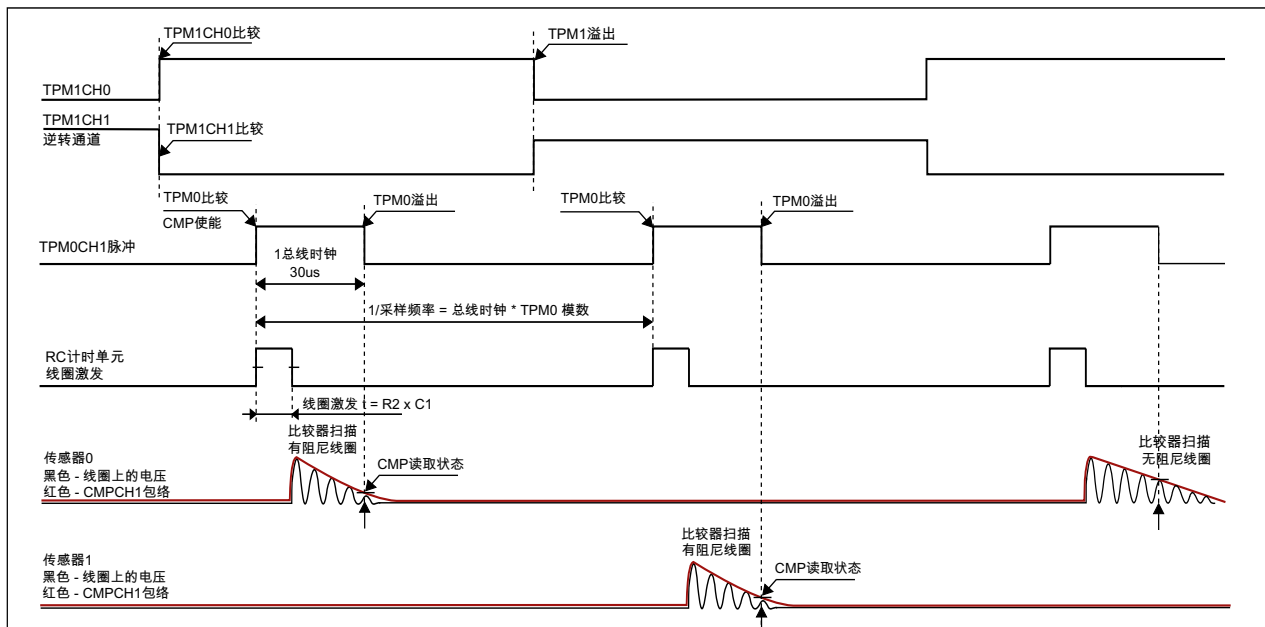


图 5. 传感器控制信号时序图

当 32 kHz 总线时钟产生一个粗略的时间分辨率时，请仔细选择 LC 回路谐振频率。线圈激发与包络电压测量之间的时间约为 30us，阻尼谐振（通过靠近线圈的金属实现）的持续时间必须略大于 30us。47nF 电容和 20uH 线圈产生 150 kHz 谐振频率，相当于每个总线时钟约为 5 次振荡 (32 kHz)。这足以实现良好的灵敏度以确定阻尼系数。

3.3 如何控制比较器

在前面章节中，我们提到过比较器会将包络电压与 TMP0 溢出事件上的预设电压比较。那么，它是如何被驱动以及如何处理比较器输出状态的呢？

传感器采样机制的核心是无需 CPU 干预、在 VLPS 模式下运行的三个直接存储器访问 (DMA) 外设通道。DMA 外设可以在 RAM 和任意外设寄存器之间移动数据。该技术允许 DMA 通过设置其控制寄存器来控制外设。在相反的传输方向，DMA 可读取外设状态寄存器，并将内容存储在 RAM 中。

DMA 传输在触发模式下进行，这意味着每次触发 DMA 通道时执行单次 DMA 传输（1 个或 2 个字节）。

定时器 TPM0 在 VLPS 模式下运行，并且 TPM 也可触发 DMA 传输。正确设置 TPM0 定时器和相关的定时器通道 TPM0CH1 后，两者均可用作相关 DMA 通道的触发事件，并且可形成状态机。状态机通过 CMP 比较器输入执行扫描，并将结果存储在 RAM 存储器。预定义扫描周期数之后，CPU 唤醒并处理 RAM 中的结果。

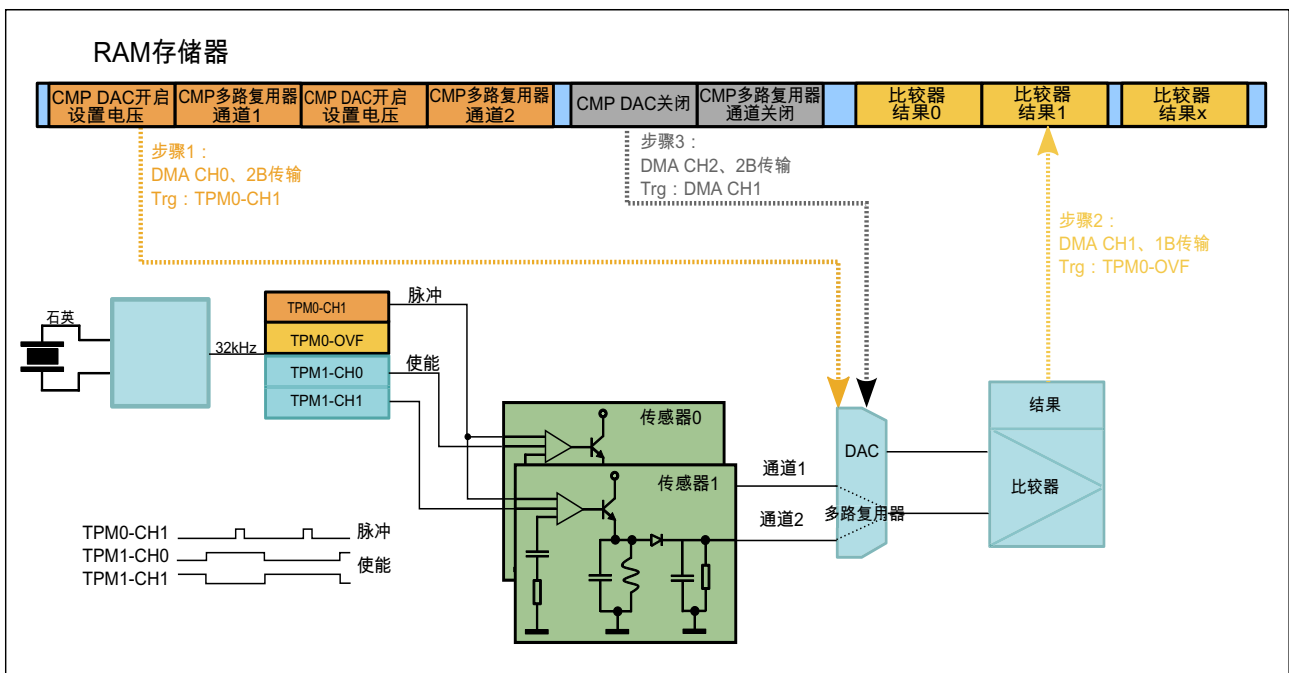


图 6. 设置 DMA 通道以构建扫描机

图 6 为连接外设和 DMA 通道的框图。CPU 处于 VLPS 模式，时钟源由石英晶振生成，因此，总线时钟频率为 32,768 kHz。TPM0 和 TPM1 定时器通过总线时钟计时。TPM0 定时器用于触发 DMA 以及激发 LC 传感器。TPM1 定时器用于使能 / 禁用实际扫描的传感器。

TPM0 模数寄存器值定义定时器溢出以及采样率。如果总线时钟频率为 32 kHz 且采样率为 320 Hz，则必须将 TPM 模数寄存器值设为 32k/320。

常见浆轮流量计在最大流量时的旋转速度约为 40 转 / 秒，最小采样率为 320 个样本 / 秒（每个传感器 180 个样本 / 秒）。

扫描状态机重复执行以下步骤：

- 步骤 1: TPM0 CH1 比较事件触发关联的 DMA CH0 通道传输。将 DMA CH0 设为将两个字节传输至 CMP DAC 寄存器和 CMP MUX 寄存器。CMP DAC 寄存器包含用于使能 DAC 和设置输出电压水平的控制位。因此，第一个传输的字节将使能 CMP DAC 并设置比较阈值电压。DMA 传输的第二个字节将设置 CMP MUX 寄存器，以将比较器连接至传感器 1 和 CMP DAC 输出。CMP 多路复用器通道设置还控制比较器打开 / 关闭状态。如果 CMP 多路复用器通道设为非有效状态（例如比较器正负输入均接至相同引脚），则比较器关闭。正确选择多路复用输入将会再次打开比较器。TPM CH0 逻辑状态将传播至输出引脚并生成 LC 回路激发脉冲。
- 步骤 2: TPM0 溢出事件触发 DMA CH1，将 CMP 输出状态寄存器值传输至 RAM。两个传感器的结果通过以下形式存储在单个数组中：*sensor1result*、*sensor2result*、*sensor1result*、*sensor2result*、*sensor1result*、*sensor2result*、...
- 步骤 3: DMA CH2 链接至 DMA CH1，并且在 DMA CH1 完成传输之后被触发。再次将 DMA CH2 设为将两个字节传输至 CMP DAC 寄存器和 CMP MUX 寄存器。DMA CH2 将来自 RAM 的第一个字节传输至 CMP DAC 寄存器并将其关闭。第二个字节通过设为相同引脚的两个输入传输至 CMP MUX 寄存器并关闭比较器。

完成预定义的 DMA 周期数后，CPU 唤醒并且处理存储在 RAM 中的结果。在 RAM 存储器中，字节数组填充传感器状态的对应值。

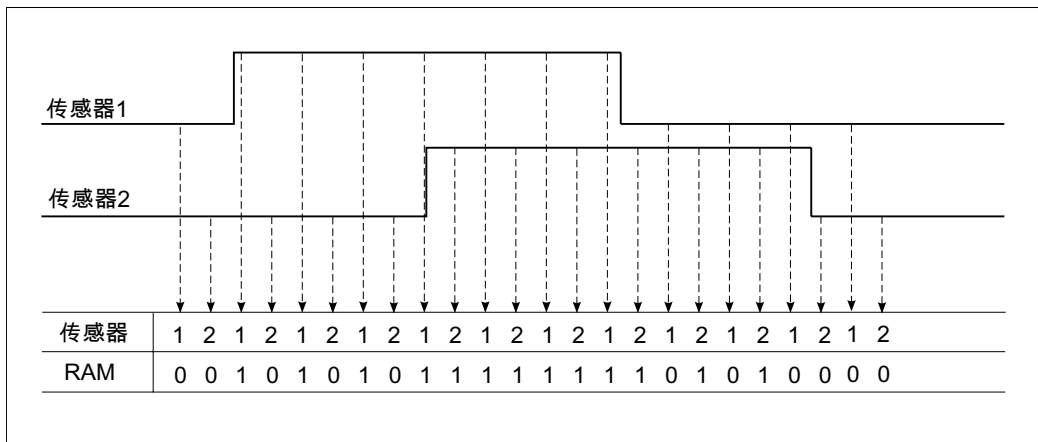


图 7. 比较器输出状态存储在 RAM 的数组中

¹ 传感器状态以及相应 RAM 数组内容如此图所示。

3.4 用于实施编码器的硬件

编码器框架在 Kinetis MKL46Z256 处理器上实现。为了进行开发，使用的是塔式系统。开发集包含处理器板 TWR-KL46Z48M、主次侧板（TWR-ELEV-PRIMARY、TWR-ELEV-PRIMARY）以及外设模块 TWR-FLOW-LC。请参见下图中的各板。



图 8. 用于基于 LC 传感器编码器的塔式系统集

TWR-FLOW-LC 板设计用于测试编码器功能。此板装配有两个 LC 传感器和一个主轴，与编码器磁盘相连。在图 9 中，使用的是铝制浆轮，而不是编码器磁盘。该板配备多个跳线和测试引脚。

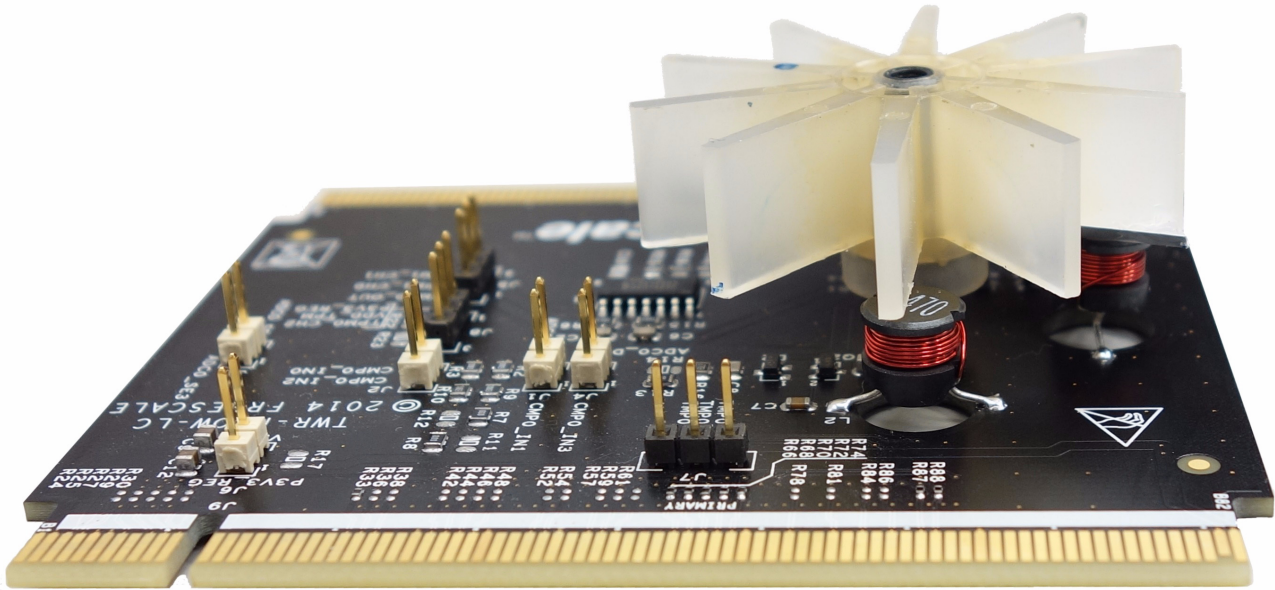


图 9. 带浆轮的 TWR-FLOW-LC 板作为编码器

板上具有多个跳线和接头连接器，用作测试点以及用于配置线圈和相关逻辑的电源。有关详细信息，请参阅附录 A “电路板示意图”。

电路板详细描述：

- J1：用于观察传感器 1 包络检测器输出的测试点。
- J1[Pin1] 连接至 MKL46 比较器 CMP0_IN1。
- J1[Pin2] 连接至模数转换器 ADC_SE0
- J2、J3：板上有两个分压器，用于创建与传感器电压比较的阈值电压从包络检测器输出。
- J2[Pin1]：R2-R9 分压器，用于与传感器 1 输出电压比较 – CMP0_IN 输入
- J2[Pin2]：R3-R10 分压器，用于与传感器 2 输出电压比较 – CMP0_IN2 输入
- J3[Pin1] - R2-R9 分压器连接至模数转换器 ADC0DP3
- J3[Pin2] - R3-R10 分压器连接至模数转换器 ADC0DP3

在实际应用中，阈值电压由比较器数模转换器生成 CMP_DAC 而不是来自分压器的电压。

- J4: 用于观察传感器 2 包络检测器输出的测试点。
- J4[Pin1] 连接至 MKL46 比较器 CMP0_IN3。
- J1[Pin2] 连接至模数转换器 ADC_SE4
- J5: 测试点
- J5[Pin1]: 比较输出 CMP0_OUT
- J5[Pin2]: 定时器 TMP1_CH0 输出
- J5[Pin3]: 定时器 TMP1_CH1 输出
- J6: 线圈的 VDD 电源。它必须通过跳线连接，以允许传感器激发。接头还可用于测量线圈的输入电流。
- J8: VDD_TPM 分压器和 AND 逻辑门的电源。可通过两种方式为其供电
- J8[Pins 1-2]: AND 门和分压器由 3.3V 板电源供电
- J8[Pins 2-3]: AND 门和分压器由 TPM0CH2 输出供电。该选项允许为仅供短时测量的电路供电，从而可降低供电电流。

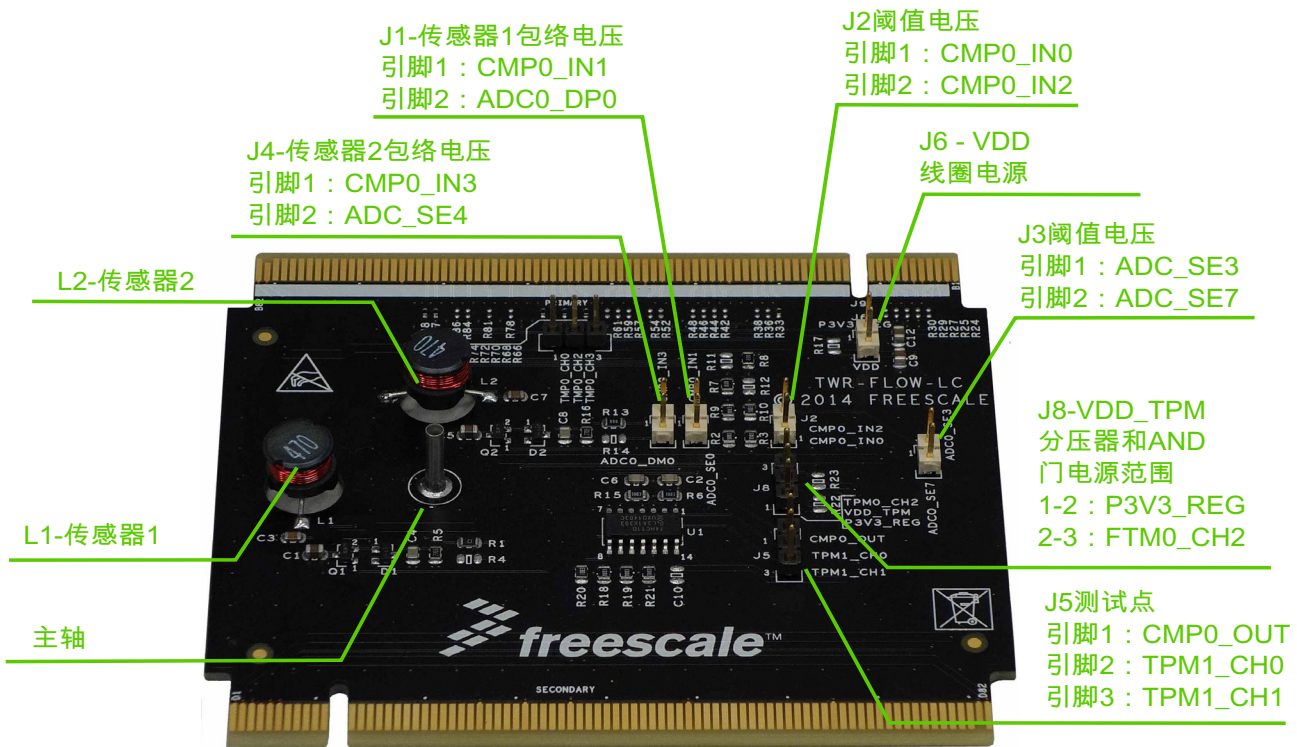


图 10. TWR-FLOW-LC 详细说明

4 软件

本章节介绍 Kinetis L LC 传感器编码器的软件应用程序演示。软件应用程序包含所需外设的设置、浆轮转数的测量和计算。

应用程序软件使用 IAR Embedded Workbench for ARM（版本 6.50.6），用 C 语言编写和编译，并且已针对执行速度全面优化。软件应用程序基于 Kinetis-L 裸机软件驱动程序。

软件包含一些基本模块。开始时，处理器通过上电复位唤醒。首先，时钟变为 32 kHz 晶振。

接下来，配置所有外设。仅为使用的外设使能时钟，以节省电流。使能低压检测装置以感测电池电量水平。配置 TMP0 和 TMP1 定时器并将相关引脚设为输出。

然后初始化 LCD 驱动器外设和 RTC。最后，设置比较器 CMP 和 DMA 通道，然后使能中断。

CPU 进入 VLPS 模式并且可通过以下一种中断唤醒：

- 在预定义的 DMA 周期数之后触发中断，并调用 DMA 中断服务例程。处理存储在 RAM 中的传感器状态值，并更新转数。
- 重新初始化 DMA 通道，并刷新 LCD 屏幕。
- 无论何时按下 SW4 按钮，均会调用 PORT 中断服务例程。在此例程中，将更新 LCD 屏幕索引
- 每秒调用一次实时时钟 (RTC) 中断服务例程，并且调用任务调度程序。任务调度程序是一个以定义时间调用的任务表
- LVD 中断服务例程当电源电压降至预设电平以下时将调用此中断，表示电池为空。用户应进行必要的维护，避免 CPU 失控和数据损坏。

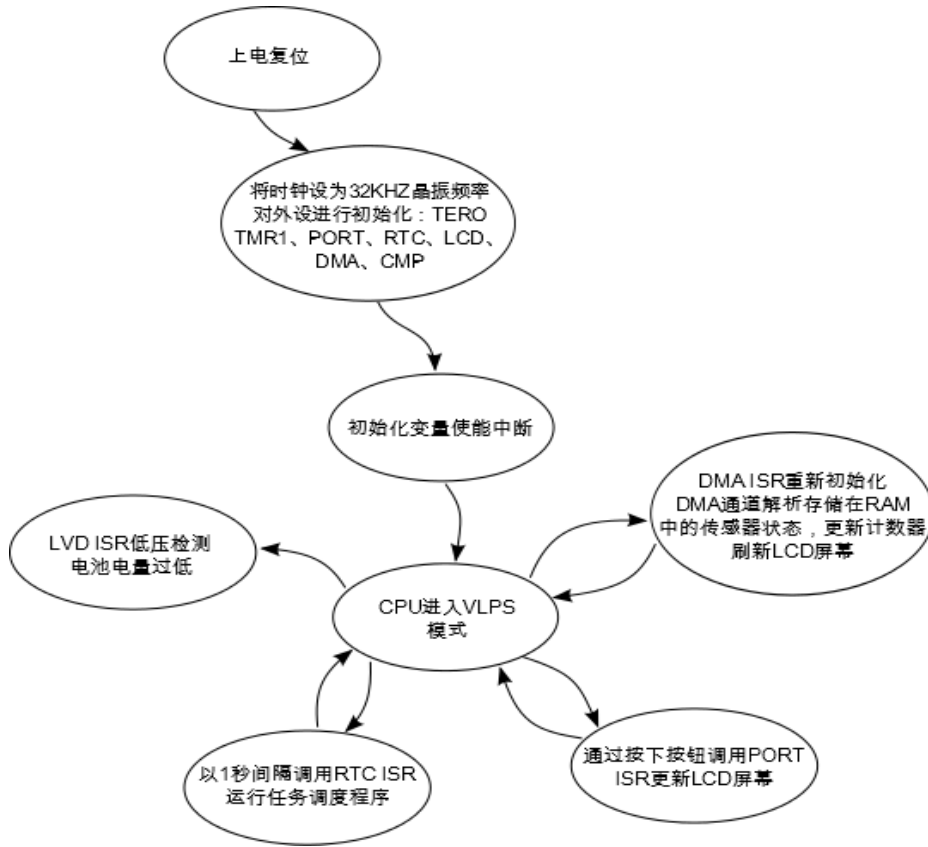


图 11. 软件流程图

5 结语

5.1 电流消耗

所有测量均在 25°C 实验室环境下完成，用 HP34401A 万用表测量电流。

表 1 分为两部分：前五行显示静态外设的电流消耗，其与选定的采样率无关。外设电流消耗列于“电流”这一行。“总计”这一行显示的是累积电流。

CPU 电流消耗是在 VLPS 模式下，消耗 2.8uA 电流。将 TMP0、TMP1、RTC 和 LCD 与处理器电流相加，将得出总的静态电流消耗为 4uA。在此状态下，CPU 始终保持在 VLPS 模式下。

表的第二部分是动态的，取决于选定的采样率，按每秒 100、200、300 和 400 个样本测量。选定的采样率还影响比较器和比较器数模转换器 CMP_DAC 外设的电流消耗，因为它们仅在测量时使能，在其他时间则保持禁用。

表 1. 电流消耗

增加的静态电流部分	电流 [uA]	总计 [uA]
CPU VLPS 模式	2.8	2.8
TPM0	0.3	3.1
TPM1	0.3	3.4
RTC	0.1	3.5
LCD	0.5	4
增加的动态电流部分 / 采样率		
DMA + CMP + CMP ADC @ 100Hz	4	8
DMA + CMP + CMP ADC @ 200Hz	7	11
DMA + CMP + CMP ADC @ 300Hz	11	15
DMA + CMP + CMP ADC @ 400Hz	15	19

附录 A 电路板示意图

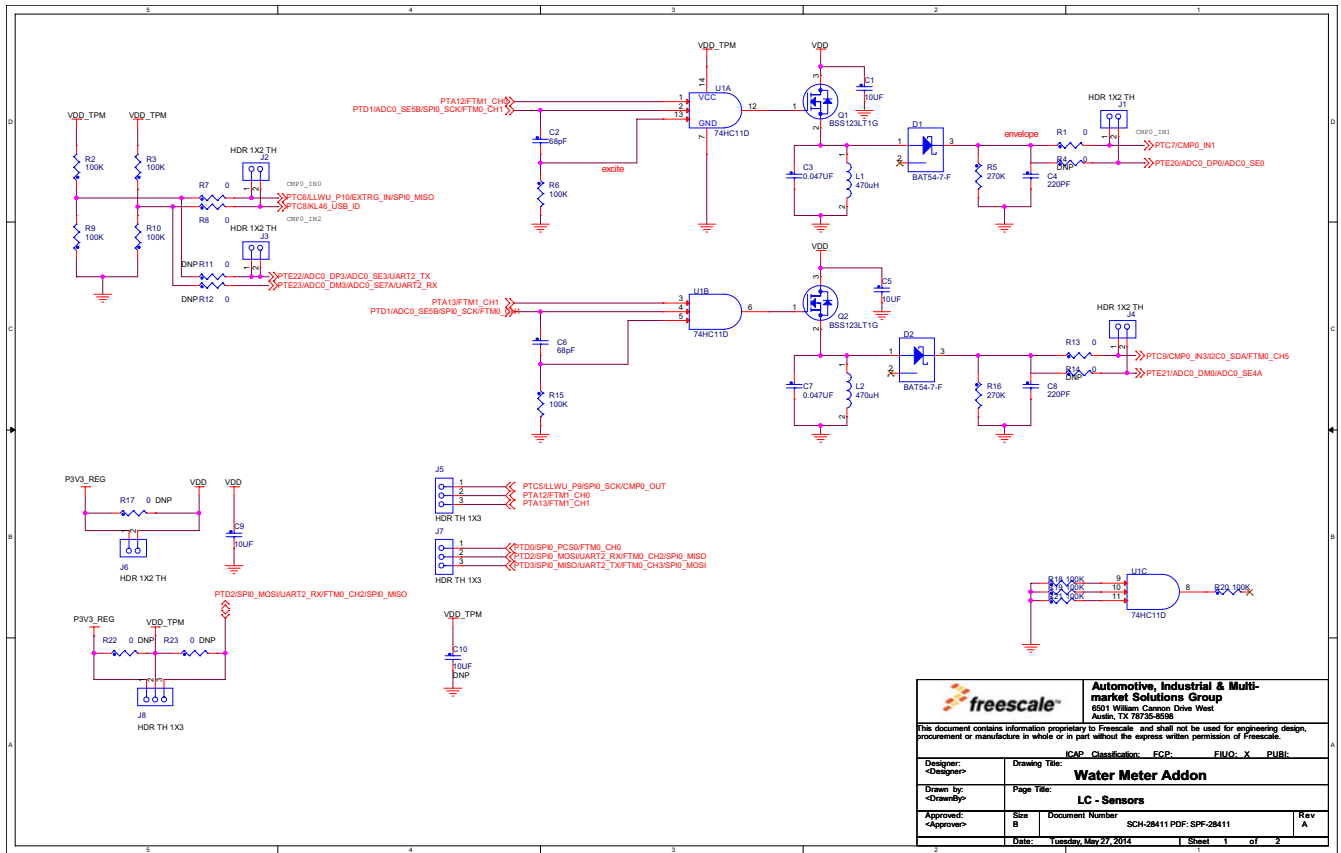


图 12. 水表插件

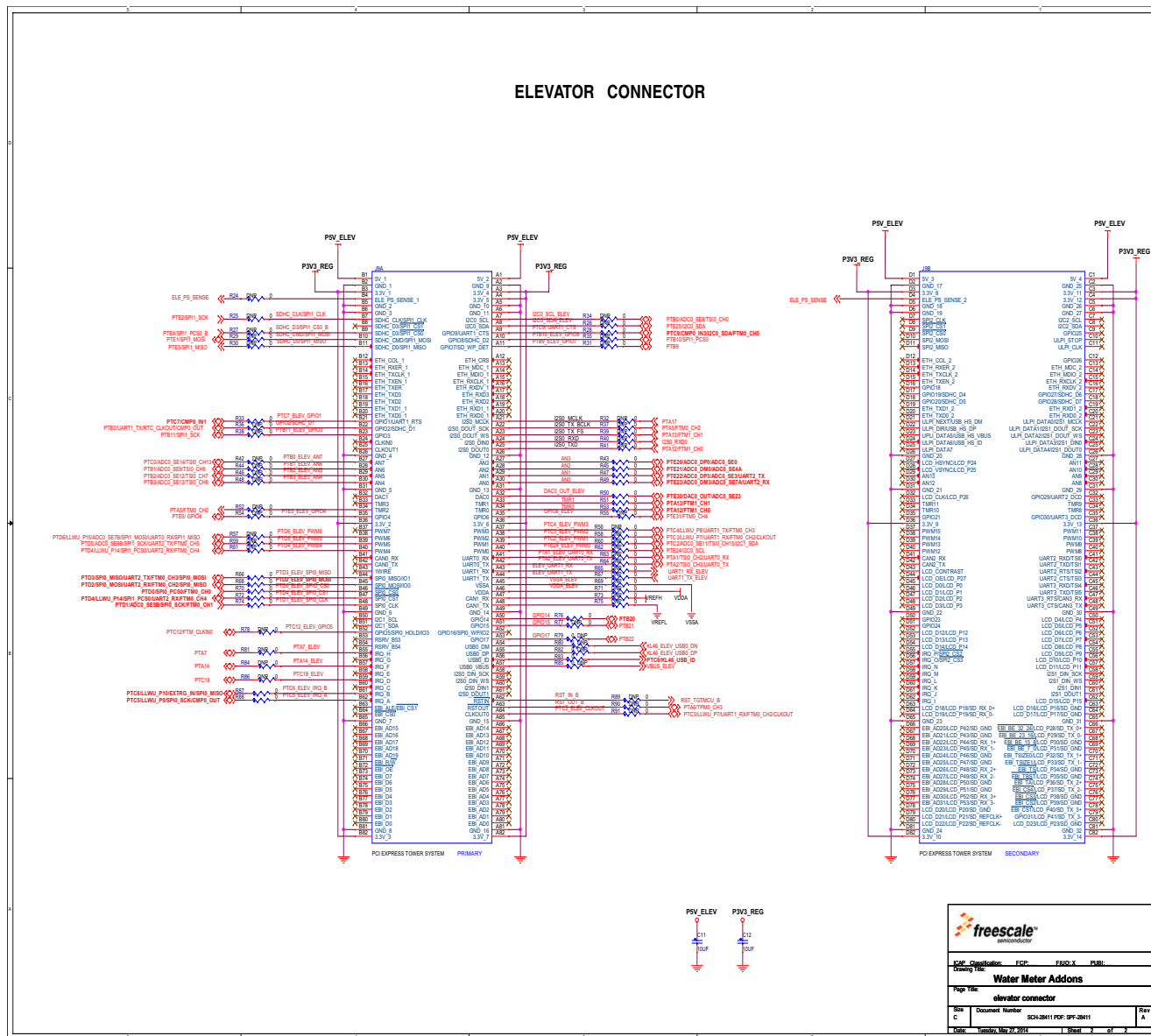


图 13. 侧板连接器

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

本文档中的信息仅供系统和软件实施方使用 Freescale 产品。本文并未明示或者暗示授予利用本文档信息进行设计或者加工集成电路的版权许可。Freescale 保留对此处任何产品进行更改的权利，恕不另行通知。

Freescale 对其产品在任何特定用途方面的适用性不做任何担保、表示或保证，也不承担因为应用程序或者使用产品或电路所产生的任何责任，明确拒绝承担包括但不限于后果性的或附带性的损害在内的所有责任。Freescale 的数据表和 / 或规格中所提供的“典型”参数在不同应用中可能并且确实不同，实际性能会随时间而有所变化。所有运行参数，包括“经典值”在内，必须经由客户的技术专家对每个客户的应用程序进行验证。Freescale 未转让与其专利权及其他权利相关的许可。Freescale 销售产品时遵循以下网址中包含的标准销售条款和条件：
freescale.com/SalesTermsandConditions.

Freescale, the Freescale logo, and Kinetis are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. All other product or service names are the property of their respective owners. ARM, the ARM Powered logo and Cortex are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere.

© 2015 Freescale Semiconductor, Inc.

© 2015 飞思卡尔半导体有限公司