

# 模块 10

实验：调试实时系统



# 实验： 调试实时系统

## 10.0 目标

本实验的目的是连接机器人将用于探索其世界的碰撞传感器，参见图 1。还将使用线传感器。

1. 您将使用数组实现最低限度的侵入式调试。
2. 您将把碰撞传感器连接到微控制器。
3. 您将学习如何实现 **SysTick** 周期中断。
4. 您将使用中断来实现多线程。
5. 您将通过将数据存储到微控制器的闪存 **ROM** 中来实现黑盒记录器。

**小知识：** 中断对于嵌入式系统非常重要，它提供了实现实时行为和多线程的机制。

## 10.1 入门

### 10.1.1 从下面的软件工程起步

浏览以下 3 个工程：

**PeriodicSysTickInt** （使用中断触发 LED）

**Flash** （将数据存储到闪存 ROM 中）

**Lab\_Debug** （本实验的开始项目）

### 10.1.2 参考资料

Meet the MSP432 LaunchPad (SLAU596)

MSP432 LaunchPad User's Guide (SLAU597)

JN\_LSA\_2RSLKBO1 line sensor datasheet

BumpSwitch.png mechanical drawing of switch

### 10.1.3 阅读材料

Volume 1 Sections 2.7, 6.2, 6.9, 9.1, 9.2, 9.4, and 9.6

Embedded Systems: Introduction to the MSP432 Microcontroller",

或

Volume 2 Sections 2.4, 3.9, 5.1, 5.4, and 5.7

Embedded Systems: Real-Time Interfacing to the MSP432 Microcontroller"

**注：** 我们选择传感器的采样率为 100 Hz，因为 10 ms 比该机器人中使用的电机的时间常数（100 ms）短约 10 倍。

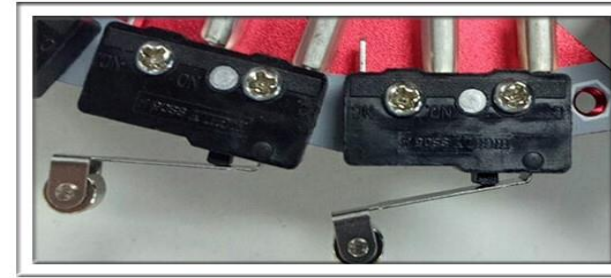


图 1. 位于机器人前部的碰撞传感器

### 10.1.4 本实验所需的组件

数量	组件描述	制造商	型号
1	MSP-EXP432P401R LaunchPad	TI	MSP-EXP432P401R
6	Bump switches	JiangNiu	
1	JN_LSA_2RSLKBO1 Reflectance Sensor Array		
12	0.5in 2-56 screw		
12	2-56 nut		

表 1. 本实验所需部件

### 10.1.5 所需实验设备

示波器（至少 10 kHz 的 1 个或 2 个通道采样）

逻辑分析仪（至少 10 kHz 的 4 个通道采样）



# 实验： 调试实时系统

## 10.2 系统设计要

本实验的第一个目标是使用 SysTick 中断实现线传感器测量；请参阅第 6 模块。与实验 6 类似，您应该每秒 100 次采样 8 位传感器。但是，必须在 SysTick 中断服务程序 (ISR) 的执行中使用传感器执行所有输入/输出。特别是，必须使用后续 SysTick 中断实现执行测量所需的 1 ms 延迟。在本实验中，此中断的优先级无关紧要，因为它是唯一的。但是，一旦我们将其它实验集成在一起，这将是一项高优先级的任务，因为测量需要是实时的。

第二个目标是在机器人上放置一至六个碰撞传感器，并将它们连接到微控制器。碰撞传感器允许软件知道机器人是否和在哪里与障碍物相撞。图 1 显示了一种可能的配置，图 2 显示了一个碰撞传感器的特写。

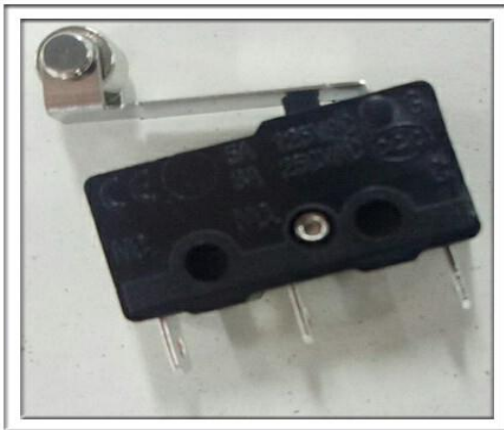


图 2. 瞬时接触开关允许碰撞检测。

第三个目标是开发一种称为转储的最小侵入式调试工具。转储是一种将战略信息记录到全局数组中的软件技术。它被归类为最小侵入性，因为与转储之间的时间 (100ms) 相比，在数组中存储数据的时间 (~1µs) 较短。在机器人操作期间，软件写入数组，当实验完成时，您使用调试器来观察值。

您将编写两个调试函数来实现转储。

- a) **Debug\_Init** 初始化 array(s)。

- b) **Debug\_Dump** 函数接收两个参数并将数据保存到 RAM 中。要记录两个 8 位参数 (8 位碰撞传感器数据和 8 位线传感器数据)。您的系统应允许至少 256 次测量 (512 字节)。每当新测量完成时，您的 SysTick ISR 将每秒调用 **Debug\_Dump** 100 次。由于数组有 256 个条目，因此这种基于 RAM 的记录允许最多 2.56 秒的调试。一旦数组已满，转储指针 (或索引) 将回绕到开头并覆盖最旧的数据。这样，在任何给定时间，数组中都有最后 2.56 秒的传感器数据。

第四个目标是扩展转储操作以将传感器数据连续存储到微控制器的闪存 ROM 上，实现黑盒记录器。闪存 ROM 大于 RAM，但它也是有限的。因此最终，ROM 空间将填满。更具体地说，您将使用 128 kibibytes 的闪存 ROM (地址 0x00020000 至 0x0003FFFF)。这些地址位于 Flash Bank1 中。您的程序将驻留在 Flash Bank 0 (地址 0x00000000 至 0x0001FFFF) 中，允许您在写入 ROM 的同时执行代码。如果记录 200 字节/秒，则可以将数据保存 655 秒或大约 11 分钟。

您将实现另外两个调试函数：

- a) **Debug\_FlashInit()** 将擦除 128 kibibytes 的 flash ROM，地址 0x00020000 到 0x0003FFFF。擦除 ROM 将数据设置为 0xFF。您可以选择从 32 字节到 512 字节的任何块大小。设  $2^n$  为您的块大小。在 128k 空间有  $2^{17}/2^n$  块。如果块的数据是 0xFF，则该块被认为是空的。
- b) 函数 **Debug\_FlashRecord()** 将  $2^n$  个字节记录到闪存 ROM 上的下一个空闲块中。稍后您可以使用调试器观察记录的数据。

**Note:** 线传感器测量和 RAM 记录将在 SysTick ISR 中进行，但所有闪存操作必须在主程序中进行。线传感器生成 8 位数，碰撞传感器生成 8 位数。在 100 Hz 的采样率下，系统产生 200 字节/秒。因此，每  $0.05 \cdot 2^n$  秒您将写入  $2^n$  个字节到闪存 ROM。当 ROM 在 655 秒后满了，您应该停止 ROM 记录。

## 10.3 实验准备

您学习了如何在模块 6 的实验中连接 JN\_LSA\_2RSLKBO1 线路传感器。您还学习了如何在模块 8 中连接开关。在本实验中，您将把碰撞开关连接到机器人的前部，并将它们连接到 LaunchPad。您可以使用 2-56 螺钉和螺母将碰撞传感器定位在机器人前部的边缘。图 3 显示了六个传感器的一种可能位置。



# 实验： 调试实时系统

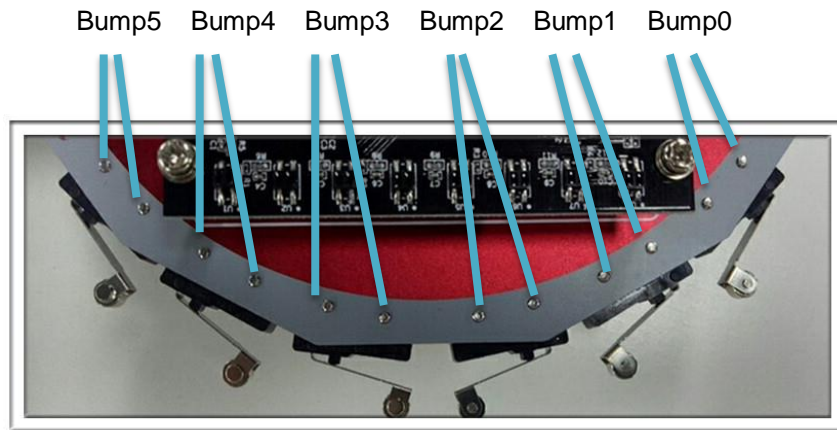


图 3. 连接到机器人前部的碰撞传感器（底视图）。

## 10.4 实验步骤

### 10.4.1 使用 SysTick 中断读取线传感器

您将使用 SysTick 中断在后台执行线传感器输入。如果 SysTick 中断发生在 1000 Hz（每 1ms），则在 ISR 执行一次

1. 将 P5.3 置为高电平（点亮 8 个红外 LED）
2. 设 P7.7-P7.0 为输出，并置为高电平
3. 延时 10 μs
4. 设 P7.7-P7.0 为输入

您将上述四个步骤定义为函数 **Reflectance\_Start()**，并在 SysTick ISR 中每隔十次调用此函数。测量的第二部分发生在随后的 ISR 中

5. Read the 8 位 sensor result
6. Turn off the 8 IR LEDs (P5.3) low
7. Store the data into a shared global variable

您将步骤 5 和 6 定义为函数 **uint8\_t Reflectance\_End()**。步骤 7 将在 SysTick ISR 中发生。如果您以 100 Hz 的频率对线路传感器进行采样，则会有 8 个 SysTick 中断，在此期间软件不执行任何操作，一个中断调用 **Reflectance\_Start()**，一个中断调用 **Reflectance\_End()**。

在每次 ISR 执行期间触发端口引脚都是很好的调试方式。将传感器数据放入存储器监视窗口并使用调试器和示波器验证传感器的行为与实验 6 类似。

### 10.4.2 连接碰撞传感器

您可以实现正逻辑或负逻辑。您可以实现内部电阻或外部电阻。

**Note:** 如果您打算使用 Wifi 实验，我们建议使用 P8.7-P8.3，P8.0 引脚作为六个碰撞传感器，因为这些引脚不会与 Wifi 实验发生冲突。如果您打算使用边沿触发中断实验，我们建议将引脚 P4.7-P4.2 用于六个碰撞传感器，因为这些引脚会引起边沿触发中断。

您将编写一个函数来初始化碰撞传感器，**Bump\_Init()**。该函数只需设置适当的端口引脚，并根据需要启用内部电阻。第二个函数，**uint8\_t Bump\_Read()**，读取开关并返回一个 8 位结果。每次 SysTick ISR 执行对 **Reflectance\_End()** 的调用时，您还应该调用 **Bump\_Read()**，并且应该将结果放在共享的全局变量中并设置信号量。

**Note:** 在后续机器人移动的实验中，在调用 **Bump\_Read()** 后。您将在 SysTick ISR 中处理冲突。因此，碰撞事件的最坏情况延迟将是 10 毫秒。

### 10.4.3 调试转储

实现调试转储有许多设计选择。您可以创建两个 8 位数组，一个用于线传感器，另一个用于碰撞传感器。或者，您可以创建一个 16-bit 阵列并将两个传感器读数存放成一个 16-bit 数据值。规格要求至少 256 个记录，但如果您愿意，可以增加此值。MSP432 微控制器具有 64 kibibytes 的 RAM，这种调试功能应该只使用一小部分可用 RAM。另一种选择是使用指针还是索引来访问数组。这通常涉及执行速度和软件风格之间的权衡。您可以实现这两者，并观察每个版本生成的汇编代码。为减少干扰，我们建议您选择执行速度最快的方法（即最少的汇编指令）。

定量测量调试工具的侵入性的一种方法是计算实现 **Debug\_Dump()** 所需的汇编指令数。更好的方法是使用示波器和未使用的端口引脚，如 Program10\_1 所示。



## 实验： 调试实时系统

```
int Program10_1(void){ uint8_t data=0;
    Clock_Init48MHz();
    Debug_Init();
    LaunchPad_Init();
    while(1){
        P1->OUT |= 0x01;
        Debug_Dump(data,data+1); // linear sequence
        P1->OUT &= ~0x01;
        data=data+2;
    }
}
```

### 10.4.4 黑盒记录器

在调试黑盒记录器时，我们建议您首先使用简单的主程序，如 Program10\_2 和 Program 10\_3。这样您就可以分别测试两个功能。使用调试器验证 ROM 是否已正确编程。

```
// Driver test
#define SIZE 256 // feel free to adjust the size
uint16_t Buffer[SIZE];
int Program10_2(void){ uint16_t i;
    Clock_Init48MHz();
    LaunchPad_Init(); // built-in switches and LEDs
    for(i=0; i<SIZE; i++){
        Buffer[i] = (i<<8)+(255-i); // test data
    }
    i = 0;
    while(1){
        P1->OUT |= 0x01;
        Debug_FlashInit();
        P1->OUT &= ~0x01;
        P2->OUT |= 0x01;
        Debug_FlashRecord(Buffer);
        P2->OUT &= ~0x01;
        i++;
    }
}
```

```
int Program10_3(void){ uint16_t i;
    Clock_Init48MHz();
    LaunchPad_Init(); // built-in switches and LEDs
    for(i=0; i<SIZE; i++){
        Buffer[i] = (i<<8)+(255-i); // test data
    }
    P1->OUT |= 0x01;
    Debug_FlashInit();
    P1->OUT &= ~0x01;
    i = 0;
    while(1){
        P2->OUT |= 0x01;
        Debug_FlashRecord(Buffer);
        P2->OUT &= ~0x01;
        i++;
    }
}
```

使用示波器或逻辑分析仪测量函数的执行时间。计算 ROM 编程实现的最大数据速率（块中的字节数除以写入块的时间）。此数据速率将比 200 字节/秒的数据生成速率快得多。但是，根据缓冲区的大小，编程一个缓冲区可能需要或不需要超过 10 毫秒。

当将黑盒记录器与传感器测量值集成时，使用 SysTick ISR 中设置的共享全局标志 (**Semaphore**)，当缓冲区满时。您将读取主程序中的标志，以了解何时进行下一次调用 **Debug\_FlashRecord()**。确保系统以正确顺序正确记录所有数据的一种方法是测量实际的线路和碰撞传感器输入，但记录在 Program10\_2 中生成的测试数据。使用测试数据可以查看是否有任何点丢失或重复。你的最终主程序会有这种行为。

```
while(1){
    if(Semaphore==1){ // wait for flag to be set
        P2->OUT |= 0x01;
        Debug_FlashRecord(Buffer);
        P2->OUT &= ~0x01;
        Semaphore = 0; // clear flag
    }
}
```



# 实验： 调试实时系统

## 10.5 疑难解答

### 碰撞传感器不起作用：

- 按照模块 8 中的说明检查接线
- 使用电压表，示波器或逻辑分析仪查看信号
- 查看调试器中的端口寄存器

### Flash 存储不起作用：

- 运行 Flash 项目代码测试硬件
- 观察 Flash 项目以了解如何调用低级驱动程序

## 10.6 请思考

在本节中，我们列出了完成本实验后要思考的思考问题。这些问题旨在测试您对本实验中概念的理解。

- 在本实验中，我们只存储了数据，但不是数据采样的时间。在这个应用程序中为什么我们不节省时间？
- 当我们将线传感器和碰撞传感器接口分类为实时时，它意味着什么？为什么这些实时输入是重要的？
- 为什么我们应该使用内部电阻代替外部电阻来实现碰撞接口？
- 我们将延迟指定为最大 10 毫秒。什么是平均延迟？我们怎样才能重新设计这个以减少延迟？
- 为什么将红外 LED 关闭 90% 并且仅在 10% 的时间内开启是好的设计？
- 擦除 ROM 会发生什么？如果 ROM 位已经为 0 会发生什么，我们尝试将其编程为 1？
- 假设我们在制造机器人时调用 `Debug_FlashInit()`，但是主程序的后续运行不会擦除 ROM。ROM 中会记录什么？即，如果我们删除并稍后恢复电源，这些数据会发生什么？

## 10.7 其它挑战

在本节中，我们列出了您可以执行的其它活动，以进一步探索此模块的概念。您可以扩展系统或提出完全不同的东西。例如，

- 增加记录时间的一种技术是仅在数据发生变化时存储数据。但是，在此方案中，您需要记录数据和数据更改的时间。在 `SysTick ISR` 中递增的全局变量可以以 `ms` 为单位保持系统时间。
- **临界区**是软件行为，它取决于两个看似无关的代码片段的相对执行情况。例如，在 `ISR` 中执行 `P2->OUT ^= 0x01` 然后在主程序中执行 `P2->OUT ^= 0x02` 将由于临界区而产生错误。在两个不同的线程中共享公共端口会生成一个临界区。您可以重写 `LaunchPad` 驱动程序代码（`LaunchPad.c`）以使用 **bit-banding**，这将删除一些临界区。
- 您可以在项目中包含 `UART0.c` 并创建一个额外的调试功能，将闪存 ROM 中记录的数据转储到串行端口。然后，您可以运行 `PuTTY` 或 `TEsXasDisplay` 等程序来查看数据。此功能允许您在 PC 上捕获数据以进行报告编写和文档编制。

## 10.8 接下来是哪些模块？

这是本课程中中断的第一次使用。以下模块将基于此模块构建：

模块 12) 连接电机到机器人。

模块 13) 使用定时器创建 PWM 信号，使用中断管理多个软件任务。

模块 14) 在开关触点上立即对软件任务使用边沿触发中断。

## 10.9 您应该已经学会

在本节中，我们将回顾您应该在本单元中学到的重要概念：

- 使用中断实现多线程
- 使用全局变量在线程之间进行通信
- 访问数组时优化执行速度
- 创建最低限度的侵入式调试工具
- 使用端口引脚和示波器施行执行分析
- 擦除并编程 flash ROM 以记录调试数据

## IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ("TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products (<http://www.ti.com/sc/docs/stdterms.htm>), [evaluation modules](#), and [samples](http://www.ti.com/sc/docs/sampterm.htm) (<http://www.ti.com/sc/docs/sampterm.htm>).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2018, Texas Instruments Incorporated