



模块 6

实验：通用输入输出接口



实验：通用输入输出接口

6.0 目标

本实验的目的是连接用于探索机器人世界的循线传感器，参见图 1。

1. 你将学习 C 语言中的函数，条件，循环和运算。
2. 你将使用 GPIO 执行输入和输出。
3. 你将了解光如何转换为电压，以及电压如何转换为二进制。
4. 你将连接循线传感器到微控制器。

小知识：通用输入输出（GPIO）是在微控制器上执行 I/O 的最简单，最普遍的方法。你在本实验中连接的传感器将允许机器人探索其世界。

6.1 入门

6.1.1 从下面的软件工程起步：

浏览以下 3 个工程：

GPIO（输出到四个引脚是一个非常简单的系统）

InputOutput（一个简单的系统在 LaunchPad 上，开关作为输入，LED 作为输出）

Lab_GPIO（本实验的入门项目）

6.1.2 参考资料

Meet the MSP432 LaunchPad (SLAU596)

MSP432 LaunchPad User's Guide (SLAU597)

JN_LSA_2RSLKBO1.pdf, line sensor datasheet

6.1.3 阅读材料

Volume 1 Sections 4.1 and 4.2

Embedded Systems: Introduction to the MSP432 Microcontroller",

或

Volume 2 Sections 2.2 and 2.4

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

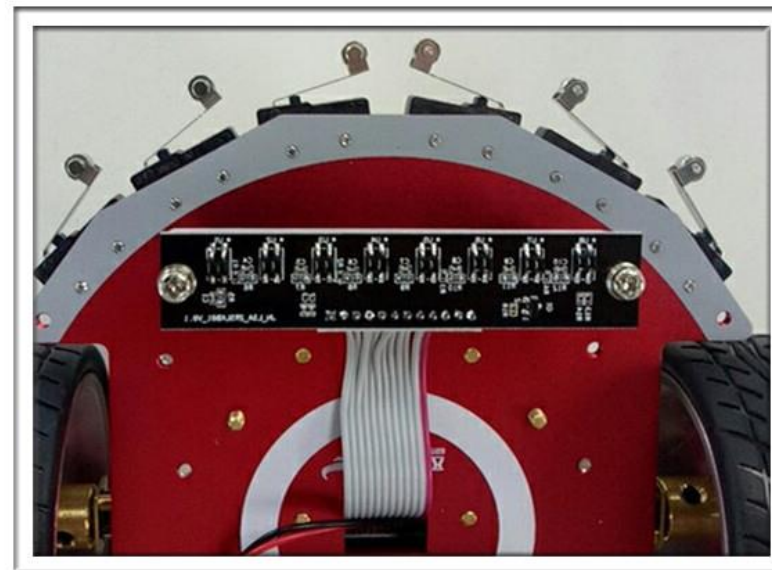


图 1.JN_LSA_2RSLKBO1 安装在机器人底部，距离地面 3mm

6.1.4 本实验所需组件

Quantity	Description	Manufacturer	Mfg P/N
1	MSP-EXP432P401R LaunchPad	TI	MSP-EXP432P401R
1	Reflectance Sensor Array	JiangNiu	JN_LSA_2RSLKBO1

表 1

6.1.5 所需实验设备

示波器（1 或 2 个至少 10kHz 采样的通道）

逻辑分析仪（4 个至少 10kHz 采样的通道）

*注意：你还需要黑色非反光胶带和白色表面



实验：通用输入输出接口

6.2 系统设计要求

该实验的最终目标是设计一个传感器系统，测量机器人相对于线的位置。例如，白色表面上的黑色胶带可用于探索其世界的机器人挑战中。

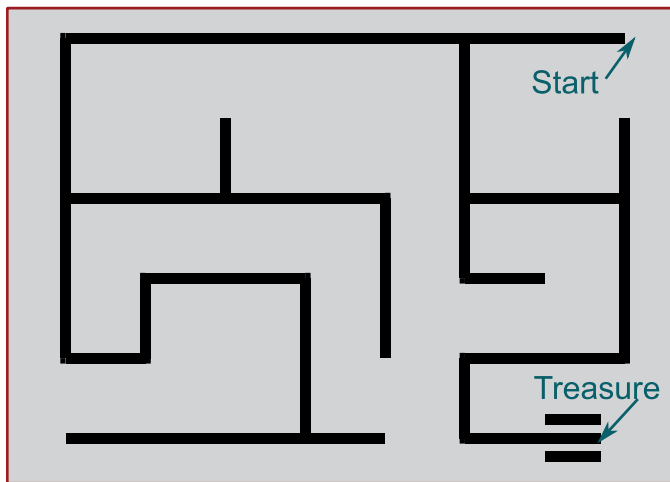


图 2. 可能最终的机器人挑战探索世界和寻找宝藏。

机器人有 8 个传感器用来检测线路，参见图 3。

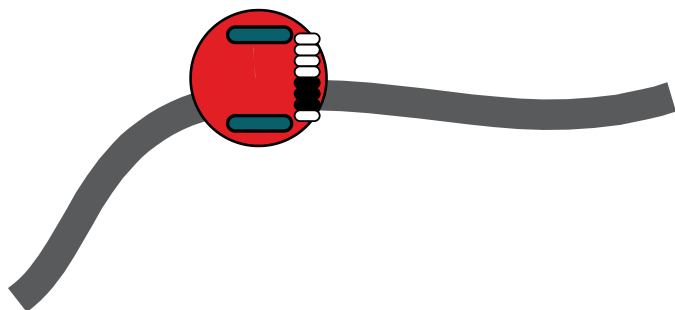


图 3. 机器人有 8 个传感器和 2 个电机轮。

每个传感器都对应二进制数，如果看到黑色则返回 1，如果看到白色则返回 0。如果机器人正确定位在线上，中间传感器将看到黑线。如果机器人向左或向右稍微偏离，一个或多个外部传感器将看到黑线。如果机器人完全脱离线路，所有传感器都会看到白色。我们将使用传感器集成将八个二进制数组合成一个位置参数。我们将机器人的位置定义为从传感器到线的距离。我们将使用的传感器宽约 66 毫米（传感器之间约为 9.5 毫米），所以我们应该能够估计距离线中心-33 到+33 mm 的机器人位置。

图 4 显示了传感器测量系统的所需输出。基本上，根据机器人相对于线的位置，要求您创建一个从-330 到+330（单位 0.1mm）不等的输出。

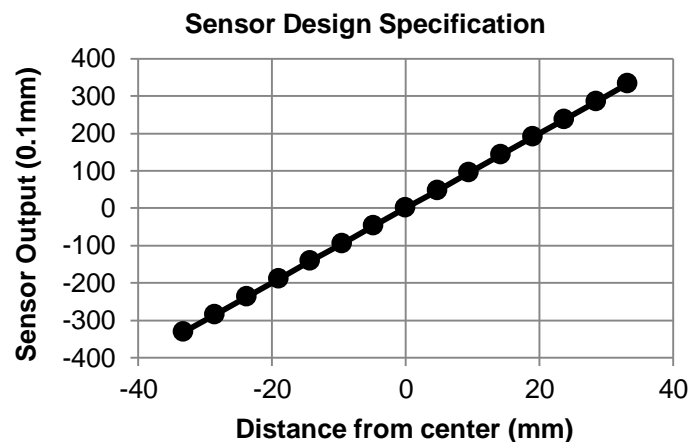


图 4. 测量系统期望的输出。

有许多性能指标可用于评估系统。**准确度**是给定时间的真实线和测量值之间的差异。

$$\text{平均准确度} = \frac{1}{N} \sum_{i=1}^N |x_i - T_i|$$

其中 x_i 是测量值， T_i 是实际或真实线。准确度是有意义的，但对于解决探险者的任务并不是很重要。

如果输入的增加从不导致输出减少，则系统是**单调的**，即，随着输入的增加，输出增加还是保持不变。同样的，随着输入的减少，输出减少还是保持不变。单调性对机器人控制系统很重要。你将通过在线路上缓慢滑动传感器并测量每个位置的系统输出来确定您的系统是否是单调的。



实验：通用输入输出接口

噪声对控制系统也很重要。**标准方差**是噪声的定量测量，由下式给出：

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$

$$\sigma = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2$$

在这种情况下， x_i 是在输入固定的情况下进行的测量。**变异系数**是标准偏差除以平均值抑或平均值。

$$CV = \sigma/\mu$$

特异性是指系统对期望信号的相对灵敏度，而不是对其它不需要的相对灵敏度。具有良好特异性的系统将仅响应感兴趣的信号并且独立于这些其他因素。你要学习的不必要的影响是与线的角度，如图 5 所示。尤其是，你将为角度 90, 75, 60 和 45 度创建响应曲线，如图 4 所示。

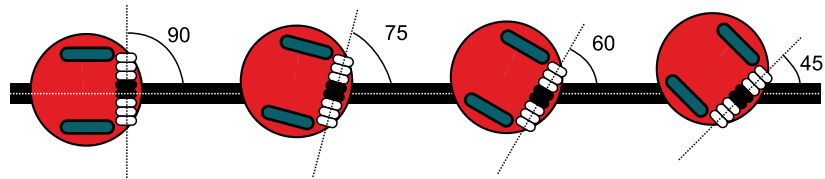


图 5. 希望系统能够在很宽的角度范围内测量到线中心的相对距离

6.3 实验准备

循线传感器的最佳感应距离为 3 mm (0.125")。你需要将循线传感器和地面之间的距离固定为约 3 mm。在这个实验里，您可以使用两个 3 毫米的物体连接到传感器的每一端来设置距离并使其平行。更好的方法是将传感器安装在机器人的底部（如果您已购买套件），再次将传感器平行放置在离地面 3 毫米的地板上。见图 1。您将需要访问挑战竞技场的模型（小代表片段）（类似于图 2 所示），以便测试循线传感器。循线传感器在标有黑色非反光胶带的白色反光表面上工作良好。

您将通过 **JN_LSA_2RSLKBO1** 循线传感器连接到 **MSP432 LaunchPad** 来实现本实验，如图 6 所示。有关连接说明，请参阅传感器的数据表。

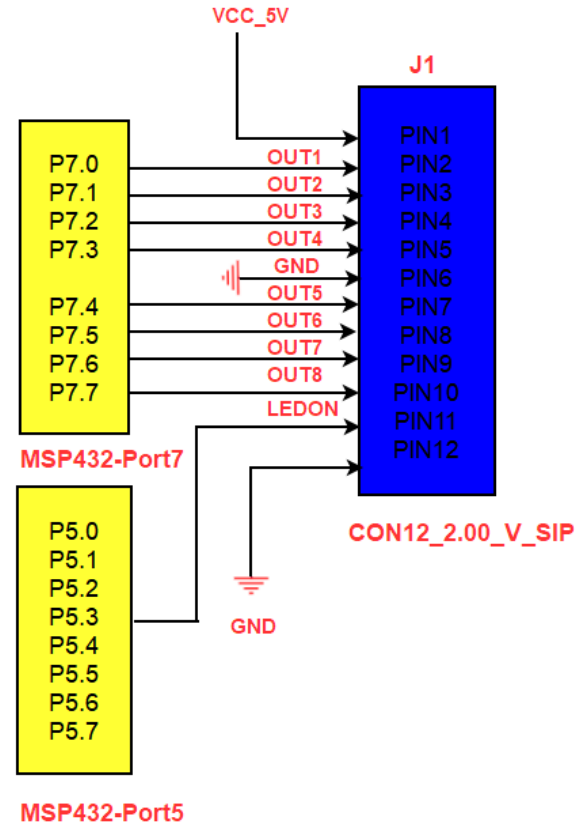


图 6. 8 通道循线传感器连接到端口 5 和 7。（P7.0）位于右侧，S8（P7.7）位于左侧。



实验：通用输入输出接口

6.4 实验步骤

6.4.1 将 MSP432 端口配置为输入和输出

要初始化通用 I/O 端口，你将执行以下步骤。首先通过将 0 写入 PxSEL0 和 PxSEL1 寄存器来指定 GPIO。其次，你将设置寄存器的方向。

打开 GPIO 项目 - 编译，下载和调试此示例。使用调试器，单步执行主程序时，请观察端口 4 的方向寄存器和输出寄存器。运行程序并观察端口 4 的 3-0 bit 的输出。注意使用函数初始化端口 4。Port 4 的输出是正常还是不正常？注意 main2 文本提供了一个抽象概念。

打开 InputOutput 项目并注意初始化可以以不同的方式完成 (main, mainb 和 main2)。main 的例子是不正常的。mainb 例子是正常的。main2 例子是正常的并提供抽象概念。

使用 InputOutput 项目，编译，下载和运行程序。使用调试器，在单步执行主程序时，观察端口 1 和 2 寄存器的方向，输入和输出。注意端口是如何初始化的，如何输入数据以及如何输出数据。在这个项目中有三个版本。请参阅项目文件中的注释。

Note: 第一个版本是“不正常的”，因为它写入整个寄存器，即使它只需要影响一些位。第二个版本是正常的，因为它只会改变所需的位数。第三个版本提供抽象概念，通过逻辑名称 (SW1IN, SW2IN, BLUEOUT, GREENOUT, REDOUT) 而不是物理实现 (P1-> IN, P2-> OUT) 来引用对象。

在实验室中执行上述步骤后，将对以下三个版本进行评论：

- 易于理解
- 能够集成到更大的系统中
- 移植性，易于在另一个微控制器上实现系统

6.4.2 配置低-级别传感器输入/输出

首先，你将分析循线传感器上的八个传感器的其中一个，并了解传感器的工作原理。

你将连接 TI MSP432 端口 到 JN_LSA_2RSLKBO1 循线传感器。

我们将使用 Port 5， bit 3 作为输出， 连接到循线传感器的 IR 引脚。 P5.3 打开红外 (IR) LED， P7.0 连接到 JN_LSA_2RSLKBO1 的一个传感器引脚， 并将 P1.0 设置为数字输出（在本节中用于测试）。见图 7。尤其是， 本节的目的是观察给定表面的反射如何影响 P7.0 上的电压， 以及微控制器如何根据反射表面的颜色将 P7.0 上的电压转换为二进制数据。

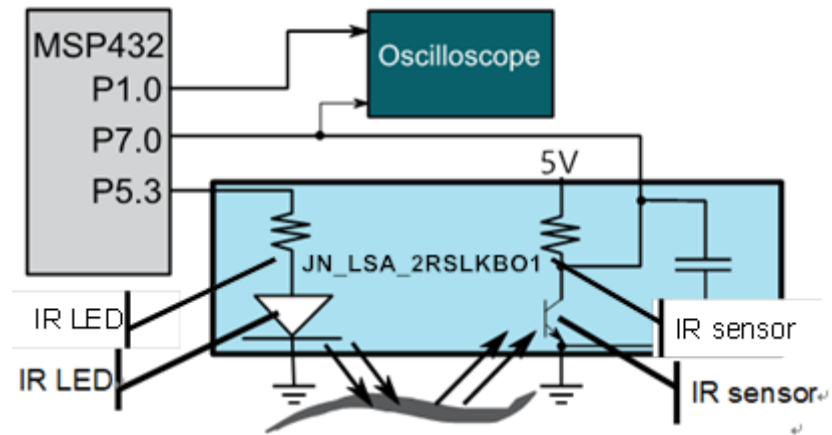


图 7.关于一个传感器的硬件配置

解决方案将放在 Reflectance.c 中，以便该软件可用于后续实验。有关你需要编写的软件的详细规范，请参阅 Reflectance.h。将传感器 S1 (P7.0) 放在白色反射表面上方 3 mm 处。执行此软件初始化：

- 1) 初始化时钟为 48 MHz， `Clock_Init48MHz();`
- 2) 将 P5.3 设为输出，且初始化为低电平
- 3) 将 P7.0 初始化为输入
- 4) 将 P1.0 初始化为输出（您可以使用任何未定义的引脚）

将 P7.0 和 P1.0 连接到双通道示波器。如果你有单通道示波器，则首先测量 P7.0，然后测量 P1.0。由于输入（循线的颜色）不会改变，因此可以对齐两个单独的测量。主程序的主体循环地执行步骤 1 到 7：



实验：通用输入输出接口

Main 函数

```

0) 初始化
while(1){
1) 设 P5.3 为高电平（点亮红外 LED）
2) 将 P7.0 设为输出，并设为高电平（给电容充电）
3) 等候 10 us, Clock_Delay1us(10);
4) 将 P7.0 设为输入
5) 运行此循环 10000 次
    a) 读取 P7.0（将 P7.0 上的电压转化为二进制）
    b) 将二进制数赋给 P1.0（使您能实时看到二进制数）
6) 设 P5.3 为低电平（关闭红外 LED，节省电力）
7) 等候 10 ms, Clock_Delay1ms(10);
}

```

请注意，读取传感器 10000 次，寻找输入信号从 1 变为 0。你将使用示波器或逻辑分析仪观察 P1.0 的变化

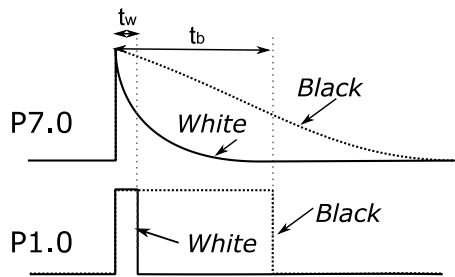


图8.光转换为电压，电压转换为二进制

运行系统并观察示波器上 P7.0 和 P1.0 上的电压。对于白色表面，二进制在 P1.0 上从 1 切换到 0 的电压是多少？用黑色非反射表面重复实验。

注：基本上，正在发生的事情是 P7.0 上的 10us 输出脉冲对 JN_LSA_2RSLKBO1 电路板上的电容充电（见图 7）。一旦微控制器将引脚设置为输入，光敏晶体管上的反射光就会使电容器放电。晶体管在白色反射面上具有更多的光，并通过集电极 - 发射极传导更多电流。该电流使电容器放电。注意白色面放电更快，P7.0 下降得更快。使用 P7.0 的上升沿作为 0 的时间参考，测量白色

和黑色表面的 P1.0 下降时间，见图 8。对于白色表面，测量时间 t_w 。对于黑色表面，测量时间 t_b 。

6.4.3 底层传感器接口

现在你已了解传感器的工作原理，你将创建一个低级别软件驱动程序，可同时测量 8 个传感器。你需要连接 8 个传感器输入，如图 6 所示。你将假设 MSP432 以 48 MHz 时钟运行。初始化包括：

- 1) Make P5.3 an output, and set it initially low
- 2) Make P7.7 – P7.0 inputs

创建一个测量 8 个传感器的 C 函数。让 `time` 成为传递给此函数的参数，选择 `time` 在 t_b 和 t_w 之间。基本上，如果我们等待 1000 μ s，那么白色将衰减到零，而黑色仍将是高。这使我们可以区分白色和黑色。

按以下顺序执行 8 个步骤：

- 1) 设 P5.3 为高电平（点亮红外 LED）
- 2) 设 P7.7 – P7.0 为输出，并置为高电平（给 8 个电容充电）
- 3) 等待 10 us, **Clock_Delay1us(10);**
- 4) 设 P7.7 – P7.0 为低电平
- 5) 等待 `time` us, **Clock_Delay1us(time);**
- 6) 读取 P7.7 – P7.0 输入（将电压转化为二进制）
- 7) 将 P5.3 置为低电平（关闭红外 LED，节省电力）
- 8) 返回第 6 步中的 8 位二进制测量值

如果机器人或竞技场发生变化，通过该功能可以轻松地重新调整传感器。根据示波器的测量结果显示，时间约为 1000 μ s。你将使用类似于 Program6_1 的简单主程序测试你的系统，并使用调试器观察 Data 变量，以确保它符合预期的行为（“0”表示白色，“1”表示黑色）。

```

// Use this program to test the Read function
uint8_t Data; // JN_LSA_2RSLKBO1
int Program6_1(void) {
    Clock_Init48MHz();
    Reflectance_Init(); // your initialization
    TExaS_Init(LOGICANALYZER_P7);
    while(1) {
        Data = Reflectance_Read(1000); // your measurement
        Clock_Delay1ms(10);
    }
}

```



实验：通用输入输出接口

注：会注意到此测量会消耗所有处理器时间。当电容放电的等待时间为 1ms 时则改时间的浪费的（或不放电）。在主循环内的 10ms 延迟中再次浪费时间。一旦我们学习了模块 10 中的中断，我们可以恢复这个浪费的时间。

6.4.4 顶层传感器集成

在这章节，你将把八个二进制测量值组合成一个参数，表示机器人远离线路中心的数量。我们假设传感器 S1 (P7.0) 位于机器人的右侧，距离中线 33.2 毫米。另外，我们假设传感器 S8 (P7.7) 位于机器人的左侧，距离中线 -33.2 毫米。如前面提到的，另一个目标是使这个参数对角度不敏感。如果传感器运行正常，存储在 Data 中的 8 位二进制数分为四类：

- 1) <全部为 0> (偏离线路或白色平面)
- 2) <左边为 0, 右边为 1>, 例如 00000111 (偏左)
- 3) <左边为 0, 中间为 1, 右边为 0>, 例如 00110000 (在线上)
- 4) <左边为 1, 右边为 0>, 例如 11110000 (偏右)

图 9 显示传感器相距约 9.5 mm，机器人中心在传感器 4 和 5 之间对齐 (P7.3 和 P7.4)。

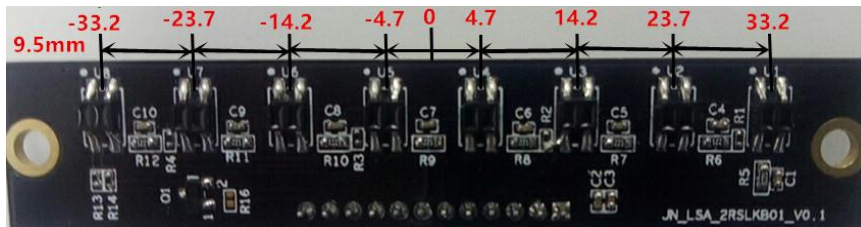


图 9. 位置定义为到机器人中心的相对距离。

以 0.1mm 为单位定义 8 个距离值的数组， W_i 为 $i = 0$ to 7。
 $W = \{332, 237, 142, 47, -47, -142, -237, -332\}$

对于 Reflectance_Read 函数返回的每个二进制位，将 b_i 定义为 0 (白色) 或 1 (黑色)。

一种可能的传感器集成功能是计算 8 个传感器读数或二进制结果的加权平均值。假设至少有一个黑色读数，估计距离 Reflectance_Position 为：

$$d = \frac{\sum_{i=0}^7 b_i W_i}{\sum_{i=0}^7 b_i}$$

在连接 MSP432 LaunchPad 开发板和 JN_LSA_2RSLKB01 之后，在实验的这一部分中，你将如图 1 所示进行安装，然后根据图 9 所示的规格定义真实距离。

将传感器缓慢滑过循线 (图 2 显示了迷宫线)，并将测量的真实距离和由 Reflectance_Position 估计/计算的距离绘制为真实距离的函数，类似于图 4 (首先在图 5 中定义的 90 度)。

评论反应是否是单调的。解释是否存在测量噪声的区域。对其他角度重复实验 (图 5) 以估计测量的特异性。

注：如果测量结果是单调且低噪声，则传感器将有效。Another issue is offset. 读数为 0 时机器人在哪里？由于控制器将尝试将距离测量值驱动为 0，这是控制器将寻求的位置。



实验：通用输入输出接口

6.5 疑难解答

传感器不起作用：

- 检查接线，如图 5 所示，包括电源和接地
- 按照第 6.4.2 节所述查看信号 P5.3 和 P7.0
- 单步执行并验证方向寄存器是否正确
- 确认处理器以 48MHz 运行
- 尝试其他 LaunchPad 或 尝试其它循线传感器

6.6 请思考

在这章节中，我们列出了完成本实验后要思考的思考问题。这些问题旨在测试你对本实验中概念的理解。

- 什么是函数？如何使用函数来简化软件？
- 什么是方向寄存器？我们如何使用它们？为什么我们有它们？
- 是否可以使用相同的引脚作为输入和输出？
- 头文件 **Reflectance.h** 中有哪些信息？代码文件 **Reflectance.c** 中有什么？这种抽象概念有什么好处？
- **MSP432** 认为哪些电压较低？**MSP432** 认为哪些电压较高？
- 逻辑分析仪和示波器之间有什么区别？
- 什么是单调性，为什么它很重要？

6.7 其它挑战

在这章节中，我们列出了你可以做的其他活动，以进一步探索此模块的概念。你可以扩展系统或提出完全不同的东西。例如，

- 你可以用这种方法来测量滑块或操纵杆的位置。将电容器与电位器并联，使用引脚作为输出对电容充电，然后使用引脚作为输入，看看电容器放电需要多长时间。
- 开发替代方法，将 8 个传感器读数集成到一个参数中，该参数确定机器人在线路上的相对位置。

- 向 **Reflectance_Position** 添加验证检查，以确保传感器读数是四种预期模式之一。特别是，添加检查以确定机器人是否位于宝藏之上（参见图 2 的末尾）。

6.8 接下来是哪些模块？

GPIO 引脚是将数据输入微控制器或从微控制器输出数据的简单但常用的方法。除了这种循线传感器，GPIO 将用于机器人，用于碰撞传感器，电机方向，LCD 输出，转速计输入，超声波 I/O，蓝牙低功耗（BLE）和 wifi。以下模块将基于此模块构建：

Module 7) 研究有限状态机作为控制机器人的方法

Module 8) 将实际开关和 LED 连接到微控制器
这将允许更多的输入和输出增加系统的复杂性。

Module 9) 开发简单的 PWM 输出以调整占空比

Module 10) 学习 SysTick 周期中断，所以这些测量以非常省时的方式在后台进 r

Module 12) 将循线传感器和电机连接到机器人

6.9 您应该已经学会

在这章节中，我们将回顾您应该在本单元中学到的重要概念如何：

- 使用函数提供软件抽象感念
- 在 C 中执行条件，循环和计算
- 使用调试器单步执行并可视化 I/O 寄存器
- 设置 GPIO 引脚的方向寄存器
- 对 I/O 寄存器执行友好访问
- 可视化输入引脚将电压转换为二进制
- 使用示波器可视化时间行为
- 执行增量设计和测试
- 进行实验以确定准确性，单调性，特异性和噪音
- 将多个传感器测量值集成到一个值中

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