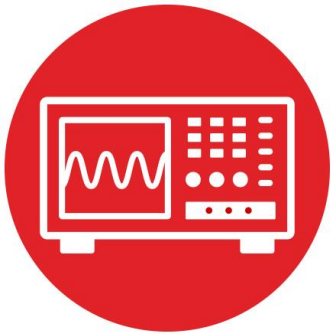

模块 9

实验： SysTick 定时器

实验：SysTick 定时器

9.0 目标

本实验的目的是学习如何使用 SysTick 定时器来管理时间。

1. 您首先将实现准确的时间延迟。
2. 您将使用时间延迟来创建 PWM 输出。
3. 使用硬件低通滤波器，您将使用 PWM 来实现 DAC。

小知识： 定时器，如 SysTick，在机器人中用于管理时间。SysTick 将用于定期执行任务。机器人中使用的超声波传感器通过测量声音传播所需的时间来计算距离，从表面反射，然后返回传感器。PWM 的概念将用于向机器人电机施加可变功率。

9.1 入门

9.1.1 从下面的软件工程起

浏览以下 2 个工程：
SysTick （使用 SysTick 定时器的示例）
Lab_SysTick （本实验开始的项目）

9.1.2 参考资料

MSP432P4xx Technical Reference Manual (SLAU356)
Meet the MSP432 LaunchPad (SLAU596*)*
MSP432 LaunchPad User's Guide (SLAU597*)*
MSP432P401R Datasheet, msp432p401m.pdf (SLAS826)
CarbonFilmResistor.pdf, resistor datasheet
CeramicCapacitor.pdf, capacitor data sheet

9.1.3 阅读材料

Volume 1 Sections 4.4 and 8.7
Embedded Systems: Introduction to the MSP432 Microcontroller",
或
Volume 2 Sections 2.6 and 6.3
Embedded Systems: Real-Time Interfacing to the MSP432 Microcontroller"

9.1.4 本实验所需组件

数量	组件描述	制造商	型号
1	MSP-EXP432P401R LaunchPad	TI	MSP-EXP432P401R
1	Ceramic capacitor, 0.47 µF		
1	Carbon 1/6W, 5%, 470 Ω		
1	solderless breadboard		

9.1.5 所需实验设备

电压表
示波器 （ 至少 1 个 10 kHz 采样的通道 ）
逻辑分析仪 （ 至少 4 个 10 kHz 采样的通道 ）

9.2 系统设计要求

在实验的第一部分中，您将使用 TI Launchpad 开发套件上的红色 LED 生成心跳波。 您将使用该概念并生成 PWM DAC。

LED 将从明亮到昏暗到暗淡的震荡，几乎呈现出正弦波。所以 LED“看起来”就像呼吸一样。本实验将使用 LaunchPad 上的两个开关来激活和停用心跳。

- 当操作者按下 SW1 时，心跳激活（并且无限期地继续）
- 当操作者按下 SW2 时，心跳停用

操作者可以多次按下开关，并且如上所述心跳应该开始和停止。如果您可以在心跳处于活动状态时忽略启动按钮，并在心跳处于非活动状态时忽略停止按钮。

实验：SysTick 定时器

基本思路是使用 SysTick 等待功能在 P1.0 上创建数字输出信号。有效时，此信号的周期应固定为 10,000µs。但是，软件会调整占空比作为控制 LED 亮度的方法。设 H 为 LED 开启的时间，L 为 LED 关闭的时间。该软件将保证 H + L 总是为 10000µs。但是，当激活时，H 将在 100 到 9900 之间变化。占空比定义为

Duty = H/(H+L) = H/10000

LED 的亮度与占空比线性相关。为了让心跳更具有天赋，您将以正弦的方式震荡占空比周期，如图 1 所示。当占空比很大时，LED 会亮，当占空比为 50%时，LED 将变暗，当占空比较低时，LED 将熄灭。

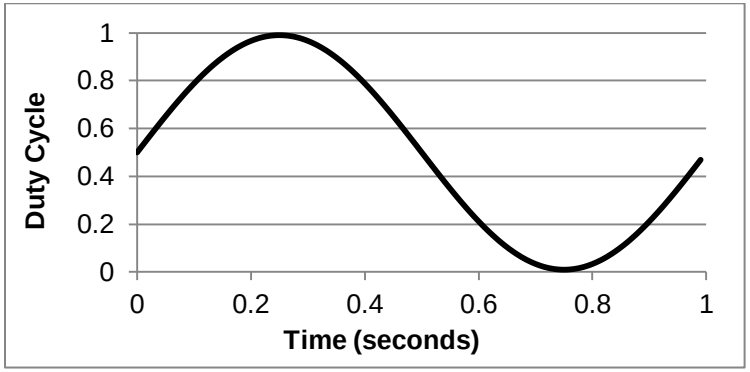

图 1. 占空比= H / 10000 作为时间的函数。

图 2 展示了 MSP432 LaunchPad。您将使用红色 LED 连接 P1.0 引脚。您将使用开关 1 连接 P1.1，开关 2（SWM2）连接 P1.4。

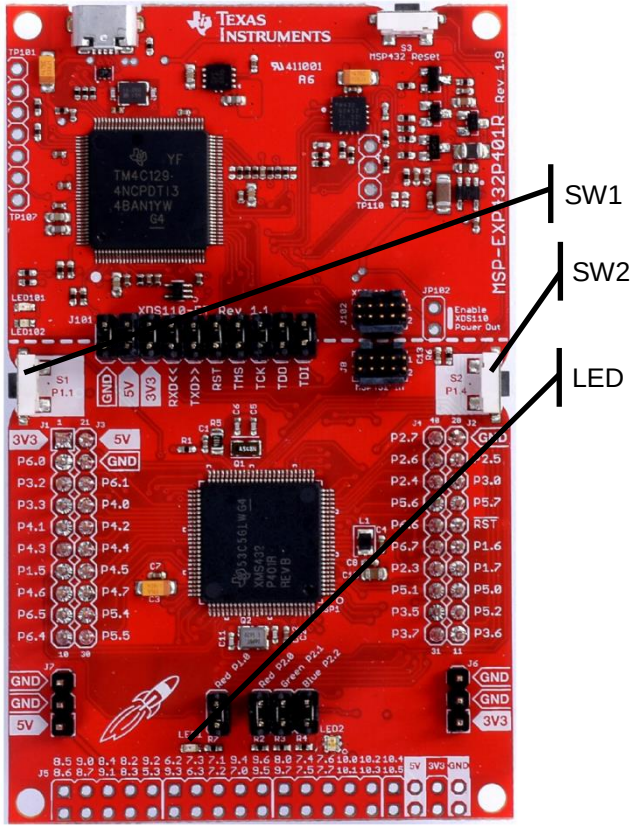

图 2. 本实验中使用的是没有外部电路的 LaunchPad。

实验：SysTick 定时器

9.3 实验准备

9.3.1 周期性心跳的硬件

LED 呼吸将通过 MSP432 LaunchPad 实现，无需额外的电路，请参见图 3。

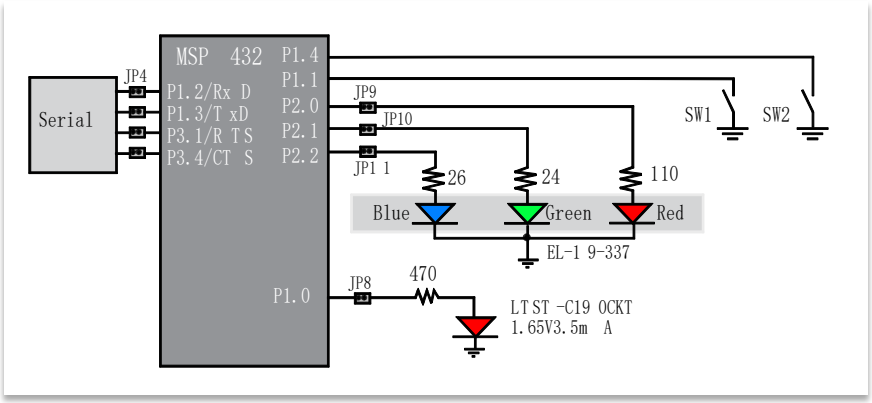

图 3. 在 P1.0 上创建周期性心跳。

9.3.2 PWM DAC 的硬件

要实现 PWM DAC，您需要构建模拟低通滤波器。电压表和示波器应连接在电容器两端，见图 4。

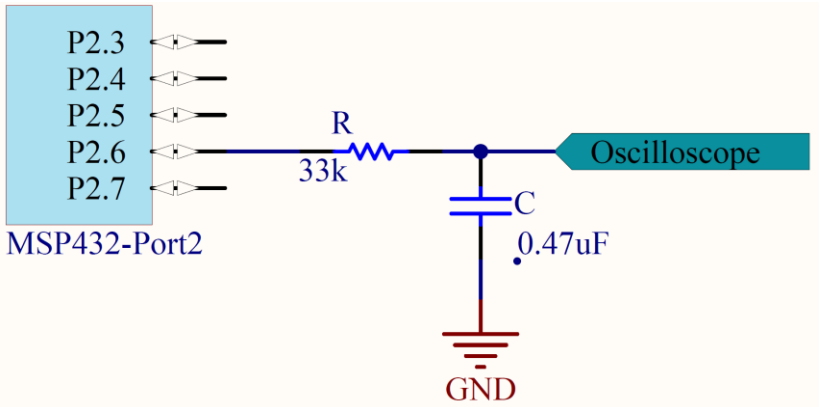

图 4. 使用外部无源 10 Hz 模拟低通滤波器将 PWM 信号（本例中为 P2.6）转换为 DAC 模拟输出电压。

9.4 系统设计要求

9.4.1 SysTick 等待

第一步是编写，开发和测试 SysTick 等待函数。

以下是初始化 **SysTick** 的软件驱动程序函数。 在这个实验里，我们不会使用中断。此初始化函数在主程序开始时调用一次，但在软件使用 **SysTick** 之前调用。

该函数的原型是：

void SysTick_Wait1us(uint32_t delay);

其中延迟是指定的等待时间，单位为 µs。您可以假设延迟大于 2µs 且小于 349,000µs。

```
// SysTick Initialization
void SysTick_Init(void){
  SysTick->LOAD = 0x00FFFFFF;   // maximum reload value
  SysTick->CTRL = 0x00000005;   // enable, no interrupts
}
```

实验：SysTick 定时器

SysTick 等待函数的步骤顺序如下：

1. 将所需值写入 SysTick LOAD 寄存器
2. 清除 VAL 计数器值，该值也会清除 COUNT 位
3. 等待 SysTick CTRL 寄存器中的 COUNT 位置 1

Program9_1 显示了如何通过创建 75％占空比数字输出来测试等待功能。使用逻辑分析仪或示波器验证等待功能的正确时序。高电平信号约 7.5 ms，低电平信号约 2.5 ms。

```
int Program9_1(void){

  Clock_Init48MHz();   // makes bus clock 48 MHz
  SysTick_Init();
  LaunchPad_Init();    // buttons and LEDs
  TExaS_Init(LOGICANALYZER_P1);
  while(1){
    P1->OUT |= 0x01;     // red LED on
    SysTick_Wait1us(7500);
    P1->OUT &= ~0x01;    // red LED off
    SysTick_Wait1us(2500);
  }
}
```

9.4.2 生成 PWM 输出

第二步是扩展操作以实现具有正弦变化占空比的数字波。例如，如果 ***H*** = 5000，***L*** 为 5000， LED 将有 50% 的亮度。交替，如果 ***H*** = 100， ***L*** 为 9900，LED 将有 1% 亮度。输出固定频率和固定占空比的波形，主循环将按此顺序执行这四个步骤，一遍又一遍

1. 将 P1.0 置为高电平
2. 使用您的 **`SysTick_Wait1us`** 函数延时 ***H*** µs
3. 将 P1.0 置为低电平
4. 使用您的 **`SysTick_Wait1us`** 函数延时 ***L*** µs

PulseBuf 是一个基于 ROM 的表，包含 100 个脉冲次数，以 µs 为单位，构成正弦变化的占空比。因为 100*10 ms 是 1 秒，创建正弦变化心跳的一种方法是反复执行以下序列。如果一遍又一遍地执行步骤 1 – 7，每次使用新的 H 值循环时，LED 将以 1 Hz 的频率闪烁。

1. 查找新的 ***H= PulseBuf [i]*** 值
2. 计算 ***L*** = 10000-***H***
3. 将 P1.0 置为高电平
4. 使用您的 **`SysTick_Wait1us`** 函数延时 ***H*** µs
5. 将 P1.0 置为低电平
6. 使用您的 **`SysTick_Wait1us`** 函数延时 ***L*** µs
7. ***i*** = ***i*** +1，如果 ***i*** ==100，则使 ***i*** =0

```
// Array used in this lab to create sine wave
const uint32_t PulseBuf[100]={
    5000, 5308, 5614, 5918, 6219, 6514, 6804, 7086,
    7361, 7626, 7880, 8123, 8354, 8572, 8776, 8964,
    9137, 9294, 9434, 9556, 9660, 9746, 9813, 9861,
    9890, 9900, 9890, 9861, 9813, 9746, 9660, 9556,
    9434, 9294, 9137, 8964, 8776, 8572, 8354, 8123,
    7880, 7626, 7361, 7086, 6804, 6514, 6219, 5918,
    5614, 5308, 5000, 4692, 4386, 4082, 3781, 3486,
    3196, 2914, 2639, 2374, 2120, 1877, 1646, 1428,
    1224, 1036,  863,  706,  566,  444,  340,  254,
     187,  139,  110,  100,  110,  139,  187,  254,
     340,  444,  566,  706,  863, 1036, 1224, 1428,
    1646, 1877, 2120, 2374, 2639, 2914, 3196, 3486,
    3781, 4082, 4386, 4692};
```

使用示波器或逻辑分析仪测试您的解决方案。

但请注意，这种创建 PWM 输出的方法需要处理器的全部注意力。一旦我们开始将模块放在机器人上， 我们将使用硬件定时器 （在**定时器** 模块）创建 PWM 输出，因此 PWM 的生成不需要专门关注软件。但是，目前的目标是简单地理解 PWM。

9.4.3 添加开关功能

第三步是添加开关功能，这样一个开关启动心跳，另一个开关停止心跳。在本实验中，开关去抖无关紧要，因为您可以在心跳处于活动状态时忽略启动按钮，并在心跳处于非活动状态时忽略停止按钮。

9.4.4 创建 PWM DAC

第四步是使用 PWM 输出设计数模转换器。本节有两个动机。首先，DAC 本质上是有用的器件，使用 PWM 实现 DAC 可为低于 1 kHz 的信号提供低成本，高分辨率的实现。其次，本节中的 RC 电路模拟了电机的行为，因此我们可以认为该电路的电压输出类似于输出到电机的功率。

回想一下数字波的频率是 100 Hz。在 PWM 方法中，频率将是固定的。LED 确实足够快，能够对我们创建的这个方波做出响应。请查看 HLMP-4700 LED 的数据表。您会发现它的响应时间为 90 ns。因此，当以 100 Hz 运行时，LED 将完全打开和完全关闭。

然而，我们的眼睛无法检测到 100 Hz 的波，这就是为什么我们的眼睛在 75%的亮度下感知 75%的占空比波。我们可以使用 PWM 来控制与 100 Hz 方波相比响应缓慢的其它设备。如果器件的时间常数与 PWM 频率相比较慢，则器件响应平均信号（H /（H + L））而不是瞬时接通和关断。要在另一个示例中看到这种强大的 PWM 方法，我们需要将输出移到未使用的引脚，因此引脚不会连接到任何 LED 电路。慢速器件的一个例子是用电阻器和电容器实现的模拟低通滤波器，如图 4 所示。滤波器的截止频率为

$$f_c = 1/(2\pi RC)$$

为了使其工作，我们需要 1 Hz <fc <100 Hz，因此电路通过 1 Hz 方波并拒绝（或平滑）100 Hz 方波。事实上，我们的眼睛有一个大约 10 Hz 的截止值。所以，我们会选择

$$RC = 1/(2\pi 10\text{Hz}) \approx 0.016 \text{ sec.}$$

一种可能的组合是 R=33 kΩ，and C = 0.47 µF。它也适用于 R = 3.3 kΩ，C = 4.7 µF。

Warning： 选择大于 3.3kΩ 的电阻值，以将电流限制在 3.3V /3.3kΩ= 1 mA 以下。此外，我们建议选择一个远小于示波器探头输入阻抗的电阻值。

使用电压表静态测试 PWM 实现的数模转换器。将电压表连接到图 3 中的电容器。**Program9_2** 在 P2.6 上实现具有已知占空比（H /（H + L））的 100Hz 方波。

```
int Program9_2(void){
  uint32_t H,L;
  Clock_Init48MHz();   // makes bus clock 48 MHz
  SysTick_Init();
  TExaS_Init(SCOPE);
  P2->SEL0 &= ~0x40;
  P2->SEL1 &= ~0x40;  // 1) configure P2.6 as GPIO
  P2->DIR |= 0x40;    // P2.6 output
  H = 7500;
  L = 10000-H;
  while(1){
    P2->OUT |= 0x40;    // on
    SysTick_Wait1us(H);
    P2->OUT &= ~0x40;   // off
    SysTick_Wait1us(L);
  }
}
```

将该程序运行五个不同的工作周期，并将直流电压绘制为占空比的函数。 您的数据应该类似于图 5。

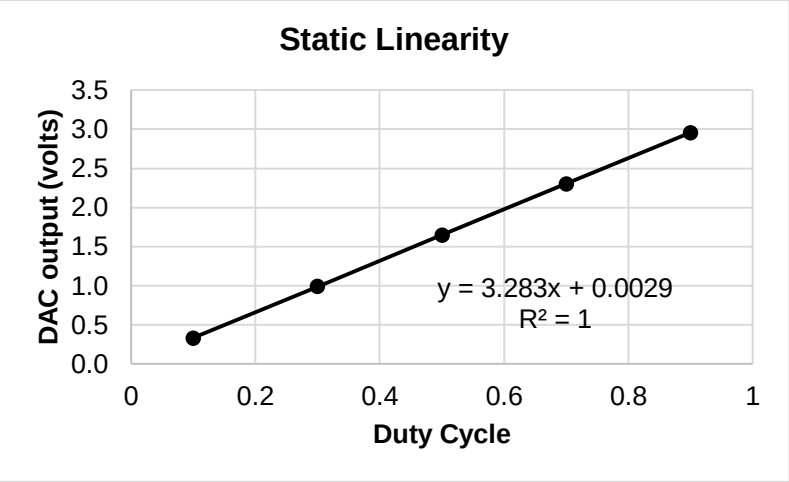

图 5. 显示 DAC 线性度的示例测量数据。

上图显示了 DAC 输出电压和占空比之间的关系。鉴于 PWM 的这种实现，您可以创建的不同占空比的数量称为 **精度** (具有备选单位)。通常我们定义位精度。

$$Bits = \log_2(Alternatives)$$

从理论上讲，如果有 10,000 个备选方案，则此 DAC 的等效位数约为 13。

分辨率 是可以创建的 DAC 电压的最小变化。如果 H 增加 1，在这种情况下，如果 H 增加 1，则 DAC 模拟输出（理论上）增加 3.3V / 10000 = 0.33 mV。

范围 是最大电压（3.3V）减去最小电压（0V）。请注意，范围（以 V 为单位），精度（以 bits 为单位）和分辨率（以 V 为单位）是相关的。

$$Range = 2^{Precision} * Resolution$$

接下来，让我们测量从图 4 构建的电路的实际系统性能，并比较实际值和理论值。

测试 (i) 使用 **Program9_2**，设置 H=9000 和 L=1000。这将占空比设置为 90%。然后使用电压表测量 DAC 的直流电压。电容器上的电压应约为 0.9*3.3V。 设 *S* (信号) 为该直流电压测量值。

接下来，在不改变占空比的情况下，更改电压表设置并测量 DAC 的交流电压。设 *N* (噪声) 为伏特的这种测量值。计算信噪比为 *SNR = S/N* 。在此测量中，我们将 RMS AC 电压定义为 DAC 的分辨率。同样，我们将 DAC 范围近似为 90%的值。此，考虑噪声的等效位数是

$$Precision\ (bits) = \log_2 S/N$$

测试 (ii) 对于与图 4 所示相同的电路，我们将使用示波器。将示波器探头连接到电容器上。现在运行正弦变化的负载系统并观察示波器上的输出。图 6 显示了使用 TExaS 示波器测量的典型模拟输出。图 6 显示滤波器未移除所有 100 Hz 组件；它确实通过了 1 Hz，但也通过了一些 100 Hz。由 PWM 信号引起的信号中存在大的 100Hz 分量。如果您的示波器具有频谱分析仪功能，您可以使用它来查看由 PWM 频率引起的 100 Hz 幅度。

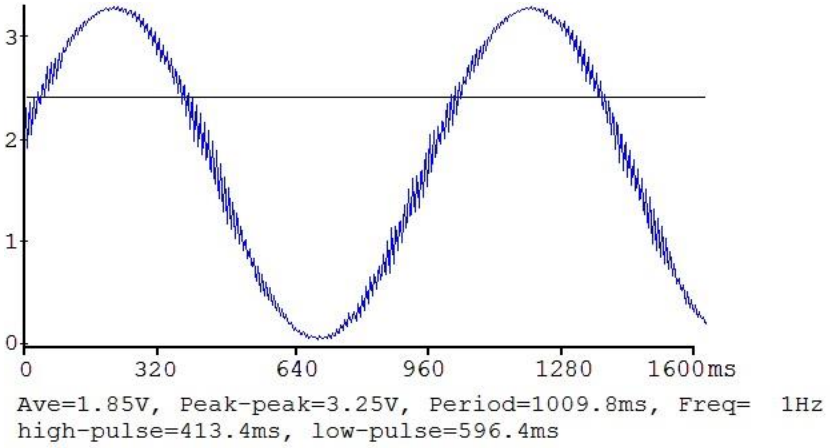

图 6. PWM DAC 的示例模拟输出。

9.5 疑难解答

无法编程 LaunchPad：

- 检查 LaunchPad 开发板上的电缆和跳线。
- 检查 Windows 驱动程序以查看操作系统是否识别该板。

- 在此计算机上尝试另一个 LaunchPad。 在另一台计算机上试用此 LaunchPad。

SysTick 延迟不正确：

- 确保 MSP432 时钟以 48 MHz 运行。
- 确保 SysTick 函数中使用的所有整数都适合 24 位（小于 16,777,216）。

LED 不变暗：

- 测量示波器或逻辑分析仪上的端口输出。确保频率固定，但占空比变化。
- 将输出回送到微控制器的两个引脚（将相同值输出到两个引脚）。 您的端口引脚可能已损坏。

DAC 不是模拟的：

- 验证电阻和电容值。Calculate f=1/(2πRC)， f 应介于 1 和 100 Hz 之间。

9.6 请思考

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

- 精度是可以生成的不同 PWM 输出的数量。该实验描述了系统能够创建大约 10,000 个不同的 PWM，相当于大约 13 位。你能做些什么来提高精度？
- PWM 周期（10ms），SysTick 定时器等待（1us）的分辨率与 PWM 精度之间的关系是什么？ 给出这种关系的等式。
- 如果您尝试将 PWM 周期设置为大于 350ms，您的执行会发生什么？RC 电路以何种方式模拟（表示）我们眼睛和大脑的视觉处理行为？
- 为什么 RC 电路被归类为模拟低通滤波器？

您将如何通过实验确定视觉系统的频率响应？其中一个早期的口袋妖怪动漫节目有一个 5-sec 12 Hz 的场景，引起了儿童的神经反应（搜索“口袋妖怪诱发的癫痫发作”）。

9.7 其它挑战

在本节中，我们列出了您可以执行的其他活动，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

- 通过减少定时器等待功能的单位来提高精度（例如，通过将定时器等待从 1us 减少到 250ns，从 13 位变为 15 位）。
- 通过从 100 Hz PWM 切换到 1000 Hz PWM（精度将从 13 位降至 10 位）消除 PWM DAC 中的噪声（图 5 中的 100 Hz 纹波）。
- 使用硬件定时器实现本实验（我们最终会切换 PWM 以使用定时器）。

9.8 接下来是哪些模块？

事实证明，使用机器人的电机也具有大约 10 Hz 的时间常数。我们将使用 PWM 输出来允许软件设置传送到电机的功率。但是，我们不能使用 100％的处理器时间来为我们的机器人实现 PWM。因此，我们将使用内置于 MSP432 微控制器的硬件定时器，因此软件可以自由执行其他任务，而硬件则自动生成 PWM。然而，软件将设置一次周期，并动态调整占空比以控制机器人的行为。这些是未来模块，它基于在这个模块中学习的概念。

模块 10) 使用软件阵列验证系统的正常功能
模块 12) 使用此 PWM 输出调整机器人上直流电机的功率
模块 13) 使用周期性中断在硬件中创建 PWM 输出

9.9 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：

- 测量电阻和电压
- 使用逻辑分析仪和示波器测量时间
- 创建准确的时间延迟
- 实现 PWM 输出
- 使用 PWM 输出创建时变行为
- 创建一个简单的模拟低通滤波器
- 平衡范围，分辨率和精度之间的权衡

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/sampterms.htm).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265