



模块 5

简介：循线迷宫-有限状态机



简介：循线迷宫-有限状态机

教学目标：

回顾 C 程序设计

理解 变量，数字，指针，结构体，数组

开发 调试技术

学习 如何解决有限状态机的问题

设计、构建和测试一个系统

控制循线机器人

需要预先学习的模块：模块 1，和 4

- 使用 CCS 在启动板上运行代码(模块 1)
- GPIO (模块 4)

推荐阅读材料：

- Volume 1 Sections 6.1, 6.2, 6.4 and 6.5
Embedded Systems: Introduction to the MSP432 Microcontroller
ISBN: 978-1512185676, Jonathan Valvano, copyright (c) 2017
- Volume 2 Section 3.5
Embedded Systems: Real-Time Interfacing to the MSP432 Microcontroller, ISBN: 978-1514676585, Jonathan Valvano, copyright (c) 2017

软件抽象允许我们用一套基本抽象原则定义复杂问题。然后，我们可以使用这些抽象构建块构建系统解决方案。使用抽象可使我们更好地理解问题及其解决方案。这是因为我们可以将系统所做的事情（策略）与系统工作方式（机制）的细节分开。这种分离首先是通过描述系统的功能简化了设计过程，然后我们可以将描述转换为实现该描述的系统。抽象提供了正确的函数证明，并简化了扩展和定制。本节介绍的抽象是有限状态机（FSM）。FSM 开发的抽象原则是输入，输出，状态和状态转换。FSM 状态转换图（STG）定义了其输入和输出之间的时间依赖关系。如果我们解决复杂问题并将其映射到 FSM 模型，那么我们可以使用简单的 FSM 软件工具来解决它。我们的 FSM 软件实现将易于理解，调试和修改。

该问题被映射到一个定义良好的模型，该模型具有一组抽象但功能强大的规则。然后，软件解决方案是实现模型规则的关键。在我们的例子中，一旦我们证明我们的软件正确地解决了一个 FSM，那么我们就可以对状态转换图进行更改，并确信我们的软件解决方案正确地实现了新的 FSM。

嵌入式系统通常部署在安全关键系统中。在这些情况下，我们必须证明解决方案完全符合预期。像有限状态机（FSM）这样的抽象方法允许我们将它的作用与它的工作原理分开。FSM 的复杂性在状态转换图中，而控制器应该是非常简单的。一旦我们认证低级控制器正常运行，我们就可以在高级或抽象级别中验证系统。在与该模块相关的实验室中，我们将使用有限状态机为一个简单的循线机器人创建控制器。输入将来自两个开关（模拟两个线路传感器），输出将转到两个 LED（在一个差动驱动机器人上模拟两个电机）。控制器的目标是遵循直线。本实验的目的是提供另一个 C 编程实验，并作为机器人控制的简介。在之前的模块 4（GPIO）中，您连接了一个实际的循线传感器。其它实验将为机器人控制器提供额外的传感器。在 8（ADC）中，您将添加红外距离传感器。在 3（定时器）中，您将连接机器人电机，这将是真正的 FSM 控制器的输出。

系统开发的基本方法是创建组件，然后将组件拼凑在一起以创建系统。在本单元中，您将学习如何将 FSM 用作系统的中央控制器。

