



# 模块 9

简介：UART 通信



# 简介：UART 通信

## 教学目标：

- 理解** 先进先出队列的操作和使用
- 连接** 使用串行接口将机器人连接到 PC
- 创建** 两个先进先出队列
- 设计** 一个命令解释器，用于协助机器人挑战

- 需要预先学习的模块**（模块 10）
- 使用 SysTick 中断（模块 10）

在本模块中，您将使用通用异步接收器/发送器（UART）开发中断设备驱动程序。该串行端口允许微控制器与其它设备进行通信，例如其它计算机，输入传感器以及输出显示器。串行传输涉及一次发送一个比特（bit），这样数据就会随时间展开。每秒传输的总位数称为**波特率（baud rate）**。图 1 显示了 MSP432 发送一个字节数据时产生的波形。

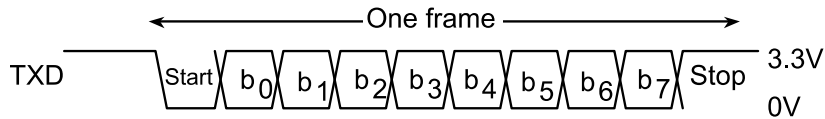


图 1. 一个串行数据帧，包含 1 个起始位，8 位数据，1 个停止位和 1 奇偶校验位。

先进先出循环队列（FIFO）对于实现缓冲 I/O 接口非常有用。它可用于缓冲输入和缓冲输出。保存顺序的数据结构临时保存源（producer）创建的数据，然后再由接收器（consumer）进行处理。在该模块中开发的 FIFO 类为静态分配的全局结构。因为它们是全球变量，也就是说他们将永久存在并且可以被多个线程共享。使用 FIFO 结构解决数据流问题的优点是我们可以解耦生产者线程和使用者线程。如果没有 FIFO，我们每次只能生成一个数据，对该数据进行处理，然后才能生成第二个数据并处理。使用 FIFO，生产者线程可以继续生成数据，而无需等待使用者处理完之前的数据。这种去耦可以显著提高系统性能。

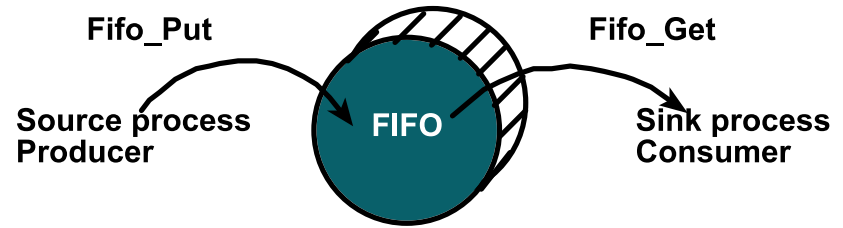


图 2. FIFO 用于缓冲生产者和使用者的数据。

由于多个线程正在访问共享的全局结构，因此考虑临界部分非常重要。如果 put 和 get 函数都读取 - 修改 - 写入相同的全局变量，则如果一个函数启动（读取），被第二个函数中断，第二个函数执行，然后第一个函数完成（修改 - 写入），则会发生错误。例如，此方法有一个临界部分，具有对 **Size** 的共享访问权限。

```
Put
Size=Size+1;
Get
Size=Size-1;
```

总之，UART 驱动程序将使用 FIFO 队列的中断同步。这种缓冲方法将分离数据的生成和消耗。例如，主程序可以生成要打印（print）的数据，并将其放入 FIFO 队列中。当发送 UART 硬件空闲时，ISR 可以从 FIFO 获取数据并将数据写入硬件。因此对于复杂系统来说，缓冲数据是一种有效的机制。