TI-RSLK MAX

Texas Instruments Robotics System Learning Kit

Texas Instruments

Module 18

Introduction: Serial Communication

Introduction: Serial Communication

Educational Objectives:

UNDERSTAND The operation and use of first in first out queue
INTERFACE The robot to the PC using a serial channel
CREATE Two first in first out queues
DESIGN A command interpreter to assist in the robot challenge

Prerequisites (Module 10)
- Interrupts using SysTick (Module 10)

Recommended reading materials for students:
- Chapter 18, **Embedded Systems: Introduction to Robotics,** Jonathan W. Valvano, ISBN: 9781074544300, copyright © 2019

In this module you will develop an interrupting device driver using the Universal Asynchronous Receiver/Transmitter (UART). This serial port allows the microcontroller to communicate with devices such as other computers, input sensors, and output displays. Serial transmission involves sending one bit at a time, such that the data is spread out over time. The total number of bits transmitted per second is called the **baud rate**. Figure 1 shows the waveform produced when the MSP432 sends one byte of data.

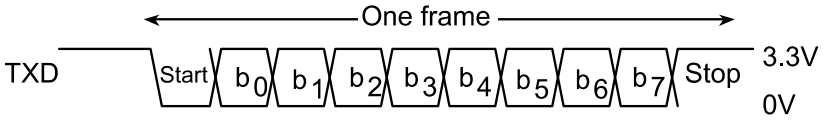


Figure 1. A serial data frame with 1 start bit, 8-bit data, 1 stop bit, and no parity bit.

The first in first out circular queue (FIFO) is quite useful for implementing a buffered I/O interface. It can be used for both buffered input and buffered output. The order preserving data structure temporarily saves data created by the source (producer) before it is processed by the sink (consumer). The class of FIFOs developed in this module will be statically allocated global structures. Because they are global variables, it means they will exist permanently and can be carefully shared by more than one thread. The advantage of using a FIFO structure for a data flow problem is that we can decouple the producer and consumer threads. Without the FIFO we would have to produce 1 piece of data, then process it, produce another piece of data, then process it. With the FIFO, the producer thread can continue to produce data without having to wait for the consumer to finish processing the previous data. This decoupling can significantly improve system performance.

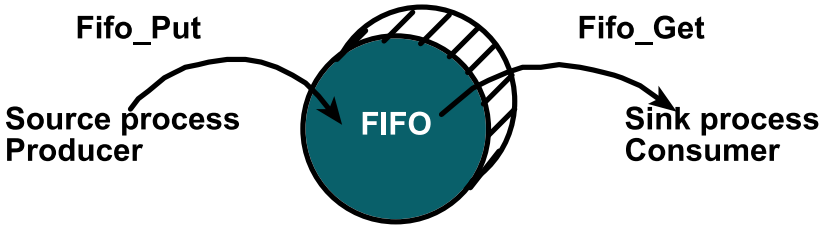
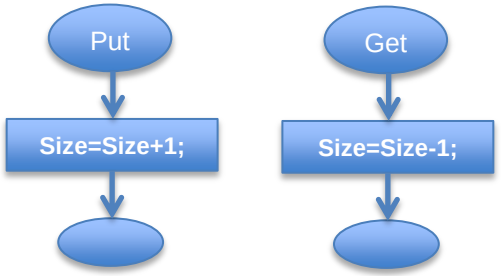


Figure 2. The FIFO is used to buffer data between the producer and consumer.

Because multiple threads are accessing the shared global structure, it is important to consider critical sections. If the put and get functions both read-modify-write the same global, an error would occur if one function started (read), was interrupted by the second, the second executes, and then the first completes (modify-write). For example, this approach has a critical section with the shared access to **Size**.



In summary, the UART driver will use interrupt synchronization with FIFO queues. This buffered approach will decouple the production of data from the consumption of data. For example, the main program can generate data to print, and put it into the FIFO queue. When the transmit UART hardware is idle, the ISR can get from the FIFO and write the data to the hardware. Buffering data in this matter is an efficient and effective mechanism for complex systems.

ti.com/rslk

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

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