INTRODUCTION

The Serial Peripheral Interface (SPI) is an embedded hardware unit in charge of communicating with external peripheral devices and microcontrollers serially. The SPI standard bus was established by Motorola and is supported by various integrated circuit manufacturers. It is based upon a shift register that the user can access in order to send or receive data bytes at a high bit rate. SPI communication is synchronous in the sense that data bits are transmitted and received along with a synchronizing clock. Hence, the receiver does not need to know the bit rate a priori as is the case with the EUSART. The SPI is part of the Master Synchronous Serial Port (MSSP) module on PIC18 microcontrollers. This module may also be used in I2C mode covered in Chapter 14. The SPI is rather simple to work with and is very convenient for low-pin count microcontrollers. For instance, a serial 16-bit A/D converter usurps only 2 microcontroller pins: the data line and the clock line. This is a great advantage in comparison with a parallel 16-bit A/D converter that would appropriate 16 I/O pins.

THE BASIC PRINCIPLE

SPI communication consists of a so-called master and one or more slaves. The master is the communication end that drives the clock line in order to exchange data with the slave. Figure 1 shows a simple configuration depicting only one slave. When the master decides to communicate with the slave, it asserts the Slave Select SS pin. Consequently, the slave leaves the high-impedance state and becomes electrically connected to the serial bus. Upon writing a byte to the master’s transmit register, an internal shift register pops out these bits one at a time starting with the MSB. The bits transferred from the master on SDO (Serial Data Out) or MOSI (Master Output Slave Input) will in turn “kick-out” the bits residing in the slave’s shift register (see Figure 2). The bits out of the slave appear on the master’s SDI (Serial Data In) or MISO (Master Input Slave Output). In brief, the shift registers of both the master and the slave act like one circular shift register.

DOI: 10.4018/978-1-68318-000-5.ch013
Although the master always initiates data transfer, communication between the master and the slave may take one of the following forms:

- The master sends a dummy byte to the slave. In return, the slave pops out data residing in its shift register. This data is received by the master. The slave may be an analog-to-digital converter (or any other input device) with serial interface. Such a configuration requires the SCK line between master and slave to be connected. The MOSI line becomes redundant and may be disconnected.
- The master sends useful data to the slave which in turn ejects a dummy byte from its shift register. This configuration is applicable whenever the slave is an output device such as a digital-to-analog converter. No physical connection of the MISO line between master and slave is required in this case.
Related Content

Input/Output Ports
www.igi-global.com/chapter/inputoutput-ports/190446?camid=4v1a

The Software Enterprise: Preparing Industry-Ready Software Engineers
www.igi-global.com/chapter/software-enterprise-preparing-industry-ready/29596?camid=4v1a

From Use Case to Use Case Slices: An Automated Approach
Yangyang Chen, Cong Chen, Hao Wen, Jian-min Jiang, Qiong Zeng, Hongping Shu and Zhong Hong (2019). International Journal of Information System Modeling and Design (pp. 24-50).
www.igi-global.com/article/from-use-case-to-use-case-slices/243438?camid=4v1a

A Test-Driven Approach to Behavioral Queries for Service Selection
www.igi-global.com/chapter/test-driven-approach-behavioral-queries/77715?camid=4v1a