

Experiment 14 – Extra Credit –Arduino Microcontroller II: Communicating via the SPI Interface

Introduction

The SPI, or Serial Peripheral Interface, is a 4-wire protocol for communication between a microcontroller (like the Arduino's ATmega328P) and another chip. The four signals are SS (Slave Select, to enable the external chip), SCK (Serial Clock, for synchronizing the transfer of data), MOSI (Master Out Slave In, for commands from the master, or microcontroller, to the slave, or external chip), and MISO (Master In Slave Out, for data from the slave to the master). In this experiment, you will use only the first three signal lines for communications from the Arduino to the MCP41010, a 10-k Ω potentiometer (variable resistor). The resistance of the MCP can be set anywhere within its range to 8-bit precision, in accordance to commands from the Arduino over the SPI interface.

The Assignment

Your challenge is connect the Arduino and the MCP41010 via the SPI interface and to write a control program that allows the user, via the Serial Monitor window, to set the resistance in ten increments (zero, 10% of full scale, 20% of full scale, etc.) The resistance can be measured by a voltmeter connected between the wiper and one fixed end of the resistance on the MCP chip. Demonstrate the successful operation of your circuit to the instructor to collect your Extra Credit!

Advice

You can find an explanation of the Arduino's SPI interface on the arduino.cc site under Documentation→Reference→Libraries→SPI. You can find the datasheet for the MCP41010 chip on the jameco.com site. The important parts to look at are the pinout diagram and the description of command format the chip expects.

As with the previous Arduino lab, planning to complete the project in stages will aid in debugging. First, check that the Arduino can send data back and forth over the Serial Monitor. Then wire up the MCP and see if Arduino can set a fixed midrange resistance value (that is, don't worry about using the Serial Monitor yet). Finally, extend your program so the resistance output can be controlled by the user via the Serial Monitor.