School of Advanced Technology

Lab 4: Serial to Parallel Conversion, Synchronous Interrupts

Lab Objectives:

1. Show basic control of a single bit travelling through a set of eight LEDs and one shift register
2. Implement a bar graph level meter, consisting of sixteen LEDs.
3. Run a 2-digit 7-segment display from two shift registers.
4. Implement a synchronous interrupt.

Bonus demo:

5. Manually clock in an 8-bit number into the shift register.

Required Equipment:

- Two 8-bit shift register (2 x SN74HC595)
- Sixteen LEDs: green (4 x), yellow (4 x), red (8 x) – or your choice
- Two common anode 7-segment displays
- Resistors – suitable for maintaining LED currents to the limits specified for the shift register
- Teensy board, USB cable, various connecting wires
- Computer with Arduino IDE & Teensy extensions installed and working

References and Resources:

- Theory class circuit schematic.
- Course textbooks: “Beginning Arduino” and “Getting Started with Arduino”, or other…
- Parts List file with part numbers, for referencing data sheets and the corresponding part data sheet
- http://arduino.cc/en/tutorial/ShiftOut - very handy tutorial from Arduino reference pages
- Teensyduino IntervalTimer https://www.pjrc.com/teensy/td_timing_IntervalTimer.html

Task: Determine Operating Parameters for the 74HC595 Shift Register

1. Find the data sheet for the shift register (either online in Blackboard, or use the parts list to identify the part number and do an Internet search for a datasheet).
2. Using the datasheet, identify the pinout and the markings which designates pin 1.
3. Using the datasheet, identify the appropriate voltage & current parameters.
4. Use Ohm’s law and other formulae as required when determining a suitable value for any current limiting resistors for LEDs.

Demo #1: Master basic operation of a shift register using ***eight*** LEDs and ***one*** shift register.

1. Using the data sheet from the previous task, plan out the placement of a shift register with a set of 8 LEDs and their accompanying resistors. Shift register signals that you will need to control via the Arduino are SERial input (SER), the Shift Register clock (SRCLK), and the output Register clock (RCLK). Shift register signals that you will need to control using the power rails are OE and MRCLR.
2. Wire up a shift register and a set of eight LEDs.
3. Consider carefully whether the LSB or MSB needs to be loaded first! Implement the code to light up a single LED and make it "travel", bouncing from end to end at some pleasing speed.

Demo #2: Add 8 more LEDs and one more shift register

1. Wire up the second shift register with an additional 8 LEDs.
2. Implement a bar graph which displays a pattern smoothly varying between 0 LEDs lit–> All LEDs lit–> 0 LEDs lit. The LEDs are lit up one by one, until all of them are lit up, and finally count backwards, turning off all of the LEDs one at a time, until none of them are lit. The display should change at a speed which is clearly discernible.

Task: Determine Operating Parameters for 7-Segment Display

1. Find the data sheet for the 7-segment display and examine the data sheet.
2. Identify the pinouts and note whether the device is common-***anode*** or common-***cathode***. Find the required operating voltage and current parameters for this device and compare the current requirements with what you know about the specifications (current limits) of the ARM processor. Make an engineering decision regarding what current to design for. Use Ohm's law and other formulae as required when determining a suitable value for the current limiting resistor.

Task: Basic set-up and functional verification of the 7-segment display

1. Remove the LEDs and modify your circuit so that the shift registers are now connected to the 2x7-Segment displays. Install each 7-segment display with the five pin rows placed on opposite sides of the center gap of the protoboard. If uncertain about this step, look at the uploaded photo of a sample circuit, or ask your lab prof. Don't accidently short any pins to each other.
2. The data sheet of the 7-segment display specifies that pins 3 and 8 (the common anode) are internally connected to each other. Only one of these pins is required to be connected to the Vcc (power) rail.
3. Connect each of the each display's segments a-g, to the shift register outputs, with a current limiting resistor installed in series. One shift register will be connected to the7-segment display that is considered to be the "10's" display and the other shift register will be connected to the7-segment display that is considered to be the "1's" display.

Demo #3: Two Digit Decimal Counter

You will use **multiplexing** to control a pair of 7-segment displays. You will use the minimum possible number of teensy connections (3 connections) to control 15 segments (a-g and one decimal point).

1. Modify your program to see the two digits count from 00 – 99, pausing at each value long enough so that it is distinguishable from the next. There should be no flickering of lights.

Demo #4: Synchronous Interrupt

1. Modify your code from demo #3 by adding an interrupt. The interrupt will reset the count at a number to be determined by your lab instructor.

Deliverables:

1. Shift register single bit travelling up and down through 8 LEDs.
2. Shift register bar graph with 16 LEDs and 2 shift registers.
3. A pair of 7-segment digital displays, driven programmatically from two shift registers.
4. Operation of a synchronous interrupt to reset the 7-segment display count.

Bonus Demo: Manually Clock in an 8-bit number to Shift Register (2 bonus marks)

You are eligible for the bonus demo if the regular demonstrations 1-4 are complete and the demo is complete before the due date.

Refer to demonstration completed in theory class or this video:
https://www.youtube.com/watch?v=6fVbJbNPrEU

1. Connect the inputs of a shift register to switches that have been connected as either pull-up or pull-down as dictated by either an active low or an active high input. At the very minimum, the three shift register inputs that need to be wired to pushbuttons are SER, SRCLK and CLK.
2. The output enable and the master reset pins can be hardwired into either the Vcc or GND rails as needed instead of pushbuttons. Be certain to leave the master reset pin wire easily accessible and relatively longer – you will occasionally need move this wire to and from the Vcc and GND rails.
3. Clock in an 8-bit number and demonstrate.