Brandon Pollack

Lab5

Pre Prelab

My switch bounced quite a bit, sometimes up to 5 times. I added some nops to the interrupt subroutine to compensate for this. Even after adding many NOPs, it is worth noting that this switch is very bouncy, and the NOPs don’t compensate for bounce when pushing the switch up.

PortC pins 1,2,3 are used for USART communication to the USB port.

Pre Lab

1. I have to use a baud rate of 38400 because that is what the lab manual told me to use, although I could generate a baud rate all the way up to a maximum of 32\*2(if CLK2X is enabled) MHZ (limited by our Fper which is 2MHZ)
   1. Data- contains the data to be transmitted or the received data from the shift reg
   2. Status-contains interrupt flags, error flags, and the bit 8 bit if you are reicieving 9 bit transmission.
   3. CTRLA-controls the interrupt levels for Data Reg empty, tx sent, rx receive
   4. CTRLB-enable receiver/transmitter, double transmission speed, enabling bit 8, MPCM
   5. CTRLC, select IR, SPI, or UART, or Master SPI, Parity mode, stop bitmode, and character size
   6. BaudCTRLA- sets Bsel’s lower 8 bits
   7. BaudCTRLB- sets Bsel’s upper 4 bits and BSCL
2. Serial has one (or 2 rx and tx) data lines and and sends data over time. Parallel sends each bit on its own line.
3. Synchronous uses a clock to send the data, Asynchronous uses a predetermined Baud rate that must match at the receiving and sending end.
4. Double buffering allows you to hold your data, while still receiving the next transmission (or sending) without overwriting your data. Thus, you need not immediately read the data as it is received.

Problems encountered:

For some reason my IOPORT only works outside of the main loop, I am not sure why

Learned:

Serial communication is awesome.