Lab 2

Name: Dylan Christopherson

CEE-345 Microprocessor System Design

Introduction:

The goal of this lab was to code in C. We were to design the Knight Rider light display which sends a light back and forth across the LEDS and also a Traffic Light controller which was designed to display a certain pattern of lights based on traffic lights.

LDI – Stands for Load Immediate. This loads an 8 bit constant directly to register 16 to 31.

SPL – Stack Pointer Low

SPH – Stack Pointer High

RAMEND – A label that represents the last memory address in SRAM. It’s a 16 bit word so we use the functions to split 16 bits into 8 bits so it can be handled.

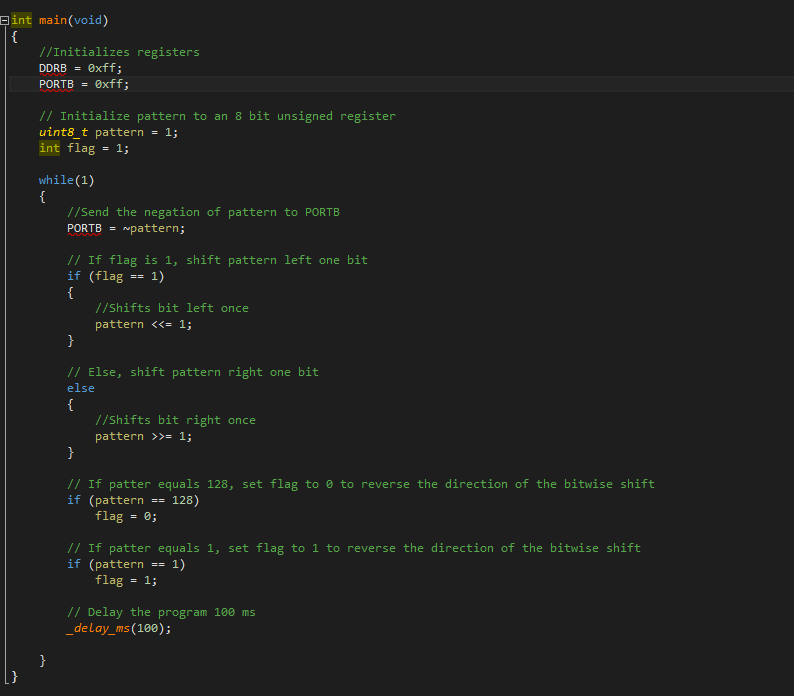
Using low() and high(), we can return the low byte and high byte respectively of a 16-bit word

rol (Rotate left): The MSB is rotated to the carry flag, the carry flag is rotated to the lsb, all other bits are shifted left. The carry flag is initially 0.

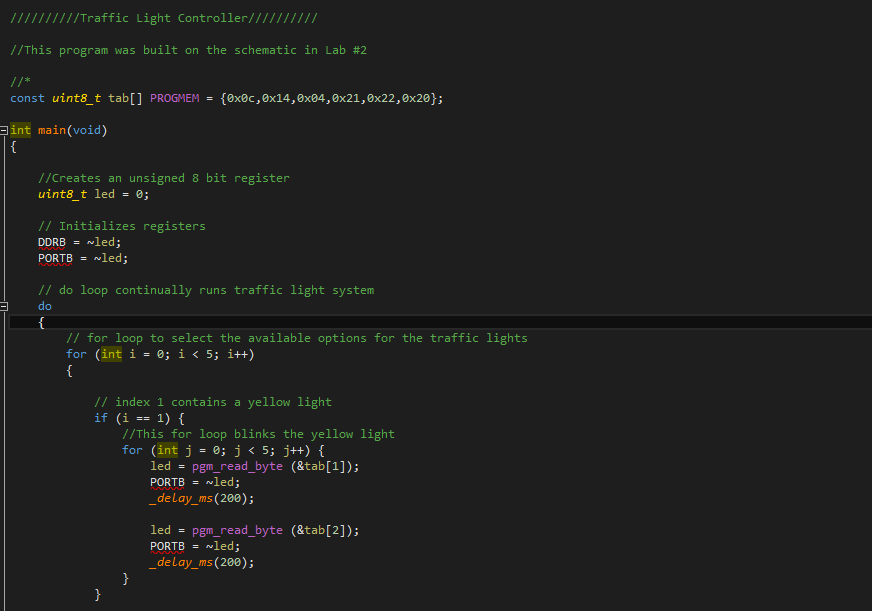
All of my comments for both programs are within the code as I wanted to have them available if I ever went back and used this code.

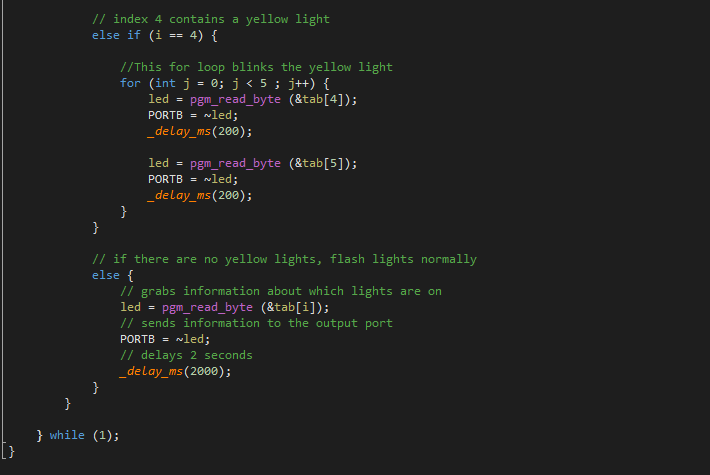
Knight Rider:





Traffic Light Controller:





There were some struggles from this lab. I’ve been refamiliarizing myself with C and its syntax and uses. This has been an excellent opportunity to focus my attention on it. I also had some troubles with making the LED flash 5 times when it turned on. I did this by having if statements recognize when we were at an index where a yellow light appeared. I then alternated the state when the yellow light was on, and a state that was similar but had the yellow light off. This caused the light to flash on and off quickly. I also divided the time into 5 segments so the yellow light was the same time as all of the other lights, but still blinked 5 times.