Lab 3

Name: Dylan Christopherson

CEE-345 Microprocessor System Design

2/18/2019

Introduction:

The goal of this lab was to implement a PmodKeypad using C and Assembly code. The user pressed a digit on the keypad which ranged from 0 to D. The output of the keypad was sent into DDRD. The AVR board parsed the data, and then sent the output out DDRB to the LEDs. The LEDs were blinked based on the number selected.

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.

brne: branch not equal

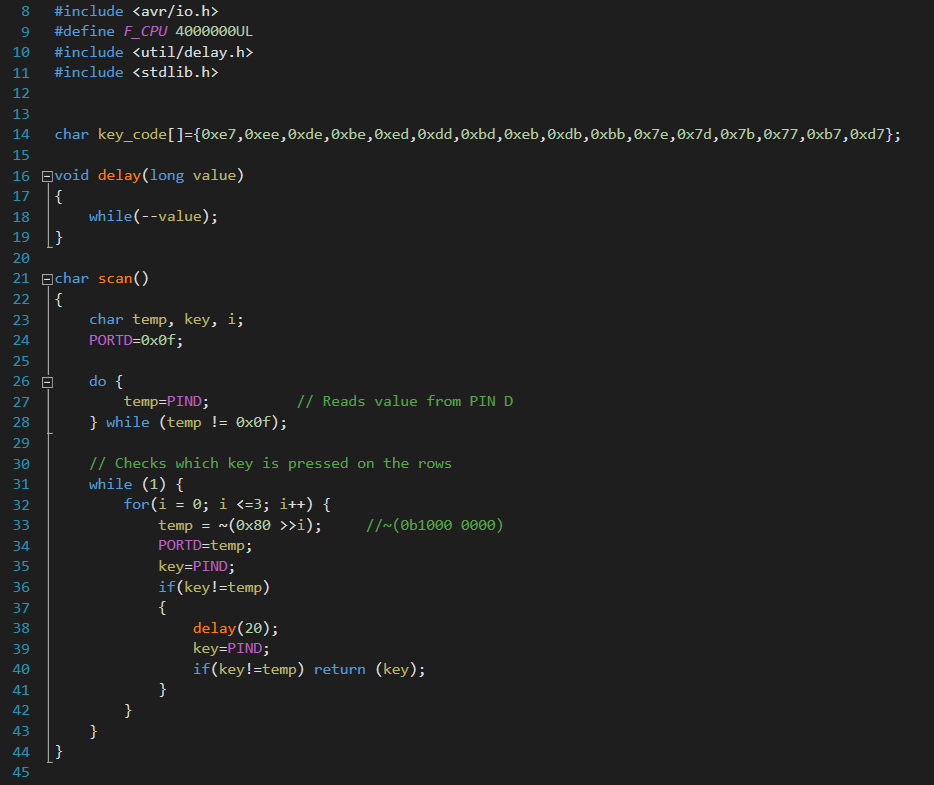
breq: branch equal

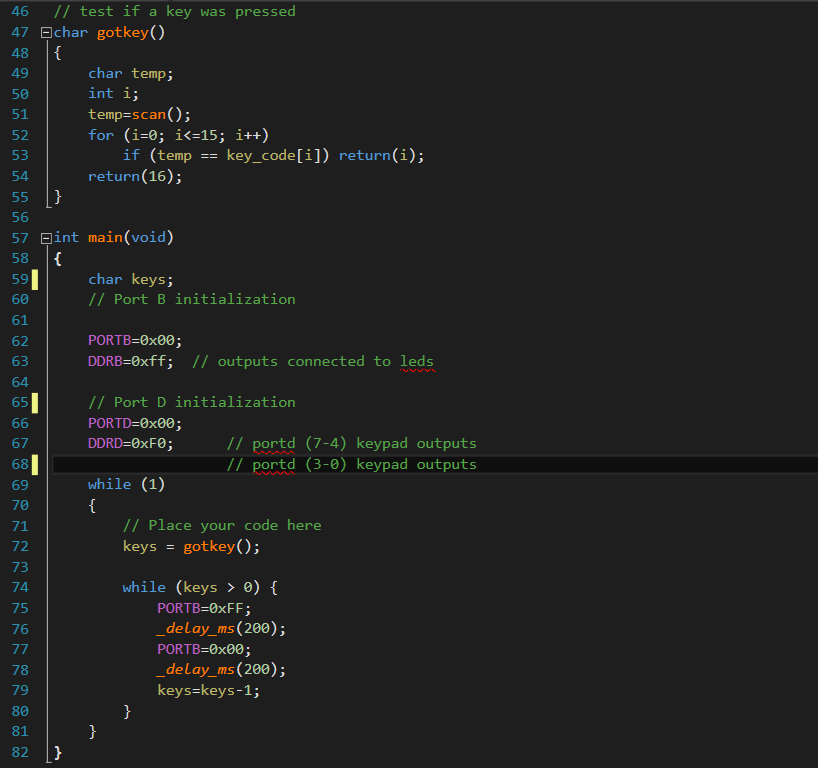
xor: eor

What’s the difference between mov and ldi?

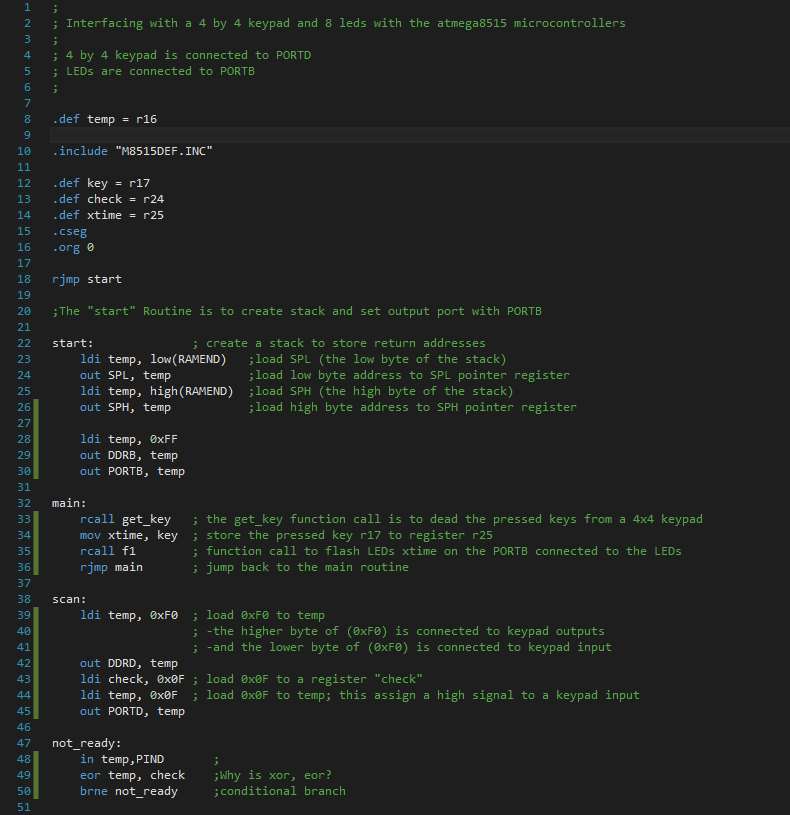
Comments are in code on page

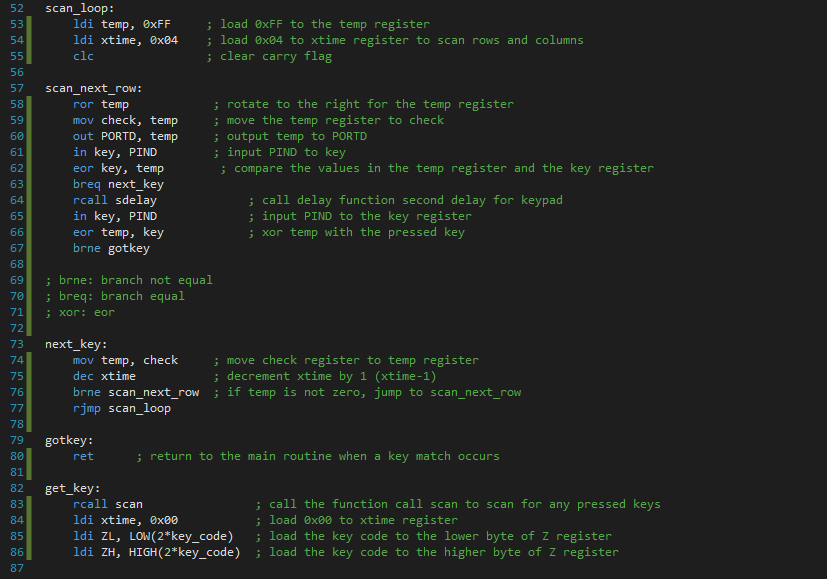
**C**

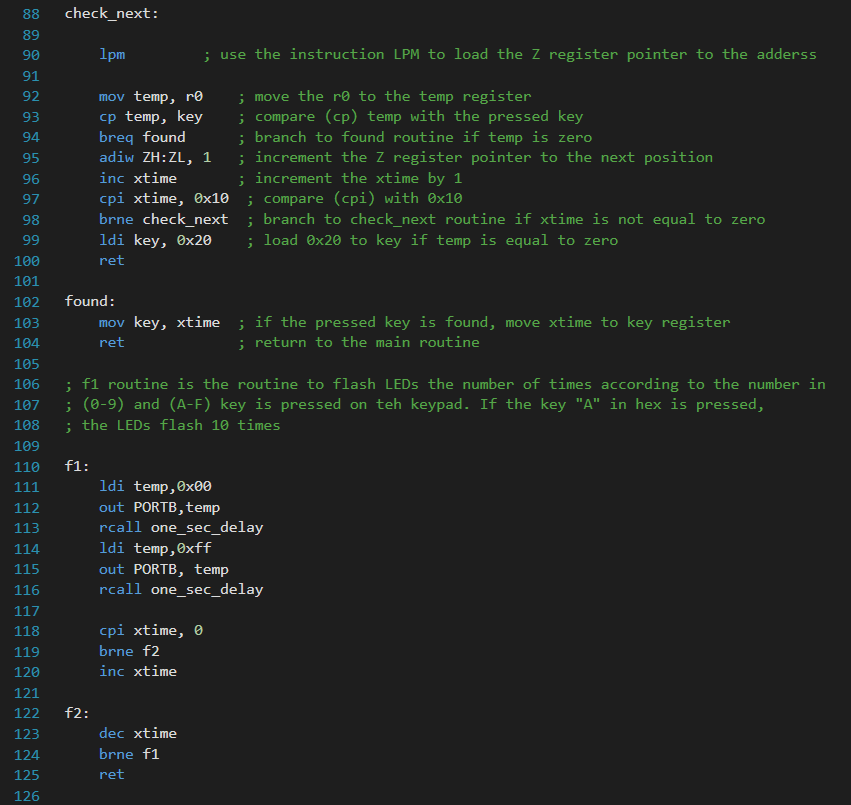


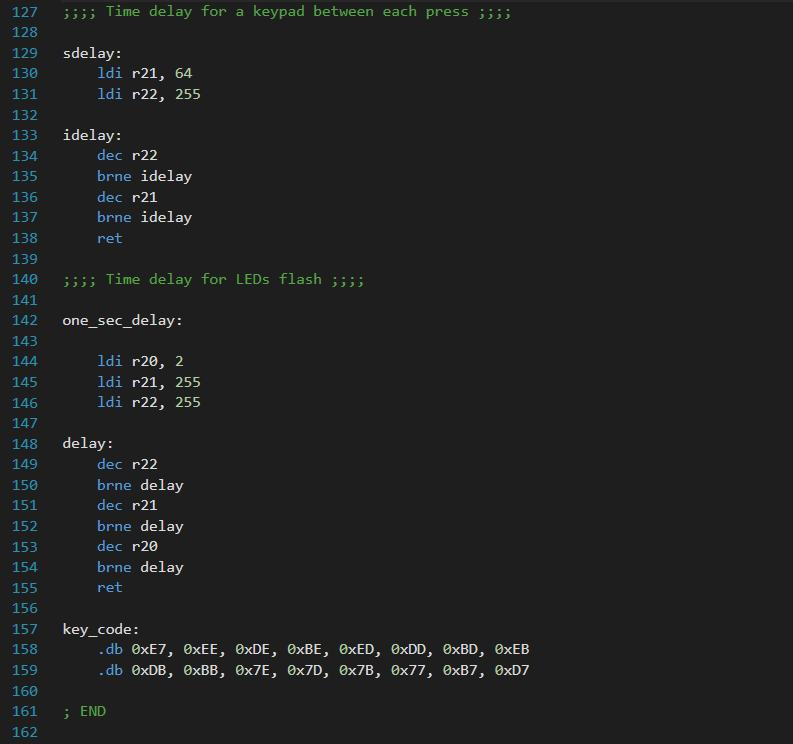


**Assembly**









Conclusion:

This lab had its difficulties. The C program wasn’t too bad. The hardest part was understanding how the values were scanned on the keypad and how the values were sent to the AVR board. In the main function, it was also important to figure out how to blink the LEDs while using the while loop that was counting down based on the number pressed.

The Assembly was most difficult. There was a steep learning curve in learning some of the new commands and figuring out what they do. It was also difficult to figure out some of what the Assembly code was doing and the goal of the code. I’m still planning on going over it some and analyzing what it does.