Lab 4
Jacob Hillebrand
CEE-345 Microprocessor System Design
Keypad controlling stepper motor

This lab continued our exploration of interfacing with a Digilent keypad with the Atmel 8515 microprocessor, and added a twist to it. The idea was that the keypad would be connected to the PORTD pins on the processor and when "F" was pressed, the stepper motor connected to PORTC would rotate in the forward direction, and when "B" was pressed, the stepper motor would rotate backwards. This assignment was written in C.
The code loaded the board's registers, set up PORTC and PORTD, performed a rapid column vs. row scan of the keypad, accurately recognized and debounced a keypad press, took in the value of the keypress and determined whether an "F" or "B" had been pressed. Then, either the forward or backwards function would be called, and the function would loop through sending the appropriate 8 different values to the stepper motor to enable its rotation for the predetermined amount of time. At this point, the process would start over, and the system would wait for another keypress. The code for this program is shown on the next pages.

```c
/*
 * Part_1.c
 *  This lab exercise is to interface a stepper motor and a keypad
 *  with the STK-600 board. The stepper motor requires a +5 V EXTERNAL
 *  power source because the STK-600 board doesn't have enough current
 *  for motor controls
 *
 *  PORTC IS CONNECTED TO THE MOTOR INPUTS
 *  PORTD IS CONNECTED TO THE KEYPAD INPUTS
 *
 * Created: 2/18 2:40:23
 * Author : Jacob Hillebrand
 */

#include <avr/io.h>
#define F_CPU 4000000UL
#include <util/delay.h>
#include <stdlib.h>

// write a sequence of digital pulses for the stepper motor to run forward

// reverse the sequence to run the motor backwards

// also need a method for scanning the keypad and returning the key

// Black Keypad - Digilent
char key_code[] = {0xE7,0xEE,0xDE,0xBE,0xED,0xDD,0xBD,0xEB,0xDB,0xBB,0x7E,0x7D,0x7B,0x77,0xB7,0xD7};


// Sequence to run motor forward
char forward[] = {0x08, 0x0C, 0x04, 0x06, 0x02, 0x03, 0x01, 0x09};

// Sequence to run motor backward
char backward[] = {0x09, 0x01, 0x03, 0x02, 0x06, 0x04, 0x0C, 0x08};

// Delay to control motor speed
void delay(long value){
    while(--value);
}
```

Figure 1: First Snippet

```
// Scan for the pressed key
char scan(){
    //Declare necessary variables
    char temp,key,i;

    //Begin the scan loop
    PORTD = 0x0f;

    //Make temp == PIND
    do {
        temp = PIND;
    } while (temp != 0x0f);

    while (1) {
        for (i = 0; i < 4; i++){
            temp =~(0x80 >> i);     //~(0b1000 0000) = 0b0111 1111
            PORTD = temp;
            key = PIND;
            if (key != temp){
                delay(20);      //delay for the switch debounces
                key = PIND;     //a pressed key is detected and return return its value
                if (key != temp) return (key);
            }
        }
    }

}

//Test if a key was pressed
char gotkey(){
    char temp;
    int i;
    temp = scan();
    for (i = 0; i <= 15; i++){
        if (temp == key_code[i]) return(i);
    }
    return (16);
}
```

Figure 2: Second Snippet

```
// Run motor forward
void runForward(){
    // Declare incrementors
    int i = 0;
    int j = 0;

    // motor runs for 300 iterations of the hex sequences
    while (j < 800){
        _delay_ms(10);          //delay adjusts the speed of the motor, user defined
        PORTC = forward[i];     // send PORTC the hex value from the sequence

        i++;                    //increment i and j
        j++;

        // Resets i back to 0 when it reaches the end of the sequence
        if (i == 8){
            i = 0;
        }
    }
}

// Run motor backwards
void runBackward(){
    // Declare incrementors
    int i = 0;
    int j = 0;

    // motor runs for 300 iterations of the hex sequences
    while (j < 800){
        _delay_ms(10);          //delay adjusts the speed of the motor, user defined
        PORTC = backward[i];        // send PORTC the hex value from the sequence

        i++;                    //increment i and j
        j++;

        // Resets i back to 0 when it reaches the end of the sequence
        if (i == 8){
            i = 0;
        }
    }
}
```

Figure 3: Third Snippet

```
122
123 int main(void)
124 {
125
126     char keys;
127
128     // PORTC initialization as motor input
129     PORTC = 0x00;
130     DDRC = 0xFF;
131
132     // PORTD initialization as the keypad
133     PORTD = 0x00;
134     DDRD = 0xF0;
135
136     while (1) {
137         keys = gotkey();
138         if (keys == 15 ){
139             runBackward();
140         }
141         if (keys == 11){
142             runForward();
143         }
144
145     }
146 }
147
148
```

Figure 4: Fourth Snippet