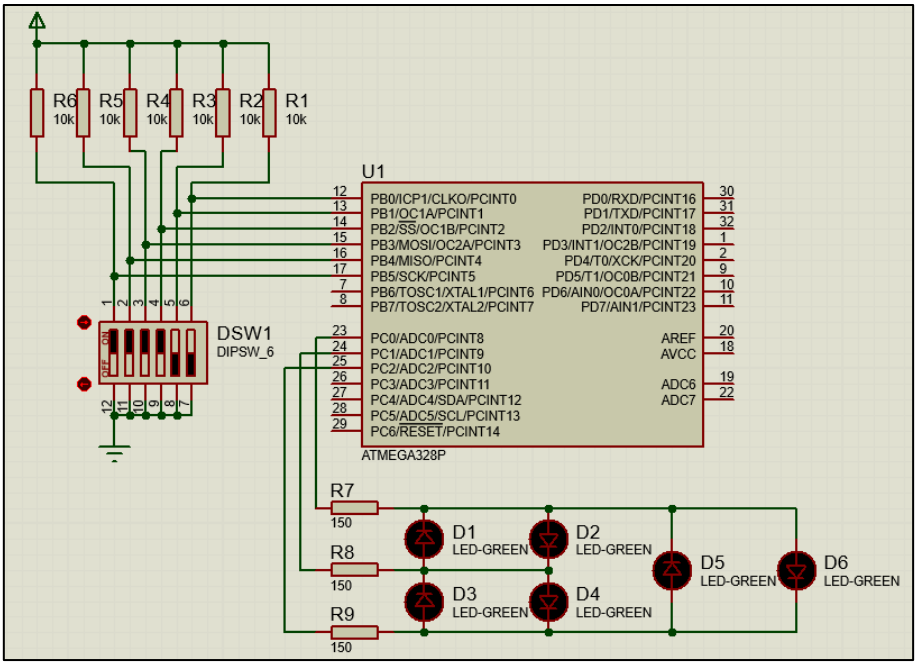
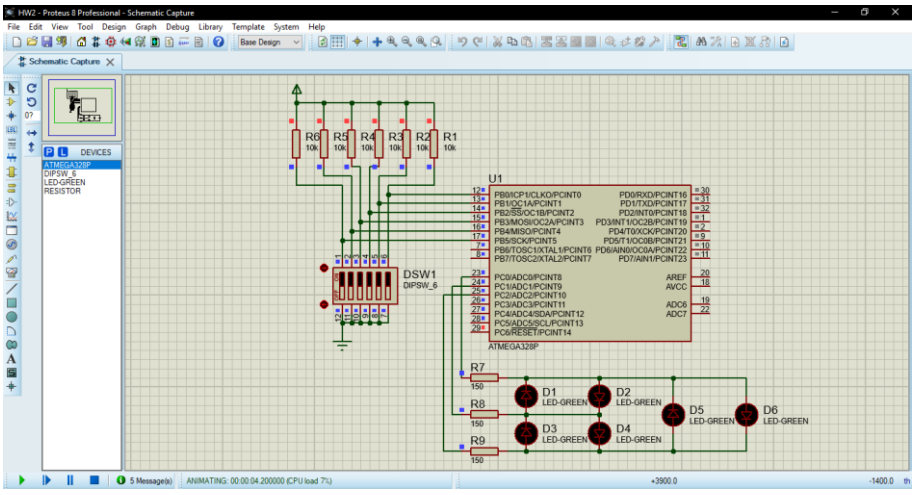
โค้ดภาษาซี

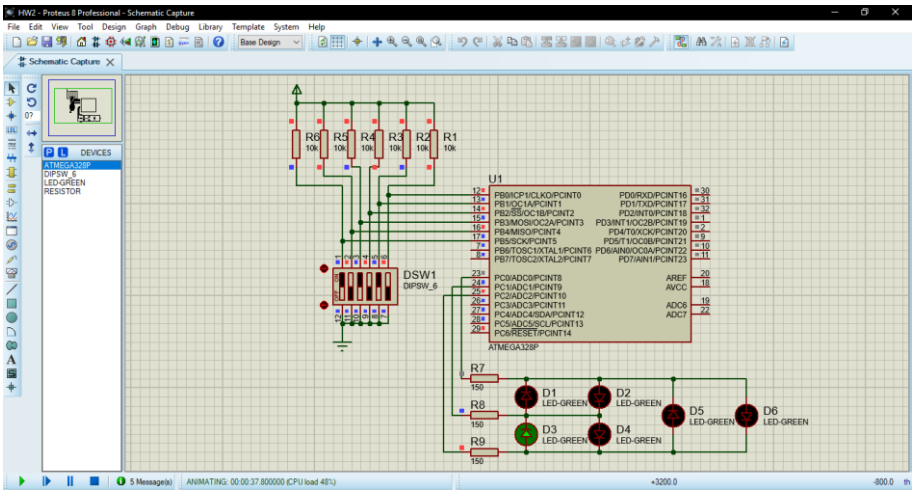
```
#include <avr/io.h>
#include <avr/interrupt.h>
#define F_CPU 16000000
#include <util/delay.h>

unsigned char LOOKUPTB[] = {
        //      DDRC    PORTC
        0b1111011, 0b0000010, //      LED 1
        0b1111011, 0b0000001, //      LED 2
        0b1111110, 0b0000100, //      LED 3
        0b1111110, 0b0000010, //      LED 4
        0b1111101, 0b0000100, //      LED 5
        0b1111101, 0b0000001  //      LED 6
};

unsigned char SWITCH_V, i, test_bit;

int main(void)
{
        DDRB   = 0xC0;
        PORTB  = 0x3F;
        SWITCH_V = PINB;
        //Set Pin Change Interrupt
        PCICR  = 0x01;
        PCMSK0 = 0x3F;
        sei();
        while (1)
        {
                if(SWITCH_V)
                {
                        for(i=0; i<6; i++)
                        {
                                test_bit = SWITCH_V & (1 << i);
                                if(test_bit)
                                {
                                        DDRC   = LOOKUPTB[2*i];
                                        PORTC  = LOOKUPTB[2*i + 1];
                                        _delay_ms(7);
                                }
                        }
                }
                else
                {
                        DDRC   = 0xFF;
                        PORTC  = 0x00;
                }
        }
}

ISR(PCINT0_vect)
{
        SWITCH_V = PINB;
}
```

ภาพวงจรบน Proteus



ภาพการทำงานของวงจรบน Proteus





นายวิรินทร โรจนศิรินทร์ 5810110302 Sec02

SUBJECT: NO: DATE: / /

การทำงานของโปรแกรม

1. ตั้งพอร์ต C บิตที่ 0 ถึง 5 เป็น อินพุตรับมาจาก Dip switch
2. อ่านค่าจาก switch เริ่มต้น เก็บไว้ในตัวแปร SWITCH_V
3. ตั้งให้ ตอบรับ อินเทอร์รัพท์จาก PCINT0 บิตที่ 0 ถึง 7
4. ตั้งให้ ตอบรับอินเทอร์รัพท์ เฉพาะ ขา บิตที่ 0 ถึง 5 ของ ขา PCINT0
5. กำหนดให้ตอบรับการ อินเทอร์รัพท์ของ AVR
6. เป็นส่วนการทำงานหลัก
   - 6.1 ตรวจสอบค่า SWITCH_V ที่อ่านมาว่า เท่ากับ 0 หรือไม่
     - 6.1.1 หากไม่เท่ากับ 0 ให้วนอ่าน ค่า SWITCH_V และตรวจสอบว่า บิตไหนเป็น 1 ให้แสดง LED (LED สว่าง) ที่ดวงนั้นโดย นำค่าจาก LOOKUPTB มา แสดง. โดยการส่งค่าให้ DDRC และ PORTC. และมีการ delay 7 ms ก่อนแสดงดวงถัดไป.
     - 6.1.2 หากเท่ากับ 0 ให้ LED ทุกดวงดับ
   - 6.2 วนกลับไปทำงานดังเดิม ในข้อ 6.1
7. ในส่วนของฟังก์ชันรองรับอินเทอร์รัพท์. มีการทำงานเดียว เมื่อ เกิด อินเทอร์รัพท์. อ่านค่าจาก Dip switch เก็บค่าใน SWITCH_V เพื่อนำไปใช้ แสดงผลในฟังก์ชัน หลักต่อไป.

สรุป

การทำงานอ่าน ค่าจาก Dip switch โดยอ่านเมื่อเกิด อินเทอร์รัพท์. แล้วมาแสดงผลในฟังก์ชัน การทำงานหลัก. การแสดงผล LED แบบ Charlieplexing ต้องมีการ สลับให้ LED ติด สลับกัน. ด้วยความถี่มากกว่า 60 Hz เพื่อให้มองเห็นว่า LED ติด พร้อมกัน.

วิเคราะห์

การทำให้ LED แต่ละดวงติด ต้อง ให้ค่า HIGH , LOW และ Hi-Z กับขา PC0 , PC1 และ PC2 สลับไปจะทำให้ ได้ผลลัพธ์ ที่ LED ติดแตกต่าง กัน / ติด คนละดวงในแต่ละ แบบที่ให้ขา PC0 ถึง PC2 สำหรับการ delay เพื่อให้ได้ ความถี่มากกว่า 60 Hz ต้องใช้ ค่า delay น้อยกว่า 1/60 s = 0.017 s = 17 ms. จากการทดลองจึงเลือกใช้ 7 ms