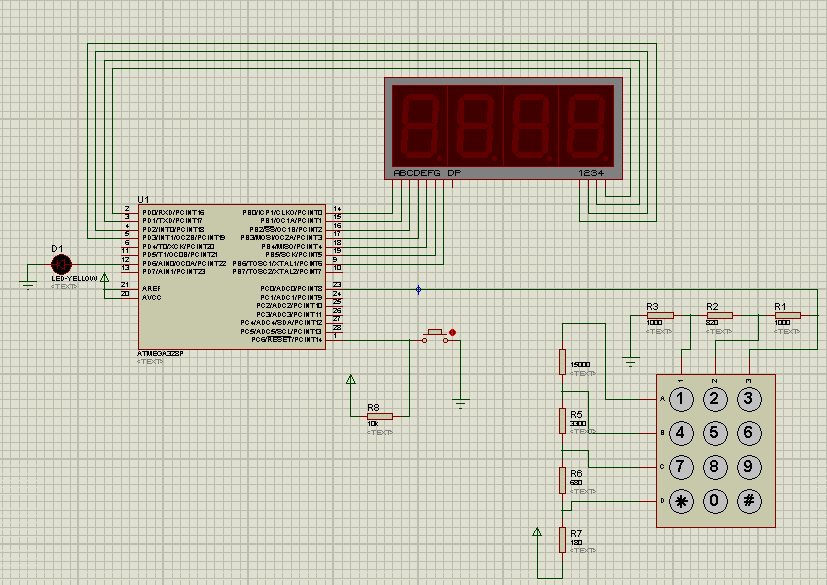
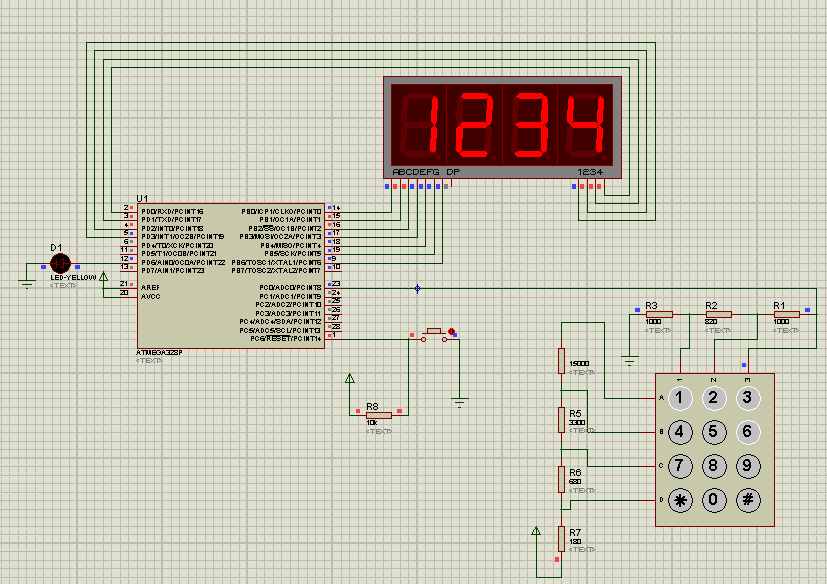
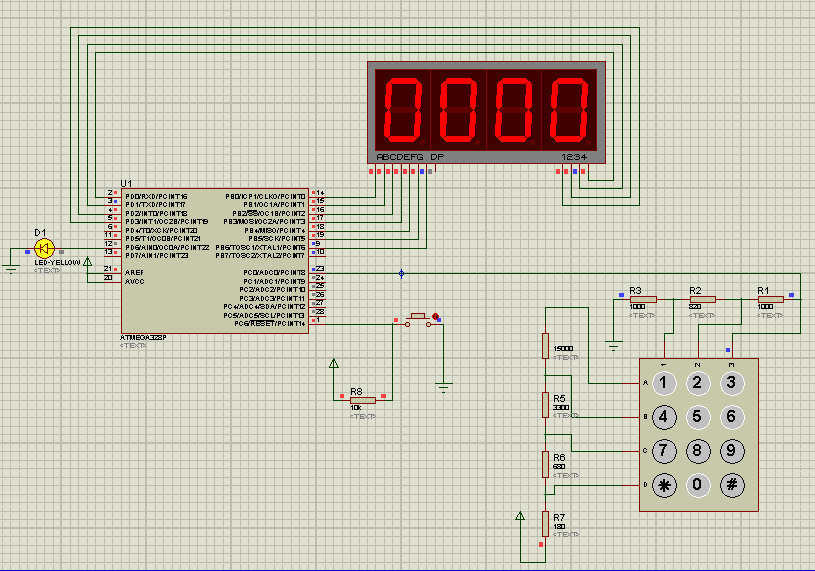
**วงจรนับถอยหลังเปิด-ปิด LED**

**รูปวงจรจากโปรแกรม Proteus**







**หลักการทำงาน**

วงจรนี้เป็นวงจรนับถอยหลังเปิด-ปิด LED มีขั้นตอนการทำงานดังนี้

1.ตั้งเวลาผ่าน keypad สามารถตั้งเวลาได้มากที่สุด 9999 โดยจะมีการแสดงเวลาที่ตั้งผ่านทาง 7-segment

2.การเริ่มนับถอยหลังทำได้โดยกดปุ่ม \* บน keypad

3.เมื่อวงจรนับถอยหลังจนถึง 0000 หลอด LED ก็จะดับ

4.สามารถทำการ reset ค่าที่ตั้งไว้โดยการกดปุ่ม # บน keypad เพื่อตั้งเวลาใหม่

5.ในขณะที่วงจรกำลังนับถอยหลัง หากผู้ใช้ต้องการตั้งเวลาใหม่ก็สามารถกดปุ่ม Button เพื่อ reset หยุดการนับถอยหลัง และตั้งค่าเวลาใหม่ได้

**เทคนิคที่ใช้ในวงจร**

1. Timer/Counter1 (วงจรจับเวลา/นับ ขนาด 16 บิต ) และโปรแกรมบริการอินเตอร์รัพต์ของ Timer/Counter1 ใช้ Timer1 ในการให้จังหวะสัญญาณนาฬิการ่วมกับการแปลง Analog to Digital ของ Keypad
2. การติดต่อ Keypad โดยใช้ Analog to Digital converter ใช้เพื่อช่วยลดอินพุตจาก 7 ขา เหลือเพียง 1 ขา พร้อมทั้งใช้ ADC จัดการ bounces
3. การขับ LED แบบ 7 เซกเมนต์ด้วยวิธี Multiplexing ใช้ในการแสดงผ่าน 7 เซกเมนต์แบบ 4 หลัก

**Code**

//Assignment

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

#define TURN\_ALL\_LED\_OFF 0x00

#define INTERVAL\_time 65528

unsigned char TB7SEG[] = { 0b00111111,0b00000110,

0b01011011,0b01001111,

0b01100110,0b01101101,

0b01111101,0b00000111,

0b01111111,0b01101111,

0b01110111,0b01111100,

0b00111001,0b01011110,

0b01111001,0b01110001,

0b00000000};

unsigned char DIVISION ,DIVISOR,QUOTIENT ,DECODED;

unsigned short ACCUMULATE, LIGHT = 0x0F;

unsigned char D[] = {16,16,16,16} , result = 0 ;

unsigned char i = 0 , j , k , l , m ;

ISR(TIMER1\_OVF\_vect)

{

TCNT1 = INTERVAL\_time;

}

ISR(ADC\_vect)

{

//unsigned char DECODED;

if(ADCH > 9)

{

ACCUMULATE += ADCH;

DIVISOR++;

}

else

{

ACCUMULATE = 0;

DIVISOR = 0;

}

if(DIVISOR == 10)

{

QUOTIENT = ACCUMULATE/DIVISOR;

if ((QUOTIENT>9)&&(QUOTIENT<16)) DECODED = 1;

else if((QUOTIENT>18)&&(QUOTIENT<26)) DECODED = 2;

else if((QUOTIENT>29)&&(QUOTIENT<37)) DECODED = 3;

else if((QUOTIENT>46)&&(QUOTIENT<53)) DECODED = 4;

else if((QUOTIENT>74)&&(QUOTIENT<81)) DECODED = 5;

else if((QUOTIENT>99)&&(QUOTIENT<107)) DECODED = 6;

else if((QUOTIENT>133)&&(QUOTIENT<141)) DECODED = 7;

else if((QUOTIENT>169)&&(QUOTIENT<177)) DECODED = 8;

else if((QUOTIENT>192)&&(QUOTIENT<199)) DECODED = 9;

else if((QUOTIENT>212)&&(QUOTIENT<220)) DECODED = 10;

else if((QUOTIENT>228)&&(QUOTIENT<236)) DECODED = 0;

else if((QUOTIENT>236)&&(QUOTIENT<244)) DECODED = 11;

else DECODED = 16; //assume no keypad

if(DECODED != 16 && DECODED != 10 && DECODED != 11 )

{ // สำหรับการเก็บตัวแปลที่รับมาทั้ง 4 หลัก

D[i] = DECODED;

i++ ;

}

else if (DECODED == 11 ) // reset value

{ // สำหรับการรีเซตค่าบน 7 segment

D[0] = 16 ;

D[1] = 16 ;

D[2] = 16 ;

D[3] = 16 ;

i = 0 ;

}

else if (DECODED == 10 && i > 3 ) // start count

{ // สำหรับการเริ่มต้นนับถอยหลังเพื่อให้โปรแกรมทำงาน

j = D[0] ;

k = D[1] ;

l = D[2] ;

m = D[3] ;

for (;j >= 0 ; j--)

{

if (j == 255)

{

j = 9;

D[0] = 0;

break;

}

for (;k >= 0 ; k--)

{

if (k == 255)

{

k = 9;

D[1] = 0;

break;

}

for (;l >= 0 ; l--)

{

if (l == 255)

{

l = 9;

D[2] = 0;

break;

}

for (;m >=0 ; m--)

{

if (m == 255)

{

m = 9;

D[3] = 0;

break;

}

PORTD = 0b0111 ;

PORTB = TB7SEG[j] ;

\_delay\_ms(15);

PORTD = 0b1011 ;

PORTB = TB7SEG[k] ;

\_delay\_ms(15);

PORTD = 0b1101 ;

PORTB = TB7SEG[l] ;

\_delay\_ms(15);

PORTD = 0b1110 ;

PORTB = TB7SEG[m] ;

\_delay\_ms(15);

}

}

}

}

}

}

}

int main(void)

{

DDRB = 0xFF; //port B is connected to 7seg ment

PORTB = TURN\_ALL\_LED\_OFF; //turn off all segment

DDRD = 0x4F; //common cathod

PORTD = 0x40 ;

//ADMUX[7:6] = 00 -> using AREF pin

//ADMUX[5] = 1 -> ADLR = 1

//ADMUX[3:0] = -> ADC0 pin

ADMUX = 0b00100000;

//ADCSRA[7]=1 -> ADC enable

//ADCSRA[6]=0 -> ADC start coversion chang auto

//ADCSRA[5]=1 -> enable auto trigger

//ADCSRA[4]=0 -> ADC interrupt Flag

//ADCSRA[3]=1 -> ADC interrupt enable

//ADCSRA[2:0]=101 -> Division factor/32

ADCSRA = 0b10101101;

//-----------------ADCSRB--------------

//ADCSRB[2:0] = 110 -> Trigger source = Timer1 OVF

ADCSRB = 0x06;

//------Timger1 setup-----------------------------------

TCNT1 = INTERVAL\_time;

//Normal mode, N = 1024,

TCCR1A = 0x00;

TCCR1B = 0x05;

TIMSK1 = 0x01;

sei();

DIVISOR = 0;

//----------loop------------------

while(1)

{

PORTD = 0b0111 | LIGHT;

PORTB = TB7SEG[D[0]] ;

\_delay\_ms(15);

PORTD = 0b1011 | LIGHT;

PORTB = TB7SEG[D[1]] ;

\_delay\_ms(15);

PORTD = 0b1101 | LIGHT;

PORTB = TB7SEG[D[2]] ;

\_delay\_ms(15);

PORTD = 0b1110 | LIGHT;

PORTB = TB7SEG[D[3]] ;

\_delay\_ms(15);

if (D[0] == 0 && D[1] == 0 && D[2] == 0 && D[3] == 0)

{

LIGHT = 0xF0;

}

else

{

LIGHT = 0x00;

}

}

}