

ASSIGNMENT

วงจรแปลงเวลาตามโซน (UTC)

รายวิชา 242-309 MICROCONTROLLER & INTERFACING

ภาคการศึกษา 2/2560

โดย

นายจตุภัทร์ ปานน้อย รหัสนักศึกษา 5735512002

นายณราธิป ทองดาสว่าง รหัสนักศึกษา 5735512153

กลุ่มที่ 20 Section 02

เสนอ

อาจารย์พัชรี เทพนมิตร

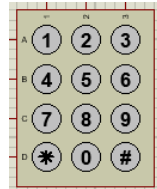
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตภูเก็ต

วงจรแปลงเวลาตามโซน (UTC)

Input

- Keypad 3x4



Output

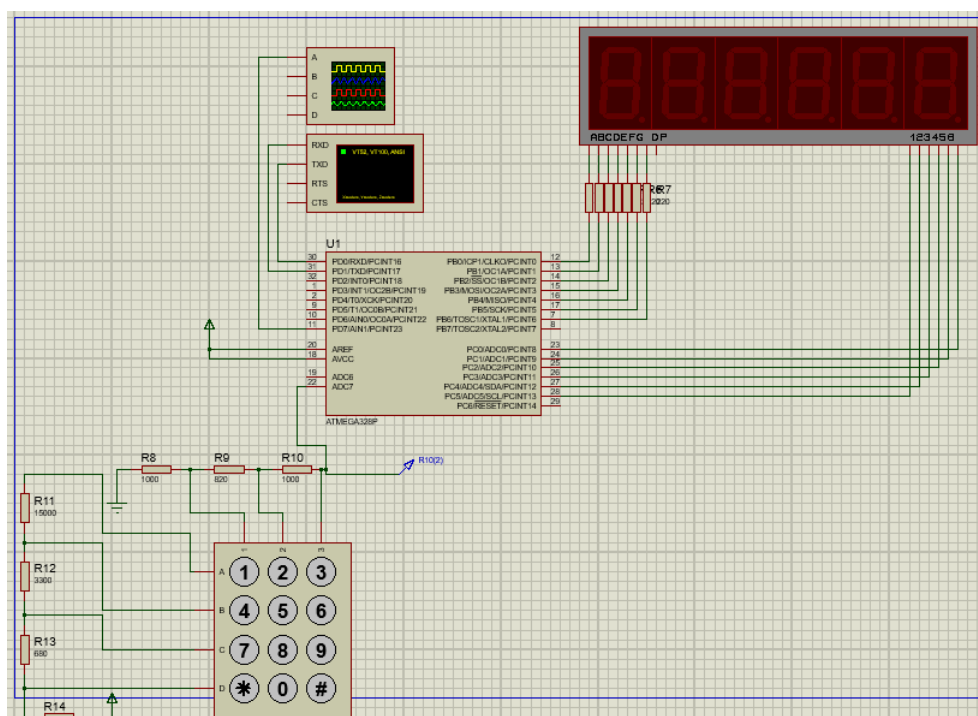
- Virtual Terminal (Usart)



- 7 Segment 1 digit X 1



รูปวงจร



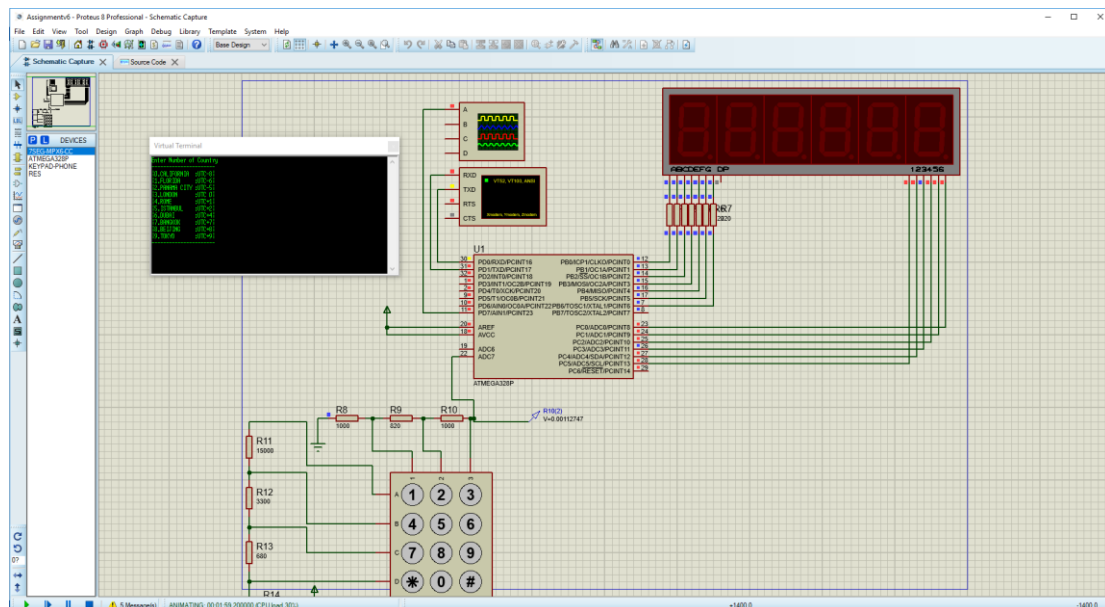
โหมดในการทำงานและวิธีการใช้งาน

- 1) ทำการกดปุ่มตัวเลขเพื่อทำการตั้งเวลานาฬิกาเวลาปัจจุบัน
- 2) หลังจากตั้งเวลานาฬิกาปัจจุบันแล้วให้ทำการกด * เพื่อทำการเลือกโซน โดยสามารถเลือกโซนได้ 0-9 โดยเป็นเลือกเวลาท้องถิ่นตามเมืองดังนี้หมายเลขโซนเวลาโดยมีหมายเลขระบุโซนเวลาดังต่อไปนี้

หมายเลข 0	เมืองแคลิฟอร์เนีย	UTC -8	CALIFORNIA
หมายเลข 1	เมืองฟลอริดา	UTC -6	FLORIDA
หมายเลข 2	เมืองปานามาซิตี	UTC -5	PANAMA CITY
หมายเลข 3	เมืองลอนดอน	UTC 0	LONDON
หมายเลข 4	เมืองโตเกียว	UTC +9	TOKYO
หมายเลข 5	เมืองปักกิ่ง	UTC +8	BEIJING
หมายเลข 6	เมืองกรุงเทพฯ	UTC +7	BANGKOK
หมายเลข 7	เมืองดูไบ	UTC +4	DUBAI
หมายเลข 8	เมืองอิสตันบูล	UTC +2	ISTANBUL
หมายเลข 9	เมืองโรม	UTC +1	ROME

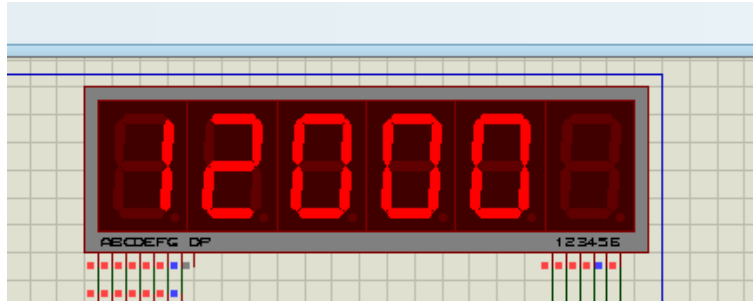
- 3) กดปุ่ม # คือ การรีเซ็ตการทำงานของนาฬิกา

Screenshot

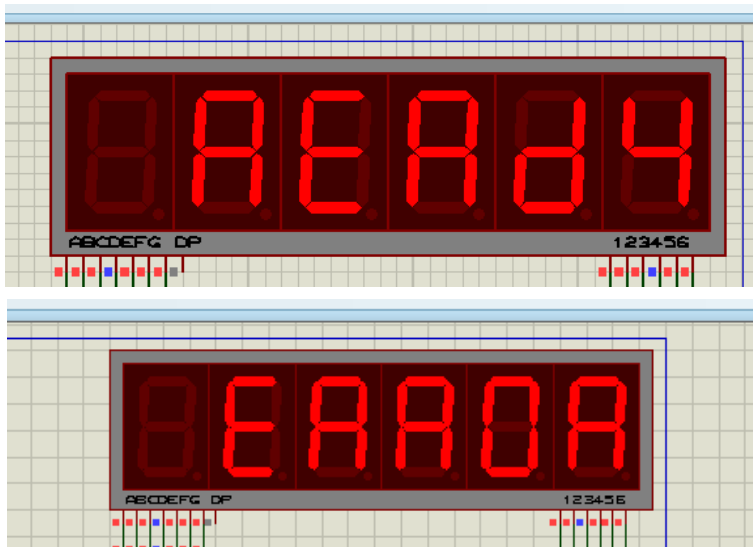


ตัวอย่างการทำงาน

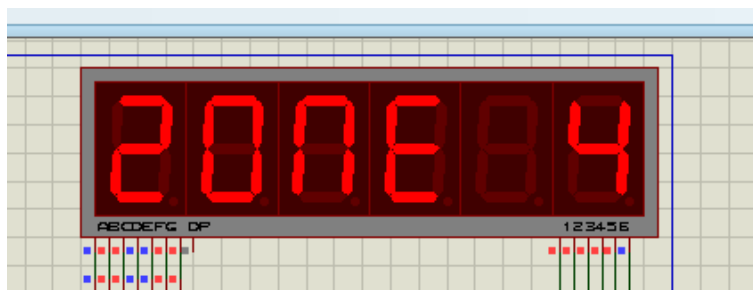
-ทำการตั้งเวลา โดยในตัวอย่างเป็นเวลา 12.00



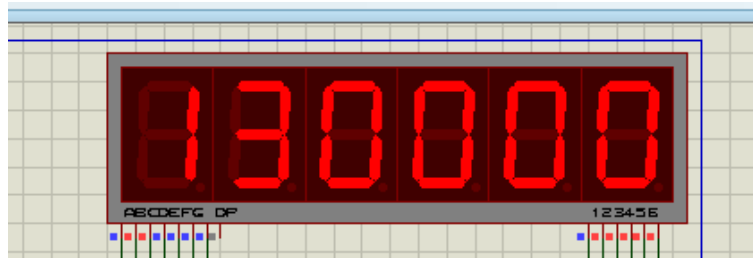
-จากนั้นโปรแกรมจะทำการเช็คว่าการตั้งนั้นเกิน 24.00 หรือไม่ ถ้าเกินจะแสดง ERROR แต่ถ้าอยู่ในเงื่อนไขจะแสดง READY ดังรูปดังกล่าว



-จากนั้นทำการเลือกโซนเวลา โดยในที่นี้ทำการเลือกโซนเป็น โซนหมายเลข 4 ซึ่งเป็นเมือง ROME โดยมี UTC +1



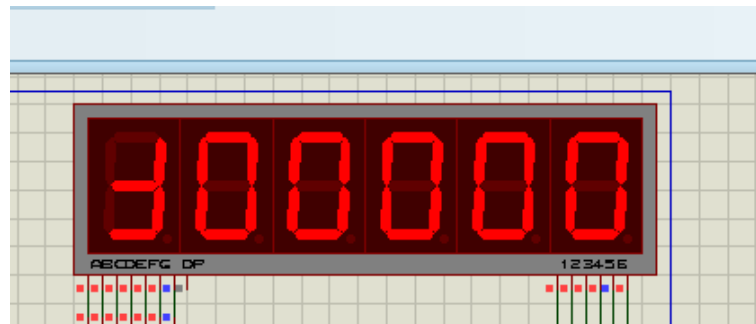
-เวลาที่ได้จะเป็นดังรูปดังกล่าวโดยจะแสดง 13.00 โดยมาจาก $12.00 + 1.00$



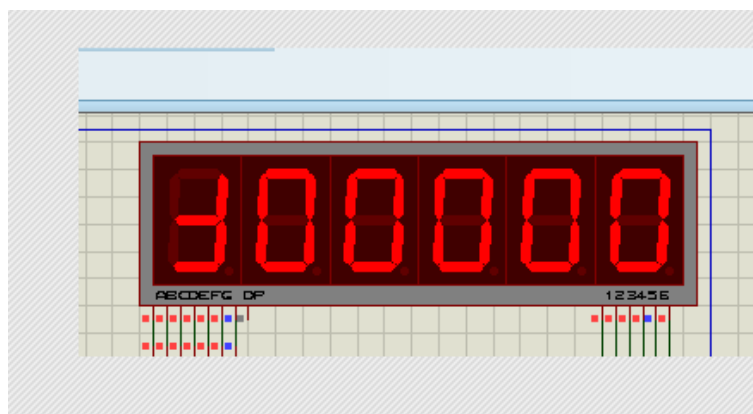
หมายเหตุ

- *ในกรณีที่เมื่อทำการตั้งเวลาที่น้อยกว่าเวลาของโซนจะไม่สามารถแสดงเวลาได้เนื่องจากเป็นเวลาของเมื่อวาน สามารถแก้ไขได้โดยการนำเวลาที่ได้ทำการตั้งมาทำการลบกับ UTC แล้ว Modulo กับ 24
- **ในกรณีที่เมื่อทำการตั้งเวลาที่ใกล้กับ 24.00 โดยเมื่อทำการแปลงโซนเวลาแล้วผลลัพธ์ที่ได้จะเป็นเวลาของวันพรุ่งนี้ และสามารถแก้ไขได้โดยการเพิ่มอุปกรณ์เพื่อเป็นสัญลักษณ์ของวันพรุ่งนี้
- กรณีปัญหาดังกล่าวเช่นดังรูปต่อไปนี้

-ทำการตั้งเวลาที่ 00.00.00 และเลือกโซน UTC-9 จะแสดงดังรูป



-ทำการตั้งเวลาที่ 23.00.00 และเลือกโซน UTC+9 จะแสดงดังรูป



เทคนิคที่ใช้ในวงจร

- 1) ใช้ความรู้เรื่อง Timer/Counter1 (วงจรจับเวลา/นับ ขนาด 16 บิต) และโปรแกรมบริการอินเทอร์รัพต์ของ Timer/Counter1 ใช้ Timer1 ในการให้จังหวะสัญญาณนาฬิกา ร่วมกับการแปลง Analog to Digital ของ Keypad
- 2) ใช้ความรู้เรื่องการสื่อสารข้อมูลแบบอนุกรม (serial Communication) เพื่อแสดงค่าต่าง ๆ
- 3) ใช้ความรู้เรื่อง Delay ในการแสดงผล 7segment โดยจะใช้ Delay ที่เหมาะสมเพื่อให้หลอดตามองตัวเลขที่แสดงนั้นแสดงต่อเนื่องกันในแต่ละ digit

Source Code

```
#include<avr/io.h>
#include<avr/interrupt.h>
#include <util/delay.h>
#define TURN_ALL_LED_OFF 0x00
#define VALUE_OF_TCNT1 65528

volatile unsigned char TIME1,ZONE;
unsigned char DIVISION, DIVISOR, QUOTIENT, DECODED;
unsigned short ACCUMULATE;
volatile unsigned char i2 = 0,STATE=0,STATE2=0,MODE =0;
volatile unsigned char D1,D2,D3,a,b,c,d,e,f;
volatile unsigned char D[] = {16,16,16,16,16,16},A[];
unsigned char TB7SEG[] = { 0b00111111,0b00000110,0b01011011,0b01001111, //0-3
                          0b01100110,0b01101101,0b01111101,0b00000111, //4-7
                          0b01111111,0b01101111,0b01110111,0b01111100, //8-B
                          0b00111001,0b01011110,0b01111001,0b01110001, //C-F
                          0b00000000,0b00110111,0b01110000 };
unsigned char USART_Receive( void )
{
    while (!(UCSR0A & (1 <<RXC0)) );
    return UDR0;
}

void USART_Transmit(unsigned char data)
{
    while(! (UCSR0A & (1 << UDRE0)));
    UDR0 = data;
}

unsigned char TEXT[40] = {"Enter Number of Country"};
unsigned char m1[30] = {"|0.CALIFORNIA :UTC-8|"};
unsigned char m2[30] = {"|1.FLORIDA :UTC-6|"};
```

```

unsigned char m3[30] = {"|2.PANAMA CITY :UTC-5|"};
unsigned char m4[30] = {"|3.LONDON      :UTC 0|"};
unsigned char m5[30] = {"|4.ROME       :UTC+1|"};
unsigned char m6[30] = {"|5.ISTANBUL   :UTC+2|"};
unsigned char m7[30] = {"|6.DUBAI     :UTC+4|"};
unsigned char m8[30] = {"|7.BANGKOK    :UTC+7|"};
unsigned char m9[30] = {"|8.BEIJING   :UTC+8|"};
unsigned char m10[30] = {"|9.TOKYO     :UTC+9|"};
unsigned char m11[30] = {"|-----|"};
unsigned char i;
int main(void)
{
    TIME1=0;
    DDRB = 0xFF;
    PORTB = TURN_ALL_LED_OFF;
    DDRC = 0xFF;
    DDRD = 0b11111111;
    PORTD = 0b11111110;

    ADMUX = 0b00100111;
    //ADMUX[7:6] = 00 -> using AREF pin
    //ADMUX[5] = 1 -> ADLR = 1
    //ADMUX[3:0] = -> ADC0 pin
    ADCSRA = 0b10101101;
    //ADCSRA[7]=1 -> ADC enable
    //ADCSRA[6]=0 -> ADC start conversion change 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
    ADCSRB = 0b00000110;
    //ADCSRB[2:0] = 110 -> Trigger source = Timer1 OVF
    TIMSK1 = 0b00000001;

```



```

//Overflow Interrupt Enable
TCCR1A = 0b00000000;
//set normal mode (bit0,1 is normal mode,bit4,5,6,7 is normal port operation)
TCCR1B = 0b00000101;
//clk 256 (bit0,1,2 is 100,bit3,4 is normal mode)
cli();
//Disable global interrupt
TCNT1 = VALUE_OF_TCNT1;
sei();
//Enable global interrupt
ACCUMULATE = 0;

UCSR0A = 0x02;
UCSR0B = 0x98;
UCSR0C = 0x06;

UBRR0H = 0;
UBRR0L = 207;

TEXT [26] = 13;
TEXT [27] = 10;

for (i = 0; i<30; i++)
{
    USART_Transmit(TEXT[i]);
}
m11[26] = 13;
m11[27] = 10;
for (i = 0; i<30; i++)
{
    USART_Transmit(m11[i]);
}

```

```

m1[26] = 13;
m1[27] = 10;
for (i = 0; i<30; i++)
{
    USART_Transmit(m1[i]);
}
m2[26] = 13;
m2[27] = 10;
for (i = 0; i<30; i++)
{
    USART_Transmit(m2[i]);
}
m3[26] = 13;
m3[27] = 10;
for (i = 0; i<30; i++)
{
    USART_Transmit(m3[i]);
}
m4[26] = 13;
m4[27] = 10;
for (i = 0; i<30; i++)
{
    USART_Transmit(m4[i]);
}
m5[26] = 13;
m5[27] = 10;
for (i = 0; i<30; i++)
{
    USART_Transmit(m5[i]);
}
m6[26] = 13;

```

```

m6[27] = 10;
for (i = 0; i<30; i++)
{
    USART_Transmit(m6[i]);
}
m7[26] = 13;
m7[27] = 10;
for (i = 0; i<30; i++)
{
    USART_Transmit(m7[i]);
}
m8[26] = 13;
m8[27] = 10;
for (i = 0; i<30; i++)
{
    USART_Transmit(m8[i]);
}
m9[26] = 13;
m9[27] = 10;
for (i = 0; i<30; i++)
{
    USART_Transmit(m9[i]);
}
m10[26] = 13;
m10[27] = 10;
for (i = 0; i<30; i++)
{
    USART_Transmit(m10[i]);
}
m11[26] = 13;
m11[27] = 10;
for (i = 0; i<30; i++)

```

```

{
    USART_Transmit(m11[i]);
}

while(1)
{
    PORTC = 0b11111110; // ตำแหน่งที่ 1
    PORTB = TB7SEG[D[5]];
    _delay_ms(5);

    PORTC = 0b11111101; // ตำแหน่งที่ 2
    PORTB = TB7SEG[D[4]];
    _delay_ms(5);

    PORTC = 0b11111011; // ตำแหน่งที่ 3
    PORTB = TB7SEG[D[3]];
    _delay_ms(5);

    PORTC = 0b11110111; // ตำแหน่งที่ 4
    PORTB = TB7SEG[D[2]];
    _delay_ms(5);

    PORTC = 0b11101111; // ตำแหน่งที่ 5
    PORTB = TB7SEG[D[1]];
    _delay_ms(5);

    PORTC = 0b11011111; // ตำแหน่งที่ 6
    PORTB = TB7SEG[D[0]];
    _delay_ms(5);
}
}

```

```

ISR(TIMER1_OVF_vect)
{
    TCNT1 = VALUE_OF_TCNT1;

//    PORTD = PORTD ^ 0x80;

}

ISR(ADC_vect)
{
    //PORTD = PORTD^0x80;
    //unsigned char DECODED;
    if(ADCH > 9)
    {
        ACCUMULATE += ADCH;
        DIVISOR++;
        //USART_Transmit(DIVISOR+'0');
    }
    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

```

```

PORTD ^= 0x80;

```

```

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

```

```

{
    D[i2] = DECODED;
    /*if(STATE2==0)
    A[i2] = DECODED;*/
    i2++ ;
}

```

```

else if(DECODED == 11 )

```

```

{
    D[0]=16;
    D[1]=16;
    D[2]=16;
    D[3]=16;
    D[4]=16;
    D[5]=16;
    i2=0;
    STATE=0;
}

```

```

    }
    else if(DECODED == 10 )
    {
        D[0]=16;
        D[1]=16;
        D[2]=16;
        D[3]=16;
        D[4]=16;
        D[5]=16;
        i2=0;
        STATE++;
    }

    if((DECODED <= 9 && i2==6 && STATE == 0))
    {
        D1=D[0]*10+D[1];//24
        D2=D[2]*10+D[3];//60
        D3=D[4]*10+D[5];//60

        if(D1<24 && D2<60 && D3<60)
        {
            D[0]=16;
            D[1]=10;
            D[2]=14;
            D[3]=10;
            D[4]=13;
            D[5]=4;
        }
    }

```

```

        // STATE2++;
    }
    else
    {
        D[0]=16;
        D[1]=14;
        D[2]=10;
        D[3]=10;
        D[4]=0;
        D[5]=10;
    }
}
if(STATE == 1)
{
    ZONE = D[5];
}
if(DECODED == 10 && STATE ==1)
{

    D[0]=2;
    D[1]=0;
    D[2]=17;
    D[3]=14;
    D[4]=16;
    D[5]=16;
    i2=5;

}
else if(DECODED ==10 && STATE == 2)
{

```



```

if(ZONE == 0)
{
    D1 = D1-8;
    D[0] = D1/10;
    D[1] = D1%10;
    D[2] = D2/10;
    D[3] = D2%10;
    D[4] = D3/10;
    D[5] = D3%10;
}
else if(ZONE == 1)
{
    D1 = D1-6;
    D[0] = D1/10;
    D[1] = D1%10;
    D[2] = D2/10;
    D[3] = D2%10;
    D[4] = D3/10;
    D[5] = D3%10;
}
else if(ZONE == 2)
{
    D1 = D1-5;
    D[0] = D1/10;
    D[1] = D1%10;
    D[2] = D2/10;
    D[3] = D2%10;
    D[4] = D3/10;
    D[5] = D3%10;
}
else if(ZONE == 3)
{

```

```

        D1 = D1;
        D[0] = D1/10;
        D[1] = D1%10;
        D[2] = D2/10;
        D[3] = D2%10;
        D[4] = D3/10;
        D[5] = D3%10;
    }
    else if(ZONE == 4)
    {
        D1 = D1+1;
        D[0] = D1/10;
        D[1] = D1%10;
        D[2] = D2/10;
        D[3] = D2%10;
        D[4] = D3/10;
        D[5] = D3%10;
    }
    else if(ZONE == 5)
    {
        D1 = D1+2;
        D[0] = D1/10;
        D[1] = D1%10;
        D[2] = D2/10;
        D[3] = D2%10;
        D[4] = D3/10;
        D[5] = D3%10;
    }
    else if(ZONE == 6)
    {
        D1 = D1+4;
        D[0] = D1/10;

```

```

        D[1] = D1%10;
        D[2] = D2/10;
        D[3] = D2%10;
        D[4] = D3/10;
        D[5] = D3%10;
    }
    else if(ZONE == 7)
    {
        D1 = D1+7;
        D[0] = D1/10;
        D[1] = D1%10;
        D[2] = D2/10;
        D[3] = D2%10;
        D[4] = D3/10;
        D[5] = D3%10;
    }
    else if(ZONE == 8)
    {
        D1 = D1+8;
        D[0] = D1/10;
        D[1] = D1%10;
        D[2] = D2/10;
        D[3] = D2%10;
        D[4] = D3/10;
        D[5] = D3%10;
    }
    else if(ZONE == 9)
    {
        D1 = D1+9;
        D[0] = D1/10;
        D[1] = D1%10;
        D[2] = D2/10;
    }

```

```

        D[3] = D2%10;
        D[4] = D3/10;
        D[5] = D3%10;
    }

    else
    {
        D[0]=16;
        D[1]=14;
        D[2]=10;
        D[3]=10;
        D[4]=0;
        D[5]=10;
    }
}
}
}

```