



## รายงาน

โปรแกรมคำนวณปริมาตรรูปทรง 4 แบบ (ทศนิยม 2 ตำแหน่ง)

จัดทำโดย

นางสาวชาติศิริ	เกตุแก้ว	รหัสนักศึกษา	5735512011
นายวีรศักดิ์	ราชวังเมือง	รหัสนักศึกษา	5735512083
นางสาวสุธิดา	แลบัว	รหัสนักศึกษา	5735512134

Section 01

เสนอ

อาจารย์ พัชรีย์ เทพนิมิตร

รายงานนี้เป็นส่วนหนึ่งของวิชา 242-309 Microcontroller and Interfacing

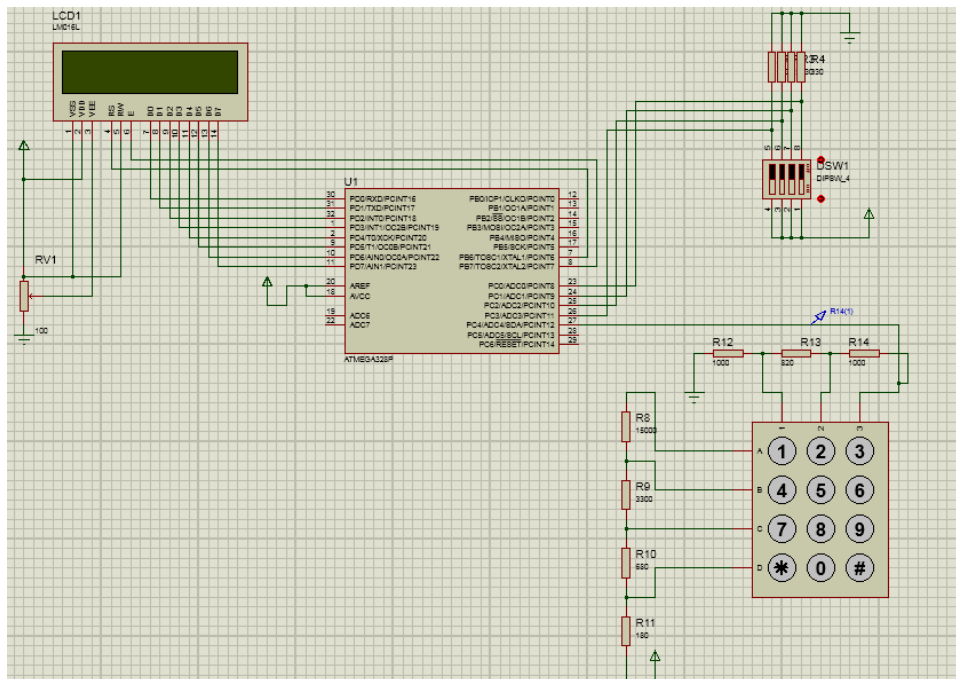
ภาคการศึกษาที่ 2 ปีการศึกษา 2559

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยสงขลานครินทร์

## โปรแกรมคำนวณปริมาตรรูปทรง 4 แบบ (ทศนิยม 2 ตำแหน่ง)

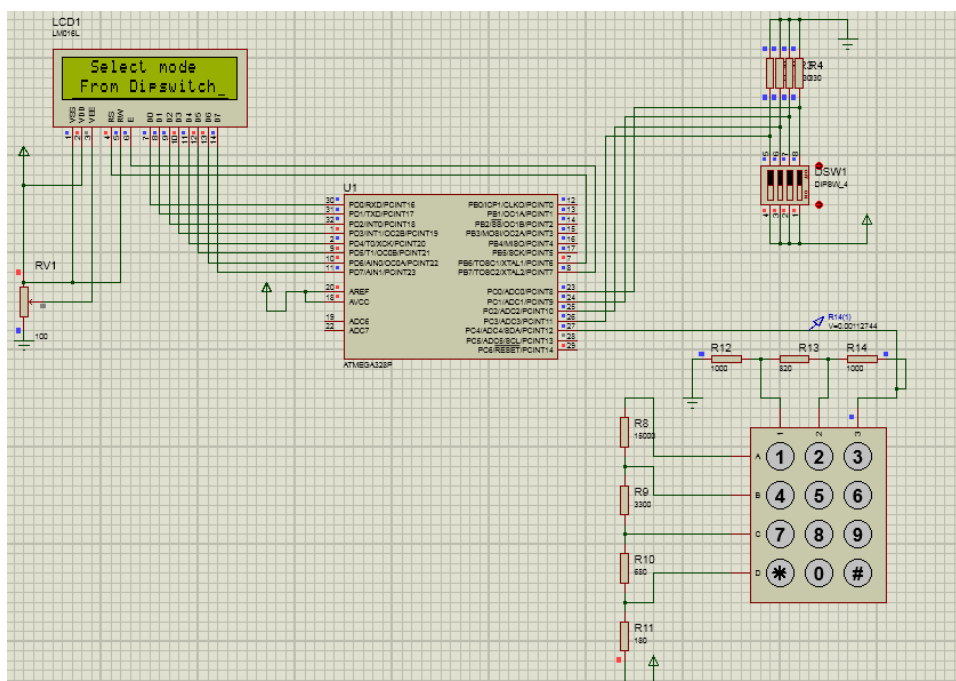
รูปร่าง:



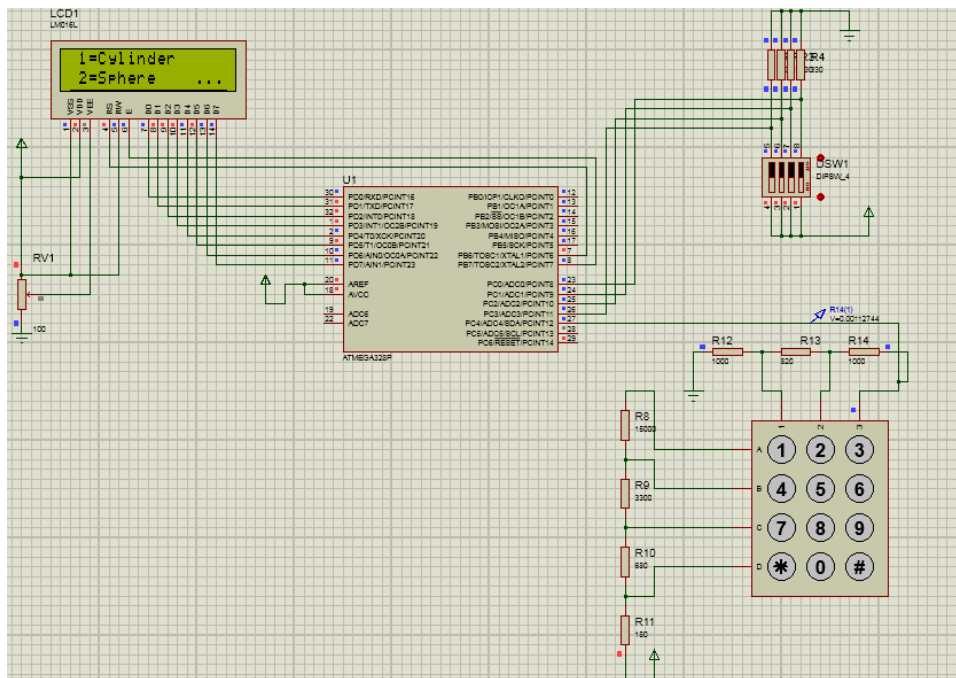
ภาพรวมวงจรก่อนการ Run Program

รายละเอียดการทำงาน:

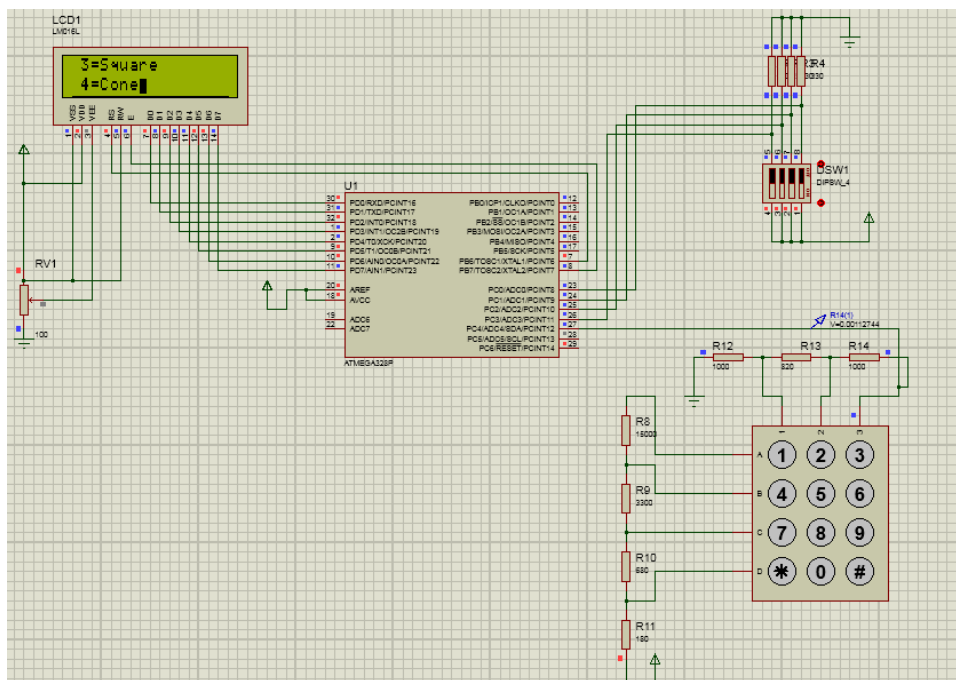
เมื่อ Run Program จะขึ้นข้อความจาก LCD ว่า “Select mode from Dipswitch”



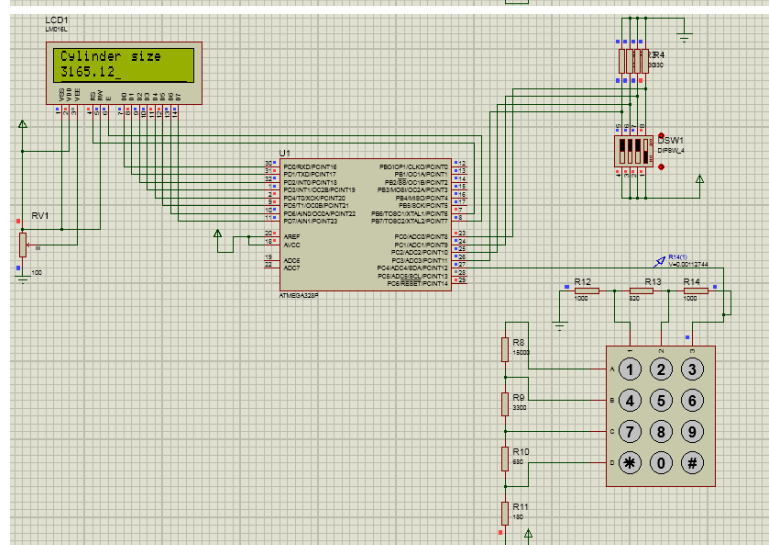
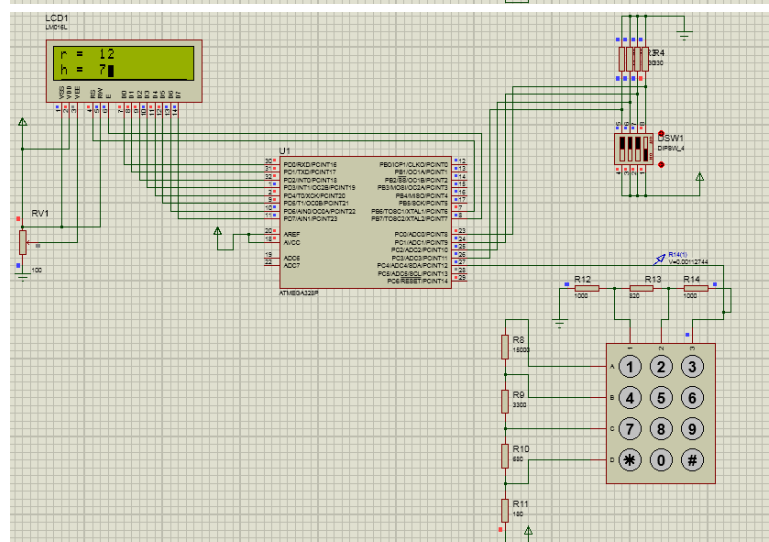
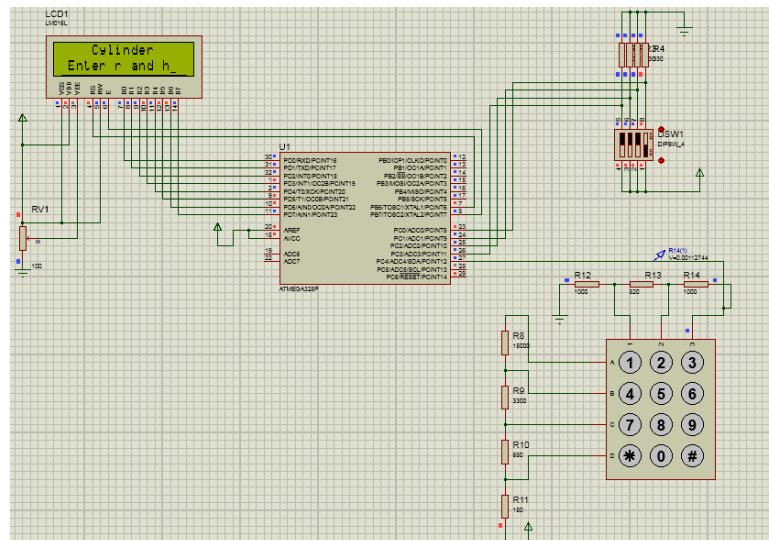
เลือก Mode การทำงาน 1=Cylinder 2=Sphere



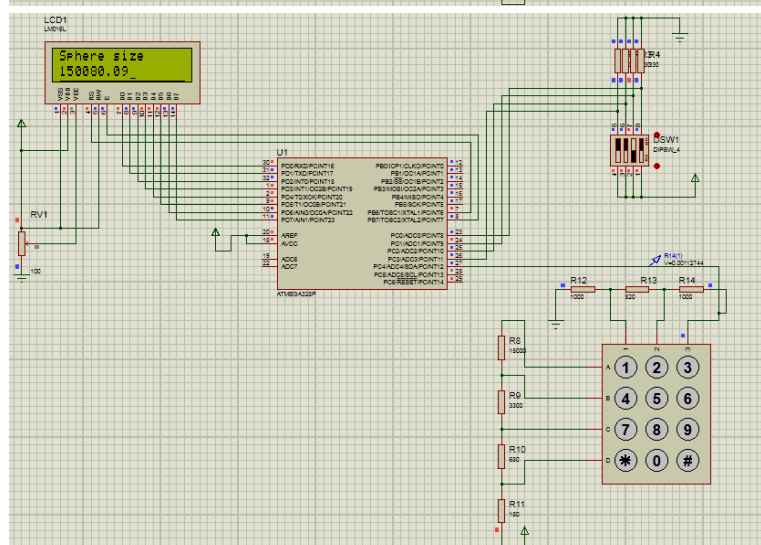
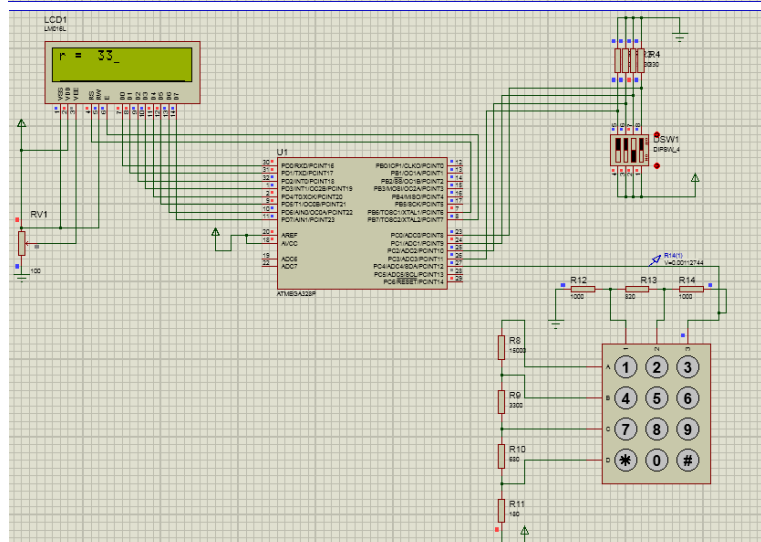
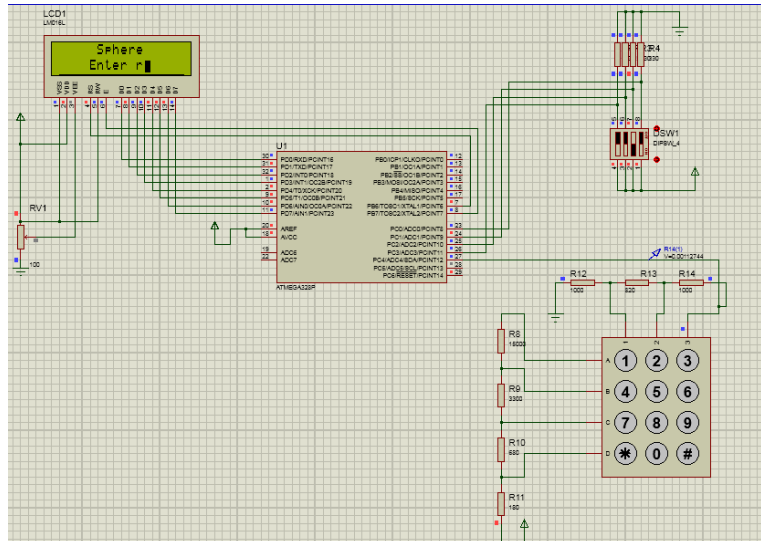
เลือก Mode การทำงาน 3=Square 4=Cone



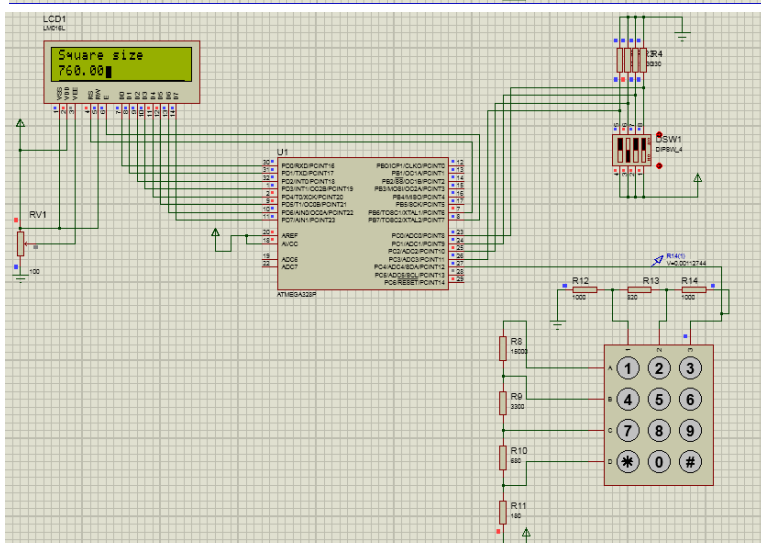
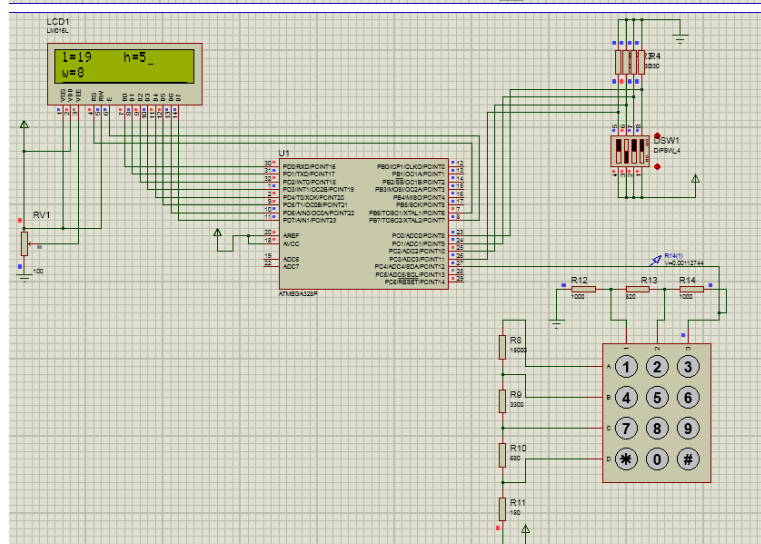
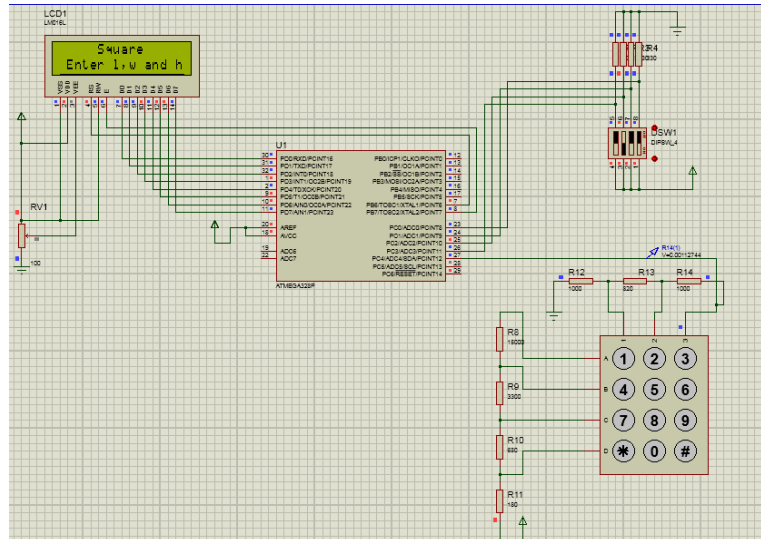
เมื่อผู้ใช้เลือก mode ที่ 1 (0001) จาก dipswitch LCD จะแสดงข้อความบอกชื่อรูปทรงและให้ผู้ใช้ป้อนค่าต่าง ๆ โดยกดป้อนข้อมูลจาก Keypad แล้วกด \* เพื่อป้อนค่าตัวถัดไปเมื่อป้อนค่าจนครบจากนั้นกด # เพื่อส่งค่าไปคำนวณ LCD จะแสดงค่าคำตอบออกมา



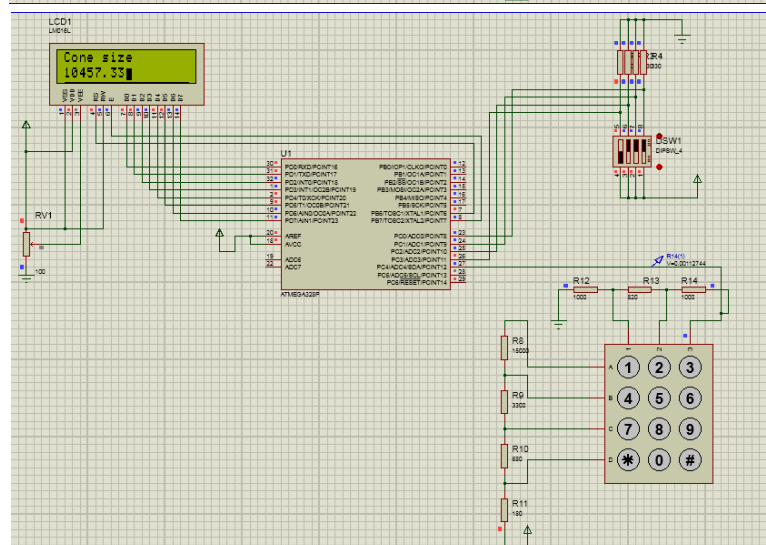
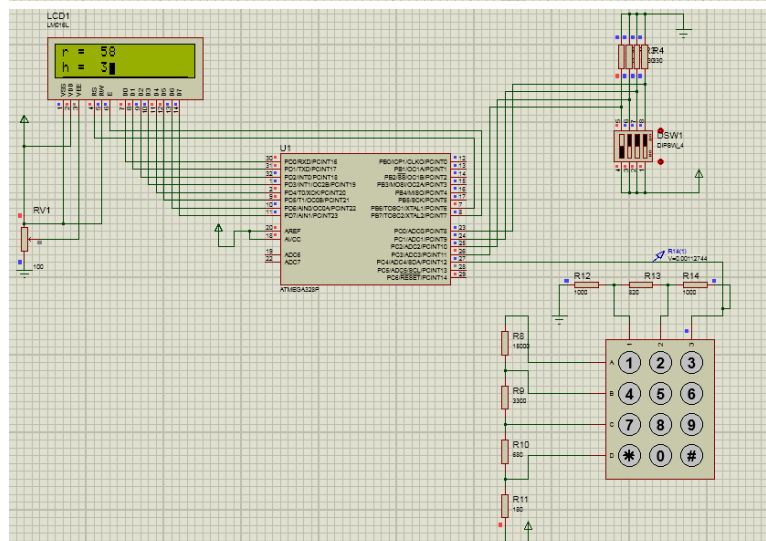
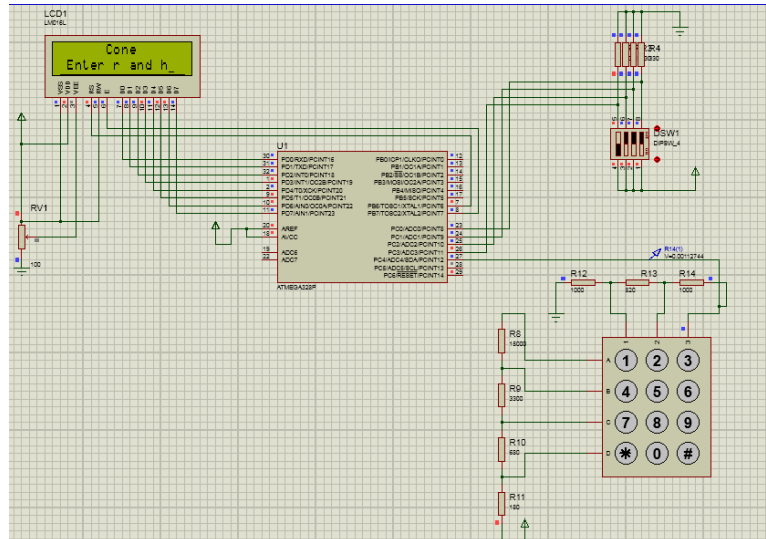
เมื่อผู้ใช้เลือก mode ที่ 2 (0010) จาก dipswitch LCD จะแสดงข้อความบอกชื่อรูปทรงและให้ผู้ใช้ป้อนค่าต่าง ๆ โดยกดป้อนข้อมูลจาก Keypad แล้วกด \* เพื่อป้อนค่าตัวถัดไปเมื่อป้อนค่าจนครบจากนั้นกด # เพื่อส่งค่าไปคำนวณ LCD จะแสดงค่าคำตอบออกมา



เมื่อผู้ใช้เลือก mode ที่ 3 (0100) จาก dipswitch LCD จะแสดงข้อความบอกชื่อรูปทรงและให้ผู้ใช้ป้อนค่าต่าง ๆ โดยกดป้อนข้อมูลจาก Keypad แล้วกด \* เพื่อป้อนค่าตัวถัดไปเมื่อป้อนค่าจนครบจากนั้นกด # เพื่อส่งค่าไปคำนวณ LCD จะแสดงค่าคำตอบออกมา



เมื่อผู้ใช้เลือก mode ที่ 4 (1000) จาก dipswitch LCD จะแสดงข้อความบอกชื่อรูปทรงและให้ผู้ใช้ป้อนค่าต่าง ๆ โดยกดป้อนข้อมูลจาก Keypad แล้วกด \* เพื่อป้อนค่าตัวถัดไปเมื่อป้อนค่าจนครบจากนั้นกด # เพื่อส่งค่าไปคำนวณ LCD จะแสดงค่าคำตอบออกมา



#### Input:

1. Dipswitch 4 pin
2. Keypad

#### Output:

16 x 2 Character LCD

#### เทคนิคที่ใช้ในวงจร:

1. ใช้ความรู้เรื่อง PIN Change interrupt ในการเก็บสถานะการเปลี่ยนแปลงขาที่ต่อกับ Dipswitch เพื่อใช้ในการเลือกโหมดรูปทรง 4 รูปทรง
2. ใช้ความรู้เรื่อง Timer/Counter1 (วงจรจับเวลา/นับ ขนาด 16 บิต ) และโปรแกรมบริการ Interrupt ของ Timer/Counter1 ใช้ Timer1 ในการให้จังหวะสัญญาณนาฬิกาการร่วมกับการแปลง Analog to Digital ของ Keypad
3. ใช้ความรู้เรื่อง Delay ในการแสดงผลของจอ LCD โดยจะใช้ Delay ในการหน่วงเวลาเพื่อแสดงข้อความใหม่บนจอ LCD

#### Code:

```
#ifndef F_CPU
#define F_CPU 16000000UL // 16 MHz clock speed
#endif

#define D0 eS_PORTD0
#define D1 eS_PORTD1
#define D2 eS_PORTD2
#define D3 eS_PORTD3
#define D4 eS_PORTD4
#define D5 eS_PORTD5
#define D6 eS_PORTD6
#define D7 eS_PORTD7
#define RS eS_PORTB6
#define EN eS_PORTB7

#include <inttypes.h>
#include <avr/io.h>
#include <avr/interrupt.h>
```



```
#include <avr/sleep.h>
#include <stdlib.h>
#include <util/delay.h>

// set LCD Functions จาก electroSome

#define eS_PORTB0 0
#define eS_PORTB1 1
#define eS_PORTB2 2
#define eS_PORTB3 3
#define eS_PORTB4 4
#define eS_PORTB5 5
#define eS_PORTB6 6
#define eS_PORTB7 7
#define eS_PORTC0 10
#define eS_PORTC1 11
#define eS_PORTC2 12
#define eS_PORTC3 13
#define eS_PORTC4 14
#define eS_PORTC5 15
#define eS_PORTC6 16
#define eS_PORTD0 20
#define eS_PORTD1 21
#define eS_PORTD2 22
#define eS_PORTD3 23
#define eS_PORTD4 24
#define eS_PORTD5 25
#define eS_PORTD6 26
#define eS_PORTD7 27

#ifndef D0
#define D0 eS_PORTD0
#define D1 eS_PORTD1
#define D2 eS_PORTD2
#define D3 eS_PORTD3
#endif

#define INTERVAL_time 65528
void pinChange(int a, int b){
    if(b == 0){
```

```

        if(a == eS_PORTB0) PORTB &= ~(1<<PB0);
        else if(a == eS_PORTB1) PORTB &= ~(1<<PB1);
        else if(a == eS_PORTB2) PORTB &= ~(1<<PB2);
        else if(a == eS_PORTB3) PORTB &= ~(1<<PB3);
        else if(a == eS_PORTB4) PORTB &= ~(1<<PB4);
        else if(a == eS_PORTB5) PORTB &= ~(1<<PB5);
        else if(a == eS_PORTB6) PORTB &= ~(1<<PB6);
        else if(a == eS_PORTB7) PORTB &= ~(1<<PB7);
        else if(a == eS_PORTC0) PORTC &= ~(1<<PC0);
        else if(a == eS_PORTC1) PORTC &= ~(1<<PC1);
        else if(a == eS_PORTC2) PORTC &= ~(1<<PC2);
        else if(a == eS_PORTC3) PORTC &= ~(1<<PC3);
        else if(a == eS_PORTC4) PORTC &= ~(1<<PC4);
        else if(a == eS_PORTC5) PORTC &= ~(1<<PC5);
        else if(a == eS_PORTD0) PORTD &= ~(1<<PD0);
        else if(a == eS_PORTD1) PORTD &= ~(1<<PD1);
        else if(a == eS_PORTD2) PORTD &= ~(1<<PD2);
        else if(a == eS_PORTD3) PORTD &= ~(1<<PD3);
        else if(a == eS_PORTD4) PORTD &= ~(1<<PD4);
        else if(a == eS_PORTD5) PORTD &= ~(1<<PD5);
        else if(a == eS_PORTD6) PORTD &= ~(1<<PD6);
        else if(a == eS_PORTD7) PORTD &= ~(1<<PD7);
    }
    else {
        if(a == eS_PORTB0) PORTB |= (1<<PB0);
        else if(a == eS_PORTB1) PORTB |= (1<<PB1);
        else if(a == eS_PORTB2) PORTB |= (1<<PB2);
        else if(a == eS_PORTB3) PORTB |= (1<<PB3);
        else if(a == eS_PORTB4) PORTB |= (1<<PB4);
        else if(a == eS_PORTB5) PORTB |= (1<<PB5);
        else if(a == eS_PORTB6) PORTB |= (1<<PB6);
        else if(a == eS_PORTB7) PORTB |= (1<<PB7);
        else if(a == eS_PORTC0) PORTC |= (1<<PC0);
        else if(a == eS_PORTC1) PORTC |= (1<<PC1);
        else if(a == eS_PORTC2) PORTC |= (1<<PC2);
        else if(a == eS_PORTC3) PORTC |= (1<<PC3);
        else if(a == eS_PORTC4) PORTC |= (1<<PC4);
    }

```

```

        else if(a == eS_PORTC5) PORTC |= (1<<PC5);
        else if(a == eS_PORTC6) PORTC |= (1<<PC6);
        else if(a == eS_PORTD0) PORTD |= (1<<PD0);
        else if(a == eS_PORTD1) PORTD |= (1<<PD1);
        else if(a == eS_PORTD2) PORTD |= (1<<PD2);
        else if(a == eS_PORTD3) PORTD |= (1<<PD3);
        else if(a == eS_PORTD4) PORTD |= (1<<PD4);
        else if(a == eS_PORTD5) PORTD |= (1<<PD5);
        else if(a == eS_PORTD6) PORTD |= (1<<PD6);
        else if(a == eS_PORTD7) PORTD |= (1<<PD7);
    }
}

```

// LCD 8 Bit Interfacing Functions set การใช้งาน 8 bit โหมด

```

void Lcd8_Port(char a){
    if(a &1) pinChange(D0,1);
    else pinChange(D0,0);

    if(a &2) pinChange(D1,1);
    else pinChange(D1,0);

    if(a &4) pinChange(D2,1);
    else pinChange(D2,0);

    if(a &8) pinChange(D3,1);
    else pinChange(D3,0);

    if(a &16) pinChange(D4,1);
    else pinChange(D4,0);

    if(a &32) pinChange(D5,1);
    else pinChange(D5,0);

    if(a &64) pinChange(D6,1);
    else pinChange(D6,0);

    if(a &128) pinChange(D7,1);
    else pinChange(D7,0);
}

```

```

void Lcd8_Cmd(char a){
    pinChange(RS,0);        // => RS = 0
    Lcd8_Port(a);           // Data transfer
}

```

```

        pinChange(EN,1);          // => E = 1
        _delay_ms(1);
        pinChange(EN,0);          // => E = 0
        _delay_ms(1);
    }

void Lcd8_Clear(){
    Lcd8_Cmd(1);
}

// set ตำแหน่งของ cursor
void Lcd8_Set_Cursor(char a, char b){
    if(a == 1) Lcd8_Cmd(0x80 + b);
    else if(a == 2) Lcd8_Cmd(0xC0 + b);
}

void Lcd8_Init(){
    pinChange(RS,0);
    pinChange(EN,0);
    _delay_ms(20);
    //////////// Reset process from datasheet ////////////
    Lcd8_Cmd(0x30);
    _delay_ms(5);
    Lcd8_Cmd(0x30);
    _delay_ms(1);
    Lcd8_Cmd(0x30);
    _delay_ms(10);
    //////////////////////////////////////
    Lcd8_Cmd(0x38);  // function set
    Lcd8_Cmd(0x0F);  // display on,cursor off,blink off
    Lcd8_Cmd(0x01);  // clear display
    Lcd8_Cmd(0x06);  // entry mode, set increment
}

void Lcd8_Write_Char(char a){
    pinChange(RS,1);          // => RS = 1
    Lcd8_Port(a);              // Data transfer
    pinChange(EN,1);          // => E = 1
    _delay_ms(1);
    pinChange(EN,0);          // => E = 0
}

```

```

        _delay_ms(1);
    }
    // set การแสดงข้อความออกทาง LCD
    void Lcd8_Write_String(char *a){
        int i;
        for(i=0;a[i]!='\0';i++)
            Lcd8_Write_Char(a[i]);
    }
    // set ให้ข้อความเลื่อนไปทางขวา
    void Lcd8_Shift_Right(){
        Lcd8_Cmd(0x1C);
    }
    // set ให้ข้อความเลื่อนไปทางซ้าย
    void Lcd8_Shift_Left(){
        Lcd8_Cmd(0x18);
    }
    // End LCD 8 Bit Interfacing Functions

    // *****Function การทำงานหลัก*****
    unsigned char sw=0;
    unsigned short ACCUMULATED, DIVISOR, QUOTIENT, DECODED;
    unsigned int m,keep=0;
    unsigned long NUMBERS=0,r=0,h=0,w=0,l=0;

    int main(void){
        // setinput, output
        DDRD = 0xFF;
        DDRB = 0xFF;
        DDRC = 0x00;
        // setLCD
        Lcd8_Init();
        // set การแปลงสัญญาณ Analog เป็น Digital ของ Keypad
        ADMUX = 0b00100100;
        ADCSRA = 0b10101101;
        ADCSRB = 0x06;
        // set timer ของ Keypad
        TCNT1 = INTERVAL_time;
    }

```

```

TCCR1A = 0x00;
TCCR1B = 0x05;
TIMSK1 = 0x01;
// set Interrupt ของ Dipswitch
PCICR = 0x02 ;
PCMSK1 = 0x0F ;

sei();
// เริ่มต้นโชว์ข้อความออกจาก LCD ให้ผู้ใช้เลือกโหมด
Lcd8_Set_Cursor(1,2);
Lcd8_Write_String("Select mode");
Lcd8_Set_Cursor(2,1);
Lcd8_Write_String("From Dipswitch");
_delay_ms(1000);
Lcd8_Clear();
Lcd8_Set_Cursor(1,1);
Lcd8_Write_String("1=Cylinder");
Lcd8_Set_Cursor(2,1);
Lcd8_Write_String("2=Sphere   ...");
_delay_ms(1000);
Lcd8_Clear();
Lcd8_Set_Cursor(1,1);
Lcd8_Write_String("3=Square");
Lcd8_Set_Cursor(2,1);
Lcd8_Write_String("4=Cone");

while(1);
}

ISR(TIMER1_OVF_vect){
    TCNT1 = INTERVAL_time;
}

// Dipswitch interrupt Function
ISR(PCINT1_vect){
    sw = PINC ;      // รับค่ามาจาก Port C
    sw &= 0x0F ;     // ตัด bit 4-8
    // ถ้าไม่มีการสับ Switch จะโชว์ข้อความให้ผู้ใช้เลือกโหมด
    if(sw == 0x00){
        Lcd8_Clear();
    }
}

```

```

    Lcd8_Set_Cursor(1,2);
    Lcd8_Write_String("Select mode");
    Lcd8_Set_Cursor(2,1);
    Lcd8_Write_String("From Dipswitch");
    m=0;
}
// ถ้าผู้ใช้เลือกโหมดที่ 1 (Cylinder) จะแสดงข้อความให้ผู้ป้อนค่า Input ต่าง ๆ
else if(sw == 0x01){
    Lcd8_Clear();
    Lcd8_Set_Cursor(1,4);
    Lcd8_Write_String("Cylinder");
    Lcd8_Set_Cursor(2,1);
    Lcd8_Write_String("Enter r and h");
    _delay_ms(250);
    Lcd8_Clear();
    _delay_ms(250);
    Lcd8_Set_Cursor(1,0);
    Lcd8_Write_String("r = ");
    Lcd8_Set_Cursor(2,0);
    Lcd8_Write_String("h = ");
    Lcd8_Set_Cursor(1,5);
    m=1;
}
// ถ้าผู้ใช้เลือกโหมดที่ 2 (Sphere) จะแสดงข้อความให้ผู้ป้อนค่า Input ต่าง ๆ
else if(sw == 0x02){
    Lcd8_Clear();
    Lcd8_Set_Cursor(1,5);
    Lcd8_Write_String("Sphere");
    Lcd8_Set_Cursor(2,4);
    Lcd8_Write_String("Enter r");
    _delay_ms(250);
    Lcd8_Clear();
    _delay_ms(250);
    Lcd8_Set_Cursor(1,0);
    Lcd8_Write_String("r = ");
    Lcd8_Set_Cursor(1,5);
    m=2;
}

```

```

}
// ถ้าผู้ใช้เลือกโหมดที่ 3 (Square) จะแสดงข้อความให้ผู้ป้อนค่า Input ต่าง ๆ
else if(sw == 0x04){
    Lcd8_Clear();
    Lcd8_Set_Cursor(1,5);
    Lcd8_Write_String("Square");
    Lcd8_Set_Cursor(2,1);
    Lcd8_Write_String("Enter l,w and h");
    _delay_ms(250);
    Lcd8_Clear();
    _delay_ms(250);
    Lcd8_Set_Cursor(1,0);
    Lcd8_Write_String("l=");
    Lcd8_Set_Cursor(1,8);
    Lcd8_Write_String("h=");
    Lcd8_Set_Cursor(2,0);
    Lcd8_Write_String("w=");
    Lcd8_Set_Cursor(1,2);
    m=3;
}
// ถ้าผู้ใช้เลือกโหมดที่ 4 (Cone) จะแสดงข้อความให้ผู้ป้อนค่า Input ต่าง ๆ
else if(sw == 0x08){
    Lcd8_Clear();
    Lcd8_Set_Cursor(1,6);
    Lcd8_Write_String("Cone");
    Lcd8_Set_Cursor(2,1);
    Lcd8_Write_String("Enter r and h");
    _delay_ms(250);
    Lcd8_Clear();
    _delay_ms(250);
    Lcd8_Set_Cursor(1,0);
    Lcd8_Write_String("r = ");
    Lcd8_Set_Cursor(2,0);
    Lcd8_Write_String("h = ");
    Lcd8_Set_Cursor(1,5);
    m=4;
}

```



```

}
// function การทำงานของ Keypad
ISR(ADC_vect){
    if(ADCH >9){
        ACCUMULATED += ADCH;
        DIVISOR++;
    }
    else{
        ACCUMULATED = 0;
        DIVISOR = 0;
    }
    if(DIVISOR == 10){
        QUOTIENT = ACCUMULATED/DIVISOR;
        if((QUOTIENT>9)&&(QUOTIENT<16)){
            DECODED = 1;
            Lcd8_Write_String("1");
        }
        else if((QUOTIENT>18)&&(QUOTIENT<26)){
            DECODED = 2;
            Lcd8_Write_String("2");
        }
        else if((QUOTIENT>29)&&(QUOTIENT<37)){
            DECODED = 3;
            Lcd8_Write_String("3");
        }
        else if((QUOTIENT>46)&&(QUOTIENT<53)){
            DECODED = 4;
            Lcd8_Write_String("4");
        }
        else if((QUOTIENT>74)&&(QUOTIENT<81)){
            DECODED = 5;
            Lcd8_Write_String("5");
        }
        else if((QUOTIENT>99)&&(QUOTIENT<107)){
            DECODED = 6;
            Lcd8_Write_String("6");
        }
    }
}

```

```

else if((QUOTIENT>133)&&(QUOTIENT<141)){
    DECODED = 7;
    Lcd8_Write_String("7");
}
else if((QUOTIENT>169)&&(QUOTIENT<177)){
    DECODED = 8;
    Lcd8_Write_String("8");
}
else if((QUOTIENT>192)&&(QUOTIENT<199)){
    DECODED = 9;
    Lcd8_Write_String("9");
}
else if((QUOTIENT>212)&&(QUOTIENT<220))
    DECODED = 10;
else if((QUOTIENT>228)&&(QUOTIENT<236)){
    DECODED = 0;
    Lcd8_Write_String("0");
}
else if((QUOTIENT>236)&&(QUOTIENT<244))
    DECODED = 11;
else
    Lcd8_Write_String("Error!");
// เมื่อผู้ใช้เลือกโหมดที่ 1 (Cylinder) ผู้จะต้องป้อน Input ต่าง ๆ จาก Keypad
if(m==1){
    if(DECODED != 10&& DECODED != 11){
        NUMBERS += DECODED;
        if(keep==0){
            r = NUMBERS;
            NUMBERS = NUMBERS*10;
        }
        else{
            h = NUMBERS;
            NUMBERS = NUMBERS*10;
        }
    }
    else if(DECODED == 10){
        keep++;
    }
}

```

```

    Lcd8_Set_Cursor(2,5);
    NUMBERS = 0;
}
// ส่วนของการคำนวณและแสดงผลออกทาง LCD
else if(DECODED == 11){
    long ans = 314 * r * r * h;
    long d1,d2;
    d1 = ans/100;
    d2 = ans%100;
    char c1[20],c2[20];
    ltoa(d1,c1,10);
    ltoa(d2,c2,10);
    Lcd8_Clear();
    Lcd8_Set_Cursor(1,0);
    Lcd8_Write_String("Cylinder size");
    Lcd8_Set_Cursor(2,0);
    Lcd8_Write_String(c1);
    Lcd8_Write_String(".");
    if(d2<10) Lcd8_Write_String("0");
    Lcd8_Write_String(c2);
    r=0;
    h=0;
    NUMBERS = 0;
    keep=0;
}
else
    Lcd8_Write_String("Error!");
}
// เมื่อผู้ใช้เลือกโหมดที่ 2 (Sphere) ผู้จะต้องป้อน Input ต่าง ๆ จาก Keypad
else if(m==2){
    if(DECODED != 10&& DECODED != 11){
        NUMBERS += DECODED;
        r = NUMBERS;
        NUMBERS = NUMBERS*10;
    }
    else if(DECODED == 10){
        Lcd8_Set_Cursor(2,0);

```

```

    Lcd8_Write_String("Press #");
    NUMBERS = 0;
}
// ส่วนของการคำนวณและแสดงผลออกทาง LCD
else if(DECODED == 11){
    long ans = 314*(r * r * r);
    ans *= 133;
    long d1,d2;
    d1 = ans/10000;
    d2 = ans/100%100;
    char c1[20],c2[20];
    ltoa(d1,c1,10);
    ltoa(d2,c2,10);
    Lcd8_Clear();
    Lcd8_Set_Cursor(1,0);
    Lcd8_Write_String("Sphere size");
    Lcd8_Set_Cursor(2,0);
    Lcd8_Write_String(c1);
    Lcd8_Write_String(".");
    if(d2<10) Lcd8_Write_String("0");
    Lcd8_Write_String(c2);
    r=0;
    NUMBERS = 0;
    keep=0;
}
else
    Lcd8_Write_String("Error!");
}
// เมื่อผู้ใช้เลือกโหมดที่ 3 (Square) ผู้จะต้องป้อน Input ต่าง ๆ จาก Keypad
else if(m==3){
    if(DECODED != 10&& DECODED != 11){
        NUMBERS += DECODED;
        if(keep==0){
            h = NUMBERS;
            NUMBERS = NUMBERS*10;
        }
        else if(keep==1){

```

```

        w = NUMBERS;
        NUMBERS = NUMBERS*10;
    }
    else{
        l = NUMBERS;
        NUMBERS = NUMBERS*10;
    }
}
else if(DECODED == 10){
    Lcd8_Set_Cursor(2,2);
    if(keep==1){
        Lcd8_Set_Cursor(1,10);
    }
    keep++;
    NUMBERS = 0;
}
// ส่วนของการคำนวณและแสดงผลออกทาง LCD
else if(DECODED == 11){
    long ans = h * w * l;
    char c1[20];
    ltoa(ans,c1,10);
    Lcd8_Clear();
    Lcd8_Set_Cursor(1,0);
    Lcd8_Write_String("Square size");
    Lcd8_Set_Cursor(2,0);
    Lcd8_Write_String(c1);
    Lcd8_Write_String(".00");
    h=0;
    w=0;
    l=0;
    NUMBERS = 0;
    keep=0;
}
else
    Lcd8_Write_String("Error!");
}

```

// เมื่อผู้ใช้เลือกโหมดที่ 4 (Cone) ผู้จะต้องป้อน Input ต่าง ๆ จาก Keypad

```
else if(m==4){
    if(DECODED != 10&& DECODED != 11){
        NUMBERS += DECODED;
        if(keep==0){
            r = NUMBERS;
            NUMBERS = NUMBERS*10;
        }
        else{
            h = NUMBERS;
            NUMBERS = NUMBERS*10;
        }
    }
    else if(DECODED == 10){
        keep++;
        Lcd8_Set_Cursor(2,5);
        NUMBERS = 0;
    }
}
```

// ส่วนของการคำนวณและแสดงผลออกทาง LCD

```
else if(DECODED == 11){
    long ans = 314 * 33 * r * r * h;
    long d1,d2;
    d1 = ans/10000;
    d2 = ans/100%100;
    char c1[20],c2[20];
    ltoa(d1,c1,10);
    ltoa(d2,c2,10);
    Lcd8_Clear();
    Lcd8_Set_Cursor(1,0);
    Lcd8_Write_String("Cone size");
    Lcd8_Set_Cursor(2,0);
    Lcd8_Write_String(c1);
    Lcd8_Write_String(".");
    if(d2<10) Lcd8_Write_String("0");
    Lcd8_Write_String(c2);
    r=0;
    h=0;
}
```

```
        NUMBERS = 0;
        keep=0;
    }
    else
        Lcd8_Write_String("Error!");
}
}
}
```