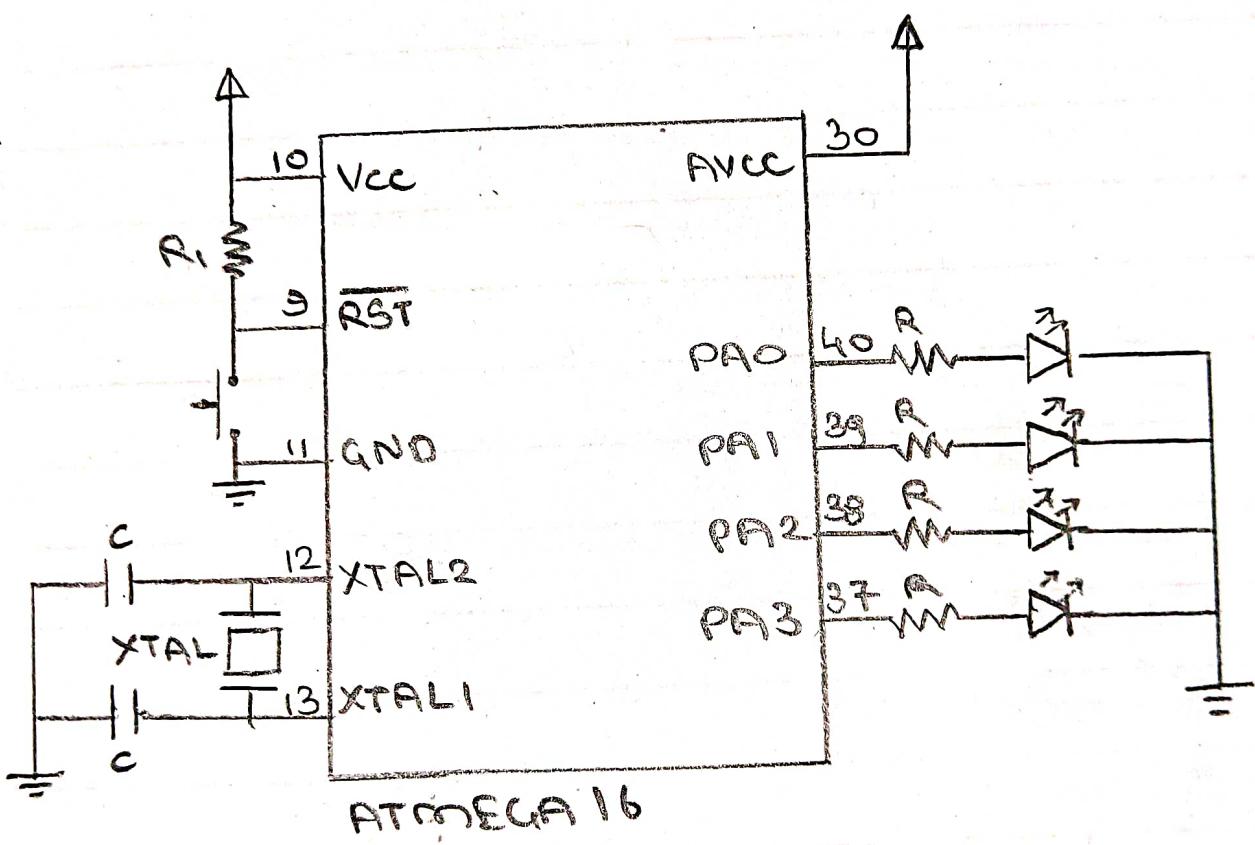


Expt. No.

Date :

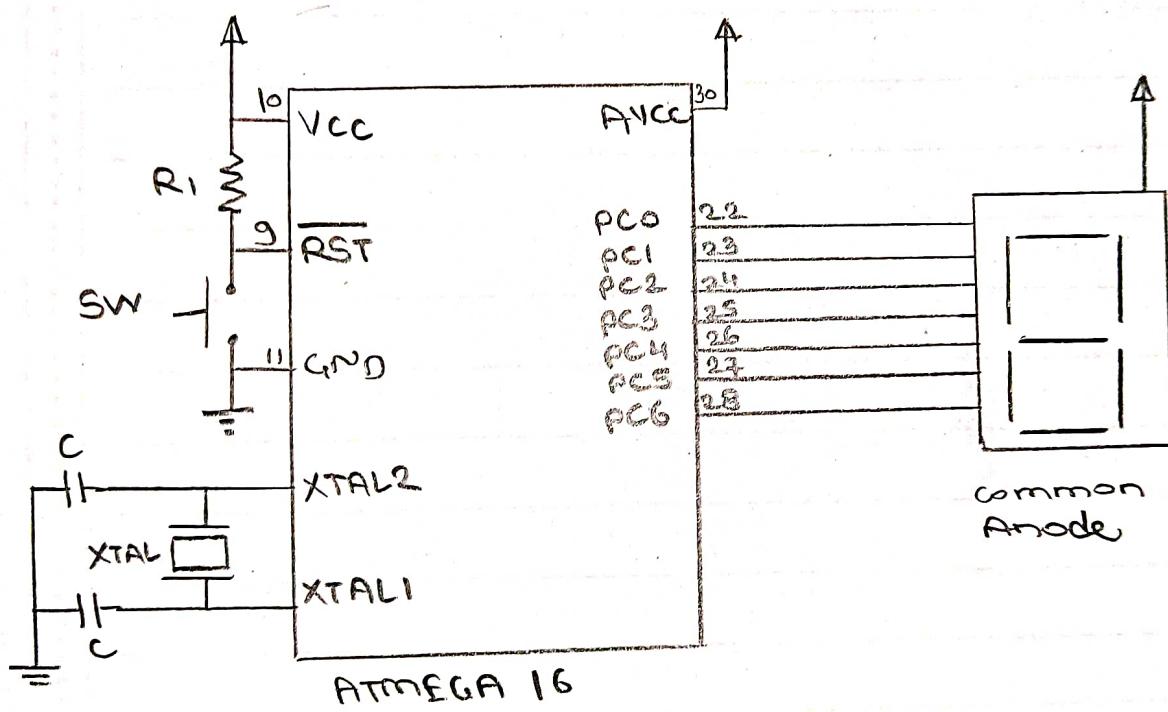
```
#include <avr/io.h>
#include <util/delay.h>
#define F_CPU 16000000
void main()
{
    DDRD = 0xFF;
    int arr[] = {0x01, 0x02, 0x04, 0x08};
    while (1)
    {
        for (int i=0; i<4; i++)
        {
            PORTD = arr[i];
            delay_ms(100);
        }
    }
}
```



Expt. No.

Date :

```
#include <avr/io.h>
#include <util/delay.h>
#define F_CPU = 16000000
int main (void)
{
    DDRA = 0xff;
    while (1) {
        for (int i=0 ; i<4 ; i++) {
            PORTA = (0x01<<i);
            -delay-ms(50);
        }
        for (int j=0 ; j<4 ; j++) {
            PORTA = (0x08>>j);
            -delay-ms(50);
        }
    }
}
```

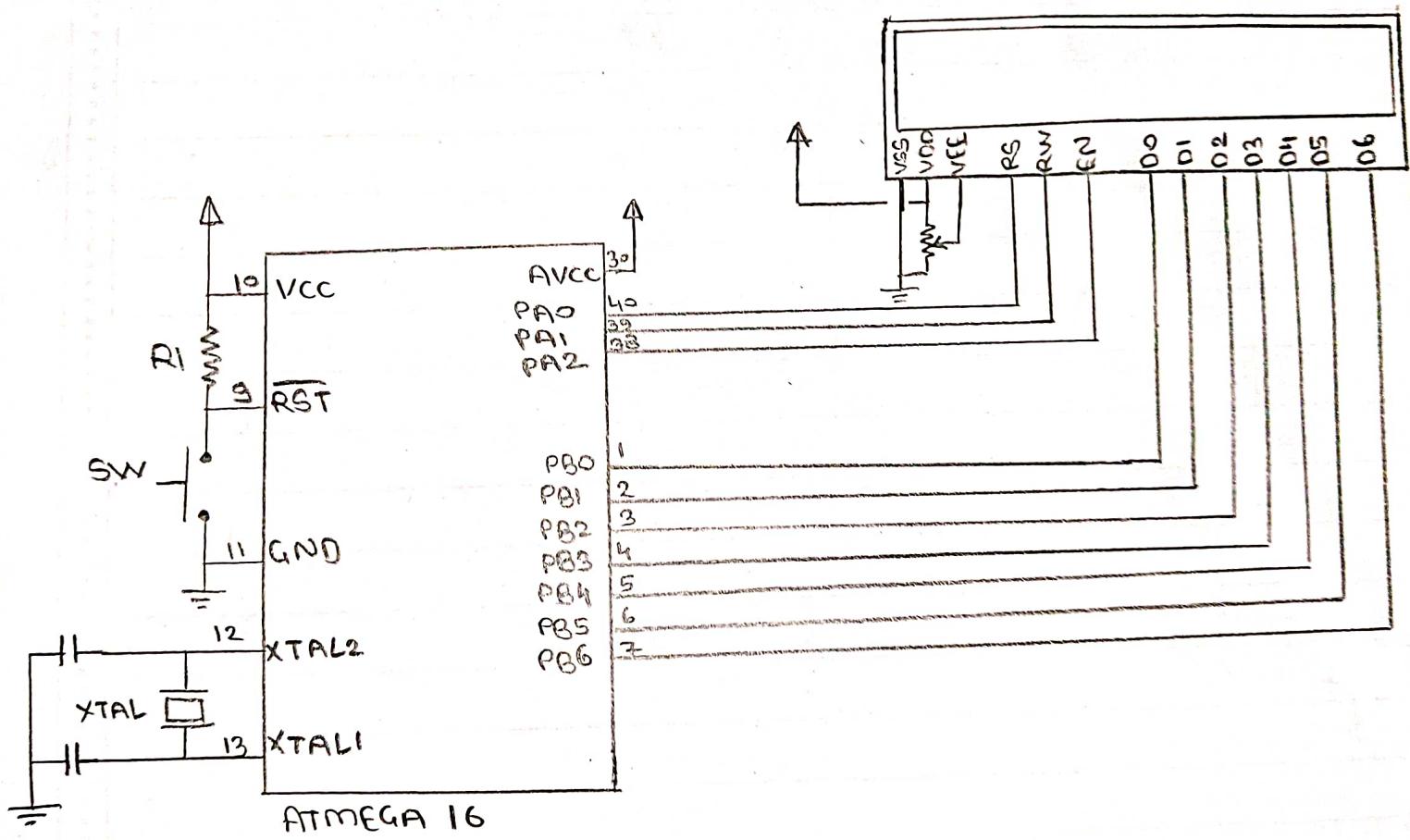


Expt. No.

Date :

```
#include<avr/io.h>
#include<util/delay.h>
#define f_CPU 16000000
int main()
{
    char Seq_code[] = {0x00, 0xf9, 0xa4, 0xb0,
                       0x99, 0x92, 0x82, 0xfa,
                       0x80, 0x90, 0x88, 0x83,
                       0xc6, 0x11, 0x86, 0x90};

    int cnt=0;
    DDRD=0xFF;
    while(1)
    {
        PORTC = Seq_code[cnt];
        _delay_ms(25);
    }
}
```



Expt. No.

Date :

```

#include <avr/io.h>
#include <util/delay.h>
#define F_CPU 16000000
#define LCD_Data_Dir DDRB
#define LCD_Command_Dir DDRA
#define LCD_Data_Port PORTB
#define LCD_Command_Port PORTA
#define RS PA0
#define RW PA1
#define EN PA2

```

void LED_Command (unsigned char cmd)

```

{
    LCD_Data_Port = cmd;
    LCD_Command_Port &= ~(1<<RS);
    LCD_Command_Port &= ~(1<<RW);
    LCD_Command_Port |= (1<<EN);
    -delay_ms(1);
    LCD_Command_Port &= ~(1<<EN);
    -delay_ms(1);
}

```

void LCD_Init (void)

```

{
    LCD_Command_Dir = 0xFF;
    LCD_Data_Dir = 0xFF;
    -delay_ms(20);
}

```

```
LCD_Command(0x38);  
LCD_Command(0x0C);  
LCD_Command(0x06);  
LCD_Command(0x01);  
LCD_Command(0x80);
```

```
3 Void LCD_Char (unsigned char char_data)  
{
```

```
    LCD_Data_Port = char_data;  
    LCD_Command_Port |= (1<<RS);  
    LCD_Command_Port &= ~(1<<RW);  
    LCD_Command_Port |= (1<<EN);  
    -delay_ms(1);  
    LCD_Command_Port &= ~(1<<EN);  
    -delay_ms(1);
```

```
4 Void LCD_String (char *str)
```

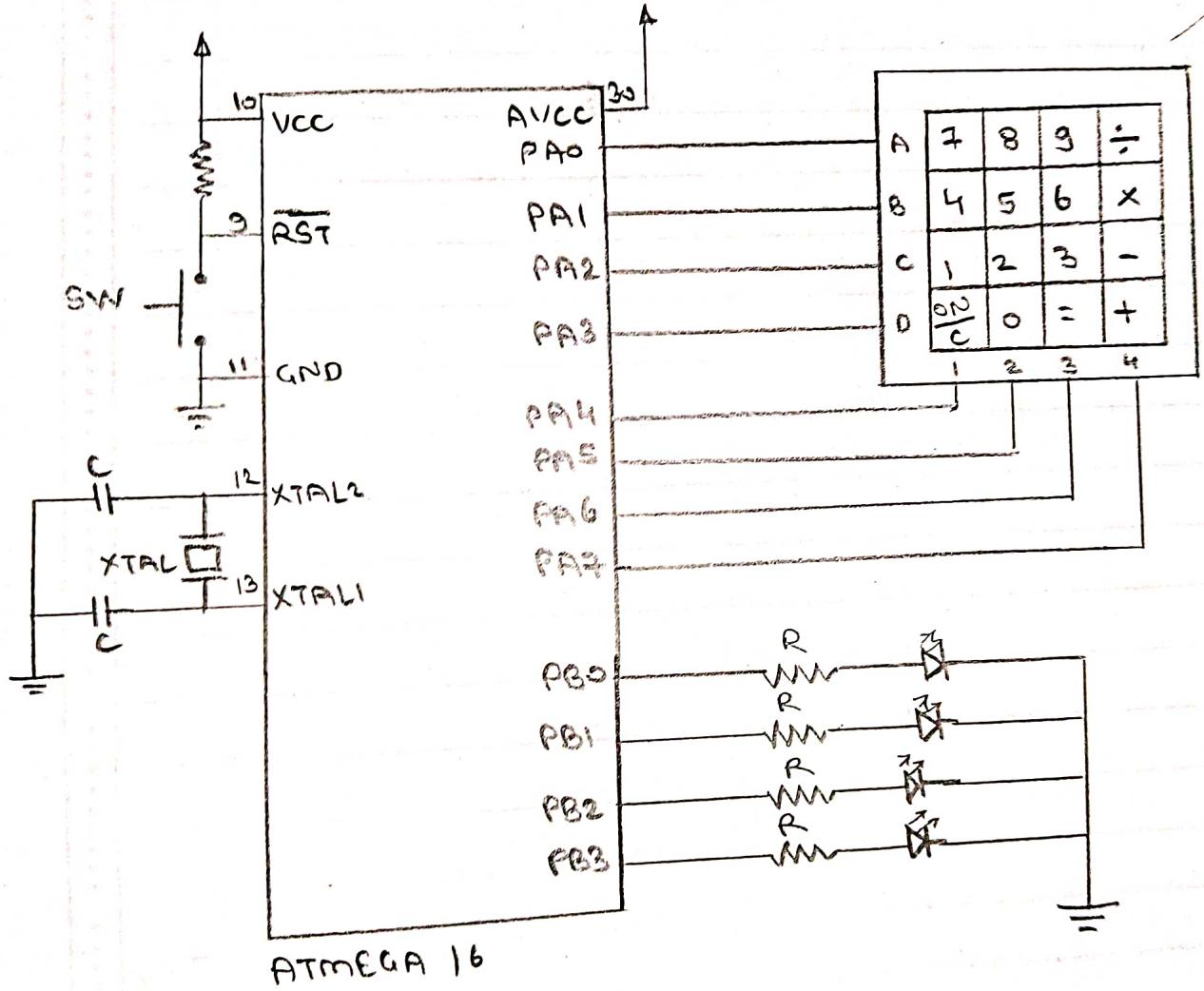
```
{  
    int i;  
    for (i=0; str[i] != 0; i++)  
    {  
        LCD_Char (str[i]);  
    }  
}
```

```
5 Void LCD_Clear ()
```

```
{  
    LCD_Command(0x01);  
    LCD_Command(0x80);  
}
```

```
6 int main()
```

```
{  
    LCD_Init();  
    LCD_String ("SSGmCE");  
    LCD_Command(0x0C);  
    LCD_String ("HELLO GUYS");  
    return 0;  
}
```



Expt. No.

Date :

```
#include <avr/io.h>
#include <util/delay.h>
#define KEY_PRT PORTA
#define KEY_DDR DDRB
#define KEY_PIN PINA
#define LED_PRT PORTB
#define LED_DDR DDRB
```

```
unsigned char keypad[4][4] = {{0x0F, 0x0E, 0x0D, 0x0C},
{0x0B, 0x0A, 0x09, 0x08},
{0x07, 0x06, 0x05, 0x04},
{0x03, 0x02, 0x01, 0x00}};
```

```
unsigned char colloc, rowloc;
```

```
char keyfind()
```

```
{
```

```
    while (1)
```

```
{
```

```
    KEY_DDR = 0xFF;
```

```
    KEY_PRT = 0x0F;
```

```
    do
```

```
{
```

```
        KEY_PRT &= 0x0F;
```

```
        asm (Nop);
```

```
        colloc = (KEY_PIN & 0x0F);
```

```
} while (colloc != 0x0F);
```

```
do
do
{
    -delay-ms(20);
    colloc = (KEY_PIN & 0x0F);
    }while (colloc == 0x0F);

    -delay-ms-(40);
    colloc = (KEY_PIN & 0x0F);
}while (colloc == 0x0F);

KEY_PRT = 0xEF;
asm ("NOP");
colloc = (KEY_PIN & 0x0F);
if (colloc != 0x0F)
{
    rowloc = 0;
    break;
}

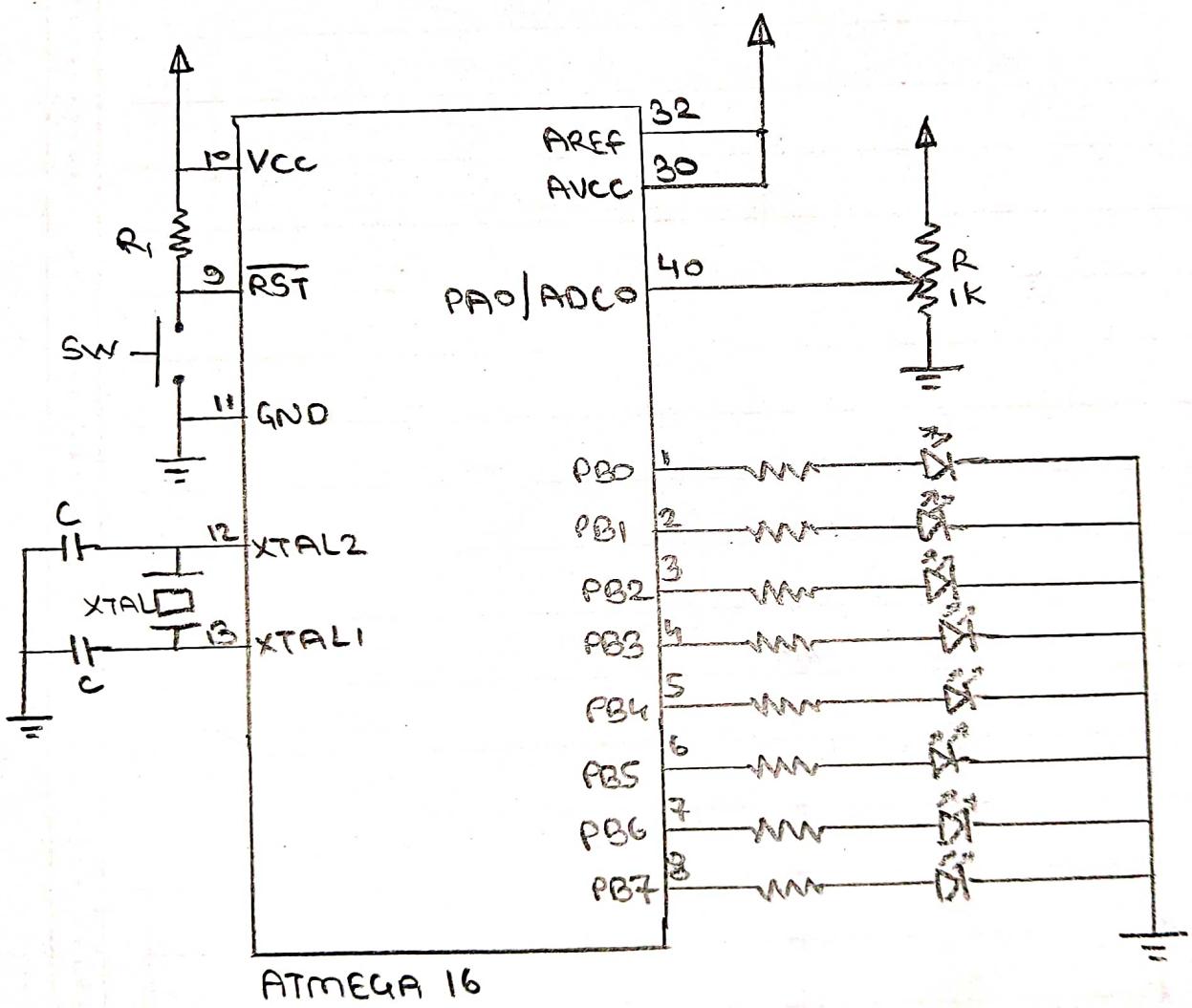
KEY_PRT = 0x0F;
asm("NOP");
colloc = (KEY_PIN & 0x0F);
if (colloc != 0x0F)
{
    rowloc = 1;
    break;
}

KEY_PRT = 0xBF;
asm("NOP");
colloc = (KEY_PIN & 0x0F);
if (colloc != 0x0F)
{
    rowloc = 2;
    break;
}
```

Expt. No.

Date :

```
KEY_PRT = 0x7F;  
asm ("NOP");  
colloc = (KEY_PTN & 0x0F);  
if (colloc) == 0x0F)  
{  
    rowloc = 3;  
    break;  
}  
if (colloc == 0x0E)  
    return (Keypad [rowloc][0]);  
else if (colloc == 0x0D)  
    return (Keypad [rowloc][1]);  
else if (colloc == 0x0B)  
    return (Keypad [rowloc][2]);  
else  
    return (Keypad [rowloc][3]);  
}  
int main (void)  
{  
    while ()  
    {  
        LED_PRT = Keyfind ();  
    }  
}
```



Expt. No.

Date :

```
#include<avr/io.h>
#include<util/delay.h>
int ADC_Read (char channel)
{
    int Ain, Ainlow;
    ADMUX = ADMUX | (channel & 0x0F);
    ADCSRA = (1<<ADSC);
    while ((ADCSRA & (1<<ADIF)) == 0);
    -delay - us(10);
    Ainlow = (int) ADCL;
    Ain = (int) ADCH * 256;
    Ain = Ain + Ainlow;
    return(Ain);
}
```

```
int main()
```

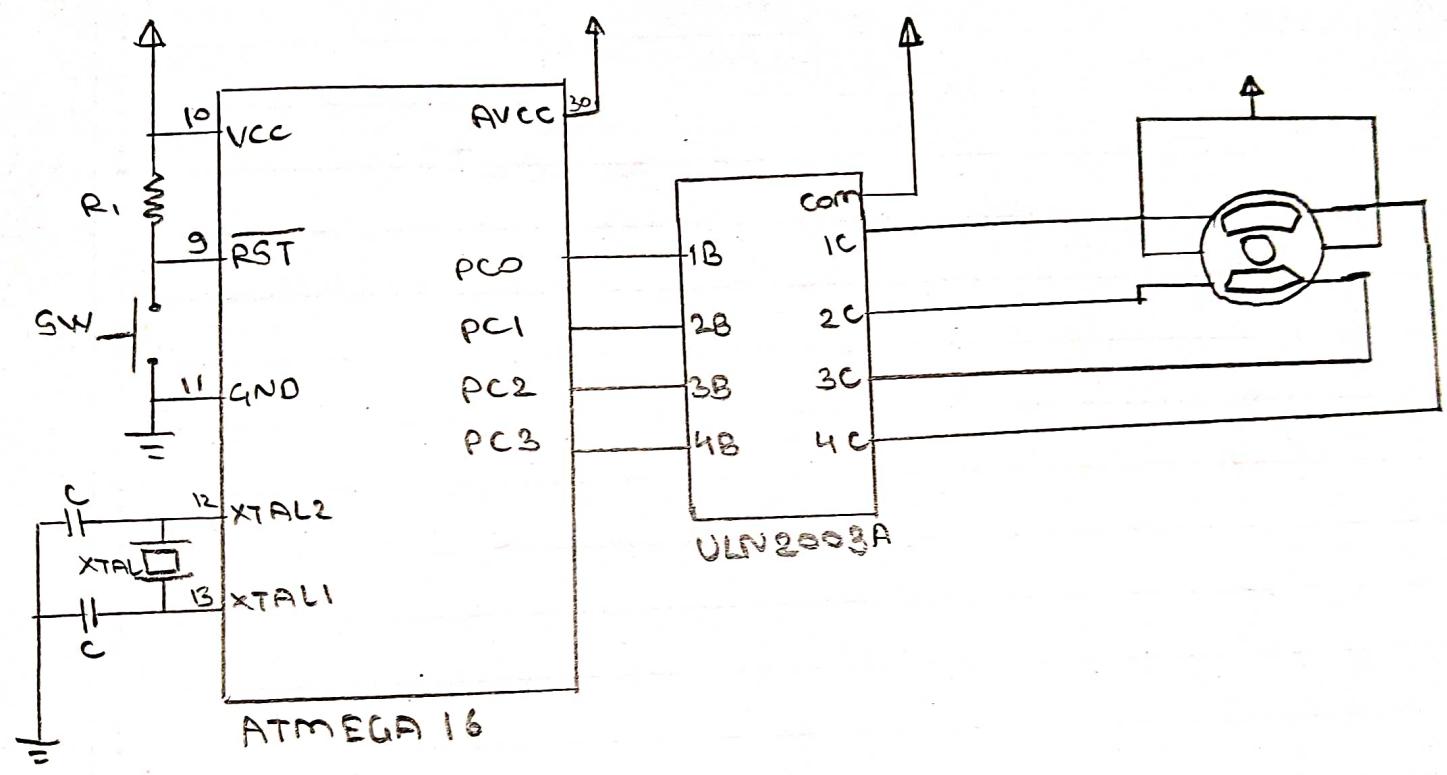
```
{
```

```
    int value;
    DDRB = 0xFF;
    PORTB = 0x00;
    DDRA = 0x00;
    ADCSRA = 0x87;
    ADMUX = 0X40;
    while () {
        value = ADC_Read (0);
        PORTB = value;
    }
```

```
}
```

```
return 0;
```

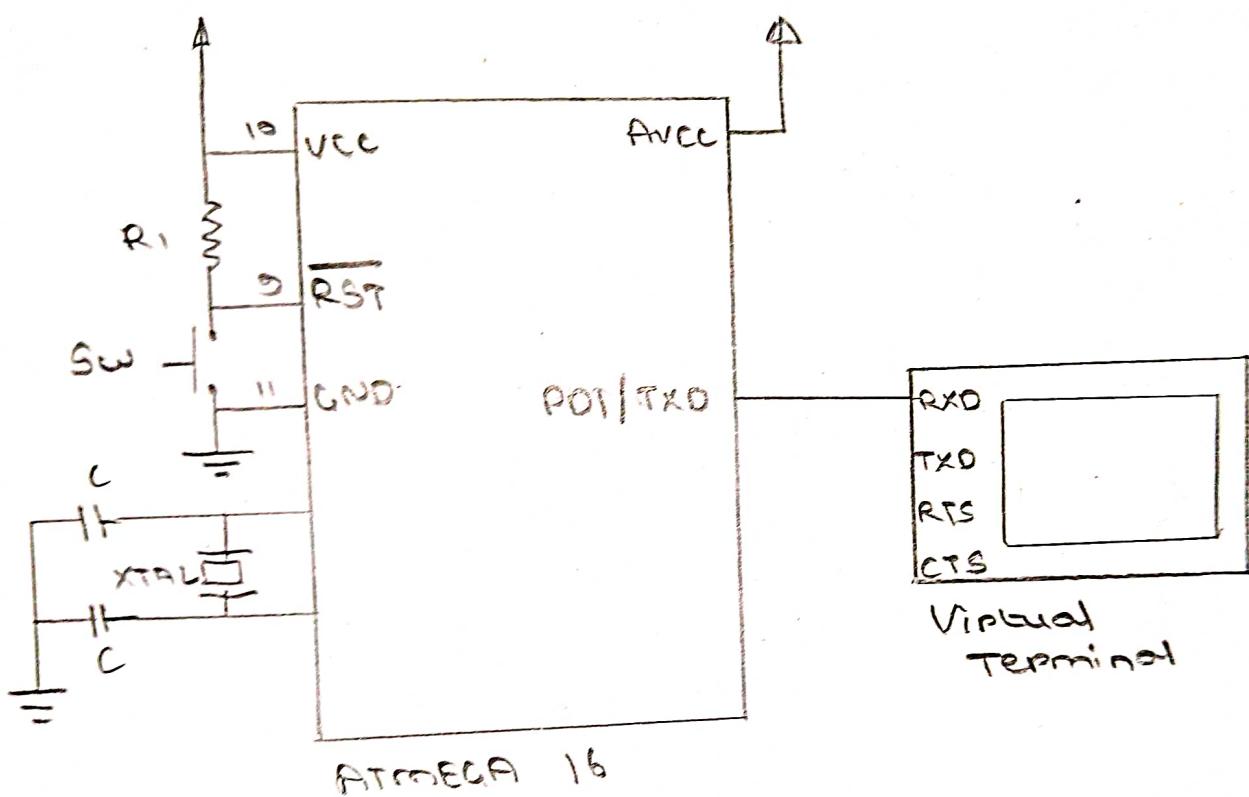
```
}
```



Expt. No.

Date :

```
#include<avr/io.h>
#include<util/delay.h>
int main(void)
{
    int PORTC = 0x0f;
    while(1)
    {
        for (int i=0 ; i<12 ; i++)
        {
            PORTC = 0x09;
            _delay_ms(10);
            PORTC = 0x08;
            _delay_ms(10);
            PORTC = 0x0c;
            _delay_ms(10);
            PORTC = 0x04;
            _delay_ms(10);
            PORTC = 0x06;
            _delay_ms(10);
            PORTC = 0x03;
            _delay_ms(10);
            PORTC = 0x01;
            _delay_ms(10);
        }
    }
}
```

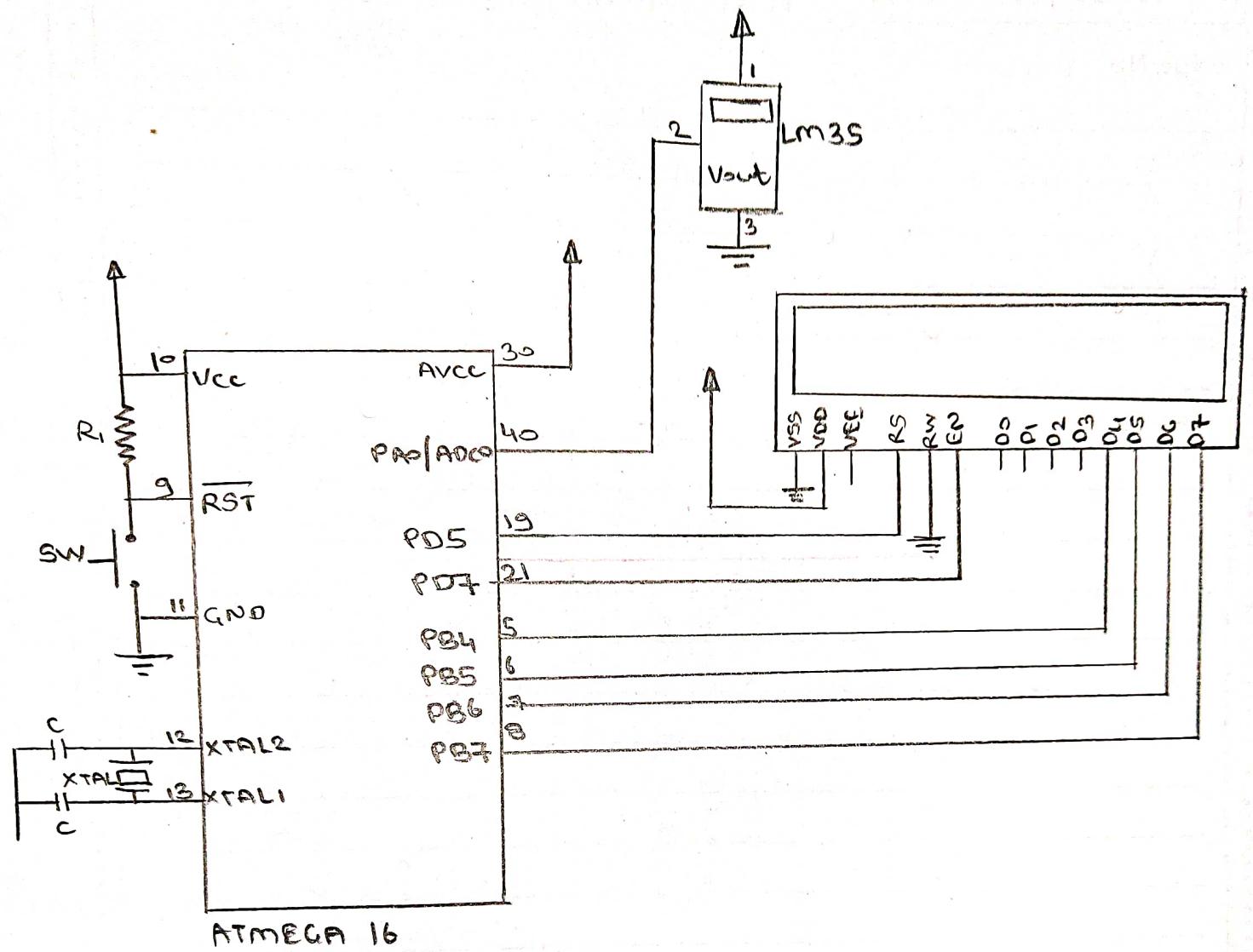


Expt. No.

Date :

```
#include <avr/io.h>
#include <util/delay.h>
void serial_out(char *str)
{
    UCSRB = (1 << TXEN);
    UCSRC = (1 << UCSZ1) | (1 << UCSZ0) | (1 << URSEL);
    UBRRH = 51;
    for (unsigned int i = 0; str[i] != 0; i++)
    {
        UDR = str[i];
        while (!(UCSRA & (1 << UDRE)));
        -delay_ms(500);
    }
}

void main(void)
{
    while (1)
    {
        serial_out("SSAMCE");
    }
}
```



Expt. No.

Date:

```
#define F_CPU 16000000
```

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

```
#include <util/delay.h>
```

```
#define LCD PORTB
```

```
#define EN 7
```

```
#define RS 5
```

```
#define RW 6
```

```
unsigned char data;
```

```
void lcdcmd(unsigned char cmd)
```

```
{
```

```
PORTD = ~(1<<RS);
```

```
PORTD = ~(1<<RW);
```

```
LCD = cmd & 0XF0;
```

```
PORTD = ~(1<<EN);
```

```
delay_ms(1);
```

```
PORTD = ~(1<<EN);
```

```
LCD = cmd <<4;
```

```
PORTD = ~(1<<EN);
```

```
-delay_ms(1);
```

```
PORTD = ~(1<<EN);
```

```
}
```

```
void lcddata(unsigned char data)
```

```
{
```

```
PORTD = ~(1<<RS);
```

```
PORTD = ~(1<<RW);
```

```

LCD = data & 0xFF;
PORTD |= (1 << EN);
delay_ms(1);
PORTD &= ~(1 << EN);

LCD = data << 4;
PORTD |= (1 << EN);
delay_ms(1);
PORTD &= ~(1 << EN);2
}

Void lcd_init()
{
    DDRB = 0xFF;
    DDRD = 0xFF;
    PORTD &= ~(1 << EN);

    lcdcmd(0x33);
    lcdcmd(0x32);
    lcdcmd(0x28);
    lcdcmd(0x0F);
    lcdcmd(0x01);
    -delay_ms(2);
}

void lcd_print(char *str)
{
    unsigned char i = 0;
    while (str[i] != 0)
    {
        lcddata(str[i]);
        i++;
    }
}

int main(void)
{
    lcd_init();
    lcd_print("Temperature:");
    DDRA &= ~(1 << 0);
    ADCSRA = 0x87;
    ADMUX = 0xE0;
    while (1)
}

```

Expt. No.

Date :

{

```
ADC0SR0 |= (1<<ADSC);
```

```
while((ADC0SR0 & (1<<ADLF)) == 0);
```

```
data = ADC0H;
```

```
convertndisplay (data);
```

```
-delay-ms (500);
```

{ }

```
void convertndisplay (unsigned char value)
```

{

```
unsigned char x, d1, d2;
```

```
x = value / 10;
```

```
d1 = x;
```

```
d2 = value % 10;
```

```
lcdcmd(0x80);
```

```
lcddata (d1 + 0x30);
```

```
lcddata (d2 + 0x30);
```

{ }