



**Annexure I**

## Annexure I

### Software Program of Capacitance Measurement

```
#include <stdio.h>
#include <AT89X52.H>
#include <intrins.h>

extern void delay1(void);           // Declaring assembly functions
extern void delay100(void);
extern void delay1000(void);

typedef unsigned char uc;

sbit scount = P1^6;                // Initialization of variables
sbit ecount = P1^7;
sbit range1 = P1^3;
sbit range2 = P1^4;
sbit range3 = P1^5;
sbit enablelcd = P1^2;
sbit readwrite = P1^1;
sbit regsel = P1^0;

void delay(void);                  // Declaring functions
void sdelay(void);
void display(void);
float calculate();
showmessage(char);
initlcd();
lcdcom(uc);
lcddata(uc);

int ovr_flow;                      // Variable Declaration
uc lcdstatus;
uc dispstr[17];

void timer_int() interrupt 1      // Timer Declaration
{
    ovr_flow++;
}

void main()                         // Starting of Main Program
{
    float count, capvalue;         // local variables declaration
    SCON = 0X50;
    TMOD = TMOD | 0X21;
    TH1 = 0XFD;
    TL1 = 0XFD;
    TR1 = 1;
    TH0 = 0;
    TLO = 0;
    ET0 = 1;
    EA = 1;
    TI = 1;
    ovr_flow = 0;
}
```

```

range1 = 0;
range2 = 1;
range3 = 1;

initlcd();
delay100();
printf("\nCapacitance Measurement\n\r");
sprintf(disptr,"%s"," WELCOME "); // Messages to display on LCD
showmessage(0x01);
sprintf(disptr,"%s"," DEPT OF INST ");
showmessage(0x02);
delay1000();

sprintf(disptr,"%s"," SK UNIVERSITY ");
showmessage(0x01);
sprintf(disptr,"%s"," ANANTAPUR ");
showmessage(0x02);
delay1000();

sprintf(disptr,"%s","CAPACITANCEMETER");
showmessage(0x01);
while(1)
{
sprintf(disptr,"%s","INSERT CAPACITOR");
showmessage(0x02);
delay100();
calculate();
if(count > 100)
{
capvalue = (count * 1.442)/10000;
sprintf(disptr,"VALUE:%2.5fuF",capvalue);
showmessage(0x02);
}

else
{
range1 = 1;
range2 = 0;
calculate();
sprintf(disptr,"VALUE:%2.5fnF",capvalue);
showmessage(0x02);
}

delay1000();
ovr_flow=0;
TH0 = 0;
TL0 = 0;
}}

initlcd() // function for initializing LCD
{
lcdcom(0x38);
delay();
lcdcom(0x38);
delay();
lcdcom(0x38);
}

```

```

delay();
lcdcom(0x80);
delay();
lcdcom(0x06);
delay();
lcdcom(0x0a);
delay();
lcdcom(0x0c);
delay();
}

```

```

showmessage ( char line )
{
    char s;
    lcdcom((line == 1)? 0x80:0xc0);
    for ( s = 0; s <16; s++ )
    {
        lcddata(' ');
    }
    lcdcom((line ==1)? 0x80:0xc0);
    for ( s = 0; s <16; s ++ )
    {
        if ( dispstr[s] == 0x00 )
            break;
        lcddata(dispstr[s]);
    }
}

```

```

lcddata( uc dat ) // function for sending data on to LCD
{
    regsel = 1;
    readwrite=0;
    P0 = dat;
    sdelay();
    enablelcd=1;
    sdelay();
    enablelcd = 0;
    sdelay();
}

```

```

lcdcom( uc command ) // function for sending command to LCD
{
    regsel = 0;
    readwrite=0;
    sdelay();
    sdelay();
    P0 = command;
    sdelay();
    sdelay();
    enablelcd=1;
    sdelay();
    enablelcd = 0;
    sdelay();
}

```

```

void delay(void)                                     // function for long delay
{
    int delaycnt;
    for ( delaycnt = 0; delaycnt <=20000; delaycnt ++);
}

void sdelay(void)                                   // function for short delay
{
    int countdelay;
    for ( countdelay = 0; countdelay <=2; countdelay ++);
}

float calculate()                                   // function for counting the pulses
{
    scount = 1;
    scount = 0;
    TR0 = 1;
    while(ecount ==0);
    TR0 = 0;
    count = TL0 + TH0 * 256 + ovr_flow*65536;
}

NAME delay                                         // assembly program for delay functions
    VAR1 DATA 70H
    VAR2 DATA 71H
    VAR3 DATA 72H
    VAR4 DATA 73H
    VAR5 DATA 74H
    VAR6 DATA 75H
?PR?delay?delay SEGMENT CODE
    PUBLIC delay1000
    PUBLIC delay100
    PUBLIC delay50
    PUBLIC delay10
    PUBLIC delay1
    RSEG ?PR?delay?delay
    USING 1
delay1000: MOV VAR6,#4H
D1000:     CALL delay100
          DJNZ VAR6,D1000
          RET
delay100:  MOV VAR1,#14H
D100:     CALL delay50
          DJNZ VAR1,D100
          RET
delay50:  MOV VAR2,#80H
D50:     CALL delay10
          DJNZ VAR2,D50
          RET
delay10:  MOV VAR3,#80D
D10:     CALL delay1
          DJNZ VAR3,D10
          RET
delay1:   MOV VAR4,#1D
D1:     DJNZ VAR4,D1
          RET
          END

```