CONSTRUCTION

Microcontroller Based Inductance Capacitance Meter

MUDIT AGARWAL

This is the Inductance / Capacitance Meters circuit. One can easily build this LC Meter measure inductances starting from 1mH to 100mH, 1μH to 1000μH, 10nH to 1000nH and capacitance from 0.1pF to 0.9μF. This inductance capacitance meter has Zero out switch that will reset the initial inductance capacitance, making sure that the final readings of the LC Meter are as accurate as possible. Further this Inductance Capacitance Meter circuit uses an auto ranging system, to come over the headache to select ranges manually. The resonance frequency of LC can be determined by using the frequency formula given below.

$$f_r=1/(2\pi\sqrt{LC})$$

Note that there are three variables that we can work with; fr, L and C (fr represents a frequency, L inductance and C capacitance). If we know the values of the two variables we may calculate the value of the third variable. For example if we want to determine the value of an unknown inductor with X inductance. We plug X inductance into the formula and we also use value of a known capacitor. Using this data we can calculate the frequency. Once we know the frequency we can use the power of the algebra and rewrite the above formula to solve for L (inductance). This time we will use the calculated frequency and a value of a known capacitor to calculate the inductance. We just calculated the value of unknown inductor, and we may use the same technique to solve for the unknown capacitance and even frequency.

The LC Meter uses a LM311 IC that functions as a frequency generator and this is exactly what we need. If we want to calculate the value of an unknown inductor we use a known1000pF capacitor and the value of an unknown inductor. LM311 will generate a frequency that we can measure with a frequency meter. Once we have this information we can use the frequency formula to calculate the inductance. The same thing can be done for calculating the value of a unknown capacitor. This time we don't know the value a capacitor so instead we use the value of a known inductor to calculate the frequency. Once we have that information we apply the formula to determine the capacitance.

All this sounds great, however if we want to determine the value of a lot of inductors / capacitors then this may become a very time consuming process. This circuit uses PIC16F84A microcontroller from microchip. PIC16F84A is like a small computer that can execute HEX programs that are written using an assembly language. PIC16F84A is a very flexible microcontroller. PIC16F84A IC requires very minimal number of external components like 4MHz crystal / resonator and few resistors depending on what project we are building. Before we can use PIC16F84A microchip we have to program it with a HEX code which has to be sent from the computer. In the next step we use the frequency generated by LM311 IC and pass it

Pins	Symbol	Function
1	Vee	Ground
2	Vdd	+5V
3	Vo	Contrast
4	RS	Register Select
5	RW	Read Write
6	En	Enable Signal
7	D0	Data Bit 0
8	D1	Data Bit 1
9	D2	Data Bit 2
10	D3	Data Bit 3
11	D4	Data Bit 4
12	D5	Data Bit 5
13	D6	Data Bit 6
14	D7	Data Bit 7
15	VA	Backlight +5V
16	VK	Backlight GND

Table 1.

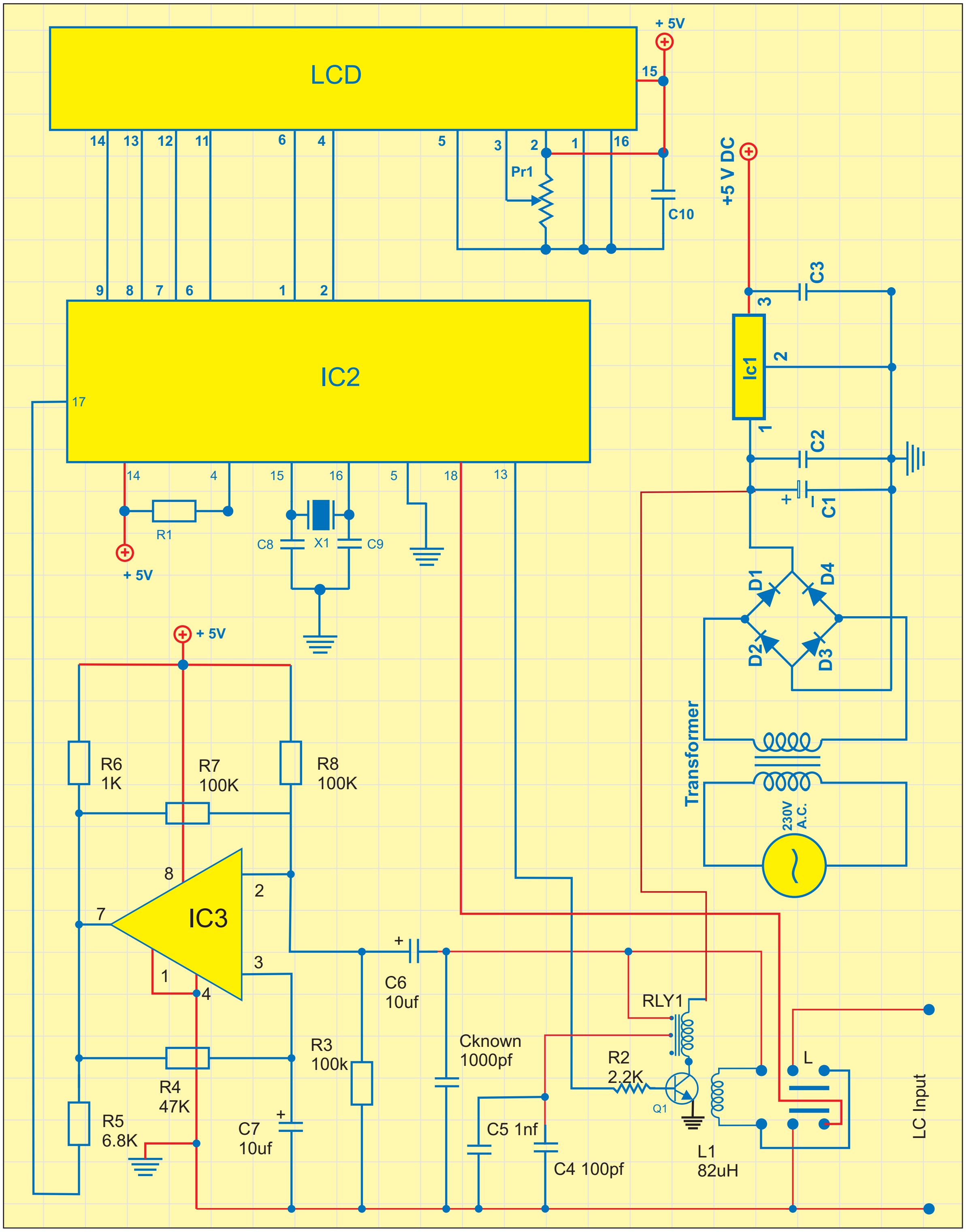

Fig.1 : Circuit Diagram of Microcontroller Based Inductance Capacitance Meter

on to PIC16F84A's PIN 17. We designate this pin as an input, as well as all other pins that are directly connected to switches and jumpers. User can use these inputs to tell the microchip to execute specified set of instructions or perform calculations. Once the microchip will calculate the unknown inductance or capacitance it will use PINs that are designated as outputs and pass the results on to the

16 character LCD display.

Most of the character LCD displays have 14 or 16 PINs. LCD with 14 pins donot have backlight. The LCD pins function is shown in table 1.

Software

```
list          p=pic16f84a
 include        p16f84a.inc
  __config  _hs_osc  &  _wdt_off  &
_pwrte_on & _cp_off
c10m    equ     h'0c'
c01m    equ     h'0d'
 c10s    equ    h'0e'
c01s    equ     h'0f'
lcd7_0  equ     b'00110011'
lcd7_1  equ     b'00110010'
lcd7_2  equ     b'00111000'
 lcd7_3  equ    b'00001110'
lcd7_4  equ     b'00000110'
lcd7_5  equ     b'00001100'
lcd7_6  equ     b'00100111'
lcd7_7  equ     b'00000001'
lcd7_8  equ     b'00100111'
lcd7_9  equ     b'10000000'
lcd70   equ     h'10'
 lcd71   equ    h'11'
lcd72   equ     h'12'
 lcd73   equ    h'13'
 lcd74   equ    h'14'
lcd75   equ     h'15'
lcd76   equ     h'16'
lcd77   equ     h'17'
lcd78   equ     h'18'
```

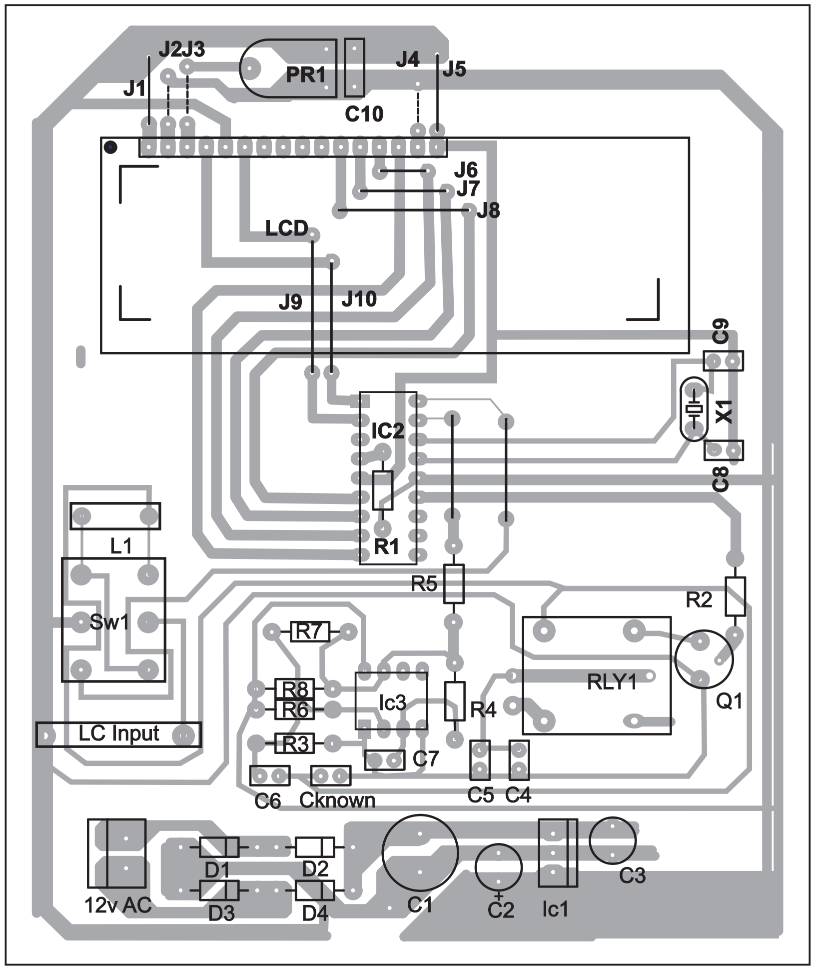

Fig.2: Component Layout o Microcontroller Based Inductance Capacitance Meter.

COMPONENT LIST

SEMICONDUCTOR DEVICES

IC1	7805
IC2	PIC16f84A
Ic3	LM311
D1-D4	1N4007

CAPACITORS

C1	1000uf/25V
C2,C3,C10	0.1uf
C8,C9	22pf
C4	100pf
C5	1nf
C6,C7	10uf
Cknown	1000pf

RESISTORS

R1	10K
R2	2.2K
R3	100K
R4	47K
R5	6.8K
R6	1K
R7,R8	100K
Pr1	2K

MISCELLANEOUS

Transformer	12V
X1	4 MHZ
LCD	16X 2 Liquid Crystal Display
Sw1	Double Pole Double Way Switch

```
 lcd79   equ     h'19'
tm_cnt  equ     h'1a'
time_f  equu    h'1b'
 ttl_in  equ    h'1c'
w_save  equ     h'1d'
s_save  equ     h'1e'
cnt500u equ     h'1f'
 cnt1m   equ    h'20'
ra0     equ     0
ra1     equ     1
ra2     equ     2
 ra3     equ     3
ra4     equ     4
rb6     equ     6
 org    0
goto    init
 org    4
goto    int
 org    5
init
bsf     status,rp0
movlw   b'00010000'
movwf   trisa
movlw   b'00000111'
movwf   option_reg
bcf     status,rp0
```

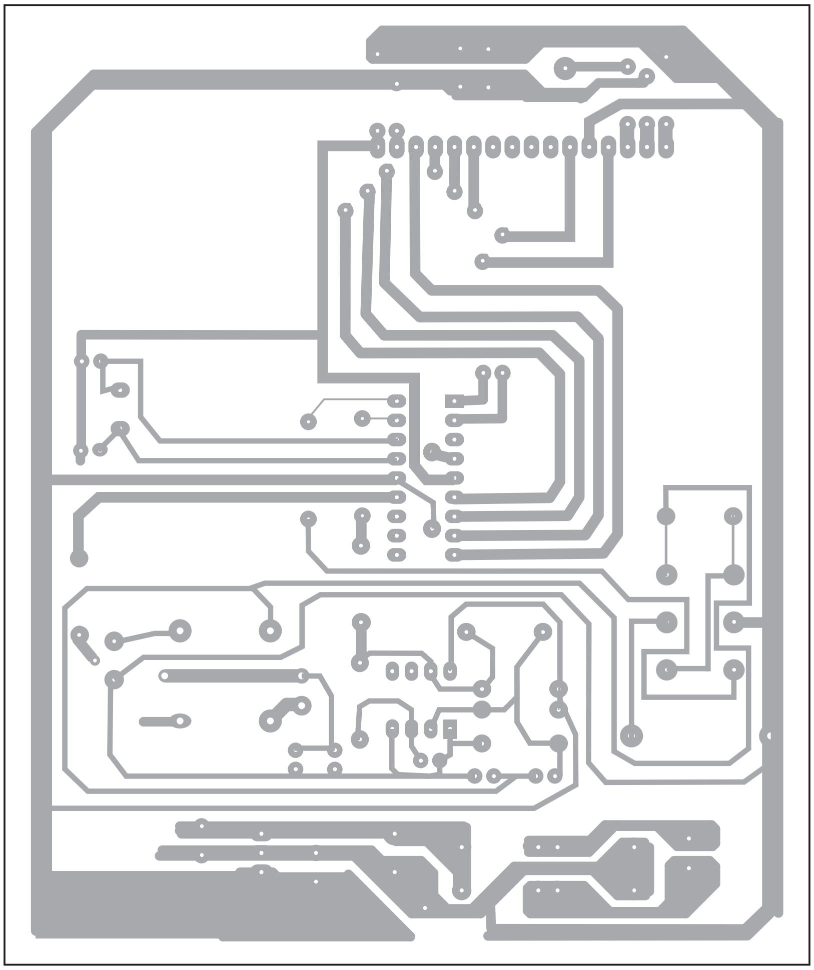
Fig.3 : PCB Layout of Microcontroller Based Inductance Capacitance Meter.

```
bsf     status,rp0
movlw   h'ff'
movwf   trisb
bcf     status,rp0
bcf     porta,ra0
bcf     porta,ra1
bcf     porta,ra2
#ifdef_debug
movlw   h'fe'
#else
call    t1m
movf    portb,w
#endif
xorlw   h'ff'
andlw   h'0f'
movwf   c10m
bsf     porta,ra0
#ifdef_debug
movlw   h'ff'
#else
call    t1m
movf    portb,w
#endif
xorlw   h'ff'
andlw   h'0f'
movwf   c01m
call    led_cont
movf    c10m,w
btfss   status,z
goto    sw_check
movf    c01m,w
btfsc   status,z
goto    stand_by
sw_check:
bsf     status,rp0
bsf     trisb,rb6
bcf     status,rp0
movlw   b'00001000'
movwf   porta
movlw   lcd7_0
movwf   lcd70
movlw   lcd7_1
movwf   lcd71
movlw   lcd7_2
movwf   lcd72
movlw   lcd7_3
movwf   lcd73
movlw   lcd7_4
movwf   lcd74
movlw   lcd7_5
movwf   lcd75
movlw   lcd7_6
movwf   lcd76
movlw   lcd7_7
movwf   lcd77
movlw   lcd7_8
movwf   lcd78
movlw   lcd7_9
movwf   lcd79
clrf    c10s
clrf    c01s
bsf     time_f,0
stand_by:
bcf     porta,ra0
bsf     porta,ra1
bcf     porta,ra2
#ifndef_debug
call    t1m
btfsc   portb,rb6
goto    stand_by
#endif
Start:
bcf     porta,ra3
#ifdef_debug
        movlw   d'255'
#else
        movlw   d'43'
#endif
        movwf   tmr0
#ifdef_debug
        movlw   d'2'
#else
        movlw   d'46'
#endif
        movwf   tm_cnt
movlw   h'a0'
movwf   intcon
loop:
        call    led_cont
```

```
movf    time_f,w
btfsc   status,z
goto    time_out
#ifndef _debug
        btfsc   porta,ra4
#endif
        goto    loop
time_out
        clrf    intcon
goto    init
bsf     status,rp0
clrf    trisb
bcf     status,rp0
bcf     porta,ra0
bcf     porta,ra1
bsf     porta,ra2
movf    c10m,w
movwf   ttl_in
call    ttl_7lcd
bsf     porta,ra0
Movf    c01m,w
movwf   ttl_in
call    ttl_7lcd
bcf     porta,ra0
bsf     porta,ra1
movf    c10s,w
movwf   ttl_in
call    ttl_7lcd
bsf     porta,ra0
movf    c01s,w
movwf   ttl_in
call    ttl_7lcd
return
ttl_7lcd
movlw   lcd70
addwf   ttl_in,w
movwf   fsr
movf    indf,w
 movwf  portb
 #ifndef _debug
 call   t1m
 #endif
 return
t1m     movlw   2
movwf   cnt1m
 tm1lp1  movlw   d'249'
movwf   cnt500u
tm1lp2  nop
nop
decfsz  cnt500u,f
goto    tm1lp2
decfsz  cnt1m,f
goto    tm1lp1
 return
 int:
goto    tm1lp2
decfsz  cnt1m,f
 goto   tm1lp1
return
int:
movwf   w_save
movf    status,w
movwf   s_save
bcf  status,rp0
btfsc   intcon,t0if
goto    timer_int       int_end
movf    s_save,w
movwf   status
swapf   w_save,f
swapf   w_save,w
retfie  timer_int:
 bcf    intcon,t0if
#ifdef _debug
movlw   d'255'          #else
 movlw  d'43'           #endif
movwf   tmr0
decfsz  tm_cnt,f
goto    int_end         #ifdef _debug
movlw   d'2'            #else
movlw   d'46'           #endif
movwf   tm_cnt
decfsz  c01s,w
goto    countdown
movf    c10s,w
btfss   status,z
goto    countdown
movf    c01m,w
btfss   status,z
goto    countdown
movf    c10m,w
btfss   status,z
 goto   countdown
time_out1:
clrf    time_f
goto    int_end
movf    c01s,w
btfss   status,z
goto    cd_c01s
movlw   9
movwf   c01s
movf    c10s,w
btfss   status,z
goto    cd_c10s
movlw   5
movwf   c10s
movf    c01m,w
btfss   status,z
goto    cd_c01m
movlw   9
movwf   c01m
movf    c10m,w
btfss   status,z
goto    cd_c10m
goto    time_out1
cd_c01s:
decf    c01s,f
goto    int_end
cd_c10s:
decf    c10s,f
goto    int_end
cd_c01m:
decf    c01m,f
goto    int_end
cd_c10m:
decf    c10m,f
goto    int_end         End
```