An LC meter in C

By Juan H la Grange, ZS6SZ

1. Introduction

This article and project is based on “Digital LC Meter Version 2” by Phil Rice VK3BHR [https://sites.google.com/site/vk3bhr/home/index2-html] and it predecessors. It is for use by private enthusiasts and hobbyists.

An inductance meter has been sorely needed by this author for his projects. When he read the above article, he wanted to implement it with a PIC 16F1619 chip using a Curiosity board, because that is what he had at hand. Unfortunately, that meant a big change, because the original software is written in assembler and he does not know PIC assembler. Instead of either learning assembler or guessing if it would work or what needs to change (port and pin assignments are different for example), he decided to rewrite it in C, using the Microchip MPLABX-IDE development tool.

The PIC16F1619 has an 8k program memory so program size is not an issue. This also means that 32-bit floating point can be used, for improved precision. Just below 4k words are used.

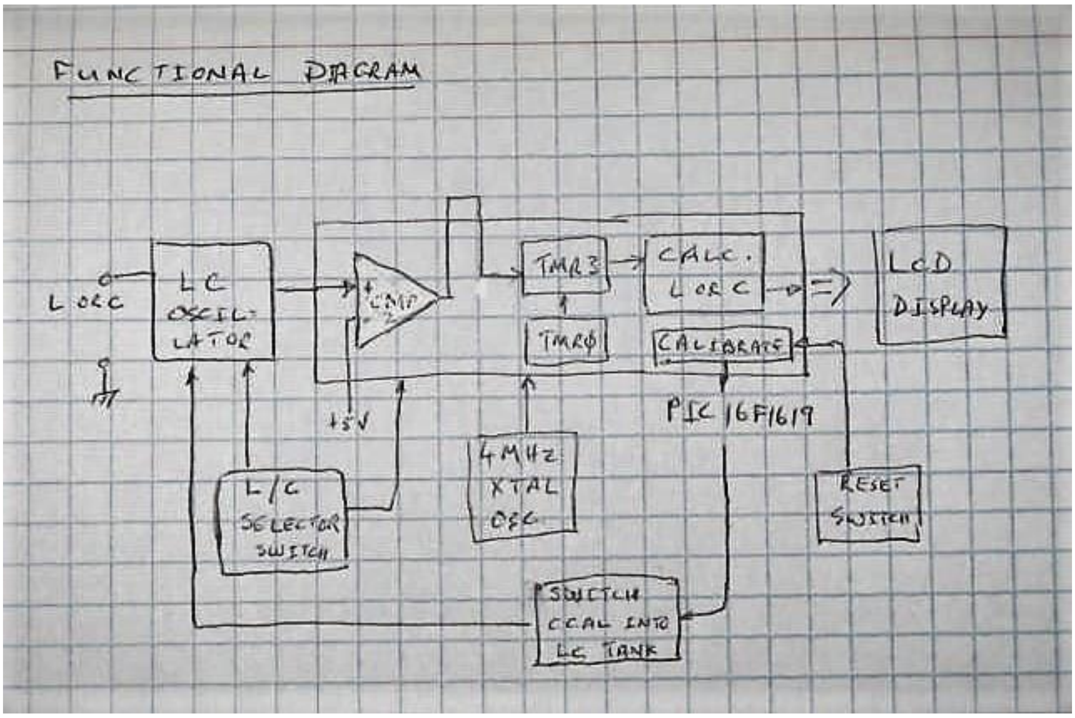
He could not get the comparator-based oscillator to work, maybe because of the selected chip. So, the decision was to use a different oscillator. Many L-C oscillators have tapped capacitors or inductors, or transformation, which would not work here, since the application requires a simple parallel L-C tank circuit. Thus, the decision was to use an FET Franklin two-stage oscillator, with the second stage ensuring the 360° phase shift required for oscillation.

The measurement ranges are:

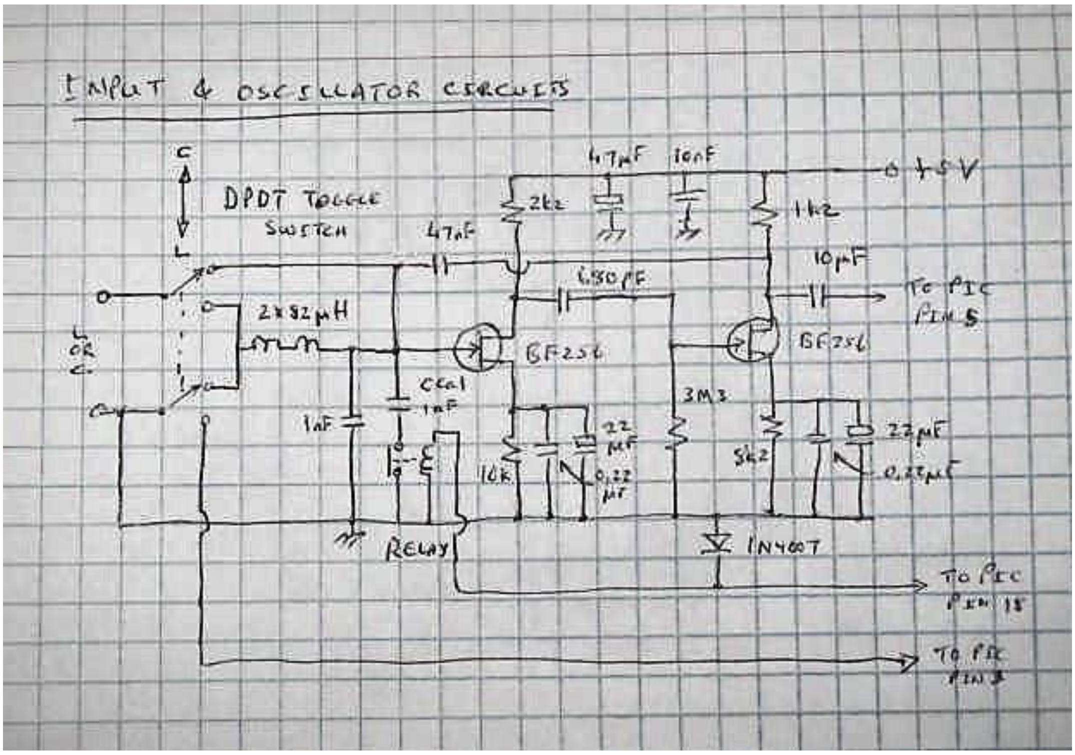
- Capacitance: 1 pF – 100 nF
- Inductance: 100 nH - >20mH

2. Circuit diagram

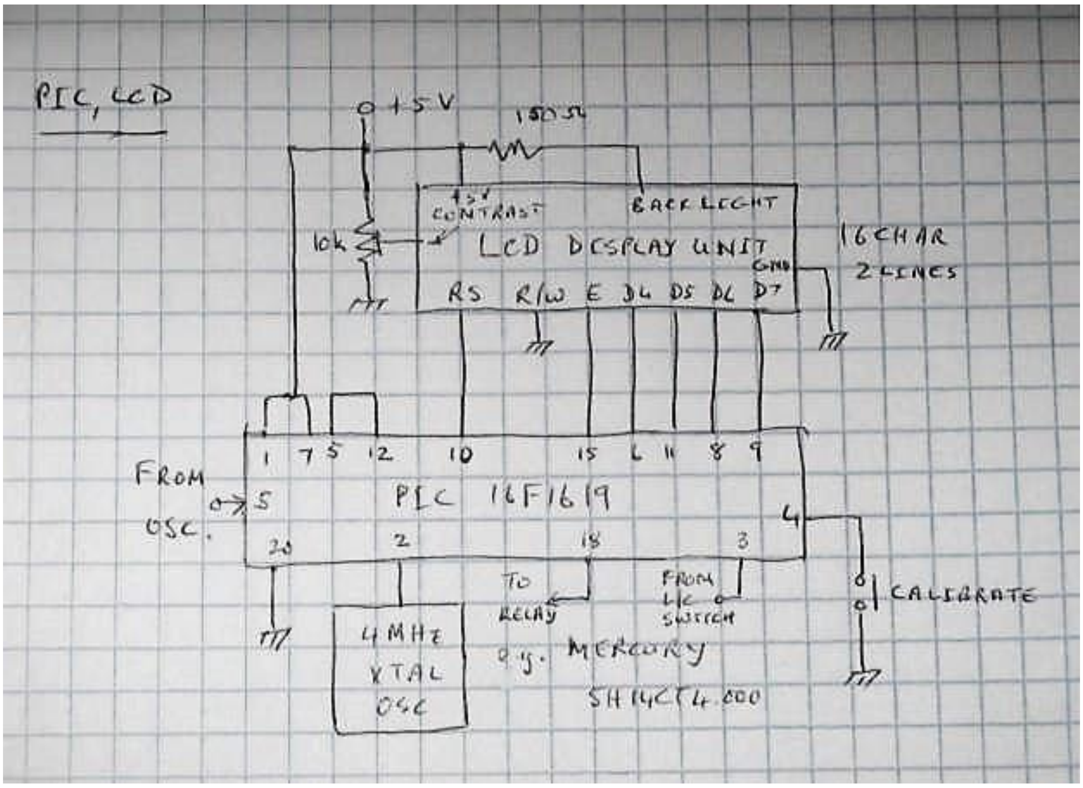
The overall functional diagram is:



The oscillator and input circuits look as follows:



The details of the PIC and LCD circuits are as follows:



Different from the original circuit, an inductance of 164 micro Henry is used in the oscillator, which did not want to oscillate otherwise with the larger capacitance values. The frequency ranges from about 25 kHz (large external L or C) to about 350 kHz (no external L or C).

Comparator 2 of the PIC chip is used to interface the analogue oscillator with the digital device.

A 4.5 V 16mA miniature relay is used to switch CCal in and out. In general, components at hand were used. CCal and the other 1nF cap in the LC tank circuit should not be ceramic for stability reasons.

The project was constructed on copper strip board. A regulated external 5V power supply is used.

3. Program

The XC8 compiler defaults to 24-bit floating point, so the option ”—Float=32” had to be included.

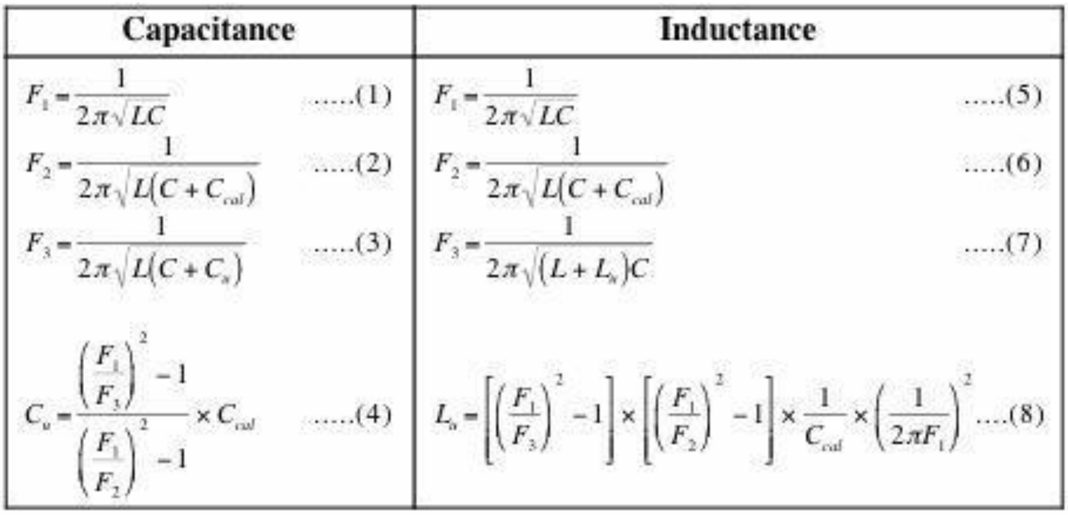
The standard compiler routines for floating point multiplication and division were utilised.

The frequency measurement is based on the design by John Main [www.best-microcontroller-projects.com/pic-frequency-counter.html] where Timer3 of the PIC counts the number of incoming pulses over the period determined by Timer0. The period was selected to be 250 ms, as a compromise between accuracy and speed of update for the display. This seems to work well. When the actual frequency is required, such as in the inductance calculations, the measured value is multiplied by four.

The LCD routines come from Craig Lee [www.piclist.com/techref/microchip/language/**c**/io/lcd/hitach**i**lcd-cl/].

The CPU clocks at 16 MHz driven by the 4 MHz crystal oscillator.

After initialization, the program runs in an infinite loop, checking if the TMR0 interrupt routine has calculated a new frequency (which it does every 250 ms). Then the program calculates the values as per the formulas (4) or (8) below, scales and writes them to the display unit.



Capacitance	Inductance
$F_1 = \frac{1}{2\pi\sqrt{LC}}$ .....(1)	$F_1 = \frac{1}{2\pi\sqrt{LC}}$ .....(5)
$F_2 = \frac{1}{2\pi\sqrt{L(C + C_{cal})}}$ .....(2)	$F_2 = \frac{1}{2\pi\sqrt{L(C + C_{cal})}}$ .....(6)
$F_3 = \frac{1}{2\pi\sqrt{L(C + C_x)}}$ .....(3)	$F_3 = \frac{1}{2\pi\sqrt{(L + L_x)C}}$ .....(7)
$C_x = \frac{\left(\frac{F_1}{F_3}\right)^2 - 1}{\left(\frac{F_1}{F_2}\right)^2 - 1} \times C_{cal}$ .....(4)	$L_x = \left[\left(\frac{F_1}{F_3}\right)^2 - 1\right] \times \left[\left(\frac{F_1}{F_2}\right)^2 - 1\right] \times \frac{1}{C_{cal}} \times \left(\frac{1}{2\pi F_1}\right)^2$ ....(8)

Upon start-up or reset, the program measures the frequency of the oscillator (F1); switches in CCal, measures the frequency (F2) and switches out CCal again. It then does a few basic calculations before entering the infinite loop. Connecting an unknown L or C of course changes the frequency to F3.

There is no provision for on-line calibration adjustment. Since the accuracy of CCal (and the crystal oscillator) determines the accuracy of the result., the CCal value in the C-code should be adjusted if required and the program recompiled and reloaded.

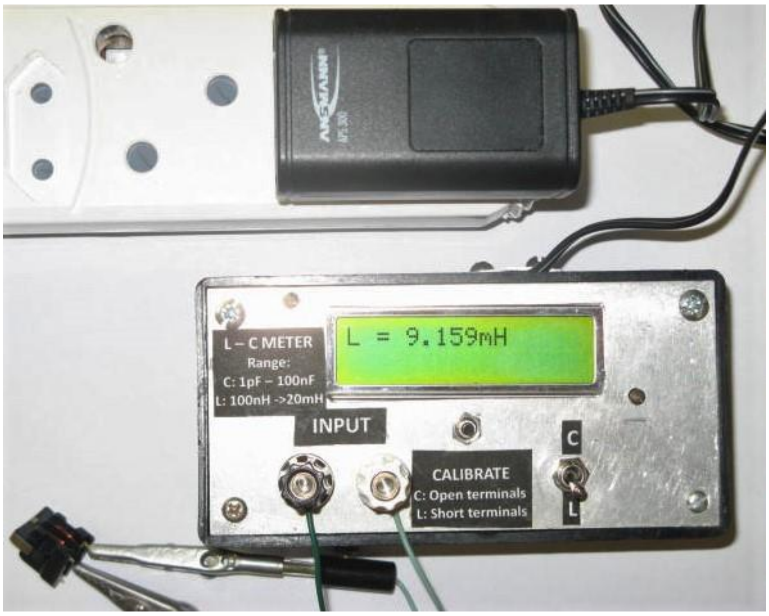
As before, the Calibrate pushbutton resets the chip.

The software files are attached.

4. Conclusion

The meter works well in practice and has already been very useful.

For this author, the concepts behind this LC meter are brilliant and the originator(s) needs to be congratulated. This project strikes the right chord with him!


ANSMANN
APS 300
L = 9.159mH
L – C METER
Range:
C: 1pF – 100nF
L: 100nH ->20mH
INPUT
CALIBRATE
C: Open terminals
L: Short terminals
C
L

May 2018