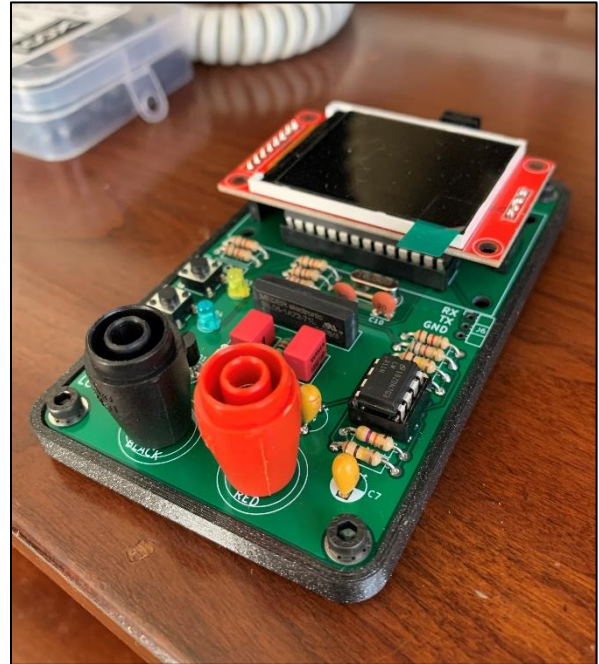


Build your own LC Meter

Part 1: Builder's Guide

Bruce E. Hall, [W8BH](#)

An LC meter is a handy tool for any electronics hobbyist. There are ready-built devices which you can purchase, but what's the fun in that? You can make your own meter for less money and with better accuracy. Read on if you are interested in building your own LC meter based on the famous AADE design.



Introduction

Several DIY meters are available on the web. One of the most accurate and widely acclaimed was designed and sold by Neil Hecht and his company AADE (Almost All Digital Electronics). Neil passed away in 2015 but his design lives on.

An AADE-inspired design was recently implemented by “coreWeaver”, using an ATmega328 for the MCU and a Nokia 5110 LCD for the display. The documentation for his project, including links to his videos, is found on his GitHub page at: <https://github.com/coreWeaver/LC-Meter>.

I decided to design my own. I wanted a meter that would be easy to build, using through-hole components, and easy to program using the Arduino IDE. This document describes, step-by-step, how to build your own meter. I believe in starting small and testing as I go.

Not a Kit.

Forgive me for starting these notes with a disclaimer: this is not a kit, I am not selling anything, and there is no guarantee of success or suitability for any particular purpose. Still interested?

Project Files

[Part 1 \(this document\)](#)
[Part 2: User's Guide](#)
[Part 3: Software Review](#)
[Part 4: Enclosures](#)
[Schematic](#)
[Interactive BOM](#)
[Source Code](#)
[PCB Gerbers](#)
[Enclosure STL files](#)

Bill of Materials.

Qty	Reference(s)	Value	Description
1	BZ1	Piezo Buzzer	TDK PS1240P02BT
1	C1	1000pF	WIMA Film, 2.5%
1	C2	1000pF	WIMA Film, 1%
3	C3, C6, C7	10uF	Electrolytic, 16V
1	C4	10uF	Tantalum, 16V
1	C5	0pF	Ceramic, 0.1" or 0.2", not fitted
2	C9, C10	22pF	Ceramic disc, 0.1"
2	C11, C12	100nF	MLCC, 0.1"
1	D1	BAT42	Schottky diode
1	D2	1N4148	small signal diode
2	D3, D4	3mm LED	D3 yellow, D4 blue
4	H1, H2, H3, H4	Mounting Hole	M3 hardware
1	J1	Red Probe	CalTech CT3151V1-2
1	J2	Black Probe	CalTech CT3151V1-0
1	J3	DC Barrel Jack	CUI PJ-102AH
1	J4	9V Battery Header	not fitted
1	J5	ICSP Header	2x3 male, 0.1"
1	J6	UART Header	1x3 male, 0.1", not fitted
2	J7, J8	Supports A & B	1x4 male, 0.1", not fitted
1	K1	DPDT Relay	Kemet UA2-5NE or equiv.
1	K2	SPST Reed Relay	Medex SIL05-1A72-71L or equiv.
1	L1	82uH	Fixed Inductor, 5%
1	Q1	2N3904	TO-92 NPN
8	R1, R9-R11, R14-R17	1K	0.25W, 5%
1	R2	120R	0.25W, 5%
3	R3, R4, R13	10K	0.25W, 5%
3	R5, R6, R7	100K	0.25W, 5%
1	R8	47K	0.25W, 5%
1	R12	6.8K	0.25W, 5%
1	R18	68R	0.25W, 5%
2	S1, S2	SPST Switch	6mm TACTILE
1	SW1	SPDT Slide Switch	OS102011MA1QN1
1	U1	L4931CZ50-AP	IC REG LINEAR 5V 250MA TO92-3
1	U2	LM311	IC DIP-8
1	U4	ATmega328P_DIP28	IC 8-bit MCU
1	U5	TFT Display	MSP1803, 128x160 pixel
1	Y1	16MHz crystal	HC49-U

Additional miscellaneous components: AVR external programmer, LC Meter PCB, 9VDC Adapter, 8pin DIP sockets for the LM311 IC, 28-pin DIP socket for the ATmega328, 4-pin and 8-pin male headers for the display module, 4 pin and 8-pin female headers for the display socket.

The PCB.

To order a circuit board from a manufacturer you must provide a digital set of design files, called ‘[Gerbers](#)’. You can obtain the Gerbers on my [GitHub account](#).

Almost all board fabricator websites allow Gerber-file uploads. Choose the manufacturer you like. I’ve used [OSH Park](#) (USA, excellent quality) in the past and have been very satisfied. For this project I used [JLCPCB](#) (China, prototype quality). Choose the default 2-layer options, get your quote, and choose whatever shipping option you want. At JLCPCB I paid \$2.00 + shipping for a set of 5 boards.

All components are mounted on the front face of the PCB.

Refer to the [interactive BOM](#) to identify the location for each part.

Step 0: Preparation.

Obtain the PCB and parts listed above. In addition, you will need a decent soldering iron, solder, an external programmer for your microcontroller (such as an AVR-ISP or USBtinyISP), a digital multimeter, and a 6V-10V DC power supply with barrel connector (2.1mm center pin positive). You should be familiar with soldering and have successfully kitted other projects using ICs and other through-hole components. You should have sufficient lighting, magnification, and workspace area.

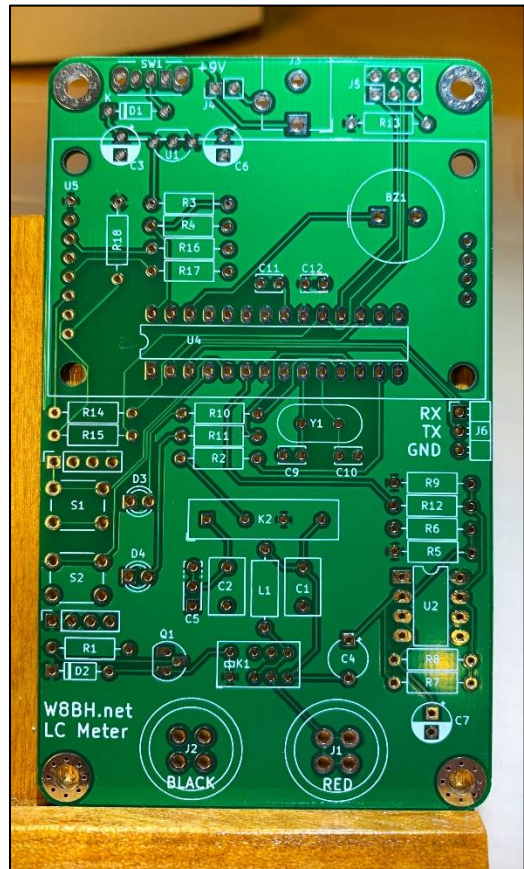
This builder’s guide may change from time to time. Refer to the most recent version at: http://w8bh.net/LCmeter_BuildersGuide.pdf.

As I mentioned earlier, I believe in starting small. Build the smallest something that you can and test it. Don’t forge ahead until you are sure everything is working. We will build and test, step by step.

These instructions assume that you are comfortable using the Arduino IDE, and that you will program the microcontroller using the onboard 6-pin ICSP header. However, you may program the MCU in any manner that that suits you.

Take a moment to familiarize yourself with the PCB. In general, the build will start near the top of the board and work downward. The board has 5 main sections: the power supply, the MCU, the display, the control circuit, and the oscillator. We will tackle one section at a time and test each section with code and/or multimeter before moving on.

Note: When soldering components, I recommend soldering the first pin and checking component alignment on the front face of the PCB before soldering the remaining pin(s).

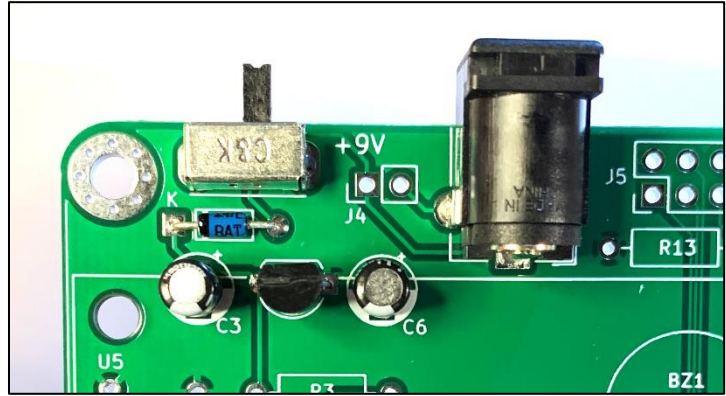


Step1: Build the power supply.

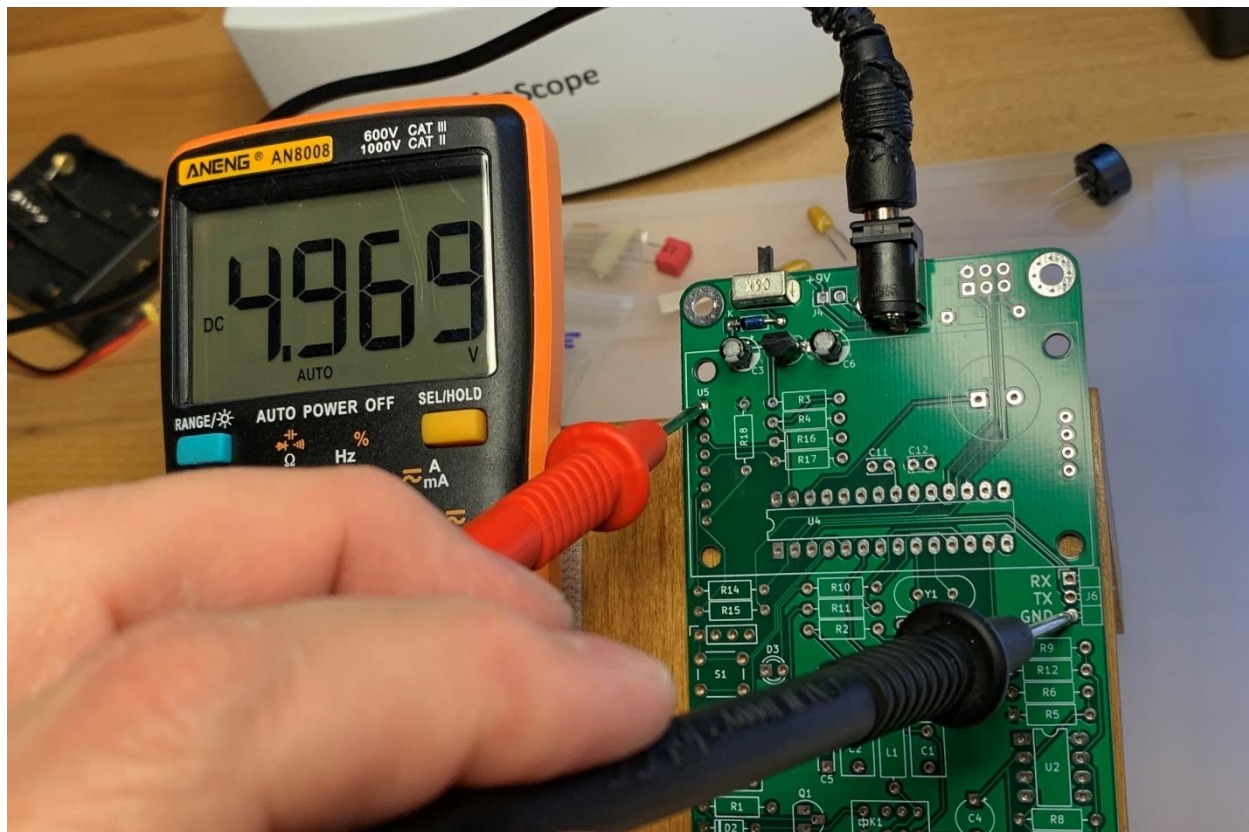
You will use the following components:
J3, SW1, D1, C3, C6, U1.

Install

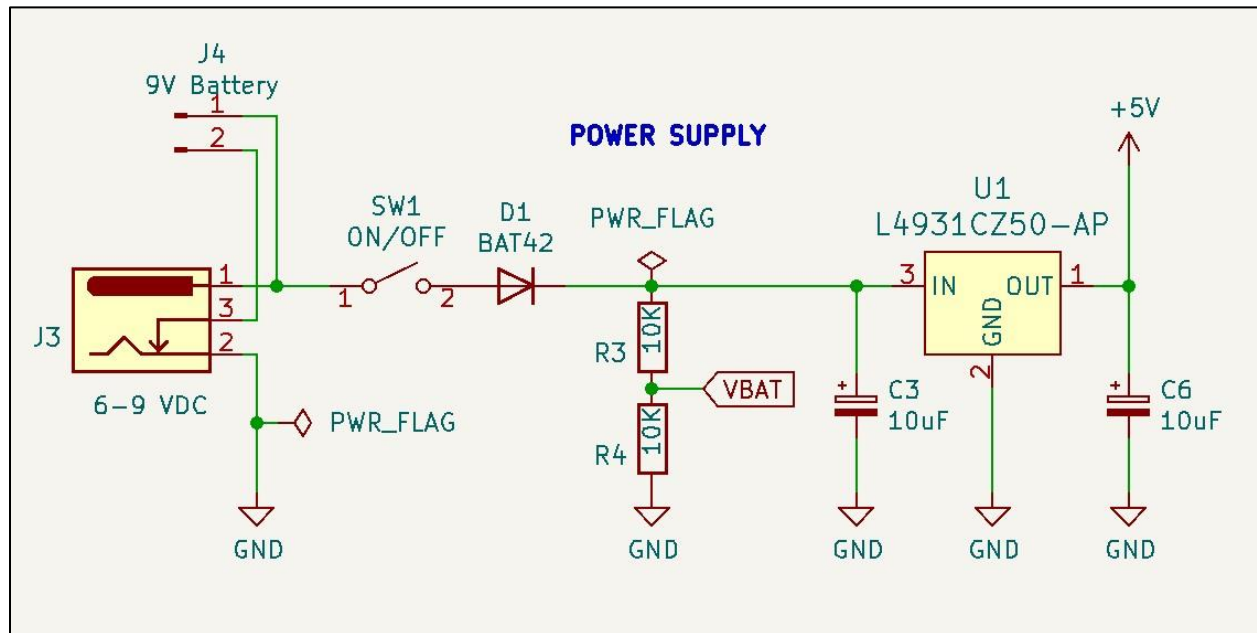
- Switch SW1.
- Diode D1, with the dark band facing left.
- Electrolytic caps C3 and C6, with the white band (-) facing downward.
- 5V regulator U1, flat side facing up.
- DC barrel connector J3.
- **Double-check:** diode band LEFT, regulator flat face UP, capacitor white bands DOWN.



Test 1: After installing these components, apply 9VDC via the barrel connector (center pin positive) and slide the switch right to the on position. Use your DMM to confirm +5V between the pin adjacent to the label “U5” and the pin labelled “GND”, as shown in the photo below. Disconnect Power. If you did not observe the correct voltage, please refer to the troubleshooting guide.



The circuit you just built provides regulated 5V power to board. Here is the schematic:



Power is provided through the DC barrel connector, J3. There is also an optional power input at J4. Note that when a power plug is inserted into J3, the mechanical connection between pins 2 and 3 is broken, removing power source J4 from the circuit. This internal switch prevents current from J3 flowing into J4, and vice versa.

Diode D1 provides protection against reversed polarity. Voltage loss across the diode is minimized by using a Schottky diode, rather than a standard 1N4007 power diode.

Voltage regulator U1 can handle an input voltage of 20V, and supply up to 300mA of current at 5V.

Capacitors C3 and C6 are required for regulator stability. The capacitor values are non-critical: values as low as 0.1uF for C1 and 2.2uF for C6 will provide adequate regulator stability.

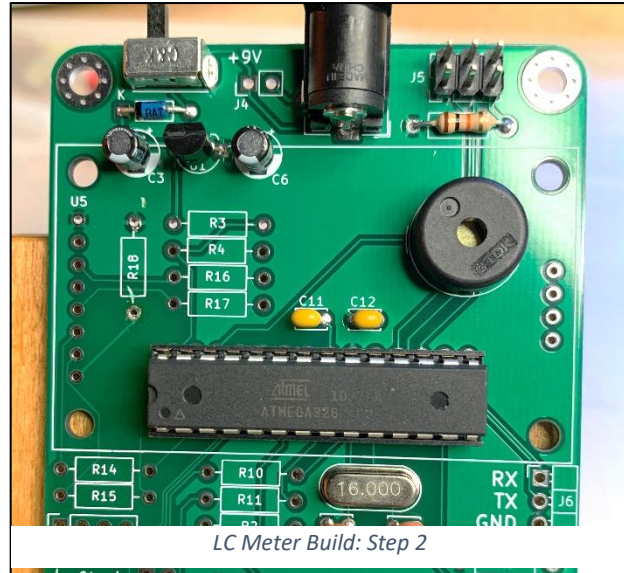
Resistors R3 and R4 provide a way of measuring the voltage input. They form a [voltage divider](#), providing a voltage on pin VBAT that is one half of the unregulated input voltage. The microcontroller cannot tolerate voltages greater than 5V on its input pins, therefore power input for this device is limited to 10V. Could you increase R3 to allow higher input voltages? Of course! Just be mindful of U1's power dissipation.

Get ready to build and program your own microcontroller.

Step 2a: Build the microcontroller circuit.

For this step you will use the following components: J5, BZ1, C9, C10, C11, C12, U4, R13, Y1. Install, from top to bottom,

- J5, the 6-pin programming header
- 10K resistor R13, below the header
- Piezo BZ1
- 100nF capacitors C11 and C12
- 16MHz crystal Y1
- 22pF capacitors C9 and C10
- 28-pin socket at the position for U4, with the notch facing left.
- U4, the ATmega328 microcontroller, with the notch facing left.
- **DOUBLE-CHECK:** U4 notch on LEFT.



Before going further, turn the board over and inspect your solder joints. Make sure that each joint is shiny and well-formed, with no solder bridging adjacent pads.

2b. Install the bootloader.

The next procedure is to install the bootloader. This may seem odd since we don't use the bootloader. However, it confirms communication between your computer and the LC meter. It also sets the MCU hardware fuses, configuring the LC meter to function like an Arduino. So,

1. Prepare your MCU programmer according to the manufacturer's instructions and connect it to your computer. I use the [AVRISP-mkII](#). The [USBtinyISP](#) programmer is another good choice.
2. Confirm an intact USB link between the computer and programmer. If you are using a PC, open the device manager and make sure that your programmer is listed. For example, on my PC the programmer appears as "libusb-win32 devices -> AVRISP mkII"
3. Start the Arduino IDE on your computer. From the Tools menu of the Arduino IDE,
 - Select Board > "Arduino UNO"
 - Select Programmer > AVRISP mkII (or your device)
4. Plug the 6-wire programming cable onto to the header J5 of your LC meter, making sure that all 6 pins are firmly seated. If the programming cable has a red wire it will be facing LEFT toward the DC jack.
5. Apply power to the LC meter, as in step 1. Slide the switch right to the ON position. (At this point, an AVRISP-mkII's LED will turn from red to green, confirming power to the board.)
6. From the Tools menu, select "Burn Bootloader".
7. Check the bottom of the IDE screen for any error messages. If it says, "Done burning bootloader", all is well. Red text with the words "Error while burning bootloader" will require further investigation. See the troubleshooting guide for suggestions.

2c. Program the LC meter.

If the bootloader burned successfully, compile and upload the [LCM_Test2 sketch](#), using the IDE command “Upload using programmer” <Ctrl><Shift>U. The sketch is working if you hear the piezo softly beeping. Here is what you just programmed:

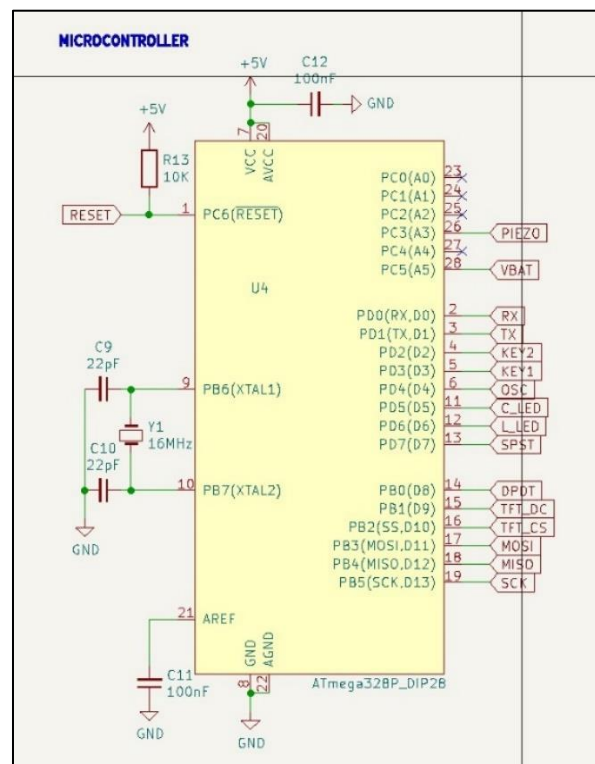
```
void beep() {  
    tone(A3,2000,100);    // 2000 Hz tone for 0.1 seconds on pin A3  
    delay(200);  
}  
  
void setup() {  
}  
  
void loop(void) {        // repeat forever:  
    beep(); beep();      // two beeps  
    delay(1000);         // and a pause  
}
```

This code is easiest to read from the bottom up. Looking at the loop() routine, we see two beeps followed by a one-second delay. Above that, beep() calls the standard Arduino tone() routine to emit a 2000Hz tone for 100ms. A3 is the MCU pin connected to the piezo element.

The microcontroller circuit requires a few support components. R13 keeps the Reset line high so that the MCU does not reset accidentally. Y1, C9, and C10 are used to provide a 16 MHz clock. C12 reduces noise on the power lines, and C11 reduces noise on the MCU internal analog circuit.

The programmer sends data to the MCU via 3 pins: MOSI, MISO, and SCK. The 3 remaining pins on the programming header are +5V, GND, and NC (no connection).

Now that we have a working microcontroller, it's time to add a display.



Step 3: Build the display circuit.

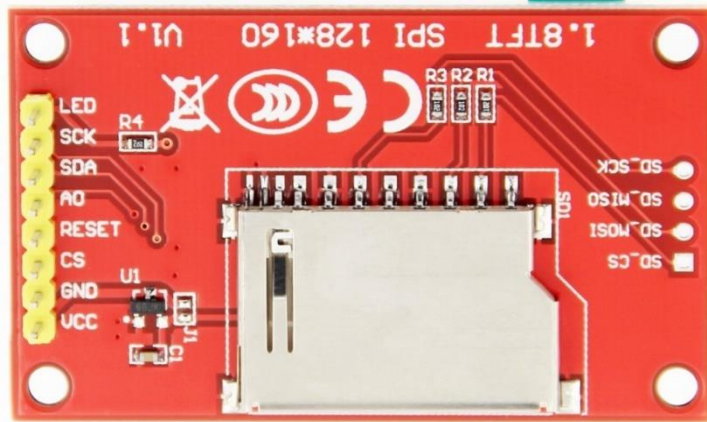
For this section you will use components R3, R4, R14-R17, Q2, TFT-Display. Install

- 10K resistors R3 and R4
- 1K resistors R14, R15, R16, R17
- 68-ohm resistor R18
- 8-pin (if not already present) and 4-pin male headers on the display module
- Fit 8-pin and 4-pin female headers on the corresponding the male headers
- Solder both female headers to the PCB.

Test 3: Turn the power on, as before, and observe the display. The display backlight should turn on. It might be difficult to see the backlight on an unlit display, so try sliding the switch on-and-off if the backlight is not immediately apparent. Go the troubleshooting guide if you can't see the backlight.

Using the display

The LC meter display is a generic LCD module commonly known as the "[MSP1803](#)". This is a 128x160 pixel TFT display mounted on a red carrier board, using the ST7735 driver and an SPI interface. It is a 3.3V device. Search [Amazon](#) and [eBay](#) for "ST7735 1.8" and you will find many vendors. The current price for the red Chinese no-brands, shown at right, is \$7. My display has 8 pins on the right with a yellow header already attached. On the left there is an additional row of 4 holes, without headers, for the SD card socket.



The 1.8" ST7735 display module (rear view)

To use this module, add the following library to your Arduino IDE:

1. Go to the GitHub page: https://github.com/Bodmer/TFT_ST77355
2. Click on the 'Code' button and select "Download ZIP"
3. Start the Arduino IDE and select Sketch > Include Library > Add .ZIP library
4. Navigate to where you downloaded the zip file and select "Open". If successful, the IDE status line will report "Library added to your libraries..."

This display library is configured for our hardware by editing its User_Setup.h file:

1. Go to the folder where the Arduino Libraries are stored. For example, on my computer, the libraries are at C:\Documents\Arduino\Libraries
2. Open the folder "TFT_ST7735-master" (or TFT_ST7735, depending on your system)
3. Open the file "User_Setup.h" with any text editor. I use Notepad.
4. Edit the following lines in the file, changing the values as shown.


```

#define TFT_CS    10
#define TFT_DC     9
#define TFT_RST   -1

```

5. Save the file.

Test 3, continued: Apply power as before. Get the sketch “[LCM_Test3](#)” from GitHub and upload it to your board. Do you see the words “ST7735 Display Test” on your display? You should see white-on-black text, with a solid blue background, and no extraneous colors. Don’t be alarmed if it isn’t right the first time. Most issues can be solved by editing the User_Setup.h file.

Here is the sketch you just uploaded:

```

#include "TFT_ST7735.h"           // display library
TFT_ST7735 tft = TFT_ST7735();  // display variable

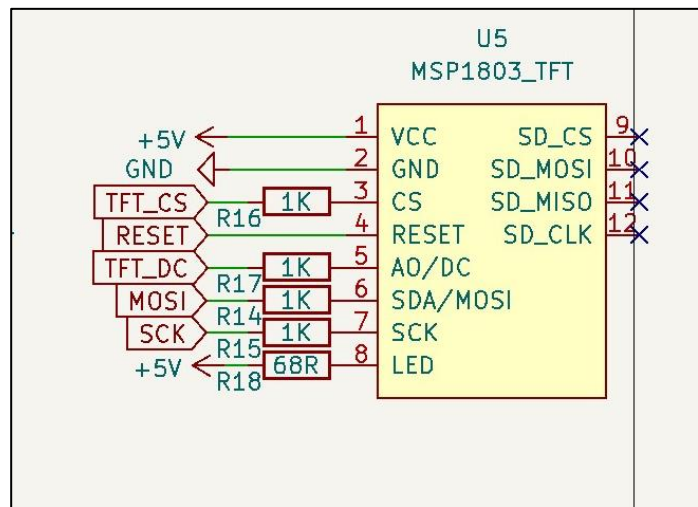
void setup() {
  tft.init();                    // initialize the display
  tft.setRotation(3);            // in landscape mode
  tft.fillScreen(TFT_BLUE);      // with blue background.
  tft.drawString("ST7735 Display Test",10,20,2); // add white-on-black text
}

void loop(void) {}

```

The first line, prefaced with “#include”, pulls in the library that we just configured. The next line instantiates the display variable “tft”. In setup(), there are a few lines to set landscape orientation and clear the display. Finally, tft.drawString() is called to show text on the screen.

Here is the display schematic. The 1K resistors R14-R17 allow us to use this 3.3V device with a 5V microcontroller. Resistor R18 provides appropriate current limiting for the backlight LEDs. Display pins 9-12, which connect to the onboard SD card, are not used in this project.



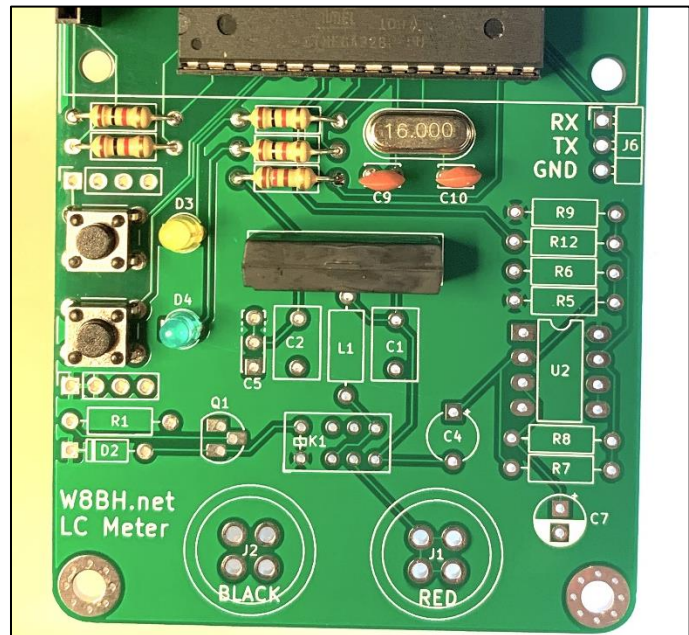
Display Schematic

Step 4a: Add the control circuits.

It's time to add a few buttons, LEDs, and relays. For this section you will use components R10, R11, R2, D3, D4, S1, S2, and K2. Install:

- 1K resistors R10 and R11.
- 120-ohm resistor R2.
- LEDs D3 (yellow) and D4 (blue). For both, the short lead goes to the square pad, so that the flat side of LEDs face left.
- Tactile switches S1 and S2.
- Reed relay K2, with the small body notch facing left.

Test 4a: Download sketch [LCM Test4](#) from GitHub. Apply power & upload the sketch to the meter. Press and hold S1. LED D3 should flash while S1 is depressed. Put your ear close to the reed relay and listen for the faint sound of its contacts opening and closing as D3 flashes. Now press and hold S2 and observe LED D4.



LC Meter Build: Step 4a

The 2 buttons and 2 LEDs are each tied to a microcontroller pin. I like starting my sketches with a list of the pins and the devices they are connected to.

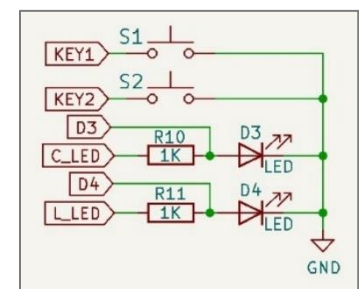
```
#define C_KEY          3                // D3: "KEY1" pushbutton
#define L_KEY          2                // D2: "KEY2" pushbutton
#define C_LED          5                // D5: LED for C_MODE
#define L_LED          6                // D6: LED for L_MODE
```

Now we can refer to the interface components by name, rather than the pin they are connected to. Next, at the start of the sketch, each of these MCU pins is declared as an input or output:

```
void initPorts() {                    // INITIALIZE MCU PINS
  pinMode(L_KEY, INPUT_PULLUP);        // L pushbutton
  pinMode(C_KEY, INPUT_PULLUP);        // C pushbutton
  pinMode(C_LED, OUTPUT);              // C LED
  pinMode(L_LED, OUTPUT);              // L LED
}
```

Note that the keys are on input pins and the LEDs are on output pins. INPUT_PULLUP means that the MCU internally applies a pullup resistor to the pin, ensuring that its value is logic 1 in the absence of any grounding input. Look at the schematic and notice that pressing S1 or S2 pulls its corresponding pin to ground.

To check for key presses, one has only to check for voltage on the corresponding pin via `digitalRead()`. A depressed switch will be at logic 0.



Resistors R10 and R11 limit LED current and set their brightness. To turn on the LEDs, the MCU applies 5V to their corresponding pins via digitalWrite(). We can control both LEDs with a single routine:

```
void setLEDs(int led1, int led2) {    // control the LEDs
    digitalWrite(L_LED,led1);        // turn the L LED on/off
    digitalWrite(C_LED,led2);        // turn the C LED on/off
}
```

Step 4b: Add more control circuits.

For this section you will use components: R1, D2, Q1, and K1. Install

- 1K resistors R1.
- 1N4148 diode D2, with band facing LEFT.
- NPN transistor Q1, with flat side facing LEFT.
- DPDT relay K1.
- **DOUBLE-CHECK:** D1 band LEFT, Q1 facing LEFT.

Test 4b: Apply power, using the same Test4 sketch as before. Press and hold S2. Do you hear the DPDT relay? It is louder than the reed relay. If you hear it, set your DMM as a continuity tester/ohmmeter. Place the black probe to the top pad of L1 and the red probe to one of the four pads on the “Red” jack J1. Press and hold S2 again. The ohmmeter should beep and show a low resistance with the same timing as the flashing D2 LED.

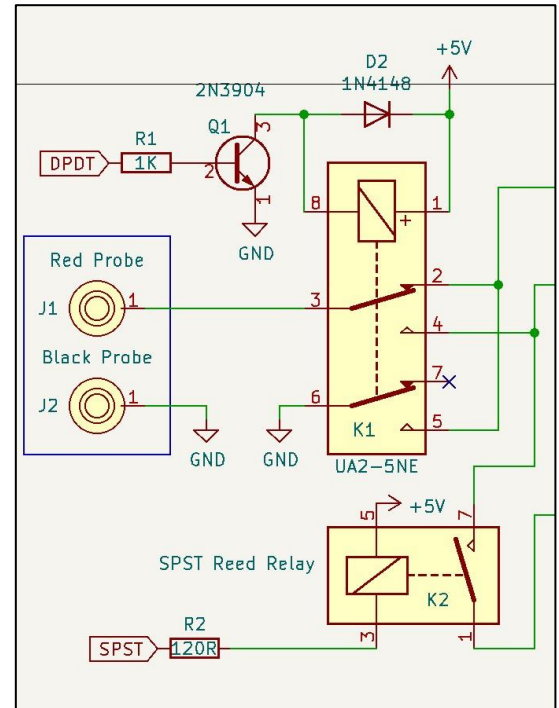
The partial schematic for both relays is at right.

In the top part of the circuit, DPDT relay K1 is controlled by transistor Q1. When the MCU pin “DPDT” is high, the base of Q1 is also high, turning on Q1 and allowing current to flow through the relay coil, causing its contacts to close. D2 is a “[snubber diode](#)”, preventing large voltage spikes on the power lines when the coil current collapses.

Reed relay K2 is a low-current device, so it does not require a driver transistor. Resistor R2 establishes a coil current of about 8 mA.



LC Meter Build: Step 4b (closeup)



Relay Schematic

Step 5: Build the oscillator.

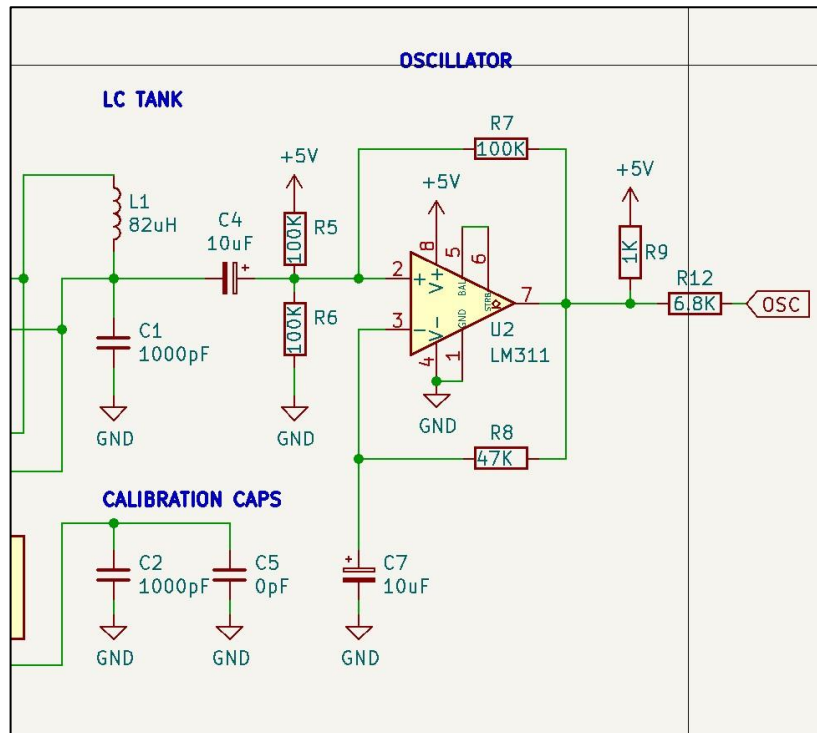
For this section you will use components L1, C1, C4, C7, R5-R9, R12, U2. Install them in the following order:

- Fixed inductor L1 and WIMA 2.5% film capacitor C1.
- C4, the tantalum capacitor, with its (+) facing UP.
- Resistors R9 (1K), R12 (6.8K) and R5-R6 (100K).
- Resistors R8 (47K) and R7 (100K).
- 10uF electrolytic cap C7, with the white stripe DOWN.
- 8-pin DIP socket for U2, with the notch UP.
- Insert U2 into its socket, with the notch UP.

Test 5: Replace the display, apply power, and upload sketch "[LCM Test5](#)". This sketch measures the oscillator's frequency output. It should display a frequency of about 500-600 kHz. If so, press your index finger to the pads of red jack J1. The added capacitance from your body should reduce the frequency of oscillation by roughly 10-20 kHz.

U2, an LM311 comparator is configured as an oscillator. The frequency-determining components of this oscillator are a coil L1 and capacitor C12, which resonate at approximately 600 kHz. The component under test is added to the LC tank -- either a coil in series with L1 or a capacitor in parallel with C12 -- and the new frequency is measured. The change in frequency is mathematically related to the unknown component's value. The accuracy of this meter depends on internal calibration capacitor C2, which is nominally a 1000pF component with 1% tolerance.

The LC meter works by measuring the oscillator frequency, calculating the unknown component's value based on the frequency, and displaying the result on the LCD screen.



Oscillator Schematic

Step 6: Complete the meter.

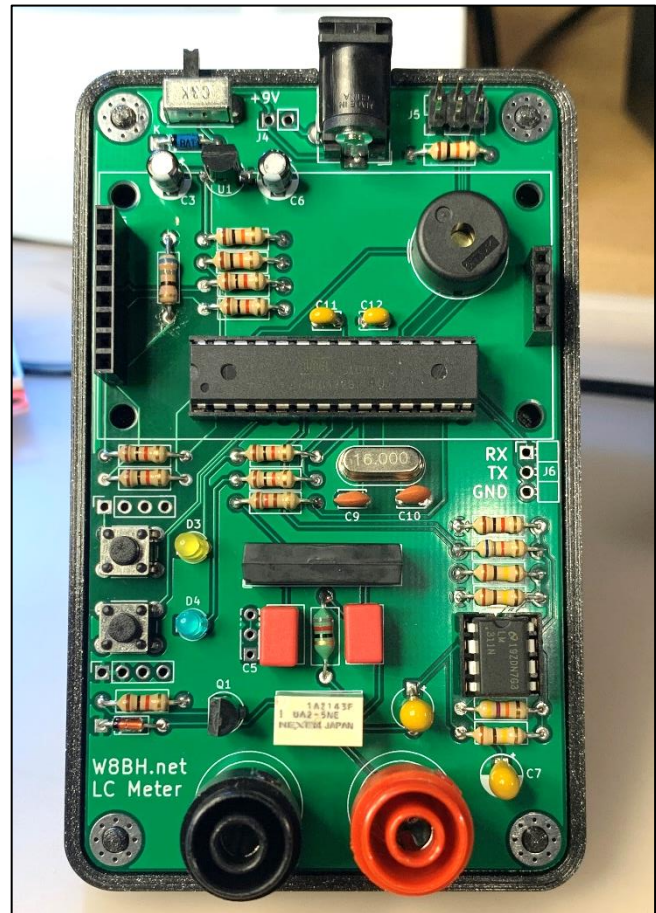
The meter is almost complete. For this section you will use the remaining components C2, J1, J2. Install

- The WIMA 1% film capacitor C2.
- Banana Jacks J1 and J2.

The banana jacks are tall components with narrow bases, so take care to mount them perpendicular to the board and parallel to each other. To do this, solder one pin of each jack then check their orientation. Solder the remaining pins when you are satisfied with jack position.

Apply power, as before. Download and install the final sketch “[LCmeter.ino](#)”.

Read the User’s Guide for tips on using your new LC meter.



Congratulations! Your LC meter is complete. I hope you enjoyed the build.

I would enjoy seeing a photo of your work.

73, Bruce.

References Links:

Part 1: Builder’s Guide: http://w8bh.net/LCmeter_BuildersGuide.pdf

Part 2: User’s Guide: http://w8bh.net/LCMeter_UsersGuide.pdf

Part 3: Software Review: http://w8bh.net/LCmeter_Part3.pdf

Part 4: Enclosures: http://w8bh.net/LCmeter_Enclosure.pdf

Source Code: <https://github.com/bhall66/LC-meter>

Test Sketches: <https://github.com/bhall66/LC-meter/tree/main/diagnostics>

Schematic: http://w8bh.net/LCmeter_Schematic.svg

Interactive BOM: [LC Meter - BN49 BOM \(w8bh.net\)](#)

PCB Gerbers: <https://github.com/bhall66/LC-meter/tree/main/PCB>

Appendix A: Troubleshooting

If you followed the step-by-step approach, you have a good idea of WHERE any problem is. Here are a few follow-up suggestions to isolate the problem further.

Problem	Suggestions
Test 1: No voltage	<ul style="list-style-type: none">a) It sounds crazy, but... make sure your power supply is connected to AC power. Then check your DC adapter output voltage with a DMM.b) If your power adapter checks out, firmly seat the adapter's plug in the corresponding PCB.c) Slide the power switch all the way to the right (ON).d) Confirm that the black band of diode D1 is on the LEFT.e) Confirm that the flat face of regulator U1 is pointed UP.f) Using your DMM, check the voltage across C3. You should see the unregulated input voltage. Now do the same across C6, and you should observe 5V.
Test 2a: Bootloader error	<ul style="list-style-type: none">a) Make sure that the MCU is correctly seated in its socket with the notch facing LEFT.b) Confirm that the remaining MCU hardware is installed, especially crystal Y1, resistor R13, and capacitors C9, and C10.c) Confirm the programmer cable is centered on the J5 programming header, with no header pins visible. The cable's red wire should be adjacent to the power jack.d) Ensure the programmer USB cable is attached correctly and that your computer can "see" the programmer (via Windows Device Manager, etc). If not, reinstall drivers for your programmer.e) Confirm that power is being supplied to the board, as in Test1.
Test 2b. No beeps/programming error	<ul style="list-style-type: none">a) Redo the bootloader burning, looking for errors.b) Make sure that power is being applied to the board.c) You must use the 'Upload using programmer' command, which is NOT the same as the usual 'Upload' command.d) If the sketch uploaded with errors but you don't hear the piezo, make sure you've uploaded the correct sketch and that piezo BZ1 is installed correctly.
Test 3: No backlight	<ul style="list-style-type: none">a) Confirm 5V power to the board, as in Test1 aboveb) Confirm proper placement of the 68-ohm resistor R18. Its color markings should be Blue-Gray-Black-Gold.c) Check power to the backlight by measuring voltage at pin8. With the display removed, check this with a small jumper wire inserted into the bottom hole of the 8-pin female header U5.d) If power is present on pin8 but there is no visible backlight, confirm that you are using an MSP1803-compatible display.

Test 3: No display text	<ul style="list-style-type: none"> a) Triple check the settings in your User_Setup.h file, located in the TFT_ST7735 library folder. b) Confirm that the sketch you uploaded is LCM_Test3. c) Confirm that power is applied and that you are using an MSP1803-compatible display. d) Check all soldering connections between U5, the 8-pin header, and the MCU.
Test 3: The display shows wrong colors or uninitialized pixels	<ul style="list-style-type: none"> a) Edit the #define TAB_COLOR setting in the User_Setup.h file. There are several choices available to you. For me the BLACKTAB choice worked the best. For each option, save the file and uploading the sketch again.
Test 4: No response to pressing S1 or S2.	<ul style="list-style-type: none"> a) Confirm the display says "Test 4" b) If the display does not respond to button presses, check soldering connections between the switches and MCU. You can simulate a button press by briefly touching MCU pins 4 or 5 to ground. c) If the display responds to button presses but the LEDs do not light, check the orientation of D3 and/or D4. It is easy to install them backwards.
Test 4: No reed relay clicks	<ul style="list-style-type: none"> a) It's very quiet; listen again with your ear against the relay. b) Confirm proper installation of R2, a 120-ohm resistor with color codes: Brown-Red-Brown-Gold. c) Confirm that the tiny body-notch on the reed relay is in the bottom-LEFT corner. d) To confirm that the relay is working, set your DMM as a continuity tester/ohmmeter. Place the black probe to the top pad of C1 and the red probe to the top pad of C2. Press and hold S1 again. The ohmmeter should beep and show a low resistance with the same timing as the flashing D3 LED.
Test 4: No DPDT relay clicks	<ul style="list-style-type: none"> a) Confirm that Q1 is a working NPN transistor, facing LEFT b) Confirm that the black band of diode D2 is facing LEFT c) Check soldering connections at R1, Q1, K1, and D2. d) Check the output of MCU pin 14 while pressing S2. It should alternate between 0 and 5V at the same rate as D4.
Test 5: No frequency output	<ul style="list-style-type: none"> a) Check solder connections on all oscillator components b) Confirm proper placement of U2 with the notch UP. Remove U2 and confirm all pins are straight. Reseat U2. c) Check polarity on capacitors C4 and C7. For both, (+) is UP and (-) is DOWN.

Appendix B: Part Sourcing

Obtaining parts has become more difficult in recent years. Below are links to selected components used in the LC meter. This is a partial list. Please be aware component availability changes from time to time. In some cases, I have listed more than one source, or have listed possible alternatives. I have not tested these alternatives, so carefully check the specifications of any part before purchasing.

Part	Supplier/Link	Cost
ATmega328P MCU 28-pin DIP	Mouser 556-ATMEGA328P-PU	2.60
28-pin IC Socket 0.3"	1-2199298-9 TE Connectivity Mouser	0.79
	1-2199298-9 TE Connectivity AMP Connectors DigiKey	0.81
8-in DIP Socket	1-2199298-2 TE Connectivity Mouser	0.25
DPDT Relay 5V	UA2-5NE KEMET Relays DigiKey	2.64
	UA2-5NU KEMET Relays DigiKey	2.63
	G6J-2P-Y DC5 Omron Electronics Inc-EMC Div DigiKey	4.73
SPST Relay 5V	Standex-Meder: Mouser 876-SIL05-1A72-71L	3.64
	9007-05-00 Coto Technology DigiKey	1.49
	9001-05-00 Coto Technology DigiKey	4.41
	HE3621A0500 Littelfuse Inc. DigiKey	1.46
BAT42 200mA diode DO-35	Mouser 511-BAT42	0.38
DC Barrel Jack PJ-102A	PJ-102A CUI Devices Mouser	0.76
	PJ-102A CUI Devices DigiKey	
	PJ-102AH CUI Devices Mouser	
WIMA 1000pf Film Cap, 2.5%	Mouser 505-FKP2D011001DHC00	0.61
	FKP2G011001D00HSSD WIMA Capacitors DigiKey	0.55
WIMA 1000pF Film Cap, 1%	Mouser 505-FKP2D011001D00ES	2.51
Electrolytic Cap, 10uF 16V	Mouser 647-UMA1C100MDD1TP	0.28
Tantalum 10uF Cap	TAP106K016CRS KYOCERA AVX DigiKey	0.53
LM311 Voltage Comparator	Mouser 926-LM311N/NOPB	1.38
	LM311P Texas Instruments Mouser	
Bourns 82uF Fixed Inductor 5%	Mouser 542-78F820J-RC	0.33
Yageo 6.8K Resistor, ¼ w, 5%	Mouser 603-CFR-25JT-52-6K8	0.10
Yageo 120R Resistor, ¼ w, 5%	Mouser 603-CFR-25JT-52-120R	0.10
Banana Jack CT3151V1 red	CT3151V1-2 Cal Test Electronics DigiKey	2.05
	CT3151V1-2 Cal Test Mouser	
Banana Jack CT3151V1 black	CT3151V1-0 Cal Test Electronics DigiKey	2.05
	CT3151V1-0 Cal Test Mouser	
Voltage Reg L4931 – 5V	L4931CZ50-AP STMicroelectronics DigiKey	0.89
	L4931CZ50-AP STMicroelectronics Mouser	0.90
Piezo Buzzer PS1240	PS1240P02BT TDK Mouser	0.60
SPDT Right-Angle Switch	OS102011MA1QN1 C&K Mouser	0.38

Last updated: 4/15/2022 10:52 AM