Arduino Multifunction 24-Bit Measuring Shield

Here's a low-cost PC-linked measuring system project which provides four accurate DC voltage measurement ranges, together with an audio frequency level and power meter, plus an optional RF level and power meter that can operate to 500MHz.

WANT TO accurately measure voltages, decibels and power levels using an Arduino? This design combines a 4-range, 24-bit DC voltmeter with both audio frequency and RF level and power meters. The audio meter can measure up to 60V RMS (up to 900W into a 4Ω load) while the RF section will measure up to 1kW into a 50Ω load, with a frequency range extending up to around 500MHz.

Apart from the RF measurement head, everything fits into a small die-cast aluminium box (119 × 94 × 57mm) which is hooked up to a PC via USB. No separate power supply is required. The RF Head is tiny; just 51 × 51 × 32mm and connects to the main box via a standard 3.5-to-3.5mm tip/ring/sleeve (TRS) (or stereo phono jack) cable.

The unit is designed to be controlled from a Windows PC (the software is for Windows 7 or later) but you could also write your own Arduino 'sketch' to suit other measurement tasks.

In short, this is a seriously useful and accurate test instrument that can be built at moderate cost. Refer to the specification panel for more details.

What's inside the box

Fig.1 on the facing page shows a simplified block diagram of the *Multifunction*

Part 1: By Jim Rowe

The *MFM* shield PCB is housed in a metal diecast case measuring 119 × 94 × 57mm. The optional *RF Head* PCB is housed in a small diecast case measuring 51 × 51 × 32mm.

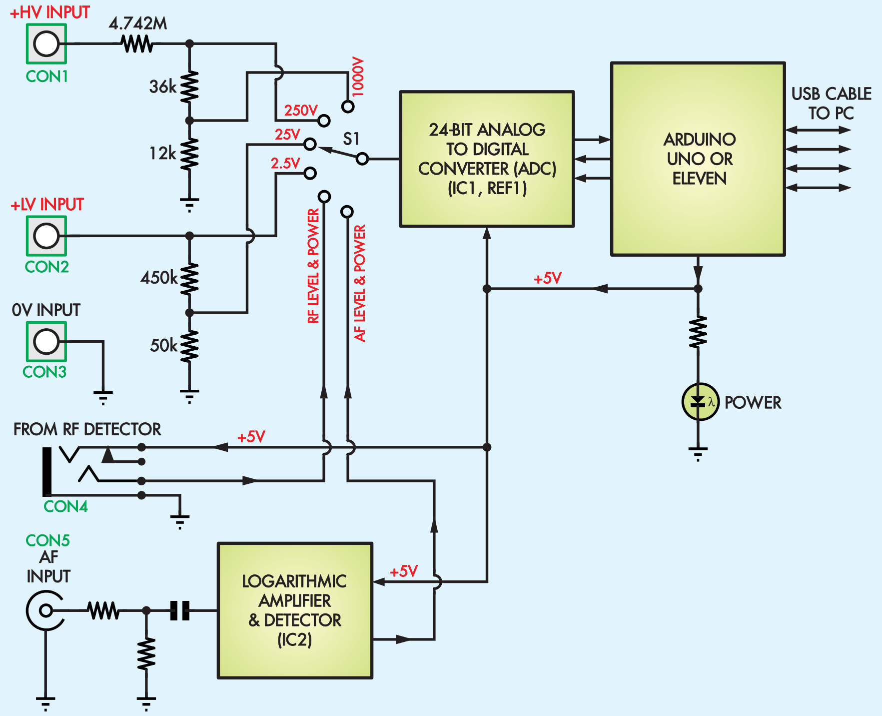
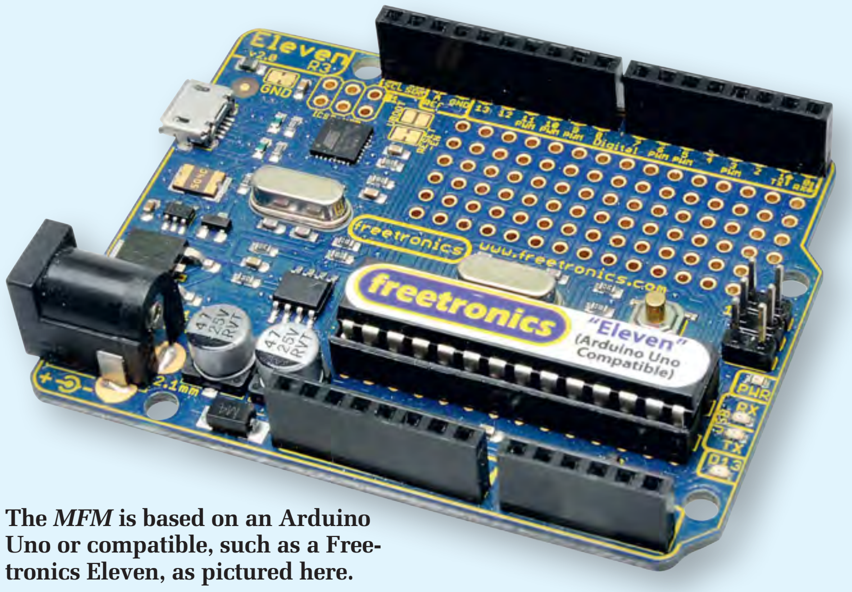


Fig.1: a simplified schematic showing the general operating principle of the *Multifunction Meter*. There are four inputs: low and high-voltage DC, audio level/power and RF level/power. S1 connects these inputs to a high-precision analogue-to-digital converter (ADC) which is controlled by the Arduino. This in turn passes the measured values on to the PC for logging or display. IC2 provides AC-to-DC conversion for audio signals, while an identical chip mounted off-board (but configured differently) feeds in RF level and power measurements via jack CON4.



The *MFM* is based on an Arduino Uno or compatible, such as a Freetronics Eleven, as pictured here.

Meter (*MFM*). At its heart is an Arduino Uno or compatible (eg, Freetronics Eleven or Duinotech Classic). This is shown at upper right and it controls the rest of the circuitry, including the USB link to the PC. The USB cable also provides 5V DC to power all of the meter's circuitry.

Coupled closely to the Arduino is the digital sampling 'engine' shown to its left. This comprises a Linear Technology LTC2400 24-bit precision ADC (analogue-to-digital converter), together with an LT1019ACS8-2.5V precision voltage reference. This combination forms a high-resolution, high accuracy digital DC voltmeter with a basic range of 0-2.5V, a resolution of 150nV (nanovolts!) and a basic accuracy of around ±0.06% (ie, ±1.5mV).

The LTC2400 is a very impressive device. It comes in an SO-8 SMD package, uses delta-sigma conversion technology and connects to the Arduino via a flexible 3-wire interface compatible with SPI and Microwire communication protocols. Other nice features include a built-in pin-selectable notch filter providing better than 110dB rejection at either 50Hz or 60Hz (ie, to reject mains hum fields), very low offset and noise, and a low supply current of only 200μA. It operates from a single 2.7-5.5V supply.

Range switch S1 controls the connections between the ADC and the various inputs and has six positions. Four of these are for the DC voltage ranges, while the remaining two are used for the *Audio Level & Power* and the *RF Level & Power* ranges respectively.

In the third position of S1, the +LV input connector CON2 is connected directly to the input of the ADC, giving a measurement range of 0-2.5V. In the next position, the ADC is connected via a 10:1 voltage divider, giving a

Features and specifications

- **Description:** A PC-linked digital measurement system combining the functions of an accurate DC voltmeter, an audio level and power meter and an RF level and power meter. PC link is via USB with an adjustable sampling rate. The application software allows saving data in CSV format for later loading, plotting or analysis.
- **Power supply:** All power comes from the host PC. Draws less than 65mA from the USB port (<325mW @ 5V).

DC Voltmeter

- **Four ranges**: 0-2.5V, 0-25V, 0-250V and 0-1000V.
- **Resolution:** 24 bits (1 part in 16,777,216) corresponding to 150nV, 1.5µV, 15µV and 60µV for each range.
- **Basic accuracy:** approximately ±0.06% on 2.5V range (±1.5mV), ±0.5% on higher ranges (±125mV, ±1.25V and ±5V respectively).
- **Input resistance:** 500kΩ on the 2.5V and 25V ranges, 4.79MΩ on the 250V and 1000V ranges.

Audio Level & Power Meter

- **Input range:** 4.2mV RMS (–37.5dBV) to 60V RMS (+35.5dBV) (83dB range)
- **Frequency response:** from below 20Hz to above 200kHz.
- **Input resistance:** 60kΩ.
- **Power readout:** watts, dBm and dBV for load impedances of 600Ω, 75Ω, 50Ω, 32Ω, 16Ω, 8Ω, 6Ω, 4Ω or 2Ω.
- **Maximum readings:**
  600Ω: 6W, +37.8dBm
  75Ω: 48W, +46.8dBm
  50Ω: 72W, +48.6dBm
  32Ω: 112.5W, +50.5dBm
  16Ω: 225W, +53.5dBm
  8Ω: 450W, +56.5dBm
  6Ω: 600W, +57.8dBm
  4Ω: 900W, +59.5dBm
  2Ω: 1800W, +62.5dBm

RF Level & Power Meter

- **Input range:** 15.8mV RMS (–36dBV) to 223.6V RMS (+47dBV) (83dB range)
- **Frequency response:** approximately 10kHz to 500MHz.
- **Input resistance:** 101kΩ.
- **Power readout:** watts, dBm and dBV for load impedances of 75Ω or 50Ω (600Ω, 32Ω, 16Ω, 8Ω, 6Ω, 4Ω and 2Ω modes also available).
- **Maximum power readings:**
  75Ω: 667W, +58.2dBm
  50Ω: 1000W, +60.0dBm
- **Minimum readings:**
  75Ω: 3.33µW, –24.8dBm
  50Ω: 4.99µW, –23.0dBm

Front panel artwork

Artwork for the two labels can be downloaded from the *EPE* website as a PDF file.

Print these out (or photocopy Fig.6), then laminate them in clear plastic for protection and then finally attach them using thin double-sided tape.

measurement range of 0-25V. In either case, the load impedance is 500kΩ.

In the fifth / sixth positions of S1, the ADC input is connected to the +HV input at CON1 via a 2-step voltage divider giving a division ratio of 100:1 for the fifth position and 400:1 for the sixth position. This gives two further DC voltage ranges of 0-250V and 0-1000V respectively.

The first position of S1 is used for the *Audio Level & Power* function, with the signal fed into a separate BNC input socket. Signal processing is performed by IC2, an Analog Devices AD8307ARZ logarithmic amplifier/detector device, which uses a progressive compression/successive detection technique to provide a dynamic range of more than 88dB with a linearity of ±0.3dB, at all frequencies between 20Hz and 100MHz.

In effect, the AD8307 is a wideband AC-to-DC converter with a logarithmic transfer function. It converts AC signals to a DC output voltage which varies from 0.25V to 2.5V, with a nominal slope of 25mV/dB. This can be adjusted via external components, as we shall see shortly.

Finally, the second position of switch S1 connects the ADC input to the tip of CON4, which interfaces with the *RF Detector Head*. We haven't included this circuitry in Fig.1, but it will be covered below. It's based on another AD8307 logarithmic amplifier detector, but with a different configuration, to allow it to operate at frequencies up to about 500MHz.

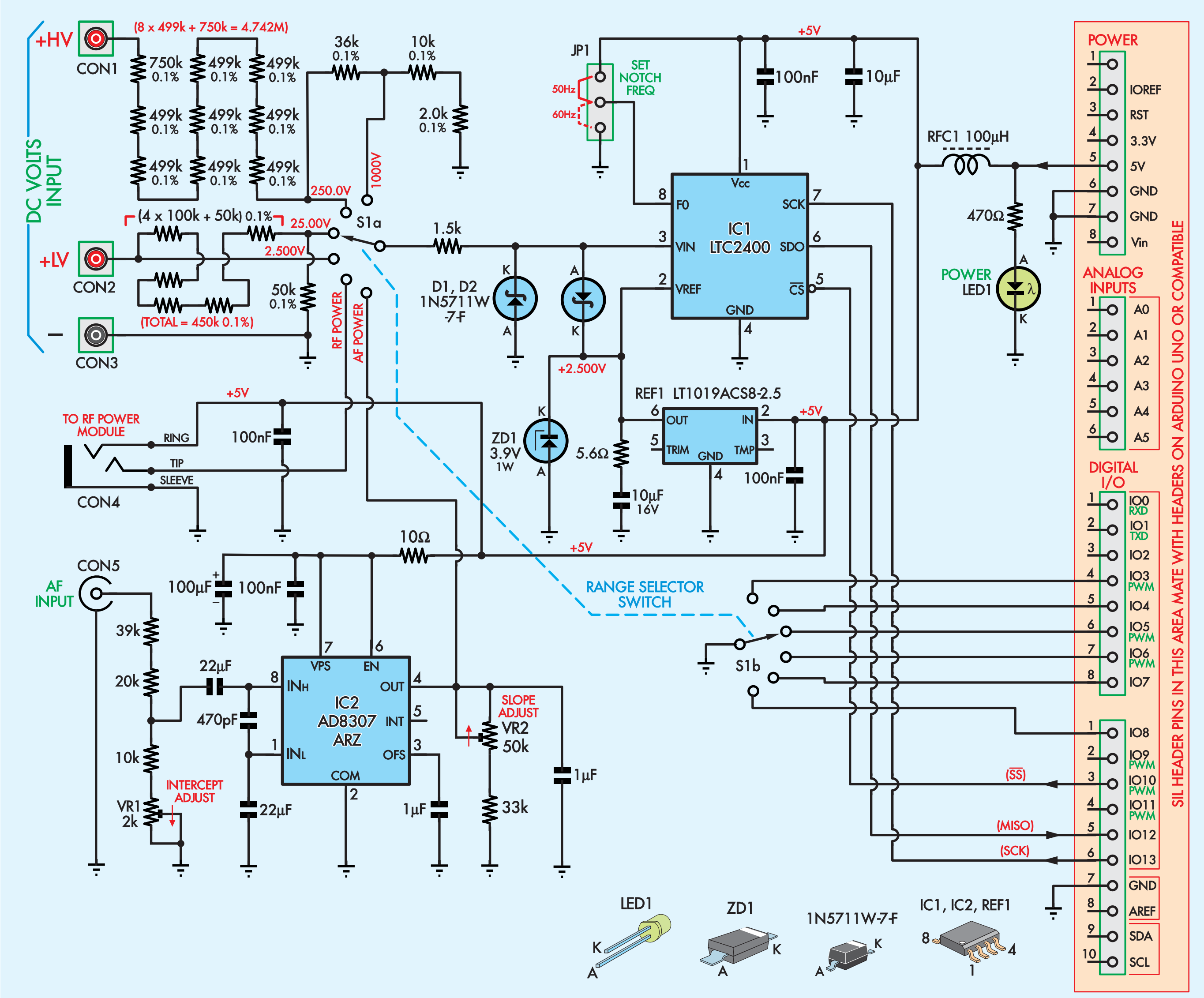
Circuit description

Fig.2 shows the full circuit diagram for the *Multifunction Meter's* main 'shield' PCB, ie, everything apart from the Arduino itself and the *RF Head*. The Arduino interface is via the SIL header pins shown on the right-hand side of the circuit, while the RF level and power measurement head connects via CON4 at centre left. More on this shortly.

Fig.2 includes the LTC2400 24-bit ADC (IC1) and LT1019ACS8-2.5 2.5V reference (REF1) which were both mentioned previously. Pin 3 of IC1 is the analog input (V_{IN}), with Schottky diodes D1 and D2 providing input overvoltage protection, in conjunction with a 1.5kΩ series resistor.

The 2.5V reference from REF1 is fed to pin 2 of IC1 (V_{REF}), while ZD1 and the series RC circuit between this pin and ground both reduce any noise present in the output of REF1 and limit any voltage rise of the 2.5V reference output due to accidental input voltage overload.

Pin 8 of IC1 is used to set its internal notch filter to either 50Hz or 60Hz, by connecting it to +5V or ground via JP1. Pins 5, 6 and 7 of IC1 form the SPI serial interface connecting it to the Arduino.



ARDUINO MULTIFUNCTION 24-BIT MEASURING SHIELD

Fig.2: the full circuit diagram for the *MFM* shield (the optional RF power head and the components on the Arduino PCB are not shown). The input switching and attenuation circuitry is shown at upper left, while the 24-bit ADC (IC1) and associated components is at upper right. The audio level and power measurement circuitry is based on IC2 at lower left, while the Arduino interface headers are on the righthand side of the circuit.

Pin 5 is the 'slave select' (enable) input, while pin 7 is the serial clock (SCK) input. Sample data emerges from pin 6, the serial data output (SDO), which connects to the MISO (master in/slave out) serial input of the Arduino, via digital I/O pin 5 (IO12).

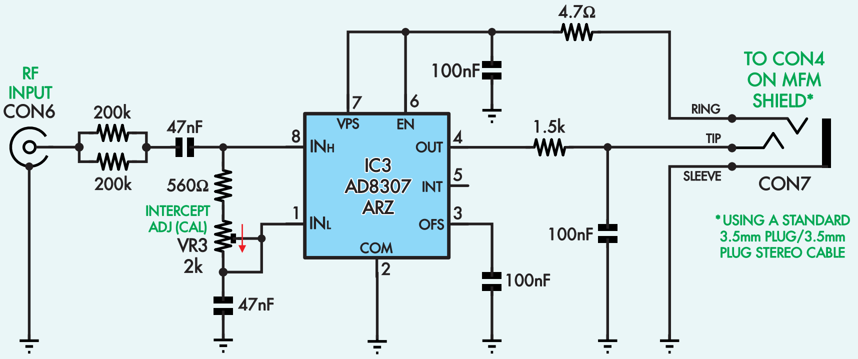
Fig.2 also shows a second pole for range switch S1 (S1b). This allows the Arduino to monitor which range the user has selected. Each position goes to a different digital input on the Arduino (pins IO3-IO8), while the rotor is grounded. These inputs have internal pull-up resistors so the firmware can tell the position of S1b (and thus S1a) by sensing which pin is pulled to ground.

The audio level and power meter circuitry is shown at lower left in Fig.2. As noted previously, this is based around IC2, an AD8307 logarithmic amplifier/detector. Its DC output voltage at pin 4 rises by a maximum of 25mV per decibel increase in the input AC voltage, which is applied between pins 8 and 1.

The 25mV/dB slope can be easily reduced by connecting an external load resistance, and this is the purpose of trimpot VR2 (50kΩ) and the 33kΩ series resistor. With VR2 adjusted to give a total of 50kΩ, IC2's output slope drops to 20mV/dB.

The audio input from CON5 is fed to pin 8 (IN_H) via a 60:1 resistive voltage divider and a 22µF capacitor, while pin 1 (IN_L) is connected to ground via another 22µF capacitor. Note that while the IN_H and IN_L pin names imply polarity, they are in fact interchangeable.

In determining the resistor values in the divider, we must consider the 1.1kΩ input resistance of the AD8307. So when trimpot VR1 in the lower leg of the nominal divider is set to 1kΩ (ie, to mid-range), the external lower leg resistance is 11kΩ, giving an effective lower leg resistance of 11kΩ // 1.1kΩ = 1.0kΩ. In conjunction with the upper leg resistance of 59kΩ, this gives the required 60:1 ratio.



RF HEAD FOR ARDUINO MFM SHIELD

Fig.3: the optional *RF Head* circuit. Note the similarity to the audio level and power measurement circuitry in Fig.2; the main difference being that various component values and placements having been changed to expand the bandwidth out to 500MHz. 5V power comes from the main board via the ring connection of CON7, while the measurement output goes to the tip.

As a result, this section produces an output of 2.5V DC for an audio input of 60V RMS, falling at a rate of 20mV per dB, down to around 840mV DC with an input of 4.26mV RMS. That results in a measurement range from +35.5dBV to –47.5dBV, for signal frequencies from below 20Hz to above 200kHz.

You might wonder why VR1 is labelled 'Intercept Adjust'. This is because by adjusting the input divider ratio, it controls the input level that corresponds with 0V output from IC2.

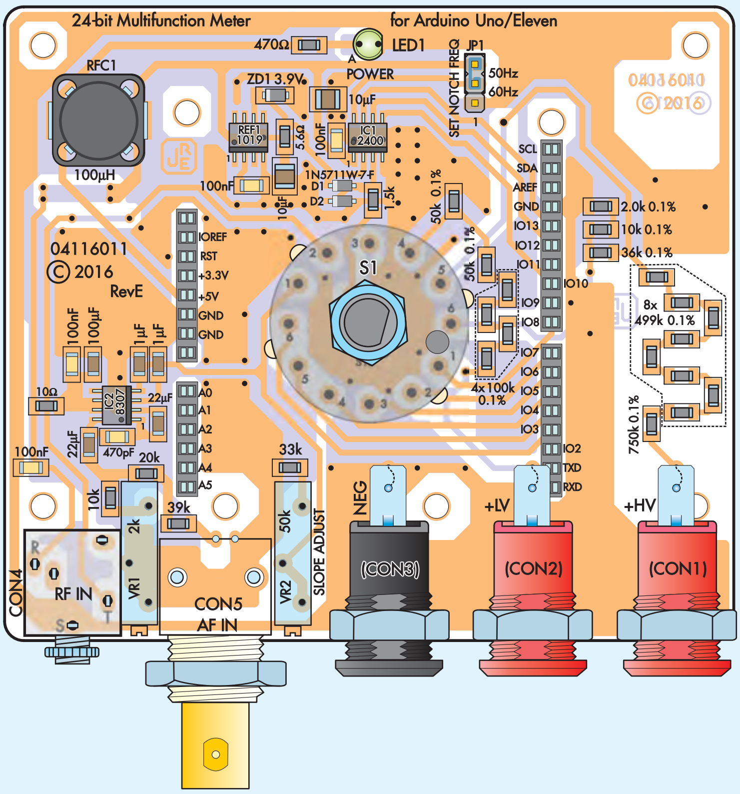
The 470pF capacitor connected between input pins 8 and 1 of IC2 is there to attenuate input frequencies above about 300kHz. This is necessary because the AD8307 can respond to frequencies up to above 500MHz, which would result in the circuit being affected by RF interference. The 1µF capacitor between IC2's output pin 4 and ground acts as a noise filter, while the 1µF capacitor bypassing pin 3 (OFS) has a similar function.

The 5V DC rail that powers all these ICs comes from the PC's USB port, via the Arduino. Inductor RFC1 provides some RF filtering in case any unwanted signals have been picked up by the wiring. LED1, in conjunction with a 470Ω current-limiting resistor provides power indication.

RF input head

The RF input section is housed in a second smaller metal box and connects to CON4 on the main PCB via a 3.5mm 'stereo' cable. Fig.3 shows the circuit of this section and again, it's based on an AD8307 logarithmic amplifier/detector (IC3), with a changed configuration to make it suitable for measuring RF signals up to about 500MHz. We put it in a separate metal box, to prevent any RF radiation from affecting the operation of the rest of the circuit.

In this case, the input divider values provide a nominal division ratio of 158:1, while the output from pin 4 has no external resistance to ground, giving a nominal slope of 25mV/dB. As a result, the *RF Head* provides a 2.5V DC output for an RF input of 223V RMS, corresponding to 1000W into 50Ω or 666.6W into 75Ω. The minimum input voltage level is lower than



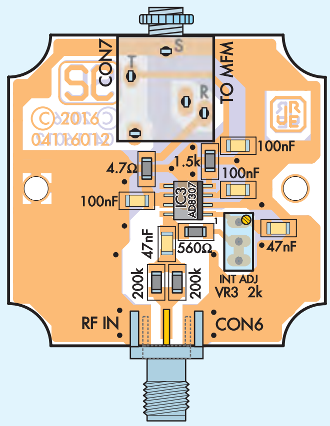


Fig.4: follow these two parts layout diagrams to build the *MFM* shield and *RF Head* PCBs. All parts are mounted on the top side of each board, while CON1-CON3 are chassis-mounted and connect to the *MFM* shield PCB via short lengths of tinned copper wire. Be sure to fit the SMD components first before moving on to the through-hole types.

7.07mV RMS, corresponding to 1µW into 50Ω or 0.66µW into 75Ω.

The *RF Head* receives its +5V DC power from the main *MFM* shield, via the same cable used to carry the output from IC3 back to CON4.

Construction

Most of the parts of the *MFM* are mounted on a single 96 × 83mm 'shield' PCB, available from the *EPE PCB Service* (code 04116011) which plugs directly into the Arduino PCB via SIL pin headers. Range switch S1 is at the centre of the shield PCB, while the RF and AF input connectors are at lower left. The complete assembly fits into a diecast box, along with three panel-mount DC input sockets (CON1-CON3) which are mounted just above the shield PCB.

All parts for the *RF Head* are fitted on a second PCB also available from the *EPE PCB Service* (code 04116012) which measures 42 × 41mm. This slips into a smaller diecast box (see photos). Use Fig.4 as a guide to assemble both boards. All components mount on the top sides of the boards.

Begin by fitting the SMD resistors, taking care not to overheat the 0.1% types. Follow with all of the SMD capacitors, which are not polarised.

Next, mount the diodes (D1, D2 and ZD1) on the main PCB, then IC3 on the *RF Head* and IC2, REF1 and IC1 on the main PCB, preferably in that order. Finally, fit inductor RFC1. After that, only the through-hole parts are left.

Install the trimpots, taking care not to get VR1 and VR2 mixed up. BNC socket CON5 can then be fitted, followed by 3.5mm stereo socket CON4. Then fit the four SIL headers used to make the interconnections between the *MFM* shield and the Arduino. As you can see from the photos, these mount on the top of the *MFM* shield PCB, with their pins soldered to the pads underneath.

Take care to use the minimum solder necessary to make a reliable joint, as the main length of each pin needs to be free from solder, flux and dirt in order to make good contact with the matching clips in each Arduino SIL socket.

With the headers all in place, fit the 3-pin SIL header for JP1 at upper right. Then pass 20mm lengths of 0.75mm tinned copper through the three holes at lower-right on the main PCB, soldering them to the pad underneath. These will later be soldered to DC input connectors CON1-CON3. Alternatively, you can use cut-down IC socket pins as shown in the photo below, but soldered connections may be more reliable.

Rotary switch S1 can now go in, after first having its spindle cut to around 17mm long (remove any swarf with a small file or hobby knife). Make sure it's oriented correctly, with the plastic post on the right side, as shown in Fig.4, so the knob will go on the correct way around.

The *RF Head* PCB fits inside a metal diecast case measuring just 51 × 51 × 32mm. It's mounted on 12mm spacers, as shown in Fig.5.

The shield PCB can now be completed by fitting LED1. This is mounted at top/rear centre with its leads at their full length so that the underside of the LED's body is 24mm above the top of the PCB (use a cardboard spacer).

To finish the RF Head PCB, fit SMA input socket CON6 to the bottom edge of the PCB and 3.5mm socket CON7 at the top. The SMA socket is 'edge mounted', with the PCB passing through its side slots and its centre pin resting on the central rectangular pad on the top of the PCB. Solder this pin to that pad, then solder the earthy 'side bars' to the matching copper pads on both the top and bottom of the PCB.

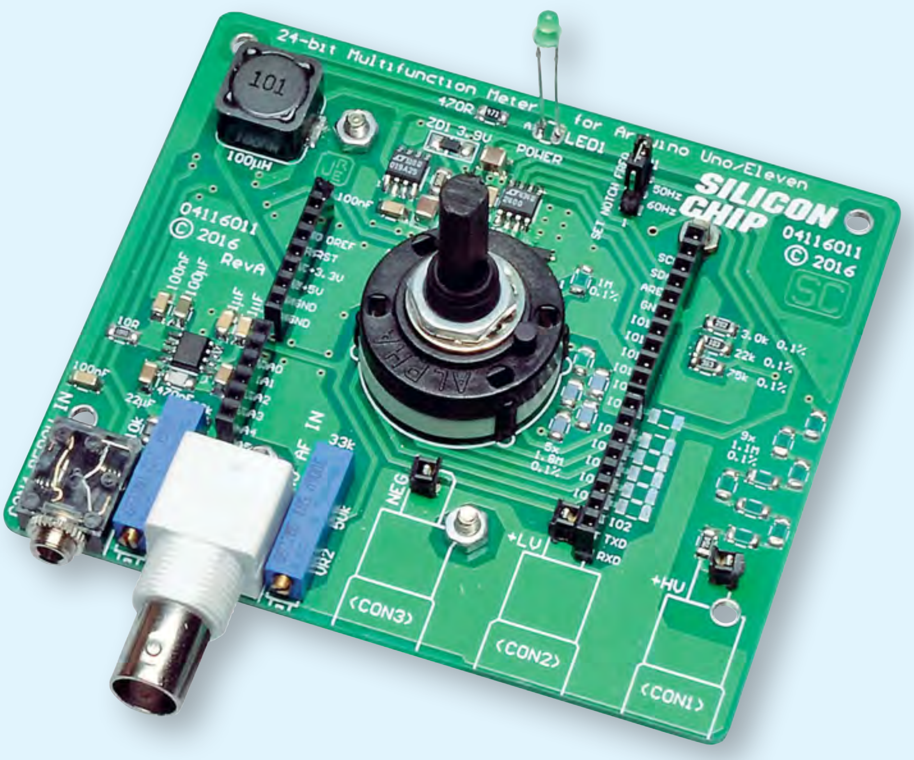
Both of your *MFM* boards should now be complete and ready to be mounted in the two boxes. We will describe how to do that next, but **we recommend you go through the test procedure (to be described in Part 2) first, before mounting it in the box.**

Preparing the boxes

The drilling and cutting details for both boxes can be downloaded from the *EPE* website. There are 11 holes to drill in the main case, plus one rectangular cut-out, and seven holes in the *RF Head* case.

Note that the rectangular cut-out is sized to suit a USB type B plug, as required by the Arduino Uno or Duinotech Classic. You could get away with making a smaller cut-out for the micro-USB input on a Freetronics Eleven.

Once all the holes and cut-outs have been made, remove any burrs from the



This view shows the fully-assembled PCB. We used cut-down IC socket pins to accept the tinned copper wire leads from CON1-CON3.

Parts List

1 double-sided PCB, 96 × 83mm, code 04116011. Available from the *EPE PCB Service*
1 diecast aluminium box, 119 × 94 × 57mm
1 lid panel label, 103 × 84.5mm
1 Arduino Uno, Freetronics Eleven or Duinotech Uno module
1 USB cable to suit Arduino module
1 2-pole 6-position rotary switch, PCB mounting (S1)
1 instrument knob, 24mm diameter
1 100µH SMD RF inductor
1 3-pin SIL header with jumper shunt (JP1)
2 red panel-mount banana sockets, fully insulated (CON1-CON2)
1 black panel-mount banana socket, fully insulated (CON3)
1 3.5mm stereo switched jack socket (CON4)
1 PCB-mount BNC socket (CON5)
1 2kΩ multi-turn horizontal trimpot (VR1)
1 50kΩ multi-turn horizontal trimpot (VR2)
1 set Arduino male/female headers
4 M3 × 25mm tapped spacers
8 M3 × 6mm machine screws
4 4.5mm OD, 12mm-long untapped spacers
4 M3 × 20mm machine screws
4 M3 hex nuts
4 stick-on rubber feet
1 14-pin DIL socket (for CON1-3)

Semiconductors
1 LTC2400CS8#PBF 24-bit ADC, SOIC-8 (IC1)
1 AD8307ARZ logarithmic amplifier/detector, SOIC-8 (IC2)
1 LT1019ACS8-2.5#PBF precision 2.5V reference, SOIC-8 (REF1)
1 3.9V 1W zener diode, SC-109B (ZD1)
1 3mm green LED (LED1)
2 1N5711W-7-F Schottky diodes, SOD-123 (D1,D2)

Capacitors (all 1206 SMD)
1 100µF 6.3V X5R
2 22µF 10V X5R
2 10µF 16V X5R
2 1µF 16V X7R
4 100nF 16V X7R
1 470pF 100V C0G/NP0

Resistors (0.25W 1206 SMD)

1 750kΩ 0.1%	1 10kΩ 1%
8 499kΩ 0.1%	1 10kΩ 0.1%
4 100kΩ 0.1%	1 2.0kΩ 0.1%
2 50kΩ 0.1%	1 1.5kΩ 1%
1 39kΩ 1%	1 470Ω 1%
1 36kΩ 0.1%	1 10Ω 1%
1 33kΩ 1%	1 5.6Ω 1%
1 20kΩ 1%	

RF Head (optional)
1 double-sided PCB, Available from the *EPE PCB Service,* coded 04116012, 42 × 41mm
1 AD8307ARZ logarithmic amplifier/detector, SOIC-8 package (IC3)
1 diecast aluminium box, 51 × 51 × 32mm
1 front panel label, 45.5 × 45.5mm
1 PCB edge-mount SMA socket (CON6)
1 3.5mm stereo switched jack socket (CON7)
1 2kΩ multi-turn vertical trimpot (VR3)
2 12mm × 4.5mm OD untapped spacers
2 M3 × 20mm machine screws
2 M3 hex nuts
1 3.5mm stereo jack to 3.5mm stereo jack cable, length to suit user requirements
4 stick-on rubber feet

Capacitors (all 1206 SMD)
3 100nF 16V X7R
2 47nF 50V X7R

Resistors (0.25W, 1% 1206 SMD)

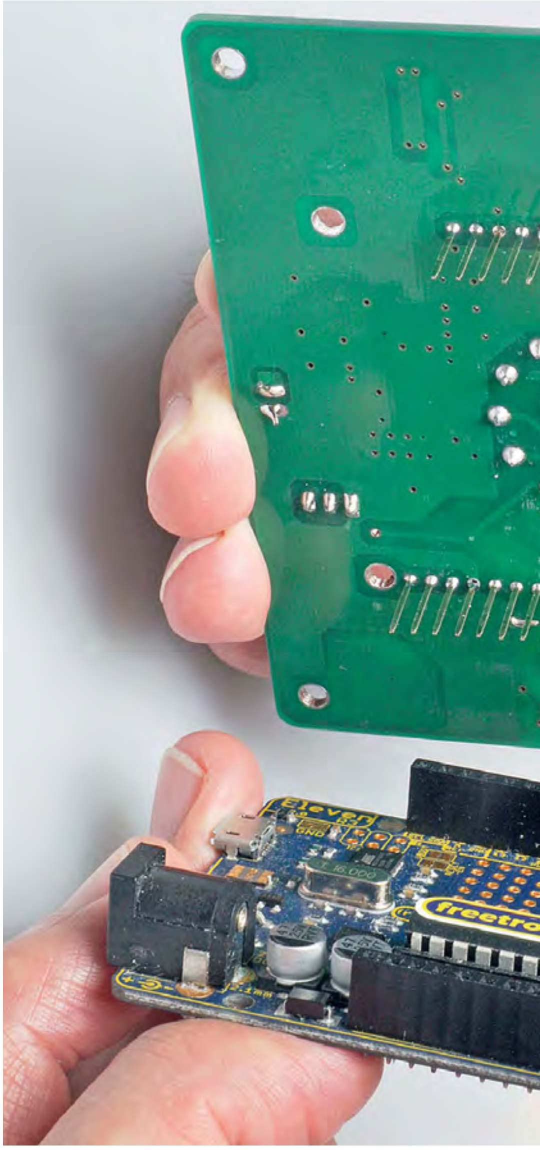
2 200kΩ	1 560Ω
1 1.5kΩ	1 4.7Ω

Reproduced by arrangement with SILICON CHIP magazine 2017. www.siliconchip.com.au

inner and outer edges with a large drill or needle file.

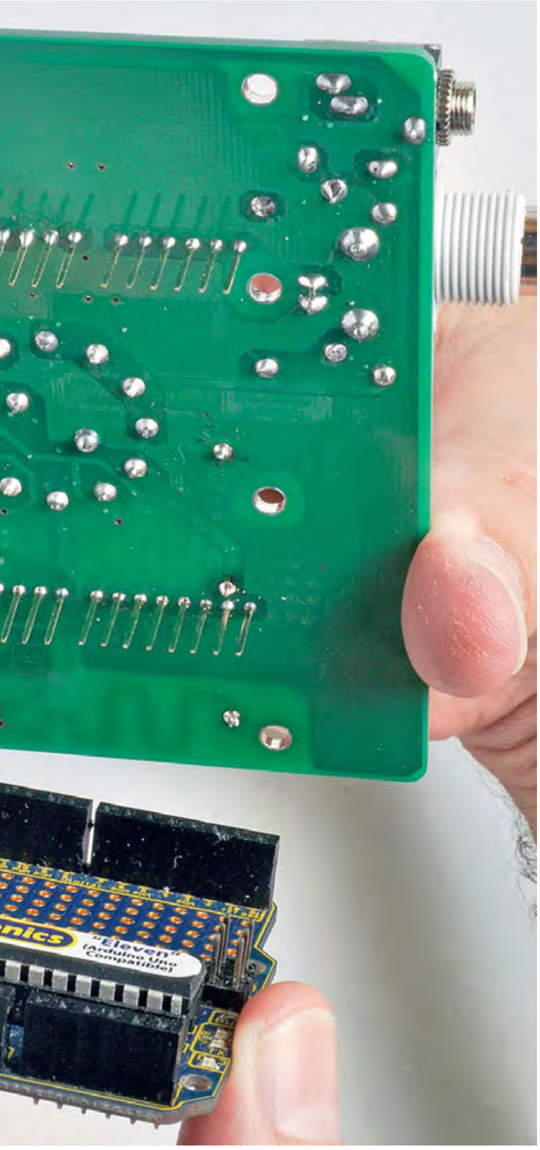
Box assembly

Fig.5 shows how the two PCBs are mounted. The procedure is as follows:
(1) Plug the *MFM* Shield PCB into the Arduino module, making sure they are properly aligned. Don't push both boards together as far as they'll go.
(2) Slip four 12mm long untapped spacers into the locations for the Arduino mounting screws between the two boards.



The header pins on the back of the *MFM* shield PCB are plugged into matching headers on the Arduino PCB. The assembly is then secured using 12mm-long untapped spacers and M3 × 20mm machine screws and nuts (see Fig.5).

(3) Push M3 × 20mm machine screws up through the Arduino PCB and each spacer and fit M3 hex nuts on the top.
(4) Gradually tighten each screw and nut until the two PCBs are held together. **Note that you will need to file a small amount of metal off one 'flat' of the nut used on the mounting screw that is very close to the rear end of the 10-way SIL header (the one at upper right in Fig.4) so it doesn't interfere.**
(5) Fit four M3 × 25mm tapped spacers into the bottom of the case using M3 × 6mm screws. Don't tighten these screws up fully yet.
(6) Remove the nut and lockwasher from the front of BNC socket CON5.



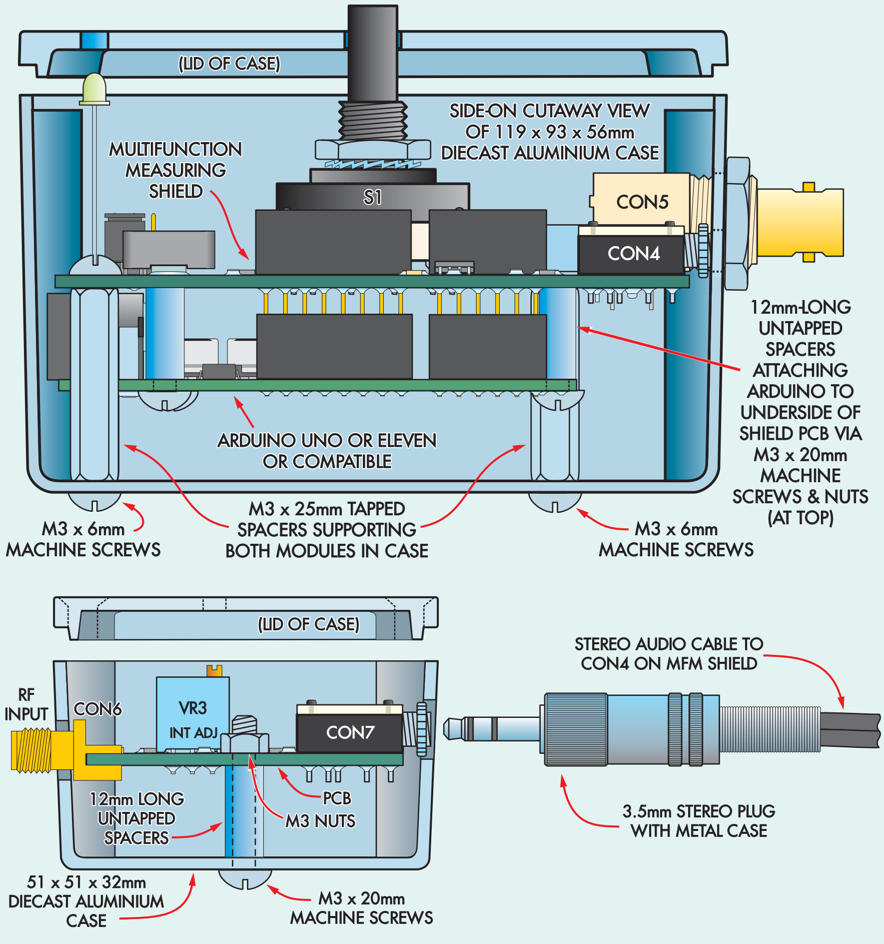


Fig.5: here's how the PCB assemblies are mounted inside the diecast cases. The Arduino board is mounted on the *MFM* shield PCB using four 12mm untapped spacers and secured with machine screws and nuts. The entire assembly is then mounted on the bottom of the case on M3 × 25mm tapped spacers. The *RF Head* PCB is secured inside its case on two 12mm-long untapped spacers.

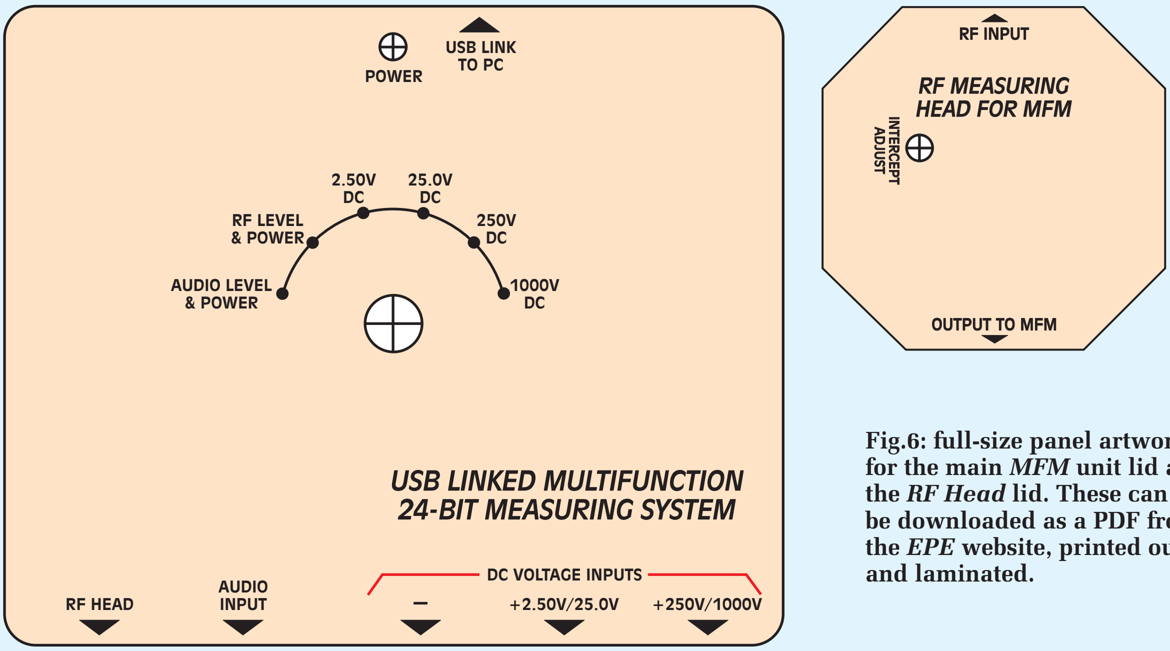
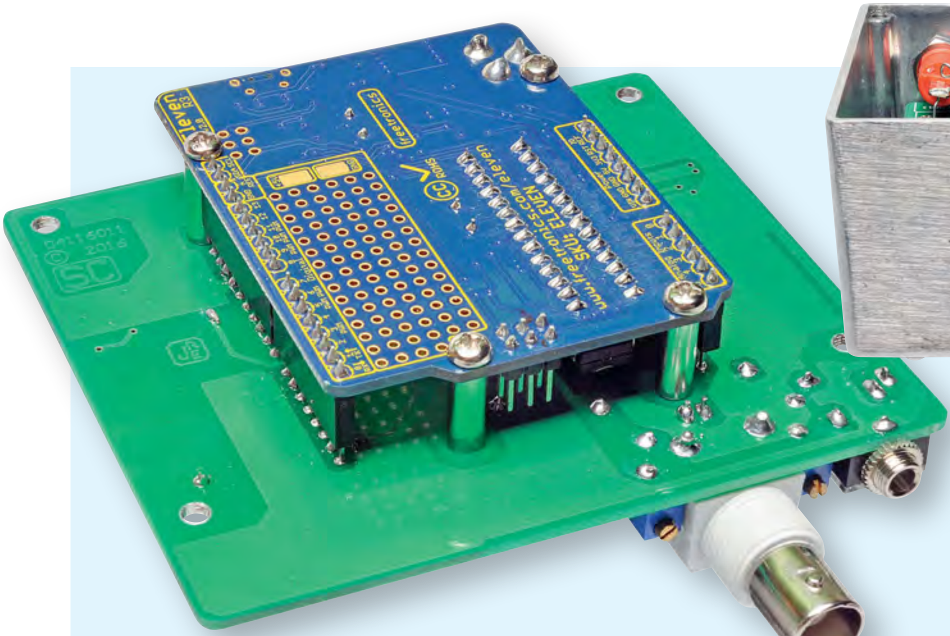


Fig.6: full-size panel artwork for the main *MFM* unit lid and the *RF Head* lid. These can also be downloaded as a PDF from the *EPE* website, printed out and laminated.



Above: this view shows the Arduino PCB (in this case, a Freetronics Eleven) mounted on the rear of the *MFM* shield PCB.

The rear panel has a cut-out to access the USB socket on the Arduino module. Note that this cut-out can be made smaller than shown if the module is fitted with a micro-USB socket.

The PCB assembly is installed in the case by first angling the BNC socket down through its hole, then forcing the rear of the assembly down into the case. The three banana sockets are then fitted and wired to the shield PCB.

(7) Insert BNC socket CON5 through its corresponding hole in the front of the box, then force the PCB assembly down into the case until it is resting on top of the four 25mm spacers.
(8) Attach the *MFM* shield board to these spacers using M3 × 6mm screws, then tighten the corresponding screws in the bottom of the case.
(9) Screw the lockwasher and nut back onto the ferrule of BNC socket CON5.
(10) Remove the nuts from the three banana sockets for CON1-CON3, push them through the 12mm holes in the front of the box and slip the nuts back on. Make sure the black socket (CON3) is closest to BNC socket CON5.
(11) Rotate the banana sockets so that their tabs are horizontal and do the nuts up tight.
(12) Bend the previously soldered lengths of tinned copper wire so they pass through the corresponding banana socket holes, then solder them in place (or push tinned copper wire leads into cut-down IC sockets and then solder these to CON1-CON3).
(13) Attach the label to the lid.
(14) Drop the lid in place, ensuring that LED1 passes through its 3mm hole, then screw the lid down and attach the knob.

RF Head assembly

Use the following steps to assemble the *RF Head*:

1) Push M3 × 20mm machine screws up through the two holes in the bottom of the box, then slip a 12mm-long untapped spacer over each.
2) Insert CON6 through the larger hole in the side of the box, then lower the PCB into place and slide it back so that CON7's ferrule lines up with its smaller hole opposite.
3) Attach the board using two M3 hex nuts. Do these up nice and tight.
4) Attach the label to the lid.
5) Secure the lid in place, making sure that the hole to allow trimpot VR3 to be adjusted is positioned above the trimpot's screw.

Next month

Your *Multifunction Meter* hardware is now complete and you're ready to tackle the remaining steps such as installing the firmware and software, calibration and finally putting the instrument to use. These topics will be covered in Part 2 next month.