

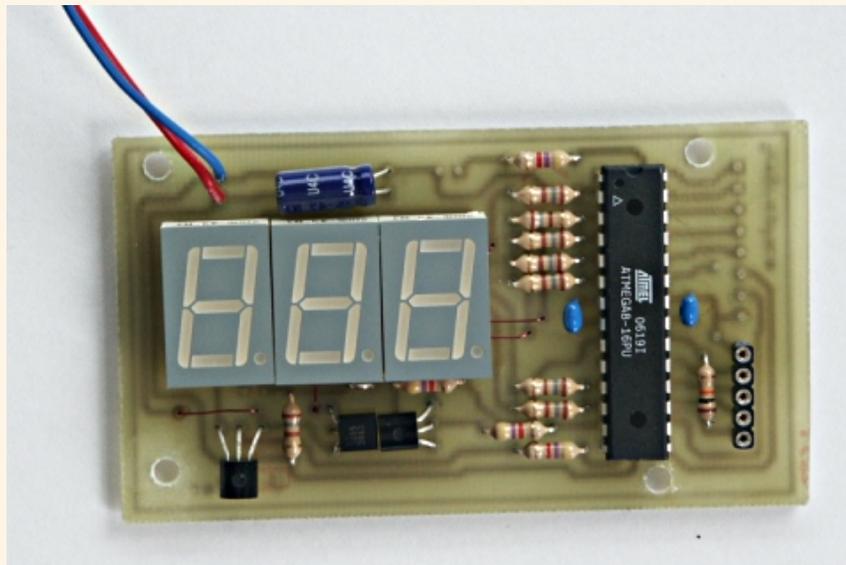


Content:

- [The problem with most commercial DVM modules](#)
- [The the solution: a simple microcontroller based DVM circuit](#)
- [A universal 3 digit display module](#)
- [Using the display](#)
- [References](#)

By Guido Socher
<guido_at_tuxgraphics.org>

Mini 3 digit display, an inexpensive digital voltmeter module



Abstract:

We build a simple and inexpensive three digit display. It's a voltmeter module but it has also general purpose digital IO pins. You could use it as well to read a digital sensor and display the value.

It can be freely programmed, calibrated and even be programmed with a non linear formula. It's a display where you can define the relation between the measured value and the displayed number.

This module is available as a kit from our online shop: <http://shop.tuxgraphics.org>

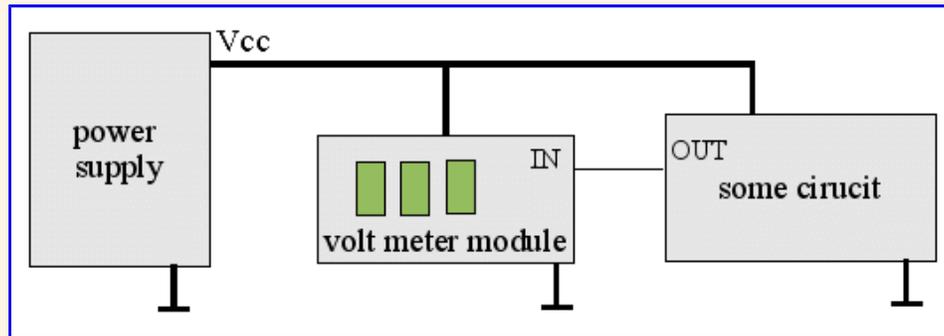
The problem with most commercial DVM modules

These days you can get at almost every local electronic store digital voltmeter modules (DVM-modules). Those instruments are meant to be placed into any equipment. They have however one big problem and usually there is no

mention of that on the box. If it is mentioned than it is usually only in the fine print inside the leaflet that comes with it:

You can't power the module from the same power that you intend to measure.

In other words something like this is not possible:



The supply voltage for the voltmeter module must be an independent ground free power source. You end up building a second transformer into your equipment.

This problem is a very annoying limitation of those little voltmeter modules and a main reason why you can not use them as a plug-in replacement for analog moving coil instruments.

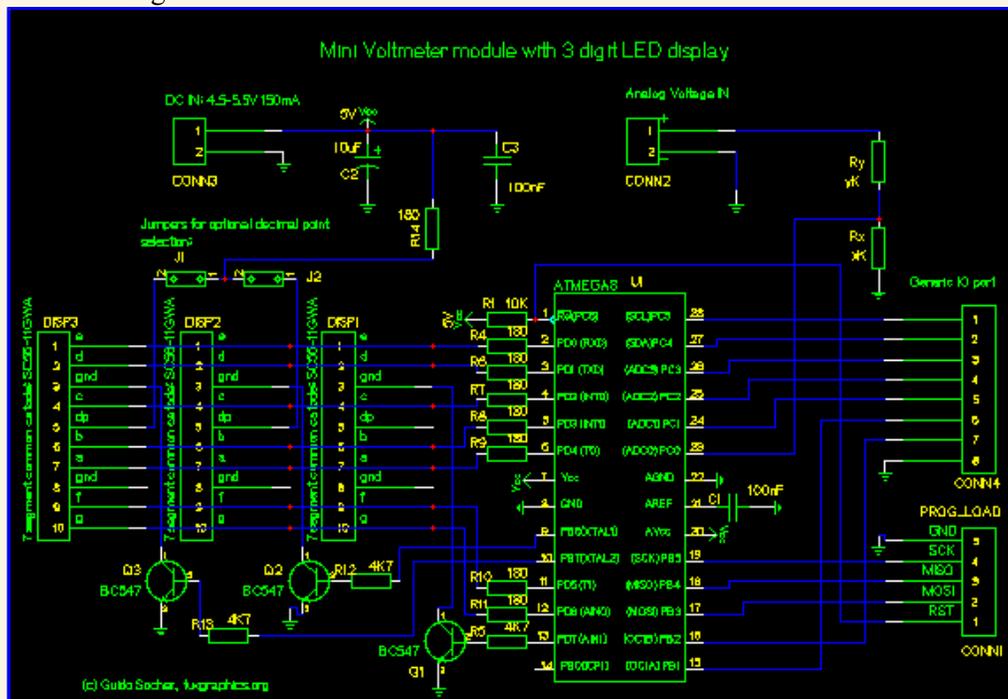
The the solution: a simple microcontroller based DVM circuit

The best solution to the problem is to take a small microcontroller with build-in ADC (analog to digital converter) and add a display.

How do you attach a simple and cheap numeric display to a microcontroller? Here is an idea that I had recently and it works very well: Take a seven segment LED display. The only problem is that 3 digits would normally require $3 \times 7 = 21$ output-pins on the microcontroller. A way to reduce the pins would be to use 74HC4511 seven segment drivers. This adds however complexity and extra cost.

A much better solution is to drive only one seven segment display at a time and one display after the other. If you do this quickly enough then the human eye will see a standing flicker free 3 digit number.

Here is the schematic drawing:



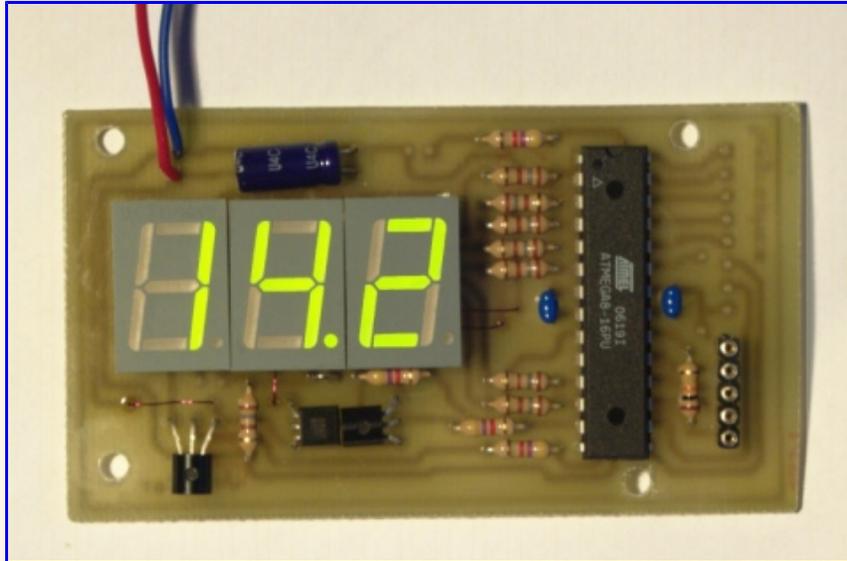
Click on the picture to see a larger version. The circuit diagram is also available as [PDF](#)

A universal 3 digit display module

This is really much more than a DVM module. Since the code is available as source it is very easy to change the mathematics inside the voltmeter. You can define what ADC (analog to digital converter) reading should result in what number on the display.

It's also much more than a voltmeter. You can not only read analog values but also digital sensors via the IO-pins on the right.

It is important for a flicker free display to refresh the LEDs frequently. You must therefore keep the code small. To read an ADC and to do some extra calculations is however no problem at all.



The power consumption of 21 LEDs is of course higher than the power consumption a LCD display but it is not problem at all if you have anyhow a transformer inside your equipment for other reasons. The module needs just under 100mA at 5V.

Using the display

The ADC inside the microcontroller is a 10bit converter and the atmega8 has internal reference of 2.56V. To be able to measure more than 0-2.56 Volt we put the voltage divider resistors (Ry and Rx) in-front. The impedance of the ADC is several Mega-Ohms. In other words you can neglect the impedance if you use resistors in the range of 10-100K for Ry and Rx. The relation between the voltage on CONN2 and the digital ADC readings is:

$$V_{\text{measure}} = \frac{\text{ADCread} * 2.56\text{V} * R_x}{1024 * (R_x + R_y)}$$

This is the formula implemented in the file analog.c Dependent on what kind of voltmeter you want you can choose appropriate values for Rx and Ry and calculate the voltage range:

Javascript voltage divider calculator. Please enter the value of Rx and Ry in KOhm without units (e.g for 10K you enter 10 into the form).

Rx: <input type="text"/>	Ry: <input type="text"/>	calculate =>	$(R_x+R_y)/R_x = \text{VDIV} =$	<input type="text"/>	Input
voltage range = 0 to <input type="text"/>					
reset					

The factor $(R_x+R_y)/R_x$ is the constant VDIV in the analog.c. You can change it as needed. I recommend to use 1% metal film precision resistors for R_x and R_y . To measure voltages in the range of 0-2.5V I recommend to use a 4K7 resistor for R_y and leave R_x open. This will provide some protection for the microcontroller against accidental voltage peaks.

If you like this type of articles then have also a look at <http://shop.tuxgraphics.org>. It is always nice to see that there are people who support my work.

Have fun and happy soldering!

References

- Software, documents and future updates: [Download page for this article](#)
- A complete kit to build this display is available from shop.tuxgraphics.org

[<--, tuxgraphics Home](#)

[Go to the index of this section](#)