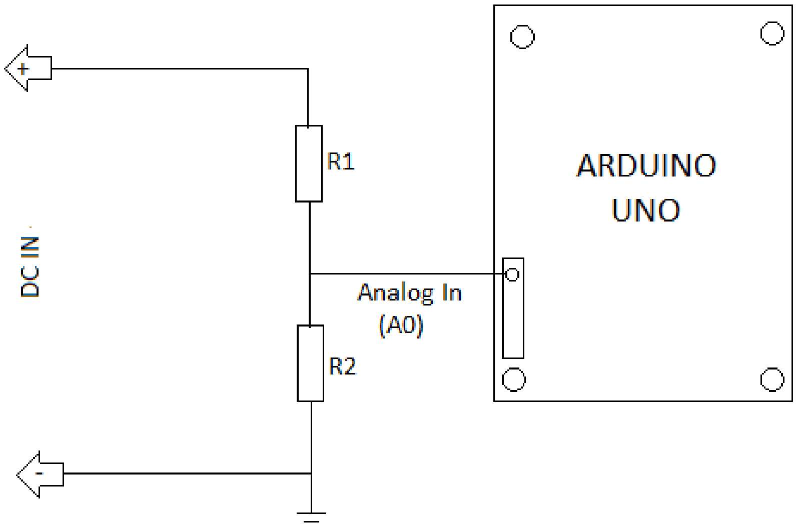
Arduino DC Volt Meter

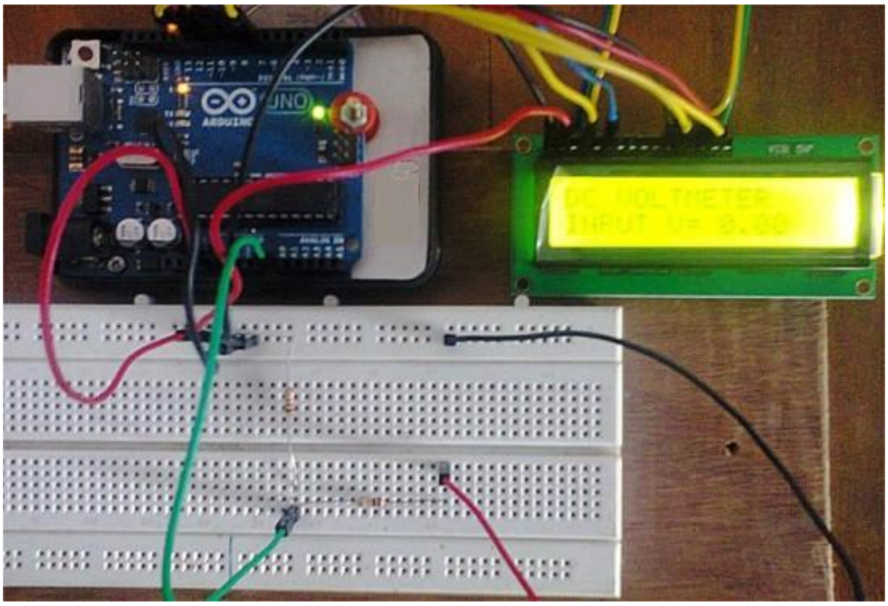
By Kevin Mc Donald ZS6KMD

This month we look at building a very useful little circuit that can be housed in your power supply or installed in a suitable enclosure as a piece of bench test equipment. A simple digital voltmeter, which can safely measure input dc voltages in 0 to 30V range. The Arduino board can be powered from a standard 9V battery pack or integrated into your (suitably smoothed and regulated) power supply circuit.

As you may be aware or are learning as we go along, the Arduino's analogue inputs can be used to measure DC voltage between 0 and 5V (when using the standard 5V analogue reference voltage) and this range can be increased by using two resistors to create a voltage divider. The voltage divider decreases the voltage being measured to within the range of the Arduino analogue inputs. Code in the Arduino sketch is then used to compute the actual voltage being measured.



The analogue sensor on the Arduino board senses the voltage on the analogue pin and converts it into a digital signal that can be processed by the microcontroller. Here, we are feeding the input voltage to the analogue pin (A0) using a simple voltage divider circuit comprising resistors R1 (100K) and R2 (10K). With the values used in the voltage divider it is possible to measure voltages from 0V to 55V using the Arduino. The junction on the voltage divider network connected to the Arduino analogue pin is equivalent to the input voltage divided by 11, so 55V ÷ 11 = 5V. In other words, when measuring 55V, the Arduino analogue pin will be at its maximum voltage of 5V. So, in practice, it is safer to label this voltmeter as "0-30V DVM" just to add a safety margin.



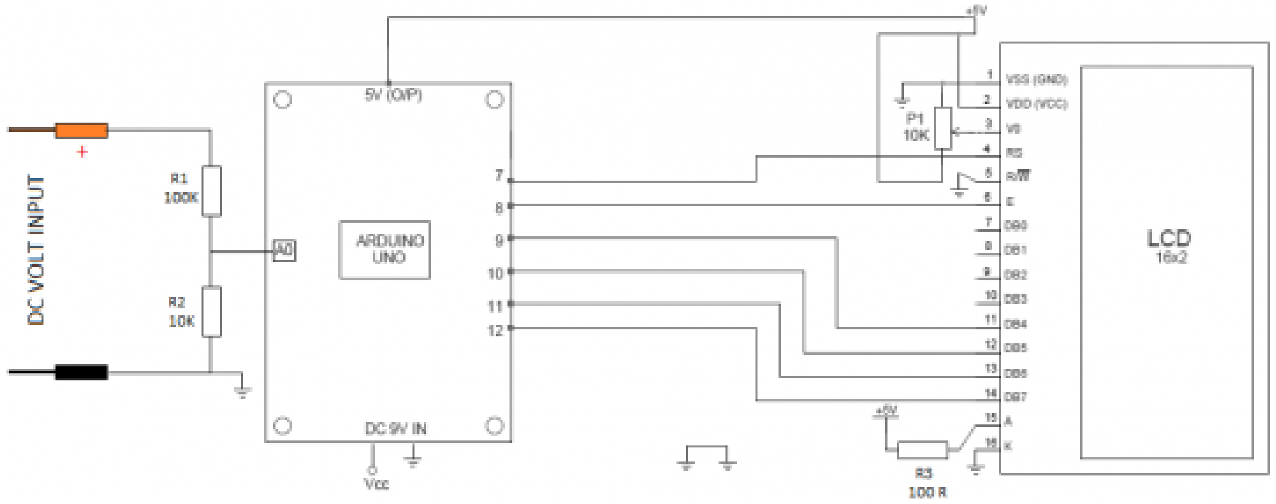
Notes

- If the display reading does not match when comparing it with your lab DVM, use an accurate DMM to find the actual resistance of R1 and R2, and replace R1=100000.0 and R2=10000.0 in the code with those values. Next check the 5V supply with the lab DVM at GND and 5V pins on the Arduino board. It might show a slightly lower value (e.g. 4.95V), replace the value in the code vout = (value * 5.0) / 1024.0 (i.e. replace the 5.0 value with the actual V reading, in this case 4.95V). Furthermore, always try to use precision 1% tolerance resistors for R1 and R2.
- The resistor values (R1 & R2) in the circuit diagram provide some over-voltage protection. Keep it in mind that any input voltage higher than about 55V could fry the Arduino. No other protection (for voltage spikes, reverse voltages or higher voltages) is incorporated in this circuit.

Arduino DVM Sketch

```
/*
DC Voltmeter
Arduino DVM based on A-D voltage divider
ZS6KMD
*/
#include <LiquidCrystal.h>
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
int analogInput = 0;
float vout = 0.0;
float vin = 0.0;
float R1 = 100000.0; // resistance of R1 (100K) NOTE -see text!
float R2 = 10000.0; // resistance of R2 (10K) NOTE - see text!
int value = 0;
void setup(){
   pinMode(analogInput, INPUT);
   lcd.begin(16, 2);
   lcd.print("DC VOLTMETER");
}
void loop(){
   // read the value at analog input
   value = analogRead(analogInput);
   vout = (value * 5.0) / 1024.0; // NOTE - see text
   vin = vout / (R2/(R1+R2));
   if (vin<0.09) {
   vin=0.0;//statement to limit any undesired reading
}
lcd.setCursor(0, 1);
lcd.print("INPUT V= ");
lcd.print(vin);
delay(500);
}
```

Schematic of the Arduino DVM Circuit



Components

- KMDuino or Arduino Uno Board
- KMDuino DVM Shield or you can build it on vero board
- 100K Resistor
- 10K Resistor
- 100R Resistor
- 10K Preset Pot
- 16×2 Parallel LCD (Hitachi HD44780 driver compatible)
- Jumper wire

Construction

Construction is simple and straight forward, if using the KMDuino DVM Shield, solder the voltage divider components, the jumper wire, the trimmer pot and backlight current limiting resistor, solder the header pins for the LCD display and check all connections. Lastly solder the LCD to the header pins.

If using vero board for the voltage divider circuit, I leave the layout design to you, but please make sure that of orientation of the components and that all connections are securely soldered. A short section of ribbon cable with header pins can be used between the Arduino or KMDuino and the LCD display.

Connect your KMDuino or Arduino to your PC, load the sketch and you will have a very accurate DVM perfect for including in a PSU project or just for testing purposes…

As always the sketch and PCB artwork can be downloaded on my web site. Please submit photos of your finished project so we can feature them in future articles.

PCB Artwork: http://www.zs6kmd.za.net/DVM.rar

Sketch: http://www.zs6kmd.za.net/DVM.txt (open the text file and copy, save as DVM.ino)

Happy Building

73 de Kevin ZS6KMD