



A2D TrueRMS Voltmeter

Internship Electronics Design Problem

To learn about crest factors and TrueRMS methods and implementations. To demonstrate my functional knowledge of modern engineering practices in electronics design.

Daniel Brandon Johnson
12/31/2009

Table of Contents

Problem Statement	3
Design Questions.....	3
Decisions	3
Circuit.....	4
Theory of Operation	10
Code.....	11
Circuit Board	15
Conclusion.....	16
About the Author	17
Self Summary	17
Daniel Brandon Johnson	17
Resume (Selected Content)	18

Problem Statement

Design the electronics for an RMS voltmeter that uses a microcontroller. The voltmeter must be able to function as a standalone system or as a module of another system.

Design Questions

The problem statement left much to be desired.

- What type of RMS quality were you looking for? TrueRMS or $DC_{AVG} * 0.707$?
- Is it meant only to measure low distortion, low frequency, sine waves? Or complex pulse trains and triangle waves?
- Did you only require the use of a microcontroller to prove that I have knowledge with digital designs? (i.e. running a display) Or did you want to see complex processing.
- Did you want me to implement a chip solution off the shelf, or to perform complex digital signal processing?

Decisions

With my design I decided to take a more challenging route. I wanted to design a TrueRMS meter that was more capable than my DMM which only does the inaccurate for most waveforms $DC_{AVG} * 0.707$ calculation. Another reason I chose to use a design like this is I wanted to have less error and have more versatility in what I was asked to design. Researching how TrueRMS meters worked, I had many choices of the type of TrueRMS methods to implement in my design. I was left with two choices: an off the shelf chip solution (AD636 Low Level, True RMS-to-DC Converter) or to do some analog work and then put the microcontroller to work.

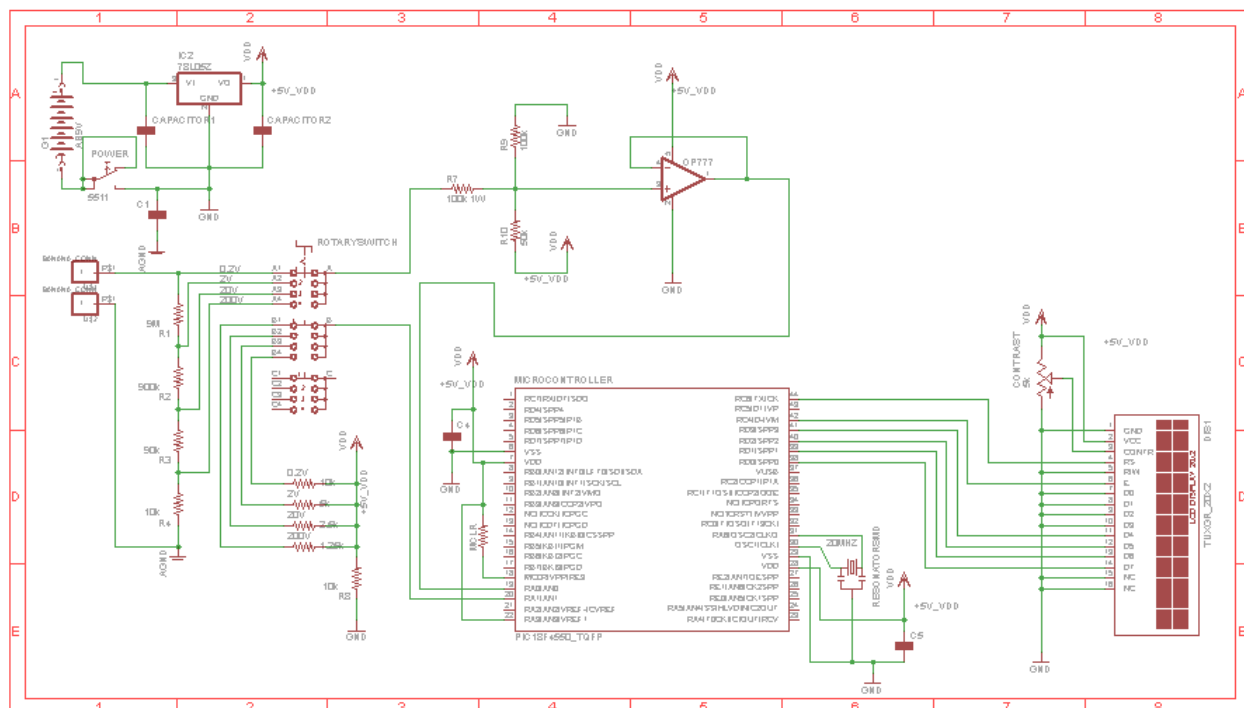


Figure 1: Full Circuit

My circuit is composed of six different blocks. And they are as follows with descriptions:

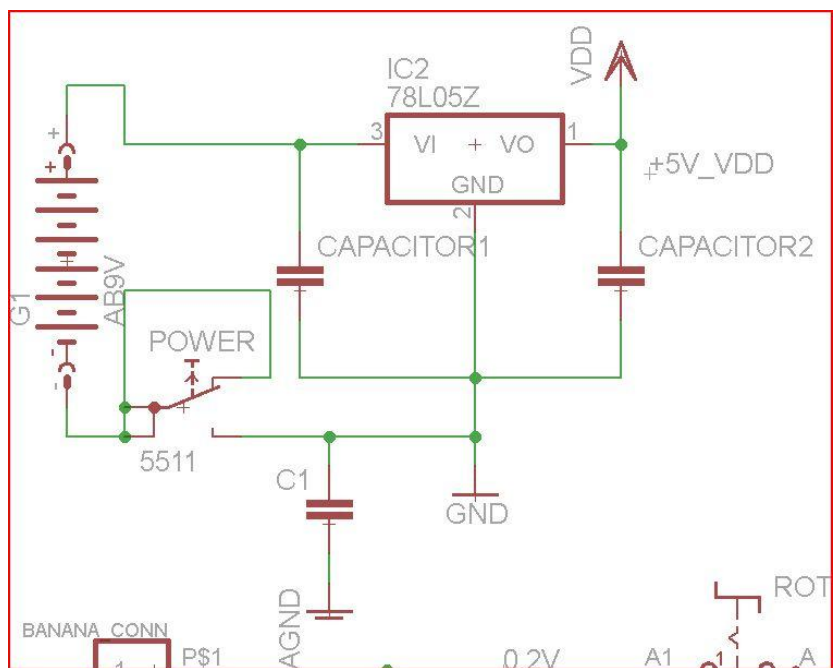


Figure 2: Power supply 9V to 5V DC with power switch

I implemented a simple 7805 voltage regulator but used a low power high precision version. The power switch disconnects the 9v battery out of the circuit. If I wanted to protect this circuit, I would have to change the LM7805 to an LM317 and add an output protection diode and a current limiting PTC resettable fuse to protect the circuit from short circuits.

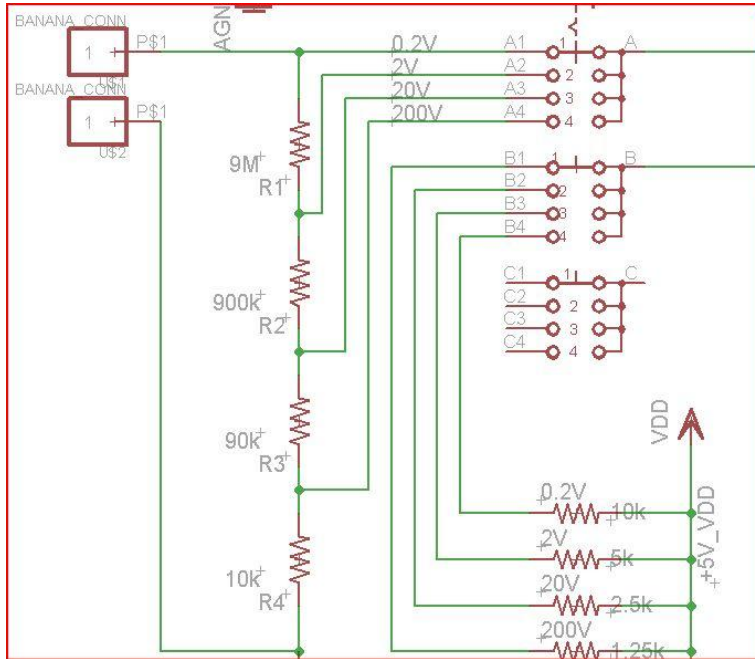


Figure 3: Input range circuit

This circuit is two banana plug style female jacks into a resistor divider network which is selectable by a rotary switch. The ranges are handled like so:

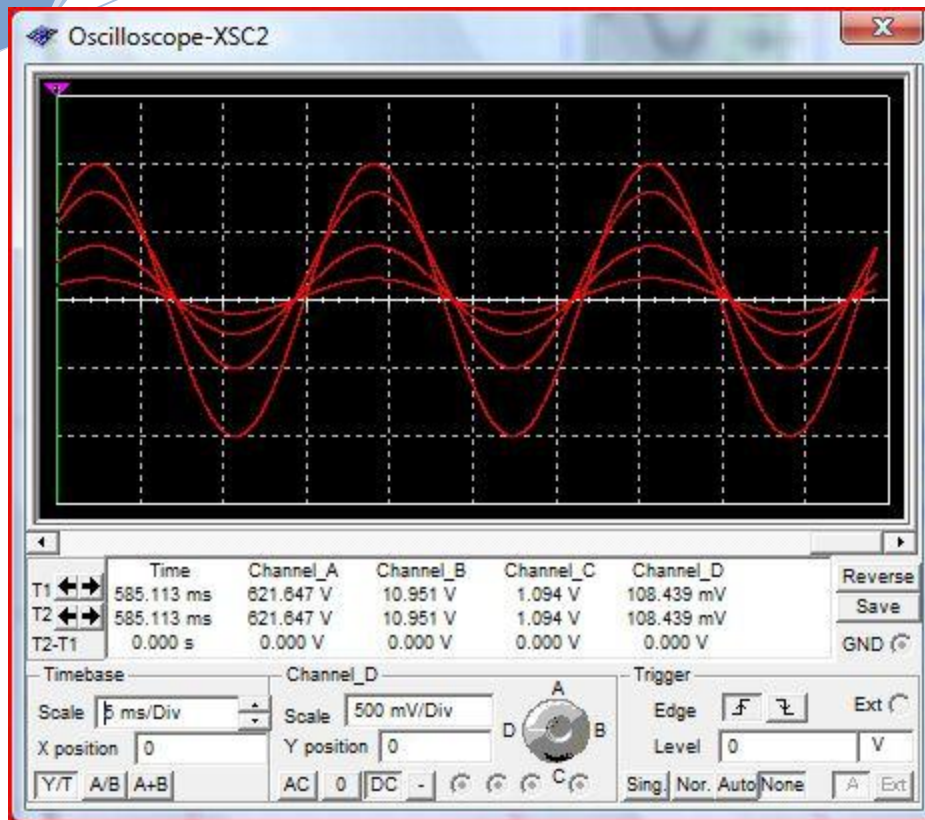
$$\frac{10M}{9M + 900k + 90k + 10k} = \frac{10M}{10M} = 1.0$$

$$\frac{10M - 9M}{9M + 900k + 90k + 10k} = \frac{1M}{10M} = 0.1$$

$$\frac{10M - 9M - 900k}{9M + 900k + 90k + 10k} = \frac{0.1M}{10M} = 0.01$$

$$\frac{10M - 9M - 900k - 90k}{9M + 900k + 90k + 10k} = \frac{0.01M}{10M} = 0.001$$

This changes the amount of voltage seen by the circuit and divides down the input voltage to a voltage that can be passed to the next stage.



This is what 1000V Sine wave looks like on all ranges. Notice at the time the screen capture was taken, channel A is in the hundreds, channel B is in the tens, channel C is in the ones, and channel D is in the millivolts range.

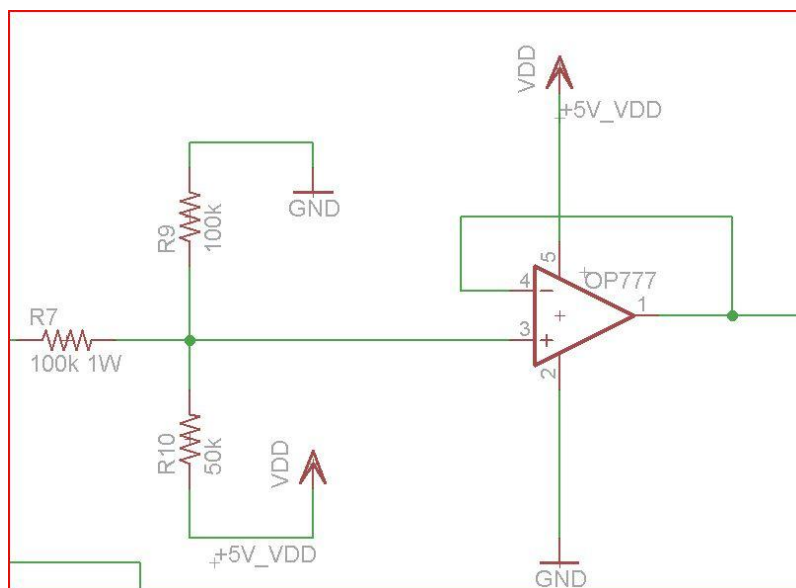
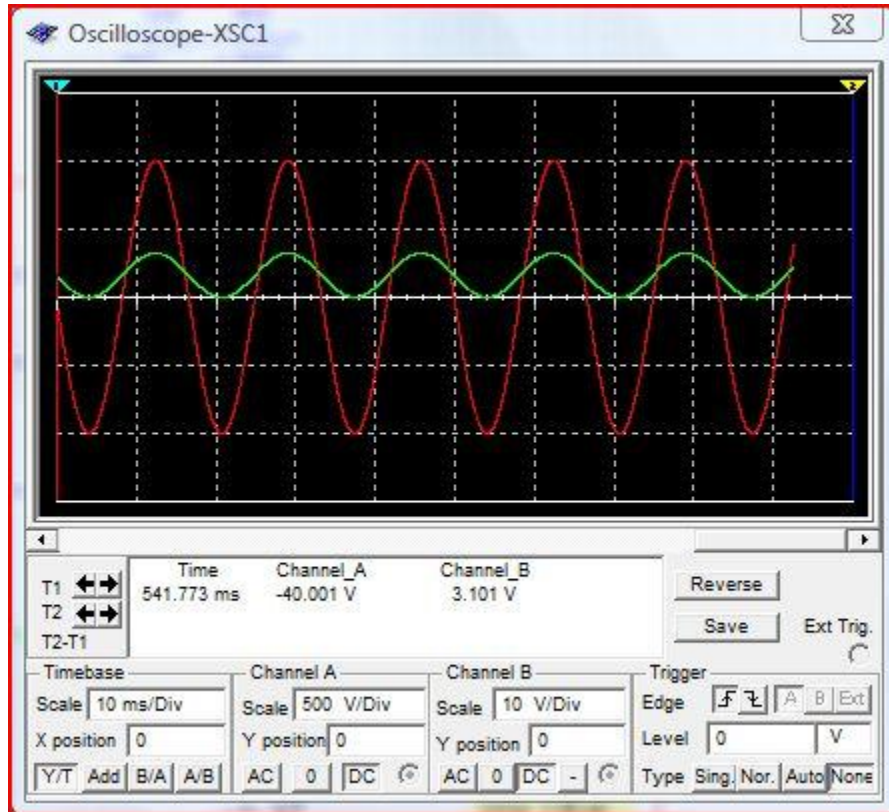


Figure 4: Signal shifter and A2D buffer driver

This is the signal shifter. I had to do analog work to implement the ADC on the microcontroller. The valid input ranges for the microcontroller V_{ref+} was +5 DC and V_{ref-} was ground or 0 DC. So I used a voltage shifter network and a voltage divider to push with DC the input signal into the positive.



Now doing this did affect the input range a little, luckily it affected it linearly so I could correct this later in software using math and a single constant for each range. Then I had to implement an analog buffer from an OP777 operational amplifier as without it, the input could not correctly drive the PIC microcontrollers analog to digital converter (too much impedance).

To determine the range of the circuit I did a simple voltage divider and used another A2D channel to have the microcontroller determine the range. Now this might seem complicated but It isn't too bad. I was forced to use this approach because of the rotary switch that was available to me (Quad Pole Quad Throw to single ended output).

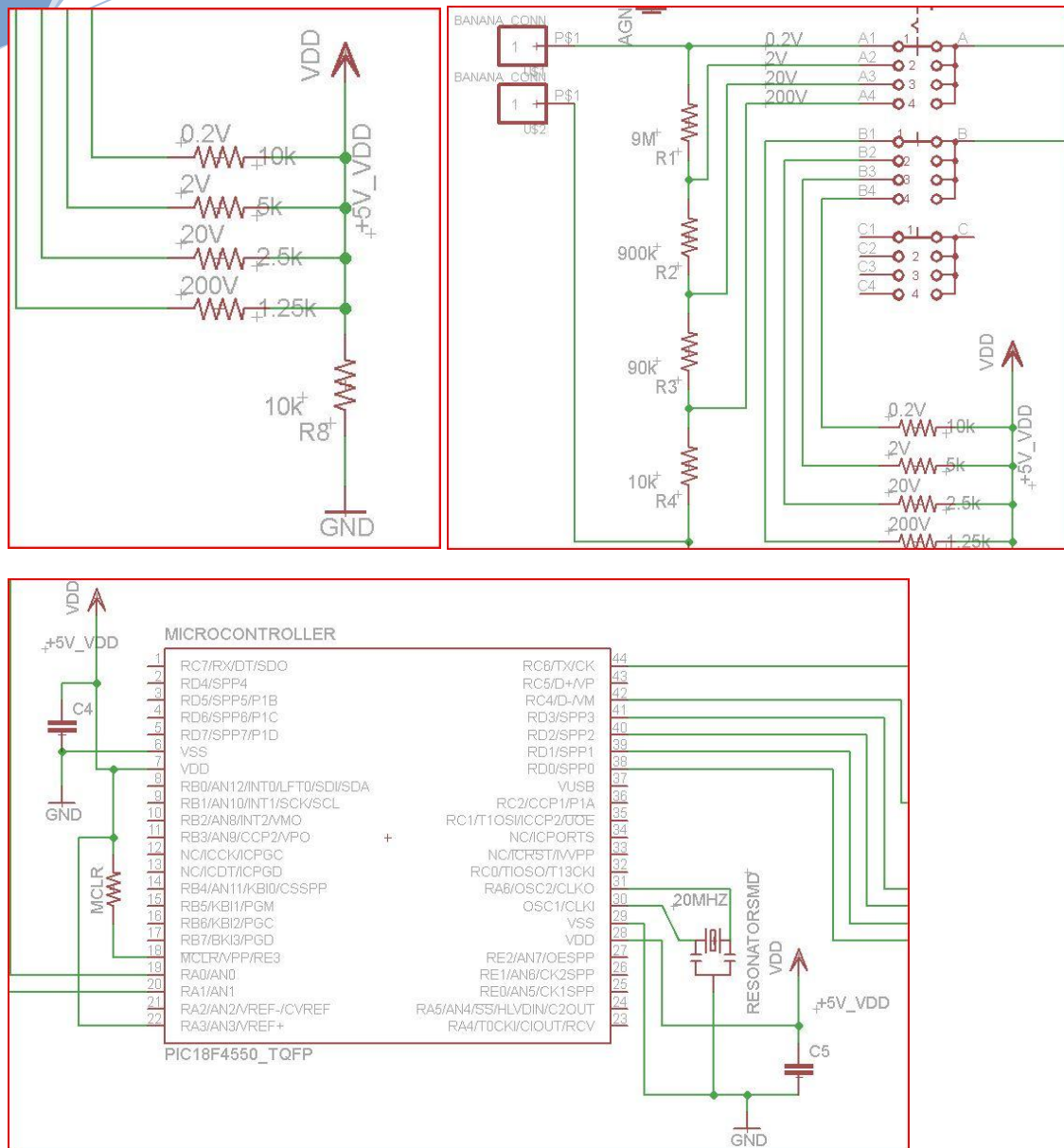


Figure 5: Microcontroller

The microcontroller I chose for this project was the PIC18F4550 in a TQFP package. It has an abundance of I/O that can be used for expandability. I am driving the clock externally with a 20Mhz ceramic resonator. Inside the chip this is being divided down to 4Mhz and then ran through a Phase Lock Loop up to 96Mhz in case I ever needed to add USB connectivity. It is then ran though one more divider by 2 down to 48Mhz which is then input into the CPU clock internally. Otherwise I am only using MCLR pin for brown out detect and 6 I/O lines to drive the LCD display and 2 ADC channels to sample the signal and check the range.

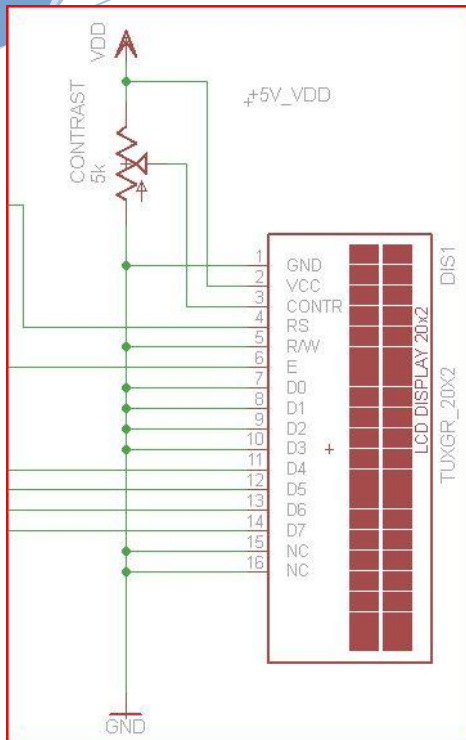


Figure 6: LCD 20x2 Character type

This is just a basic LCD which I use to display 4 pieces of calculated data: DC_offset, RMS, RMS+DC_offset, and the calculated Crest factor of the waveform.

Theory of Operation

My design samples the signal and corrects for the voltage dividers in the circuit. It then computes averages and square roots, and calculates the needed values. For example here are some of the things I needed to compute on the data: $V_{RMS} = [\text{Avg}(v^2)]^{\frac{1}{2}}$. All of the calculations can be seen in the code below, in the section labeled code.

I wrote a small program using Matlab to compute the Nyquist rate and the maximum frequency that should ever be attempted to be measured with my design:

```
>> OSC_freq = 64000000

OSC_freq = 64000000

>> OSC_period= inv(OSC_freq/2)

OSC_period = 3.1250e-008

>> Time_per_sample = 64*OSC_period

Time_per_sample = 2.0000e-006

>> Frequency_per_sample = inv(Time_per_sample)

Frequency_per_sample = 500000

>> nyquist=Frequency_per_sample/2

nyquist = 250000
```

Theoretically you should not try to measure any frequencies over 250kHz as their will be aliasing based on the maximum conversion time of the PIC18F4550's internal ADC.

Code

```
/*
Project Name: TrueRMS Voltmeter
            RMS measuring Voltmeter not limited to low frequency,
            low distortion sine waves.

Configuration:
            MicroController: PIC18F4550
            Oscillator:      HS, 48.0000 MHz
Author:      Ceramic Resonator
            Daniel Johnson
            California State University, Fullerton
            Graduate Student, December 2009
            johnson.danielb@yahoo.com

*/

#define TRUE  1
#define FALSE 0
#define r200mv 1*0.25
#define r2v    10*0.00322
#define r20v   100*0.00144
#define r200v  1000*0.000985

// Lcd pinout settings
sbit LCD_RS at RC6_bit;
sbit LCD_EN at RC4_bit;
sbit LCD_D7 at RD0_bit;
sbit LCD_D6 at RD1_bit;
sbit LCD_D5 at RD2_bit;
sbit LCD_D4 at RD3_bit;

// Pin direction
sbit LCD_RS_Direction at TRISC6_bit;
sbit LCD_EN_Direction at TRISC6_bit;
sbit LCD_D7_Direction at TRISD0_bit;
sbit LCD_D6_Direction at TRISD1_bit;
sbit LCD_D5_Direction at TRISD2_bit;
sbit LCD_D4_Direction at TRISD3_bit;

// Structures
```

```

// Shared Variables
unsigned long range;
long AvgV , DC_offset;
long RMS_Plus_DCOffset;
unsigned long RMS_value;
unsigned long CrestFactor;
unsigned long Vread_sqrd;

void setup_IO()
{
    // Interrupt settings
    //   RCON.IPEN  = 1;   // Enable Interrupt High/Low feature
    //   IPR1.RCIP  = 0;   // USART Rx set_Low
    //   INTCON.GIEH = 1;   // Enable Global Interrupts
    //   INTCON.GIEL = 1;   // Enable Global Peripheral Interrupts
    //   PIE1.RCIE  = 1;   // Enable Usart_Rx Interrupt
    // Disable USB!!!
    UCON = 0x00;
    UCFG = 0x08;
    // A/D
    ADCON1 = 0x1B;      // Set A/D pins 0,1,2,Vref+ Enabled
    TRISA  = 0x0F;      // PORTA.RD0-RD3 is input
    // 20x LCD
    TRISC = 0;
    TRISD = 0;
    Lcd_Init();
    Lcd_Cmd(_Lcd_CLEAR); // Clear display
    Lcd_Cmd(_Lcd_CURSOR_OFF); // Turn cursor off

    // Initialize delay
    Delay_ms(35);
}

void update_display()
{
    char disp_DC[5];
    char disp_RMSDC[5];
    char disp_RMS[5];
    char disp_CF[3];

    DC_offset = range * DC_offset;

```

```

FloatToStr(DC_offset, disp_DC);

RMS_Plus_DCoffset = range * RMS_Plus_DCoffset;
FloatToStr(RMS_Plus_DCoffset, disp_RMSDC);

RMS_value = range * RMS_value;
FloatToStr(RMS_value, disp_RMS);

CrestFactor = range * CrestFactor;
FloatToStr(CrestFactor, disp_CF);
Lcd_Cmd(_Lcd_CLEAR);    // Clear display
Lcd_Out(1, 1, "DC:");
Lcd_Out(1, 5, disp_DC);
Lcd_Out(1, 11, "RMS:");
Lcd_Out(1, 15, disp_RMS);
Lcd_Out(2, 1, "DC+RMS:");
Lcd_Out(2, 9, disp_RMSDC);
Lcd_Out(2, 13, "CF:");
Lcd_Out(2, 17, disp_CF);
}

void find_range()
{
    unsigned int range_check;
    range_check = adc_read(RA1); // Range    Lower    Upper
    if ( 767 < range )    // 0.2    768    1023
        range = r200mv;
    if ( 511 < range < 768 )    // 2    512    767
        range = r2v;
    if ( 255 < range < 510 )    // 20    256    511
        range = r20v;
    if ( range < 254 )    // 200    0    255
        range = r200v;
}

void sample_and_compute()    // VRMS = [ Avg(v^2) ] ^1/2
{
    long Vread;
    unsigned long Vpeak = 0;
    unsigned long Vtotal = 0;
    unsigned int s = 0;

```

```
for ( s = 0 ; s < 7000; s++ )
{
    Vread = adc_read(RA0);
    if(Vread > 512){
        Vread = Vread-512; }
    else if(Vread < 512) {
        Vread = -1*(512-Vread); }

    if (fabs(Vread) > Vpeak) {
        Vpeak = fabs(Vread); }

    Vread= Vread*0.004883;
    Vread_sqrd = Vread * Vread;
    Vtotal = Vread_sqrd+Vtotal;
}
AvgV = Vtotal/7000;

DC_offset = AvgV-3.23;
RMS_Plus_DCoffset= sqrt(AvgV);
RMS_value = RMS_Plus_DCoffset - DC_offset;
CrestFactor = Vpeak/RMS_Value;
}

void main()
{
    setup_IO();           // Initializations
    while(TRUE)
    {
        sample_and_compute();
        update_display();
    }
}
```

Circuit Board

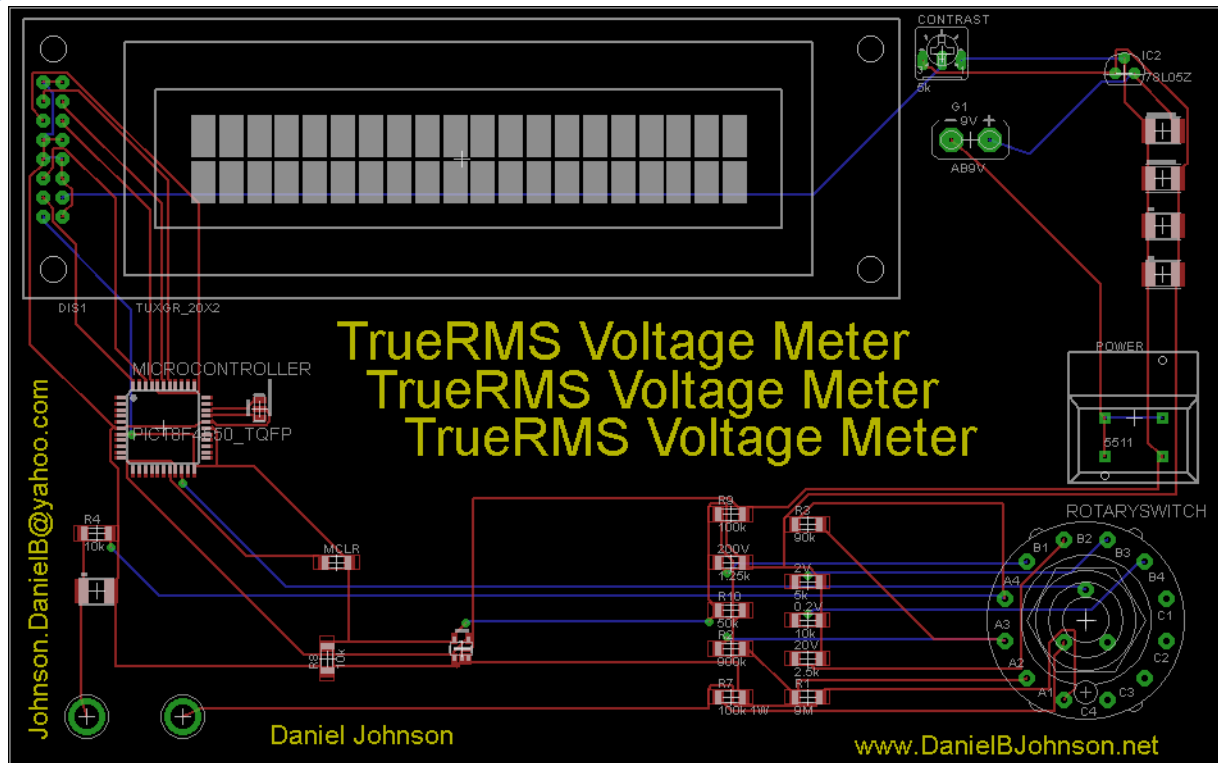


Figure 7: PCB board ready to be fabricated

I just did a quick board layout to show I have experience with building circuit boards.

Green : vias , holes

Blue : bottom layer traces

Red : top layer traces

Yellow : top layer silk screen

Grey : part outline and smd pads

Conclusion

I am confident in my design. There are quite a few small issues that even though, theoretically my design is sound there could be problems. For instance, ambient temperature adds significant error to my input because even though I am using 1% resistors, they can react differently. I opted to do more work on my design to showcase my skill set. If this was to be a finished product I likely would have opted to use the AD636 RMS-to-DCdB chip because it is likely much more accurate and would be simpler to design using it as a drop in solution. The PIC microcontroller I used had an 8x8 hardware multiplier on board which would be extremely efficient at handling the RMS calculations, however I have about my compiler, that it is making use of it, or if I would have to go in and write some assembly to make use of the extra functionality. Seeing as how I learned quite a bit from working on this project, I added some extra functionality to the design and compute the crest factor of the input under test. This is ideal as you want to know everything you can about what you are testing. I am not sure if the nicer meters on the market will do that for you. The Crest factor is an important parameter to understand when trying to take accurate measurements of low frequency signals for describing the quality of an AC power waveform.

About the Author

Self Summary

DANIEL BRANDON JOHNSON

12490 Fairbanks Drive	◆ 1(714) 679-8560	◆ Johnson.DanielB@yahoo.com	◆ www.DanielBJohnson.net
Tustin, CA 92782	Cell Phone	Email	Website

I am currently pursuing a Masters degree in Electrical Engineering with an emphasis in control systems, robotics, and automation. I am interested in electronics, engineering and biomedical engineering. I am on the Dean's list for my last semester of undergraduate coursework, and am heavily active in both my schools activities and the Institute of Electrical and Electronics Engineers club.

Activities and Club Affiliations:

- ◆ Institute of Electrical and Electronics Engineers (**IEEE.org**). Volunteered as an elected officer of student section and presented workshops to teach peers new skills. Positions held: Project Manager, Webmaster
- ◆ International Society of Automation (**ISA.org**)
- ◆ Presented a neuromuscular electrical stimulator design for biomedical engineering applications at the 2009 Savant International System on a Chip Conference in Newport Beach, California. This research paper won third place, competing against 14 other entries including group research doctorate theses.

Resume (Selected Content)

Electrical Engineer

To assist Masimo as a United States citizen and graduate student college intern.

ENGINEERING SKILLS

<i>Accelerometers</i>	<i>Digital Signal processing</i>	<i>Microcontrollers</i>
<i>Altium Designer</i>	<i>Embedded Systems</i>	<i>MPLAB IDE</i>
<i>Assembly Programming</i>	<i>EW MultiSIM and MultiHDL</i>	<i>Nano Electrical Devices</i>
<i>AutoCAD</i>	<i>Feedback Control Systems</i>	<i>PCB Design and Fabrication</i>
<i>Automation</i>	<i>FPGA and VHDL</i>	<i>Robotics, Robotic Arms</i>
<i>C Programming</i>	<i>GPS</i>	<i>Solid State Design</i>
<i>Cadence Pspice Simulations</i>	<i>LabVIEW</i>	<i>SolidWorks</i>
<i>Cadsoft EagleCAD</i>	<i>Mathworks Matlab and Simulink</i>	<i>System Stability</i>
<i>Digital Filter Design</i>	<i>Microchip PIC</i>	<i>Xilinx ISE</i>

EDUCATION & CERTIFICATIONS

California state university, Fullerton — Fullerton, California

In Progress:

Expected Completion: 2011

Masters of Science (MS) in Electrical Engineering with emphasis in Control Systems Engineering.

Completed:

Bachelor of Science (BS) in Electrical Engineering, August 2009.

Electrical Engineering + Physics + Computer Science: **GPA 3.15**

Certifications: **Comptia A+ Hardware Certification**, 2003

PROFESSIONAL EXPERIENCE

Coast Fulfillment Corp. — Ontario, California

2007 - 2009

Led a team to edit media and upload multimedia to an online ecommerce product catalog.

The logo studio. — Riverside, California

2005 – 2007

Moderated a team to edit media and upload multimedia to the logo product automation system.

Icarrots. — Lake Forest, California

2002 - 2005

Worked on a team to update the product database for the online prize catalog.