Build a dsPIC Oscilloscope and Spectrum Analyzer: Scopey II

by **vithrar** on December 24, 2007

Table of Contents

intro: Build a dsPIC Oscilloscope and Spectrum Analyzer: Scopey II

Once upon a time I had a nice, but old, Tek analog scope. I had to move, and thus had to move my junk, so I shipped the scope to where I was heading. Unfortunately, although packed in the box it was shipped to me in, it met an unfortunate end, and I now have a nice, but old, Tek analog scope with a dead CRT.
Despondent, I was reading through some issue of Circuit Cellar, and noticed that there was a contest involving dsPICs, and there were coupons for discount stuff! So why not, scrounge up some parts, read a couple datasheets, and a few months later, boom, a fairly useful little doodad that doesn't quite replace the old scope, but has some nifty new features that an old analog scope would be hard-pressed to reproduce.

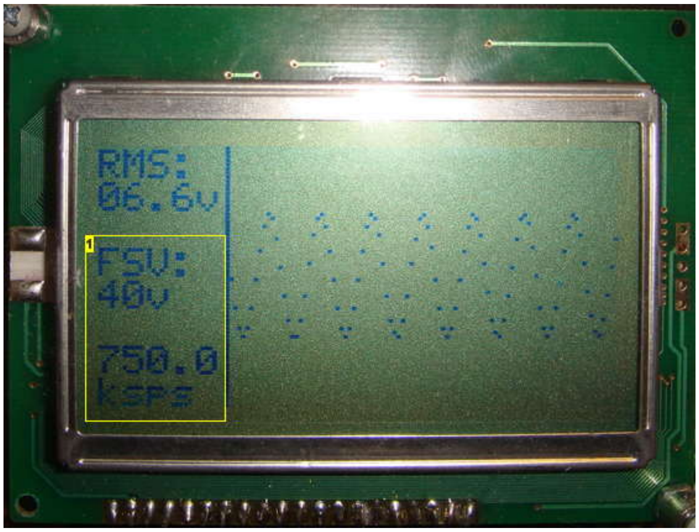


Image Notes
1. 55.555kHz sine wave on highest input voltage setting and fastest sampling

Image Notes
1. The Input Amplifier and Attenuator
2. Hey, look, you can see my fan!
3. Potentiometers for control knobs

File Downloads

scopey II schematic.pdf (334 KB)
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'scopey II schematic.pdf']

Scopey II.pdf (247 KB)
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'Scopey II.pdf']

step 1: The Big Picture

Here is a basic outline of how this thing works.
Everything comes in through the BNC jack on the front. The signal is then attenuated/amplified by the attenuator and amplifier. For the ultimate in excitement and because I dislike switches, the gain of the input stage is controlled by the PIC. The input stage also level shifts the input so it centers around 2.5v (half of full scale) to enable reading negative voltages. The front panel controls (potentiometers) are also read by the ADC. Finally, all the exciting info the PIC gathers is displayed on a handy 128x64 Graphics LCD.

The 'scope can sample an input at up to 750,000 samples per second allowing for signals up to 375kHz to be viewed (sort of). The RMS value of the input is displayed on the main Oscilloscope screen. The FFT function separates the input into 128 frequency bins, and displays the frequency of the bin with the highest amplitude.

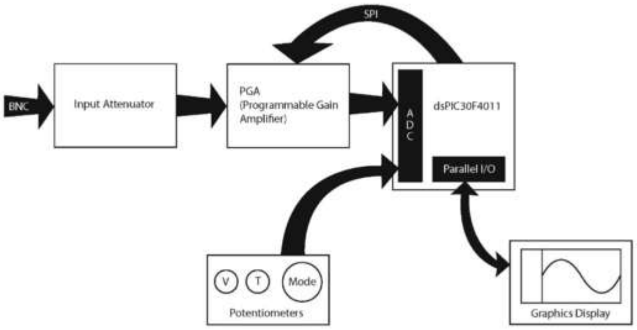
Here's all the stuff you need to make your own 'scope:
1x dsPIC 30f4011 (40 PDIP)
1x MCP6S21 (8 PDIP)
1x TLV2374 (14 PDIP)
1x BNC Connector
1x Transformer (like this one)
3x Linear Taper Potentiometers, around 1k-10k or so (and knobs, if you'd like)
9x 0.1uF ceramic capacitors
2x ~1000uF 35v electrolytic capacitors
2x ~1000uF 6.3v electrolytic capacitors
1x SPST or SPDT switch
1x DPST or DPDT switch
4x Rectifier Diodes
3x 10 ohm resistors
7x 10k ohm resistors
1x 10k trimmer potentiometer (the small blue square type fits the iron-on patterns)
2x 127k resistors (1% if possible)

1x 750k resistor (1% if possible)
1x 7805 or similar voltage regulator
1x 7905 or similar voltage regulator
1x 5-pin length of square header pins
1x IEC socket
1x Lumex LCM-S12864GSF Graphic LCD (or similar 128x64 GLCD with KS0107/KS0108 controller)
Some rainbow ribbon cable (easier to troubleshoot than grey)
Some single-sided Copper Clad PCB material if you plan on making PCBs

You'll also need some tools to put everything together:
A soldering iron
some solder
wire cutters and needle nose pliers
wire stripper (unless you're brutal like my middle school science teacher and use your teeth)

For PCBs and case making:
dremel or drill press and PCB drill bits (some normal sized drill bits are useful, too!)
a hand reamer

And, of course, you'll need some way to program your chip. I personally use the ICD2 from Microchip. I think ICD2 clones will also work on dsPICs.
You'll want MPLAB for creating all your future projects, too. Get it free!



step 2: Design an Input Stage

When designing an input attenuator/amplifier, there are a few things to consider.

1. Analog signals can have portions that travel below ground, while digital circuits can typically only handle positive voltages.
2. An ADC can't tell you about a signal that falls outside it's supply voltage range, so for this PIC we need to transform any input signal to be between 0v and 5v.
3. Sometimes little AC signals ride on big DC offsets, so AC coupling is a nice feature to have, but it is probably a bad idea to not also allow for DC coupling.
4. High input impedance is good

So, taking all of this into account, I came up with the input stage you see below.

First, there is a capacitor with a switch. If the switch is open, the signal must pass through the cap, therefore no DC can get through. If the switch is closed, the cap is shorted, and thus DC can get through. Next is a resistive divider, R9, R10, and R11. This attenuates the signal by about 1/8 (close enough) and gives an input impedance of about 1 Megaohm (close enough). Then we have a buffer, IC2a. Next, the level shifting, where R5 and R6 provide the 2.5 volts that will serve as "ground" for the ADC, and IC2c buffers this. IC2a does the actual shifting. Note that 2.5v appears across R7 and R8, which means that 1.25v appears at V+ (the non-inverting input to the op-amp). For an input of 0v (Ground) from the buffer, the op-amp will try to force both inputs to the same voltage, which means 1.25v at V- (the inverting input). To do this, it will put 2.5v on the output so that the feedback resistors (R3 and R4) will divide the voltage down to 1.25v at V-. Voila, 2.5 is the output for 0V input, and following similar logic you can see that an input of -2.5v will be bumped up to 0v and 2.5v will be bumped up to 5v. Magic!

This signal now feeds into the amplifier with programmable gain. Vref is connected to the 2.5v output of IC2c as well, so that this stage amplifies signals with respect to 2.5v. Recall that the input voltage divider gives an output that is 1/8 the input. The MCP6S21 from Microchip can be set to various gains, but I only used the powers of 2 so that the gains would come out as nice values. With gains of 1, 2, 4, 8, 16, and 32 from the MCP6S21, the total gain of the input can be set to 1/8, 1/4, 1/2 , 1, 2, and 4. This means that on the 1/8 setting, a 40V peak-to-peak signal will give max and min ADC values, and thus fill the screen (the largest input allowed) and at the 4 setting, a 1.25V peak-to-peak signal will fill the screen. Now we can hand this signal to the PIC and it can go and do some fun stuff.

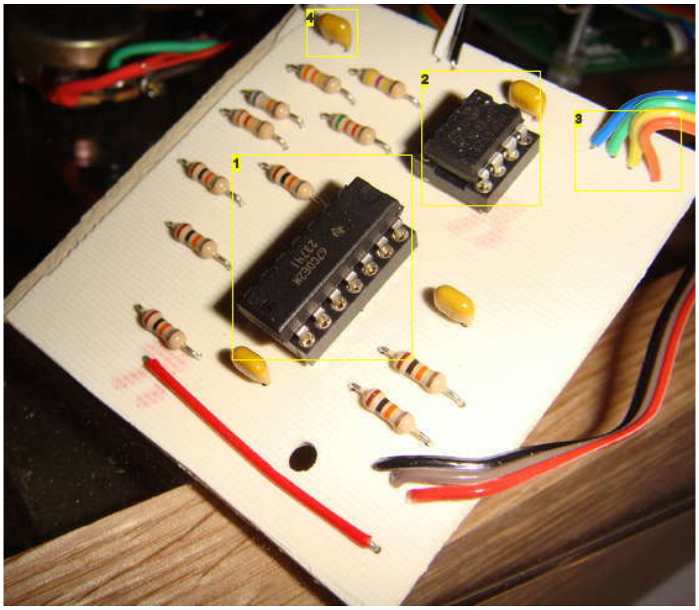


Image Notes
1. TLV2374
2. MCP6S21 PGA (programmable gain amplifier)
3. SPI to main board
4. a happy little yellow mono cap

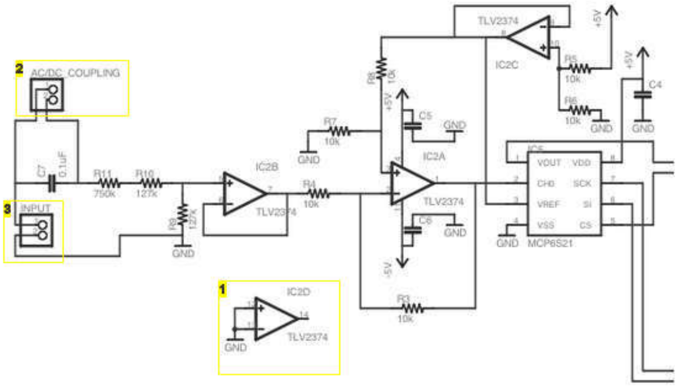


Image Notes
1. The fourth op-amp in the package, unused
2. goes to an SPST switch
3. Connects to BNC jack on front pannel

step 3: Make a power supply!

Taking note of the previous step, we need the following supply voltages:

1. +5v for the logic, as usual, as well as the positive rail of the op-amps
2. -5v for the negative rail of the op-amps

Being completely lazy and unimaginative, I used the usual center-tapped transformer with full-wave bridge rectifier to get positive and negative DC supplies, and used the standard 3-terminal 7805 for +5v and 7905 for -5v. Toss in some big caps for filtery goodness, and bam, power supply. You should probably throw a switch in there somewhere if you like to turn things off without unplugging them.

I originally had this bad boy running off the usual two 9v configuration for positive and negative voltages. This worked well except that the dsPIC and the LCD backlight use a lot of current. The batteries would die after only a couple of hours, so early on in the first prototype, I switched to wall power. I used a relatively large Radioshack transformer because that's what I had handy. Just about any center-tapped transformer will work, as long as it's high enough voltage and rated for probably at least 500mA. A non-center-tapped transformer will also work, if you modify the circuit a bit. I made a iron-on PCB pattern that works with a center-tapless transformer. You can also use a transformer with dual secondaries in order to effectively get a center tap.

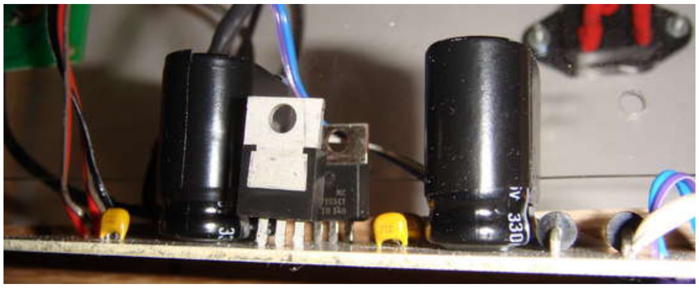


Image Notes
1. IEC connectors are your friend. Just make sure you insulate connections well to avoid death. Also, don't forget to get a power switch...

Image Notes
1. use some big electrolytic in parallel with a small ceramic mono cap. Values don't matter too much, the ceramic one should be around 0.1uF
2. Diode choice doesn't matter that much, just use what you have handy. Something rated 1 A or better is a good idea
3. 7905
4. 7805
5. Center tap goes to center pin

step 4: You can PIC your friends, and you can PIC your nose...

It seems like everyone these days is using AVRs for their projects, and I can see why, as I think they're great and have used them in several projects. Of course, at the time I used them I though "geez, it seems like everyone is using PICs in their projects..." I'm not a big fan of the 8-bit PICs, as I'd rather use an AVR, but these newfangled 16-bit PICs are pretty awesome. Just like the AVR, these chips have an instruction set designed for use with a C compiler, and in fact they are fairly similar. The bonuses include speed, an extra 10 MIPS or so, as well as 16-bit goodness, which makes assembly programming nicer, in my opinion, and probably makes C code a bit faster if you need to do things bigger than 255. The best part is the fast ADC, which can run at speeds up to 2Msps. Sort of...

Make sure you read the errata for a chip before you believe everything you see in the spec table...

Anyway, there is a good reason for choosing 750ksps as the maximum sampling speed...

Oh, and if you use a 30f/33f part, which this project does, you get some powerful math instructions that make crazy stuff like FFT (Fast Fourier Transform) run at reasonable speeds.

This entire project is written in assembly, so I'd recommend looking over the programmer's manual for the 30f series here.
There is also the ever-useful data sheet for the dsPIC 30f4011 which is at the heart of the project.

step 5: dsPIC + GLCD = almost a 'scope

This part of the design is pretty straightforward. First, check the pinout of the display you are using. I used the Lumex LCM-S12864GS, a 128x64 display. The code for this project will likely only work with 128x64 displays, mostly because that's what I had on-hand, but they seem to be pretty common and are cheap if you know where to look. I hooked up all the control pins willy-nilly to the dsPIC, arranging things so that most of the design can fit on a one layer PCB with as few jumpers as possible. All the stuff dangling from the LCD connector is according to its data sheet. The three three-pin connectors at the top of the schematic are for the potentiometers that control the scope. Each has its wiper connected to an analog pin. There is an ICSP 6-pin header, which is compatible with my hacked-up ICD2 cable, and possibly the PICkit2, although I don't have one so I don't know. The resistors and caps on AN0 and AN1 are according to the 30f4011 data sheet, as the fastest conversion rates require using an external voltage reference, in this case it's just the 5v supply. There are also some wires flying off the schematic, which go to the SPI port of the 6S21 programmable gain amplifier, as well as one for the output of the amplifier. In the picture you can see the ribbon cables that stand in for these black lines in real life.

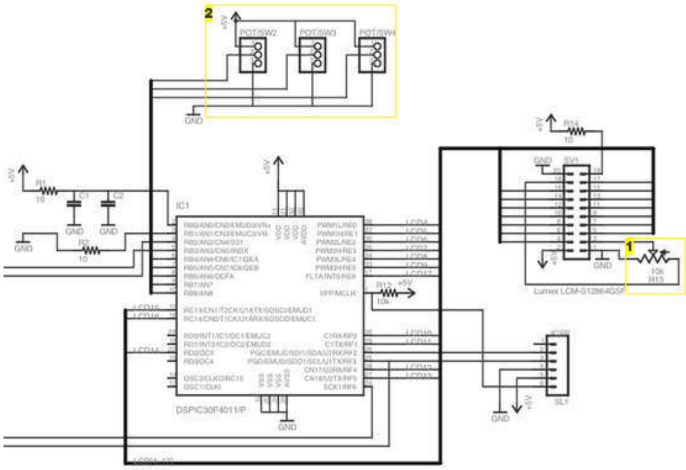
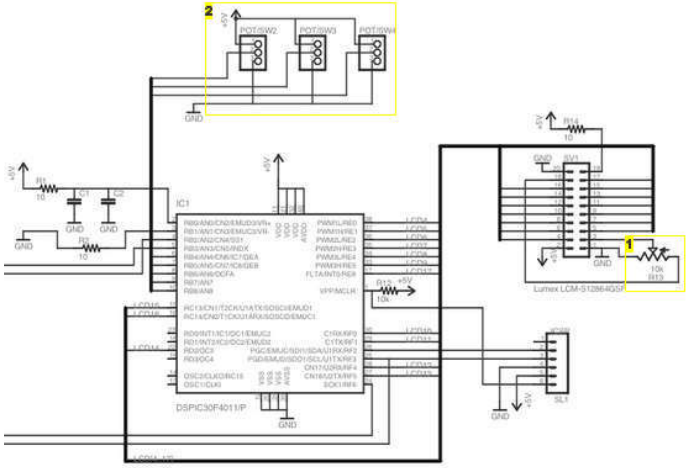


Image Notes
1. Pot for LCD contrast
2. Pots for 'scope control



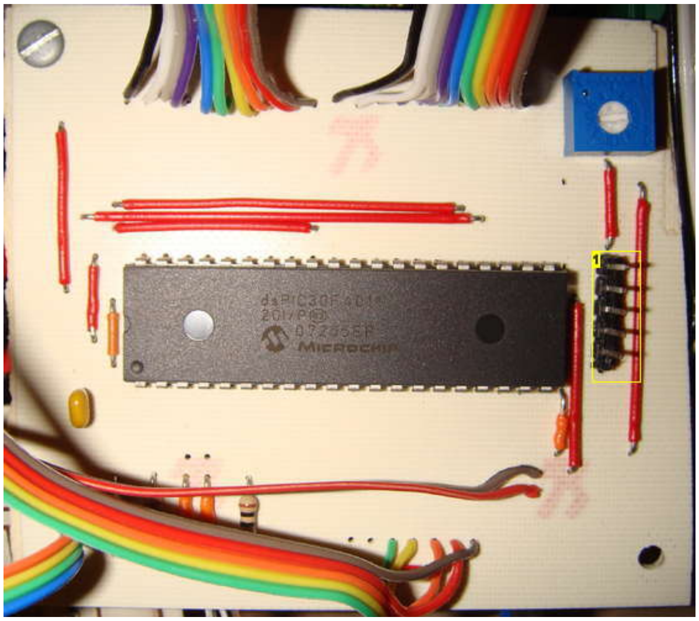


Image Notes
1. ICSP header

step 6: Code it!

This bad boy won't do much without some code to make it go. So get to it!

...

Just kidding. Have some code. It's a lot of assembly, with varying degrees of commenting, as well as varying degrees of actually being used. There is some code for a cursor that I never incorporated as I ran out of pots to control everything, as well as for an icon displaying mechanism that didn't make it into this version of the project. I'm working on a second version that will have multiple channels, as well as (possibly) higher sampling rates and maybe some sort of link to a real computer. Anyway, back to the code...

There are only a few interesting tidbits that I will point out, as there is a lot, and maybe (or not) it would be more instructive to look at the code and try to follow my crappy comments.

First off, the FFT is my favorite part. I used the wonderful DSP libraries available for free from Microchip to do this stuff. FFT is a complex subject, and there are lots of books on the subject, so I'm not going to bother with that. Let's just say it can be useful to tell you what frequencies your input signal contains. There are various memory location requirements for the use of these functions, so read all the documentation about the libraries, the C30 user's manual to figure out the calling conventions for assembly calling C (which are actually written in assembly) functions... blah blah... The code provided seems to work, so you can follow that, or come up with something else, or leave it out... rewrite everything in C, who knows?

Also worthy of mention is the True RMS meter incorporated into the oscilloscope display screen. It actually takes each sample, squares it using the MAC (multiply and accumulate) instruction and then takes the square root, for pretty good results. A 40-bit accumulator comes in handy for this sort of thing.

The software has the ability to set the trigger level at any "pixel" on the screen, and positive or negative slopes, but the interface right now is set for triggering at either positive slope at 0v or always trigger, where the scope will sample the wave whenever it would anyway, and takes those samples and displays them regardless of where the signal starts. The FFT triggers whenever it feels like it, although that could be changed fairly easily.

I think I spent the most time writing the Graphics LCD code, which is fairly self-contained and could be used on other projects. It involved hours of looking at the data sheet for the display and looking at other people's code for the KS0107/0108 controller.

Something worth mentioning is that there are two ways to run the screen, as I see it. You can keep everything stored on the screen's controller in your local memory, add and subtract pixels locally, and then send the bytes to the display. This requires 1k of ram if you use 1 bit per pixel, which seemed like a lot to me, so instead, on every write to the screen, I read a byte from the display, modify it, and write it back to the display. It takes a read and a write instead of just a write, but saves 1k of memory, which allow me to have enough room for a 256 point FFT. A good tradeoff, I think.

Perhaps if there is interest, I can write "Part 2: The Software". But for now, I'll just give everyone the software.

The code contains everything except the DSP libraries from Microchip. They are part of the install of MPLAB, so if you actually build this, you can just add them to the project.

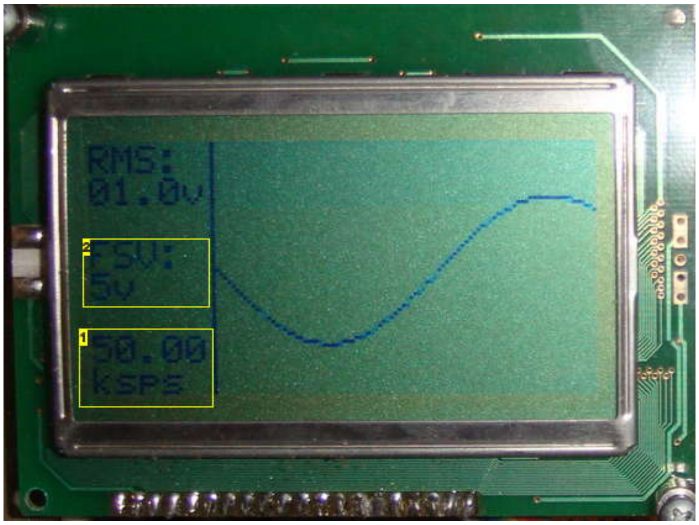


Image Notes
1. Not too shabby...
2. FSV = Full scale voltage, the voltage from the bottom of the screen to the top

File Downloads

source.zip (21 KB)
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'source.zip']

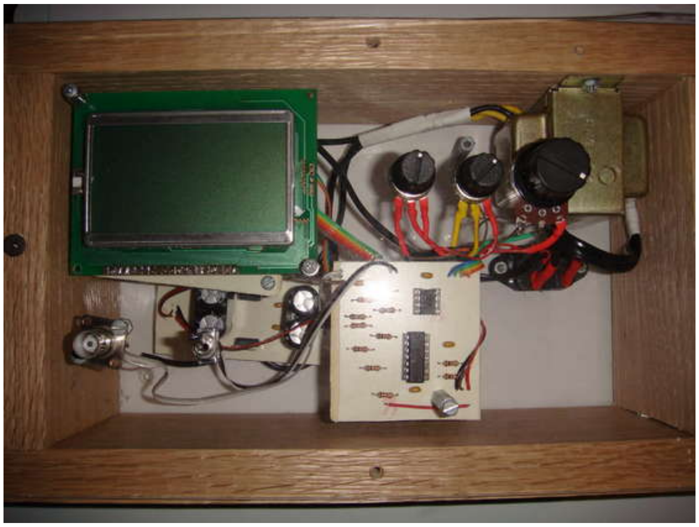
step 7: Done! Mostly!

So there you have it, a handy little scope that can sample at speeds up to 750ksps, which means you could theoretically view signals up to 375kHz, but that's kind of pushing it. It has a spectrum analyzer, too, which is something not every home built oscilloscope has.

I put it in a nifty box I made out of red oak and some 1/8" plexi from the local home supply depot, which fit the original nicely but now seems too big for the PCB version.

One improvement I'd strongly suggest is a power switch. I ran out of switches, ha, so there you go.

Here are some nifty pictures of the scope in action, with my trusty laptop standing in for a real function generator.



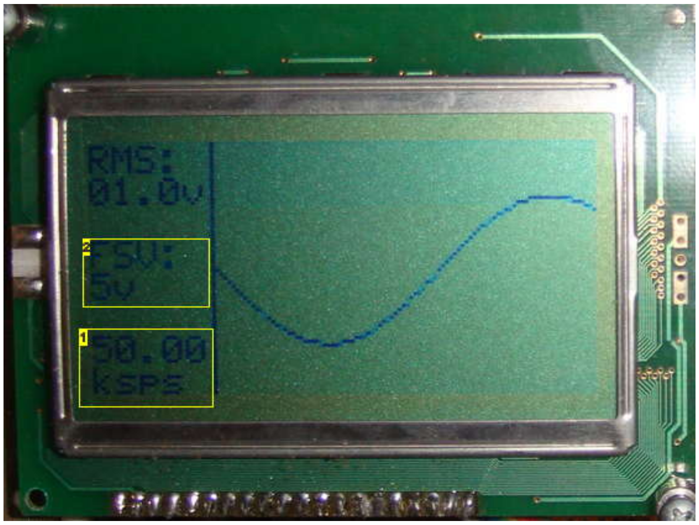


Image Notes
1. Not too shabby...
2. FSV = Full scale voltage, the voltage from the bottom of the screen to the top

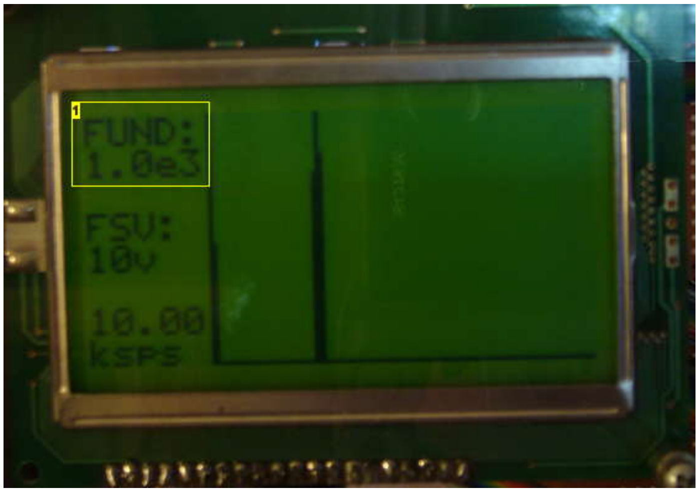


Image Notes
1. Spectrum of a 1kHz sine wave. Nice big spike and some DC crap at the left edge. FUND = frequency of largest spike (the FUNDamental)

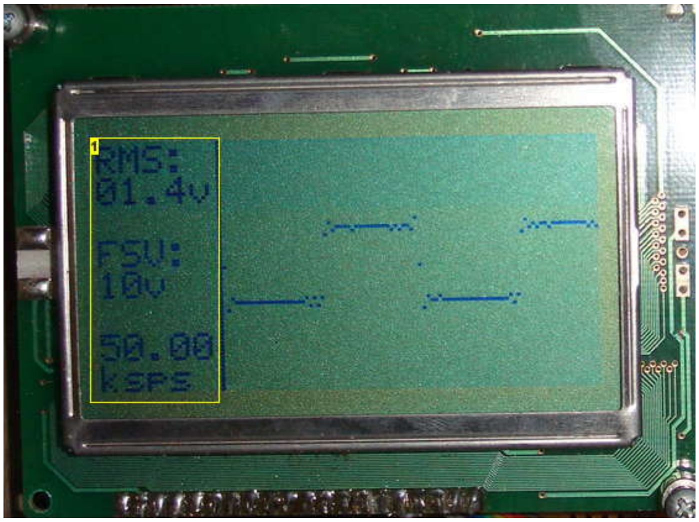


Image Notes
1. A 1kHz square wave

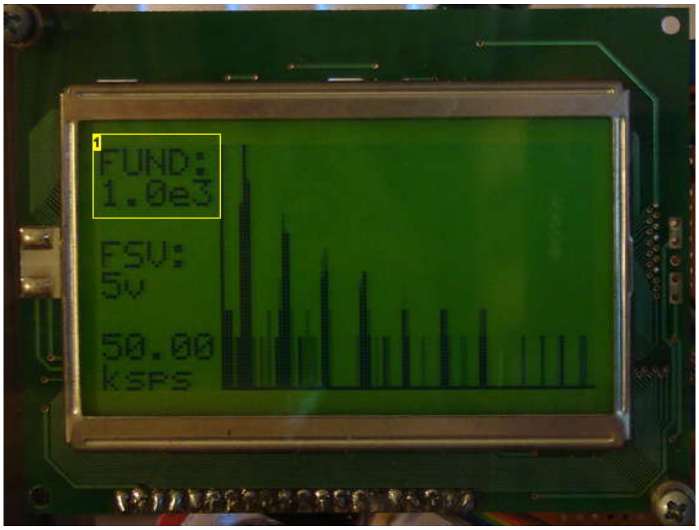


Image Notes
1. 1kHz square wave, with the noticeable odd harmonic spikes, and some ghostly spikes that appear and disappear...

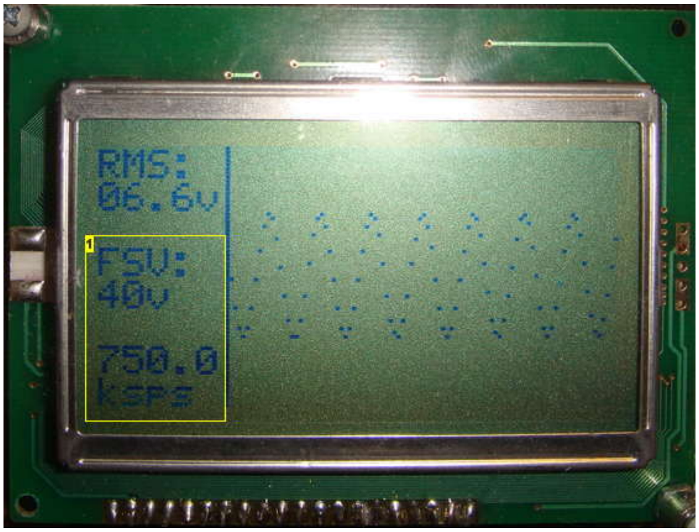


Image Notes
1. 55.555kHz sine wave on highest input voltage setting and fastest sampling

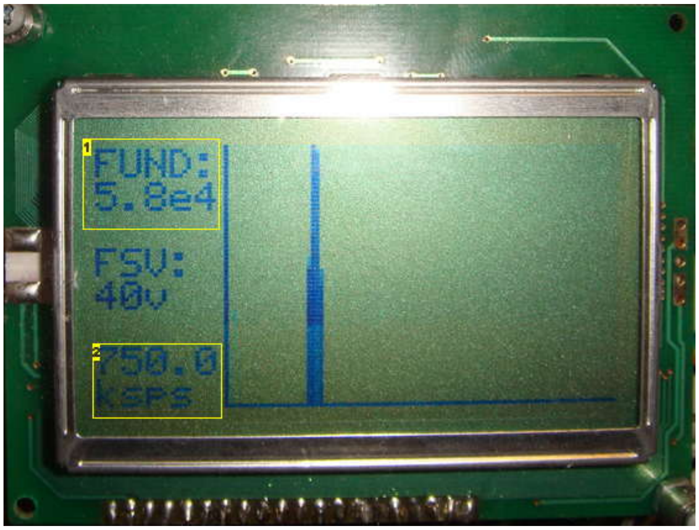
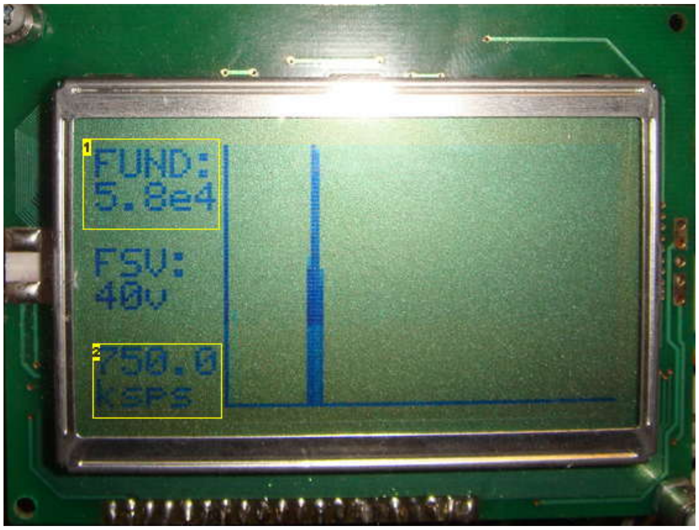


Image Notes
1. 55.555kHz Sine, so the frequency of the bin is a bit off
2. Fastest sampling rate!



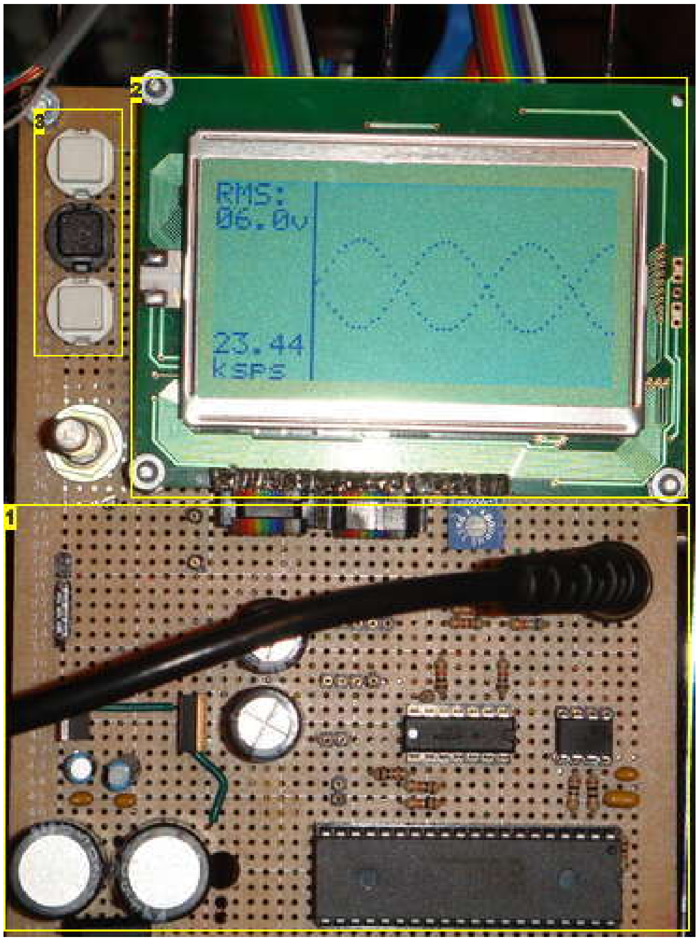


Image Notes
1. Original prototype on Radioshack proto board. Now lives on PCBs.
2. Under sampling a sine wave at one of the weird original sampling rates. Cool pattern. Also, thanks Nyquist!
3. Mysterious buttons that didn't make it to the final version

Related Instructables

Comments

29 comments Add Comment

soshimo says: Dec 1, 2008. 11:26 PM REPLY

Great project! Exactly what I needed and straightforward circuitry. The software is a work of art. One question though regarding the power supply. Could I use a wallwart jack and regulator circuitry with a wallwart supply instead of the onboard PS? I was hoping to prototype this before I etched it and am a little wary of having 120v AC laying around on bare poles. Thanks for a great weekend project though!

jrv_boots says: Feb 18, 2008. 7:38 AM **REPLY**

This is a great project to make!

I would probably make this one with a higher sampling DSP(2Msps), add a dual trace feature, and give it a larger display (320x240).

vithrar says: Feb 18, 2008. 10:47 AM **REPLY**

It actually uses the built-in ADC of the uC, and it was supposed to be 1Msps, but the "man" doesn't modify his data sheets when the errata informs you of the lower sampling speed. I was also thinking about a bigger screen and more channels, but addition screen real estate was unusually expensive compared to the $10 or so you can get a 128x64 screen for, and with additional channels in this design the sampling rate would be divided among the channels, so that the maximum sampling rate would go down with each additional active channel. But, if anyone can find 320x240 screens for cheap, or even color ones, let us all know!

2Msps is pretty wimpy, still. Microchip does make dsPICs with ADC that fast... unless you look at their errata. I'm working on a newer design that will sample at 30 or 40Msps and only be a bit more complicated than the current design.

jsprat2012 says: Nov 8, 2008. 12:07 AM **REPLY**

Okay, I have the printed circuit board layouts and photo etching ready to go.

I have the chips and all the parts.

Now, how do I make the software into something I can program into the PIC?

jrv_boots says: Feb 21, 2008. 5:29 AM **REPLY**

Yes, 2Msps is VERY wimpy, especially when other component can get 10Msps and better.

Microchip has some DSP controllers with 2 ADCs built in.

320x240 displays can vary in price with the high end in the $100 or so.

Any thoughts about adding in a USB port?

vithrar says: Feb 21, 2008. 10:25 AM **REPLY**

They do have some with two, but in order to get the fastest speed you have to configure the device so that one adc samples while the other is converting, so you're back down to one, effectively.

I definitely think a usb port would be sweet. I ordered a few 18F parts with USB, so maybe...

The Lightning Stalker says: Nov 6, 2008. 11:03 PM **REPLY**

My congrats to you for building this and doing all the programming in asm. It really is quite an achievement.

jsprat2012 says: Oct 25, 2008. 9:22 PM **REPLY**

The HEX code would certainly make like easier.

Who knows, maybe someone could actually build one, besides Vithrar, that is.

This construct with mystery code reminds me of Circuit Cellar. I purchased a "Build this" article only to find they do not have any printed circuit board artwork. Now who, in their right mind, is going to waste six months making, testing, and remaking a PCB for a "one-of" construction project?

Vithrar must be a Circuit Cellar fan. They do not have artwork and he only supplies mystery code.

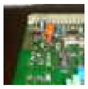
lieshamza says: Aug 7, 2008. 7:38 AM **REPLY**

Could you share the source please.
Yes, because it does not work with my version of MPlab.
Thank you in advance

subin.sam says: Jun 12, 2008. 12:21 PM **REPLY**

sir ,

can u provide us the hex code.



pbj says: Aug 1, 2008. 7:55 AM **REPLY**

Good morning!
Do You have the hex code? And others(Source code)?
Can you send me?

Thank you?

j9k says: Jun 18, 2008. 1:56 PM REPLY

just bought aan icd2 for the 33f devices and have been looking on the net for projects purely coded in assembly. everything is coded in C... even the examples from Microchip. anyway i want to learn assembly and this projects code has been a big help.

THANKS!!!!!!!!!

j9k

subin.sam says: May 30, 2008. 12:09 AM REPLY

sir
can i also use AGM1264F-FL-GTS-D

michaelmill says: May 17, 2008. 7:58 PM REPLY

HI.
I like this scope . But I have a question.
In the schematic you have lines RB4,RB5,RB,RB&,RB8 of the pic going to only 3 pot's ? And a parts placemeat for the pcb's Any help would be appreciated.

thank you michael

subin.sam says: May 14, 2008. 10:54 PM REPLY

sir
can i use LCM-H12864GSF

testor says: Apr 14, 2008. 6:41 AM REPLY

Very Fine! It looks like you have done the hardest parts of a portable oscilloscope. Perhaps you or another would look into adapting the circuit to work with a laptop through the USB port.

thegimpster says: Jan 20, 2008. 8:38 PM REPLY

Great project. I have been looking for a cheap o-scope to build. For those of you out there wondering wether to build it or not I have been doing some searching and you can get the 3 main chips for free. The two microchip ones you can get as free samples, and you can get the other chip from TI as a sample too. That just leaves the display to buy assuming you have or can scavange the other stuff.

vithrar says: Jan 26, 2008. 10:26 PM REPLY

free samples... coincidence? heh, probably not...

Also, for those interested, the displays can be had for about $20 with shipping from ebay, just search for 128x64. There are also some displays that will come close to the same price (with shipping, if that is all you order) at BGmicro.com, but I'm not sure they have a built-in negative voltage generator. That is something I forgot to mention, this design requires a glcd that has a built-in negative voltage generator, so keep that in mind if you want to use the boards.

mig says: Feb 3, 2008. 1:41 PM REPLY

Vithrar,

I agree on switched cap filters. The only source I have found is direct from Maxim. Maybe someone (like sparkfun.com) shoud re-sell them.

However it's not too hard to build a filter with a few op-amps.

I do respect the simplicity that you are after with this project. For me being able to store a reasonable 2 chanel trace and inspect (zoom, pan) would be great. My analog scope is no good for seeing non repeditive signals.

I know what you mean about screens, I have spent a long time looking for decent screens. There has been some good work done with monocrhrome OLED scopes that look good..

For those looking for a source for LCD's - www.futurlec.com (in thailand will ship international) have some reasonable prices. otherwise www.eio.com for thos of you in the states.

Anyway, just my thoughts.

- Mig.

thegimpster says: Jan 27, 2008. 8:09 AM REPLY

ya, that is the one that i was going to get. here is the spec sheet on it: http://docs.bgmicro.com/pdf/lcd1030.pdf it says that there is an optional neagtive voltage on board from the AZ display website, but I dont know if this one has the option or not. If I knew the full part number I could tell.

thegimpster says: Jan 31, 2008. 5:24 PM **REPLY**

I checked w/ BG MICRO and then AZ Displays and it will work, it does have the negative voltage generator on board. I have one ordered and it is in the mail now. Cant beat it for $0 bucks.

mig says: Feb 3, 2008. 4:06 AM **REPLY**

Hi, Firstly, good effort. I have often thought about something similar. However, you have failed to include a suitable anti aliasing filter on the input stage to the ADC's.

Whilst this is arguably not required for the scope function it is certainly needed for the spectrum analyzer. Without this you will se overlap in the FFT. Typically unless you are over sampling a lot (10+ times) you will need 6 or 8 order active filter (switched cap?).
Also, clearly the number of samples you can take here is not an issue. The update rate of the screen is the limiting factor. OLED Screens are better for replicating cathode ray.

Nice to see this sort of thing shared!

-Mig

vithrar says: Feb 3, 2008. 10:10 AM **REPLY**

I scoured various US suppliers, surplus dealers, and even ebay before I found an affordable display of any type, and any size, that was widely available, as that was one of my personal requirements for the project. I found that most OLED screens were too small physically, (almost always) too expensive, or their connectors were too fine pitched (and fragile) to be useful without an actual manufactured board.

As for an anti-aliasing filter, I thought long and hard before I decided to leave it out. It was a choice, and it is not necessarily needed for the FFT, and as I designed the scope for myself, I figured that I usually know what around what bandwidth and frequency my probing should reveal, so it's all good.

And don't get me started on switched cap filters. Good idea, but they're hard to find cheap or from the usual distributors, plus, just as an example, the highest corner frequency you can get from maxim for a low-pass switched cap filter is 50khz, which would be unfortunately low, I think. However, if anyone knows of a <$2 switched cap low pass filter that has a very low clock to corner ratio and a wide bandwidth, and is available from a normal US distributor, let me know.

4z4rb says: Jan 22, 2008. 9:48 AM **REPLY**

This is great !Tnx I will try to build it.

CyberBill says: Dec 26, 2007. 11:13 AM **REPLY**

Wow, this is a really awesome project! :)

The only thing I can recommend that you add is to draw lines from each sample so that the graph is fully connected.

burzvingion says: Dec 25, 2007. 12:38 PM **REPLY**

strong sauce! i wanted to build one of these a while ago, but ended up abandoning the project. For anyone who was wondering, yes, the pickit 2 is capable of programming the 30f4011. I may have to take a peek at your FFT routines when I get to building the 80x16 bicolor graphic LED (not LCD) display i'm working on.

very impressive.

BOOOM! says: Dec 25, 2007. 10:47 AM **REPLY**

wow ,2 micro controller based oscilloscope projects in 1 day what are the chances

dosadi says: Dec 25, 2007. 12:11 AM **REPLY**

Very nice! This is the best microcontroller scope I've seen so far. You even convinced me to check out the dsPICs. Thanks for sharing it...

GorillazMiko says: Dec 24, 2007. 10:03 PM **REPLY**

ahh, looks cool :-D