

# DPScope - Build Your Own USB/PC-Based Oscilloscope

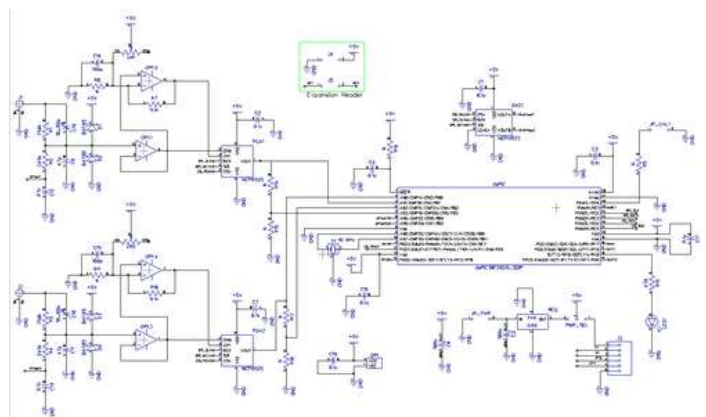
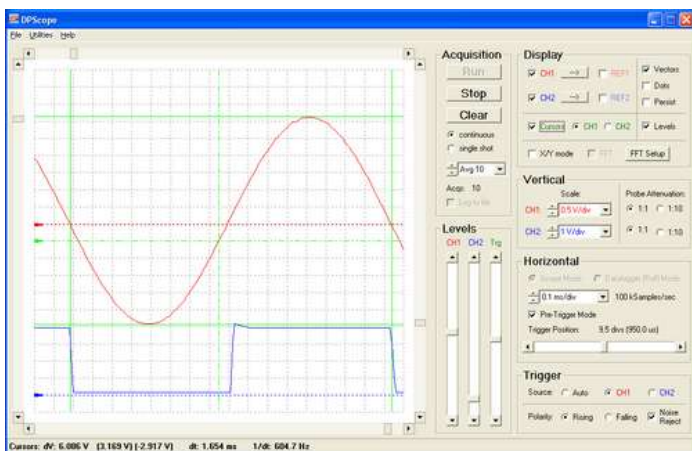
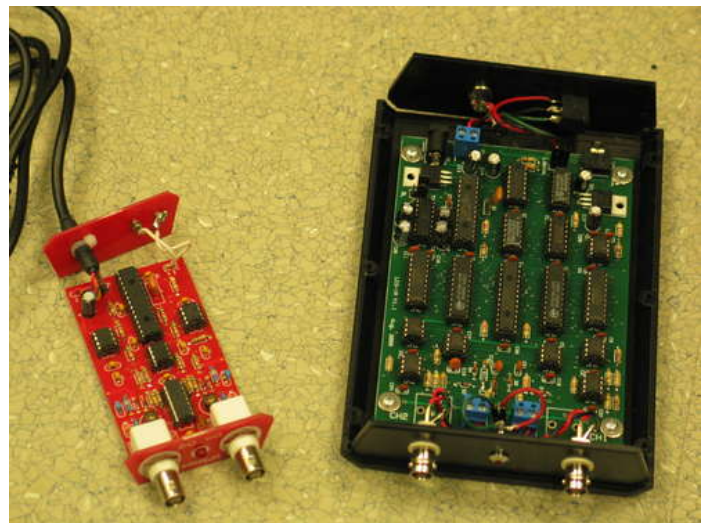
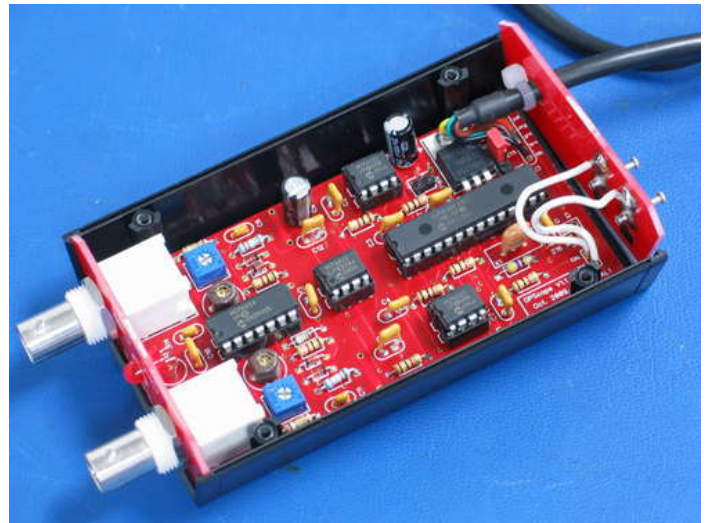
by [womai](#) on February 18, 2010

## Table of Contents

DPScope - Build Your Own USB/PC-Based Oscilloscope	1
Intro: DPScope - Build Your Own USB/PC-Based Oscilloscope	3
Step 1: Instrument Specifications	4
Step 2: Key Specs	5
Step 3: Design Overview	5
Step 4: Circuit Schematic	6
Step 5: Input Stage / Analog Frontend (part 1)	6
Step 6: Input Stage / Analog Frontend (part 2)	7
Step 7: Input Stage / Analog Frontend (part 3)	7
Step 8: Microcontroller	8
Step 9: USB Interface, Power Supply	9
Step 10: Board Layout	9
Step 11: Assembly - Tools	10
Step 12: Assembly Step 1: Unpacking the Kit	11
Step 13: Assembly Step 2: Components	11
Step 14: Assembly Step 3 - Component List	12
Step 15: Assembly Step 4 - Component Placement	12
Step 16: Assembly Step 5: Installing the First Part - R19	12
Step 17: Assembly Step 6: Soldering the First Part	13
Step 18: Assembly Step 7: Next Components	14
Step 19: Assembly Step 8: Remaining Resistors	14
Step 20: Assembly Step 9: Capacitors	15
Step 21: Assembly Step 10: Diodes	15
Step 22: Assembly Step 11: Trimmers, Resonator, Jumper	16
Step 23: Assembly Step 12: First Socket	16
Step 24: Assembly Step 13: Remaining Sockets	17
Step 25: Assembly Step 14: BNC Connectors; Frontpanel	17
Step 26: Assembly Step 14: BNC Connectors; Frontpanel	18
Step 27: Assembly Step 15: LED Indicator	18
Step 28: Assembly Step 16: IC Installation	18
Step 29: Assembly Step 17: USB Cable	19
Step 30: Assembly Step 18: Probe Calibration Output	20
Step 31: Assembly Step 19: Putting It All Together	20
Step 32: Assembly Step 20: Software Installation	20

Step 33: Assembly Step 21: Software Start .....	21
Step 34: Assembly Step 22: Offset Adjustment .....	22
Step 35: Assembly Step 22: Offset Adjustment .....	24
Step 36: Assembly Step 23: Probe Compensation .....	24
Step 37: Scope Software .....	25
Step 38: All Done! .....	26
Related Instructables .....	27
Comments .....	27

## Intro: DPScope - Build Your Own USB/PC-Based Oscilloscope



## Step 1: Instrument Specifications

Below you see the specifications of the instruments. If you are familiar with oscilloscopes you will see that the DPScope has pretty much all the features you'd expect from a decent lower-end instrument.

If you aren't a number freak, feel free to skip this page as fast as you can :-)

On the next page I'll discuss a few of the key specifications.

### Input:

Number of channels: 2  
Analog bandwidth: > 1.3 MHz  
Input impedance: 1 MOhm || 15 pF  
Probe connection: BNC  
Usable probe types: Standard 1:1, 1:10, 1:20 probes

### Vertical (voltage) scale:

Vertical sensitivity (20 divisions):  
- 5 mV/div to 1 V/div (1:1 probe)  
- 50 mV/div to 10 V/div (1:10 probe)  
- 100 mV/div to 20 V/div (1:20 probe)

Vertical offset: 0 - 20 divisions

Maximum voltage range  
-12V ... 20V (1:1 probe)  
-120V ... +200V (1:10 probe)  
-240V ... +400V (1:20 probe)

Probe compensation: yes (2 kHz calibration output)

Offset adjustment: yes

### Horizontal (time) scale:

Max. sample rate (single shot): 1 MSample/sec  
Max. sample rate (repetitive signals): 20 MSamples/sec

Timebase settings (scope mode): 0.5 usec/div ... 1 sec/div  
Timebase settings (datalogger/roll mode): 0.5 sec/div ... 1 hr/div

### Trigger:

Trigger source: CH1, CH2, auto (free run)  
Trigger polarity: rising edge, falling edge  
Trigger noise reject: yes (selectable)  
Pre-trigger capability (i.e. can show what happened before the trigger event): 0 - 20 divisions  
Post-trigger delay (delayed scan, to look at the signal long after the trigger event but with high resolution): 0 - 200 divisions

### Acquisition:

Record length (normal mode): 200 points/channel  
Record length (FFT mode): 400 points/channel  
Max. screen refresh rate: up to 40+ frames/sec  
Datalogger mode (roll mode): yes (data can be logged to file in real time)

### Display:

Real-time FFT: yes  
FFT filters: Rectangular, Hanning, Hamming, Blackman  
Averaging: yes (2 / 5 / 10 / 20 / 50 / 100)  
X-Y mode: yes  
Display styles (can be combined): Points, Vectors (Lines), Infinite Persistence

Time and level measurements: yes (using cursors)

### Save & Restore:

Waveform export (e.g. to Excel): yes (CSV format)

Save/restore of scope setups: yes

### PC Software:

PC connection: USB, 500 kbaud  
PC software: Windows 2000, XP (SP3), Vista, 7  
Minimum screen size: 800 x 600 pixel

### Mechanical construction:

Power supply: through USB (5V / 250mA)

<http://www.instructables.com/id/DPScope-Build-Your-Own-USBPC-Based-Oscilloscope/>

(external supply 7.5 - 9V / 300mA optional)

Approx. size (in enclosure): 4.5" x 2.6" x 1.2" (114 mm x 66 mm x 31 mm)

Component count: ~50

Solder connections to make: ~200

Required skill level for assembly: moderate; only through-hole components and DIP packages (no surface mount or fine pitch parts)

Printed circuit board: Professional printed circuit board with corrosion-resistant, gold-plated pads and contacts (not cheap solder finish), with silkscreen to denote component locations.

Enclosure: Sturdy ABS plastic enclosure with custom glass-fiber front- and back-panel, silkscreen. All holes pre-drilled - no drilling required.

Microcontroller and USB interface: Fully pre-programmed; no programming required

## Step 2: Key Specs

As promised, here a short discussion some of the DPScope's key features:

First, it is a two-channel instrument. This is a very important feature. Many low-end oscilloscopes offer only a single channel, which is a severe handicap: It doesn't allow you to look at two signals in relation to each other (e.g. clock signal and data signal), e.g. to see which one changes first and by how much. It also prevents you from triggering on a signal different from the one you want to look at. Thus I consider two channels an absolute must for any serious oscilloscope; everything else is a toy, not a real instrument.

Second, the bandwidth - the DPScope has about 1.3 MHz. While that may sound small compared to "big iron" scopes, it actually is quite usable for a large variety of tasks (in parentheses I show the approximate maximum frequency in the particular application):

- audio (20 kHz)
- infrared remote control signals (38 kHz)
- ultrasound (200 kHz)
- servo signals (a few kHz)
- bio signals, medical instruments (< 100 Hz)
- I2C (1 MHz)
- RS-232 (115 kHz)
- one-wire
- SPI (as long as  $\leq 1$  MHz)

The capture rate is a very important measure as well; it needs to be fast enough so ideally you instantly any changes on the signal or to the scope settings - this makes for a very responsive feel during practical use of the scope. Now that means it should at least be around 15 - 20 records per second (your eye isn't much faster than that anyway). The DPScope manages to do around 35 - 40 frames/sec (assuming a sufficiently fast timebase setting), so passes that criterion easily.

The DPScope also offers a datalogger mode (roll mode) for slow sample rates (between 10 samples/sec and 1 sample/hour); in that mode the waveform continuously scrolls to the left, and you can record it directly into a file. That's very useful to record slow-varying signals, e.g. temperature.

But now let's dive into the design, and start with some pictures!

## Step 3: Design Overview

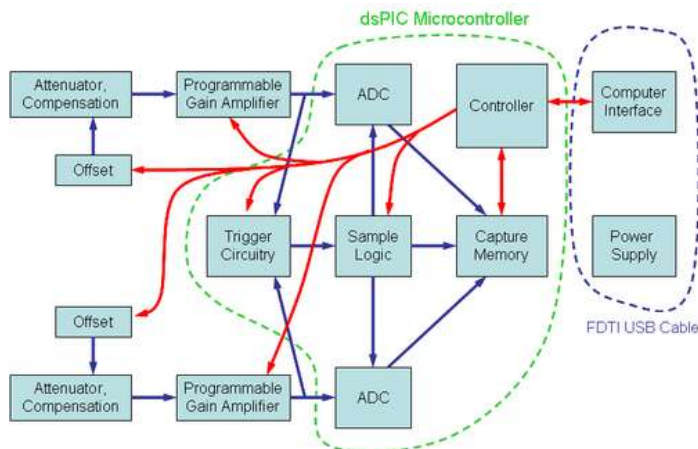
Below is a block diagram of the oscilloscope. The architecture is fairly traditional; the input signals are conditioned in the analog frontend which depending on the signal levels either attenuates them or amplifies them, and can also add offset. All this is necessary to make optimum use of the fixed voltage range that the analog-to-digital converters can convert into digital information.

The analog-to-digital converters (ADCs) take the analog signals and convert them into digital numbers. The sample logic controls the sampling process and the storage of the converted data in the capture memory. The trigger circuitry decides when to start the sample process.

The controller takes care of setting signal gains and offset, setting up the sample logic, selecting trigger source, trigger level, and trigger polarity, and communicating the the PC.

As you can see, the dsPIC30F2020 microcontroller allows to implement most of it without external circuitry - controller, memory, sample logic, analog-to-digital converter (ADC), trigger and trigger level control all reside within a single chip. This makes the design very compact, inexpensive, and easy to build.

USB interface to the PC as well as power supply for the scope are provided by the FTDI232R serial-to-USB converter cable - again a very user-friendly solution since there is nothing to assemble.

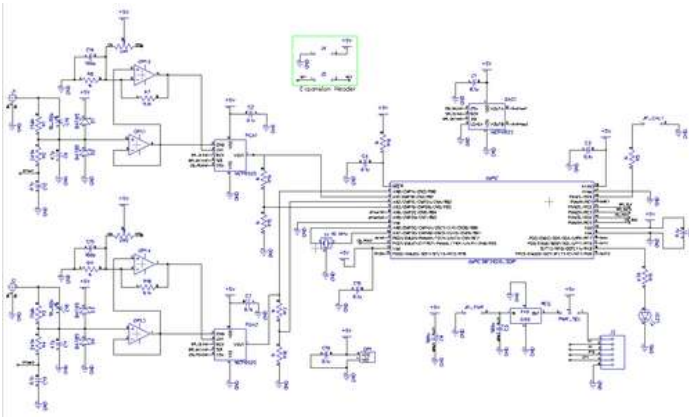




## Step 4: Circuit Schematic

Below is the full schematic. If you aren't an experienced electrical engineer it may seem daunting at first, but we'll break it down into easier-to-digest subsections in the following few slides.

If you want to get the schematic with better resolution (which is much easier to read), you can [download it in PDF format](#).



## Step 5: Input Stage / Analog Frontend (part 1)

The microcontroller's analog-to-digital converters (ADCs) have a fixed input range of 0 to 5V. Signals smaller than that range will have reduced resolution, and larger signals will get clipped. Since the input signal that the scope is supposed to measure can span a wide range from quite small to quite large, we need an input stage that can attenuate and/or amplify the incoming signal to make it suitable for the ADC. The circuit shown here is for channel 1, but channel 2 looks identical.

First, the incoming signal is attenuated by a factor of 4. This increases the maximum voltage range to 20V. Since the subsequent circuits cannot deal with negative voltage (to keep the circuit simple, the scope has only a single +5V supply and no negative supply), the only way to measure negative signals is to shift them up with a programmable offset voltage (between 0 and 4 V) provided by a digital-to-analog converter (shown later). This way the scope can display voltages between -12V and +20V max with a 1:1 probe (-120V to +200V with a 1:10 probe - but be VERY CAREFUL whenever working with such high voltages!).

The offset is fed in on the bottom of the voltage divider (between C12 - which buffers fast transients - and R2).

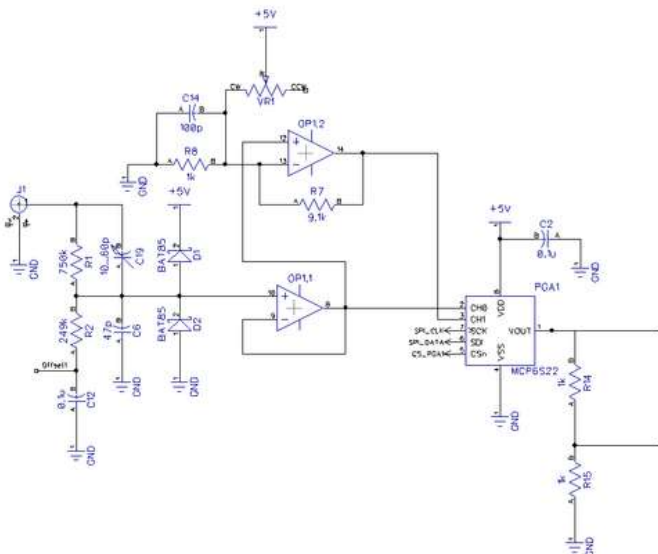
The input divider deserves some further consideration. It is a so-called compensated attenuator and consists of a combination of a fixed ohmic divider (R1 and R2) and an adjustable capacitive divider (C19 and C6). The reason for adding the capacitive divider is the fact that the protection diodes (D1 and D2) as well as the input of the op-amp (OP1.1) have some unavoidable parasitic capacitance in the order of a few pF. With only R1 and R2 this would create a low-pass R-C filter (the divider feeding the parasitic capacitance which would need some time to charge up), severely limiting the achievable bandwidth.

Quick estimate: output impedance of the divider is  $R1 \parallel R2 = 187 \text{ k}\Omega$ , parasitic  $C_{\text{par}}$  maybe 20 pF, that would give a time constant of  $187\text{k} * 20\text{p} = 3.74\mu\text{s}$  and a bandwidth of just  $0.35/3.74 = \text{approx. } 90 \text{ kHz}$ . This is much too low for our scope!

The solution - if you can't beat them, join them. Adding the capacitive divider and adjusting it to the same division ratio (1:4) as the resistive divider makes the frequency response flat from DC to light (at least in theory - but close enough for our purpose). The necessary condition is:

$$(C6 + C_{\text{par}}) / C19 = R1 / R2$$

Adjustment is done with C19. Since nothing comes for free in life, it's not surprising there is a price to pay - the capacitive divider causes the scope's input impedance to drop for higher frequencies. Still this is a worthwhile tradeoff and thus such a compensation can be found in virtually every oscilloscope.



## Step 6: Input Stage / Analog Frontend (part 2)

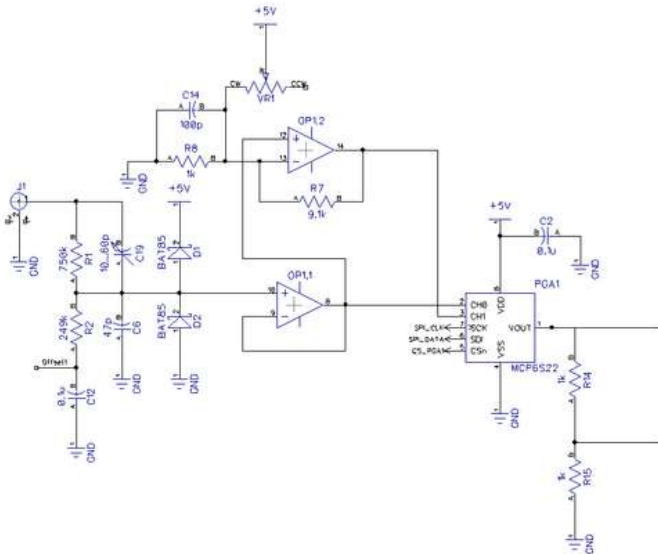
The two diodes (D1, D2) act as input protection, clipping any signals to the preamplifier that exceed either +5V or 0V by more than one diode drop. They need to be fast-switching (both in conduction as well as in recovery) and have low input capacitance. On the other hand, given the high input resistance value (750 kOhm) they don't need to shunt a lot of current even at large overvoltages at the scope input.

The signal is then fed into a simple op-amp follower stage (OP1.1, which is one of the four op-amps inside the Microchip MCP6024). This buffering is also necessary because the following stage (the MCP6S22 **p**rogrammable-**g**ain **a**mplifier or PGA) does not react kindly to an input source with too high an impedance - wild oscillations would be the result (yes, I tried and it is true!). The input divider's output impedance ( $R1||R2$ ) is around 187 kOhm while the PGA requires a source impedance of less than 1 kOhm.

The buffered signal drives one of the PGA's inputs (CH0) directly, and also feeds the input of a 1:10 gain stage that produces a signal amplified by 10, which in turn goes to CH1 of the PGA. That way the PGA can choose between less pre-amplification for large input signals, and large amplification for small signals. The PGA has a specified bandwidth (not gain-bandwidth product!) of between 2 and 12 MHz (depending on amplification setting), so we are in safe territory here; the scope actually uses only gain settings of 1, 2, 5, and 10 - according to my experiments higher settings (up to gain = 32 would be possible) are quite sensitive and tend to exhibit excessive noise (an indication that oscillation may not be far away).

The MCP6024 has a gain-bandwidth product of 10 MHz, which is more than sufficient for the buffer stage (gain = 1, so BW = 10 MHz), but marginal for the gain=10 stage (OP1.2) - we can only expect ~1 MHz of bandwidth here, and the other stages (buffer stage, PGA, ADC inside the microcontroller) will further reduce that number somewhat. For that reason I added C14 which increases the gain at higher frequencies. It is chosen so that the gain increase starts approximately at the frequency where otherwise the gain would start to drop off, that way the flat gain region is extended to higher frequencies. On my prototypes I measured a gain-stage bandwidth of around 800 kHz without this compensation but almost 1.3 MHz with C14 in place - quite some bang (50% improvement) at virtually no cost! Its effect is also clearly visible - much faster settling transitions - when using the scope to look at a fast-rising square wave. Ideally C14 would be adjustable, but its value is not overly critical so I stuck with fixed 100pF which was very close to the optimum I determined experimentally as well as by simulating the stage with Microchip's free Spice tool. If C14 were too larger, overshoot would occur.

The resistor trimmer (VR1) is here to allow minor offset adjustments in the high-gain path. The main reason for this is leakage current through the clamp diodes (D1, D2) which introduces a small positive offset onto the signal. This offset is small, but becomes noticeable when multiplied by 10. (VR1 also impacts the exact gain, but the effect is small enough to be ignored (less than 1%), especially when compared to the tolerances of the gain-setting resistors (R7, R8).



## Step 7: Input Stage / Analog Frontend (part 3)

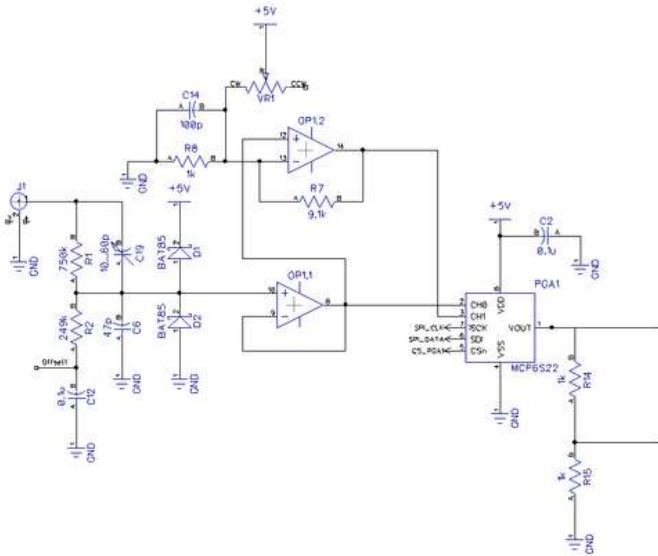
The signal now arrives at the previously mentioned programmable-gain amplifier (PGA), a Microchip MCP6S22. Such a PGA is a beautiful device because it allows us to change the channel gain (amplification) without any moving parts like relays. This reduces cost, component count, size, and improves reliability (no mechanical wear-out) at the same time. The only reason I can see why this isn't more widely used in other oscilloscopes is that there aren't many PGAs available for very high bandwidths (several 100 MHz or even GHz). But for the DPSScope with it's design goal of 1 MHz or maybe slightly above this is just what the doctor ordered - the PGA's minimum specified gain is 2 MHz.

Since the two selectable inputs of the PGA are fed with signals differing by a factor of 10, with the PGA we can effectively choose a total amplification of 1, 2, 5, 10, 20, 50, and 100 only use gain settings 1, 2, 5, and 10.

The microcontroller controls the PGA through a standard SPI interface consisting of three signal lines - clock, data, and chip select.

The output of the PGA feeds the analog-to-digital converter (ADC) inside the dsPIC microcontroller. It also drives into a second input which goes to a comparator inside the dsPIC. This comparator creates an interrupt whenever the input voltage crosses a certain, programmable threshold in a pre-defined direction (rising or falling signal level, respectively; this is also called a rising - or falling - edge). The threshold voltage is generated by a 10-bit digital-to-analog converter (DAC) inside the dsPIC (it's amazing how much peripherals Microchip has put inside a single, inexpensive microcontroller - all this reduces cost and complexity of the scope a lot). Since the DAC can only produce up to 2.5V (i.e. half the maximum signal level), the easiest solution was to divide the incoming signal by two with a voltage divider (R14 and R15). That way the trigger threshold can be set anywhere within the incoming signal range.

Finally, on the right you see the external 12-bit digital-to-analog converter (DAC, a Microchip MCP4822) which sets the offset for the input signal. It has two independent output channels, one used for each of the two scope input channels. The microcontroller drives the DAC through the SPI interface as well (the dsPIC uses the chip select (CS) line of the respective device - DAC, PGA1 or PGA2 - to determine which device is getting the SPI data at any given time).



## Step 8: Microcontroller

The Microcontroller is the heart of the DPSCOPE. It is a 16-bit Microchip dsPIC30F2020 which has a lot of desirable features: Not only is it blazingly fast (it runs 32 MIPS on a 128 MHz clock - this is very slightly outside spec - max. 120 MHz - but works perfectly fine) and comes in a hobbyist-friendly DIP package; since it is geared towards digital signal processing or DSP (that's where the "ds" in the name comes from) it also has a bunch of useful mixed-signal periphery already built in:

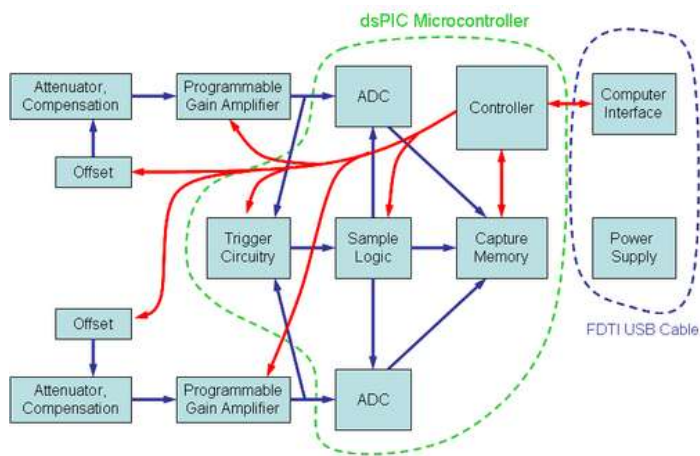
First and most important, it has an analog-to-digital converter that can acquire samples at 2 MSamples/sec. At least that's what the spec seems to say at first glance. The reality is - it can acquire two channels simultaneously at 1 MSample/sec each, and the Microchip marketing guys simply added those two numbers... anyway, that's perfectly fine for us because we need two channels anyway. The ADC has a resolution of 10 bits, but in order to maximize acquisition speed and reduce memory requirements only the upper 8 bits get used (i.e. one byte per sample). The reason for using a 128 MHz clock is the same - at that speed the dsPIC is just barely fast enough to repeatedly sample two channels and store the data into internal memory at 1 MHz sample rate.

The analog bandwidth of the ADC is well above 1 MHz, so it's not a limiting factor in the signal chain. For sample rates faster than 1 MSample/sec - up to 20 MSamples/sec - the DPSCOPE employs a technique called "equivalent time sampling" - basically it runs at 1 MSample/sec real sample rate and acquires only a subset of the data points at each sweep, and then overlays two or more subsequent sweeps (each with slightly increased start delay after the trigger) for a composite picture with higher effective timing resolution. You can read more details about this technique in the Tektronix application note "The XYZ of oscilloscopes".

The second great feature of this microcontroller is a set of comparators with finely controllable threshold (10 bit resolution); most other Microchip devices have only very coarse compare threshold steps. These comparators are all we need to implement a full-blown scope trigger with adjustable threshold and selectable edge polarity (rising or falling edge, respectively), which cuts down on overall component count and thus cost and complexity.

The only sore point with this dsPIC is its small RAM size - just 512 bytes. Some of that is taken up by program overhead (e.g. global variables, parameter stack and so on), and it was a challenge to get at least 200 bytes per channel (actually 205, since this works out to 410 points for FFT - where only one channel is acquired at a time - and 410 is 4/5 of the 512 points needed for the FFT, which makes interpolating it to 512 points fairly straightforward); a future version of the scope may use a different dsPIC device (but right now there isn't any that has all the other features, runs at 5V, and is available in DIP package). Adding external RAM is not an option either - first, it would add cost and complexity, second, the dsPIC does not have enough output pins to control it, and third, at maximum sample rate there is no time for additional control tasks anyway. But 200 points is good enough for a full display, and in most applications the DPSCOPE's delayed trigger capability provides exactly the same functionality that a longer capture memory would.

Finally, the dsPIC supports SPI and USART communication, which it uses to control the other devices in the DPSCOPE (adjustable gain amplifiers, offset DAC) and communicate with the PC, respectively.





## Step 9: USB Interface, Power Supply

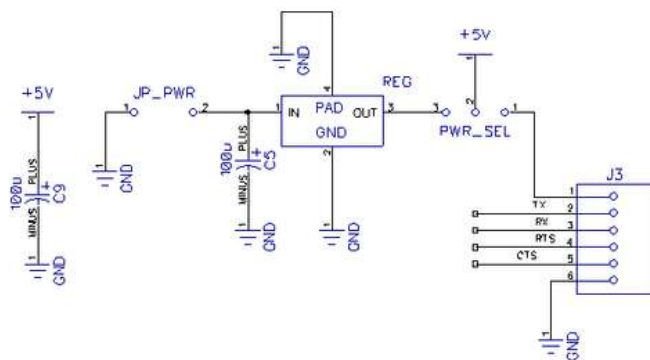
The connection to the PC is very straightforward. The DPSCOPE uses a USB-to-serial converter cable from FTDI. This cable has a FT232R chip built into the USB connector and translates the USB data stream into a standard RS-232 asynchronous serial data stream (output levels on the serial side are TTL, not true RS-232 - this is what the microcontroller needs anyway). That way the scope does not have to deal with a complex USB interface but sends and receives data as if the connection were a simple RS-232 link.

On the PC side there is a driver that emulates a RS-232 connection as well, so the scope application only has to deal with a standard serial connection, too. Data transfer rate is still a respectable 500 kBaud (the converter cable could go up to 1 Mbaud but the transmitted data volume is too small to really need full bandwidth - each record is approx. 0.5 KByte, so even at 40 frames/sec this amounts to just 20 KByte/sec or around 200 kBaud average serial data rate (note that each data byte needs a start bit and a stop bit, so for each byte actually 10 bits are sent).

The interface uses software handshaking (the scope responds to each transmission from the PC with an acknowledge packet), so while CTS and RTS are physically connected - just in case a later revision wants to use them - they are not used in the current design.

The converter cable also makes the USB port's 5V supply available. The DPSCOPE needs about 250mA of supply current, a USB port can provide up to 500 mA, so the scope can get its power from the USB line and does not need an external power supply. One word of caution though, some USB ports have rather large variation in supply voltage (sometimes down to little more than 4V). In my experience this is mostly true when connecting to an unpowered USB hub or when many other power-hungry devices are connected to the same USB hub, so try to avoid this. The DPSCOPE assumes a 5V supply and uses that as its voltage reference, so while it will function even at lower voltage, any deviation directly impacts its voltage accuracy. Some USB ports (laptops are notorious) are also quite noisy, this will show up as noise on the measured signals.

If you are absolutely unable to provide close to 5V from USB, or the USB supply turns out to be too noisy, then the DPSCOPE allows to use an external power supply (7.5V/500mA). All you need to do is add the regulator (REG, a simple 7805 linear regulator) and a buffer capacitor (C5), hook up the power supply, and set the jumper (PWR\_SEL) to "external supply". REG and C5 are very generic types and should be available in any hobby electronics store (and in any case are most likely already in your drawer).



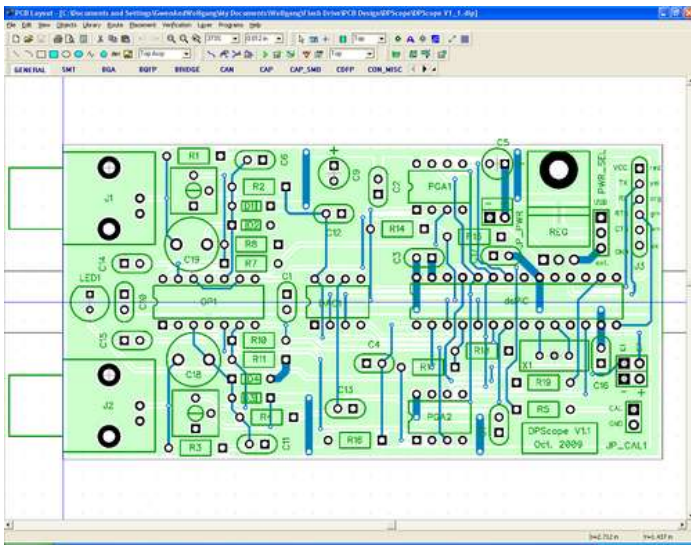
## Step 10: Board Layout

Once I had the design complete - including a working breadboard assembly of the prototype - it was time to lay out a printed circuit board for the scope. I did both the schematic capture as well as the board layout with a design tool called Diptrace . This is quite a capable tool that also comes with great customer support - the developers really listen to user feedback! - at a fraction of the price of so-called professional tools. In fact, they offer a free version that is restricted to two copper layers and a maximum of 250 component pins - which is actually more than the DPSCOPE layout needs (it has two layers but only about 200 pins). So you could do this design without paying a dime for the tool.

The final layout is shown below. You can see that I placed the components in a logical order: To the left, close to the BNC probe connectors, is the input circuitry (attenuator, input amplifier, clamp diodes, probe compensation). In the middle is the offset DAC and the programmable gain amplifiers, followed by the dsPIC microcontroller. On the right finally you have the USB cable connector and the power supply. All the chips are placed in the same orientation (pin 1 is bottom left, notch is left) to minimize the chance for assembly mistakes.

As for the trace layout, the ground plane is a solid copper fill on the bottom plane - this minimizes supply inductance and provides some shielding against interference. The power supply traces are made pretty wide (50 mil / 2.25mm) since they carry considerable current, and again to minimize inductance. With some exceptions the top layer carries the vertical traces, and the bottom layer the horizontal traces.

Now let's see how to build up this scope in practice!



## Step 11: Assembly - Tools

While the kit that is robust and simple to assemble, there are some components that are sensitive to mishandling, e.g. putting them in with wrong polarity, so please pay close attention to the description for every step. Moderate soldering experience is required (If you have never soldered before, there are many good introductions to be found on the web, and you should probably practice a little bit before attemptation to put together a larger circuit like the DPScope).

You will need a few tools for the assembly:

- Small soldering iron (about 17 Watts power) with sufficiently fine tip
- Solder wire
- Flat-nosed pliers (to bend component leads)
- Small wire cutter (to cut off component legs)
- 14mm wrench (to fasten the BNC connectors to the front panel)
- Small Philips screwdriver (to adjust the trimmer resistors and for the enclosure)
- Small non-metal screwdriver (to adjust the trimmer capacitors)

The time required for putting the scope together will depend on your experience – a seasoned hobbyist should be able to do it in maybe two hours (I do it in 45 minutes but of course I know the component locations by heart now), but if you are new to this it will take longer.



## Step 12: Assembly Step 1: Unpacking the Kit

Below you see the kit as it comes out of the box:

- Probe cables
- Enclosure, which holds all the small components
- FTDI USB cable

Unscrew the two screws on the bottom of the enclosure and carefully lift the bottom – make sure not to drop any components.

We don't need the probes and the USB cable right now, so put them to the side for the moment.

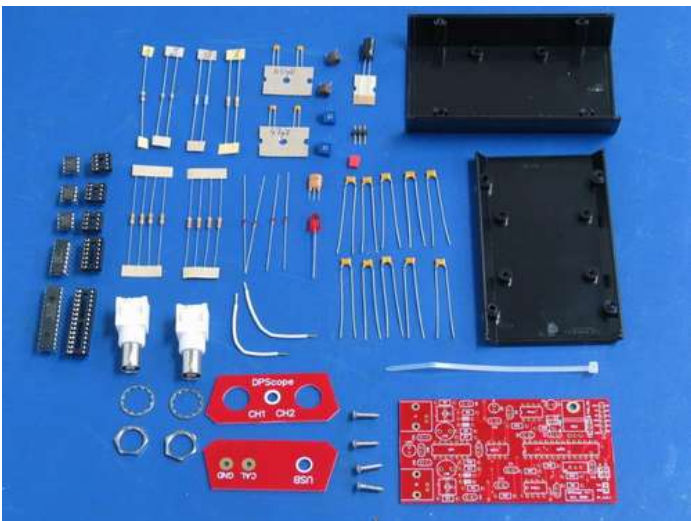


## Step 13: Assembly Step 2: Components

Below you see all the components spread out. Everything you need to build the oscilloscope is included. Please use the component list (on the next page) to verify that you have all the parts shown below.

Some general remarks:

- All components that are difficult to distinguish (e.g. resistors) are clearly labeled with their respective value (e.g. "100 pF").
- The integrated circuits and the diode are sensitive to electrostatic discharge – it is good practice to use a grounded wrist strap to avoid damage to them during assembly, and to place all the components on an antistatic surface. Don't wear clothes that get easily charged up (e.g. wool sweater).



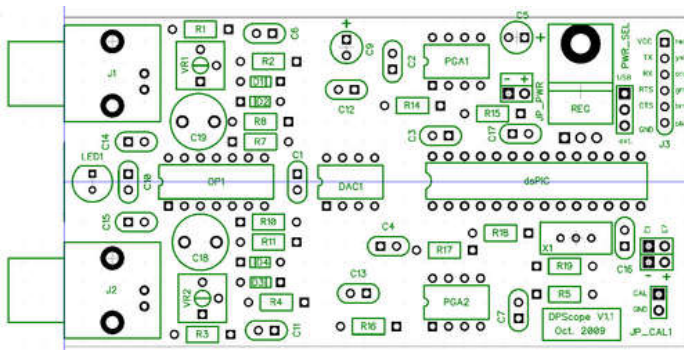
## Step 14: Assembly Step 3 - Component List

Below is the list of components - you can see that there really aren't too many. The microcontroller is already pre-programmed, so we'll only need to put it in and get going.

RefDes	Description	Value	Quantity
R19	resistor, 0.25W	470	1
R4, R6, R11, R14, R15, R16, R17, R18	resistor, 0.25W	1k	8
R7, R10	resistor, 0.25W	0.1k	2
R2, R4	resistor, 0.25W	24k	2
R5, R9	resistor, 0.25W	750k	2
VR1, VR2	trimmer, 0.25W	500k	2
C8, C11	ceramic capacitor	47p	2
C14, C15	ceramic capacitor	100p	2
C1, C2, C3, C4, C7, C10, C12, C13, C16, C17	ceramic capacitor	0.1u	10
C9	electrolytic capacitor	100u	1
C18, C19	trimmer capacitor	10, 60p	2
D1, D2, D3, D4	diode	1N514A	4
V1	ceramic resonator, 3 pin	16 MHz	1
LED1	LED	Red, 1.3kV	1
DAC1, PGA1, PGA2	DIP socket	8 pin	3
Q1	DIP socket	14 pin	1
uPIC	DIP socket	28 pin, 0.3"	1
DAC1	Integrated Circuit	MP948022	1
Q1	Integrated Circuit	MCP6024	1
PGA1, PGA2	Integrated Circuit	MCP6022	2
dsPIC	Integrated Circuit	dsPIC30F2020	1
	Terminal block for Cat & GND		2
PWR_SEL	3-way jumper		1
	Shunting block		1
J1, J2	BNC connector		2
	Tab & washer for BNC connector		2
	Enclosure		1
	Screws for enclosure		4
	Printed circuit board		1
	Front panel		1
	Back Panel		1
	Holding wire		2
	Cable ties		1
	USB to serial converter cable		1
	BNC-to-MicroBNC probe cables		2

## Step 15: Assembly Step 4 - Component Placement

Below is a picture showing the component placement on the board. This is the same as the layout picture, but with only the top silkscreen marking visible. Such a print is useful to locate the different component positions during assembly.

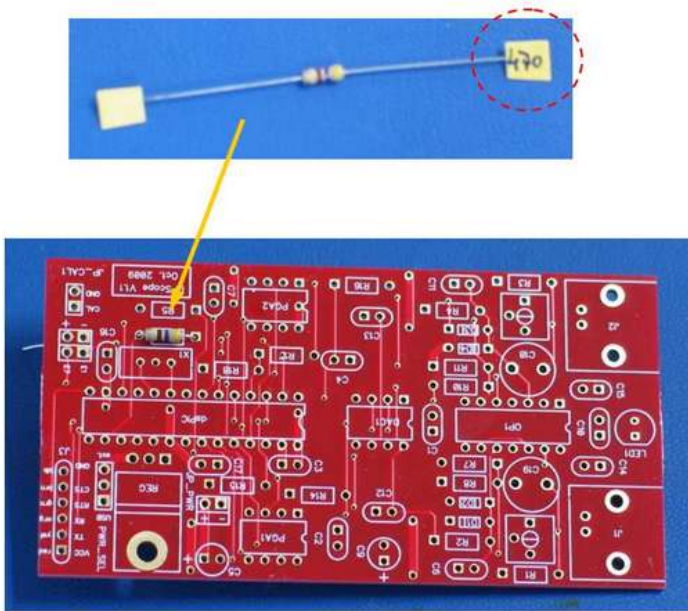


## Step 16: Assembly Step 5: Installing the First Part - R19

Let's start out with a simple part – resistor R19, which has 470 Ohm. All the other parts will get assembled in a very similar way.

Find the resistor among all the parts – it is labeled with “470” (see picture below). On the printed circuit board (PCB) locate the component outline labeled “R19”. Bend the resistor leads and stick them through the board. The final result is shown below.



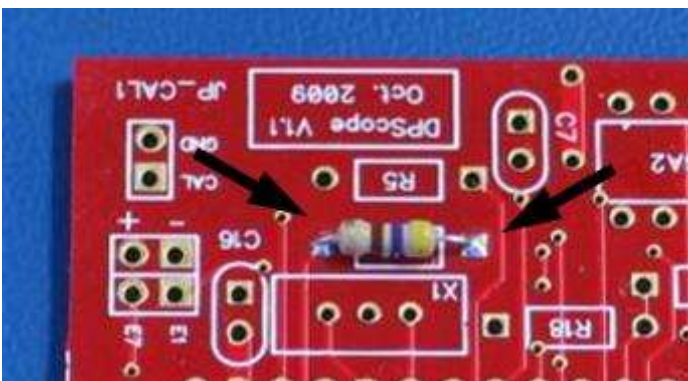
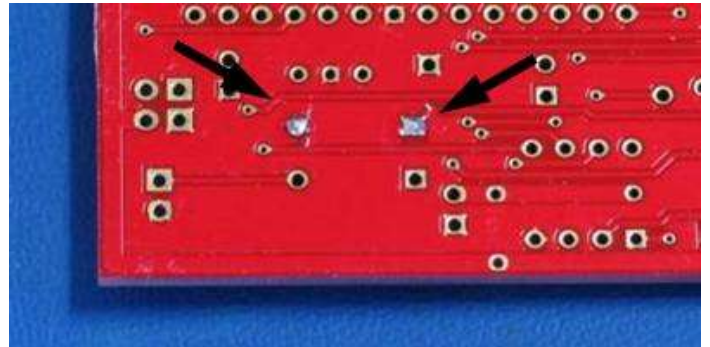
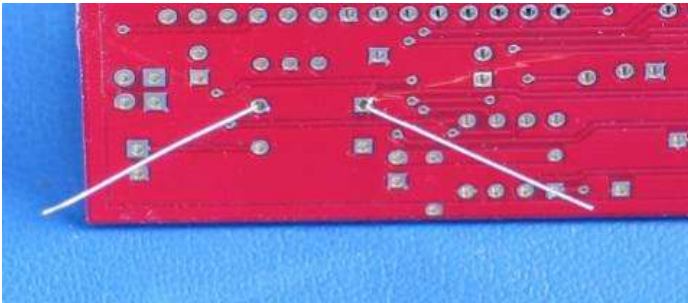


### Step 17: Assembly Step 6: Soldering the First Part

On the PCB bottom bend the component leads apart – this will hold the resistor securely when you turn over the PCB to solder it on.

To solder, touch the component lead (wire) with the soldering iron and the solder wire at the same time. Surface tension will pull the solder into the via holes automatically. Add enough solder so it completely fills the hole and leaves a small “hill” of solder. Remove the solder wire and the soldering iron but don’t move the board before the solder has become completely solid again. The solder joint should be shiny and bright. Repeat for the other lead.

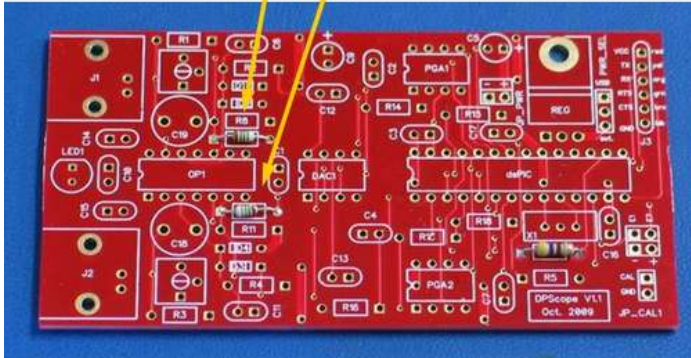
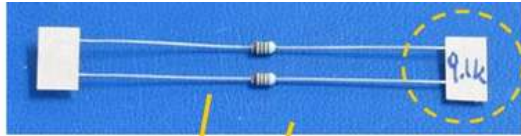
Check the topside of the board – you should see solder protruding a bit from the via holes – this shows the holes are nicely filled with solder (see picture below):





### Step 18: Assembly Step 7: Next Components

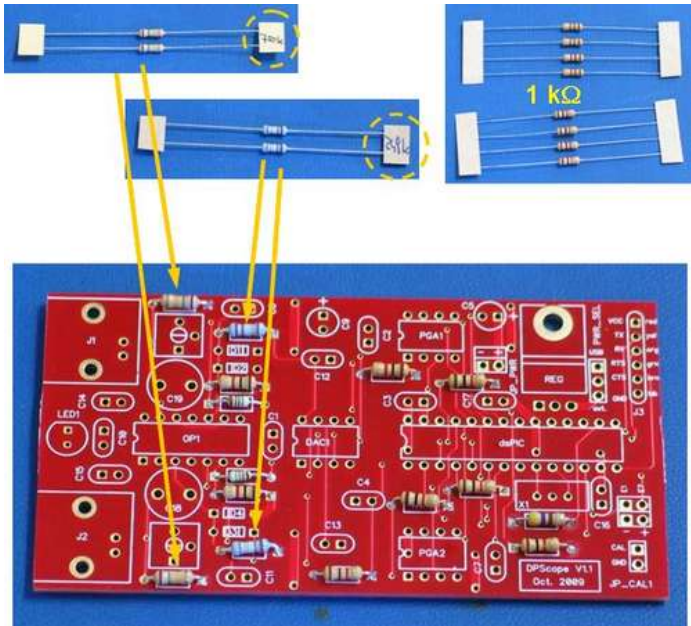
Next is a pair of resistors – R7 and R10 (the 9.1 kOhm resistors, thus labeled with “9.1k”). Installation and soldering works just like for R19. Below you see what the board looks like with them installed.



### Step 19: Assembly Step 8: Remaining Resistors

Now let's finish up the resistors. We still got R2 and R4 (249 kOhm), R1 and R3 (750 kOhm), and 8 pieces of 1 kOhm resistors which are not labeled because they are the last resistors left. Install the two pairs first, and then install the 1 kOhm resistors in all remaining resistor outlines on the PCB (refer to the parts list if you find out which resistor is which value).

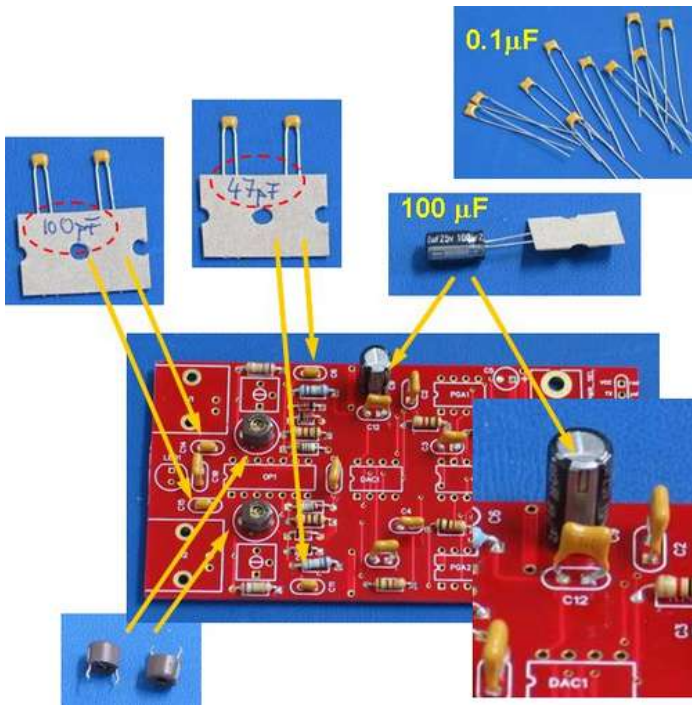
The picture on the bottom shows the PCB populated with all the resistors.



## Step 20: Assembly Step 9: Capacitors

Installing the capacitors works the same as for the resistors. We have 5 different types: C6 and C11 (ceramic, 47 pF), C14 and C15 (ceramic, 100 pF), C9 (electrolytic, 100 uF), C18 and C19 (trimmer), and 10 pieces of 0.1uF ceramic capacitors (not labeled because they are the only type left).

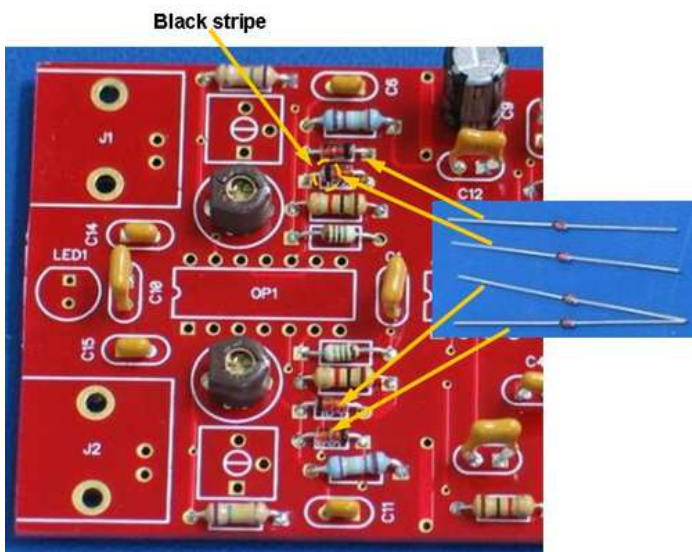
**Note: Be careful when installing C9 – it has to be put in with correct polarity.** The negative side of C9 is clearly labeled with a white stripe and “-” (minus) symbols – make sure you install it as shown in the pictures below with this white strip facing inwards on the PCB!



## Step 21: Assembly Step 10: Diodes

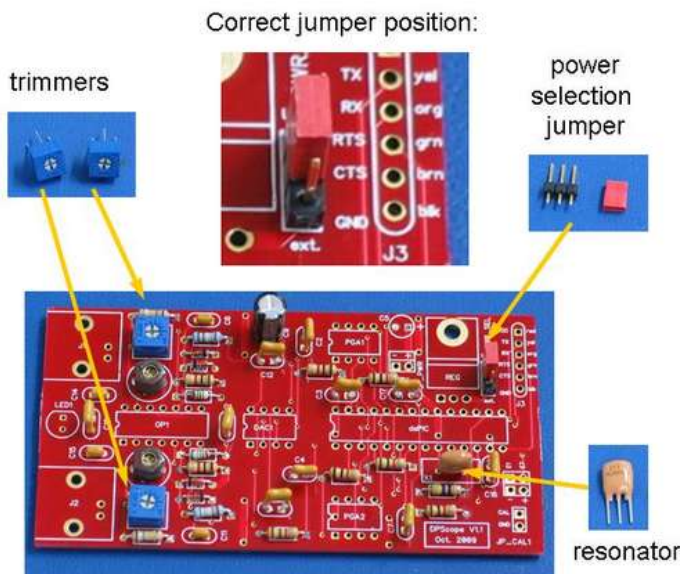
Next in the row are the clamping diodes. These components need more care than the resistors.

- Make sure you insert them with correct polarity (correct orientation); the negative end is denoted by a black stripe around the diode body. The silkscreen outline also shows a (white) stripe – this is the side the black stripe must lie. Note that the orientation is not the same for all diodes.
- Diodes are quite sensitive to heat. Thus try to minimize soldering time. The best approach is to first solder only one end of all diodes, then the other end – this gives them enough time to cool down in the time between.



### Step 22: Assembly Step 11: Trimmers, Resonator, Jumper

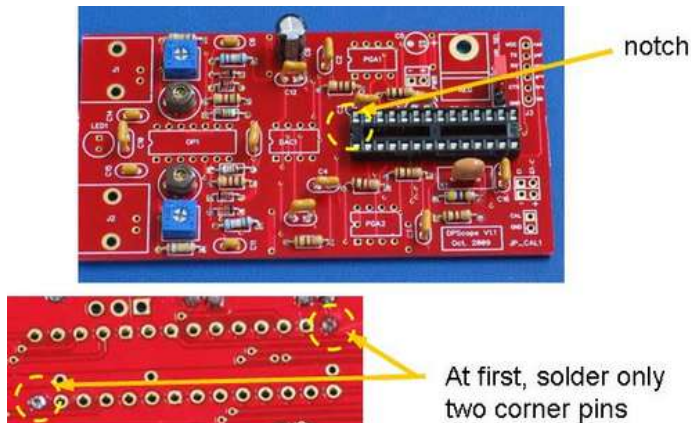
A few more small components are left: The two trimmers (VR1 and VR2, the blue blocks with the adjustment knob on the top), the ceramic resonator (X1), and the power selection jumper (PWR\_SEL). Install them as shown below and **put the red shorting block on the jumper in the position indicated in the zoomed-in picture (labeled "USB")**; this jumper is used to select the optional external power supply & voltage regulator as the scope's power source; per default these are not installed because the instrument gets its power through the USB connection.



### Step 23: Assembly Step 12: First Socket

Now comes the first socket – let's start with the 28-pin one, which goes into the outline labeled "dsPIC" and will later hold the microcontroller. Be careful with its installation because once soldered down it is almost impossible to remove. The best is to do it step by step:

- (1) Place the socket on the board as shown in the big picture. Note the **position of the notch** on the left side of the silkscreen outline – make sure to orient the notch on the socket to the same side. This will make installing the chip less error-prone.
- (2) Turn the board around and **solder only two of the corner pins** as shown in the zoomed-in picture below. The reason is simple – two pins diagonally opposed will securely hold the socket in place, but still allow you to make corrections.
- (3) Press the socket onto the board and re-heat both of these solder joints – this allows the socket to sit flush against the board. Visually inspect the socket to make sure this is really the case.
- (4) Only now solder all the other pins.

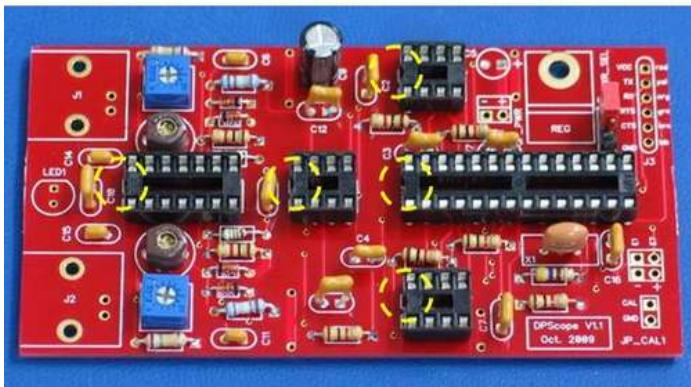
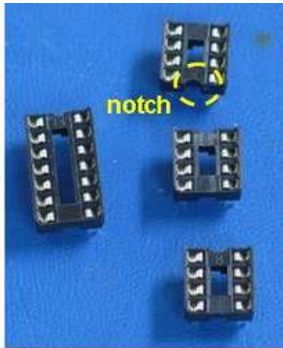




### Step 24: Assembly Step 13: Remaining Sockets

Now install the remaining sockets – one 14-pin and three 8-pin ones. Proceed in the same manner as for the first socket. **Again make sure all the notches match the silkscreen outline** (they are all on the left side in the picture).

Below you see how the board should look like after this step.

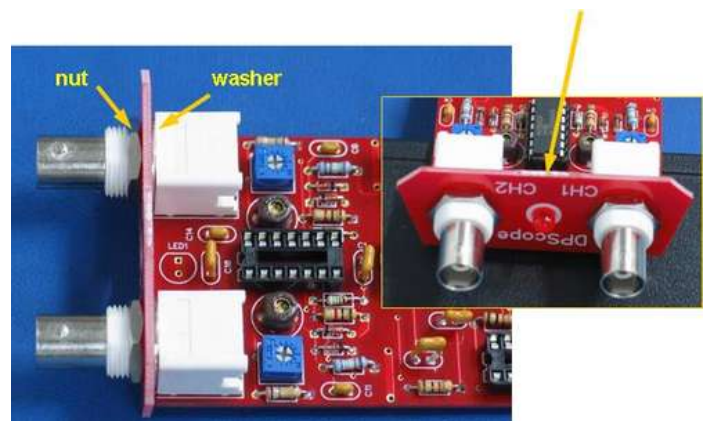


### Step 25: Assembly Step 14: BNC Connectors; Frontpanel

Snap the BNC connectors in place. **Don't solder them down yet!**

Add the washers, the frontpanel, and finally the nuts. Tighten the nuts by hand (not too strongly, just so the frontpanel no longer moves freely).

**Note that the frontpanel's narrow end – close to the "DPScope" label! – is on the bottom (PCB) side, and the wide end – close to labels "CH1" and "CH2" – is on the top (away from the PCB).**



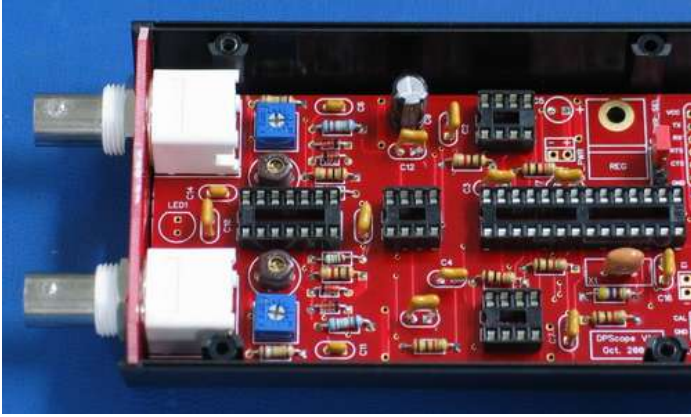
### Step 26: Assembly Step 14: BNC Connectors; Frontpanel

Place the board with BNC connectors and frontpanel into the enclosure as shown (into the deep half of the enclosure).

Make sure the board fits and sits loosely on – but does not push too hard against – the standoffs inside the enclosure. If necessary loosen the nuts a bit so the connectors can move against the frontpanel.

Tighten the nuts again sufficiently to hold the BNC connectors in place.

Only now solder the BNC connectors onto the board – start with the clamped-in feet, and finish with the signal wires. The clamped feet will need a lot of solder to fully fill up the mounting holes – don't be shy, because that's the only thing holding the PCB in place.

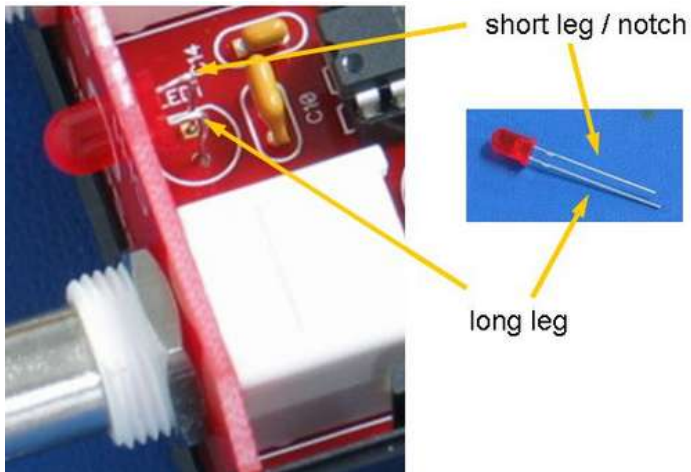


### Step 27: Assembly Step 15: LED Indicator

Remove the frontpanel again. Take the LED (light emitting diode) and bend its legs by 90 degrees as shown in the picture. Fit it into the board. **Make sure the short leg of the LED goes into the side where the silkscreen circle has its notch (flattened side).** The diode body also has a notch at the same side.

Now put on the frontpanel – the LED must go through the center hole. Put on the nuts again and tighten them down with a wrench. Be careful not to over-tighten them, damaging the connectors threads!

The frontpanel now holds the LED securely in place. Turn the board around and solder the LED's legs onto the board.



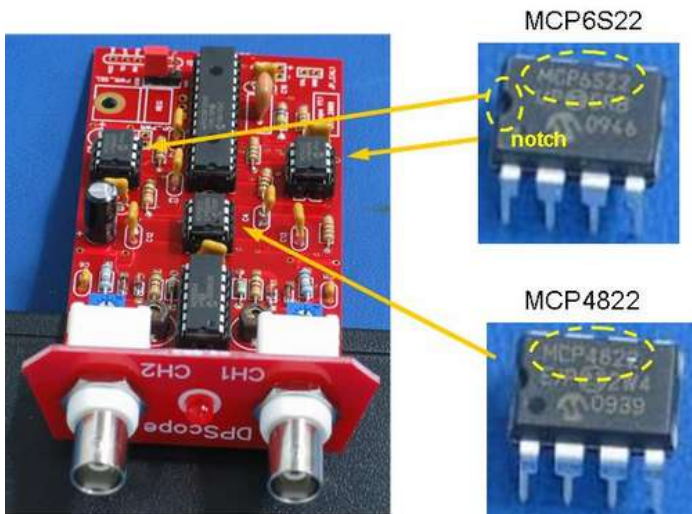
### Step 28: Assembly Step 16: IC Installation

Now it's time to install the ICs (integrated circuits). There are five of them.

**Note 1: Make sure to put the chips on in the correct orientation.** Each chip has a notch on one end – this notch must go on top of the notch in the silkscreen outline (and the notch in the socket if you installed those correctly!).

**Note 2: There are two different types of the 8-pin ICs – make sure you install them in the correct locations as shown in the picture below.** Installing them in the wrong place (or the wrong orientation) will destroy them when you power up the oscilloscope. You can distinguish the chips by the labels printed on them (two are labeled MCP6S22, one is labeled MCP4822).



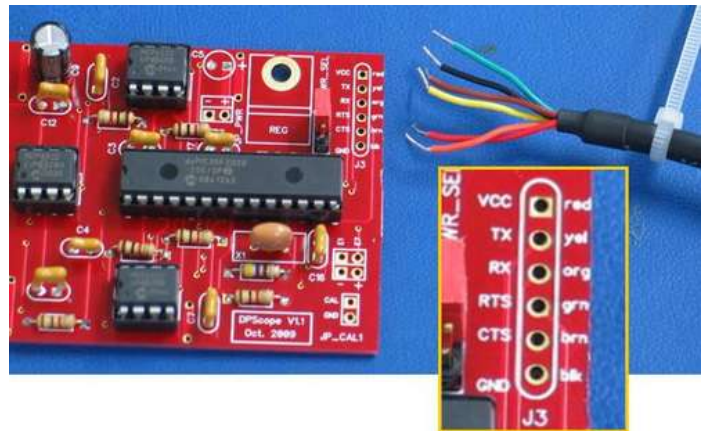
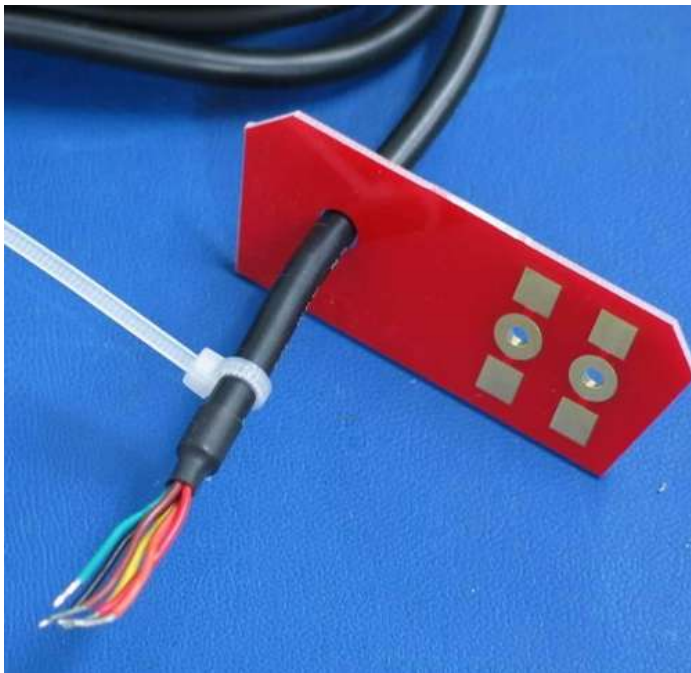


### Step 29: Assembly Step 17: USB Cable

Take the USB cable out of its antistatic bag and feed its wire ends through the hole in the backpanel plate. Secure the cable binder tightly around the cable – this will act as a stress relief so the cable can't pull on the solder joints later. Snap off the protruding part of the cable binder.

The cable has 6 wires in 6 different colors. Solder them into the respective hole of J3 (all holes have labels indicating the proper color, as shown in the zoomed insert). The sequence from top to bottom is:

Red (red)  
 Yellow (yel)  
 Orange (org)  
 Green (grn)  
 Brown (brn)  
 Black (blk)

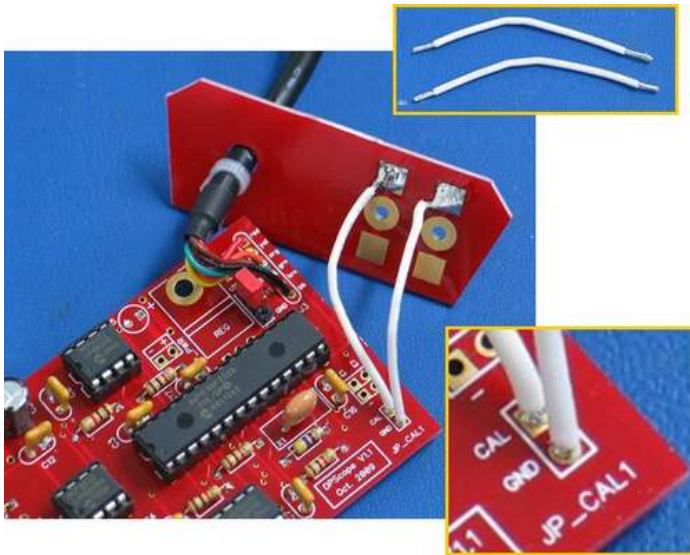


### Step 30: Assembly Step 18: Probe Calibration Output

Take the two hookup cables and solder them into the board and onto the solder pads on the backpanel as shown.

Make sure to connect the hole labeled “CAL” with the backpanel pad going to the “CAL” hole on the backpanel, and the same for “GND”.

Put the two terminal turrets (not shown) into the holes and solder them on. The long part of the turret points to the outside of the backpanel.



### Step 31: Assembly Step 19: Putting It All Together

Now put the board back into the enclosure box, and snap in frontpanel and backpanel.

Congratulations – your oscilloscope is fully assembled!



### Step 32: Assembly Step 20: Software Installation

If you haven't already done so, go to the DPSScope website ( <http://www.dpscope.com> ) and download:

- The oscilloscope software
- The FTDI USB driver
- The USB driver installation guide

First install the USB driver – follow the installation guide for that. Note that the installer will actually install two different drivers on your computer, i.e. go through two installation cycles. Make sure to complete both of them.

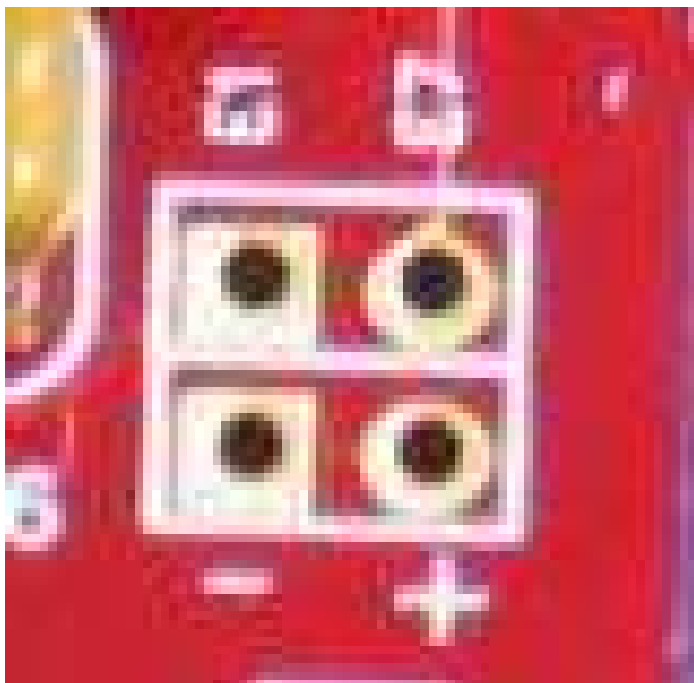
Then install the DPSScope software as well (unpack the files and click on Setup.exe).

Attach the DPSScope to a free USB port on your computer. Wait for a minute or two to give the computer time to recognize the new instrument (you should get a screen message when that happens).

The DPScope's frontpanel LED should blink a few times and then stay on. The blinking should last for about one second total. If that's the case then your oscilloscope has just passed the first functional test!

**Note 1:** The DPScope software needs a screen resolution of at least 1024 x 768 pixels.

**Note 2:** It is recommended to connect the DPScope to a USB port of your computer itself, or to a powered USB hub. Unpowered hubs tend to have large voltage drops, and the DPScope's level accuracy is dependent on a steady 5V supply voltage from the USB. You can test the voltage by connecting a voltmeter to the pins labeled "+" and "-" on the expansion header (bottom right on the PCB).

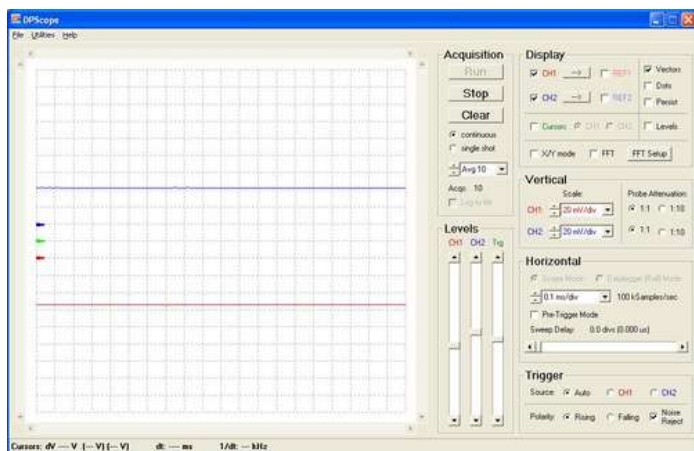


### Step 33: Assembly Step 21: Software Start

Launch the DPScope software. It should look like the picture below. Press the "Run" button – the two scope traces should come alive.

Attach the two probe cables to the BNC connectors (CH1 and CH2 on the frontpanel).

Now we need to make two simple adjustments to optimize the scope performance.

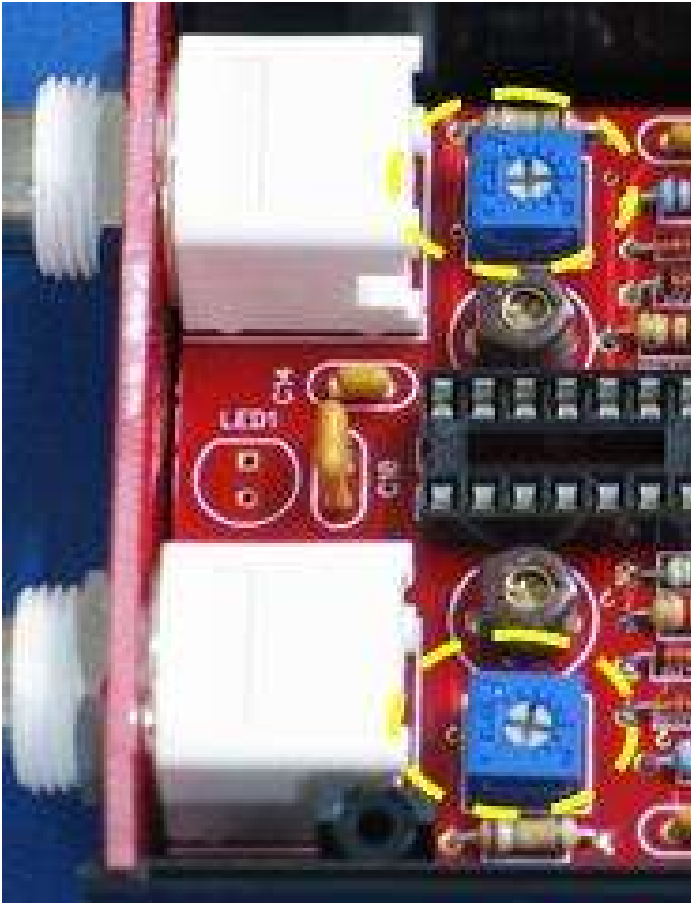


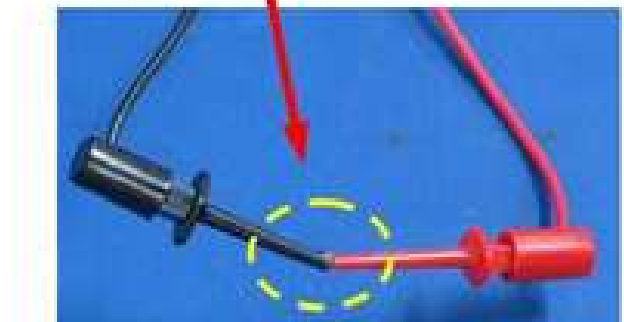
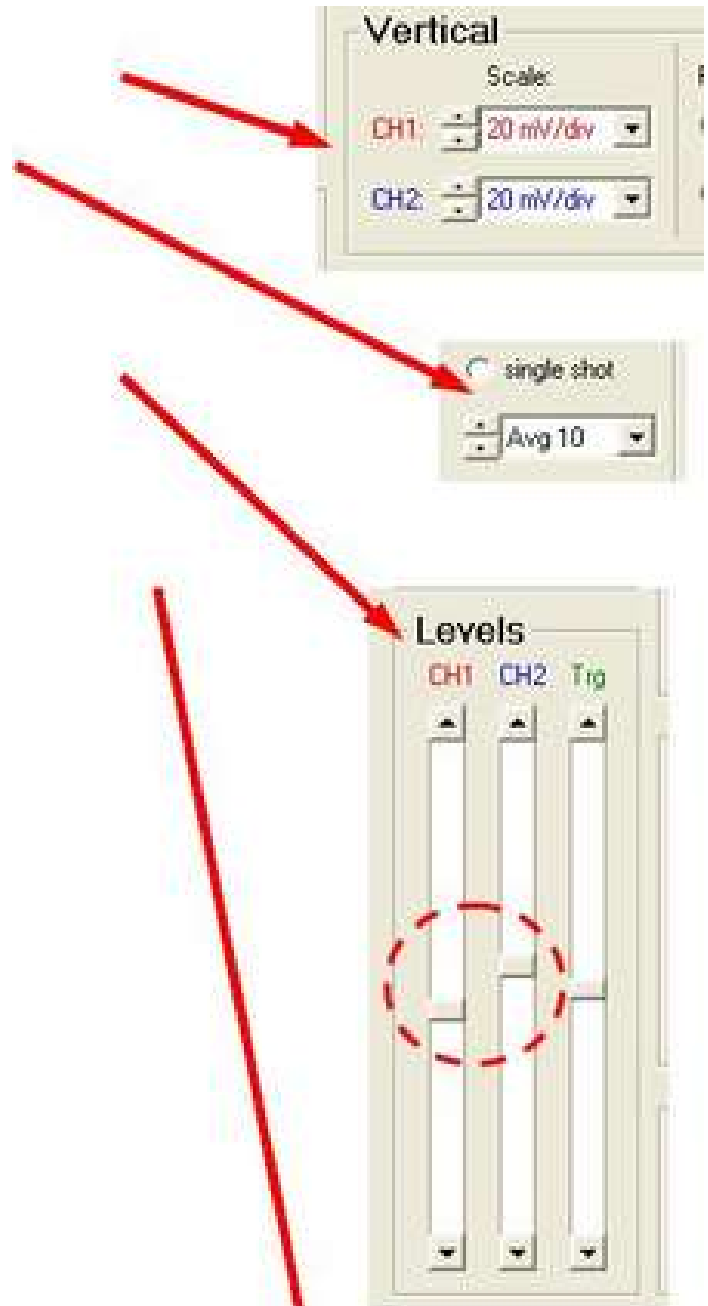
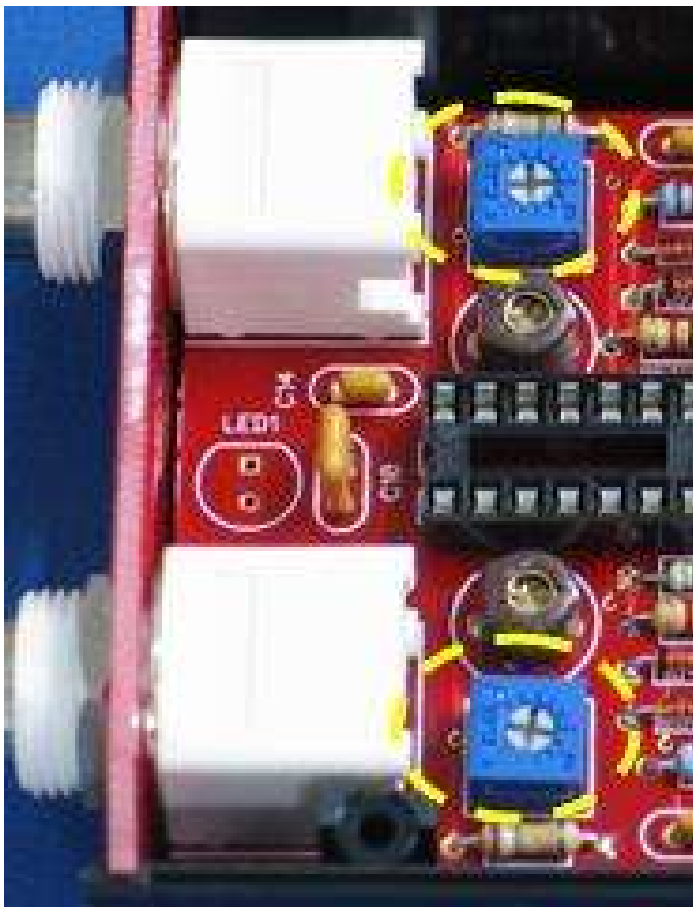
### Step 34: Assembly Step 22: Offset Adjustment

- In the "Vertical" menu change the scale for both channels to 20mV/div.
- In the "Acquisition" menu change the averaging to "Avg 10".
- In the "Levels" menu move the sliders "CH1" and "CH2" to the middle. The ground level indicators (blue and red arrow on the left in the waveform display) will be in the middle as well.
- Short the probes, i.e. connect red grabber and black grabber together.

With a small screwdriver you can now adjust the offsets of the two channels:

- Adjusting the two blue square trimmers (VR1 and VR2) will move the respective trace (red = CH1 and blue = CH2) up and down.
- Adjust the trimmers so the red trace is exactly at the height of the red arrow on the left, and the blue trace is exactly at the height of the blue arrow. Done!







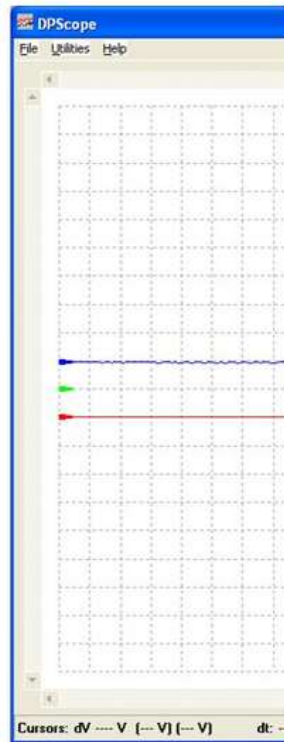
### Step 35: Assembly Step 22: Offset Adjustment

Below you see the scope display before (left) and after (right) correct offset adjustment.

Before offset adjustment:

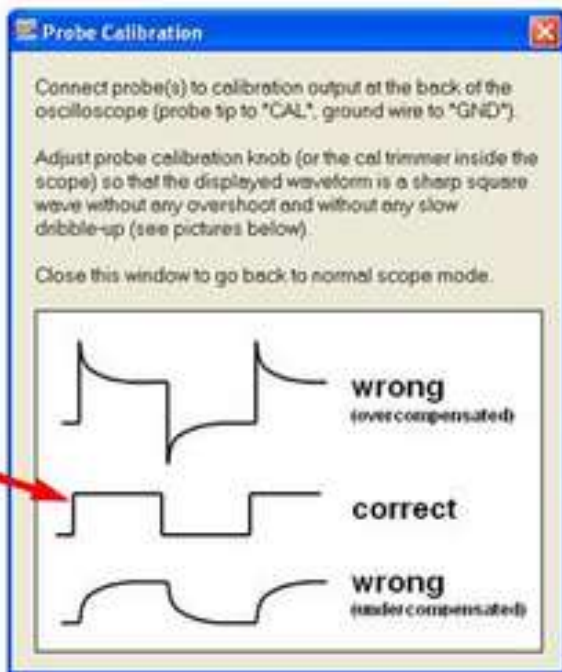
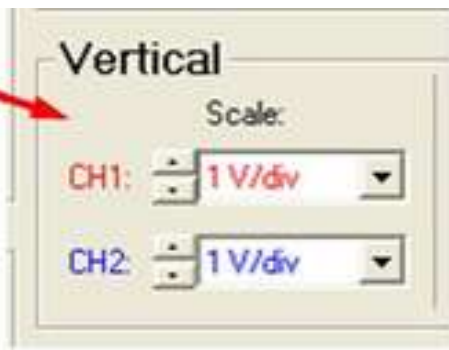


After offset adjustment:



### Step 36: Assembly Step 23: Probe Compensation

- In the "Vertical" menu change the scale for both channels to 1V/div.
- In the "Acquisition" menu leave the averaging at "Avg 10".
- In the "Levels" menu move the sliders "CH1" and "CH2" a bit below the middle.
- Connect the probes to the calibration outputs on the back side of the oscilloscope:
  - Red grabber connects to "CAL" post
  - Black grabber connects to "GND" post
- In the DPScope window on your PC select Utilities à Probe Compensation. A small window with instructions will pop up.
- With a small non-metal screwdriver you can now adjust the probe compensation capacitors (C18 and C19, respectively).
- On the right side you see examples for overcompensated, undercompensated, and compensated probes.
- The adjustment is correct when the displayed signals are nice square waves with sharp corners, i.e. when there is neither overshoot (sharp peaks after each transitions) nor slow settling (rounded edges).
- At the same time, you have tested the scope's acquisition circuitry.



### Step 37: Scope Software

The DPScope is controlled by the DPScope software running on the PC. The user interface gives you full access to all the features of the DPScope like horizontal and vertical resolution, trigger settings, acquisition settings (e.g. averaging, pre- and posttrigger range), and so on.

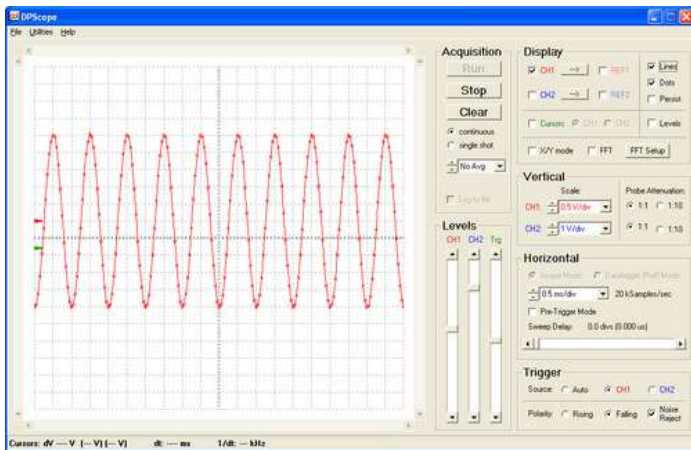
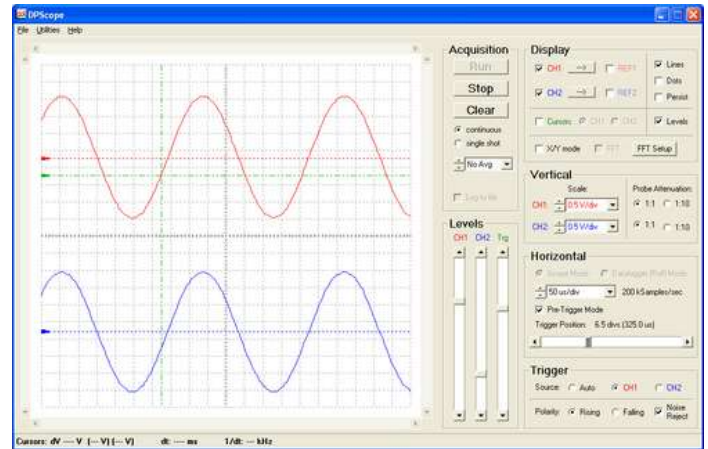
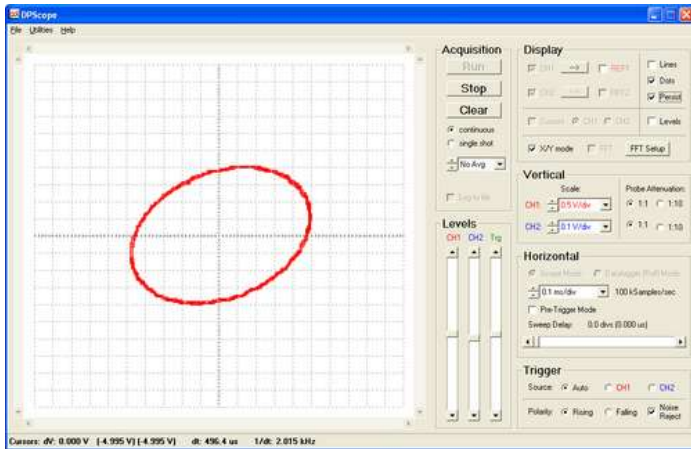
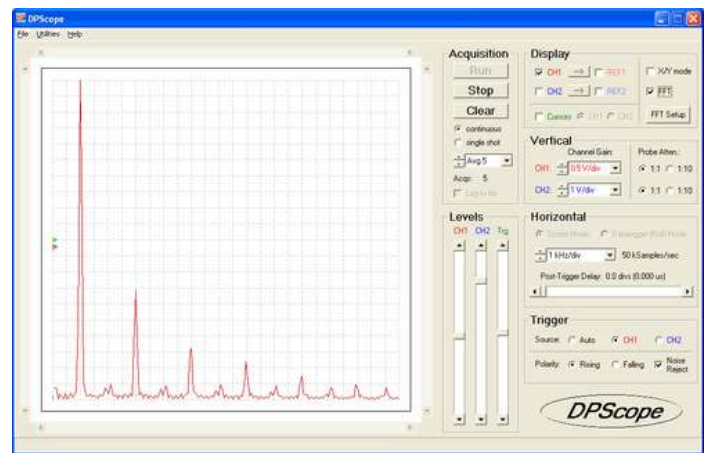
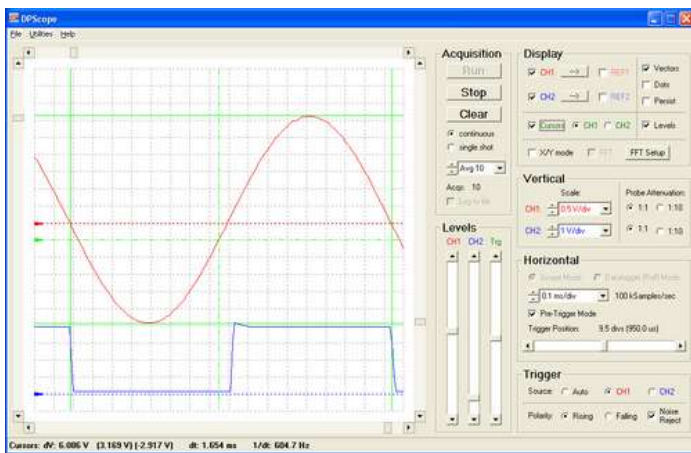
One neat feature is the FFT (frequency display) mode - in this mode the software performs a real-time **F**ast **F**ourier **T**ransform (FFT) on the data, so you see the frequency spectrum of the signal(s). This is a great tool e.g. to pinpoint small periodic noise that would be difficult to see in the normal scope display, and also to acquire an intuitive feel for the frequency domain.

Another mode is the X-Y-mode where you plot one signal versus the other (instead of both signals versus time). This allows quick characterization of components and phase shifts.

For a detailed description you should download the DPScope User Manual .

Attached below are a few screenshots that show the DPScope in action.

<http://www.instructables.com/id/DPScope-Build-Your-Own-USBPC-Based-Oscilloscope/>



### Step 38: All Done!

Now put the bottom cover on the instrument and screw it shut with the four Philips screws.

**Congratulations, you have successfully assembled, set up and tested your new oscilloscope!**

As a reminder, you can get the DPScope as a kit or fully assembled from my website:

Webpage: <http://www.dpscope.com>

From there you can also download the PC software, user manual, drivers, and other documentation.

If you still have questions do not hesitate to contact me!



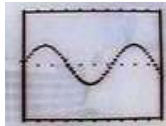
## Related Instructables



... by vithrar



**LCS-1M - A Full-Featured, Low-Cost Hobby Oscilloscope** by womai



**S2W - Scope to Web Using Baseline PIC - Step 1** by Learndy



**Hacking the Rigol DS1052E Oscilloscope with Linux** by hardwarehank



**Logic Probe Kit** by Mrdon219



**RF probe** by neelandan

## Comments

[50 comments](#) [Add Comment](#)

[view all 86 comments](#)



**saturnino** says:

Most nicely explained project. Could there be another projects ? Best wishes.

Carlos Ahumada  
Santiago de Chile

Mar 15, 2011. 5:45 PM [REPLY](#)



**womai** says:

Yes, there will be. Right now I am working on a simple AWG (arbitrary waveform generator) and a very simple, no-frills oscilloscope (much less capable than the DPScope but also even lower cost). The AWG should make a nice complement to the DPScope.

Both instruments will use the PC for control and display, just like the DPScope. The challenge here is to reduce component cost to absolute minimum. Microchip has a couple low-end, low-cost (a third of the price of a 18F2550 or 18F4550) USB-capable microcontrollers that will be fun to work with. Right now I'm waiting for the development board for them.

Stay tuned!

Mar 18, 2011. 12:01 AM [REPLY](#)



**hesslerk** says:

Hi there,

I am working on a PIC project right now with the PIC18F4550. Would it be possible for you to send me your C++ code for the oscillator, ADC, and SPI communication?  
hesslerk@gmail.com

Thanks!  
Kurt

Mar 14, 2011. 5:41 PM [REPLY](#)



**womai** says:

This scope is based on the 16-bit dsPIC30F2020, which has a very different architecture compared to the 8-bit PIC18F series, so that won't help you much.

For PIC18F series development I use Mikroelektronika's MikroC compiler and development boards. The compiler (2K limited demo download is free) comes with a few examples for the 18F4550 - just use these as a starting point, they should work right out of the box (they did for me). They use an external 8 MHz crystal and run the core at 48 MHz (maximum possible) which is probably what you want to do (USB needs exactly 48 MHz anyway). MikroC has native libraries for ADC and SPI - very easy to use - so there isn't any clever code I could share for that. E.g. adc read is my\_var = adc\_read(channel)

Mar 17, 2011. 11:48 PM [REPLY](#)



If you are interested in USB using the 18F4550, go to the MikroE web forum and search for "USB" and author "womai" - I posted a full example (including VB6 source code for the PC side).



**jumbocat** says:

Mar 4, 2011. 7:21 PM **REPLY**

Wow, Im about 4 mos into building electronic projects, and I have yet to see such amazing detailed instructions for a project! You put a lot of thought into this. I will definitely be pick your self assembly up as soon as my Solder skills are better.



**rosenred** says:

Feb 2, 2011. 12:24 PM **REPLY**

Although I cannot but marvel at the quality of your work, I could not bring myself to give you a good rating, or even consider buying your kit.

I have been a member of Instructables for quite some time and even before that, I would pop in to check on some project or another and trust me it's not the first time I see someone advertising their work. And believe me I am totally fine with that. As a matter of fact, I believe that if one can profit from a great design and/or personal work, it's not bad at all, as long as there are willing customers.

To be honest, I expected many more reactions in the comments so far, since I've seen people get upset for less obvious advertisement. But in most other cases, as I recall, the 'ibles were enough to complete the project without having to buy something from a specific someone.

In your case, you repeatedly refused to give the firmware and the circuit layout when asked and sometimes you did not even bother to answer.

I know that if I were more skilled in electronics and mc programming, I would work something out on my own, as you pointed out (using a proto-board for example).

Again, I am not giving you fault here, I just expected a little more honesty from your side. You could have stated in the beginning that this is an 'ible to build a very specific kit, as others have done. "Build your own" is a bit misleading don't you think? If I were to follow this 'ible, I would be building \*your\* USB/PC-Based Oscilloscope, not mine.

I have no intention of being offensive or anything, so I apologize beforehand if my comment upset you, I just thought I should give my two cents.



**womai** says:

Feb 6, 2011. 11:12 AM **REPLY**

No offense taken. Two things I'd like to comment on though:

First, I don't agree with "you did not even bother to answer". If you read through the list of comments, I make an effort to always comprehensively answer whatever question - technical or otherwise - somebody has. A few - not many - of the comments though were not questions at all, for all can tell these individuals only wished to express their opinion but did not really seek any response or comment from my side. They are of course entitled to their opinion, so I let them stand as-is.

Second, I do think there is a lot to learn from this project even if you have no intention of getting the board or kit. The whole hardware design - as well as the high-level software design - is documented and explained in detail - so you get a proven, usable analog frontend - a variable gain amplifier chain with ove a MHz of bandwidth - for free (of better design quality than many low-end commercial scopes have as I can assure you), as well as a good idea how a real digital sampling scope works. In fact I personally know about more than one person who copied all or part of the analog portion for his/her own project. I have seen many other instructables where the total amount of usable information was quite a bit less than this alone.

Finally, I have published a fully open scope design as well - including firmware, board layout and so on, which everybody is free to use if so desired, so overall I do feel I contribute to this community in the way intended:

<http://www.instructables.com/id/LCS-1M-A-Full-Featured-Low-Cost-Hobby-Oscillosc/>

Regards,

womai



**rosenred** says:

Feb 6, 2011. 1:32 PM **REPLY**

Thank you very much for taking the time to answer. I never said though, you are not contributing to this community, I assure that would be horrible on my part. I do get your point. I might have exaggerated a little and for that I must apologize.

Thank you for the link to your other project. Again (with my little experience in electronics) I cannot but admire the design.

I sincerely hope you were not offended or in any other way annoyed by my comment. I just expressed what was in my head at the time of reading.



**womai** says:

Feb 6, 2011. 4:29 PM **REPLY**

If you like to get a quick and easy start in electronics, have a look at the Picaxe microcontroller (on which my other scope project is based). [www.picaxe.co.uk](http://www.picaxe.co.uk) - typical times - for an absolute beginner! - to get up and running and have e.g. an LED blinking is normally measured in minutes, not hours or days (no kidding). That's what go me started with microcontrollers and re-started with electronics a few years back. There is a very beginner friendly support forum for any questions you may have.

And no, I don't get any kickbacks from them ;-)



**rosenred** says:

Feb 7, 2011. 1:20 AM **REPLY**

Thanks! I was considering the Andruino, but I am sure this one would be much simpler for a beginner like me :)

Much appreciated!



**womai** says:

Feb 7, 2011. 7:57 AM **REPLY**

The Arduino is very nice platform with a large community. But IMHO the Picaxe has a much easier learning curve for an absolute beginner. The basic circuitry is the Picaxe, two resistors for the program download connection to the PC (using a serial cable or a USB-to-serial converter cable), and potentially one 0.1uF capacitor between power and ground. Can't beat that in terms of simplicity! It's also very inexpensive - just a few \$ will get you there.

The good thing is, once you got you feet wet with the Picaxe (and gained some experience with general electronics while working with it),

<http://www.instructables.com/id/DPScope-Build-Your-Own-USBPC-Based-Oscilloscope/>



moving over (or up) to any other platform (Arduino, bare PIC or Atmel or any other microcontroller) will be much easier as all the basic concepts stay the same.



**lgeorge123** says:

Jan 4, 2011. 3:39 AM [REPLY](#)

I use visual c# to display my scope screen , your screen is 40+ frames/sec , do you feel your screen flicker ? in my code I use timer to update my ADC data and set timer to 4ms interval , I feel my screen flicker very fast :

```
private void timer1_Tick(object sender, EventArgs e)
{
    Graphics g = pictureBox1.CreateGraphics(); // pictureBox1 is picture that
    display grid
    pictureBox1.Refresh();
    // function to display adc data;

}
```



**womai** says:

Jan 19, 2011. 2:31 PM [REPLY](#)

No, my software does not suffer screen flicker. The flicker in your case is most likely due to the fact that you initialize the display (copying the grid) and then draw the traces on a control visible to the user. The drawing takes some time (and the display may update each time you draw a line, making it even slower). A better (flicker-free) method is to draw everything on a hidden control (e.g. a second display hidden behind the "real" display object) and then copy over the complete picture. That's what I use. It's faster, too, because the hidden display does not get updated while drawing.



**kalnas** says:

Sep 6, 2010. 5:16 AM [REPLY](#)

What bandwidth I can see with this oscilloscope? from 1.3 MHz to ...?(60MHz)? Thanks for answer



**womai** says:

Sep 24, 2010. 9:18 AM [REPLY](#)

The analog bandwidth is ~1.3 MHz. The maximum sample rate is 1 MSamples/sec real-time (for single-shot acquisition) and 20 MSa/sec in equivalent time mode (for repetitive signals). The 20 MSamples/sec means you can display signals up to ~2 MHz frequency (at that frequency they will be somewhat attenuated because the analog bandwidth is a bit lower than that, but you can still display them.)



**lgeorge123** says:

Aug 26, 2010. 5:09 AM [REPLY](#)

Can you disclose how you do the trigger level function of your scope , as I want to do a project in my college .



**womai** says:

Aug 30, 2010. 9:22 AM [REPLY](#)

For the trigger I use the built-in comparators of the dsPIC. It has an internal DAC (10 bits resolution) to set the compare threshold (note that it only covers 0V to VCC/2, so the scope has a 1:2 voltage divider in front of the comparator inputs). The comparator is set up so it generates an interrupt when the input rises above the threshold (i.e. compare output going from low to high); that calls an interrupt function which then performs the signal capture. You need to put a jump address to this function in the dsPIC's interrupt table at the proper location (see dsPIC data sheet). There is also a bit to invert the compare output so you can easily trigger on rising and falling edges, respectively. The beauty of all this for the scope is that apart from the voltage divider (two 1kOhm resistors) all the trigger circuit hardware resides within the dsPIC. For details about the comparators DACs etc. please refer to the dsPIC30F2020 data sheet - but basically you just write the appropriate value to a specific address in the dsPIC's RAM.



**taurus123** says:

Jul 30, 2010. 5:03 PM [REPLY](#)

This is not much of a scope, especially when you are doing digital work. It is slow, the software freezes, the screenshots are not that great. Look around before you buy this. For a few more bucks, you can do better than this.



**iEdd** says:

Apr 6, 2010. 12:07 AM [REPLY](#)

I just built mine, and I'm a bit disappointed. Here's the problems I have so far:

1. The microcontroller gets pretty hot. I think hot enough to burn your finger if you leave it there for more than 10-15 seconds.
2. The waveforms in the software will be active for about 5 seconds, then freeze for about 5 seconds, then active again, etc.
3. After about 1 minute of constant operation, it freezes. The waveforms are frozen, but the software still works (I can move the sliders, etc). So I think it's a hardware/firmware problem and the microcontroller that is freezing. I have to unplug the USB cable and plug it back in to get it going again.

Is any of this normal? I've been looking, but haven't found any shorts or bridges in my soldering. Has anyone else had these problems?



**womai** says:

Apr 6, 2010. 7:30 AM [REPLY](#)

Hello iEdd,

the microcontroller becoming hot is not a problem, but a feature :-). But seriously, this is normal, the Microchip dsPIC30F series is known for that, it uses relatively large power which produces heat accordingly. I should probably add a note to the assembly instructions so people don't wonder. I measure about 65degC on my DPScopes, which is indeed uncomfortable to touch but will not affect reliability (the chips are spec'd up to 125 degC operating temperature).

The software freezing up is not normal though. I have built dozens of DPScopes so far and had them continuous in operation for days and weeks without this happening, on different computers. Also I have not a single other user report about this happening. As a first step, please try to operate the scope on a different computer and see if the problem persists.



**Jeffrey G C** says:

Jun 16, 2010. 11:26 AM **REPLY**

Womai, i bought your dpscope also, and mine does the the same thing with the software, where it will work for a little while, but then after measuring, the waveform freezes, sometimes i cant even see the waveform, i have actually put it into a modded removable hard drive case this way i have a place to put a fan that will hopefully keep it cooler, i dont remember if that fixed it or not, i havent used it in a while, but i have another question, is it possible to not use the +5 power supply from the usb? in other words, is it possible to use a wallwart power supply to supply the power to the board, i saw that there were provisions in your schematics to run on external power, but i couldnt get it to turn it on with the wall adapter, the i put in the 5v regulator, and the capacitor as well, i just made some rough mods, i painted it, but havent finished the whole thing yet, so if anyone likes i can put pics up when i finish it, it may be a while though



**womai** says:

Jun 16, 2010. 1:25 PM **REPLY**

Hello Jeffrey, can you tell me what exactly the marking of your dsPIC is? Especially on the second line, does it say -20E or -30E? And the what is the last line (which is the batch code)? I recently encountered a batch of marginal dsPICs that had some issues (they did not freeze but their ADC was working too slowly once it heated up). That said, this only affected about 1% of the dsPICs I have tested. As for the external power supply, yes, all the hooks are there, you only need to add a 7805 regulator and another 100uF capacitor. Please refer to the assembly guide for details. From your description I suspect you simply forgot to move the power supply jumper to the "ext." position. (been there, done that :) Wolfgang



**Jeffrey G C** says:

Jun 17, 2010. 4:08 AM **REPLY**

20E/SP 0841262 and i do have it on the ext power setting, so i know that shouldnt be an issue, the light wont blink on when the power setting is on ext, so i have no idea why its not turning on.



**womai** says:

Jul 14, 2010. 9:04 AM **REPLY**

If the LED blinks when powered from USB, but not when (supposedly) running on external power, then something is not right with the external supply. Things to check: - is jumper set to "ext." when trying to run on external supply?

- is the supply feeding into the "+" and "-" points close to the 7805 regulator? (not the test points close to the CAL connector)?

- what is the spec of the external supply? - needs to be able to deliver ~500mA current, and have a voltage of ~9V (you need a bit more than 5V because the regulator adds ~2V drop)

- use a voltmeter to measure the scope supply after the regulator (the "+" and "-" test points close to the CAL connector (not the points close to the regulator)! The measurement result should be close to 5V (+/- maybe 0.2V)

By the way, I have added a user forum to the DPScope webpage (go to <http://www.dpscope.com> --> User Forum, or direct link <http://dpscope.freeforums.org/>). Going forward this is will be the best place to get help as I check it daily. Out of some reason Instructable stopped sending me notification when a comment gets posted so I may not see it for days or weeks.



**iEdd** says:

Apr 8, 2010. 9:00 PM **REPLY**

Thanks for the reply. I've tried it on my Mac through VMWare and by booting into Windows, but it's the same install. I'll have to try it on another windows machine later.

Also, wrt to software for mac - you mentioned that it would be good if an individual would get it working for mac/linux.

I can get your DPScope program running on Snow Leopard with Darwine, which is easy. Just install as normal. Problem is, it doesn't recognise the scope. Are there any clever people out there that can get winehelper to emulate the COM port through the mac's USB? I don't think it should be that difficult, I'm just not that experienced with darwine or FTDI drivers.



**womai** says:

Jul 14, 2010. 8:59 AM **REPLY**

I have added a user forum to the DPScope webpage (go to <http://www.dpscope.com> --> User Forum, or direct link <http://dpscope.freeforums.org/>). Going forward this is will be the best place to get help as I check it daily. Out of some reason Instructables stopped sending me notification when a comment gets posted so I may not see it for days or weeks when you post here (on Instructables).



**lgeorge123** says:

Jul 14, 2010. 5:59 AM **REPLY**

I found several c# and c++ programs about pc based oscilloscope , they have buffer size range from ten thousands to store data from adc . If you use a rectangle size said 700x500 pixel to display the waveform , a maximum of horizontal 700 pairs of data to display on this rectangle . But the buffer size is ten thousand long , how can this rectangle accommodate this large buffer size ??



**womai** says:

Jul 14, 2010. 8:50 AM **REPLY**

That works pretty much the same as scrolling through a text document longer than one page - only one screen of data (e.g. 700 pairs in your example) is displayed, but there is a scrollbar that you can use to look at different section of the long text (or waveform). The only difference is that the scrolling is horizontal, while in a text file you typically scroll vertically. Often there also is a possibility to fit the full waveform on the screen for a quick overview, of course you won't be able to see every little detail that way - like a thumbnail.



**UltraMagnus** says:

Jul 8, 2010. 2:31 AM **REPLY**

where is the firmware? without it this is nothing but instructions for a commercial proprietary kit.



**avscb** says:

Jul 11, 2010. 4:48 AM **REPLY**

You are right



**jolshefsky** says:

Jul 5, 2010. 4:00 PM **REPLY**

I have been using your scope for several months now, and I'm generally satisfied; my complaint is largely offset by the extremely low cost and high functionality.

My complaint is a design flaw in the offset voltage. It is reflected in the output across the 1 megohm voltage divider, sourcing several microamps of current. It is not a problem with measuring buffered signals, but it gets in the way of reading sensitive circuits. I had been using it on a high-impedance oscillator in-circuit, and the DC bias would affect the operation of the circuit in a noticeable way. I think that if the DC bias were not present, the circuit would behave better. One workaround would be to set the offset voltage to zero although that prevents reading negative voltages.

I think the input circuit needs a bit of a redesign. Offhand I can't think of a good way to do it without going to a full-blown differential input internally biased against VCC/2 which of course adds a couple op-amps to an already crowded (and power hungry) device. Simply floating the BNC ground to the offset voltage instead of tying it to system ground would alleviate the problem on isolated circuits, but grand external ground loops would probably thwart that (i.e. watching signals from the audio output of the computer would short the BNC ground to the PC ground which is tied to the USB ground, etc.)



**womai** says:

Jul 9, 2010. 9:10 PM **REPLY**

Probably the easiest solution would be to AC-couple the signal. I.e. add a series capacitor between your point of measurement and the scope input. Depending on your frequency, a ceramic capacitor of 10...100nF should be a good choice. That will float your measurement point and avoid and bias issues. Note that due to the offset voltage the scope trace will not be centered around the zero marker, but that should be a very minor issue. The way the offset generation works was a design tradeoff I decided to make, and in 99% of the cases it won't affect usability at all; on the upside it allows the whole scope to work with just a single +5V supply rail. The clean alternative would have been to add a negative supply rail - could be done e.g. with a voltage inverter. The downside of that would be more components, potentially a larger board, and an increase of about \$10 in cost. Given that the aim was to create something simple and very affordable for a hobbyist, that wouldn't have been a good choice in my opinion.



**camx** says:

Apr 26, 2010. 7:49 PM **REPLY**

Built the scope and it works well. Very nice design.



**raptorofaxys** says:

Apr 3, 2010. 2:38 PM **REPLY**

Great project! I've been looking for a long time for something around the same price point, and you really delivered a lot more than I was expecting to get for that sum.

Many thanks for such an excellent kit! I just finished assembling it and playing around with it for a bit and it delivers.



**Azayles** says:

Mar 27, 2010. 4:06 PM **REPLY**

Put it together fairly quick with no problems at all, aside from forgetting to solder the rest of the pins on one of the chips :P Works PERFECTLY! I'm so impressed, I've never been so pleased with something I've soldered together myself, and it worked first time :D Been using it to test the output of a servo controller circuit I built so I could check the PWM signals. The ability to save and load scope parameters is a BIG bonus!

Thanks again, Wolfgang, you've made an electronics geek very happy!



**Azayles** says:

Mar 27, 2010. 11:31 AM **REPLY**

Mine arrived in the post this morning. Thanks, Wolfgang! Can't wait to get this thing put together and calibrated :D

I wanna say a very big personal thank you again as I've wanted my own scope for as long as I can remember, but have never found an affordable solution which is of good design quality as this kit is. Please let me know if you start selling kits for other equipment ;-) Next stop, a really good function generator with arbitrary waveform generation :D



**MrMike** says:

Mar 25, 2010. 4:28 PM **REPLY**

Great write up and an excellent kit. Easy to follow instructions even for a beginner.

Mine arrived 3/22/10 after a 3 week parts delay, (with timely updates from the author) and took me about 2.5 hours to put together and calibrate. Thanks for marking the caps, I didn't need to pull out the the microscope or the LCR bridge.

This simple scope is a great addition to the tool box for any experimenter / hobbyist out there, and you can't beat the price.



**cardinalflyer** says:

Mar 15, 2010. 3:33 PM **REPLY**

Mine is all built up.  
[http://s794.photobucket.com/albums/yy222/cardinal634/?action=view&current=IMG\\_1593.jpg](http://s794.photobucket.com/albums/yy222/cardinal634/?action=view&current=IMG_1593.jpg)


I added a switch & capacitor to each channel to make the input similar to a Tektronix oscilloscope I have.


This change lets me switch in a capacitor in series with the input signal. That way if I want to see what is happening on a DC signal, I can switch to AC coupling - the DC is blocked by the capacitor and I can look at the signal at a higher resolution.


I found some small switches, drilled 1/4 inch holes below the seam of the box and placed so that it didn't interfere with or touch anything else. The lead from the BNC connector was not installed in the PCB - it was pulled up and wired to the center of the switch. One lead of the cap was soldered to the PCB instead. A wire was added to that same lead and connected to one side of the switch. The other side of the capacitor was connected to the other side of the switch.


The plastic tie-lock helped the board and the end plates in place while I figured out where switch would. To prevent mangling the box while drilling, I started with a 1/16" hole and then used a unibit from home depot to slowly open the hole a drill size at a time.

There are more shots before & after the portion of schematic. I am new to photobucket, can't find a way to delete stuff.

 **Azayles** says: Mar 10, 2010. 11:43 AM [REPLY](#)  
Just bought one! (told you I would :-))  
Can't wait till it comes :D I ordered the kit version 'cause I like building things myself plus it's cheaper :P  
I'll probably post back here when it arrives to leave comments :D


 **pipponum1** says: Feb 28, 2010. 1:12 PM [REPLY](#)  
Hi  
I have a question: you have a 1Ms/s interface with 8 bit per sample. It's 8Mb/s. Two channels are 16Mb/s. How can you send data to pc via a 500kbps interface? Do you throw away packages or.. How?  
Thanks


 **womai** says: Feb 28, 2010. 2:46 PM [REPLY](#)  
My scope works like any other digital sampling scope: It does not stream data to the PC at that speed. Instead, it captures one waveform record (~200 points per channel in this case) and stores it in the microcontroller's internal memory. After that it transfers that data to the PC (at 500 kbaud) where it gets displayed. Then it waits for the next trigger, captures another record, transfers that, and so on. On my computer I get about 35 - 40 records per second. For the eye it makes no difference, changes of the waveform display perfectly fluently - you won't be able to tell if it's 40 or 4000 records persecond anyway (remember, movies are just 25 pictures per second, the eye isn't faster than that).  
  
So overall, at fast sample rate the scope spends most of the time transferring and displaying the data. As I said, every other digital real-time scope works the same. There are scopes that can do 40 GSamples (yes, 40 billion samples) per second on several channels simultaneously, no data connection to the PC could get even close to carry that as streaming data.


 **pipponum1** says: Mar 6, 2010. 4:17 AM [REPLY](#)  
Thank you for the answer.  
Another question. Is it possible to buy only the firmware? Simply the .hex file (which - I think - is not editable) sent by e-mail? If it is possible, how much is it? Thanks


 **gtkindust** says: Mar 3, 2010. 6:19 PM [REPLY](#)  
Embedded link in step 10 for Diptrace is wrong. It shows www.dptrace.com should be www.diptrace.com


 **womai** says: Mar 3, 2010. 7:27 PM [REPLY](#)  
Thanks for pointing this out! Should be correct now.


 **saLZBURG** says: Mar 1, 2010. 9:25 AM [REPLY](#)  
I am interested in building this scope but microchip doesn't have a 24 F 2020. What other chip could be used instead.

 **womai** says: Mar 1, 2010. 1:25 PM [REPLY](#)  
Actually the scope uses a **dsPIC30 F2020**, not a PIC24F2020 (which does not exist as you found out). But there is no need to order your own - a pre-programmed 30F2020 is included in the scope kit (which you can get from my website, [www.dpscope.com](http://www.dpscope.com)).

 **womai** says: Mar 2, 2010. 7:10 AM [REPLY](#)  
Ok, just realized where you got the notion that it's a 24F2020 - a typo in my description. Corrected now. Sorry for the confusion.

 **media1328** says: Mar 1, 2010. 9:52 AM [REPLY](#)  
Hello. many many thanks i have been looking for this scope for long times but now i can have it. my only question is how can i get the PCB for top and bottom layers?  
all bests for you Dear Womai.

 **womai** says: Mar 1, 2010. 1:23 PM [REPLY](#)  
Hi,  
  
the PCB is included with the scope kit that you can get from my website ([www.dpscope.com](http://www.dpscope.com)), so no need to fabricate your own. The PCB isn't suitable for home-brew methods anyway since it has pretty small vias and narrow lines, and having one produce a single one professionally would cost more than the whole scope kit.

 **media1328** says: Mar 1, 2010. 8:44 PM [REPLY](#)  
Hello thanks for your mail. as I understood I have to buy your kit and it's not possible I make it by myself. and in other hand I can't buy it directly because I have no credit cart. and or pay pal or such accounts so that many thanks again. and if it was possible for you to send me the top and bottom layout i can make the PCB by myself your help will be appreciated.  
thanks and good luck Media.

[view all 86 comments](#)