

PCB Scope / Logic Analyzer

Hardware Design Description

Introduction

The PCB scope is the result of a challenge I set for myself – to build a practically usable oscilloscope with a minimum amount of components and for minimum cost. The practical benefit is of course that this is an instrument that I hope will be interesting to many teachers, students and hobbyists looking for an affordable, simple tool for their electronics work.

There were a few basic requirements for whatever circuit I came up to justify the “practically usable” label: There are quite a few simple “sort of an oscilloscope” type circuits out on the web, but all I know fall short in one or more respects:

- Only one input channel – very often one must display one signal in relation to another (e.g. clock and data, or input vs. output) or trigger on a signal different from the one to be displayed, and this needs at least two channels.
- No reasonable protection against overvoltage at the input.
- Very limited range of input voltages (e.g. just 0...5V) and no adjustable input amplification or attenuation.
- Only usable for very slow signals because of limited sample rate (a few kSa/sc) and/or low bandwidth (a few kHz).
- Non-standard input impedance – standard scope probes need 1 MOhm. Input stage not frequency compensated (limits bandwidth to a few kHz at best).
- Use of exotic, obsolete or unnecessarily expensive components.
- Unnecessarily complex circuit especially given the limited performance.

With the PCB scope I tried to address all these issues. Of course the end result can never compete with a professional Tektronix or Agilent scope costing hundred times more, but it is still good for a large number of applications.

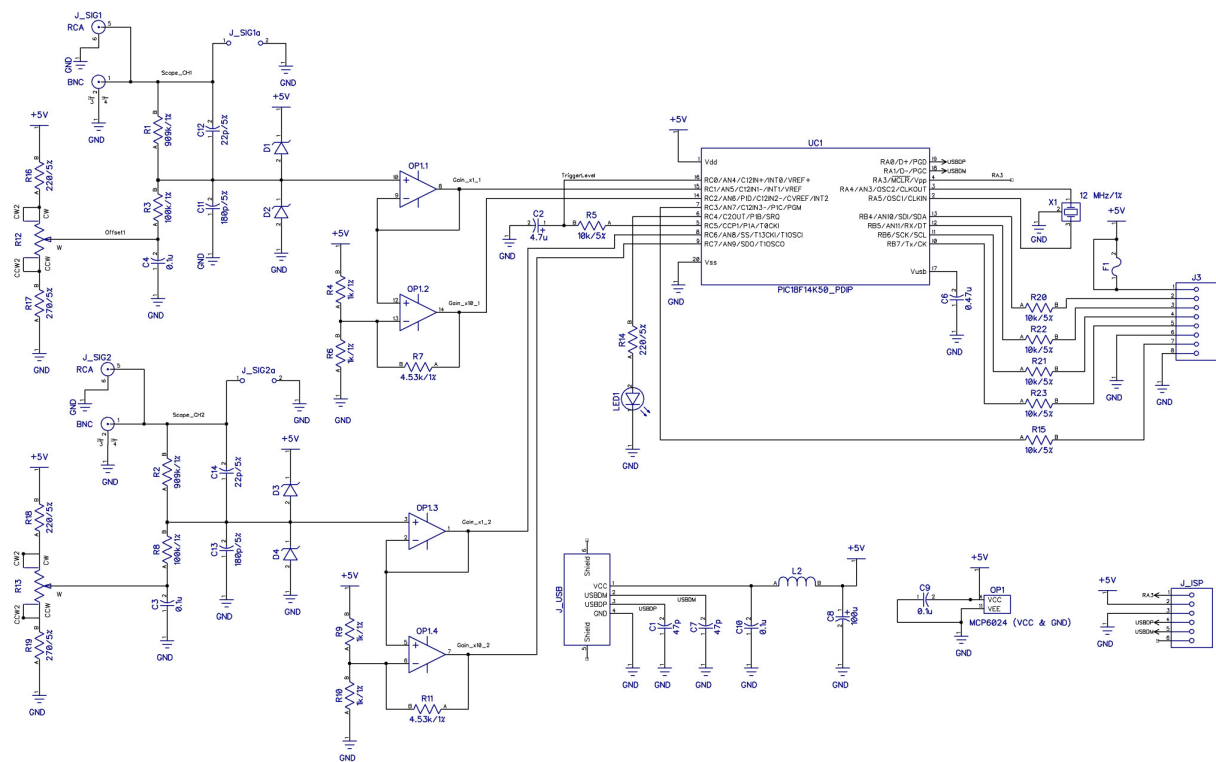
To minimize cost some preliminary considerations set the general direction:

- Try to integrate as much of the scope hardware into a single chip.
- The instrument should use a PC for control and display – saves the cost of a dedicated LCD display (while providing unmatched resolution and processing power) and front panel control knobs and buttons.
- Connection should be through USB (fast and today’s standard – legacy serial port connection require USB converters which again adds cost since most new PCs don’t even have RS-232 ports anymore).
- Power to be provided through USB connection – saves the cost of a wall-wart power supply.
- Be very careful about adding components and be diligent looking for low-cost, easily available components.

- Don't use hard-to-solder, fine pitch components so the scope can be distributed as a self-assembly kit anybody can build.

In the end I was able to squeeze almost everything onto a single microcontroller (a Microchip PIC18F14K50) – acquisition control, capture memory, USB interface, digital-to-analog converters (DAC), trigger circuitry and trigger threshold generation. The only major block outside is the analog frontend – and even this is just one quad op-amp chip and a number of resistors and capacitors.

Circuit Overview



The picture on this page shows the complete oscilloscope schematic. We will dive into each section in detail later, but first let's get a broad overview.

The signals to be displayed enter on the left side. There are two input channels, their design is identical. First the signals get attenuated to make sure they do not exceed the range of the op-amp or the ADCs (0...5V). The attenuator has very high impedance and also acts as protection against excessive voltage at the inputs, together with the clamping diodes.

A low impedance divider provides some fixed offset to shift the input signal up, that way positive as well as negative voltages can be measured.

The signal then continues to the amplifier stage, consisting of two op-amps (both on the same physical op-amp chip). The first one acts as a simple unity gain buffer, the second stage provides 10x amplification.

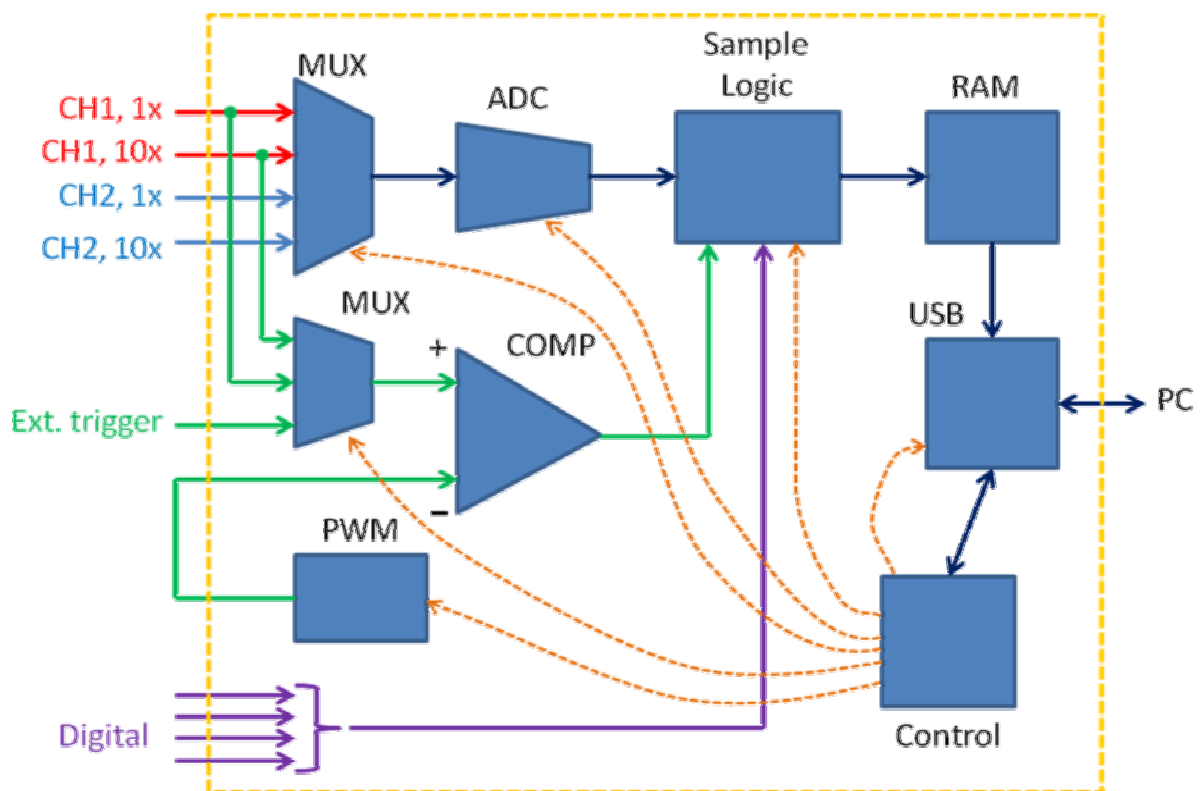
Finally the signal enters the microcontroller where it gets digitized. The scope can change its voltage resolution by selecting which version of the signal to use – the one from the unit-gain buffer (for large input signals) or the 10x amplified version (for small signals).

The microcontroller captures the signal – the acquisition parameters (sample speed, voltage scaling, trigger setup come from the controlling PC) and sends the data back to the PC for display.

There is also a logic analyzer port – four digital inputs into the microcontroller.

Finally, the USB interface transmits data between microcontroller and PC and also provides power to the whole circuit.

Microcontroller



The microcontroller (a Microchip PIC18F14K50) is the heart of the whole instrument. Above you see a high-level block diagram of its internal structure. As said before, almost all functions are integrated into this single chip, Microchip really managed to pack a ton of features into this little marvel costing just around \$3.

Even though there seem to be several ADC input pins the truth is that there really is just a single ADC whose input gets multiplexed onto these pins. That means one can only take one sample on one channel at a time; for two channels the acquisition is done interleaved, i.e. one sample from CH1, then one from CH2, again one from CH1, and so on. The maximum achievable sample rate is just above 50 kSa/sec. (For finer resolutions the scope uses equivalent time sampling, up to 2 MSa/sec equivalent sample rate). The ADC's analog bandwidth is a few hundred kHz (the data sheet does not specify bandwidth, this number is from my own measurements).

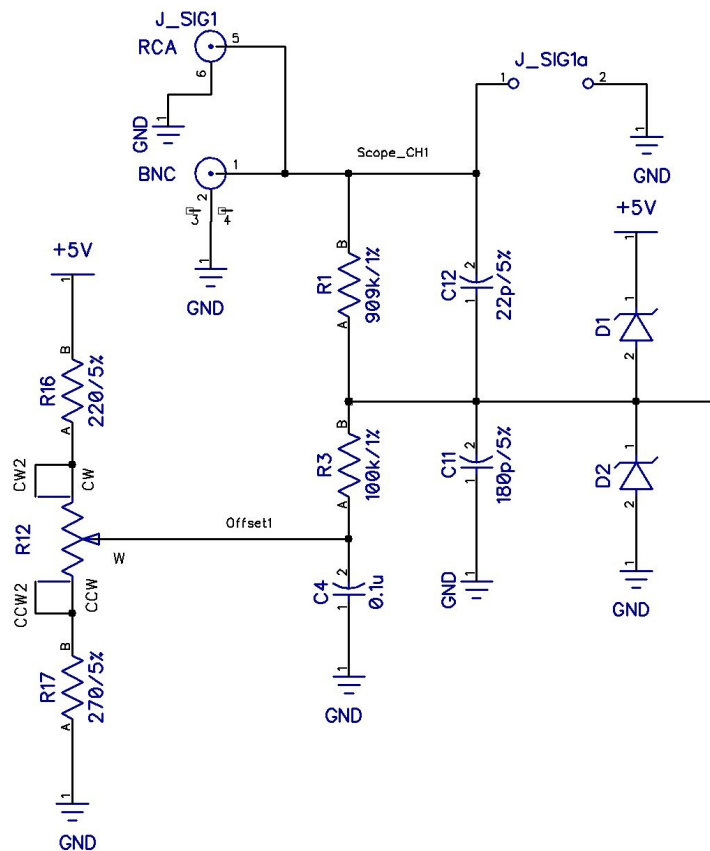
As with the ADC, the comparator inputs can get multiplexed on several different pins – here, CH1 low (1x) and high (10x) gain path and external trigger. The other side gets the trigger threshold, produced by the PWM (pulse width modulation) module. The microcontroller can set up an interrupt that reacts whenever the comparator output changes state (i.e. the input signal crosses the threshold) which allows for very fast, low-latency reaction to trigger events.

The sample logic (actually a piece of code) controls the acquisition process – processing interrupts from the comparator, timer interrupts to sample at regular intervals, transfer of the sampled signal data into the microcontroller’s internal memory (RAM). It can either capture data from the two analog scope channels or from the four digital logic analyzer inputs.

The microcontroller also has a complete USB interface built in which provides data connection to the PC. A program on the PC lets the user control the acquisition settings and displays the data, so it looks and feels like a regular oscilloscope.

The main program running on the microcontroller takes care of communicating with the PC (receiving commands and sending the captured data) and sets up the sample logic, comparator, ADC with the acquisition parameters.

Input Stage



Let’s have a look at the input stage first. Since the two channels are identical in their design we will concentrate on scope channel 1 (CH1).

To allow for connection of different types of probes the input offers three different options for the connector. The standard one is the BNC connector, which is mandatory if the scope shall be mounted in the Serpac enclosure with the provided custom end panels. For bare-board variants the user can install RCA connectors (also known as Cinch or audio connectors) – some people prefer these because it's easier to build cheap probes for them. There is also space for a 0.1" spacing jumper header – useful to hook the scope e.g. to a breadboard through jumper cables.

The microcontroller's analog-to-digital converter (ADC) has a fixed input range of 0 to 5V. The op-amp (rail-to-rail input and output, see next section) can only transmit signals in this range as well. Signals smaller than that range will get measured with reduced resolution (because they don't span the full range of possible digital ADC values), 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 analog-to-digital converters (ADC).

First, in the input stage, the incoming signal is attenuated by a factor of 10 by the voltage divider formed by resistors R1 and R3. This increases the maximum voltage range to 50V peak-to-peak. R1 and R3 in series yield 1 MOhm total resistance so the scope input has the correct, standard impedance required to use standard passive 1:10 or 1:100 scope probes.

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 an offset voltage, provided by the low impedance divider built from resistors R16, R17, and trimmer R12. Capacitor C4 assures that this offset voltage does not float around when the input signal changes quickly (the power supply could not react fast enough to quick changes – C4 takes care of that because it acts as a charge buffer).

R12 is adjusted so the voltage going to the op-amp input is at 2.5V (i.e. exactly at half the supply voltage and thus the ADC range) when 0V is applied to the scope input. The 2.5V level thus acts as sort of a "virtual ground level" for subsequent processing stages. Since the input divider reduces the input swing by a factor of 10 this means the scope can now measure input signals between -25V (resulting in 0V going to the op-amp) and +25V (resulting in +5V going to the op-amp).

If you use a 1:10 scope probe – basically a glorified (well, actually frequency compensated) 9 MOhm resistor, the input divider ratio becomes 1:100 (divider formed by probe and R1 on one side and R3 on the other) and the range increases to +/-250V (but be VERY CAREFUL whenever working with such high voltages!).

Important note: One source of confusion is the fact that the bottom of the input divider is not going to ground (0V) but rather to the offset voltage. This is different from how most other scopes work and is the price to pay for using just a single, positive supply rail. It means that if the input is left open (nothing connected to it) it will float up to the offset voltage and as a result the scope will NOT show 0V but rather some positive offset. It also means that the software has to adjust the vertical offset applied slightly when going from a low-impedance 1:1 probe to a 9 MOhm 1:10 probe.

The input divider (attenuator) deserves some further consideration. It is a so-called compensated divider and consists of a parallel combination of a resistive divider (R1 and R3) and a capacitive divider (C12 and C11). The reason for adding the capacitive divider is the fact that the protection

diodes (D1 and D2) and the input of the op-amp (OP1.1) have some unavoidable parasitic capacitance C_{par} on 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 (in our case to just a few kHz).

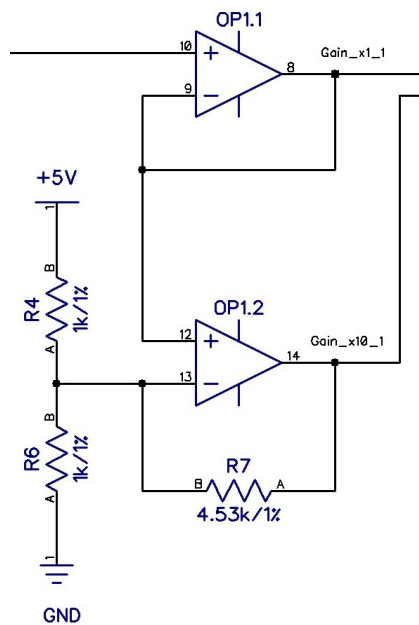
The solution - if you can't beat them, join them. Adding the capacitive divider and adjusting it to the same division ratio (1:10) as the resistive divider makes the frequency response flat from DC to light (at least in theory - but close enough for our purpose). The necessary condition is that the division ratio of the resistive divider (R1, R3) is the same as the division ratio of the capacitive divider:

$$(C11 + C_{par}) / C19 = R1 / R3$$

In many higher end scopes one of the capacitors is made adjustable to trim the exact capacitance ratio, but to keep things simple (and low cost) I opted for fixed capacitors. One less thing to adjust! 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 circuit can be found in virtually every oscilloscope.

Finally 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 (900 kOhm) between input and diode they don't need to shunt a lot of current even at large overvoltages at the scope input. The diodes used (1N914) are very easily available yet offer very fast switching. (Actually almost any CMOS IC, including the op-amp, has a similar protection structure on their inputs and outputs, but for scope inputs that may be subject to significant overvoltages I did not want to rely on this for circuit protection). R1 is rated to a static load of at least 100V or 200V (depending on specific model), so this gives good headroom compared to the specified input signal range (25V peak). For short times the input structure can withstand much higher voltages (but don't rely on that!).

Amplifier Stage



The next stage for the signal is the op-amp (operational amplifier). The PCB scope uses the MCP6024 which has 4 single op-amps in one package. Its gain-bandwidth product of 10 MHz (meaning for small signals you can get a bandwidth of 10 MHz for 1:1 gain, or 1 MHz for 10x amplification) is more than sufficient for our purposes (a few 100 kHz at most). For large signals the slew rate is actually the more important parameter to look at since that tends to be the limiting factor. The MCP6024 can do $7\text{V}/\mu\text{s}$ so it can span a 5V range in less than one microsecond – again well sufficient for the required performance here. (A rise time of $1\ \mu\text{s}$ corresponds to a bandwidth of roughly 330 kHz.)

What is also important – the MCP6024 is a so-called rail-to-rail amplifier, i.e. it works fine even if the input signal or is output signal go all the way to its supply rails (0V and 5V in our case), or at least very close to them (a few mV) as long as it does not need to drive large currents. This is by no means a given – most op-amps need some margin between their supply levels and the maximum and minimum signal levels they can work with. In any case rail-to-rail capability makes our life very easy and helps keeping circuit simple.

But back to the signal chain of our scope: The signal is first fed into a simple non-inverting op-amp follower (buffer) stage (OP1.1) which produces a 1:1 copy of the signal. This buffering is also necessary because the input stage constitutes is a high impedance source which could not drive the ADC directly (Microchip states that the maximum source impedance should be 10 kOhm or less to guarantee settling to 1 LSB within one sample period, and we need even better because the scope performs equivalent time sampling for fast sweep rates – meaning the signal has to settle faster than one “real time” sample interval).

The buffered signal (output of OP1.1) drives one of the microcontroller’s ADC inputs directly (pin RC1/AN5), and also feeds the input of a 1:10 gain stage (OP1.2, hooked up as a standard non-inverting amplifier) that produces a signal amplified by 10, which in turn goes to a second ADC input (microcontroller pin RC2/AN6). That way one can choose between less amplification for large input signals and large amplification for small signals simply by telling the microcontroller to sample on RC1

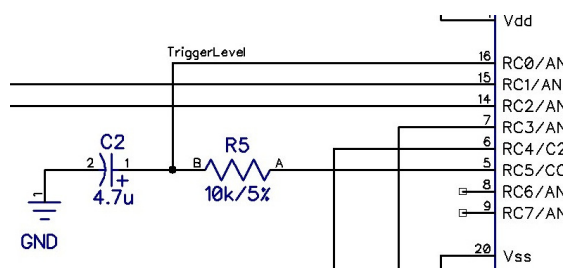
or RC2, respectively, without the use of any relays or switches (which would make the design larger, more complex, and more costly).

The negative (inverting) input to OP1.2 is biased to 2.5V through the divider R4/R6 since as outlined in the previous section our virtual ground is sitting at half the supply voltage. The gain of this stage is given by R7 and the parallel combination of R4 and R6 as:

$$\text{Gain} = 1 + R7 / (R4 * R6 / (R4 + R6)) = 1 + 4.53k / 0.5k = 10.06$$

which is the very close to the desired value of 10 (better than the 1% tolerance of the resistors used anyway).

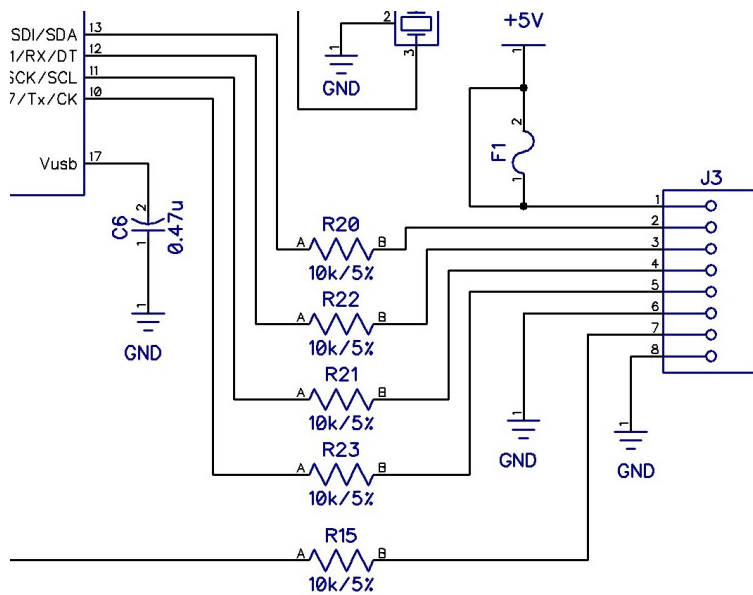
Trigger



The trigger uses the comparator built into the microcontroller – again great, no external circuitry required for this functionality. The trigger compares the incoming signal (actually the scaled copy of the input signal arriving at the ADC) with a user-controlled threshold level.

We still need to generate the required trigger threshold (the voltage level at which the trigger should fire and start the acquisition), and this threshold must be adjustable. Instead of using a digital-to-analog converter (DAC) which would add significant cost, the scope generates this level using pulse width modulation (PWM). The microcontroller has a PWM generator built into hardware so it can generate PWM signals with programmable frequency and duty cycle independently in the background while the program runs. The frequency is fixed, and the duty cycle (time within one period where the signal is high) is variable, depending on the desired trigger level. E.g. if the signal is high for 20% of the period (and low otherwise), the average level is then 20% of the supply voltage, i.e. 1V. Of course this signal is not yet a static DC level but rather a fast square wave, so the PWM output feeds a low-pass filter consisting of R5 and C2. The time constant (R5*C2) is chosen fast enough for the level to settle to a new value within less than about 200ms, but much longer than the PWM period. That way the output settles to the average voltage of the PWM signal.

Logic Analyzer Port



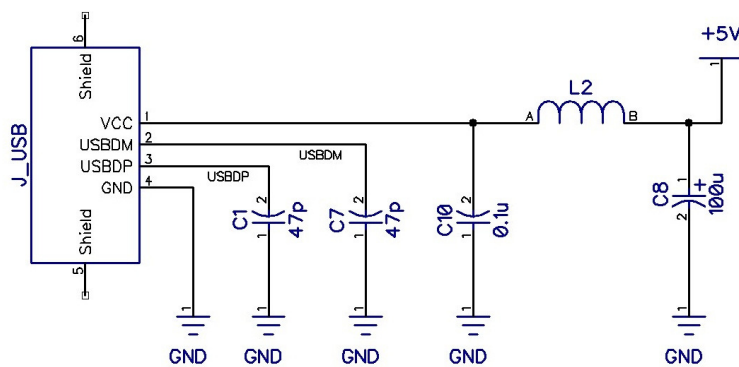
In addition to the normal scope function (to measure analog signals) the PCB scope also offers a logic analyzer mode. In this mode it can acquire up to four digital signals (i.e. at each sample instant it will only show if the signals are high or low). The advantage is that it can store longer records (since there is only 1 bit of data per channel for each sample instant, instead of 8) and sample much faster (because it simply reads the state of its digital inputs instead of performing a – much more complicated and time consuming – analog-to-digital conversion with its ADC).

The digital signals enter through the multipin connector on the backpanel of the scope and go through 10 kOhm protection (current limiting) resistors directly to four PORTB inputs of the microcontroller. Triggering in this case is done by setting up a “interrupt on change” on the channel selected as trigger channel.

Note that different to the scope channels the logic analyzer inputs do rely on the protection diodes internal to the microcontroller (plus the current limiting function of the series resistors, which makes this approach safe for up to about 15V). Microchip explicitly condones such an approach (there are even apps notes making use of it), and second the logic analyzer port is only intended to probe digital signals which normally do not exceed 5V, but take care not to exceed that.

The logic analyzer port also “breaks out” ground (must be connected to the circuit under test of course), an additional external trigger for the scope mode, and the 5V USB supply. There is an option to install a thermal fuse (F1) that limits the supply current in case of a short circuit. The main reason for the 5V pin is to enable attachment of (yet to develop) add-on boards, e.g. a signal generator. The scope supplies power to these boards and send and receive data through the trigger pin and the logic analyzer channels – the board itself then can be very simple because it needs neither its own power supply nor an extra USB connection to the PC.

USB Connection



The USB connection has two functions here – it serves as the data link between scope and PC, and in addition it provides the circuit with power.

The PIC18F14K50 microcontroller has a full USB interface already built in (actually that is one of the main reasons to use this particular microcontroller), so implementing the USB data connection becomes almost trivial on the hardware side – the connector and two 47 pF capacitors (C1, C7) is all that is needed! The device is set up as a HID (human interface device), which is one of the two most common choices for simple USB enabled devices (the other being CDC = virtual serial port). HID can transfer data at up to 64000 bytes/sec (almost 64 KB/sec), or – more precisely – up to 1000 data packets per second where one data packet can contain up to 64 bytes of data. This is absolutely sufficient for our purposes: One full data record on the PCB scope has around 420 bytes (210 bytes for each of the two analog channels), i.e. needs just 7 maximum-length packets. Then there is some overhead to set up acquisition parameters, start the acquisition and query the scope status, but still one full acquisition uses less than 20 data packets, so the USB data bandwidth is not a limiting factor for the capture rate.

The power supply was kept as simple as possible as well. It relies on the fact that the USB supply voltage is at least roughly regulated (nominally 5V but it can vary between ~4.3V and ~5.3V) and matches the voltage requirements of our circuit (5V), so there is no voltage regulator. (More advanced USB supplied circuits typically run at 3.3V so they can use a simple low-dropout regulator to provide regulated 3.3V from the raw USB supply voltage. But to get regulated 5V from the 4.3...5.3V input – i.e. sometimes lower and sometimes higher than the output voltage – we would need a buck-boost switching regulator which would roughly double the scope's complexity).

There is some minimum power supply decoupling – the bulk electrolytic capacitor C8 in combination with chock L2 and ceramic capacitor C10 filter the USB supply: C8 acts as a buffer (reservoir) to take care of surges in the circuits power consumption. C10 and L2 block high frequency noise coming from the PC to the scope or disturbances coming from the scope.