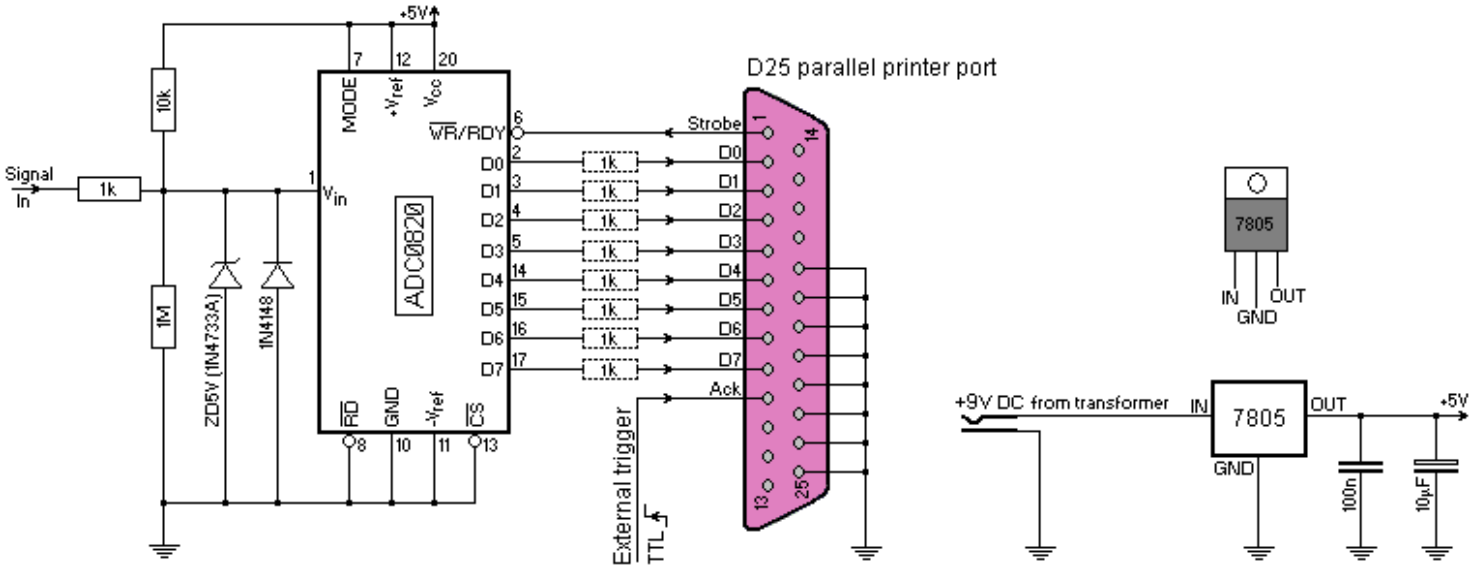
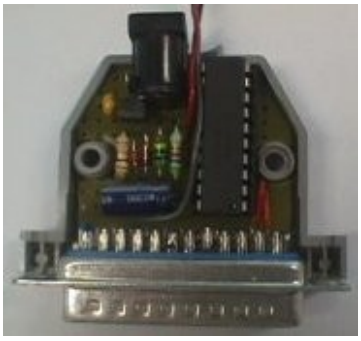
LPTScope hardware





At the beggining: I don't take any responsability if you burn out your PC or anything else. I don't even care. You are using this information at your own risk.

ADC0820 (TLC0820) is a cheap and fast ADC. I like it! See the datasheets.

This oscilloscope uses the SPP or EPP parallel port (LPT1) for reading the data from ADC. If your LPT1 is an ECP then the program automatically switches the LPT1 in byte mode (normal mode). BUT, not all parallel ports (LPT ports) are bidirectional. "Bidirectional" means, that except outputing data, the port can also read the data.
How to determine if your LPT1 port is a bidirectional one? Start the LPTscope program and on the program menu select: "Bidirectional capabilities testing" then follow the given instructions.

This is how the test is performed:

1) Program sets pins 2-9 of LPT1 port all high (5V). In other words: program puts a 0xFF in the data port of your LPT1 at adress 0x378
2) Program then continuosly reads the LPT1 data port and tracks if anything changes
3) Your task is to connect any data pin to a ground. For example, connecting pin 2(DATA0) to pin 18(GND) would change the read value from 255 to 254. You should use a 1kOhm resistor to connect a "high" pin to a GND pin
    a) If the port is not bidirectional (pin 2 keeps 5 volts), 1k resistor is big enough to limit the current at 5mA. 5mA sources from pin2 and sinks on pin18 which is acceptable on all ports without damaging it. (1kOhm means current of 5mA at 5V).
    b) If the port is bidirectional, 1k resistance is low enough to pull the pin 2 low, from 5Volts to 0 Volts.
4) If the read value changes this is a good news: your port is a bidirectional one!

If it does not change, first, try to change the settings in the BIOS of your PC (in the case when your LPT1 is on the MB). Make few tries in this order: Normal, EPP or ECP.
If the test doesn't pass, then you can't use the above shematics. But hey, do not give up. Just consider some other options. For example this. In this option you should drastically change the above schematics, and just a little the program in the LPT-reading part. For example this project uses multiplexed reading. But, using multiplexed data gathering, nibble by nibble, you obtain twice as

slower reading speed.

In addition, there is a software test upon starting the program, which checks if the hardware lets the software set the bit 5 of the LPT's control byte. Setting the bit 5 of the control port to a logical high, we put a LPT port in the bidirectional mode. There is lots of different ports around. Differently protected, or not protected at all. Read this article if you want to know more about this issue.

Those 1kOhm current limiting resistors between ADC and LPT are here just in case somebody tries to connect this to a non-bidirectional port. You can omit this resistor if you know that you are going to plug it in a bidirectional port only. I never tested this schematics with those resistors, since I'm sure that my PC is a bidirectional one. Or else this resistor could help if someone tries to send something out (printing) when the oscilloscope is plugged in LPT1...

10k, 1M resistors and 1N4148 and ZD5V are there for analog input protection. Analog input (signal in) at the pin 1 should never exceed the supply voltage of the ADC. If you are planning to measure the signals with amplitude more than 5V, you should add an additional resistor divider between signal and ADC to keep the voltage below 5V at pin1. And of course, you can omit this protection and connect the signal directly to pin1 if you are sure that your analog input will allways be in range of 0V-5V.

Pin 10 of LPT1 (Ack) acts like an external trigger. It is a TTL level trigger which fires at rising edge.

ADC0820 works in wr/rd mode (pin7 high), stand-alone operation.
To make a measuring sample, program has to set the pin6 low for a very short time. After that (820ns ater that - regarding to datasheets), the data is ready on data pins. Simple, isn't it. When doing the program, I was worrying about timings, but during the testing, I realised that ADC0820 is faster than any LPT port (available this days), and all the program has to do, is a short low impulse at pin6 and read the data as quick as it can. I never noticed any data was lost (eaten samples) during the testing. I must say that any data wasn't lost in a DOS version of the program. Windows are a bad choice for a heavy timing tasks such as a (software) oscilloscope. Windows runs other programs in the background. That's why the win version doesn't have 100% accurate time base, but in other side it looks fancy.
The bottleneck in the data transfer from ADC to PC is read/write routines of the LPT port, not the speed of ADC0820. For one data sample we need 3 I/O instructions from PC:

_out (set pin6 of ADC low)
_out (set pin6 of ADC high)
_in (read a data byte from ADC)

Under different OS and processor modes these I/O instructions take different number of processor cycles. Read this if you want to know more.
I tried to set an external oscillator for ADC clock and avoid two "out" commands to save some time, but this way we loose synchronisation between ADC and PC and data became messed. ADC must be synchronised with the PC (does it?).

Conclusion: Almost everything I know about parallel ports I learned at this great site: http://www.beyondlogic.org/spp/parallel.htm.
Since I have just few PCs available for testing, it is possible that many unpredictable things (nice or bad) happen. I'm thankful for any suggestion from you. I will update this page if you know or find out anything worth mentioning.

Other LPT handling and programming related stuff:
http://www.lvr.com/parport.htm
ftp://ftp.armory.com/pub/user/rstevew/LPT/zha96lpt.faq
...

Back home