Powering devices from PC parallel port

Introduction

PC parallel port was never designed to provide any output power to divices connected to it. it was only designed for connecting printer to your PC. But in this weird PC business everything changes. Suddenly the parallel port have become a port where you can connect very wide selection of devices like software protection dongles, removable disks, modems, network adapters, printer sharing units, parallel port extenders, soundcards, electronic telecard readers and many homebuilt circuits.

Most of those external circuit have their own power supplies. You can consider external power supply with some devices not very useful because that transformer is usually bigger than the device itself and you have yet another transformer wint it's wiring lying around your table. Sometimes it would be useful if the circuit would work without external power supply.

Because there is no power output in parallel port, operating a circuit from parallel port seems to be first quite impossible. But there is one solution to this problem: steal some power from the parallel port signal lines. This same approach is used with PC serial port powered devices like PC mouse. The same method is also useful in parallel port and some devices like copy protection dongles and some sound output devices use this method.

New features in PC parallel port (EPP,ECP, IEEE 1284) make the port faster, more flexible and more standardized. This will make the parallel port more and more attractive interface for all kinds of peripherals. But the new features have not become without a price: some old circuits just refuse to work in those new ports.

Parallel port powered devices

The most traditional printer port powered device is software protection dongle. Those small boxes are connected between the PC parallel port and the cable going to printer. To operate those dongles must take the power from somewhere: from the signal lines--a definite no-no. This method works sometime and but causes too often problems. The dongles don't work reliably with some parallel port and sometimes they need that the printer is kept powered all the time for the dongle to operate. The most common problem is that the dongle does not get enough power from the parallel port (voltage in signal pins drop too much) in some system configurations.

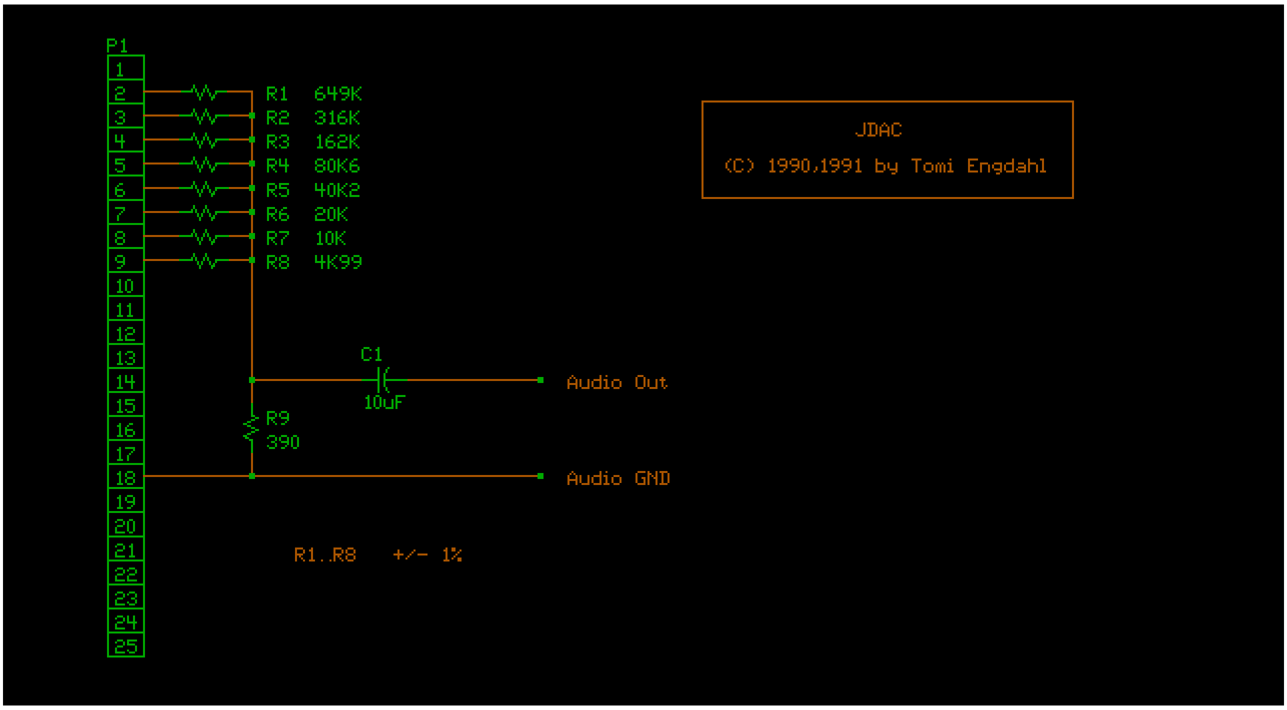
Dongles should be designed to operate all the way down to the minimum TTL low or at least to minimum Vout High of 2.4V. If your dongle doesn't work but your printer works fine then it is almost certainly the fault of the dongle--not the parallel port. Software with an incompatible dongle to the parallel port on a machine will not be usable on that machine--one more reason to not penalize the legitimate buyers of software.

All parallel port powered devices are oushing the pecs in the way or another. Usually those product work in "standard" computers on their own very well, but when more that one of them is used at the same time you can expect problems, because if the parallel port can barely power one device, then two devices is definitely too much for it.

Sound output devices

The most common parallel port powered device is simple parallel port DAC circuit. First this kind of commercial product was Covox Speech Thing, but shortly after introduction of that product many computer hackers designed their own circuit using the same basic idea.

The circuit design in all this type of circuit which do not need external power is based on some sort of resistor network which is driven directly from parallel port data pins. In this way the parallel port automatically powers D/A stage of the circuit which gives the output voltage directly. The maximum current the circuit takes is about 1 mA from datapin D7 and the total power consumption is less than 2 mA. This is well in the range what the parallel port can easily supply without problems.



Here is another popular version of the parallel port DAC circuit:

```
Paralel Port:

signal    pin        20k   20k
D0        2       >---###-+-###--0v
                     20k # 10k
D1        3       >---###-|
                     20k # 10k
D2        4       >---###-|
                     20k # 10k
D3        5       >---###-|
                     20k # 10k
D4        6       >---###-|
                     20k # 10k
D5        7       >---###-|
                     20k # 10k
D6        8       >---###-|
                     20k # 10k
D7        9       >---###-|
                         # 10k  100nF
                         +--------][----> To amplifier
                         # 10k
GND       20      >--------+------------->
                         0v
```

Telecard readers

What you need to know about electronics telecards document written by Stephane Bausson presents the simple way of reading electronic telecard using PC. The telecard itself need power of about 5V at 5 mA maximum and the power taken from the rest of the electronics is very little. The circuit takes the power from datapins D3-D7 through diodes. Using this method the telecard gets a little over 4V voltage which is enough for it to operate. This method is not very reliable way of getting power and newer versions of the reading circuit uses external power supply for more reliable operation.

Caller ID modules

In Finland the caller ID signal is sent (if you have subscribed the service) as DTMF tone before or between telephone ring signals. The DTMF signals can be easily decoded using suitable chip like MT8870C-1 from Mitel. The recommended operation range for MT8870C-1 is 4.75V to 5.25V and it takes typically around 3 mA current, which makes it quite suitable for being powered from parallel port.

```
                                       +---------+--------+--+-D1-<+5V
>-C1--R1-+---------+     MT8870C-1     |         |        |  |
         |         |    _____________  |         C3       |  +-D2-<+5V
         |         +----1          18--+         |        |  |
>-C2--R4-?--+--+--------2          17------------+        |  +-D3-<+5V
         R3 R2 +-R5-----3          16------R6----+        |
         +--+-----------4          15---->StD             |
                        5          14---->Q4              |
                        6          13---->Q3              |
               +--------7          12---->Q2              |
               +-X1-----8          11---->Q1              |
                    +---9          10---------------------+
                    |   _____________
                  Ground
```

In this circuit the power for the IC is generated from PC parallel port data pins D1-D3. The power from datapins is fed through diodes D1-D3 (type BAT42, BAT50 or similar low drop diode) to +5V power input of the IC. The driver software makes sure that parallel port data pins D1-D3 are set to logic 1 when the circuit is used to give the power. The circuit is designed by Per Elfstrom (elfstrom@siilinjarvi.fi) and I show here only the parts which are needed to understand the powering method of the circuit. The whole schematic, software and circuit description (in Finnish) is available at http://www.helsinki.fi/~metsala/cid.html.

Parallel port extenders

From: geo@netcom.com (George Pontis)

"Another area that might be of interest in your document would be some comment on the parallel port extenders. I have a xmit/rcv pair from LinkSys that I bought from Fry's Electronics for about $70. They convert the parallel signal to a serial data stream, using the signal and control lines for power. My set was working fine until I added a hardware dongle for an expensive Windows application. Then, printing ceased to work reliably. I took the transmitter apart and partially traced the schematic. They have used 7 diodes to suck power from pins 13, 14, 15, 17, 1, 2, and 3. Also, they connected pins 15 (ERR) to 16 (INIT). The strobe line is coupled in to a flip-flop, which starts clocking the parallel loaded data."

Other devices

Other devices which are totally parallel port powered are the parallel port multiplexers sold with Xircom network adapters. This multiplexer enable to connect the network adapter and prnter or two printers to one parallel port. The multiplexer device takes power from parallel power and includes some very low-power TTL chips which make all the work. I have not exactly figured out how this thing works and gets enough power, but anyway they exist and they usually work as promised. My assumptation is that the multiplexer also uses power available from the handshaking signal lines which are controlled by the external devices, because there is always some externally powered device connected to the multiplexer.

How much power can parallel port source

Comparision of different parallel port types

	Normal	UM82C11-C	IEEE 1284 level II
Data output (>2.4V)	2.6 mA	2 mA	14 mA
Data line sink (<0.4V)	24 mA	24 mA	14 mA
Control output (>2.4 V)	0.5 mA*	1.5 mA	?
Signal lines (short circuit)	1 mA	?	?
Control line sink (<0.4V)	7 mA	7 mA	14 mA

* This value is caluculated from the known data available from port.

Ordinary parallel port

The ordinary PC parallel port was based on discrete LS-TTL chips and it's performance figures are determined of the components used. A good schematic of PC parallel port is available in documents THE TYPICAL SCHEMATIC OF ORIGINAL IBM PC PARALLEL PORT by Richard Steven Walz.

Data lines

In ordinary parallel port implementations the data outputs are 74LS374 IC totem-pole TTL outputs which can source 2.6 mA and sink 24 mA Usually the output of the chis is followed by 22 ohm resistors and 2.2-10 nF capacitor, but some implementations don't use that RC circuit. The outputs are resigned so that they give at least 2.4V at 2.6 mA load. This 2.6 mA figure is for ordinary LS-TLL circuits used, the LSI implementations used in many computers can gvie more or les,. For example quite popular (few years ago) UM82C11-C parallel port chip can only source 2 mA.

```
         +5
        /    |
    ---H on   V
        \      -->
         |_________         TTL output on = 1 = high, "sourcing" current
         |   out   |
        /          Load (up to 2.6 mA @ 2.4 v)
    ---L off       |
        \__________/
        Gnd
```

If you set the outputs to high state and don't load the outputs, you get almost full +5V output. If you start to take current out of the port, the voltage will drop because of the resistances inside the parallel port electronics. The voltage is guaranteed to to stay abode 2.4V when you load the output as specified. If you try to pull more current from the port, the output voltage will drop and the parallel port electronics will start to heat up more than they are specified. Because of the different impelementations, the port might be able to give much higher than 2.4V at the full load (for example 4.75 " 2.5 mA load in modern port as specified in Circuit Cellar Ink magazine article), but you can't trust that every computer gives higher than 2.4V.

Standards called EPP (Enhanced Parallel Port) and ECP (Extended Capabilities Port) brought many enhancements. The most important one is the possibility of bi-directional communication over the data pins D0-D7, due to modified hardware design of these pins. The only difference between ECP/EPP and the "normal" SPP is, that the transistor pulling high has been replaced by a resistor (it's supposed to be 4700 Ohms, according to the standards).

```
         +5
          |
       4.7 kohm
          |
          |________  TTL output
          |   out
         /
    ---L off
         \
          |
         Gnd
```

Therefore, an ECP/EPP pin can be set to "read mode" by setting it to 1, so the transistor pulling low is open (non-conducting) and the actual logical level on the pin can be read. This system is backward-compatible with SPP in most cases; some difficulties do arise, for example you can't get much power from the data pins.

Control outputs

You can also use the Control Out pins. They can't source much of anything (about 1 mA through the 4.7K resistors to +5), and can only sink about 7mA. (The LS TTL gate actually sinks 8 mA, but one is taken up by the 4.7 K resistor to +5).

Again, check on clones with different electrical specs. For example very popular (few years ago) UM82C11-C chip implements the control outputs as standard TTL outputs with lower (1.5 mA) current capacity instead of traditional open collector circuit with pull-up resistors. 82C11 os quite popular IC for implementing parallel ports few years ago, but it was very sensitive to ESD and CMOS latchup effects. After facing problems with this the manufacturer started to use other solutions (nowadays the whole I/O card is usually integrated to one chip).

IEEE 1284

The original parallel port did not have a defined electrical specification that identified the driver, receiver, termination and capacitance requirements in order to guarantee any compatibility between devices.

Host adapters and peripherals were built with any number of pull-up values on the control lines, open collector or totem pole drivers for the data and control lines, and most offensive of all, up to 10,000pF capacitors on the data and strobe lines. This type of design variation makes it impossible to create a new interface protocol without explicitly defining the required electrical parameters with which to guarantee operation.

The 1284 standard defines two levels of interface compatibility, Level I and Level II. The Level I interface is defined for products that are not going to operate at the high speed advanced modes, but need to take advantage of the reverse channel capabilities of the standard. The Level II interface is for devices that will take advantage of those high-speed capabilities.

IEEE 1284 Level II interface drivers must be able to source 14 mA current (at least +2.4V voltage) and also sink 14 mA current (output voltage lower than 0.4V). The output impedance in normal operation range is defined to be 50+/-5 ohms.

How to use the power available from parallel port

Because the power capabilities of the parallel port is very limited, the circuit you vant to connect to parallel port must be designed so that they take as little as possible power. This is possible by using small number of low power logic chips and running then at low clock frequencies.

Directly taking the power from parallel port to power 5V logic

One method often used for powering small circuits is to use the power from data output pins directly to power the external IC. If you have one IC which takes less than 1 mA current, you can easily power it directly from one data output pin. You get around 4.5-5V to the IC very easily in this way just by using one data output pin. You can expect reliably draw about 0.5 mA from each data line which take total of 4 mA current.

If you need a little more power, you have to take the power from many data pins of the parallel port. Directly connecting all data output pins you wish to use to the +5V power rail of the circuit is not a good idea because this will cause problems. If you accidentally pull down one of those data pins you use for supplying the power, it will short the other outputs to ground which will overload the parallel port output chip. Also if you have parallel port datapins directly tied together, the port will likely fail in power-on shelf-test done every time the computer is started.

The more sophisticated way of getting the port from many data pins is to use diodes which enable only the current from the parallel port to come to the circuit, not in the other way. Those diodes will cause voltage drop (around 0.5V in silicon diodes at those current), which is what you want because you have already a slightly low voltage for your circuit. The voltage drop in the diodes can be minimized using schottky type diodes (voltage drop of around 0.2-0.3V). You might also consider using germanium diodes, but those have higher internal resistance and they are getting rare.

Very typical circuit which takes power from PC parallel port data lines through diodes is the telecard reader circuit presented in What you need to know about electronics telecards document. If you parallel the 8 data lines, you can reliably draw about 0.5 mA each. Due to the diode voltage drop in power paralleling circuit, you would be lucky to get 4-5 mA at 4.5V. There are 5 control outputs from which you can reliably draw 1 mA each. That comes to around 9 mA. If you can hold all outputs high, up to 50mA seems possible but the voltage would sag a lot. If you use switcher for generating the operation voltage for your circuit, then you can take out the maximum amount of power from printer port in this way.

In general, the more current and the higher the voltage your circuit requires the less likely it is to work on a variety of different parallel ports. Taking power directly form parallel port work well to fire up a few CMOS ICs that will work down to 3V (or less) and draw only a few mA.

Powering low voltage logic

If you design your circuit to reliably operate at voltages from 3 to 4.5V you have no problem in getting right voltage from the parallel port. You have plenty of marginal for voltage drops in diodes.

Generating +5V from power available from parallel port

If your circuit absolutely needs stable +5V power source, you have no other choice than to use a DC/DC converter to generate stable +5V voltage from the power it can take from the parallel port. Usually ready-made DC/DC converter modules are not suitable for this because they are designed mostly for bigger loads. What you are looking for is a very small current switching power supply which can supply +5V output from input voltage range of 2.4-4.5V and uses the power economically. Suitable components are those small single-chip miniature switches designed for small battery powered devices. Maxim has wide selection of suitable components.

Pinout of PC parallel port

```
<= in   DB25   Cent    Name of          Reg
=> out  pin    pin     Signal           Bit     Function Notes
------  ----   ----    ---------        ---     ----------------------------
--
=>       1      1      -Strobe          C0-     Set Low pulse >0.5 us to
send
=>       2      2      Data 0           D0      Set to least significant
data
=>       3      3      Data 1           D1      ...
=>       4      4      Data 2           D2      ...
=>       5      5      Data 3           D3      ...
=>       6      6      Data 4           D4      ...
=>       7      7      Data 5           D5      ...
=>       8      8      Data 6           D6      ...
=>       9      9      Data 7           D7      Set to most significant
data
<=      10     10      -Ack             S6+ IRQ Low Pulse ~ 5 uS, after
accept
<=      11     11      +Busy            S7-     High for Busy/Offline/Error
<=      12     12      +PaperEnd        S5+     High for out of paper
<=      13     13      +SelectIn        S4+     High for printer selected
=>      14     14      -AutoFd          C1-     Set Low to autofeed one
line
<=      15     32      -Error           S3+     Low for
Error/Offline/PaperEnd
=>      16     31      -Init            C2+     Set Low pulse > 50uS to
init
=>      17     36      -Select          C3-     Set Low to select printer
==      18-25  19-30,  Ground
              33,17,16
```