Checking a new PiDP-8

20160227

This assumes you powered up your PiDP for the first time, you see some lights come up and you wonder if all of them work. First comes a quick diagnostic, then an explanation of the how and why.

# Basic Check

Power up the PiDP and wait until you see the OS/8 command prompt. Assuming you did not do the serial mod yet, check that it says “Rpi [x] detected – Default version”. Not serial version. You get the wrong leds light up if you inadvertently run the serial-mod version of the pidp8 program.

1. Hit the STOP switch. Blinking freezes.
2. Set all the SR switches (the first 18 switches starting from the left) to 1. Meaning you press them down.
3. Toggle the LOAD\_ADD switch.

You should now see all the LEDs on the PC row (the top row) light up.

If not:

If one or more of them does not light up, there is a bad connection somewhere in the column wiring. But it could be a led not working, or the corresponding SR switch not working. If the entire PC row of 18 leds does not light up, there is a problem with the ledrow wiring, probably involving the 2981 IC. If only the leftmost 6, or only the leftmost 12 leds do not light up, it is a problem with a particular ledrow line. Explanation below. Continue the test below.

1. Hit the DEP switch. This will copy the PC row, which you just filled with 1s, into the MB register, the 3rd row of leds. It will also light up all the MA leds, the 2nd row.

So you should see the MA and MB leds all lit up now. All 12 of them, on both rows.

If not:

It may be that one of the leds on the MA and MB lines is not working. If the led on the PC line directly above the unlit MA/MB led \*was\* lit up, you know it’s a problem with this particular unlit LED not being connected correctly.

If the led on the PC line directly above the unlit MA/MB led \*was not\* lit up either, it may be that the SR switch was not working. That it did not toggle a 1 into the PC register, but remained at 0. The leds may be fine, they just show a 0. Alternatively, the switch may be OK, but a bad connection in the column wiring may have disabled the three of the leds above it. Which of the two problems do you have? Hit CTLR-E to find out. It brings you in the command line of simh. Enter “ex ma”. If it does not show “1111”, your switch is the problem. If it does, then the switches worked fine but a led just does not get powered to show it.

# Understanding what’s wrong

From here, first an explanation on how the PiDP-8 circuit works with regards to the leds. The switches come later.

It is logical to worry if you are harming the Pi by running it with a defective board. But it is extremely unlikely. The only known casualty so far has been one of my Pis, when I put the serial connector (which also can provide power) in upside down. The two serial port pins died. For the rest, if you’ve come this far, the Pi pins are sufficiently protected by resistors and the IC for it not to suffer whilst you are debugging the board. Lethal problems could be if the 5V power line shorts with the 3.3V power, or directly shorts with a 3.3 gpio pin.

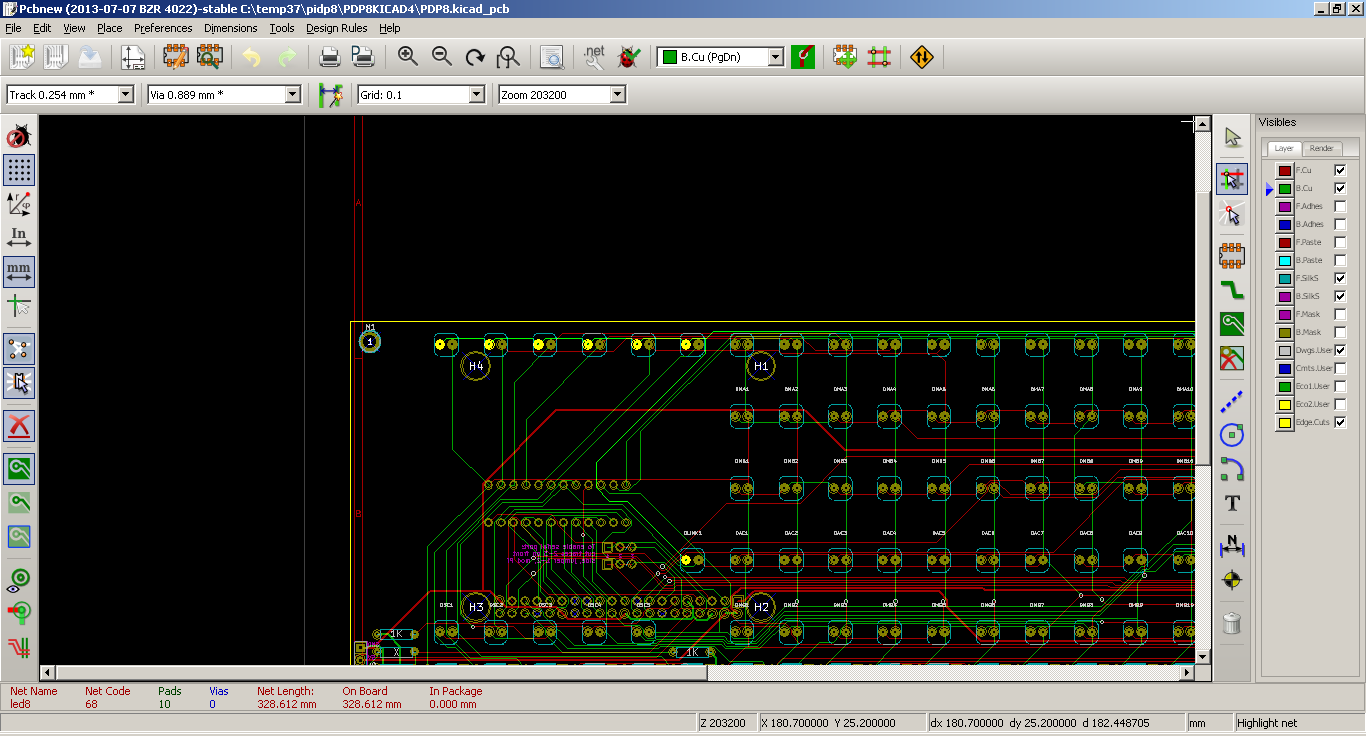
You have 8 ledrows that provide power to 12 leds each. Ledrow 1 is the 12 PC leds, ledrow 2 is the MA line, ledrow 3 the MB line. 4 the accumulator, 5 the MQ register. That’s all logical. But ledrow 6 is the first 12 vertical leds (AND to Word Count). Ledrow 7 is Current Address to Run. And, it also powers the 5 leds of the Step Counter. Ledrow 8 does the Data Field and Inst Field leds.

So if a group of leds remains dark, see if it corresponds to one of these ledrow groups. Your problem then lies with a ledrow connection.

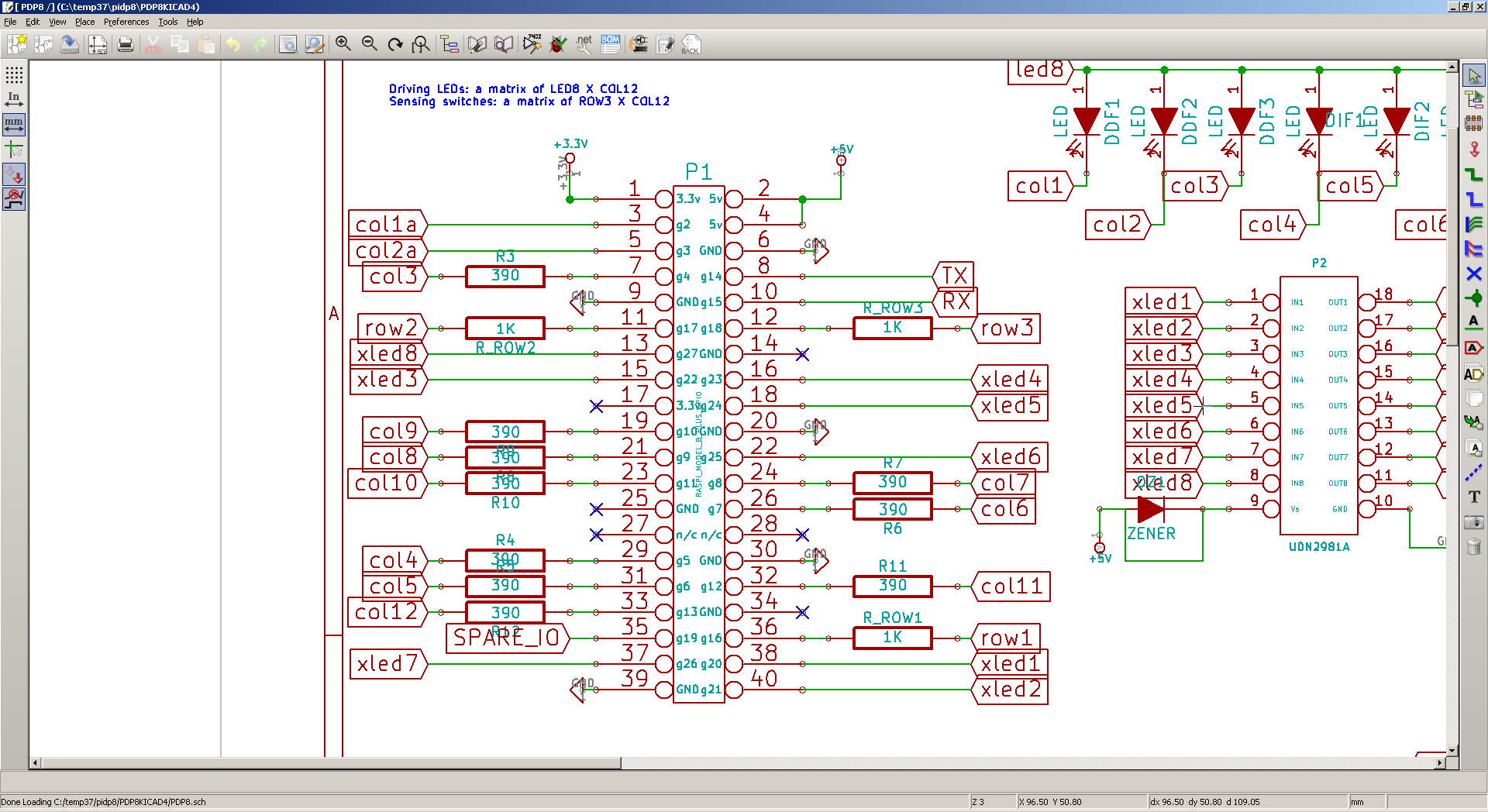
The ledrows provide the +, 12 column lines provide the – for the leds. If a ledrow pin is set to High, and a column pin to Low, that particular led will light up. All the PiDP circuit does is set each ledrow pin to High for a while, and set the column pins to Hi if a particular led is to remain dark, or to Lo to provide a current sink and thus, to light up the led.

The whole PCB basically runs traces from point to point. That makes it easy to see where you see a signal, and where you don’t anymore because of a bad solder point (95% of problems) or a broken trace (so far, 5 times in 700 PCBs but it does happen). The third explanation is a dead 2981 IC (five times so far).

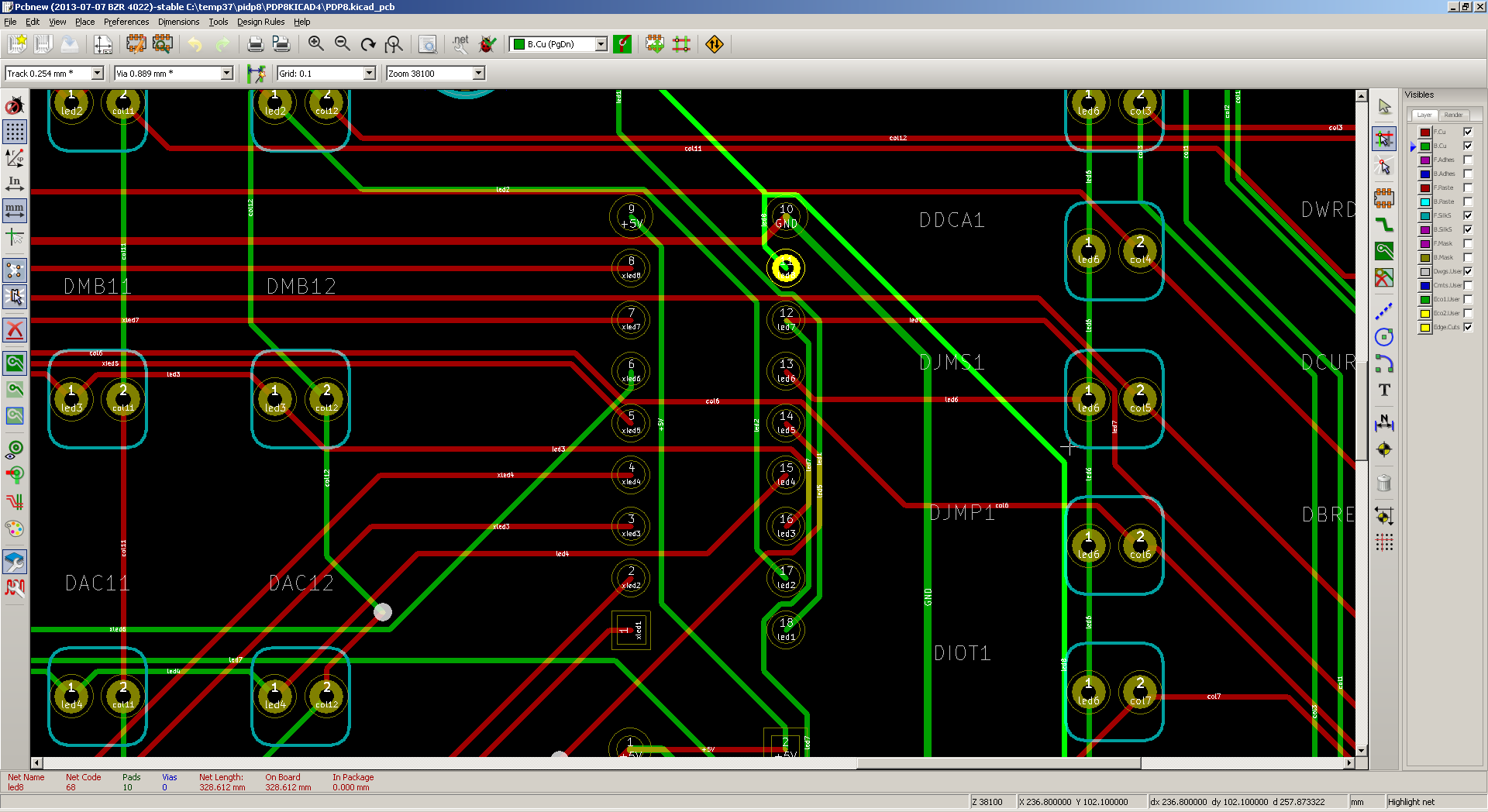
# Debugging the board

For debugging, install Kicad on your computer and load the Kicad project file from the web site’s download section. The schematic (and tracing) is virtually identical across all PiDP-8 boards. Load up the project file in Kicad’s PCBnew editor (3rd icon). You’ll see the PCB. Look for this icon, and click on it. Now, any trace you click on will get highlighted so you see where you can use a multimeter to check if the connection really is there on your PCB.

If you zoom in far enough, you will see each solder point labeled with ledx (meaning, ledrow) or colx. Take a step-by-step approach (no need to remove the Pi) and start at the source - the led or col pin on the gpio connector. See picture below (looked at from in front of the board, with the leds in sight and the switches down below).



Col traces run through a 390 ohm resistor, and then to all the leds. Ledrow traces go straight to a pin on the left-hand side of the 2981 IC and are called xled. On the right-hand side of the IC, a buffered led signal comes out and runs to the rows of leds on the rest of the PCB.

Multimeter your way through the path of the ledrow (led) and col traces and you’re bound to find what is wrong. Zero ohms or infinity? Most of the time, it’s resoldering a pin here or there. It could also be a broken trace on the PCB, which is pretty bad but has occurred. The solution is simple: solder a jumper wire from the point where you last had good contact, to the solder point where there is no contact anymore. But triple-quadruple check yourself. Because maybe you made a mistake, and you are checking the wrong solder points? Mirror-image confusion, miscounting? A wrong jumper wire can do Bad Things.

If the above does not identify the problem, it is time to suspect the 2981 IC. Power up the PiDP, and measure if signals come in to its left side, the input pins. Looking at the board from the front (where the leds are mounted), here is the pinout:

Of course, the signals will be pulsed 3.3V, so your multimeter will read a much lower voltage as it averages out over time. But the point is: 0V or V? 0V, if the trace to the gpio connector is good, means a damaged Pi[[1]](#footnote-1). You’d be the first.

Then, the question is whether you get the signal out of the IC on the pin on the opposite side. It will be higher voltage.

No voltage coming out means a dead 2981 IC. I must send you a new one. You can verify it is the IC’s fault by briefly shorting the dead output pin to the good input pin with a wire. Or a resistor. With a wire, it puts a not-OK load on the Pi but it’s been no problem in practice. You should see a row of dead leds come to life with the Pi running. Diagnosis is clear now.

# Testing with wiringPi

WirinPi is a software package that contains the very useful gpio tool. Google for details. With the gpio program (and pidp8 not running of course) you can set any ledrow pin High, any col pin Low and the corresponding led should light up. The write-up may be short, but this is the best debugging tool you have. Try it!

# Debugging the switches

All of the above concerns the leds. The switches are also in a matrix. To read the switches, the col pins now are set to Input-with-internal-pull-ups. Then, there are three row pins on the gpio, which can be set Low to read a row of up to 12 switches. If a switch is set (pressed in the down position), it will cause a short between the col pin and the row pin, driving the input pin to read a Low signal as its weak internal pull-up is trumped by the power drain of the row line.

Row 1 is connected to the 12 SR switches. Row 2 deals with the six leftmost switches (Data and Inst Field). So only 6 switches in this row. Row 3 deals with the rightmost 8 control switches.

With a multimeter, you will get confusing readings on switches because there is a diode in-between the switch and the col line. So measure continuity (ohms) from a row pin on the gpio to one end of the diode, and from the other end of the diode to the switch. Should be all 0 ohms.

From the row pin on the gpio, the signal goes through a 1K ohm resistor to protect the Pi, and then on to the middle pin of a row of switches. See where it goes wrong.

Special case: the row2 trace winds its way in-between the switch solder points to the expansion connector. In one case, it shorted out somewhere along the way with another trace. If you have problems with the leftmost switches and you cannot figure it out, maybe cut that long trace just below the IF3 switch at the back of the board. You can always use a jumper wire to re-establish the contact – and nobody is using this line on the expansion port…

Again, next to measuring continuity of connections with a multimeter, you can use WiringPi to set one of the row columns to Low, the other two to High. Set all 12 column pins to input-with-internal-pullup, and you should see which switch you are setting in gpio (gpio, as in the programwith that name).

# Other annoying problems

If youhave not fixed the problem by now, you will be frustrated. Here is what could be wrong otherwise:

* Is a LED defective (never happened, but still)? Put a battery with 1K (really, anything from 200 ohm to 1.2K) resistor in line to its pins on the board. Better remove the Pi first. Does it light up?
* Is your Pi defective? It can be that just one or two gpio pins are dead. Test with WiringPi, flipping it from High to Low (read on a multimeter) to Input (set it to input, connect it to GND and 3.3V consecutively, ideally with a resistor in-between, and see the gpio program’s output)
* Is a switch defective? Just short its top and middle pins with a wire and see if it is On then).
* Did you compile the pidp8 program with serial-mod enabled whilst you did not do the serial mod?
* Is your power supply OK (people always talk about this, but truth be told, a Pi runs off a laptop USB port so not enough amps is very very unlikely to be the problem. But bad power could be).
* If you run the Pi with pidp8 when removed from the board, do you see some voltage (not 3.3V, it’ll be much less on a multimeter) on all the row, column and ledrow pins?
* Any LED, diode put in the wrong way? The chip put in the wrong way (upside down, wrong side of the board)?

So far, all PiDPs have been brought to life. But I am jinxing it by saying so. If you still have not fixed your problem, email me or use the forum.

Clearly, an oscilloscope would make diagnosis much easier if you have one. If you do, be aware that the signal out of the 2981 looks much less clean than you would expect. It is a pretty primitive Darlington array. But a not-so-clean signal is OK, and a wrong signal is, well, wrong.

1. This assumes that the pidp8 program is running of course. Otherwise, you’d see 0V on the input pins. [↑](#footnote-ref-1)