PUMPKIN™
REAL-TIME SOFTWARE

AG-1
Assembly Guide

750 Naples Street • San Francisco, CA 94112 • (415) 584-6360 • http://www.pumpkininc.com

Assembling the SSDL/SCU PICmicro® Protoboard

Introduction

Pumpkin's SSDL/SCU PICmicro Protoboard ("protoboard") is an enhanced replacement for the existing PICmicro prototyping PCB that Stanford University's Space Systems Development Laboratory (SSDL, http://ssdl.stanford.edu/) and Santa Clara University's Santa Clara Remote Extreme Environment Mechanism Laboratory (SCREEM, http://screem.engr.scu.edu/) have been building up by hand from the PICProto64 prototyping board from microEngineering Labs, Inc (http://www.microengineering-labs.com/).

CAUTION Please review *Special Instructions for REV A Protoboards* below before assembling any protoboards. This will ensure their correct operation.

The protoboard is a two-sided printed circuit board (PCB), measuring 4.00" x 4.00" x 0.062". It has the following enhancements over the PICProto64:

- additional benchtop power connectors via banana jacks
- dedicated 9V battery clip (vertical-mount)
- three different power sources (bench, battery and system), switch-selectable
- improved microcontroller reset circuitry built-in
- two RS-232 channels built-in, with MAX233 driver
- two serial SPI EEPROMs built-in
- real-time clock crystal built-in
- I^2C pull-ups built-in
- DB-9 connectors used throughout for system power, system data, hardware USART (RS-232) and software USART (RS-232)
- improved power supply decoupling and distribution
- improved grounding and noise immunity

created by Andrew E. Kalman on Aug 13, 2000 updated on Oct 17, 2000

- distributed test points for critical voltages
- a 12 x 15 hole breadboard area on 0.100" centers with GND and +5V rails

CAD Genesis

The protoboard was designed in OrCAD SDT v4.1,[1] and laid out in PADS-PCB and PADS-2000. The PCB has been assigned a Pumpkin part number of 705-00167, and a project number of 90002. It is currently at Revision A (REV A).

The relevant files are:

- `90002A.sch`: OrCAD schematic
- `Pumpkin.lib`: OrCAD library
- `Pumpkin2.lib`: OrCAD library
- `90002A.asc`: PADS-PCB v4 ASCII job file
- `90002A.job`: PADS-2000 job file
- `90002A.net`: Futurenet netlist
- `90002A.zip`: all PCB CAD files, .zip'd
- `90002Apd.zip`: schematic and PCB verify plots, in Acrobat (.pdf) format, and Bill of Materials
- `90002Aps.zip`: PCB verify plots, PostScript format

Circuit Description

Please refer to `90002A.pdf` in `90002Apd.zip` to view the protoboard's schematic. A general description, enhancements over and differences between the protoboard and the earlier PICProto64-based version are described below.

The *CPU* U1 is specified as a Microchip® PIC16C77. Other compatible PICmicro devices in a 40-pin DIP package (e.g. PIC16F877, PIC16C65, PIC18C452, etc.) may also be used.

The two *Oscillators* and the *Serial (SPI) EEPROM* are implemented in a conventional manner.

The *Reset Circuitry* uses a MC33164P-5 micropower reset supervisor to guarantee proper startup and reset on brownout conditions. The 10kΩ value for R5 is recommended to avoid the possibility of PICmicro latch-up.

All of the CPU's pins are available on the *Expansion Connector* H3. OSC[1..2] should not be used by any external devices.

RC[0..2], RC[5..7] and RB[1..5] may or may not be available to external devices, depending on which components have been stuffed on the protoboard.

The *DPDT Power Switch* and *Power Conditioning* control power to the protoboard. In the *center position*, all power is off. In the *up position*, power comes from either the bench power supply (at J5 and J6) or the 9V battery (at B1), depending on which is greater.[2] In the *down position*, power comes directly from the system power bus through the DB-9 connector H1 and bypasses the regulator U3. LED D3 will light whenever power is on, regardless of the source.

The LM2931(A)Z-5.0 (TO-92 package) and LM2940(C)T-5.0 (TO-220 package) +5V fixed voltage regulators are automotive-grade parts which, among other things, have low quiescent currents, tolerate excess V_{in}, reverse hookup and shorts on the output, etc.

The MAX233 *RS-232 Transceiver* U2 implements two RS-232 DTE (data terminal equipment) ports. Hardware handshaking is not supported. Since these DTE ports are the same (pinout- and connector gender-wise) as is commonly encountered on a PC, you will need a *null-modem cable* to connect to a PC.

Connector H4 is connected through U2 to U1's built-in high-speed USART,[3] making this a *Hardware RS-232 Port*. This port is commonly used for *debugging*.

Connector H5 is connected through U2 to two general-purpose I/O pins on U1. This means that in order to implement RS-232 through this port, a software UART must be written to run in U1. The speeds of a software implementation are usually much lower than those of a hardware implementation. Therefore this port is a *low-speed Software RS-232 Port*. In order to use this port, jumpers J1 and J2 must be connected. *Black* shorting blocks are provided for these jumpers. If you will not be using this port, leave these jumpers disconnected.

10kΩ *I^2C Pull-Ups* are provided via jumpers J3 and J4. Only one device per I^2C bus should have pull-ups enabled. *Blue* shorting blocks are provided for these jumpers.

R7-9 provided a small measure of *noise immunity* for the three communication ports.

R10 and C8 form a *Noise Filter* between signal ground (GND) and chassis ground (the shields on the DB-9s).

Test Points TP1-9 are provided for DC voltages throughout the protoboard.

Decoupling Caps are provided for all of the digital circuitry.

A *Breadboard Area* is provided for additional on-board prototyping. GND and +5V *Power Busses* are provided on the left and right sides, respectively, of the breadboard area. H3's (and therefore U1's) pinout is labeled on the solder side of the PCB to facilitate connections between the processor and the breadboard area.

Mounting Holes are drilled on all four corners of the PCB. They will accept 4-40 fasteners and up to .250" diameter mounting hardware (e.g. standoffs).

PCB Description

The PCB is a conventional double-sided board with 12-, 25- and 50-mil traces on a 25mil grid with 12mil clearance rules. Only through-hole components are used. Signals are routed with 12mil traces and 45-degree bends, whereas power and ground[4] are routed with 50mil traces and orthogonal bends. Additionally, no trace may enter a pad with greater than 25mil width.[5]

A *star-grounding* scheme is employed, with all ground traces emanating from the "-" terminal of the 9V battery.

Some *additional text legends* have been laid down on the copper layers in case the board is fabricated without soldermask and silkscreen.

Two large *silkscreen pads* are provided for writing the serial number and a modification code directly on the PCB with a compatible marker.

Stuffing the PCB

Please refer to `cover.pdf` contained in `90002Apd.zip` to view the PCB with silkscreen part outlines and component designators. There are a few isuues to be aware of when populating the protoboard:

U1 is *socketed* to enable connecting the MPLAB-ICE in-circuit emulator. Stuff the socket, not the PICmicro!

H1, H4 and H5 are all *plug* / male (exposed pins) DB-9 connectors. H2 is a *socket* / female (no exposed pins) DB-9 connector. Do not interchange the two types!

Follow the *orientation* of all diodes D1-3, electrolytic capacitors C3 and C105 and TO-92 packages U3 and U6.

Stuff only U3 *or* U7 – not both. The holes for U7 (in a TO-220 package) are larger than, and are located above, those for U3 (in a TO-92 package).

Power switch SW2 wobbles in place before soldering – solder just a single terminal, and reheat if necessary to orient the switch in a proper, untilted vertical orientation. Then solder the remaining five terminals.

J6 is a *black* banana jack, J5 is a *red* one. These jacks are mechanically fastened (not soldered) to the PCB. To mount each one, remove both nuts and discard the solder tab. *The washer immediately under the plastic body must remain on the top (component) side of the PCB*. Also, unscrew the knurled upper body 2-3 turns[6] before proceeding. From the top side of the PCB, place the jack's screw through the PCB, screw one nut on from the back side of the PCB and tighten it down with a 5/16" nut driver – you *must* use Loctite® or a lock washer here or else each banana jack may work its way loose.

The battery clip B1 is unfortunately not keyed – it can be inserted in one of two orientations. The correct orientation is with the *smaller of the two terminals to the left*.

The correct orientations of the banana jacks and battery clip is shown in Figure 1 below:[7]

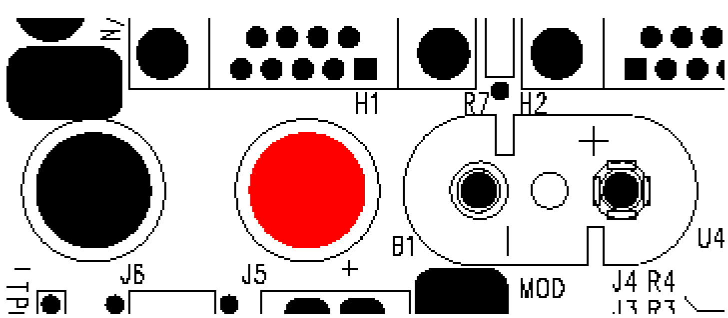

Figure 1: Orientation of Banana Jacks and Battery Clip

When you have finished soldering all of the components to the PCB, mount the 4-40 hardware (screw, lockwasher[8] and hex standoff) at each corner to give the PCB some "legs" to raise it high enough to clear the screws of the two banana jacks.

Special Instructions for REV A Protoboards

The REV A protoboards were made using Alberta Printed Circuits' (http://www.apcircuits.com/) Proto 1 protoyping service. Therefore these protoboards have neither soldermask nor silkscreen. Please note the *special instructions* below when stuffing these boards.

The *pinout* for the PCB parts decal for the reset switch SW1 is *incorrect*. If the PCB is left unmodified, the two pairs of terminals will be shorted together when the switch is soldered in place, resulting in a permanent RESET condition for U1. To correct this, *two traces have been cut on all* REV A protoboards. The location of the cuts is shown below:

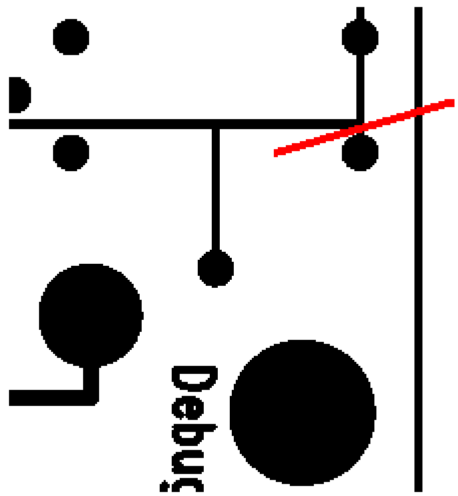

Figure 2: Cut Trace near SW2 on Layer 1

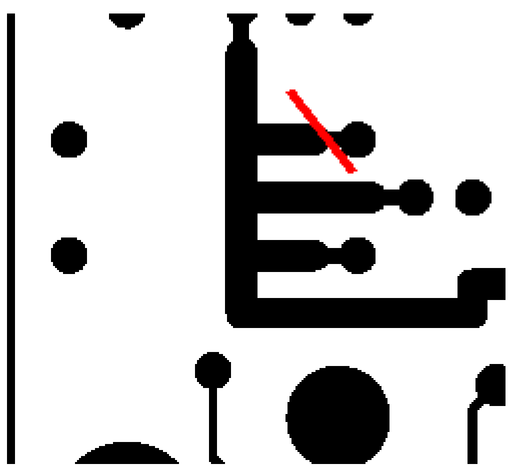

Figure 3: Cut Trace near SW2 on Layer 2

The PCB *holes* for the pushbutton switch SW1 are a *tight fit* – use a set of needle-nose pliers to *straighten* SW1's leads before inserting the switch into the PCB.

The PCB *holes* for LED D3 are a *tight fit*. By *cutting* the leads off "just before" their slight bulge, leaving approximately 0.100" total lead length, you will be able to *flush-mount* the LED and still have enough leads protruding on the other side to solder it successfully.

The PCB *holes* for battery clip B1 are a *tight fit* – make sure you have pushed it down flush with the PCB before soldering it in place.

There is a PCB *trace* on layer 1 directly beneath the body of crystal Y1. You must take care to mount Y1 *slightly above* the PCB (e.g. 0.050"-0.100") to *avoid shorting* its case to the trace beneath it.

Similarly, crystal Y2 should also be *mounted some distance above* the PCB. It *cannot be laid flat* because of the uncovered traces nearby.

Capacitors C1-2 and C6-7 have a lead spacing of 0.200", but the PCB expects *smaller* caps with a 0.100" lead spacing. Make two right-angle bends in the left lead of each capacitor to reduce the lead spacing to 0.100". By bending the left leads only, C1 and C2 will not touch Y1.

In Figure 4 below you can see how the C1's and C2's leads were bent, and how the crystals are mounted with some clearance above the PCB.

Figure 4: Bent Capacitor Leads and Crystal Mounting

We will endeavor to fix all of these errors on the REV B protoboard, if/when one is made.

Testing the Populated PCB

Once you have stuffed and soldered all the required components, you can test the complete protoboard.

Step 1. With U1's socket empty and SW2 in the center position, connect the protoboard to a lab benchtop power supply set for V_{out} = 6 to 20V_{dc}. 6V_{dc} should be adequate,[9] but if you have a 9V_{dc}, 12V_{dc} or 15V_{dc} supply that should work, too. Use test clip leads with banana plugs, black for "-" (GND) and red for "+" (+5V).[10] If your benchtop supply has a programmable current limit, set it for around 100mA.

Step 2. Turn the system on by moving SW2 to the up position. Touch each chip with your fingertip – nothing should be hot. LED D3 should light up. Check TP2 for +5V_{dc}, using TP3 for GND (0V). If you have anything other than +5V_{dc} on TP2 then you may have stuffed D2 and/or U3 backwards, or you may have stuffed another component (e.g. U4 or C3) backwards. If you have +5V_{dc} but D3 is not lit, it's probably soldered in backwards. A fully populated board (less U1) will draw approximately 30mA – if substantially more current is being drawn, you may have a problem.

Step 3. Check to see that U2 is functioning properly by testing TP1 and TP5 for approximately +10V_{dc} and –10V_{dc}, respectively – use TP4 for GND.

Step 4. Ensure that the reset circuitry is working by checking TP9 for +5V_{dc}. Press and release SW1 and observe the voltage dropping to 0V, and then back up to +5V_{dc}.

Step 5. If you have a 9V battery handy, you can test it by disconnecting the bench supply and snapping a 9V battery in place – you should also get +5V_{dc} at TP2 with SW2 in the up position. If not, verify the orientation of the battery clip B1 and diode D1.

Step 6. You can test the system power connector H1 by connecting a system power cable to it, moving SW2 to the down position and repeating the voltage tests above.

Further Testing: At this point all of the subsystems are probably working. Any further testing must be done with a CPU or in-circuit emulator.

Using the SSDL/SCU PICmicro® Protoboard

A typical usage scenario is one where the protoboard is powered from a bench supply, with H2 connected to the system I²C data bus, and perhaps with H4 connected to a terminal program running on a PC.

You can build additional circuitry on the protoboard by using the breadboard area. U7 (TO-220 regulator) can supply up to 1A for power-hungry components.

If you are using an external module (e.g. the *1.5MB SRAM Module*) you'll need to connect it to the protoboard via the expansion connector H3. External modules get all of their power from the protoboard – it's not necessary to hook power up to them separately.

The *Real-time Clock* and *Serial (SPI) EEPROMs* are already available on the protoboard if you want to use them. You'll need to use the appropriate code module to enable their functionality in your application.

If you want to use the *Software RS-232 Port*, be sure to connect jumpers J1 and J2. You'll need to configure U1's PORTC pin 2 (RC2) as an output and pin 5 (RC5) as an input via the PICmicro's TRISC register.

Schematics

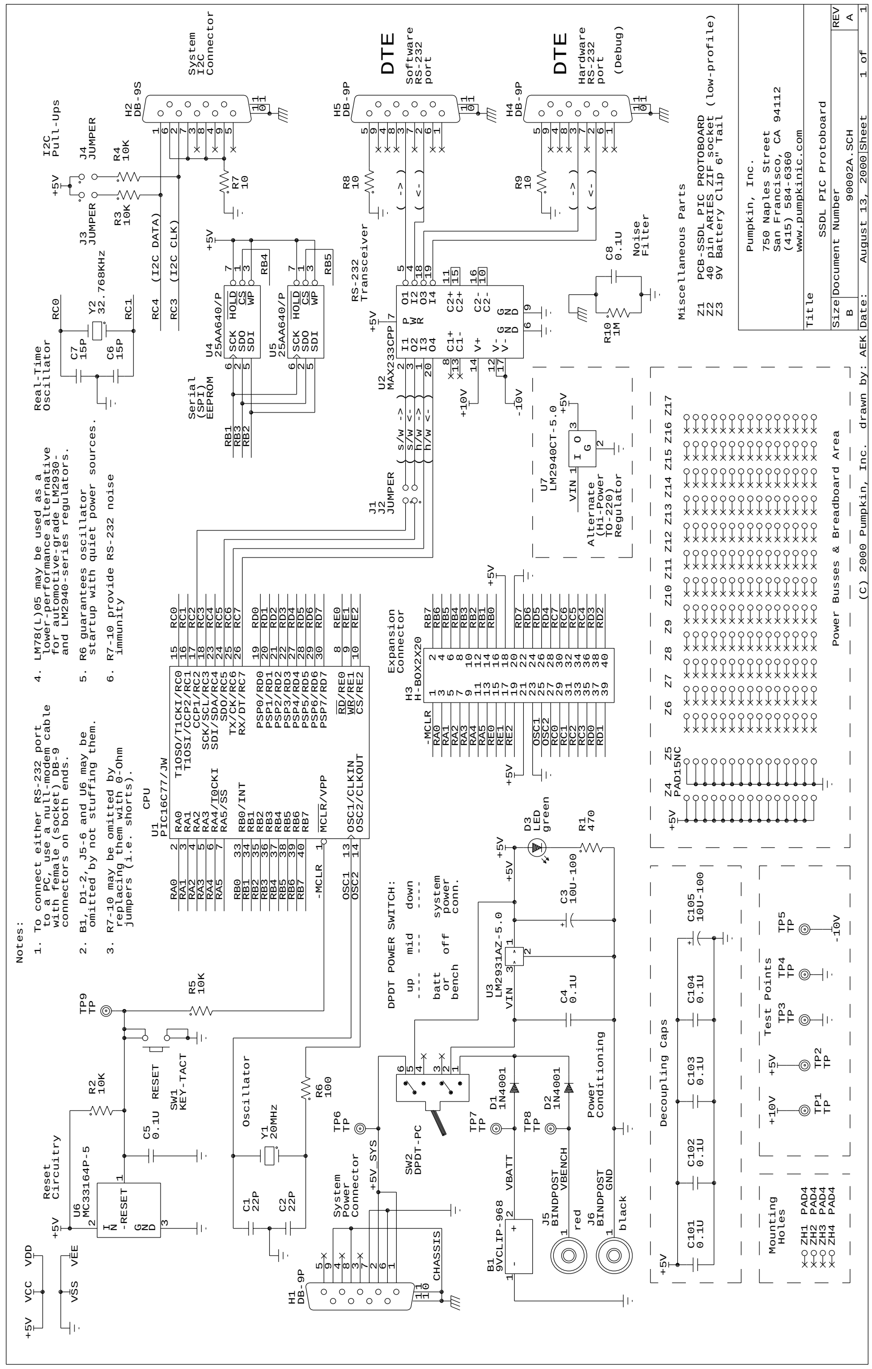

Figure 5: REV A Schematics

PCB Layers

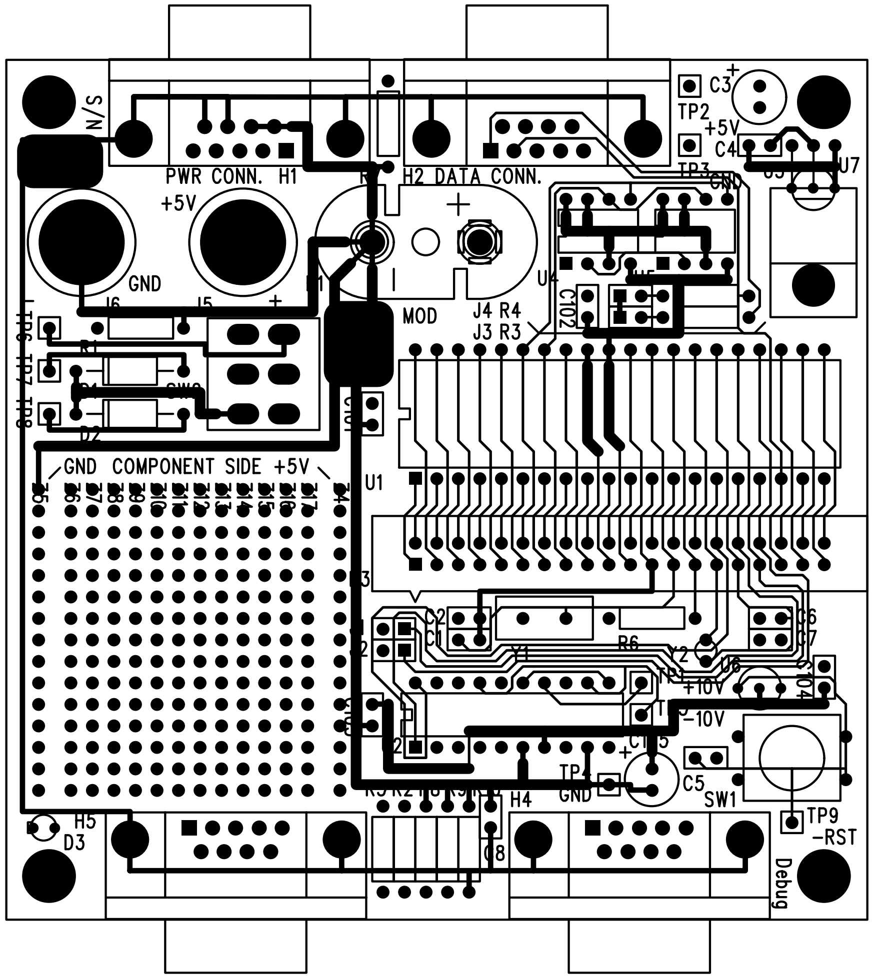

Figure 6: REV A PCB Topside Plot

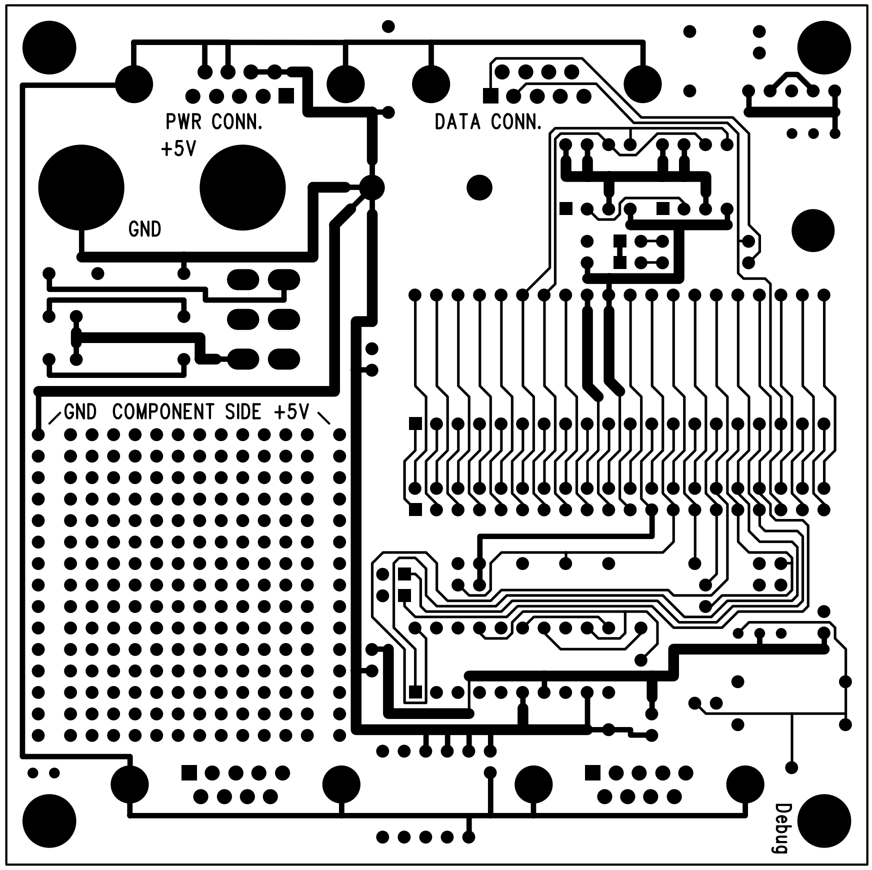

Figure 7: REV A PCB Top Routing Layer

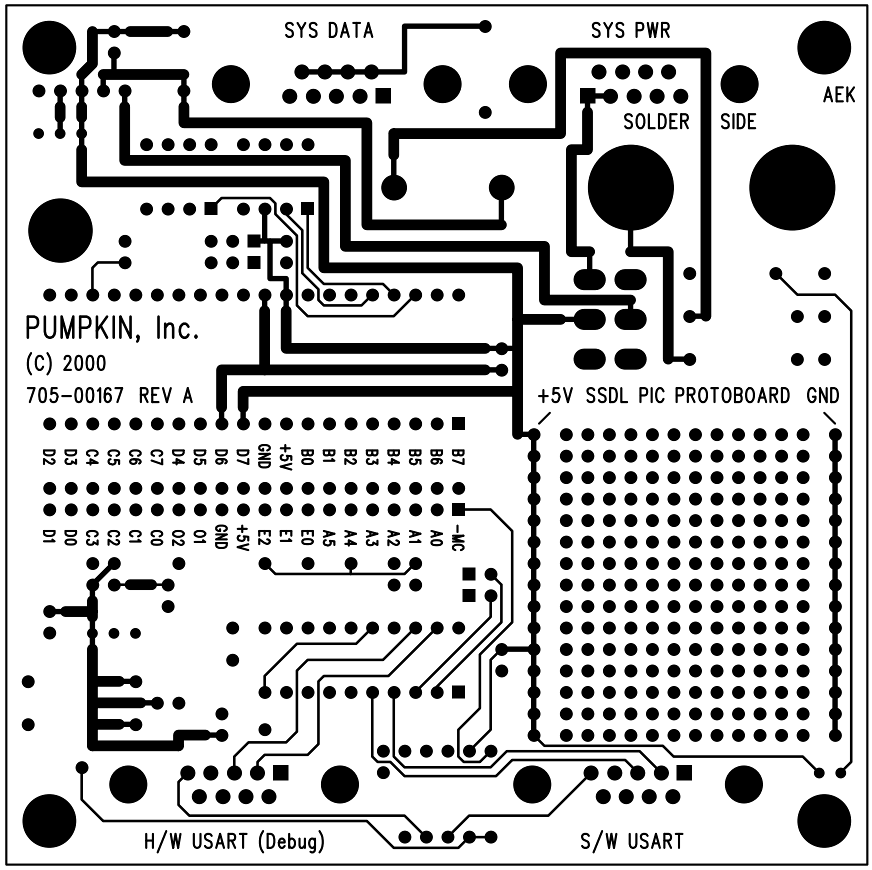

Figure 8: REV A PCB Bottom Routing Layer

[1] It may be ancient, but it's also reliable.
[2] That's what diodes D1 and D2 are for.
[3] Universal Synchronous/Asynchronous Receiver/Transmitter.
[4] Chassis shields are routed at 25mils on this PCB.
[5] This ensures that components can be desoldered – with 50mil traces entering a pad, the heatsink effect greatly reduces one's ability to successfully desolder the component without damaging the trace(s) or pad(s).
[6] This guarantees that the fastening screw that goes through the PCB is extended to its full length.
[7] J5 is shown in red.
[8] On top of PCB.
[9] This assumes you are using an LM2931- or LM2940-series low-dropout voltage regulator. If you have substituted a lower-performance part, like a 7805, you may need to raise the voltage to > $7V_{dc}$.
[10] On REV A PCBs the "+5V" label is in error – it should read "PWR".