PUMPKIN™
REAL-TIME SOFTWARE

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

**AG-5**

**Assembly Guide**

# *Assembling the SSDL/SCU PIC17 Protoboard*

## Introduction

Pumpkin's SSDL/SCU PIC17 Protoboard ("protoboard") is the second in a series of enhanced replacements 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.

**Note** The PIC17 Protoboard is derived from the SSL/SCU PICmicro Protoboard detailed in Pumpkin's *Application Guide AG-1 "Assembling the SSDL/DCU PICmicro Protoboard."* Both protoboards share many similar features.

The protoboard is a two-sided printed circuit board (PCB), measuring 4.00" x 8.00" 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
- $I^2C$ 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
- distributed test points for critical voltages
- a 12 x 15 hole breadboard area on 0.100" centers with GND and +5V rails

Additionally, this protoboard is designed around the Microchip® PIC17C756A[1] single-chip microcontroller (instead of the PIC16F877 and similar/compatible PICmicros). External memory can be used with the 17C756A when operating in microprocessor or extended microcontroller modes. External memory can be accessed by the 17C756A via the table read and table write instructions. This protoboard supports both external program memory (Flash) and external RAM with configurable memory maps.

## CAD Genesis

The protoboard was designed in OrCAD SDT v4.1,[2] and laid out in PADS-PCB and PADS-2000. The PCB has been assigned a Pumpkin part number of 705-00192, and a project number of 90003. It is currently at Revision A (REV A).

The relevant files are:

- `90003A.sch`: OrCAD schematic
- `Pumpkin.lib`: OrCAD library
- `Pumpkin2.lib`: OrCAD library
- `90003A.asc`: PADS-2000 ASCII job file
- `90003A.job`: PADS-2000 job file
- `90003A.net`: Futurenet netlist
- `90003A.zip`: all PCB CAD files, .zip'd
- `90003Apd.zip`: schematic and PCB verify plots, in Acrobat (.pdf) format, and Bill of Materials
- `90003Aps.zip`: PCB verify plots, PostScript format

## Circuit Description

Please refer to *Schematics* below or `90003A-1.pdf` and `90003A-2.pdf` in `90003Apd.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® PIC17C756A. Other compatible 16-bit processors (e.g. PIC17C752, PIC17C756, etc.) in a 68-pin PLCC package may also be used.

The *Oscillator* 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.

Most of the CPU's pins are available on the *Expansion Connector* H3. OSC[1..2] should not be used by any external devices. RA[2..5], RB[1..6] and RG[6..7] may or may not be available to external devices, depending on which components have been stuffed on the protoboard. A *Duplicate Connector* H6 is provided for easy jumpering to the signals present on H3. The *EDAC*[3] *Connector* H7 has all of U1's remaining pins,[4] and can be used to implement external memory on an expansion card instead of on the protoboard.

U8-U15 form the 16-bit-wide *External Memory System*. U8 and U9 demultiplex the multiplexed address and data lines of U1's AD[0..15]. ALE, -OE and –WE are the control signals. The 1 Megabit (128K x 8bit) *Flash* memories U10 and U11 are connected in a conventional manner to span U1's 16-bit address space. The 4 Megabit (512K x 8bit) static RAMs U14 and U15[5] are configured such that their upper 6 address lines specify an 8K page. These *Paged Address* bits PA[13..18] are driven by the serial shift register U13, which is part of the protoboard's SPI chain. U13 requires a rising edge on its strobe (STB) pin in order to transfer shifted data to its outputs QA-QH. RAM vs. Flash chip-enable *decoding* is performed by U12. This decoding can be *overridden* by setting the EN_RAM signal low (0V), in which case RAM is inaccessible and the entire 16-bit external address space (i.e. all 64K words) maps to Flash. The protoboard's Flash and RAM decoding can be altered by appropriate choice of U12:

| U12 | EN_RAM | Flash Space | RAM Space | RAM Banks[6] |
|---|---|---|---|---|
| 74HC00 | false (GND) | 0000-FFFFh (64K words) | -- | -- |
| 74HC00 | true (+5V) | 0000-7FFFh (32K words) | 8000-FFFFh (32K words) | up to 64 |
| 74HC20 | false (GND) | 0000-FFFFh (64K words) | -- | -- |
| 74HC20 | true (+5V) | 0000-DFFFh (56K words) | E000-FFFFh (8K words) | up to 64 |

**Table 1: Effect of U12 on Flash and RAM Map**

**Note** EN_RAM is reset[7] at power-on by the Reset Circuitry, i.e. EN_RAM must explicitly be set (+5V) before external RAM can be accessed. Also, unlike the Serial (SPI) EEPROMs, U13's contents cannot be read via SPI.

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.[8] 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[9] #1, making this a *Hardware RS-232 Port*. This port is commonly used for *debugging*.

Connector H5 is connected through U2 to U1's built-in high-speed USART #2, making this also a *Hardware 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 want to use USART #2 with a device other than U2, disconnect jumpers J1 and J2 and connect RX2 and TX2 to your device (e.g. a modem). If you still want to use RS-232 over H5, connect the two pads Z33 and Z34 to two general-purpose I/O pins on U1 and write a *software UART* to run in U1. The speeds of a software implementation are usually much lower than those of a hardware implementation. Therefore in this configuration H5 is part of a *low-speed Software RS-232 Port*.

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.

LED D4 is intended for use as a system status indicator. If U13 is not stuffed then D4 must be connected directly to one of U1's output pins.

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* TP1-13 are provided for DC voltages throughout the protoboard.

*Decoupling Caps* are provided for all of the digital circuitry. Special decoupling is provided for the analog supply voltages that serve U1's A/D converter.

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.

Six *Mounting Holes* are provided on 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 8-, 25- and 50-mil[10] traces on a 5 or 25mil grid with 8mil clearance rules. Only through-hole components are used. Signals are routed with 8mil traces and 45-degree bends, whereas power and ground[11] are routed with 50mil traces and orthogonal bends. Additionally, no trace may enter a pad with greater than 25mil width.[12]

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 *Schematics* below or `cover.pdf` contained in `90003Apd.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 or PICMASTER in-circuit emulator. Note that this is not a ZIF[13] socket, and therefore a PLCC extraction tool must be used to remove U1 from its socket. Therefore take extra care when removing or inserting U1.

U8, U9, and U12-U15 all are socketed using low-profile sockets. By socketing these chips you can remove the External Memory System entirely in case you wish to run in U1's microcontroller mode.

The prototype can accommodate two types of RAM chips – 62256 (32Kx8bit) and 628512 (512Kx8bit). U14 and U15 must be of the same type and speed.

62256-type RAMs can come in two different 28-pin PDIP packages: 0.600" width and 0.300" width. In both cases, the orientation is "bottom justified" with respect to the 32-pin 628512-type RAM outlines used for U14 and U15. See Table 2, Figure 1 and Figure 2 for more information.

| RAM type | Width | Position | J7 | J8 |
|---|---|---|---|---|
| 62256 | 0.600" | Pins 1-14 & 15-28 overlap pins 3-16 & 17-30, respectively, of 32-pin package outline. | on | off |
| 62256 | 0.300" | Pins 1-14 are in center of 32-pin package outline. Pins 15-28 overlap pins 17-30 of 32-pin outline. | on | off |
| 628512 | 0.600" | Occupies all 32 pins. | off | on |

**Table 2 : Jumper Settings for different RAM types**

**Note** Settings for jumpers J7 and J8 other than those shown in Table 2 are invalid.

Figure 1 shows the positions for pin 1 on the center row of 28 pins when using 62256 type RAMs with 0.600" lead widths for U14 and U15.

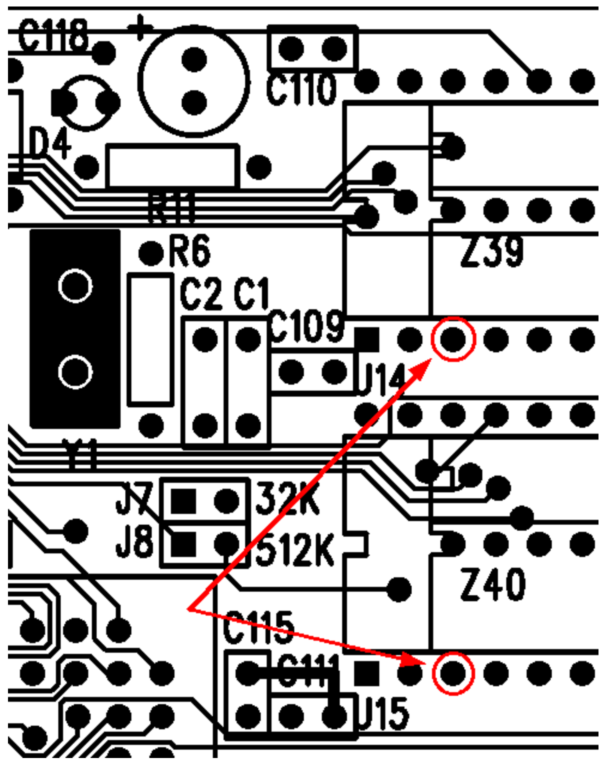


**Figure 1: Pin 1 Positions for 28-pin 0.600" RAMs**

Figure 2 shows the positions for pin 1 on the center row of 28 pins when using 62256 type RAMs with 0.300" lead widths for U14 and U15.

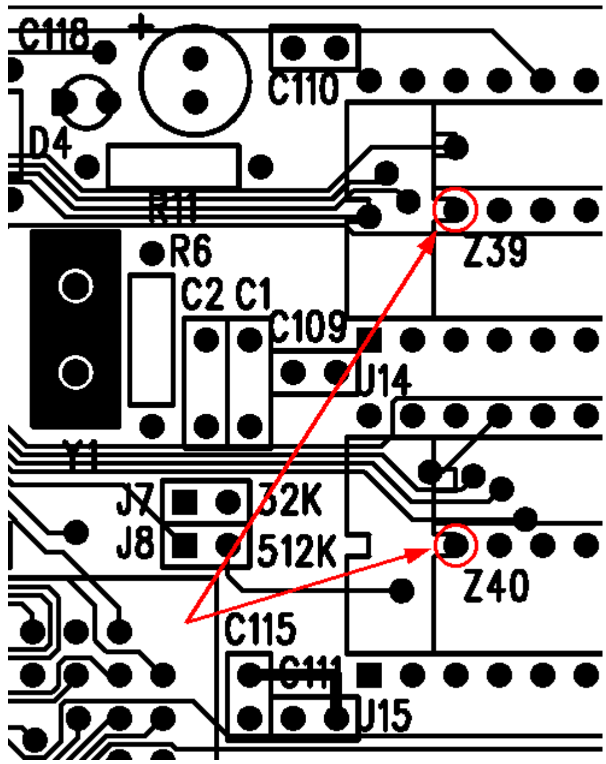


**Figure 2: Pin 1 Positions for 28-pin 0.300" RAMs**

U10 and U11 are socketed using ZIF sockets. The 28- and 29-series Flash memories do not support in-circuit serial programming – they must be programmed externally. By using high-quality ZIF sockets we're able to ensure a trouble-free, long life of removing the Flash memories for programming and re-installing them. The ZIF sockets accommodate DIP packages with 0.300" and 0.600" lead spacings.

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 C117-119 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 and make sure the upper washer is as far down the threaded shaft as possible[14] 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 3 below:[15]

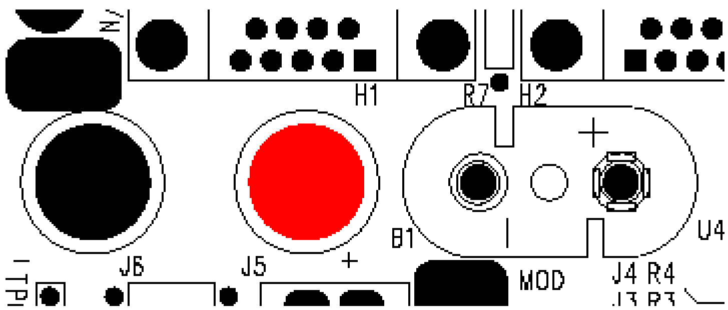


**Figure 3: Orientation of Banana Jacks and Battery Clip**

When you have finished soldering all of the components to the PCB, mount the 4-40 hardware (screw, lockwasher[16] 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.

*Be especially careful* when soldering the memory area of the board. There are many instances of two 8mil traces passing between pins on a 0.100" grid. Because there is no soldermask, it is easy to unintentionally connect a pin to a nearby trace with solder. Use a fine soldering tip and just enough solder to get the job done.

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.

The ZIF sockets for U10 and U11 *must be installed with their release levers at the edge of the PCB*. Otherwise they will not be able to open and close correctly.

The PLCC socket for U1 has a *required orientation*. The arrow in the socket points to pin 1. Insert the socket so that the arrow points towards connectors H3 and H6.

## 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,[17] 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). 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, U10, U11, U14 and U15) will draw approximately 30-60mA – 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 above are probably working. Any further testing must be done with a CPU or in-circuit emulator.

---

**Note** If your application fails to work when executing from Flash memory, you may have an unwanted connection in the External Memory System due to solder bridging between pins and traces (see above). "Stuck" address and data lines can easily been seen on an oscilloscope.

---

## Using the SSDL/SCU PIC17 Protoboard

A typical usage scenario is one where the protoboard is powered from a bench supply, with H2 connected to the system $I^2C$ 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 plan to use an external module designed for the (original) SSDL/SCU PICmicro protoboard you'll need to ensure that the module's pinout and its software driver are compatible with H3 and U1, respectively. External modules get all of their power from the protoboard – it's not necessary to hook power up to them separately.

The *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 H5 as a Hardware RS-232 Port, be sure to connect jumpers J1 and J2.

The PIC17C756A has up to 12 channels of 10-bit A/D. On the protoboard it is configured with $AV_{dd}$ = +5V and $AV_{ss}$ = GND, with extra 0.1µF and 10µF capacitors for noise rejection and the A/D converter's supply pins.  If this supply proves to be too noisy you can isolate the pins by cutting two traces located on the top side of the board near U1 and U9 as shown in Figure 4 below:

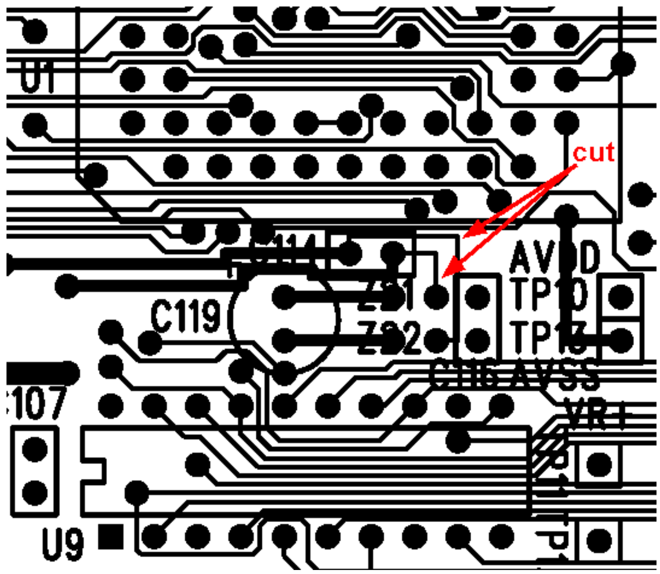


**Figure 4: Cut These Traces To Isolate A/D Converter's Power Supply Pins**

You can then feed the A/D converter with a dedicated clean supply by running wires to the two pads Z21 and Z22.

The A/D converter also has selectable positive (high) and negative (low) reference voltages. If you want a range other than 0-5V, configure U1's Special Function Registers accordingly and feed RG2 and RG3 with the appropriate voltages. These two pins ($V_{REF}$- and $V_{REF}$+) are available on H3 and H6.

The IRQ ("I"), Dallas bus ("D") and +5V Dallas ("5") signals from system connector H2 are available on individual breakout pads as shown in Figure 5 below:

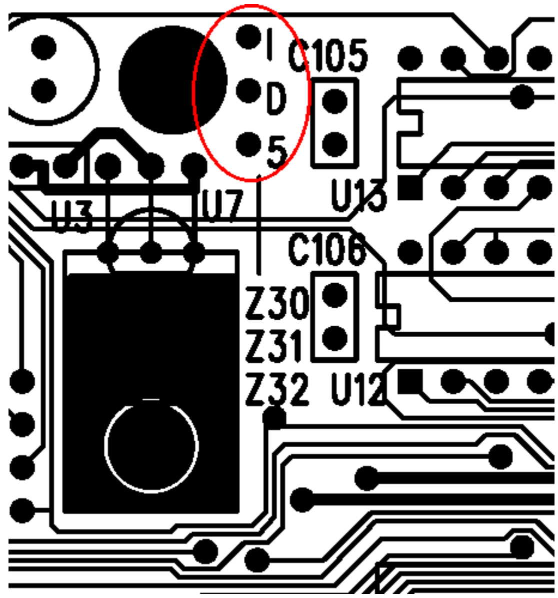


**Figure 5: Additional System Connector Breakouts**

## Integrating the Protoboard with MPLAB

If you do not have a PICMASTER or MPLAB-ICE2000 In-Circuit Emulator (ICE), you must develop PIC17 Protoboard applications by externally programming either the processor U1 itself or the Flash memory chips U10 and U11 and then inserting them into their respective sockets.

**Note** Repeatedly removing U1 for programming and reinserting it into its non-ZIF PLCC socket will quickly damage U1 and/or its socket. Therefore it is not recommended. Instead, use the Flash memories for non-ICE-based development.

In the non-ICE, Flash-memory development environment you must have a PIC17C756A with the following configuration bits:

- `PM2:PM1:PM0`: `111` (Microprocessor mode)
- `FOSC1:FOSC0`: `10` (XT oscillator)

This is not the PIC17C756A's default configuration. Therefore you must program U1's *configuration words* appropriately before it will execute the program stored in U10 and U11. You do not need to put an actual program (i.e. code) into U1. Setting the configuration bits only has to be done once. A PICSTART-Plus with a 68-pin PLCC adapter can be used to set the configuration bits.

Once U1 is properly configured, you need to program the Flash memories. U1's instruction word length is 16 bits. U10 and U11 are 8 bits wide. Therefore each time you change your application program you must reprogram both U10 *and* U11. A device programmer with the ability to "split" a source file (e.g. Intel hex) is required. Usually the first (low) device programmed will be U10, and the second (high) will be U11.

If you have an ICE for development, then you can disregard the Flash memories while testing and debugging your application. Just use the ICE with U1 in either microcontroller or microprocessor[18] mode. This bypasses the on-board Flash memory entirely.

---

**Note** If you use microcontroller mode with an ICE, remember that your final, flash-based application cannot use PORTC, PORTD or PORTE:0-2.

---

When you are finished, program U10 and U11 and insert them and a pre-configured PIC17C756A onto the protoboard. and your program should run.

## Schematics

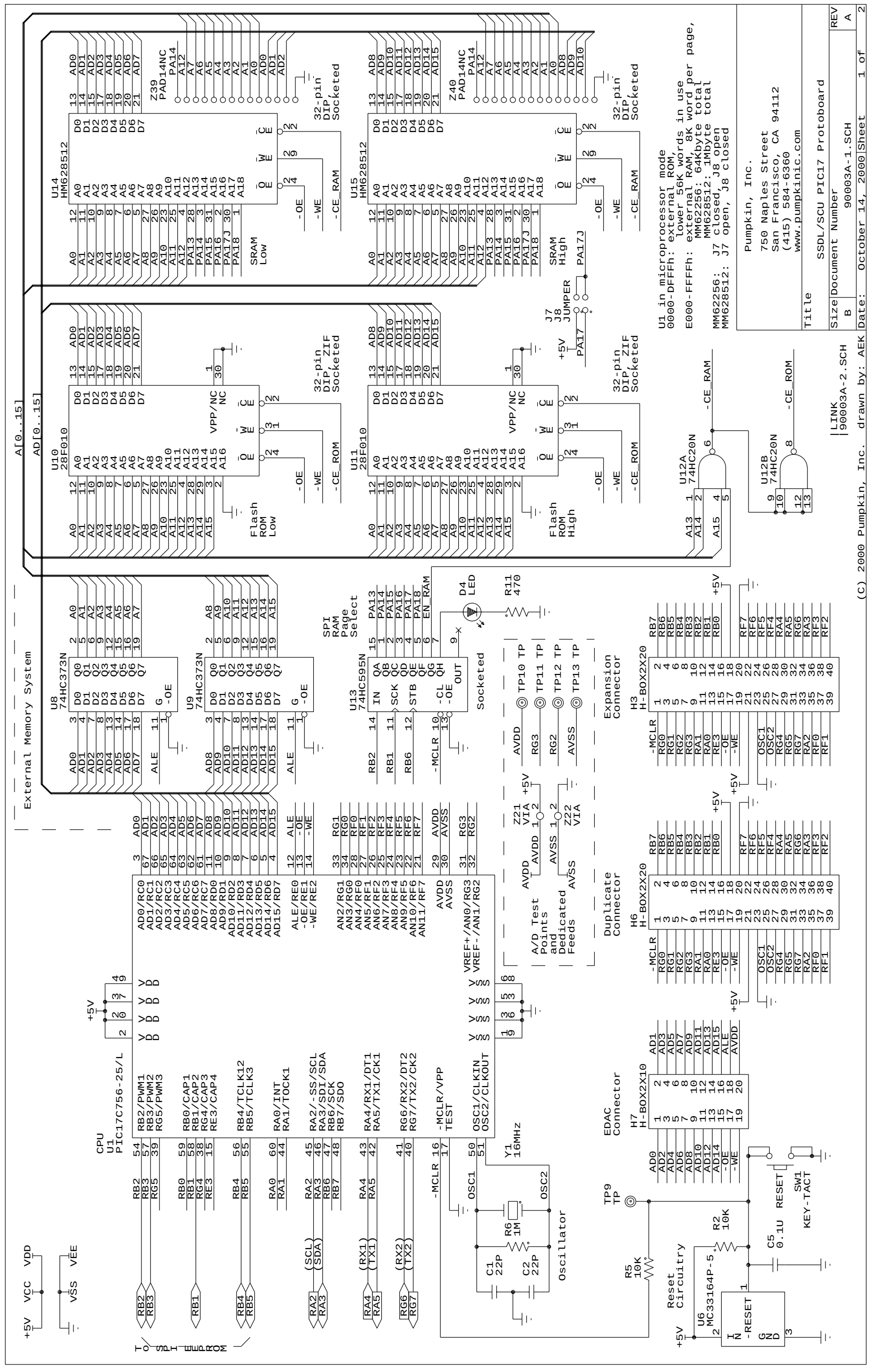

**Figure 6: First Page of Schematics**

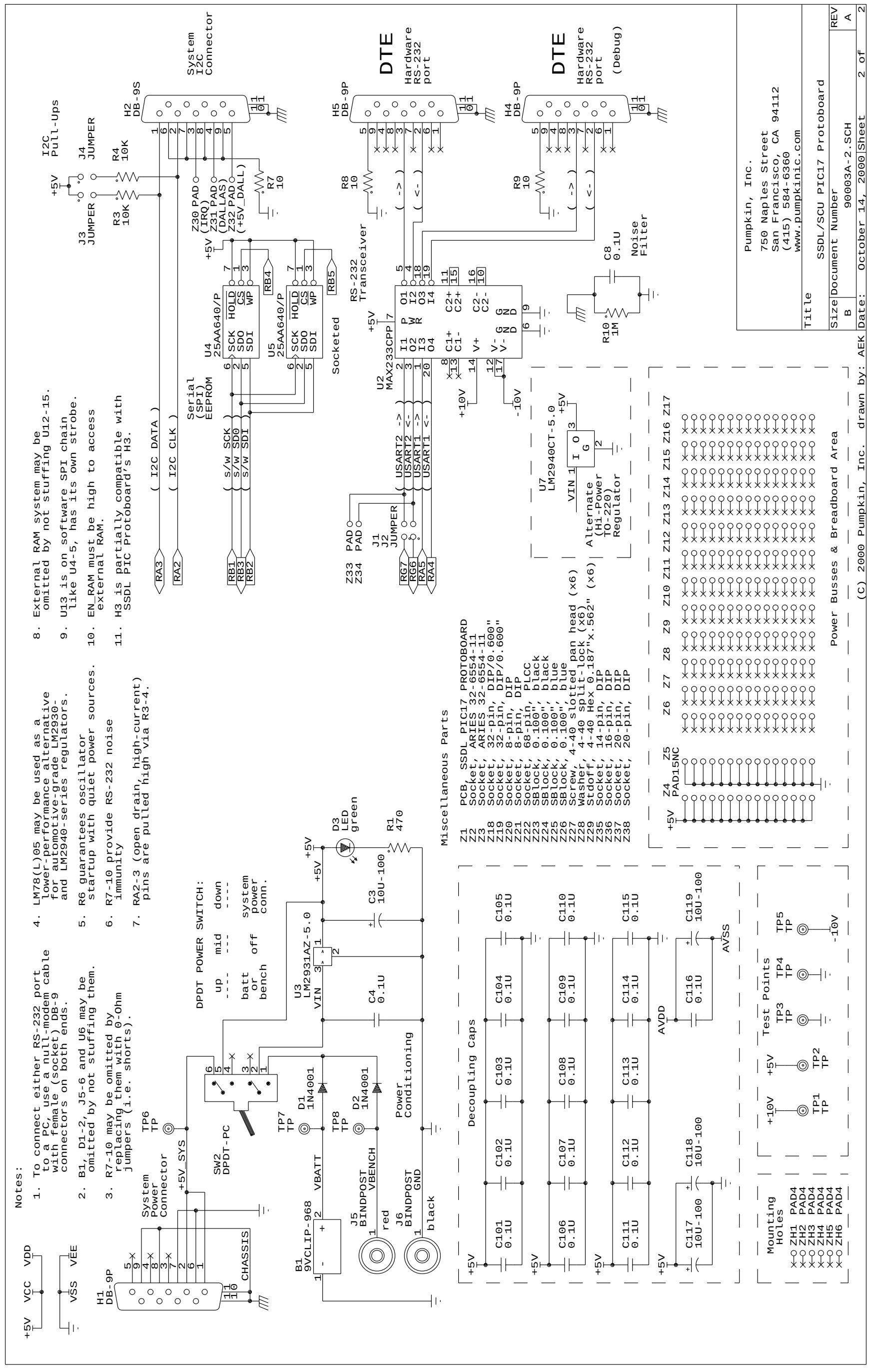

**Figure 7: Second Page of Schematics**

# PCB Layers

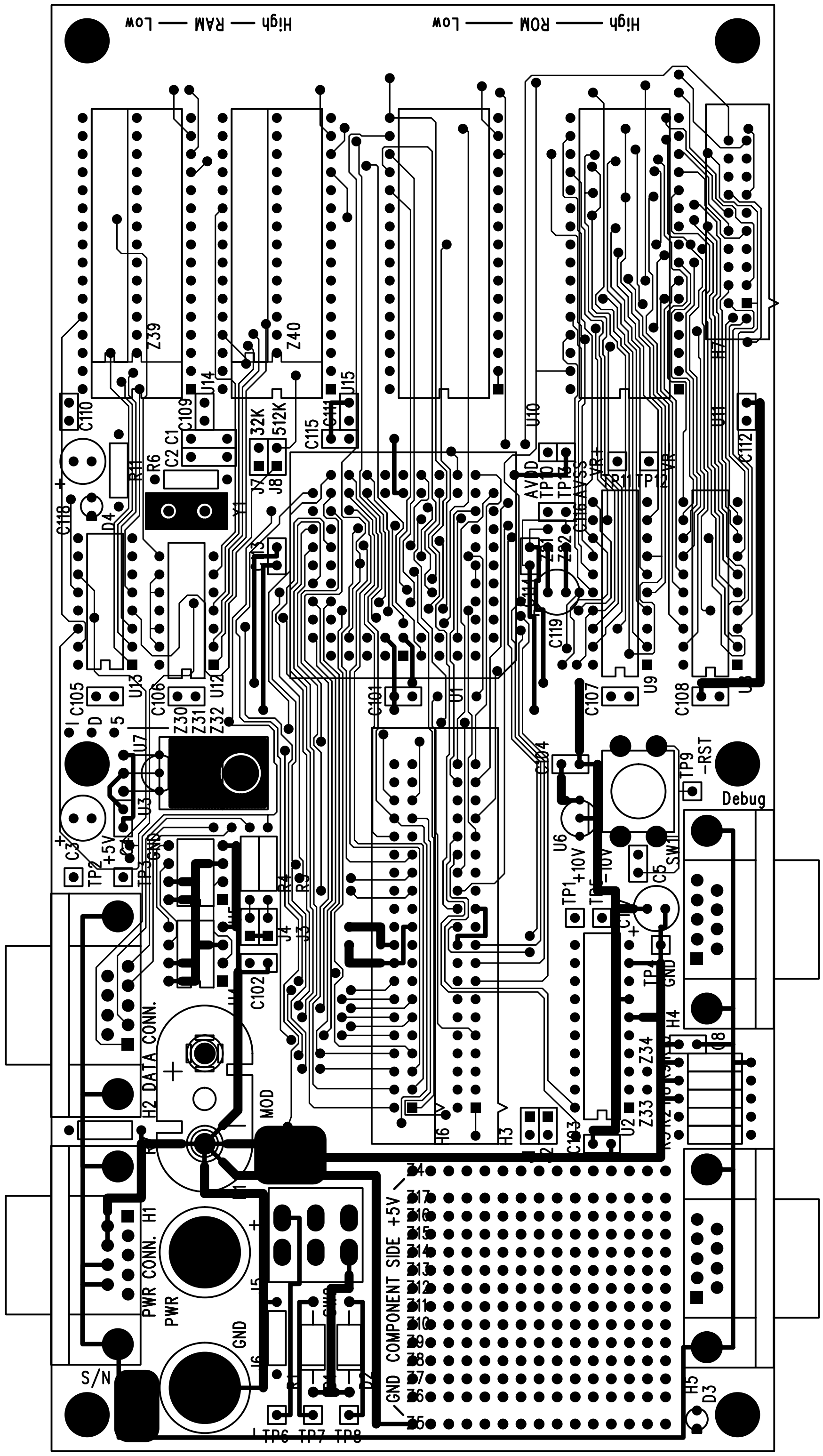

**Figure 8: PCB Topside Plot**

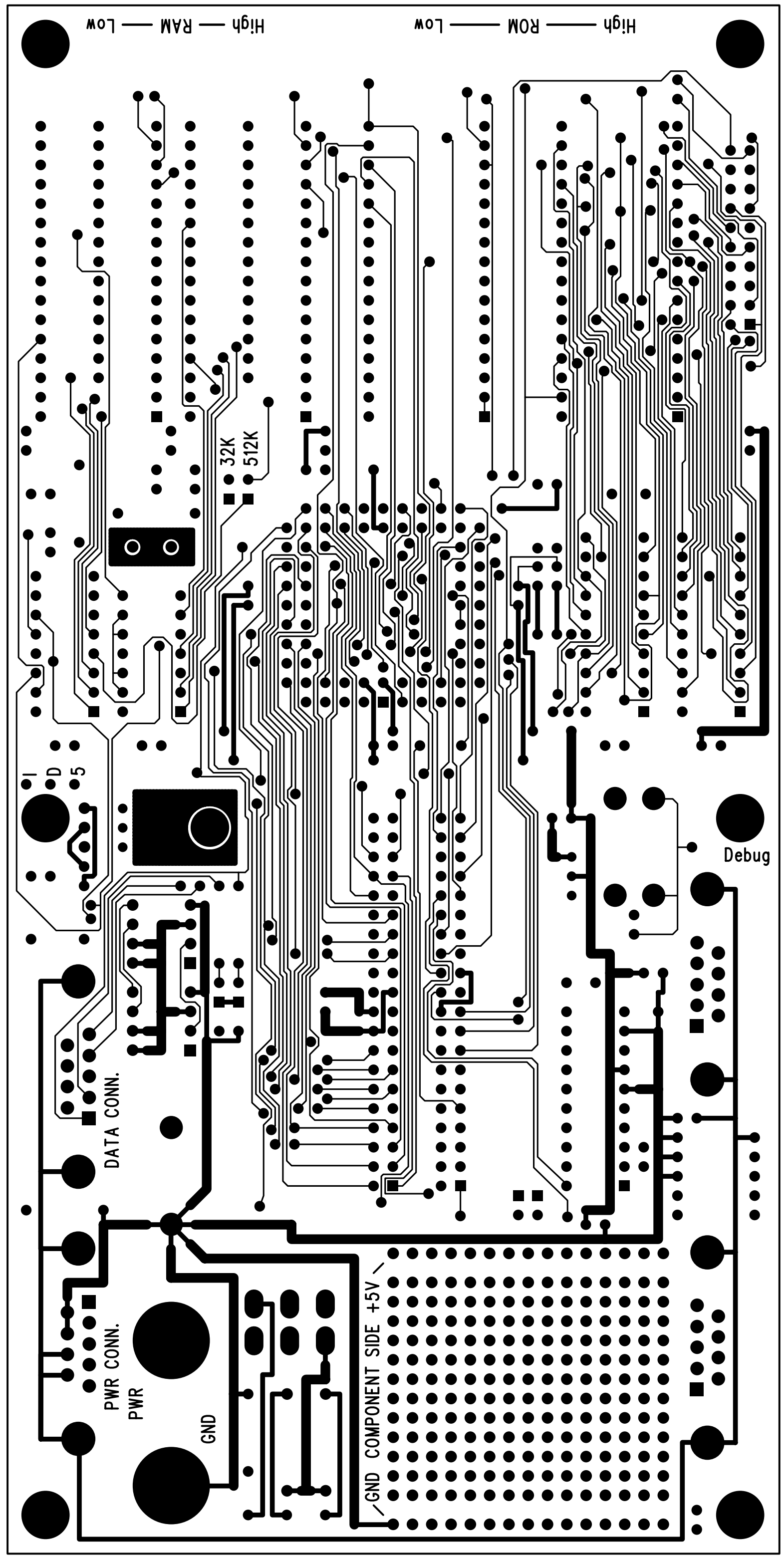


**Figure 9: PCB Top Routing Layer**

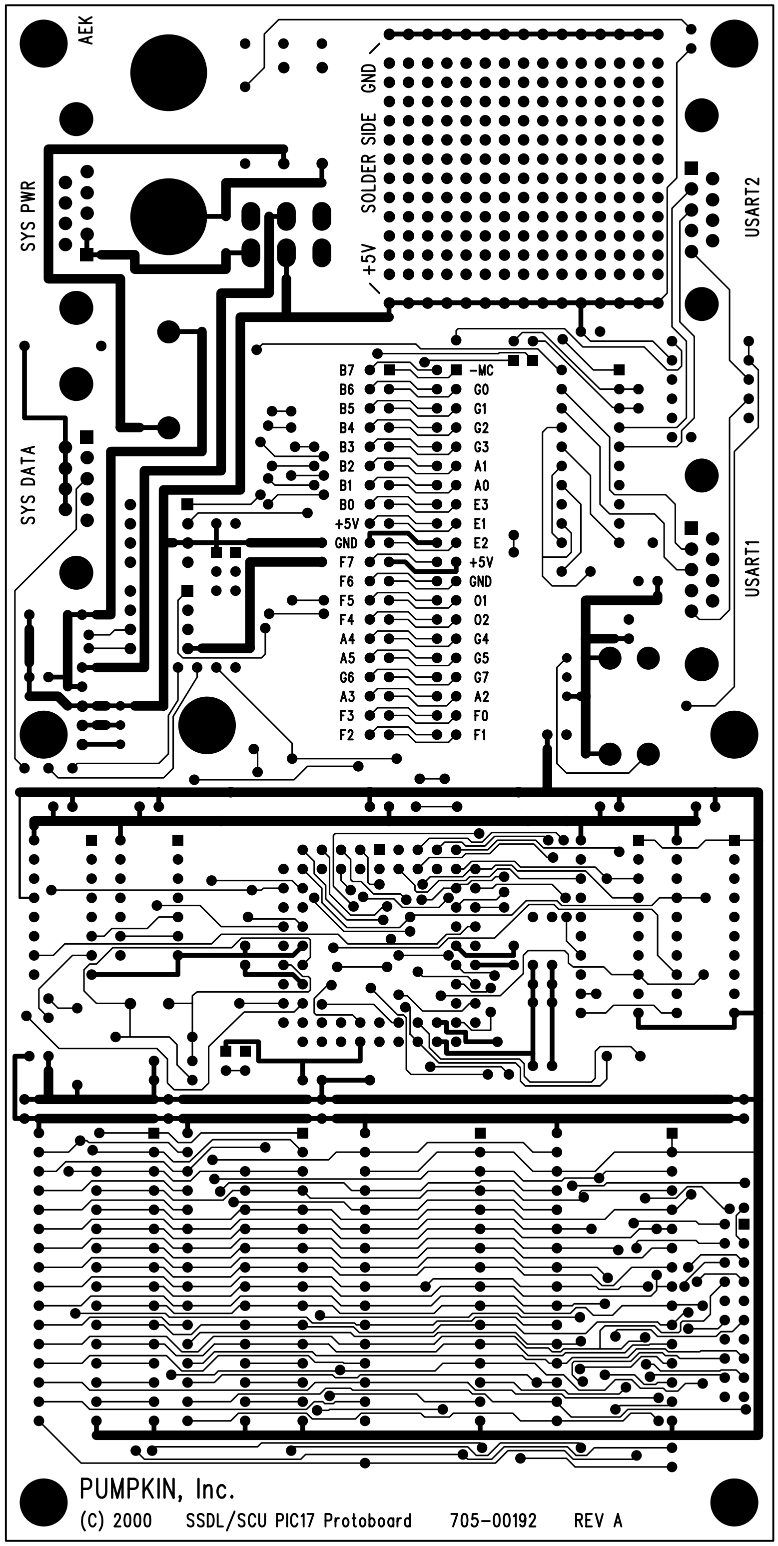


**Figure 10: PCB Bottom Routing Layer**

1 The PIC17C756 (an earlier version) may also be used. The PIC17C756A supports in-circuit serial programming (ICSP).
2 It may be ancient, but it's also reliable.
3 Error Detection And Correction. May refer to memory (and other) systems which employ a system for determining which copies of multiple instances of important information are still valid in the event of partial corruption.
4 Except $AV_{ss}$, which is not available on any connector. $AV_{ss}$ is the A/D converter's lesser supply voltage. $AV_{ss}$ must equal $V_{ss}$ on U1.
5 256 Kilobit RAMs (generic type 62256) can also be used.
6 The number of RAM banks is dependent on the size of the RAMs.
7 The resetting of EN_RAM on REV A protoboards is unreliable because of potential noise on PORTB:6 during reset. Your application should explicitly configure EN_RAM at system startup.
8 That's what diodes D1 and D2 are for.
9 Universal Synchronous/Asynchronous Receiver/Transmitter.
10 One thousandth of one inch, i.e. 0.001".
11 Chassis shields are routed at 25mils on this PCB.
12 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).
13 Zero Insertion Force.
14 This guarantees that the fastening screw that goes through the PCB is extended to its full length.
15 J5 is shown in red.
16 On top of PCB.
17 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 > $7V_{dc}$.
18 In MPLAB, select Memory: Supplied by Emulator under Options > Development Mode… > Off-Chip Memory