

---

# GETTING STARTED

---

The SPCL-0001,2,3,4, and 7 are F68HC11 based single board computers with built-in LCD display interface, a 4x5 matrix keypad interface, and a data acquisition section with up to 8-channels of 12-bit A/D and 4-channels of 12-bit D/A. Several versions of this board are available. Each version is based on the same printed circuit board and only varies by the number of A/Ds and D/As available. The SPCL-0102 is a special target version of the SPCL-0004-X20 but has fewer parts installed. Normally, a developer will use the SPCL-0004-X20 for development, and high end projects, then switch to the lower cost SPCL-0102, or one of its other NMI derivatives when volume buying begins. The SPCL-0102 circuit board supports all the features needed for program development, but is not fully populated to provide all options. The convenience of on-board power supply circuitry and easily movable jumpers are forfeited. However, the inconvenience is minimal. Rigorous development can still be accomplished on the SPCL-0102 at much lower cost.

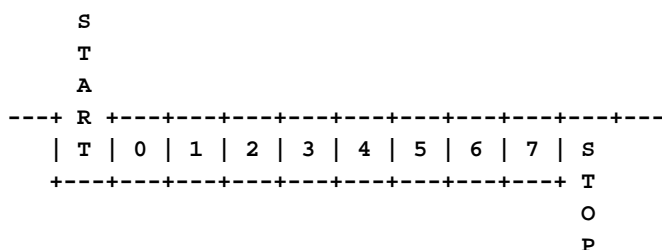
To operate the SPCL system, power and communications connections must first be made. Three power supply voltages are required. The +5V powers all the digital circuitry. The computer section will operate with only the +5V and GND connections. This allows language development that does not require use of the data acquisition section. Analog supplies are necessary to make the A/D and D/A sections fully functional. The +12V and -12V analog supplies are required by the multiplexers, the instrument amplifier, and the A/D and D/A chips themselves. The +5V supply connects to the pins labeled +5 and GND on TB2. The analog +12V and -12V supplies connect to the pins labeled +V, GND, and -V on TB3. TB1 is provided for a 7-18VAC power input from which +5VDC can be derived. However, this power supply option is not provided on the SPCL-0102.

For development, a terminal (or PC with terminal emulation software) must be connected. A cable with an RS-232 connector (DB25, or DB9) on one end and a 4-pin connector on the other end is needed for this connection. This cable is not included with the SPCL, and must be provided by the user, or purchased separately from the supplier. The serial cable connections are made to the board at J6. J6 is located just to the right of U6, the RS232 driver. Only pins 5,6,7, and 8 of J6 are used for RS232. The interconnections are detailed below for both the DB25F and DB9F connectors.

J6	DB9F	DB25F	Signal Name
5	N.C.	1	Case ground
6	3	2	SI - Serial In (to SPCL)
7	2	3	SO - Serial Out (from SPCL)
8	5	7	Electrical ground
N.C.	7 to 8	4 to 5	CTS to RTS
N.C.	4 to 6	6 to 20	DSR to DTR

'Case ground' and 'Electrical ground' both terminate to a common ground on the printed circuit board. This permits cross connecting the SI and SO lines by simply rotating the single in-line, 4-pin connector 180 degrees. The configuration shown in the table above is for Data Set Configuration. This will be the most likely configuration required.

In order to establish communications with the SPCL-0102, the terminal must also have the correct bit settings and speed. The baud rate should be set at 9600 baud for the standard 2 Mhz system (8 Mhz crystal). The SPCL-0102 sends and receives a bit protocol of one start bit, eight data bits and one stop bit.



Pin 9 of J6 is the reset line for the microprocessor. It is normally held high. A reset is accomplished by first pulling this line low, then releasing it to its original high state. A reset will occur on the rising edge of the signal applied to Pin 9. Pin 10 of J6 is ground. In addition to the RS232 connection described above, a normally open, push button should be connected between pins 9 and 10 of J6. This is the reset button.

When communications with the terminal are established, the SPCL-0102 will respond by displaying **Max-FORTH Vx.x** on the screen each time the reset button is pushed and released, if you are using the Forth language resident in the F68HC11. The signon may be different if you have disabled the Forth in favor of another language. Max-FORTH is resident in ROM in the F68HC11 normally supplied with the board. This is an excellent language to use as a monitor to check the operation of the board, even if the final application is done in another language. Basic and C are also available for use with the SPCL-0102.

The signon message means the terminal is communicating with the SPCL-0102. Press "return" on your terminal several times. If the SPCL-0102 responds with "**OK**" each time, communications are established. To further exercise the interface type "**WORDS**" in all caps and press "return". Max-FORTH should respond by showing a list of all the words in its vocabulary (several hundred). If this is successful, the digital portion of your SPCL-0102 is now running and communicating as it should with the terminal..

---

# NMIX AND NMIT DIFFERENCES

---

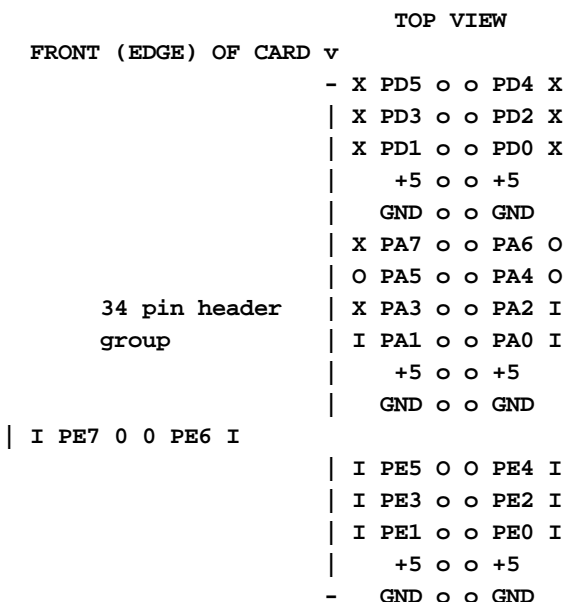
The SPCL-0102 is a special order number for the SPCL-0004-X20, which is an "X" version in the NMI board series. The SPCL-0102 is a "T" version of the board. The SPCL-0102, in the generic target configuration, is a minimum, 5 Volt only, configuration. The F68HC11, Xtal, reset circuit, various HC "glue" components and three 28 pin JEDEC sockets. Typically, a program developed in the "development configured" board will be installed in the "generic target configured" board for production of a dedicated application. The user must install the appropriate jumpers, which are not provided in the target configuration. The SPCL-0102 has one addition other "T" boards may not have. The RS-232 converter is included with the SPCL-0102. This allows the SPCL-0102 to be used for development as well as in final target applications. The negative voltage generated by the RS-232 circuit is also used to bias the LCD display, and is therefore important to its operation.

All configurations of the F68HC11 based SPCL-0102 boards use the same base PC board as the SPCL-0004-X20. Configurations differ only to the extent to which the board is filled with "nonessential" components.

# PARALLEL PORTS

The F68HC11 has five parallel ports, Port A, B, C, D and E. Ports B and C have been sacrificed to create a 64K address and data bus. Although some of the remaining port lines have special multiplexed functions, they can all be used as inputs or as outputs according to their individual designs. Some of the port lines have direction registers allowing them to be used as either inputs or outputs. The three remaining ports are brought out to connector J2 along with power and ground.

## INPUT/OUTPUT JACK J2



I=INPUT O=OUTPUT X=EITHER

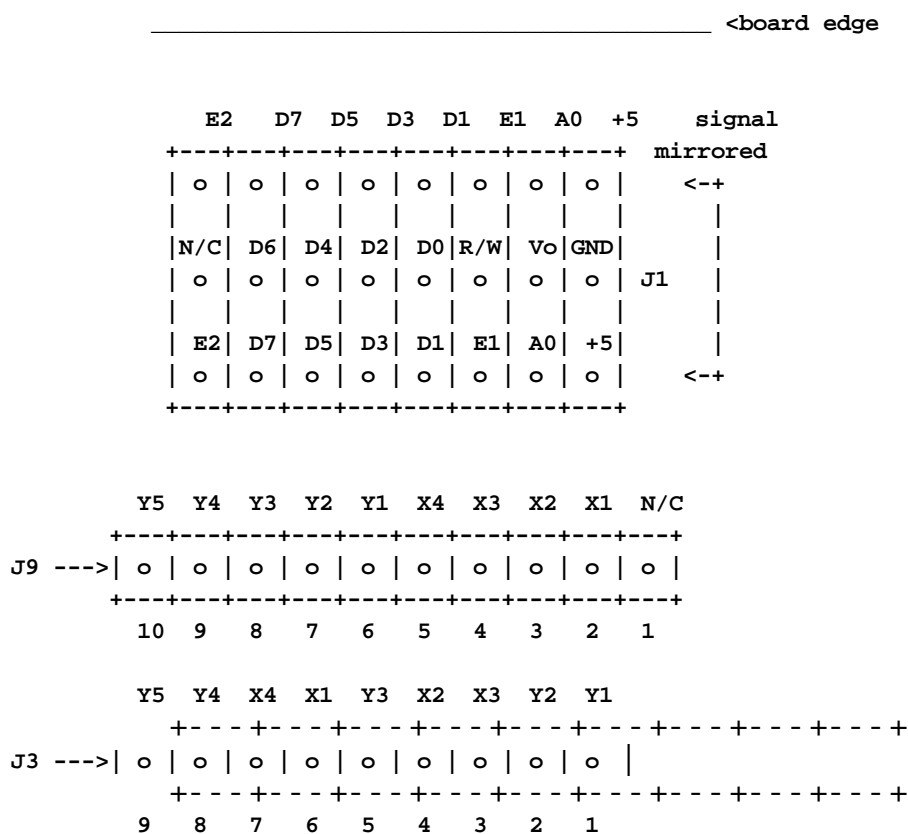
These lines can be used as individual inputs, outputs or in combination. There are very few applications, however, where pins are switched dynamically, sometimes to be used as inputs, other times as outputs. A voltage of 0.7Vcc or greater will always be recognized as a logical one. Voltages 0.2Vcc or lower will always be recognized as logic 0. **(Voltages applied above Vcc or below 0 Volts can damage the computer.)**

The outputs of the F68HC11 can sink 1.6 mA to ground while letting the pin go no higher than 0.4 Volts for a "zero" and source about .8 mA at 4.5 Volts for a "one". In terms of control, this is a very small signal. Most relays require over 50 times more current to operate. LED's typically take 5 mA to be visible. HC levels are such that the output is sufficient to drive the input on one pin of one TTL device or about a dozen of the lower power LSTTL inputs. The output is sufficient to drive VMOS FET's and Darlington's with an external pull up which can in turn control several amps of current. Usually, however, a buffer will be needed to do serious non-HC interfacing.

# KEYPAD AND LCD INTERFACE

The SPCL-0102 has a built-in Keypad Controller, the 74C923. This device scans matrixes of keys up to 4x5 without processor intervention. Connection of the 74C923 Keypad Controller to the cpu is via 68HC11 Port E pins PE3 thru PE7 for key data and Port A pin PA0 for the key valid strobe. The operation of the 74C923 Keypad Controller provides a high level Data Available Strobe to Port PA0 when a valid key is detected in the keyboard matrix. This can be detected by the 68HC11 under software control and the key data can then be read from Port E as a binary number that represents the valid key on the keyboard. Connector J9 provides one keyboard connection common to inexpensive membrane keyboards. Connector J3 is for use with keyboards similar to Grayhill Series 86 or 88 keyboards. The following is the pinout of J9 and J3. Both of these connectors are located just below the triple row LCD connector, J1.

KEYBOARD AND LCD INPUT/OUTPUT  
TOP VIEW



The SPCL-0102 has a built-in connector (J1) and decode circuitry to allow direct interfacing to many of the popularly available, intelligent LCD displays. A wide number of LCD modules can be accommodated, since many manufacturers make the modules with the same controller chips or control operation. Some of these manufacturers are AND, Densitron, Epson, Optrex, Sharp, and Sieko. They come in configurations such as 1x8, 1x16, 2x16, 1x20, 2x20, etc., up to 4x40 or 2x80.

Connector J1 has three rows of eight pins each. The pins in the top row are connected to their counterparts on the bottom row. This provides a mirroring of signals permitting a cable connection to be made to either the front or backside of the LCD display. Only two of the three rows should be used. Either the top and middle rows, or the middle and bottom rows. The two rows contain 16 pins but will accept either 14 pin or 16 pin ribbon connectors from the standard LCD modules. Pin out is identical with the exception of an additional enable signal (E2) used for the larger displays. The top two rows of J1 are configured to accept ribbon connectors that are taken off the back side of the LCD. The bottom two rows are configured to accept ribbon connectors that are taken off the front side of the LCD.

The LCD interface is hard addressed at four consecutive locations, \$B5FC hex thru \$B5FF hex. On board logic provides the necessary chip select and timing information to operate the displays. Address line A0 goes directly to the displays, so each chip select represents two memory locations. The smaller displays, with up to 80 characters, use only one display controller chip. Those with a larger number of characters use additional display controller chips. Those with 16 pin connectors have up to two controllers built-in.

The type display attached will determine its own access speed. Generally they are listed at 450ns. This is fast enough for 1 Mhz bus timing (6800 and 6500 type processors), but not fast enough for 2 Mhz. Almost all of the displays will work, however, at this higher speed, although using them this way means they are outside the manufacturer's listed specification.

The board provides little support to the display processor, other than providing the necessary signals, voltages, and gated chip selects. The handling of the displays follows the manufacturer's specifications for the particular display. Extensive example program segments are shown in Appendix B for single controller, 2 line displays. For other configurations and types refer to the manufacturer's literature.

---

# SERIAL I/O

---

The F68HC11 has a full duplex hardware serial channel that operates at CMOS levels. To use this serial channel with most standard communications interfaces, level converters are needed. Drivers for RS-232C and RS-422/485 drivers are on the boards. (It should be noted that only one combination of RS-232 driver, RS-422 drivers or RS-485 driver should be used at one time to avoid contention of their receiver outputs.)

A zero by RS-232C specification is any voltage from +3 to +15 Volts, a one is between -3 and -15 Volts. To convert the HC signals to the voltage ranges of that interface standard, the SPCL-0102 Rev. 1.0 uses a single 16 pin device, the ICL232.

The ICL232 is ideally suited for this use. It not only provides an RS-232 receiver and transmitter pair for the F68HC11 processor, but also a spare RS-232 receiver and transmitter pair which can be used with port lines for handshaking or software driven UARTS, etc.. It also generates the higher voltages needed for full RS-232 communications standards by way of an internal charge pump. This allows output swings of a nominal + and - 9V, even though the chip is only supplied +5V. (The negative output is also used to get the negative voltage bias for the display to increase contrast.)

The RS-422 standard represents a newer interface now coming into popularity, and with good reason. Unlike the RS-232 requirements which specify a single wire voltage transmission referenced to ground, the RS-422 standard uses a voltage differential on a pair of conductors. While the RS-232 at full voltage drive levels in electrically noisy environments is barely reliable at distances to 1000 feet, RS-422 signals are considered reliable at distances up to 4000 feet. The RS-422 drivers operate, requiring only a single sided 5 Volt supply, over twisted pairs of wires. A full duplex connection for RS-422 requires two twisted pairs, one for transmit, one for receive. The shield of the twisted pair should act as the common return path for the signals.

The RS-485 interface uses the same specifications for its transmitters and receivers. It, however, allows a single twisted pair to be used for incoming and outgoing messages. This is accomplished by having both a transmitter (with 3 state ability) and a receiver tied in parallel to the same twisted pair. Multiple drop point communications are possible under this scheme (up to 64 pairs by specification). Of course, in application the transmitter turns on and takes control of the lines only under software control. The actual implementation of this control will be determined by the particular protocol being used in the communication network. Usually one master sends an addresses message to one of multiple slaves and then turns off its master transmitter. The addressed slave, recognizing its address will turn on its transmitter and respond with the requested data.

These two interfaces are accommodated on the SPCL-0102 by the addition of two 8 pin 75176's, which each contain a transmitter/receiver pair. Whether the transmitter of the pair is active, or not, is controlled by a signal on one of its pins.

One of the 75176's (U11) has its receiver always enabled. It is used exclusively as the RS-422 receiver. The other 75176 (U10) can be used as the RS-422 transmitter if jumper G on the SPCL-0102 is grounded (i.e.: in 422 position), or it can be used as the receiver and transmitter for the RS-485 interface as controlled by Port A pin PA3 (i.e.: in 485 position). In this case if PA3 is low, the 75176's transmitter is not active. If PA3 is high its transmitter is active.





---

## A/D SUBSECTION

---

The SPCL-0102, Analog-to-Digital-Converter, subsection provides four channels of Analog-to-Digital (A/D) input. The converter chip is an industry standard 574, which provides differential 12-bit readings. The 574 accepts analog input in the range of 0 to 10 volts with one binary increment representing 2.44 millivolts. A HI-549 multiplexer provides four differential input channels. (The SPCL-0102 can also be configured to access four additional (eight total) input channels by adding another HI-549.)

Examination of the A/D subsection reveals the following features: At the top of the board on the right hand side, a 34-pin connector is labeled J8. The eight possible A/D differential inputs and their returns are on the right side of this connector. Below this connector, arranged vertically are places for two HI-549 analog multiplexers, U24 and U25.. Only U25 is installed on the SCPL-0102 version of the SPCL-0004-X20. The HI-549 is a dual 1-to-4 multiplexer with input protection. Below the multiplexer is an AD625 programmable-gain instrument amplifier U23. The instrument amp has a zero trim control, RV3, a 10k pot located immediately below the amp. The HADC574, U22, is a 28 pin dip located near the middle right hand side of the card. To the left of U23 you will find U21 and U20. These are standard 74HC devices providing chip select and control logic functions. U21 is a 74HC574. This is an octal D-latch used to latch the instruction byte sent to the A/D channel register (memory location B5F8 or B5F9). Only four lines (E0, E1, E2 and E3) are needed to select the eight multiplexed A/D channels. The least significant four bits of the instruction byte must be reserved for multiplexer selection. The most significant four bits can be used to control devices external to the SPCL-0102 board. All eight latched outputs from the 74HC574 are made available on J8. They are labeled E0 through E7.

The A/D subsection is memory mapped to B5F8 - B5FB hex. These addresses will be used in programming examples which follow. The following is a short test routine that will exercise one A/D channel. It can be entered by hand to verify that the A/D section is at least working. The leftmost column (column 0) displays the digital output from channel 0. The next column to the right (column 1) displays channel 1 and so on. These columns will scroll down the screen until a key is pressed. In cases where all eight possible channels are being used, take care to use only one channel at any given time. Avoid selecting any one of channels 0,1,2, or 3 (U25) while any one of channels 4, 5, 6, or 7 (U24) is active. Doing so will connect the outputs of the two multiplexers together with unpredictable results. To prevent this occurring, avoid writing values of C, D, E, and F to memory location B5F8.

The only valid channel selection data is as follows:

CHANNEL	DATA TO B5F8
0	X7
1	X6
2	X5
3	X4
4	XB
5	XA
6	X9
7	X8

NOTE: Only the four least significant bits control the multiplexer channels.

**CAUTION:** *The channel selection register is not gated with read/write. This means that if a read of the channel selection register is done the output of this register will be an unknown. DO NOT read locations B5F8 or B5F9.*

COLD

HEX

```
: DELAY 2000 0 DO LOOP ;

: ADTEST0 ( TESTS CHANNEL 0
CR
BEGIN
  7 B5F8 C! ( register offset for channel 0
  DELAY
  0 B5FA C! ( force conversion
  B5FA @ 2/ 2/ 2/ 2/ FFF AND U. ( calculate and display
  CR
  ?TERMINAL ( wait for input
  UNTIL ( Loop until key is depressed
  KEY DROP
;
```

NOTE: All of the above code must be entered using capital letters only. Comments begin with a "(" and need not be entered.

The Programming sections have further programming examples which can be used to set the A/D. Install the program or program segments to be used.

The first time the chip is powered, an unknown channel will be selected. Writing "0"s to the channel selection register is suggested. This will insure a valid conversion result. The A/D subsection of the SPCL-0102 is now running and communicating as it should.

## **PROGRAMMING**

Writing drivers for the SPCL-0102 A/D is not difficult. The HADC574 is memory mapped into a two-byte location. The processor can select inputs by doing a byte-write to the channel selection register.

The input channel is selected by writing the desired channel number to either the first or second byte. All programming examples assume that the machine is in HEX mode.

### **HEX**

```
: CH-0 7 B5F8 C! ;  
: CH-1 6 B5F8 C! ;  
: CH-2 5 B5F8 C! ;  
: CH-3 4 B5F8 C! ;  
: CH-4 B B5F8 C! ;  
: CH-5 A B5F8 C! ;  
: CH-6 9 B5F8 C! ;  
: CH-7 8 B5F8 C! ;
```

## **OR**

### **HEX**

```
: CH-0 7 B5F9 C! ;  
: CH-1 6 B5F9 C! ;  
: CH-2 5 B5F9 C! ;  
: CH-3 4 B5F9 C! ;  
: CH-4 B B5F9 C! ;  
: CH-5 A B5F9 C! ;  
: CH-6 9 B5F9 C! ;  
: CH-7 8 B5F9 C! ;
```

12-bit A/D conversion is started by writing ANY number to B5FA.

HEX

: START-12-BIT 0 B5FA ;

OR

HEX

: START-12-BIT FF B5FA ;

An 8-bit A/D conversion is started by writing any number to B5FB.

HEX

: START-8-BIT FF B5FB ;

OR

HEX

: START-8-BIT FF B5FB ;

Reading a 12-bit conversion is done by reading both the third and fourth bytes of the memory mapped address range B5F8 - B5FB. The value read will be left justified requiring scaling to obtain a true value.

HEX

: READ-12-BIT B5FA @ ;

: SCALE-12-BIT 10 / FFF AND U. ;

Reading an 8-bit conversion is done by reading the third byte (B5FA) only. Since the conversion is left justified, this is the only byte that needs to be read in an 8-bit conversion.

HEX

: READ-8-BIT B5FA C@ ;

## A/D CALIBRATION

The analog to digital conversion circuitry is calibrated at the factory before shipment. If any A/D components are replaced, or if it become necessary for any other reason, the circuitry may be re-calibrated in the field.

### Tools Required:

Digital Voltmeter

Adjustable voltage source ( 0.00 to 10.00 volts DC) (OR use the D/A converter on the SPCL-0102 board.

A small flat blade screwdriver

1) Hook up the NMI to a computer or terminal and establish communications. Connect the positive lead of the adjustable voltage source to Pin J8-33 (A/D channel #0 high input). Connect the ground return lead of the adjustable voltage source to Pin J8-34 (A/D channel #0 low input).

2.) With the power turned off, remove U23 (the AD625) from the circuit board. Install a shunt to short the two pins of J7 together. This grounds the input of the A/D converter, U22.

3.) Load and run the following routine to continually read the value from the A/D converter.

```
COLD
HEX
: CAL 7 B5F8 C!
    BEGIN
        0 B5FA C!
        100 0 DO LOOP
        B5FA @ 10 / FFF AND U.
    ?TERMINAL
    UNTIL
;
```

This routine will provide a screen display of the form:

xxx xxx xxx ...

**where XXX is the returned value of the A/D conversion.**

4.) Adjust RV2 for a screen display of "000". (Adjust up to 001, then back off to 000.)

5.) Shut off the power. Remove the jumper from J7. Install U23.

6.) Turn on the power. Re-load the program listed above, if necessary. Set the voltage source to 0.00 vdc and adjust RV3 for a screen display of "000".

7.) Set the voltage source to 10.00 vdc. Adjust RV1 for a screen display of "FFF". When "FFF" is displayed, adjust down to "FFE" , then back to "FFF". Calibration is now complete.

---

## D/A SUBSECTION

---

The SPCL-0102 provides one channel of 12-bit Digital-to-Analog output as a standard option. Analog Devices AD7248 "DAC Port", Digital-to-Analog Converter ICs are mapped into memory locations B5F0 through B5F7. They provide one 12-bit D/A voltage output each. Up to three additional (four total) "DAC Ports" can be added. The "DAC Ports" provide D/A output of 0 to +10 volts, with one binary increment representing 2.44 millivolts. The AD7248 is a completely self contained device. It does not require the external support devices normally needed by DACs. Each AD7248 contains its own internal precision buried zener reference diode and op-amp buffer.

The analog outputs of the "DAC ports" are available on J8. They are labeled 00, 01, 02, 03. Directly above each of these pins is an analog ground return pin. Analog channel 00 is the output of U18 and is the only "DAC port" delivered with the standard version of the SPCL-0102. The board provision for three additional "DAC ports" in locations U17, U16, and U15.

The following is a short test routine that will exercise all D/A outputs (up to 4). It will generate a rising staircase signal on all the D/A outputs repeatedly until a key is hit on the terminal. Use an oscilloscope or fast voltmeter to check for the signal. Only the first line of DUP B5F0 ! is required for the one supplied D/A, U18. The remainder are for U17, U16 and U15.

**COLD**

**HEX**

**: TEST**

**0**

**BEGIN**

**1+**

**DUP B5F0 !**

**DUP B5F2 !**

**DUP B5F4 !**

**DUP B5F6 !**

**?TERMINAL**

**UNTIL**

**DROP**

**;**

The Programming section has further program segments which can be used to set the D/A. Install the program or program segments to be used. The first time the chip is powered, registers need to be set up. Writing "0"s to D/A's is suggested.



The D/A subsection of the SPCL-0120 is now running and communicating as it should.

## PROGRAMMING

Writing drivers for the SPCL-0102 is not difficult. Each AD7248 is memory mapped into a two byte location. The processor can control the outputs by doing a high-byte low-byte write to the AD7248s. The latching AD7248s take any value written and convert it directly to a voltage output. The 0.00 to 10.00 volt range is available on the SPCL-0102 board. Each bit count difference represents 2.44mV. Full scale is  $4095 \times 2.44 \text{ mV}$  or 9.99 volts.

The base address of each DAC starts on a two byte boundary starting with B5F0. Each DAC appears every two bytes in the 8-byte area reserved for this function. When dealing with the board in high level FORTH, it may be convenient to store the desired output values in millivolts. Setting the output from that value would require multiplying the value by 100 and dividing by 244 before storing it into the DACs latches.

HEX

```
: DAC1-SCALED-10.0-MV-! DAC1-RAM @ 100 * 244 / DAC1-REG! ;
```

In this example, DAC1 is assumed to be set in the 10 volt range like it is on the SPCL-0102 board. Its desired output value is stored in millivolts in the RAM location defined as DAC1-RAM. Setting the output is accomplished by fetching the millivolt-scaled value, multiplying it by 100, dividing by 244, and putting the result into the DAC latches. We got 244 from :

12-bits = FFF (maximum number) = 4096

High end of voltage range = 10.000 volts

So each bit value is separated by  $10.000\text{volts}/4096\text{bits} = 0.0024414 \text{ volts}$   
or 2.4414 mV

The analog voltage output that is desired is obtained by writing a 12-bit hex number to the D/A. The following is an example of this.

HEX

```
: OUT-1 B5F0 ! ;
```

```
100 OUT-1
```

The same scheme can be applied to all channels. A set of useful software drivers are included in the PROGRAM SEGMENT appendix.



---

# BOARD MOUNTING

---

The SPCL-0102 has six mounting holes. Each hole is drilled at 0.110 inches, clearance for 4-40 hardware. Use caution when installing to prevent inadvertent grounding of printed circuit board traces. The mounting hole located above J6 is the only one that has a trace located nearby. Additional boards may be stacked above or below, as desired, on the female or male side of the Vertical Stacking Connector (VSC).

Common, 3/4 inch long, hex standoffs with a male screw on one end and a female threaded hole on the other are ideal interboard connection devices. The VSC connector was designed to work with this size spacer, giving reliable board to board mounting.

The length of the standard spacer, 0.750 inches, plus the board thickness, 0.061 inches, gives a nominal board to board spacing of 0.811 inches. Should an exact spacing of 0.800 inches be required, as in the case of standard mounting hardware having 0.800 inch PCB card guides, the standard spacer will have to be milled to reduce its length by 0.011 inches.

---

# DC POWER, BATTERY BACK UP, AND RESET

---

Connection TB2 provides a means to connect an external +5VDC power source or to access the on board +5VDC supply if the AC power connector is providing board power. Other connections on TB2 provide access to VBB and Ground.

The battery backup capability allows data retention in otherwise volatile CMOS RAMs and the processor's own internal RAM through main-board power-downs. A third terminal on the power connector, TB2, is marked VBB for Voltage Battery Backup.

The VBB terminal on TB2 is connected to the VBB supply rail on the board by diode, D1. The VBB supply rail supplies the three 28 pin JEDEC sockets, the 8054HN low voltage indicator in the reset circuit, one 74HC00 gate and the 74HC138 decoder. If no power is applied to the VBB terminal, the VBB rail is supplied through a P channel FET, Q1, to within a diode drop of the supplying 5 volt rail (~4.4 Volts). When the 8054HN low voltage indicator releases the reset line, Q1 is turned on and the VBB comes almost completely up to the 5 volt rail (~4.95 Volts). (This may cause some problems with the Dallas Semiconductor DS1223 battery sockets, as they "write protect" their RAMs at 4.75 Volts. Running an elevated 5 Volt supply may be necessary to accommodate these parts. The purpose of this feature is, however, to do away with the need for battery sockets in final system configurations.)

When the 8054HN low voltage indicator holds the reset line low (when VBB is below 3.8-4.2 Volts, Rev A), Q1 is turned off and the address decoder is disabled through the same input that is used by MEMDIS. This "access" protects the memories during the power down cycle.

To meet the full letter of the specifications of the parts involved, the correct backup voltage on the VBB pin is critical. This supply must be low enough to ensure that after the diode drop of D1, the VBB rail cause the 8054HN to issue a reset (~4.0 Volts), otherwise Q1 will remain on and the whole system will be powered by VBB. It must also be high enough to ensure that after the diode drop of D1, the VBB rail will meet the processors required backup voltage (listed as 4.0 Volts). Therefore, the ideal voltage for the VBB supply is 4.3-4.5 Volts. It should be pointed out, however, the Motorola specification appears to be overly conservative. By empirical test, VBB supplies below 3 Volts appear to be quite adequate. Most CMOS RAMs will retain data down to 2.2 Volts. Accounting for the diode drop under such low currents, the VBB supply may work as low as 2.5 Volts.

The processor battery backup supply enters the chip via the MODB pin. Jumper block D controls the setting of MODB, either to ground or to VBB. For backup of the processor's RAM to be successful

jumpers D and E must be in the Single Chip or Expanded Multiplexed settings. When the VBB supply is used on the processor, it will retain its User Area through power down and remember its linkages to the external FORTH dictionary.

TB3 is the power connection for the A/D and D/A sections. The center terminal is ground while the other two are supplied for +V and -V connections. The +/-V voltages can be as low as 11.7 volts or as high as 16.4 volts. A value of 12 volts is recommended with tracking plus and minus supplies for best results.

---

# AC POWER SUPPLY

---

The power supply circuit on the SPCL-0102 is designed to allow the board to operate from a simple, low-voltage, AC wall-transformer. It has three major sub circuits - rectification, regulation and DC to DC conversion. Battery backup capabilities are also provided to the 28 pin JEDEC sockets and the F68HC11 internal RAM, and a power-up power-down reset circuit.

Connection TB1 is for AC (9VAC) voltage input or for DC voltages greater than 8 volts to be input. The bridge rectifier converts the AC to DC. The 7805 regulates this rectified incoming voltage to a constant 5 Volts.

The upper limit of +V is set by the ability of the 7805 to dissipate heat. If a heat sink is added to the 7805, voltages in excess of 20 Volts are possible. Driving the 7805 to hard, however, will cause it to enter thermal overload and "shut down" its output.

The typical current required by the SPCL-0102 is 60mA when using a 9 VAC supply, 8K CMOS RAM and the Max-FORTH ROM at a 2 Mhz clock rate. The ICL232 RS-232 interface chip generates its own + and - V for RS-232 levels. A multiple stage charge pump produces +9V and -9V. The negative output is also used to get the negative voltage bias for the LCD display to increase contrast.

---

# TROUBLESHOOTING

---

As always the first thing to do when troubleshooting is to check the power and ground connections. An oscilloscope should be used to check signals. The heat sink of the 7805 is a convenient place to hook a ground clip. If +5 Volts is present at TB2 and the board is not operational, the next item to check is the oscillator. Putting the scope on EXTAL (Pin 7) should show an 8 mHz (4 mHz) sine wave running from about 0.5 Volt lows to 4.5 Volt peaks. XTAL (F68HC11 Pin 8) should have an identical signal, but of a much smaller amplitude. If the sine waves are not present and there is 5V present at the power pin Vcc (Pins 26), and ground at Vss (Pin 52), then either the F68HC11 or the crystal are bad and require replacement. There is one exception. If the processor has executed a STOP instruction, the oscillator will stop. When the oscillator is functioning correctly a 2 nHz (1 mHz) clean running square wave should be present at the E output (Pin 5). The E signal drives the timing for all external memory transfers. This signal should transition nearly rail to rail, a 0.4V low and a 4.6V high are normal. Less amplitude can indicate a board short or an excessive load on the line external to the F68HC11.

The serial channel should send a sign on message if no autostart ROM interferes. If not, the reset circuit could be bad, the serial converter could have failed, or the F68HC11 could be defective. With the reset button depressed the RES pin (Pin 17) should be at ground. When released, the pin should rise to 5 Volts. If the reset pin is working and still no message is seen on the terminal, check PD1, the serial output line (Pin 21). When reset is exercised, this line should go from normally high through a multitude of toggles back to a high state. The periods of the toggle transitions are multiples of approximately 100 microseconds. If this signal is not present, and there are no user ROMs in the board, the F68HC11 is suspect. If the signal is present, check pin 3 of the DB25F connector. It should normally be at -V (-9 Volts nominally) and should toggle to +V (+9 Volts nominally) at the same rate as the serial output line. If this is happening and no message is seen, the RS-232 wiring or the terminal is suspect. Check to see if J1 is connected to the DB25F RS-232 connector as follows:

DB9F	DB25F	Signal Name
----	-----	-----
	1	Case ground
3	2	Serial in (to SPCL-0102)
2	3	Serial out (from SPCL-0102)
5	7	Electrical ground
7 to 8	4 to 5	CTS to RTS
4 to 6	6 to 20	DSR to DTR

Check the voltages on pins 2 and 3. If pin 3 is very negative and pin 2 is floating, both systems are trying to talk on the same line. Pins 2 and 3 need to be swapped. Usually this is done with a "null modem" inserted where the two systems connect.

If the -V/+V signal was not found at pin 3, the RS-232 converter is not working. Check pin 2 of the ICL232 for +V and pin 6 of the ICL232 for -V. If these signals are not present, the charge pump of the ICL232 has failed. Pin 14 of the ICL232, the output, should look the same as pin 3 of J6.

Check pin 2 of J6 which is the serial input into the board from the terminal. It should normally be at a negative voltage between -3 and -15 Volts. When a key is pressed on the terminal it should pulse to positive voltages between +3 and +15 Volts. If it doesn't, the terminal or the RS-232 wiring are suspect. The same signals at inverted TTL levels, should also be at PD0, which is the serial input line of the processor (Pin 20).

Verify that jumpers are installed at jumper locations L and M. These jumpers route the PD0 and PD1 signals from the processor to either the RS-232 level converter or the RS-485/422 level converters. If these jumpers are missing, there will be no connection of either serial protocol to the processor.

The most common error in trying to use the SPCL-0102 is mismatched baud rates or bit settings. Verify that the terminal is set for 9600 baud with one start bit, eight data bits and one stop bit, with no parity generated. If using Forth, be sure to use CAPITALS. (Review this discussion in the Getting Started section.)



# MEMORY MAP

K#	HEX		
--	-----		
64	\$FFFF	+-----+	
63		RUN TIME	
62		KERNEL	
61			
60		NON RUN TIME	Max-FORTH ROM
59		CODES	
58			
57		HEADS	
56	\$E000		
	\$DFFF	V3.5	
		FLOATING	
		POINT	
	\$D000		
	\$CFFF		
	\$B800		
	\$B600		EEPROM
	\$B5FC	-----	LCD
	\$B5F8	-----	A/D
	\$B5F0	-----	D/A
	\$B000	=====	REGISTERS (KEYBOARD)
		~ ~	
		~ ~	
5			
4	\$1000	\$0B_AT_\$103B	
3	\$C000		
2	\$0800		
1	\$0400		
0	\$0000	+ON=CHIP=RAM=+	

# MISCELLANEOUS JUMPERS

#	SOURCE	DESTINATION	NORMALLY
-----	-----	-----	-----
<b>A</b>			
A13-A	ADDRESS LINE 13	U8 ADDRESS DECODER INPUT	
A-+5	+5 VOLT RAIL		
<b>B</b>			
A14-B	ADDRESS LINE 13	U8 ADDRESS DECODER INPUT	
B-+5	+5 VOLT RAIL		
<b>C</b>			
C0-U2	DECODER OUTPUT 0	U2 JEDEC SOCKET	
C1-U2	DECODER OUTPUT 1	U2 JEDEC SOCKET	
C2-U2	DECODER OUTPUT 2	U2 JEDEC SOCKET	
C3-U2	DECODER OUTPUT 3	U2 JEDEC SOCKET	
C4-U2	DECODER OUTPUT 4	U2 JEDEC SOCKET	
C5-U3	DECODER OUTPUT 5	U3 JEDEC SOCKET	
C6-U3	DECODER OUTPUT 6	U3 JEDEC SOCKET	
C7-U4	DECODER OUTPUT 7	U4 JEDEC SOCKET	
<b>D</b>			
GND-D	GROUND	MODB PIN	OPEN
D-+5	MODB PIN	+5 VOLT RAIL	CLOSED
<b>E</b>			
GND-E	GROUND	MODA PIN	OPEN
E-+5	MODA PIN	+5 VOLT RAIL	CLOSED
<b>F</b>			
XIRQ-F	XIRQ'	INT FROM J4	OPEN
F-IRQ	IRQ'	INT FROM J4	OPEN
<b>G</b>			
485-G	PA3	U10 PINS 2 & 3	
G-422	U10 PINS 2 & 3	GROUND	
<b>H</b>			
U2	U2 PIN 27 R/W LINE	U2 PIN 28 SUPPLY	OPEN*
<b>I</b>			
U3	U3 PIN 27 R/W LINE	U3 PIN 28 SUPPLY	OPEN*
<b>J</b>			
U4	U4 PIN 27 R/W LINE	U4 PIN 28 SUPPLY	OPEN*
<b>K</b>			
NOT USED			
<b>L</b>			
232-L	U6 PIN 10 T2I	PD1	
L-485	PD1	U10 PIN 4, U11 PIN 4	
<b>M</b>			
232-M	U6 PIN 9 R2O	PD0	
M-485	PD0	U10 PIN 1 RO	
<b>N</b>			
N0-U2	A14	U2 PIN 1	
N1-U2	Vbb +5	U2 PIN 1	
N2-U2	Vbb +5	U2 PIN 26	

	N3-U2	A13	U2 PIN 26	
	N4-U2	A14	U2 PIN 27	
	N5-U2	RR/W	U2 PIN 27	
O				
	O0-U3	A14	U3 PIN 1	
	O1-U3	Vbb +5	U3 PIN 1	
	O2-U3	Vbb +5	U3 PIN 26	
	O3-U3	A13	U3 PIN 26	
	O4-U3	A14	U3 PIN 27	
	O5-U3	RR/W	U3 PIN 27	
P				
	P0-U4	A14	U4 PIN 1	
	P1-U4	Vbb +5	U4 PIN 1	
	P2-U4	Vbb +5	U4 PIN 26	
	P3-U4	A13	U4 PIN 26	
	P4-U4	A14	U4 PIN 27	
	P5-U4	RR/W	U4 PIN 27	
J7				
	U23	PIN 10&11	GROUND	OPEN

\* Option of pullups on R/W lines to write protect RAMs in socket. To use install 100K pullup resistor & remove jumper from 28 pin JEDEC selection socket for pin 27. If battery backup is in use, RAM will then emulate ROM.

# GENERAL PURPOSE SOCKETS

## JUMPER ASSIGNMENTS FOR JEDEC 28 PIN SOCKETS

JUMPER	1	○	○ 28	+5	+	----	+
						○	
A12	2	○	○ 27	JUMPER			*
						○	
					+	----	+
A7	3	○	○ 26	JUMPER			
A6	4	○	○ 25	A8			
A5	5	○	○ 24	A9			
A4	6	○	○ 23	A11			
A3	7	○	○ 22	OE			
A2	8	○	○ 21	A10			
A1	9	○	○ 20	CHIP SELECT			
A0	10	○	○ 19	D7			
D0	11	○	○ 18	D6			
D1	12	○	○ 17	D5			
D2	13	○	○ 16	D4			
GND	14	○	○ 15	D3			

PIN 1	PIN 26	PIN 27
O---O	O---O	O---O
○	○	○
○	○	○
A14 +5	+5 A13	A14 RR/W

\* Option of pullups on R/W lines to write protect RAMs in socket. To use, install 100K pullup resistor & remove jumper for pin 27. If battery backup is in use, RAM will then emulate ROM.

# SOCKET JUMPER SETTINGS

GENERAL PURPOSE SOCKET - U2, U3, U4  
Jumper Settings for Standard JEDEC 24/28 Pin Devices

ALL 8K X 8 DEVICES  
2764, 2864, 6264

PIN 1	PIN 26	PIN 27
	X	X
	X	X

A14 +5V +5V A13 A14 RR/W

16K X 8 EPROM  
27128

PIN 1	PIN 26	PIN 27
	X	X
	X	X

A14 +5V +5V A13 A14 RR/W

32K X 8 EPROM  
27256

PIN 1	PIN 26	PIN 27
	X	X
	X	X

A14 +5V +5V A13 A14 RR/W

32K X 8 RAM  
62256

PIN 1	PIN 26	PIN 27
X		X
X		X

A14 +5V +5V A13 A14 RR/W

\* To write protect RAMs in socket, install 100K pullup resistor between pins 28 and 27 of the JEDEC socket. Jumper locations H, I, and J have been provided on the board for this purpose. After the pullup resistor is installed remove jumper for pin 27. If battery backup is being used, RAM will now emulate ROM.

# MEMORY ADDRESS SETTINGS

## THREE 8K DEVICES

```

+5V  A  A13
+-----+
|  o  XXXX  |
+-----+ 8K POSITION
|  o  XXXX  |
+-----+
+5V  B  A14

      C
+-----+
0000-1FFF O0 | XXXX | --U2 CHIP SELECT 0000
              |      | | 1FFF
2000-3FFF O1 |  o  o | --
              |      | |
4000-5FFF O2 |  o  o | --
              |      | |
6000-7FFF O3 |  o  o | --
              |      | |
8000-9FFF O4 |  o  o | --
              |      | |
A000-BFFF O5 |  o  o | --U3 CHIP SELECT C000
              |      | | DFFF
C000-DFFF O6 | XXXX | --
              |      | |
E000-FFFF O7 | XXXX | ---U4 CHIP SELECT E000
              |      | | FFFF
+-----+

```

# THREE 16K DEVICES

```

+5V  A  A13
+-----+
|  XXXX  o  |
+-----+ 16K POSITION
|  o  XXXX  |
+-----+
+5V  B  A14

      C
+-----+
00 |  o  o  | -+
   |      |   |
0000-3FFF 01 |  XXXX  | -+ -U2 CHIP SELECT ----
   |      |   |
   |      |   |
   |      |   |
4000-7FFF 03 |  o  o  | -+
   |      |   |
   |      |   |
   |      |   |
04 |  o  o  | -+
   |      |   |
   |      |   |
   |      |   |
8000-BFFF 05 |  XXXX  | -+ -U3 CHIP SELECT ----
   |      |   |
   |      |   |
   |      |   |
06 |  o  o  | -+
   |      |   |
   |      |   |
   |      |   |
C000-FFFF 07 |  XXXX  | ---U4 CHIP SELECT ----
   |      |   |
   |      |   |
   |      |   |
+-----+

```

# TWO 32K DEVICES

```

+5V  A  A13
+-----+
|  XXXX  o  |
+-----+ 32K POSITION
|  XXXX  o  |
+-----+
+5V  B  A14

      C
+-----+
00 | o  o  | -+
   |      |   |
01 | o  o  | -+
   |      |   |
02 | o  o  | -+
   |      |   |
0000-7FFF 03 | XXXX | +-U2 CHIP SELECT ---- 0000
   |      |   |                                     7FFF
04 | o  o  | -+
   |      |   |
05 | o  o  | +-U3 CHIP SELECT ----
   |      |   |
06 | o  o  | -+
   |      |   |
8000-FFFF 07 | XXXX | ---U4 CHIP SELECT ---- 8000
   |      |   |                                     FFFF
+-----+

```



---

# INPUT/OUTPUT JACKS

---

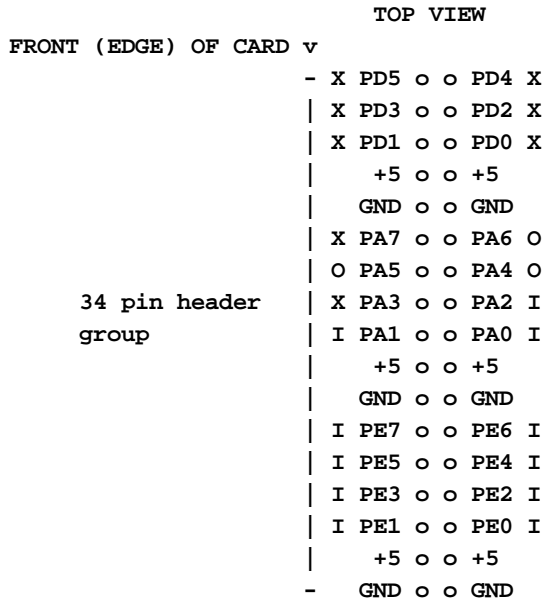
## SERIAL INPUT/OUTPUT JACK J6

TOP VIEW  
NUMBERED LEFT TO RIGHT

1 2 3 4 5 6 7 8 9 10 11 12 13 14  
-----  
o o o o o o o o o o o o o o o

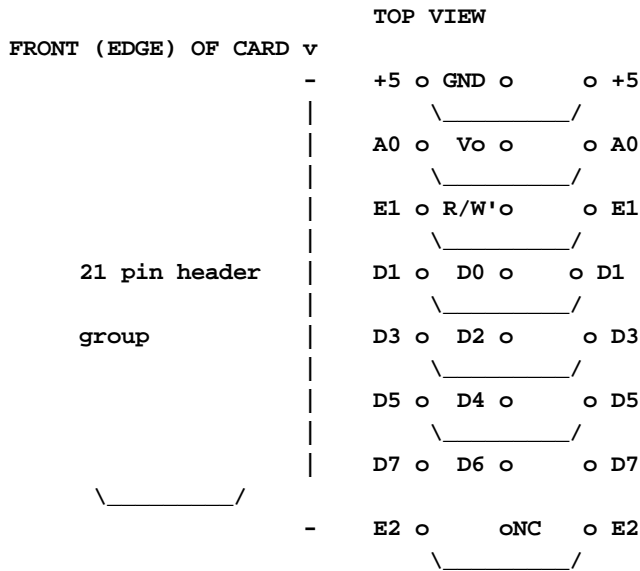
DB25F	J6	Signal Name
		-----
	1	Spare RS-232 in
	2	Spare RS-232 out
	3	Spare TTL receiver out
	4	Spare TTL transmitter in
1	5	Case ground
2	6	Serial into SPCL-0102
3	7	Serial out of SPCL-0102
7	8	Electrical ground
	9	Reset line in or out
	10	Electrical ground
	11	RS-422 Receive + Differential input or 485 xcv
	12	RS-422 Receive - Differential input or 485 xcv
	13	RS-422 Receive + Differential output
	14	RS-422 Receive - Differential output

# INPUT/OUTPUT JACK J2



I=INPUT O=OUTPUT X=EITHER

# LCD DISPLAY JACK J1



#### VSC34 EXPANSION JACK J4

MEMDIS	o	o	N.C.
E	o	o	RST
A15	o	o	INT
A14	o	o	+5
A12	o	o	R/W
A7	o	o	A13
A6	o	o	A8
A5	o	o	A9
A4	o	o	A11
A3	o	o	OE
A2	o	o	A10
A1	o	o	AS
A0	o	o	D7
D0	o	o	D6
D1	o	o	D5
D2	o	o	D4
GND	o	o	D3

The J4 expansion connector was designed to follow the JEDEC standard for byte sized memory parts in the 8, 16 and 32K Byte varieties. The J4 connector on these boards are made to most closely match the more recently available 32K JEDEC parts.

---

# PROGRAM SEGMENTS

---

COLD

FORGET TASK

HEX

100 1C ! ( V3.3 ONLY!

50 1E ! ( V3.3 ONLY!

400 DP !

```
(
*****
)
(LCD DISPLAY ROUTINES )
(
*****
)
```

: IS CONSTANT ;

B5FC IS DSP-CMD

B5FD IS DSP-DATA

```
: WAIT-NOT-BUSY BEGIN DSP-CMD C@ 80 AND 0= UNTIL ;
: CLEAR WAIT-NOT-BUSY 1 DSP-CMD C! ;
: HOME WAIT-NOT-BUSY 2 DSP-CMD C! ;
: CRLF WAIT-NOT-BUSY C0 DSP-CMD C! ;
: MOVE-CURSOR WAIT-NOT-BUSY 80 OR DSP-CMD C! ;
: RIGHT-UPPER-CORNER 27 MOVE-CURSOR ;
: CURSOR? WAIT-NOT-BUSY DSP-CMD C@ 7F AND ;
: DSP>L WAIT-NOT-BUSY 10 DSP-CMD C! CURSOR? 27 >
  IF RIGHT-UPPER-CORNER THEN ;
: DSP>R WAIT-NOT-BUSY 14 DSP-CMD C! CURSOR? 27 >
  IF HOME THEN ;

: DSP-EMIT WAIT-NOT-BUSY DSP-DATA C! ;

: DSP-TYPE
  BEGIN
    DUP 0= NOT
  WHILE
    1- SWAP DUP C@ DSP-EMIT 1+ SWAP
  REPEAT
  2DROP
;

: DSP-SPACE BL DSP-EMIT ;
: DSP-SPACES 0 MAX BEGIN ?DUP WHILE 1- DSP-SPACE REPEAT ;
: DSP-ON
  WAIT-NOT-BUSY
```

```

38 DSP-CMD C! ( GET ATTN
38 DSP-CMD C! ( SET 2 LINE DISP )
6 DSP-CMD C! ( CHARACTER ENTRY RIGHT )
E DSP-CMD C! ( DISPLAY CONTROL ON, CURSOR ON )
;

( *****
* )
( KEYPAD ROUTINES )
( *****
* )

BOOA IS KEYPAD

: KP-?TERMINAL B000 C@ 1 AND ;

: KP-KEY
BEGIN
  KP-?TERMINAL
UNTIL
KEYPAD C@ 2/ 2/ 2/
;

: KP-EXPECT
0 SPAN ! DUP
IF
  OVER + OVER
  BEGIN
    KP-KEY DUP OD ( OR DEFINE ENTER KEY INSTEAD OF OD) =
    IF
      DROP SPACE 1
    ELSE
      DUP 08 ( OR DEFINE BACKSPACE KEY INSTEAD OF 08 ) =
      IF
        DROP 2 PICK OVER U<
        IF
          SPAN 1-! 1- 08 DUP EMIT SPACE EMIT
        THEN
          0
        ELSE
          2DUP SWAP C! EMIT
          SPAN 1+!
          1+
          2DUP =
        THEN
          THEN
        UNTIL
      DROP
    THEN
  TDROP
;

```

# SPCL-0102 PARTS LIST

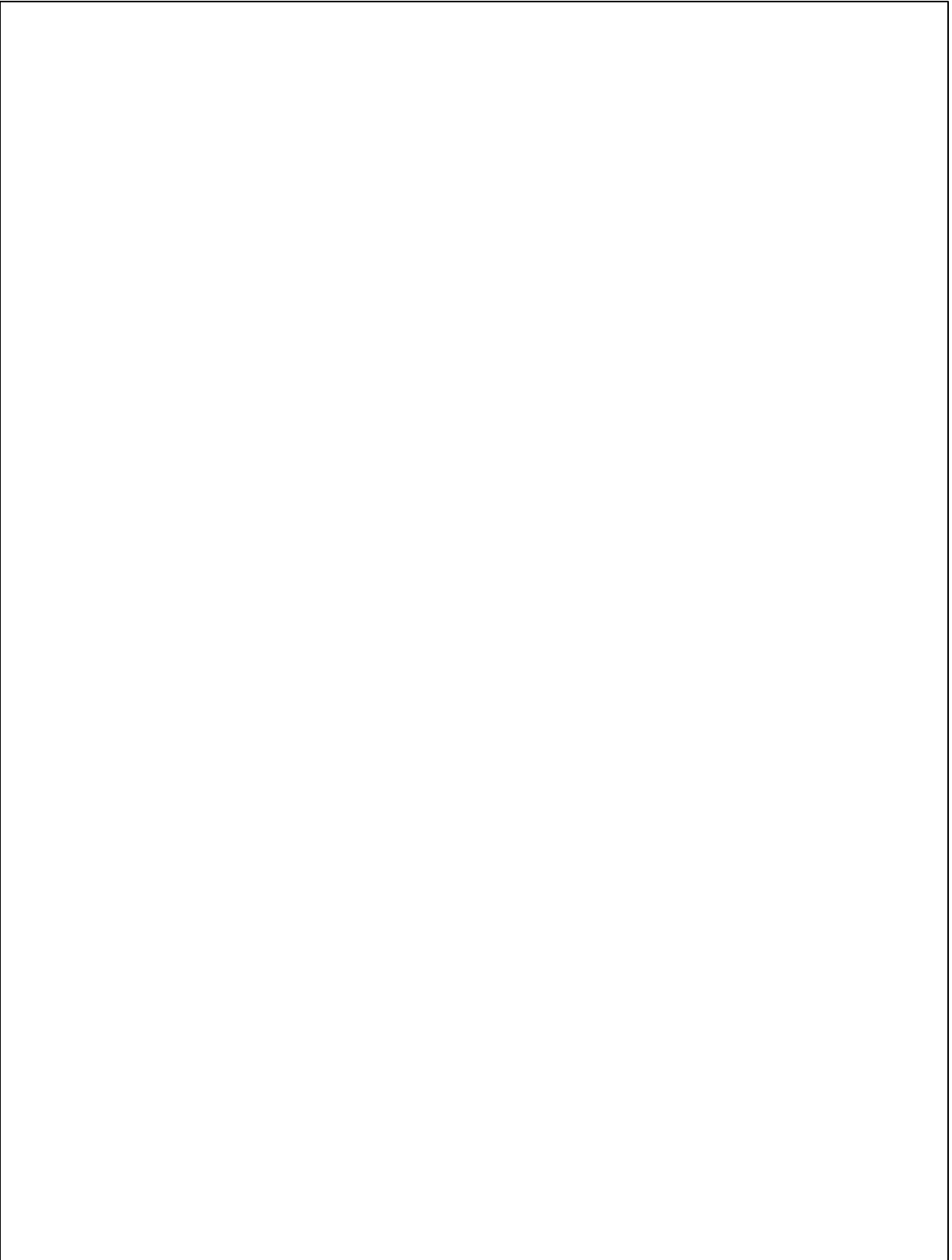
SPCL-0102 PARTS LIST REV 1.0

REF.DES.	VALUE	COMPONENT
C5,6, C14,15	.1UF	CAPACITOR, MONO.
C23-37		
C1, 4, 17, 19, 21*	. 1UF	CAPACITOR, MONO.
C9-12,22	10UF	CAPACITOR, ELECT.
C16,18,20*	10UF	CAPACITOR, ELECT.
C2,3*	220UF	CAPACITOR, ELECT.
C7,C8	22PF	CAPACITOR, CER. DISK
R0,R2-5	10K,1/8W,5%	RESISTORS
R1	10M,1/8W,5%	RESISTORS
R6	2.2K,1/4W,1%	RESISTORS
R7	10K	TRIM-POT
R8-10*	10 OHM,1/8W,1%	RESISTORS
R11	10 OHM,1/8W,1%	RESISTORS
R12	100k, 1/8W,1%	RESISTORS
R13	100 OHM,1/8W,1%	RESISTOR
R14,15	19.6K,1/8W, 1%	RESISTORS
R16,17	1.5K,1/8W,1%	RESISTORS
RV2, RV1	100K	TRIM-POT
RV3	10K	TRIM-POT
D1-6	1N4148	DIODE
D7 (NOT USED)		
Q1	VPO300M	FET
VR1*	LM7805	REGULATOR
BR1*	VMO8	RECTIFIER
TB1*	2 PIN	TERM. BLOCK
TB2,3	3 PIN	TERM. BLOCK
Y1	8MHz	XTAL, LOW PROFILE
LV11	8054	IC
U5,U7	74HC00	IC
U8,19,20	74HC138	IC
U9	74HC373	IC
U1	F68HC11	IC
U14	74HC27	IC
U12	74HC133	IC
U13	MM74C923	IC
U6	MAX232	IC
U10,U11*	75176	IC
U2	6264	IC
U21	74HC574	IC
U22	ADS574	IC
U23	AD625	IC
U24*	HI3-548-5	IC

U25  
U15,16,17\*

HI3-548-5  
AD7248

IC  
IC



U18

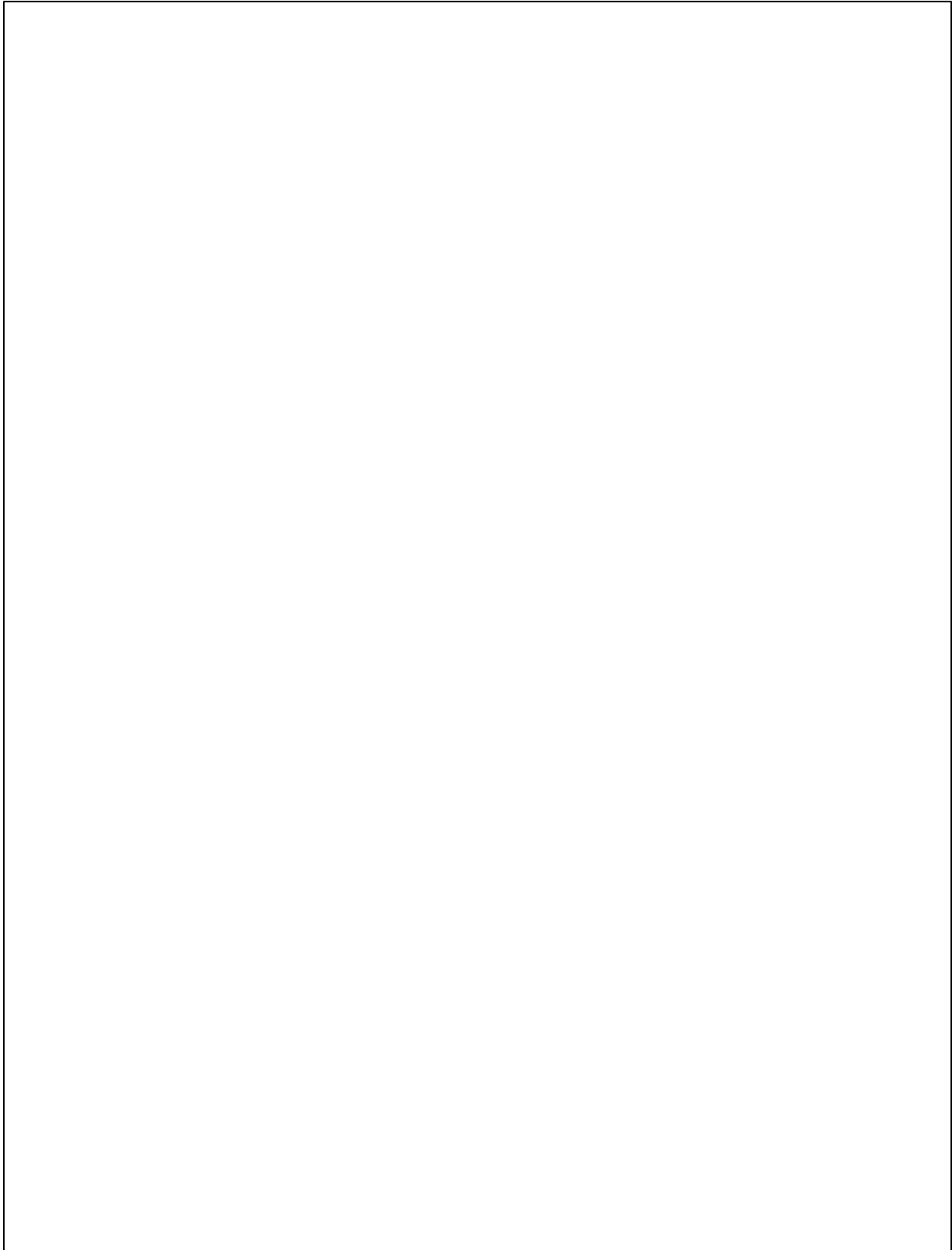
U10,U11\*

AD7248

8 PIN

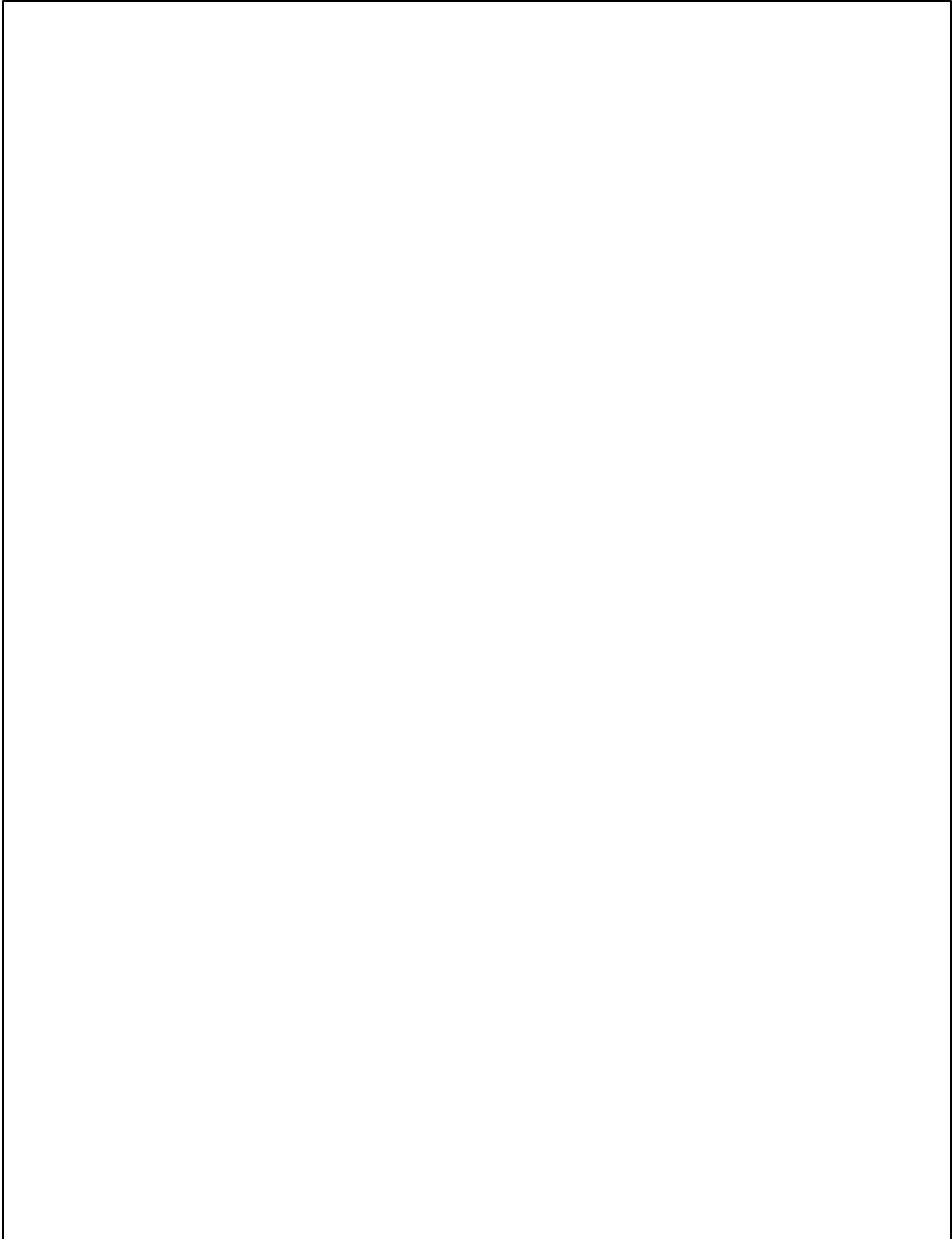
IC

SOCKET





U5,U7	14 PIN	SOCKET
U8,U12	16 PIN	SOCKET
U9,U13	20 PIN	SOCKET



U2,U3,U4	28 PIN	SOCKET
U1	52 PIN	SOCKET
J4*	34 VSC	SPACER
	34 PIN VSC	CONNECTOR
J9	1x9	HEADER PINS
J7	1x2	HEADER PINS
J6	1x6	HEADER PINS
J3	1x10	HEADER PINS
J1	1x8 (3)	HEADER PINS
	SPCL-0102	PCB
	SHUNTS	

\* Note: Parts not included in SPCL-0102, are included in SPCL-0004 X20.

---

# APPLICATION NOTE

---

## INTEL FORMAT DUMP COMMAND

The following program allows a section of memory to be dumped out the serial channel in the Intel hex format which is a standard used by many of the commercially available PROM programmers. This program should allow the use of such programmers to capture programs and data in EPROMs, which are not supported for programming by the SPCL-0102 directly.

### HEX

### VARIABLE CHKSUM

```
: CE DUP A < IF 30 ELSE 37 THEN + EMIT ; ( CONVERT AND EMIT )
: 2. R FF AND 10 /MOD CE CE ;
: 4. R 0 100 UM/MOD 2. R 2. R ;

: INTEL-DUMP ( addr count --- )
  OVER + SWAP ( CONVERTS ADDR & COUNT TO UPPER, LOWER ADDR )
  BEGIN
    CR
    2DUP 20 + MIN ( MAKE NEXT LINE OF OUTPUT UP TO 32 BYTES
LONG)
    SWAP ( BRING UP START ADDRESS, MOVE DOWN END ADDRESS )
    . " : " ( BEGIN THE RECORD )
    2DUP - ( FIND OUT # OF BYTES IN THIS RECORD )
    DUP CHKSUM ! ( BEGIN CHKSUM COMPUTATION )
    2. R ( PRINT # OF BYTES IN RECORD IN TWO DIGIT FIELD )
    DUP 100 /MOD + CHKSUM +! ( ADD START ADDRESS TO CHKSUM )
    DUP 4. R ( PRINT START ADDRESS IN FOUR DIGIT FIELD )
    . " 00" ( PRINT RECORD TYPE, NO NEED TO ADD TO CHKSUM )
    >R DUP R> ( MAKE START STOP #S FOR DO LOOP )
    DO
      I C@ 2. R ( PRINT HEX BYTE IN TWO DIGIT FIELD )
      I C@ CHKSUM +! ( UPDATE CHKSUM )
    LOOP
    CHKSUM @ FF AND NEGATE 2. R ( PRINT CHKSUM NEG 2 DIGIT
FIELD )
    2DUP =
    UNTIL ( KEEP GOING TILL LINE END IS = TO BLOCK END )
    CR . " : 00000001FF" CR ( TACK ON END RECORD )
    2DROP
  ;
```

Program and application courtesy of Danny Barger, International Computing Scale.

---

# CPU SCHEMATIC

---

---

## **A/D,D/A CONVERTER SCHEMATIC**

---

---

# SILKSCREEN

---