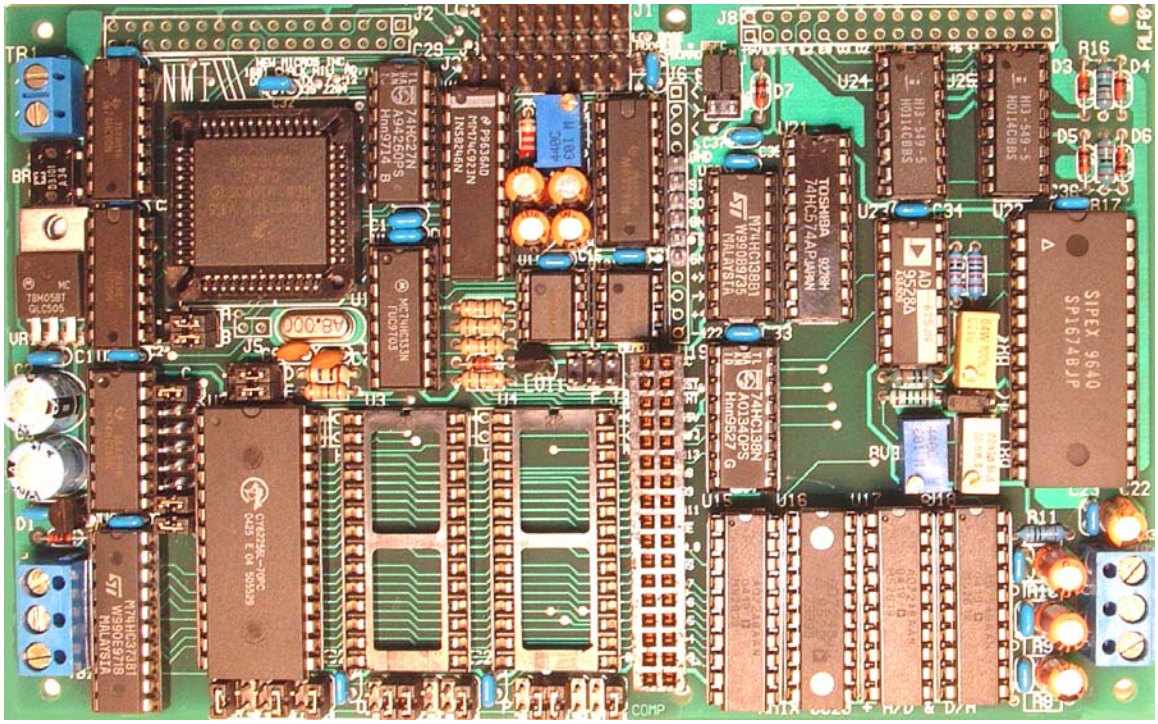
SPCL-000x-X/T20

User's Manual



GETTING STARTED

The SPCL-0001,2,3,4, and 7-X20 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-000x-T20 is a special target version of the SPCL-000x-X20 but has fewer parts installed. Normally, a developer will use the SPCL-000x-X20 for development, and high end projects, then switch to the lower cost SPCL-000x-T20, or one of its other NMI derivatives when volume buying begins. The SPCL-000x-X20 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-000x-T20 at much lower cost.

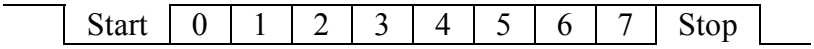
To operate the SPCL-000x-X20 system, power and communications connections must first be made. Three power supply voltages are required. The +5V powers all the digital circuitry. The computer section will operate with only the +5V and GND connections. This allows language development that does not require use of the data acquisition section. Analog supplies are necessary to make the A/D and D/A sections fully functional. The +/-12V up to +/- 15V 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 only provided on the SPCL-000x-X20, but not on the SPCL-000x-T20.

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	DB25	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	DTR to DSR

'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-000x-X20, 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-000x-X20 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-000x-X20 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. Basic11 and C are also available for use with the SPCL-000x-X20.

The signon message means the terminal is communicating with the SPCL-000x-X20. Press "return" on your terminal several times. If the SPCL-000x-X20 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-000x-X20 is now running and communicating as it should with the terminal..

-X20 AND -T20 DIFFERENCES

The SPCL-000x-X20 is an "X" version in the NMI board series where the SPCL-000x-T20, 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-000x-X20 has one addition other "T" boards may not

have. The RS-232 converter is included with the SPCL-000x-X20. This allows the SPCL-000x-X20 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-000x-X20 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

GND	+5V	PE1 (I)	PE3 (I)	PE5 (I)	PE7 (I)	GND	+5V	PA1 (I)	PA3 (X)	PA5 (O)	PA7 (X)	GND	+5V	PD1 (X)	PD3 (X)	PD5 (X)
33	31	29	27	25	23	21	19	17	15	13	11	9	7	5	3	1
34	32	30	28	26	24	22	20	18	16	14	12	10	8	6	4	2
GND	+5V	PE0 (I)	PE2 (I)	PE4 (I)	PE6 (I)	GND	+5V	PA0 (I)	PA2 (I)	PA4 (O)	PA6 (O)	GND	+5V	PD0 (X)	PD2 (X)	PD4 (X)

(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 Darlingtons 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-000X-X20 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

E2 (22)	D7 (19)	D5 (16)	D3 (13)	D1 (10)	E1 (7)	A0 (4)	+5V (1)
NC (23)	D6 (20)	D4 (17)	D2 (14)	D0 (11)	R/W (8)	Vo (5)	GND (2)
E2 (24)	D7 (21)	D5 (18)	D3 (15)	D1 (12)	E1 (9)	A0 (6)	+5V (3)

Signal (Pin #)

Keypad connectors: J3, J9

Y5	Y4	Y3	Y2	Y1	X4	X3	X2	X1	NC
10	9	8	7	6	5	4	3	2	1

J9

Y5	Y4	X4	X1	Y3	X2	X3	Y2	Y1
9	8	7	6	5	4	3	2	1

J3

X: Row, Y: Column, or vice versa

The SPCL-000x-X20 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-000X-X20 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.)

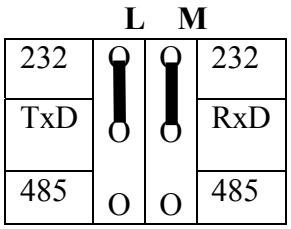
The RS-422 standard represents a newer interface now coming into popularity, and with good reason. Unlike the RS-232 requirements which specify a single wire voltage transmission referenced to ground, the RS-422 standard uses a voltage differential on a pair of conductors. While the RS-232 at full voltage 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.

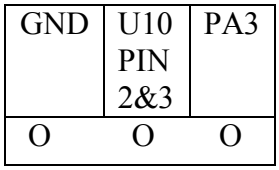
These two interfaces are accommodated on the SPCL-000x-X20 by the addition of two 8 pin 75176's, which each contain a transmitter/receiver pair. Whether the transmitter of the pair is active, or not, is controlled by a signal on one of its pins.

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-000x-X20 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.

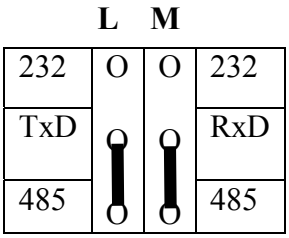
RS-232 JUMPER CONFIGURATION



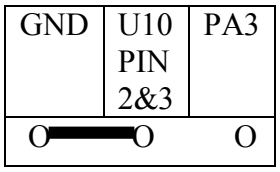
G: Don't Care



RS-422 JUMPER CONFIGURATION

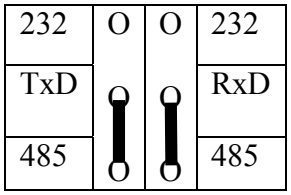


G

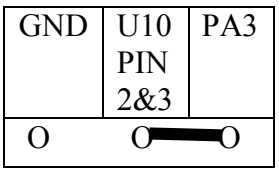


RS-485 JUMPER CONFIGURATION

L M



G



ADDRESS DECODING

The chip selects of the three JEDEC sockets are generated by one 74HC138, U8. When jumpers A and B are in the 8K position, address lines A15 to A13 are brought to this part. This means that each of the eight generated chip selects represent a single 8K byte segment out of the 64K byte memory map.

When jumpers A and B are in the 16K position, address lines A15 and A14 are brought to this part. The A13 is held high. This means that the upper four generated chip selects represent a single 16K byte segment out of the 64K byte memory map.

When jumpers A and B are in the 32K position, address line A15 alone controls the part. Address lines A14 and A13 are held high. This means that each of the two upper chip selects represent a 32K byte segment out of the 64K byte memory map.

Two other signals control the decoder - Address Strobe (AS') and On Board Memory Disable (MEMDIS'). The Address Strobe (AS') signal must be active low before any chip selects are enabled. This is the processor's signal indicating the address on the bus is valid for the off-chip memory. The On Board Memory Disable (MEMDIS') signal allows an off-board open collector source to disable the on board decoder, so off-board components can usurp a memory segment from on board memory, even if the entire 64K is filled with RAM on the main board.

Analog Input/Output Jack, J8

	GND	EN6	EN4	EN2	EN0	GND	GND	GND	GND	-IN7	-IN6	-IN5	-IN4	-IN3	-IN2	-IN1	-IN0
J8	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34
	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33
	+5V	EN7	EN5	EN3	EN1	DA3	DA2	DA1	DA0	+IN7	+IN6	+IN5	+IN4	+IN3	+IN2	+IN1	+IN0

EN0-EN7 : Digital Output signals which is used to select the analog inputs to the MUX
DA0-DA3 : Analog Outputs, 0 to +10V
+/-IN0 - +/-IN7: Analog Inputs, 0 to +10V

The 12-bit DAC is memory mapped to B5F0 - B5F7 hex
The 8-bit MUX, EN0-EN7 is memory mapped to B5F8 - B5F9 hex
The 12-bit ADC is memory mapped to B5FA - B5FB hex

A/D SUBSECTION

The SPCL-000x-X20, 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-000x-X20 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 H1-549 analog multiplexers, U24 and U25.. Only U25 is installed on the SCPL-0102 version of the SPCL-0004-X20. The H1-549 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 U21and 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-000x-X20 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 a least working. The leftmost column (column 0) displays the digital output from channel 0. The next cloumn 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
    0 B5FA C! ( force conversion
      DELAY
      B5FA @ 10 / FFF AND U. ( calculate and display
    CR
   UNTIL  ( Loop until key is depressed
   ?TERMINAL  ( wait for input
   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-000x-X20 is now running and communicating as it should.

PROGRAMMING

Writing drivers for the SPCL-000x-X20 A/D is not difficult. The ADC574 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 of second byte. All programing 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-000x-X20 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-000x-X20 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 epresenting 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-000X-X20. 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-000x-X20 is not difficult. Each AD7248 is 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-000x-X20 board. Each bit count difference represents 2.44mV. Full scale is 4095 x 2.44 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-000X-X20 board. It's desired output value is stored in millivolts in the RAM location defined as DAC1-RAM. Setting the output is accomplished by fetching tyhe millivolt-scaled value, mulitplying 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 seperated by 10.000volts/4096bits = 0.0024414 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 and example of this.

```
HEX

: OUT-1 B5F0 ! ;

100 OUT-1
```

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

BOARD MOUNTING

The SPCL-000x-X20 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 workl 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-000x-X20 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-000x-X20 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 MHz (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:

J6	DB9F	DB25	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	DTR to DSR

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.

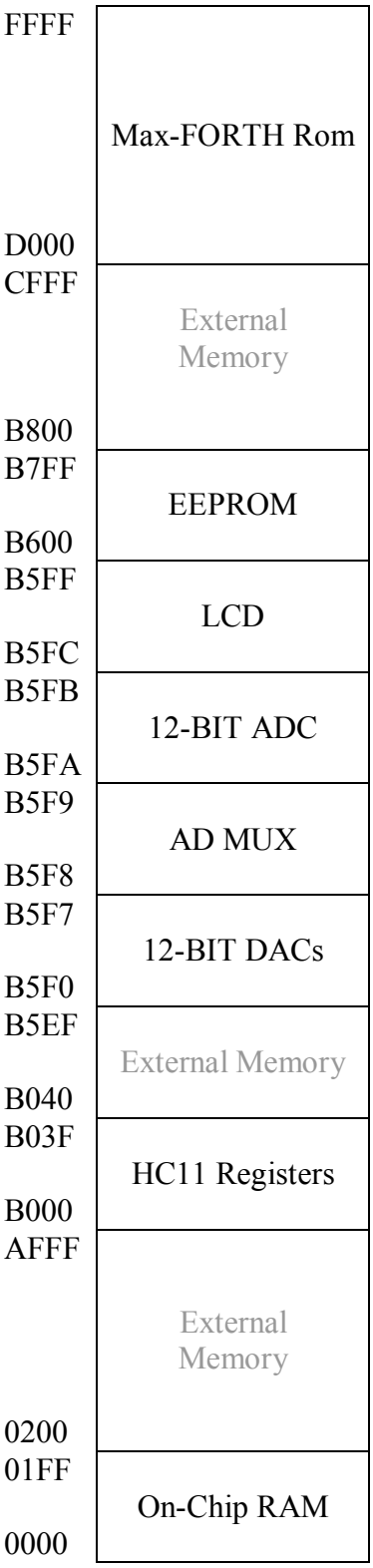
If the -V/+V signal was not found at pin 3, the RS-232 converter is not working. Check pin 2 of the ICL232 for +V and pin 6 of the ICL232 for -V. If these signals are not present, the charge pump of the ICL232 has failed. Pin 14 of the ICL232, the output, should look the same as pin 3 of J6.

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-000x-X20 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


FFFF
Max-FORTH Rom
D000
CFFF
External Memory
B800
B7FF
EEPROM
B600
B5FF
LCD
B5FC
B5FB
12-BIT ADC
B5FA
B5F9
AD MUX
B5F8
B5F7
12-BIT DACs
B5F0
B5EF
External Memory
B040
B03F
HC11 Registers
B000
AFFF
External Memory
0200
01FF
On-Chip RAM
0000

GENERAL PURPOSE SOCKETS

JUMPER ASSIGNMENTS FOR JEDEC 28 PIN SOCKETS:

U2, U3, U4

Jumper	1	28	+5V
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	CS’
A0	10	19	D7
D0	11	18	D6
D1	12	17	D5
D2	13	16	D4
GND	14	15	D3

H, I, J

VBB	2
Pin 27	1

N, O, P

PIN 1		PIN26		PIN27	
O—O		O—O		O—O	
O	O	O	O	O	O
A14	+5V	+5V	A13	A14	RR/W

Jumper ***H, I, J*** are option of pull-ups on R/W lines to write protect RAM in socket. To use, install 100K pull-up resistor & remove jumper RR/W 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	
O	O	O	O	O	O
	||	||			||
O	O	O	O	O	O
A14	+5V	+5V	A13	A14	RR/W

16K X 8 EPROM
27128

PIN 1		PIN 26		PIN 27	
O	O	O	O	O	O
	||		||		||
O	O	O	O	O	O
A14	+5V	+5V	A13	A14	RR/W

32K X 8 EPROM
27C256

PIN 1		PIN 26		PIN 27	
O	O	O	O	O	O
	||		||	||	
O	O	O	O	O	O
A14	+5V	+5V	A13	A14	RR/W

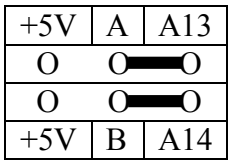
32K X 8 RAM /EEPROM
62256/28C256

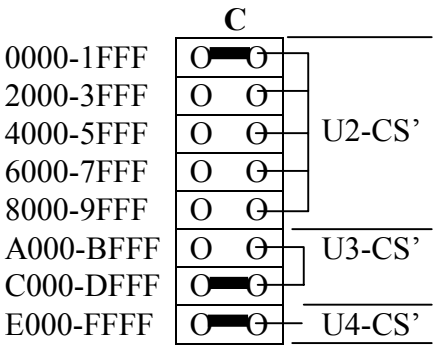
PIN 1		PIN 26		PIN 27	
O	O	O	O	O	O
||			||		||
O	O	O	O	O	O
A14	+5V	+5V	A13	A14	RR/W

To write protect RAMs in socket, install 100K pull-up 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 pull-up resistor is installed remove RR/W jumper for pin 27. If battery backup is being used, RAM will now emulate ROM.

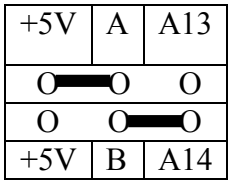
MEMORY ADDRESS SETTINGS

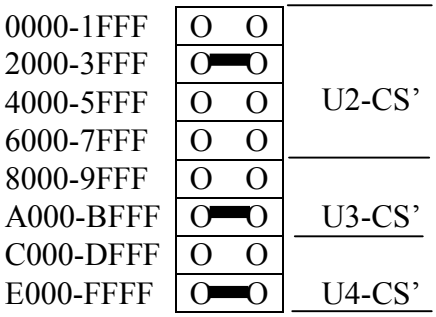
THREE 8K DEVICES



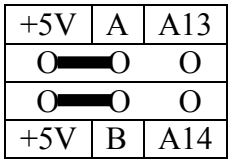


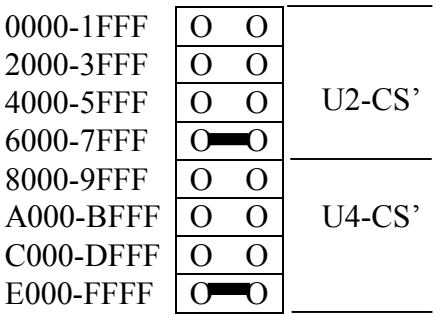
THREE 16K DEVICES





TWO 32K DEVICES





MISCELLANEOUS JUMPERS

Operating mode jumpers: D, E

GND	MODB	VBB
O	O	O
O	O	O
GND	MODA	PULL-UP

D

E

MODB	MODA	MODE
1	0	Single Chip
1	1	Expanded (default)
0	0	Bootstrap
0	1	Special test

Pheripheral Interrupt jumper: F

XIRQ'	INT	IRQ'
3	2	1

F

1 & 2 : IRQ' to INT on J4
2 & 3: XIRQ' to INT on J4

RS-422/485 jumper: G

422	U10 pin 2&3	PA3
3	2	1

G

1 & 2: RS-485 Transceiver controls by PA3
2 & 3: RS-422 Receiver always enable

RS-232/422/485 Receiver/Transmitter jumpers: L, M

232	PD1/TxD	485
3	2	1

L

1 & 2: Serial Data Input to RS-422/485
2 & 3: Serial Data Input to RS-232

232	PD0/RxD	485
3	2	1

M

1 & 2: Serial Data Output from RS-422/485
2 & 3: Serial Data Output from RS-232

SERIAL INPUT/OUTPUT JACK J6

Pin	***Signal name***
1	Spare RS-232 In
2	Spare RS-232 Out
3	Spare TTL Receiver Out
4	Spare TTL Transmitter In
5	Case Ground
6	Serial into SPCL-000x-X20
7	Serial out of SPCL-000x-X20
8	Electrical Ground
9	MPU Reset Line
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

VSC34 EXPANSION JACK J4

MEMDIS'	34	33	NC
E	32	31	RST'
A15	30	29	INT'
A14	28	27	+5V
A12	26	25	R/W'
A7	24	23	A13
A6	22	21	A8
A5	20	19	A9
A4	18	17	A11
A3	16	15	OE'
A2	14	13	A10
A1	12	11	AS'
A0	10	9	D7
D0	8	7	D6
D1	6	5	D5
D2	4	3	D4
GND	2	1	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

```
( GENERIC LCD OUTPUT ROUTINES
( -----

HEX

( LCD DISPLAY ROUTINES
( -----

B5FC CONSTANT DSP-CMD
B5FD CONSTANT DSP-DATA

: WAIT-NOT-BUSY    BEGIN DSP-CMD C@ 80 AND 0= UNTIL ;

: DSP-CLEAR   WAIT-NOT-BUSY 1 DSP-CMD C! ;

: DSP-HOME   WAIT-NOT-BUSY 2 DSP-CMD C! ;

28 CONSTANT DC/L

( position cursor on top 00-27 or bottom 28-4F line -- 2x40 DISPLAY
: DSP-AT ( n -- )
   DUP DC/L MOD  80 OR  SWAP DC/L 1- > IF  40 OR  THEN
   WAIT-NOT-BUSY DSP-CMD C! ;

( return cursor position
: CURSOR? ( -- n )
   WAIT-NOT-BUSY DSP-CMD C@ 7F AND ;

( print a character, move cursor to left
: DSP-EMIT ( char -- )   WAIT-NOT-BUSY DSP-DATA C! ;

( print a string on the lcd, writes past end of line.....
: DSP-TYPE ( addr n -- )
   ?DUP IF  0 DO  COUNT DSP-EMIT  LOOP  THEN DROP ;

( init the display
: DSP-ON
   WAIT-NOT-BUSY
   38 DSP-CMD C!   ( GET ATTN )
   38 DSP-CMD C!   ( SET 2 LINE DISP, 2x40 )
    6 DSP-CMD C!   ( CHARACTER ENTRY RIGHT )
    C DSP-CMD C! ; ( DISPLAY CONTROL ON, CURSOR OFF )
```

```
( COMPILE NON-PRINTING STRINGS
( -----

: (")
  R@ 2+ COUNT  DUP 1+ R> + >R ;

: "
  COMPILE (") 22 WORD C@ 1+ ALLOT ; IMMEDIATE

: DSP-SPACES ( n -- )
  ?DUP IF 0 DO BL DSP-EMIT LOOP THEN ;

: (D.) ( d -- addr # )
  SWAP OVER DABS  <# #S SIGN #> ;

: DSP-D. ( d -- )
  (D.) DSP-TYPE BL DSP-EMIT ;

: DSP-D.R ( d n -- )
  >R (D.)  R> OVER - 0 MAX DSP-SPACES  DSP-TYPE ;

: DSP-.  ( n -- )    S->D DSP-D. ;
: DSP-F. ( fp -- )   (F.) DSP-TYPE ;
: DSP-E. ( fp -- )   (E.) DSP-TYPE ;

: TEST
  DSP-ON
  DSP-CLEAR " THIS IS A TEST " DSP-TYPE              KEY DROP
  DSP-CLEAR " PI IS "          DSP-TYPE  PI    DSP-F. KEY DROP
  DSP-CLEAR " A NUMBER "       DSP-TYPE  1234. DSP-D. KEY DROP ;

( ---------------------------- )
( KEYPAD ROUTINES  )
( ---------------------------- )
HEX

B00A CONSTANT KEYPAD

( true if a keypad key is pressed
: KP-?TERMINAL ( -- flag )   B000 C@ 1 AND ;

( wait for keypad key to be released
: KP-RELEASE ( -- )   BEGIN KP-?TERMINAL 0= UNTIL ;

( wait for a keypad key
: KP-KEY ( -- button )
```

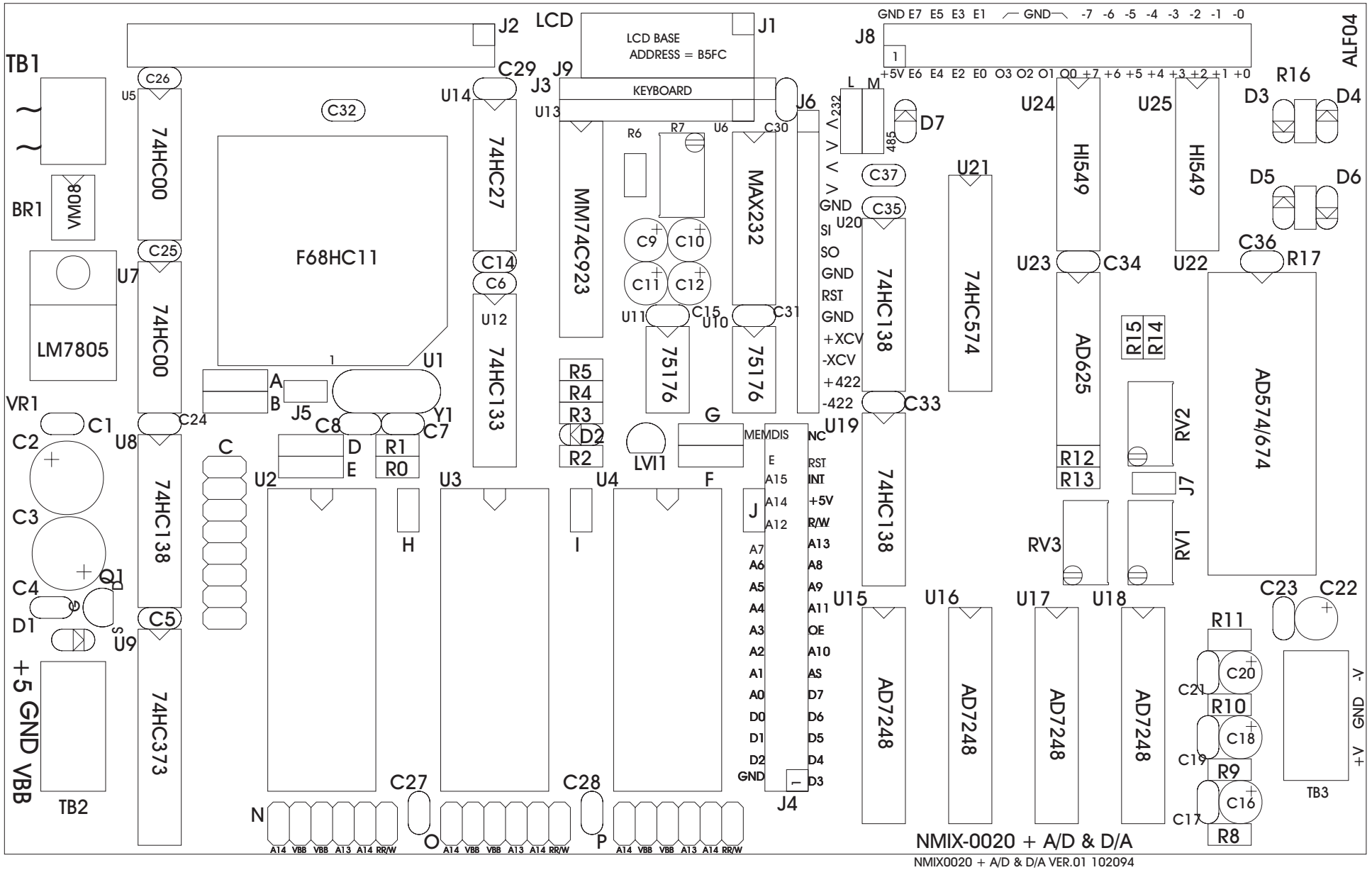
```
  BEGIN KP-?TERMINAL UNTIL  KEYPAD C@  2/ 2/ 2/ ;

CREATE ASCIIFY  ( a table for keypad translation)
   31 C, 34 C, 37 C, 45 C,      (   1    4    7  Edit )
   32 C, 35 C, 38 C, 30 C,      (   2    5    8    0  )
   33 C, 36 C, 39 C, 0D C,      (   3    6    9  entr )
   50 C, 55 C, 44 C, 54 C,      ( Prog Up   Down Trav )
   4D C, 48 C, 08 C, 53 C,      ( Mode Home Bksp Stop )

( return the ascii version of the keys, wait for key release
: PADKEY ( -- char )
   KP-KEY ASCIIFY + C@  KP-RELEASE ;


( NUMBER INPUT VIA KEYPAD AND DISPLAY

( true if char is not a digit
: NOTDIGIT? ( char -- char flag )
   DUP 30 <  OVER 39 > OR ;

( wait for a single digit. reject any non-digit and continue waiting
( return the value of the digit 0--9
: 1DIGIT ( -- n )
   BEGIN PADKEY NOTDIGIT? WHILE DROP REPEAT DUP DSP-EMIT  30 - ;

( wait for n digits, return the accumulated value
: DIGITS ( n -- n )
   0 SWAP 0 DO 0A * 1DIGIT + LOOP ;

( user input of a number
: GET-NUMBER ( -- n )
   DSP-CLEAR (  0 DSP-AT)  DSP" INPUT? " 5 DIGITS ;
```

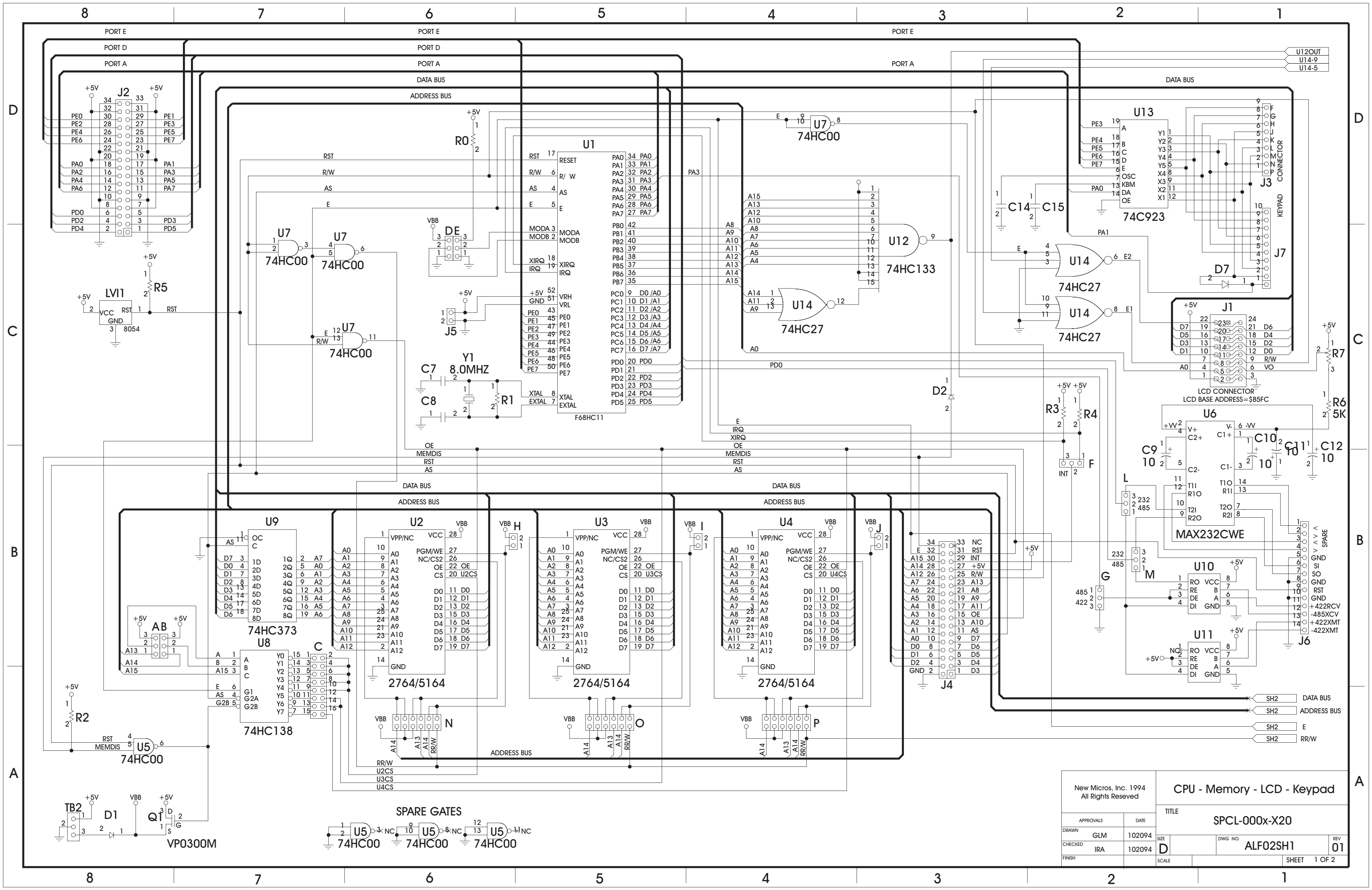
INTEL FORMAT DUMP COMMAND

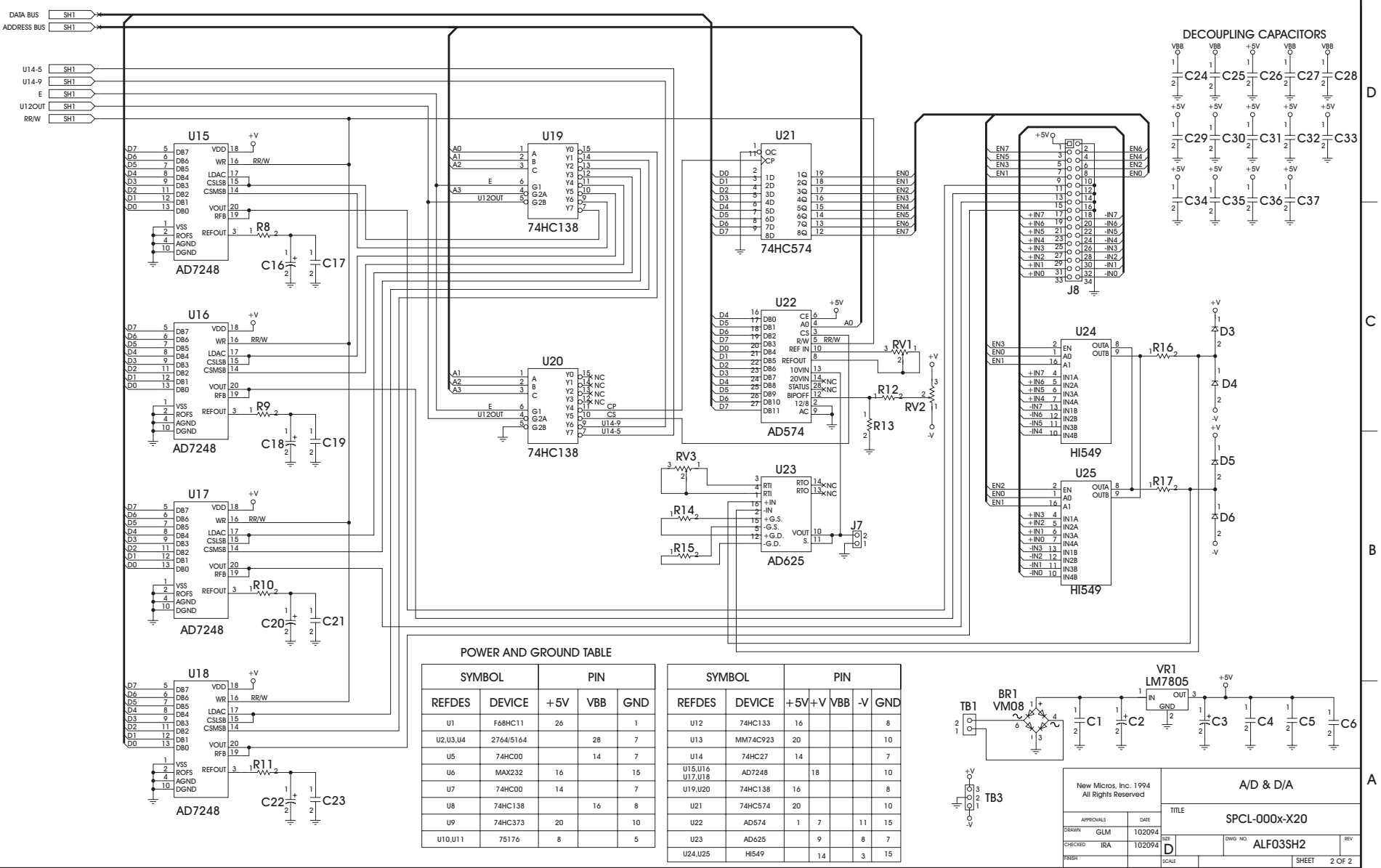
The following program allows a section of memory to be dumped out the serial channel in the Intel hex format which is a standard used by many of the commercially available PROM programmers. This program should allow the use of such programmers to capture programs and data in EPROMs, which are not supported for programming by the SPCL-000x-X20 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.


ALF04
LCD
LCD BASE
ADDRESS = B5FC
KEYBOARD
J1
J2
J3
J6
J8
J9
GND E7 E5 E3 E1 GND -7 -6 -5 -4 -3 -2 -1 -0
+5V E6 E4 E2 E0 O3 O2 O1 O0 +7 +6 +5 +4 +3 +2 +1 +0
TB1
TB2
TB3
BR1
VM08
LM7805
F68HC11
74HC00
74HC27
74HC133
74HC138
74HC373
74HC574
MM74C923
MAX232
75176
HI549
AD625
AD574/674
AD7248
U1
U2
U3
U4
U5
U6
U7
U8
U9
U10
U11
U12
U13
U14
U15
U16
U17
U18
U19
U20
U21
U22
U23
U24
U25
VR1
Y1
Q1
LVI1
J4
J5
J7
RV1
RV2
RV3
R0
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
R14
R15
R16
R17
C1
C2
C3
C4
C5
C6
C7
C8
C9
C10
C11
C12
C14
C15
C20
C21
C22
C23
C24
C25
C26
C27
C28
C29
C30
C31
C32
C33
C34
C35
C36
C37
D1
D2
D3
D4
D5
D6
D7
+5 GND VBB
+V GND -V
NMIX-0020 + A/D & D/A
NMIX0020 + A/D & D/A VER.01 102094


PORT E
PORT D
PORT A
DATA BUS
ADDRESS BUS
J2
U1
F68HC11
RESET
XTAL
EXTAL
MODA
MODB
XIRQ
IRQ
Y1
8.0MHZ
R0
R1
R2
R3
R4
R5
R6
5K
R7
C7
C8
C9
C10
C11
C12
C14
C15
LVI1
8054
U7
74HC00
U12
74HC133
U14
74HC27
U13
74C923
J3
CONNECTOR
KEYPAD
J7
D7
J1
LCD CONNECTOR
LCD BASE ADDRESS=$B5FC
U6
MAX232CWE
D2
INT
F
L
M
G
232
485
422
U10
U11
J6
SPARE
+422RCV
-485XCV
+422XMT
-422XMT
U9
74HC373
U8
74HC138
AB
C
U2
U3
U4
2764/5164
U2CS
U3CS
U4CS
RR/W
H
I
J
J4
N
O
P
SH2
U12OUT
U14-9
U14-5
U5
SPARE GATES
TB2
D1
Q1
VP0300M
New Micros, Inc. 1994
All Rights Reserved
CPU - Memory - LCD - Keypad
TITLE
SPCL-000x-X20
APPROVALS
DATE
DRAWN
GLM
102094
CHECKED
IRA
102094
FINISH
SIZE
D
DWG NO.
ALF02SH1
REV
01
SCALE
SHEET 1 OF 2



POWER AND GROUND TABLE

SYMBOL		PIN		
REFDES	DEVICE	+5V	VBB	GND
U1	F68HC11	26		1
U2,U3,U4	2764/5164		28	7
U5	74HC00		14	7
U6	MAX232	16		15
U7	74HC00	14		7
U8	74HC138		16	8
U9	74HC373	20		10
U10,U11	75176	8		5

SYMBOL		PIN				
REFDES	DEVICE	+5V	+V	VBB	-V	GND
U12	74HC133	16				8
U13	MM74C923	20				10
U14	74HC27	14				7
U15,U16 U17,U18	AD7248		18			10
U19,U20	74HC138	16				8
U21	74HC574	20				10
U22	AD574	1	7		11	15
U23	AD625		9		8	7
U24,U25	HI549		14		3	15

TITLE	A/D & D/A	
	SPCL-000x-X20	
APPROVALS	DATE	
DRAWN GLM	102094	
CHECKED IRA	102094	
FINISH	SIZE D	DWG NO. ALF03SH2 REV
	SCALE	SHEET 2 OF 2