



# MAX110 Evaluation System/Evaluation Kit

## General Description

The MAX110 evaluation system (EV system) is a complete, low-cost, two-channel data-acquisition system consisting of a MAX110 evaluation kit (EV kit) and a Maxim 80C32 microcontroller ( $\mu$ C) module. IBM PC compatible software provides a handy user interface to command the MAX110's features. Source code is provided in both C++ and C. Demonstration software includes rolling average filter and data logging applications. The MAX110 EV kit includes 1in<sup>2</sup> of prototyping area.

The MAX110 EV kit and EV system evaluate both the MAX110 and MAX111. To evaluate the MAX111, order a free sample of the MAX111 along with the MAX110 EV kit.

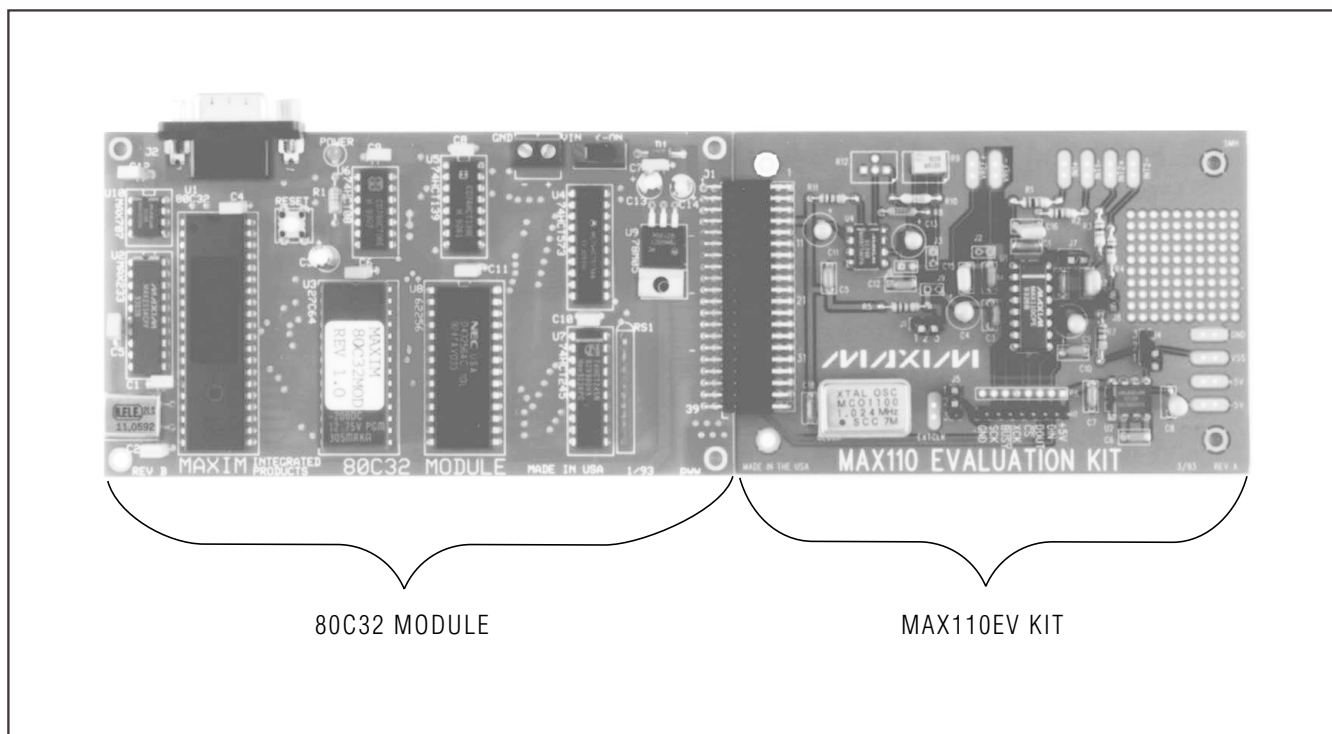
## Features

- ◆ Evaluates MAX110 and MAX111
- ◆ Complete Evaluation System
- ◆ Convenient Test Points Provided On-Board
- ◆ Data Logging Software
- ◆ Source Code Provided
- ◆ User-Selectable Resolution and Speed

## Ordering Information

PART	TEMP. RANGE	BOARD TYPE
MAX110EVC32-DIP	0°C to +70°C	Through-Hole
MAX110EVKIT-DIP	0°C to +70°C	Through-Hole
80C32MODULE-DIP	0°C to +70°C	Through-Hole

## EV System



Evaluates: MAX110/MAX111



# MAX110 Evaluation System/Evaluation Kit

## EV System Component List

QTY	DESCRIPTION
1	MAX110 evaluation kit (MAX110EVKIT-DIP)
1	80C32 $\mu$ C module (80C32MODULE-DIP)

## EV Kit Component List

DESIGNATION	QTY	DESCRIPTION
C1–C5, C7–10	9	0.1 $\mu$ F ceramic capacitors
C11–C15	5	10 $\mu$ F, 16V radial tantalum capacitors
C6–C16, C17, C18	4	1 $\mu$ F ceramic capacitors
R1–R4	4	1k $\Omega$ , 5% resistors
R5	1	10k $\Omega$ , 8-pin SIP resistor, pin 1 common
R6, R7, R8	3	100 $\Omega$ , 5% resistors
R9	1	200 $\Omega$ multi-turn potentiometer
R10	1	2.43k $\Omega$ , 1% resistor
R11	1	10k $\Omega$ , 1% resistor
R12	0	Leave this site empty
U1	1	MAX110CPE
U2	1	ICL7660CPA
U3	1	1.024MHz crystal oscillator module
U4	1	MAX873CPA
J1, J4, J5	3	3-pin headers
J7, J8	2	2-pin headers
J2, J3, J9	0	Leave these sites empty
J6	0	Unused reference designator
H1	1	10-pin header
None	1	Female data connector
None	5	Shunts
None	1	14-pin socket for U3
None	1	16-pin socket for U1
None	1	4.5" x 3" PC board
None	4	Rubber feet

## MAX110 EV System Quick Start

This section applies only to the use of the MAX110 EV kit operating with the Maxim 80C32 module.

- Copy the files from the distribution disk to your hard disk or to blank floppy disks. The MAX110 EV kit software should be in its own directory. The necessary files are in the root directory of the distribution disk, and the source code is in the SOURCE subdirectory. The SOURCE subdirectory is not required to operate the EV kit.
- Carefully connect the boards together by aligning the 40-pin header of the MAX110 EV kit with the 40-pin connector of the 80C32 module. Gently press them together. The two boards should be flush against one another.
- Connect a 9V to 15V DC power source to the 80C32 module. The terminal block is located next to the on/off switch, in the upper right corner of the 80C32 module. Observe the polarity marked on the board.
- Connect a cable from the computer's serial port to the 80C32 module. If using a 9-pin serial port, use a straight-through 9-pin female-to-male cable. If the only available serial port uses a 25-pin connector, a standard 25-pin to 9-pin adapter will be required.
- Start the MAX110 software on the IBM PC by setting the current directory to match the directory that contains the Maxim programs, and then type the program name, "MAX110". Do not turn off or disconnect the 80C32 module while the program is running; if you do, you will have to restart the program.
- The program will ask to which serial port the 80C32 module is connected. Press the space bar until the correct port is highlighted, and then press ENTER.
- The MAX110 program will be in terminal emulation mode. Turn on the power for the 80C32 module. The 80C32 module will display its logon banner and test its RAM.
- To download and run the RAM resident code on the 80C32 module, press ALT+L (i.e. hold down the ALT key as you strike the L key). The program prompts you for the file name. Press the ENTER key to download and run the file 110CODE.MAX on the 80C32 module.
- When the RAM resident program has been successfully downloaded, press ALT+C to switch to the Control Panel screen. A bank of software "switches" controls the MAX110. Two double-needle bar graphs display the MAX110's twin channels. The

# MAX110 Evaluation System/Evaluation Kit

bottom half of each bar graph shows a rolling average of the readings for that channel. The AUTO NULL switch is highlighted. With AUTO NULL and GAIN CAL both off, the bar graphs read the real voltage present at the input terminals of the MAX110.

- 10) Apply an input signal between the IN1+ and IN1- terminals of the MAX110 EV kit. Note that if jumper JU7 is installed, then IN1- is connected to ground. Observe the readout on the display screen.
- 11) To examine the bit patterns of the MAX110's data input and output, press "B" to switch to the Bit Manipulation Panel screen. The data input to the MAX110 is displayed in the upper left corner, and the output data is displayed in the lower left corner. A rolling average display of the data received is displayed underneath the most recently received data. Data is displayed in hexadecimal and in binary. To switch back to the Control Panel, press "C".
- 12) Before turning off power to the MAX110 EV kit, exit the program by pressing ALT+X.

## Stand-Alone MAX110 EV Kit Quick Start

This section applies only to the use of the MAX110 EV kit by itself, without the 80C32  $\mu$ C module.

- 1) Verify that shunts are installed at the proper locations. Table 1 shows the standard configuration. As shipped from the factory, U1 is a MAX110, and the voltage reference is 2.000V.
- 2) Connect a regulated +5V DC power supply to the terminals labeled +5V and GND. The GND pad is ground, and the +5V pad is the +5V input.
- 3) Use a voltmeter to verify that at least -4.75V appears at the -5V pad, and verify that the voltage reference between VREF+ and VREF- is 2.000V.
- 4) Connect the interface signals to the DIN, DOUT, CS, BUSY, and SCK test points. Use the GND test point as signal ground. See the MAX110 data sheet for timing information.
- 5) Apply the input voltage to the input terminals. One channel is between IN1+ and IN1-, and the other channel is between IN2+ and IN2-. If shunt J7 is installed, then IN1- is connected to GND. If shunt J8 is installed, then IN2- is connected to GND. Be sure to observe the absolute maximum ratings.

## Evaluating the MAX111

The MAX110 EV kit supports both the MAX110 and the MAX111 ICs. To evaluate the MAX111 use the following procedure:

- 1) **While the EV kit is turned off, move J4 to the 2-3 position so that V<sub>SS</sub> = ground.**
- 2) Replace U1 with a MAX111.
- 3) Note that the MAX111 uses a 1.25V reference. Replace resistor R11 with a 2.43k $\Omega$  resistor or remove jumper J3 and apply an external 1.25V reference at VREF.
- 4) Follow the quick-start instructions for the MAX110.

The MAX111 output codes are identical to the MAX110 output codes. The input voltage range is more restricted on the MAX111 because it is a single-supply device. Refer to the MAX110/MAX111 data sheet for more information on the MAX111's input voltage range and accuracy.

For the MAX111, tie the V<sub>SS</sub> supply to GND by installing the J4 shunt between pins 2 and 3. For the MAX110, tie the V<sub>SS</sub> supply to -5V by installing the J4 shunt between pins 1 and 2.

## Detailed Description of Hardware

### Jumper Options

Several jumper blocks allow different configurations of the MAX110. Jumper functions and default settings are shown in Tables 1 and 2.

### Using an External Clock

To drive the MAX110 with an external clock, use the following procedure:

- 1) Put a shunt across J5 pins 2-3, leaving pin 1 open.
- 2) Put a shunt across J1 pins 1-2, leaving pin 3 open.
- 3) Connect the external oscillator to the EXTCLK pad.
- 4) Connect the oscillator ground to the GND pin of header H1.

**Table 1. Default Jumper Settings**

JUMPER	DEFAULT SETTING
J1	1-2
J2	shorted by trace
J3	shorted by trace
J4	1-2
J5	1-2
J7	closed
J8	closed
J9	shorted by trace

# MAX110 Evaluation System/Evaluation Kit

Be sure to observe frequency and amplitude limits for the MAX110. The recommended frequency of operation is 512kHz for XCLK÷1 mode, 1024kHz for XCLK÷2 mode, or 2048kHz for XCLK÷4 mode. The conversion clock must have a constant, low-jitter frequency, otherwise the MAX110's linearity will suffer. Refer to the MAX110 data sheet for additional information.

## Using the Internal RC Oscillator

To use the MAX110's internal RC oscillator, remove the shunt from J5 and move the J1 shunt to pins 2-3. Make sure that the XCLK÷4 mode is selected in the MAX110 program. The XCLK÷4 mode must be used when the RC oscillator is enabled, since the RC oscillator frequency is approximately 2MHz. To switch back to the on-board crystal clock oscillator U3, move the J1 shunt to pins 1-2 and put a shunt across J5 pins 1-2.

Crystal clock oscillators can introduce noise into the system. For applications that demand reduced noise and do not require either precisely controlled conversion timing or 60Hz rejection, the MAX110's internal RC oscillator is recommended.

## Changing the Reference Voltage

The default reference voltage for the MAX110 EV kit is 2V. When a different reference voltage is selected, for proper scaling of the displayed output, specify the voltage in the command-line option when starting the program (see Table 3).

To use the 2.5V MAX873 reference directly, carefully cut the printed circuit trace at J3 between pins 2 and 3, and then install a 3-pin header. Next, select the 2.5V reference by installing a shunt at J3 between pins 1 and 2. Start the MAX110 program with the "-V2.5" command-line option, to tell the program that the reference is now 2.5V. Be sure to observe absolute maximum ratings for the device being evaluated. An optional 100kΩ trim pot may be added at site R12 to trim the MAX873 voltage, if desired.

To supply an external voltage reference, carefully cut the printed circuit trace at J3, between pins 2 and 3, and at J2. Connect the reference voltage between the VREF+ and VREF- pads. Run the MAX110 program, and use the "-V" command-line option to specify the reference voltage. Be sure to observe absolute maximum ratings for the device being evaluated.

## Activating Shutdown Mode

The MAX110 EV kit software can be used to measure the supply current of the MAX110 in shutdown mode. To evaluate shutdown mode, use the following procedure:

**Table 2. Jumper Settings on MAX110 EV Kit**

JUMPER	SETTING	FUNCTION
J1	1-2	XCLK is the clock input to the MAX110.
	2-3	XCLK is the output from the MAX110's internal RC oscillator.
	open	Do not operate the EV kit with J1 open.
J2	closed	VREF- connects to ground.
	open	VREF- does not connect to ground.
J3	2-3	VREF+ connects to the on-board 2V reference.
	1-2	VREF+ connects to the on-board 2.5V reference.
	open	VREF+ connects to a user-supplied reference.
J4	1-2	U1 = MAX110, V <sub>SS</sub> = -5V from ICL7660
	2-3	U1 = MAX111, V <sub>SS</sub> = ground
	open	V <sub>SS</sub> must be supplied by the user.
J5	1-2	XCLK is driven by the crystal oscillator U3 (note that when J5 is in 1-2 position, J1 must be in 1-2 position).
	2-3	XCLK connects to EXTCLK edge pad.
	open	XCLK is isolated.
J6	—	None
J7	closed	IN1- connects to ground.
	open	IN1- does not connect to ground.
J8	closed	IN2- connects to ground.
	open	IN2- does not connect to ground.
J9	closed	The V <sub>DD</sub> supply to the MAX110 flows through this jumper. Use this site to measure the supply current drawn by the MAX110.
	open	Do not operate the EV kit with J9 open.

# MAX110 Evaluation System/Evaluation Kit

- 1) To monitor the MAX110 supply current, install a current meter before applying power to the MAX110 EV kit. Refer to the *Measuring Supply Current* section. Typical supply current is less than 1mA during normal operation, and less than 4 $\mu$ A in shutdown mode if the internal RC oscillator is used.
- 2) Follow the quick-start instructions to start the MAX110 software on the IBM PC.
- 3) From the Control Panel screen, press "S" to enter the Shutdown Power Cycling screen.
- 4) From the Shutdown Power Cycling screen, press "D" to shut down the MAX110. The supply current should drop to less than 4 $\mu$ A.
- 5) From the Shutdown Power Cycling screen, press "U" to power-up the MAX110. The supply current should settle at less than 1mA. There will be short current spikes whenever a command is sent to the MAX110.

A typical application may sample the MAX110 at regular intervals, keeping the MAX110 shut down between samples. This type of usage can be evaluated by using the "P" command in the Shutdown Power Cycling screen. The power-cycling loop repeatedly puts the MAX110 into sleep mode for a user-specified length of time, wakes up the MAX110, takes a reading, and powers the MAX110 down again. The "P" command runs continuously, until halted by either the "U" or "D" command.

The parameters that pertain to the "P" command are as follows: the state of the PD and PDX bits before and after shutdown, the sleep time, and the optional NOOP, AUTO NULL, and GAIN CAL cycles.

The sampling rate is determined by setting the sleep time, which is the length of time the program keeps the MAX110 in shutdown mode. The program allows a maximum sleep time of 65 seconds.

If the internal RC oscillator is used, activating the PDX bit may cause the BUSY line to remain low. This condition should be cleared by starting the power-up sequence with a configuration word whose MSB is 0 (a NOOP cycle). If an external oscillator is used, the PDX bit has no effect and the NOOP cycle is not required. Supply current will not be as low as in the internal RC oscillator mode, unless the external oscillator is halted during shutdown.

The PDX power-down bit has no effect unless the internal RC oscillator is being used. Normally, the PD and PDX bits should be set to 1 in shutdown mode, and both should be 0 in active mode.

## Measuring Supply Current

Jumper J9 can be used to measure the supply current drawn by the MAX110 IC. To measure the supply current, use the following procedure:

- 1) Exit the MAX110 software and then turn off the MAX110 EV kit power.
- 2) Carefully cut the printed circuit board trace at location J9.
- 3) Connect a current meter between the two pins of J9. The direction of current flow is marked with an arrow on the board.
- 4) Turn on the MAX110 EV kit and restart the MAX110 program to evaluate the desired operating mode.
- 5) Observe the supply current in both operating and shutdown modes.

To restore the MAX110 EV kit, use the following procedure:

- 1) Turn off the MAX110 EV kit power.
- 2) Disconnect the current meter.
- 3) Install a 2-pin header and jumper at J9, or use a small piece of wire to reconnect J9.

If the MAX110  $V_{SS}$  is connected to -5V, the -5V supply current can be measured in a similar way. Remove the shunt from J4 1-2 and connect a current meter. Current will flow from J4 pin 2 to J4 pin 1. Or, the current meter may be connected between the  $V_{SS}$  edge pad and the -5V edge pad. Current will flow from the  $V_{SS}$  edge pad to the -5V edge pad.

## Detailed Description of Software

When starting the MAX110 program from the DOS prompt, several command-line options are available. For a list of available options, run MAX110 with the "?" command-line option. Refer to Table 3.

When the MAX110 program begins operation, it is in its Opening screen. Use the space bar to select the serial communications port to which you have connected the 80C32 module. Press the ENTER key to advance to the Terminal screen.

The MAX110 program displays its Terminal screen when it is establishing communications with the 80C32 module. When power is applied to the 80C32 module or reset is pressed, banner message identifying the 80C32 module is displayed. After the module completes its self check, it says that all tests have passed. At this point, use the ALT+L command to load the RAM resident program into the 80C32 module. After the RAM resident program has been loaded, use the ALT+C command to advance to the Control Panel screen.

# MAX110 Evaluation System/Evaluation Kit

**Table 3. Command-Line Options**

OPTION	FUNCTION
?	Display program version and list of command-line options.
-V	Specify the reference voltage ("V2.5" means 2.5 volts).
-A	Specify the size of the rolling average queue ("-A10" means average the last 10 readings).
+O	Allow the bar graph to display overrange values correctly.
-O	Do not compensate the bar-graph display for overrange values. Values beyond $\pm V_{REF}$ will display incorrectly due to code aliasing.
1	Default to COM1.
2	Default to COM2.
3	Default to COM3.
4	Default to COM4.
-Lmyfile	Specify log file "myfile" to store control-panel readings.
CONTINUE	Assume that the RAM resident program is already loaded.

The MAX110 Control Panel screen displays the current and average values for the MAX110's two input channels, and offers convenient control of the configuration settings. Pressing TAB selects a switch, and SPACE BAR manipulates the switch. The "1" and "2" keys enable and disable polling for inputs 1 and 2, respectively. The input values may be displayed as raw ADC counts or as real voltages. Data logging can be enabled if the program is started with the log file command-line option. Table 6 lists the commands available in the Control Panel screen.

The MAX110 Shutdown Power Cycling screen allows evaluation of the shutdown mode. The "D" command puts the MAX110 into its shutdown state, the "U" command brings the MAX110 out of shutdown, and the "P" command makes the program alternate between the shutdown and active states. Refer to the section *Activating Shutdown Mode*.

**Table 4. Opening Screen Commands**

KEY	FUNCTION
SPACE BAR	Select next serial port
ENTER	Use the selected serial port.
ALT+X	Exit the program.

**Table 5. Terminal Screen Commands**

KEY	FUNCTION
ALT+P	Return to Opening screen to select a different COM port.
ALT+L	Begin downloading code to 80C32.
ALT+C	Start the Control Panel screen.
ALT+B	Display the baud-rate menu.
ALT+1	Use 1200 baud.
ALT+2	Use 2400 baud.
ALT+4	Use 4800 baud.
ALT+9	Use 9600 baud.
ALT+R	Send the cold-restart command to 80C32.
ALT+X	Exit the program.

**Table 6. Control Panel Commands**

KEY	FUNCTION
TAB	Highlight the next switch.
SHIFT+TAB	Highlight the previous switch.
SPACE BAR	Toggle the highlighted switch.
1	Enable or disable polling of input 1.
2	Enable or disable polling of input 2.
C	Display readings as raw ADC counts.
V	Display readings as real voltages.
L	Enable or disable data logging. This command is available only if the MAX110 program was started with the -L command-line option.
S	Enter the Shutdown Power Cycling screen.
B	Enter the Bit Manipulation Panel screen.
ALT+X	Exit the program.

# MAX110 Evaluation System/Evaluation Kit

The MAX110 Bit Manipulation Panel screen offers direct control of the configuration word and gives a direct view of the output data word. Note that certain bits of the control word are used for factory testing and must be zero; the program prevents these bits from accidentally being set. See Table 8 for a list of commands.

## Data Logging

The MAX110 program can store measurements in a file. Once logging has been enabled, it can be started and stopped from the Control Panel screen. The numbers stored in the logging file are signed conversion counts, which may be converted to voltage values by a simple calculation.

To enable datalogging, start up the MAX110 program using the command-line option “-L” to specify the output filename. For example, the command “MAX110 -Lmyfile.dat” creates the file myfile.dat for logging. If you specify the name of a file that already exists, the program asks for confirmation before erasing the old file. An “N” answer exits MAX110 without damaging the

old file. A “Y” answer erases the old file. Once the old file has been destroyed, it cannot be retrieved.

By default, both channels are polled. To log data from only one channel, press the “1” or “2” key to turn off the unused channel. To begin logging data, press the “L” key. Logging may be suspended and restarted by pressing “L” again.

The format of the log file is straight ASCII text, with one reading per channel per line. When only one channel is enabled, each line contains one reading. When both channels are enabled, the readings for channel 1 and channel 2 are separated by a comma. The channel 1 reading is logged first. If both channels are disabled, no additional data is logged (see Table 9).

The numerical value written to the file is the signed integer conversion-count read from the MAX110. To convert the conversion count to a voltage, use the following formula:

$$\text{Real Voltage} = \frac{\text{Conversion Count} \times V_{\text{REF}}}{16384}$$

Note that due to overrange, conversion counts may extend beyond 16384 to approximately 21000.

**Table 7. Shutdown Power Cycling Panel Commands**

KEY	FUNCTION
D	Power-down the MAX110 using settings 1-2.
U	Power-up the MAX110 using settings 4-8.
P	Power-cycle the MAX110 using settings 1-8.
1	Toggle the value of the PD bit used in shutdown mode.
2	Toggle the value of the PDX bit used in shutdown mode.
3	Set the sleep time (the length of time the P command will remain in shutdown mode).
4	Toggle whether a NO OP cycle should be performed during wake-up.
5	Toggle the value of the PD bit used in active mode.
6	Toggle the value of the PDX bit used in active mode.
7	Toggle whether an AUTO NULL cycle should be performed during wake-up.
8	Toggle whether a GAIN CAL cycle should be performed during wake-up.
L	Enable or disable data logging. This command is available only if the MAX110 program was started with the -L command-line option.
C	Return to the Control Panel.
ALT+X	Exit the program.

**Table 8. Bit Manipulation Panel Commands**

KEY	FUNCTION
TAB	Highlight the next bit.
SHIFT+TAB	Highlight the previous bit.
SPACE BAR	Toggle the highlighted bit.
1	Enable or disable polling of input 1.
2	Enable or disable polling of input 2.
C	Enter the Control Panel.
ALT+X	Exit the program.

**Table 9. Description of Log-File Format**

1. <log file> ::=	{ <record> <return character> }
2. <record> ::=	<channel 1 reading>
	<channel 1 reading> <comma> <channel 2 reading>
3. <channel 1 reading> ::=	<signed integer>
4. <channel 2 reading> ::=	<signed integer>

# MAX110 Evaluation System/Evaluation Kit

**Table 10. Data-Connector Interface**

PIN No.	80C32	MAX110	FUNCTION
1–4	GND	GND	Ground
7, 8	+5V	+5V	+5V Supply to MAX110 EV Kit
27	P1.0	$\overline{\text{CS}}$	Active-Low Chip Select to MAX110
28	P1.1	SCK	Rising-Edge Clock Input to MAX110
29	P1.2	DIN	Serial Data Input to MAX110
30	P1.3	$\overline{\text{DOUT}}$	Serial Data Output from MAX110
31	P1.4	$\overline{\text{BUSY}}$	Active-Low Busy Output from MAX110

### Source Code

Complete source code for both MAX110.EXE and 110CODE.MAX is provided on disk. MAX110.EXE was written using Borland C++ version 3.0, and 110CODE.MAX was written using the Avocet 8051 Macro Assembler.

The most relevant subroutine in 110CODE.ASM is Config110 (see Listing 1). This subroutine writes the configuration word while simultaneously reading the data from the MAX110 EV kit. Note that the configuration word does not take effect until the next read/write operation. Macros have been used to help make the subroutine easier to understand.

Figures 1 and 2 show the timing specifications. The 80C32 module uses an 11.0592MHz clock, so the instruction cycle time is 1.085 $\mu$ s per instruction cycle. SCK must be low before  $\overline{\text{CS}}$  is activated. SCK's pulse width is 4.35 $\mu$ s, with period 17.33 $\mu$ s. The most significant bit is sent first. Configuration data is valid 1.14 $\mu$ s before the rising edge of SCK. Data from the MAX110 is sampled while SCK is high.

The communication protocol used between MAX110.EXE and 110CODE is very simple. MAX110.EXE sets up a

configuration word by sending "Cxxxx", where xxxx represents the hexadecimal configuration word value. MAX110.EXE then sends the "R" command, which writes and reads the MAX110, and prints the hexadecimal value it reads from the EV kit.

Note that the RAM resident code 110CODE.MAX resets the MAX110 EV kit by sending three configuration bytes in order: 8C8Ch, 8C88h, and 8C80h. These configuration words reset the MAX110 by turning on auto-null and gain-calibration mode together, then turning off auto null, and finally turning off gain calibration. The MAX110 internally divides the 1.024MHz crystal oscillator clock by two, for an effective clock of 512kHz. The program uses the 16-bit resolution mode, unless otherwise instructed by the user.

Source files SERCMD.C and MAX110.C form the core interface to the MAX110 EV kit. To write your own programs using these files, read example programs SIMPLE.C and NOISE.CPP, and the header files SERCMD.H and MAX110.H. Both SERCMD and MAX110 may be compiled under C or C++.



# MAX110 Evaluation System/Evaluation Kit

Listing 1. Sample Code to Read and Write to the MAX110

```

0020          count:      seg      dataseg
0021          CtrlH:      ds        1          ; bit index counter
0022          CtrlL:      ds        1          ; configuration word to send
0023          ByteInH:     ds        1          ; to the MAX110
0024          ByteInL:     ds        1          ; 16-bit word read back from
0025          TimeOut1:    ds        1          ; the MAX110
0026          TimeOut2:    ds        1          ; watchdog timer -- if BUSY stays
0027          TimeOut3:    ds        1          ; low for too long, then something
                                ; is wrong with the evaluation kit

;***** Config110
;*
;* Configure the MAX110 and read its value. This code is used with the MAX110 Evaluation Kit.
;*
;* On Entry:
;*   globals CtrlH and CtrlL contain the configuration word to be written to the MAX110.
;*
;* On Success:
;*   The carry flag is clear, the configuration word has been written to the MAX110,
;*   and the 16 bit word read back from the MAX110 has been stored in
;*   globals ByteInH and ByteInL.
;*
;* On Failure:
;*   The carry flag is set. This indicates that BUSY is stuck low.
;*

Config110:    seg      codeseg
              proc
              setb   BUSY          ; use BUSY as an input (BUSY is P1.4)
              setb   DOUT         ; use DOUT as an input (DOUT is P1.3)
              jnb   CtrlH.7,L?DontWaitForBusy
                                ; wait until not busy...
              mov   TimeOut3,#00Ch ; delay about 3 seconds
              mov   TimeOut2,#0FFh
              L3:   mov   TimeOut1,#0FFh ; set time-out counter
              L2:   mov   TimeOut1,#0FFh ; is it set yet?
              L1:   jb    BUSY, ??0000
              djnz  TimeOut1, L1
              djnz  TimeOut2, L2
              djnz  TimeOut3, L3
              sjmp  L?Timeout      ; give up
              ??0000:
              L?DontWaitForBusy:
              clr   SCK            ; start with clock low (SCK is P1.1)
              clr   CS            ; assert chip select (CS is P1.0)
              mov   count,#8      ; get first 8 bits
                                ; Send/Recieve High Byte
              L?Hloop:
              mov   c,CtrlH.7     ; send bit 7 of ctrlH
              mov   DIN,c         ; (DIN to MAX110 is P1.2)
              setb  SCK           ; CLOCK RISING EDGE
              mov   c,DOUT        ; read bit 7 of byteinH
              mov   ByteInH.7,c
              clr   SCK           ; clock goes low
              mov   a,CtrlH       ; shift the buffers left
              rl   a
              mov   CtrlH,a
              mov   a,ByteInH
              rl   a
              mov   ByteInH,a
              djnz  count,L?Hloop ; get next 8 bits
              mov   count,#8      ; Send/Recieve Low Byte
              L?Lloop:
              mov   c,CtrlL.7     ; send bit 7 of ctrlL
              mov   DIN,c
              setb  SCK           ; CLOCK RISING EDGE
              mov   c,DOUT        ; read bit 7 of byteinL
              mov   ByteInL.7,c
              clr   SCK           ; clock goes low
              mov   a,CtrlL       ; shift the buffers left
              rl   a
              mov   CtrlL,a
              mov   a,ByteInL
              rl   a
              mov   ByteInL,a
              djnz  count,L?Lloop
              setb  CS            ; negate chip select
              clr   c            ; clear carry flag
              ret                ; return from subroutine
              L?Timeout:
              setb  CS            ; negate chip select
              setb  c            ; set carry flag
              ret                ; return from subroutine
              endproc ;* Config110

```

Evaluates: MAX110/MAX111

# MAX110 Evaluation System/Evaluation Kit

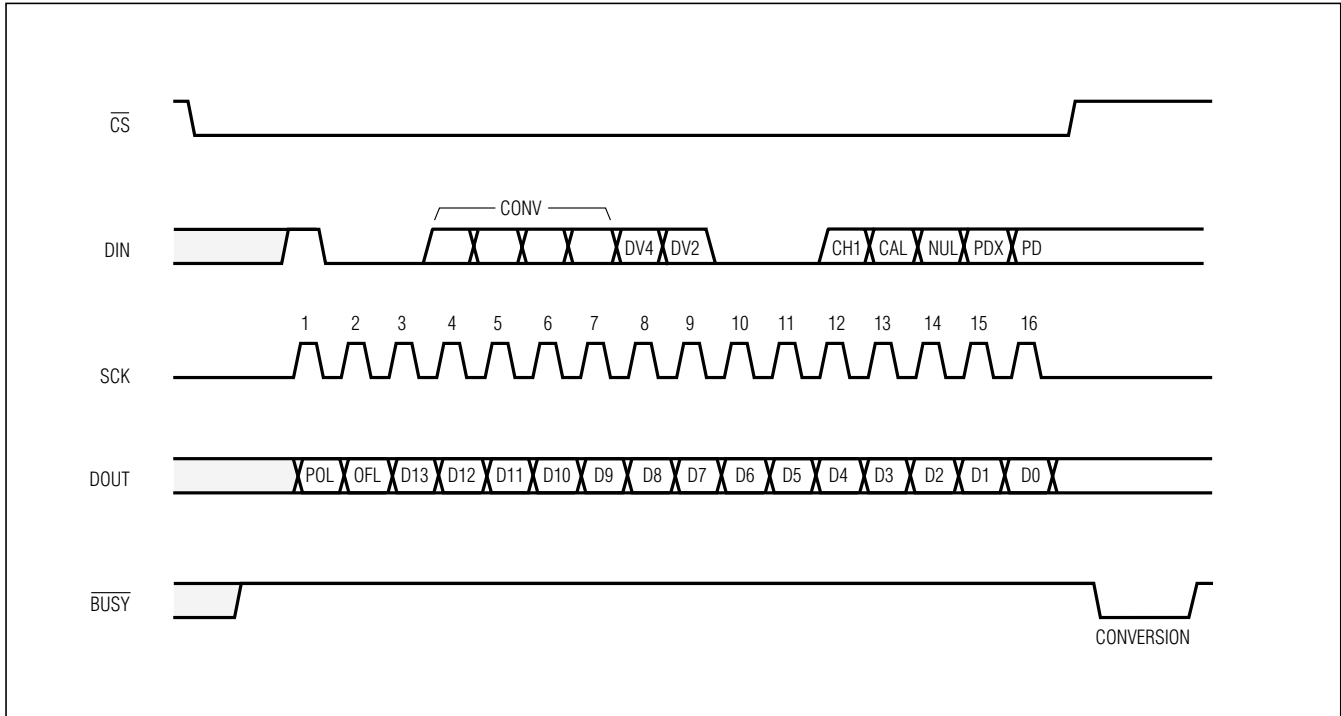


Figure 1. MAX110 EV Kit Timing Diagram

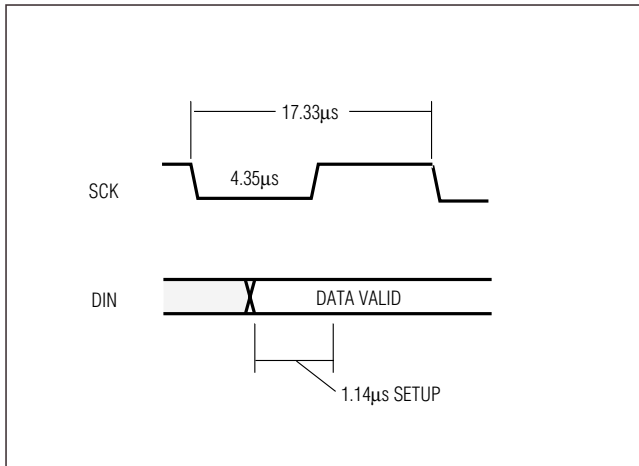


Figure 2. MAX110 EV Kit Detailed Timing Diagram

# MAX110 Evaluation System/Evaluation Kit

Evaluates: MAX110/MAX111

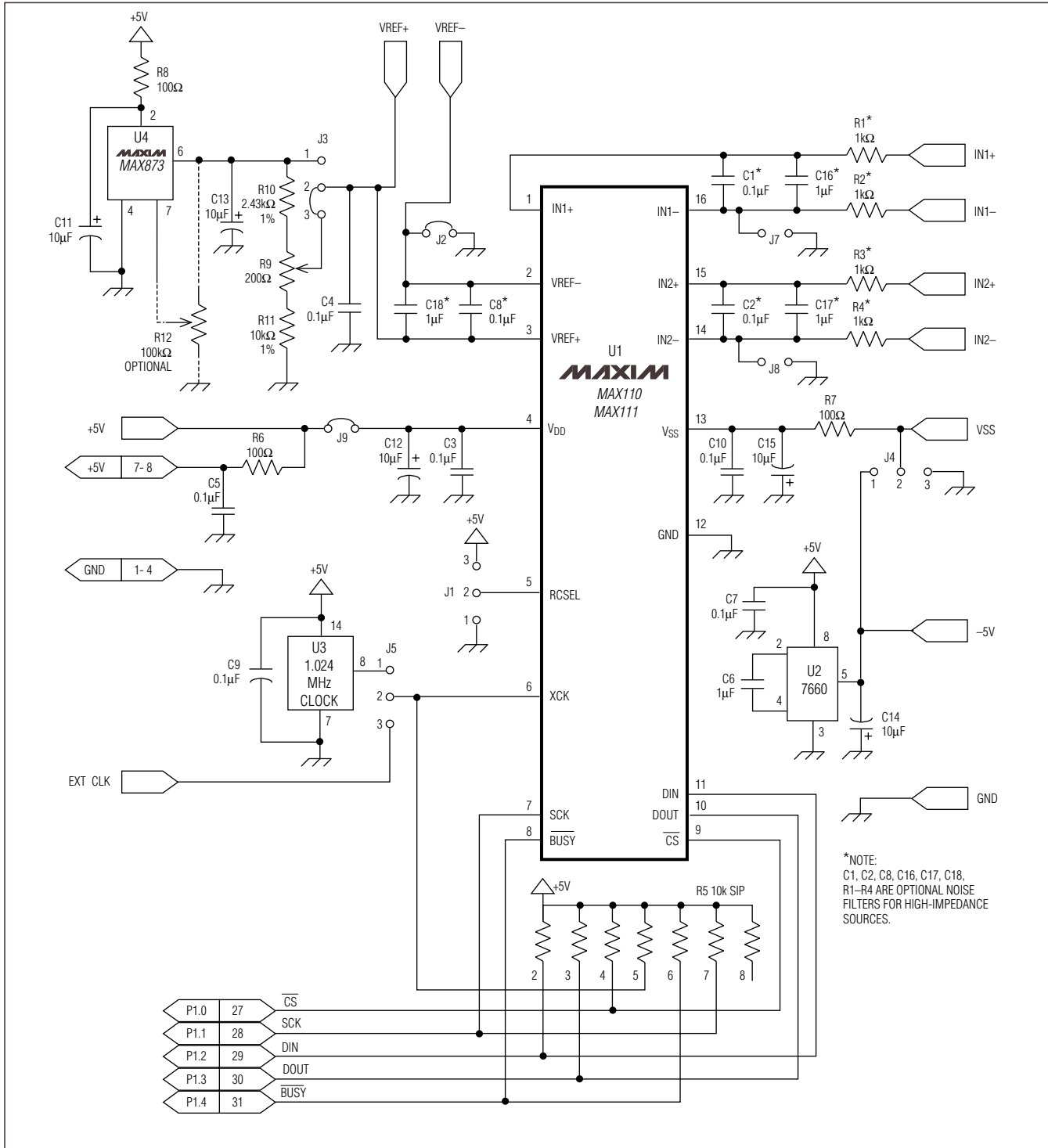


Figure 3. MAX110 EV Kit Schematic Diagram

# MAX110 Evaluation System/Evaluation Kit

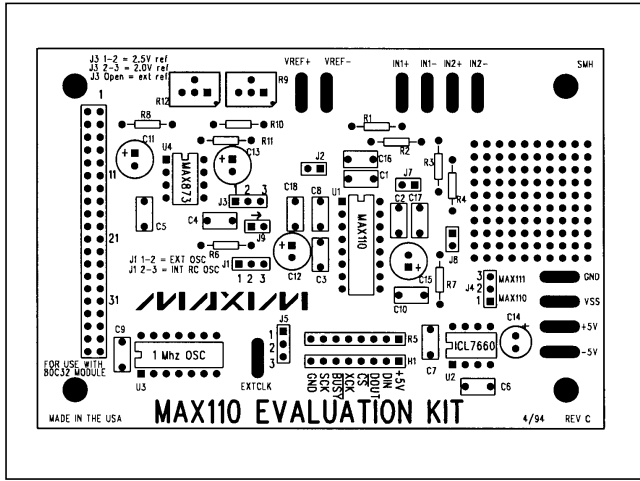


Figure 4. MAX110 EV Kit Component Placement Guide—component side

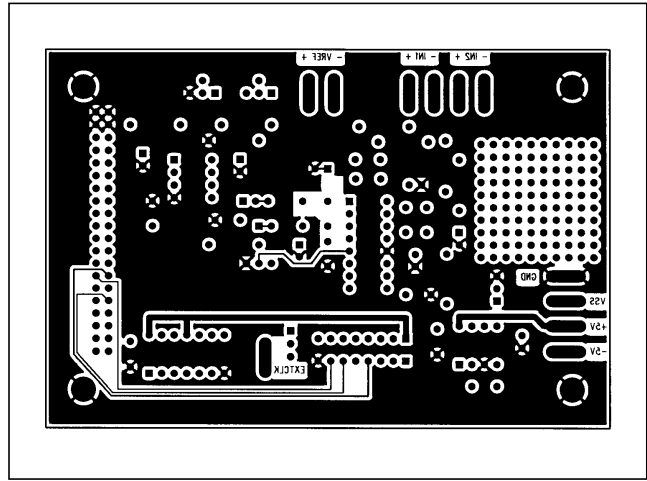


Figure 6. MAX110 EV Kit PC Board Layout—Solder Side

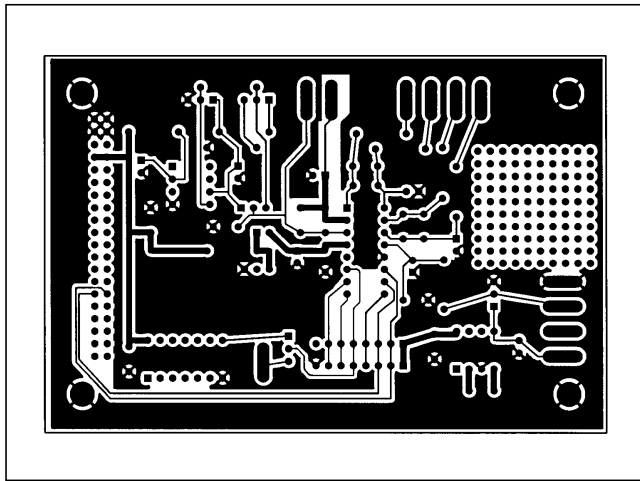


Figure 5. MAX110 EV Kit PC Board Layout—Component Side

# MAX110 Evaluation System/Evaluation Kit

**Evaluates: MAX110/MAX111**

## 80C32 Module Component List

DESIGNATION	QTY	DESCRIPTION
C1, C2	2	15pF ceramic capacitors
C4, C5, C6, C7, C8, C9, C10, C11, C12	9	0.1μF, 50V ceramic capacitors
C3, C13, C14	3	22μF, 16V radial electrolytic capacitors
D1	1	1N4001 diode
J1	1	40-pin right-angle male connector
J2	1	DB9 right-angle socket
R1	1	620Ω resistor
RS1	1	10kΩ, 10-pin, 9-resistor SIP
SW1	1	Power switch
SW2	1	Reset switch
IC1	1	80C32
IC2	1	MAX233CPP
IC3	1	27C64
IC4	1	74HCT573
IC5	1	74HCT139
IC6	1	74HCT08
IC7	1	74HCT245
IC8	1	62256
IC9	1	78M05
IC10	1	MAX707CPA
Y1	1	11.059MHz crystal
None	1	2-pin power connector
None	1	28-pin 600-mil socket for IC3 (the EPROM)
None	4	Rubber feet
None	1	3.00" x 5.50" PC board

## 80C32 Module General Description

The Maxim 80C32 microcontroller (μC) module is intended for use with this and other Maxim evaluation kits (EV kits). It contains the 80C32 μC, RS-232 interface, 8kbytes of EPROM, 32kbytes of static RAM, and address decoding logic. A 40-pin connector mates with a connector found on Maxim EV kits designed to interface with the 80C32 module.

The module is connected to an IBM-compatible personal computer over a serial communications port. Software provided with each EV kit runs on the computer and controls the unit consisting of the 80C32 module and EV kit. The program uses a routine stored in the 27C64 EPROM to download special 80C32 code for each kit. The downloaded code controls the EV kit and, together with the program running on the personal computer, displays the output data.

The board operates from a single 8V to 22V supply. Both the pre-regulated and regulated +5V levels are available to the EV kit through the 40-pin connector.

### 80C32 Module Power Supply

The Maxim 80C32 module requires an input of 8V to 22V for normal operation. An on-board 78M05 power regulator supplies the 5V required for the logic on the module, and any 5V requirements for the EV kit attached to the 40-pin connector. The pre-regulated voltage is also available on the data connector. The source must be capable of supplying 100mA for the module and meeting the load requirements of the EV kit.

### Microprocessor Supervisor

A MAX707 on the module monitors the 5V logic supply, generates the power-on reset, and produces a reset pulse whenever the reset button is pressed. A watchdog function was not included because they frequently interfere while debugging programs, and debugging is a prime function of this board.

### 80C32 Microcontroller

The 80C32 is a member of the popular Intel 8051 family of μCs. It is a low-power CMOS version that requires external ROM for program storage, 256 bytes of internal RAM, and four 8-bit I/O ports. Three of the ports are required by the system for serial communications and memory control. The fourth port (P1) is available through the data connector.

The 80C32 communicates with the PC over a serial RS-232 link. A MAX233 acts as a level shifter between the ±15V RS-232 signals and the TTL levels of the 80C32. The MAX233 also generates the output voltages necessary to drive RS-232 lines.

Port 0 (pins 32–39) of the 80C32 multiplexes the lower eight bits of memory address and the eight bits of read/write data. The lower eight bits of address data are latched during each I/O cycle by the 74HCT573 octal latch. The latch is controlled by the address latch enable (ALE) signal of the 80C32. Port 2 (pins 21–28) of the 80C32 supplies the upper eight bits of address information.

# MAX110 Evaluation System/Evaluation Kit

The port 3 pins (10–17) provide several unrelated functions. Pins 10 and 11 are used as the receive data (RxD) and transmit data (TxD) pins of the RS-232 link. Pins 16 and 17 act as the write (WR) and read (RD) control signals for the data I/O cycles. Four other pins are configured as interrupt and timer controls, but are not used on this board.

## Memory

The board has a 27C64 EPROM containing code for initializing the 80C32 and downloading additional program code to the 62256 RAM. After a reset, the EPROM resident code initializes the 80C32, determines the address range of the RAM, sets the RS-232 baud rate to 1200, and waits for communications from the PC. Receiving any character will prompt the program to send an initial banner that includes the program name, revision level, and boundaries of the on-board RAM.

The 62256 CMOS (32kbyte) static RAM is used to hold program code for the various Maxim EV kits that use the 80C32 module as the controller. Programs are transferred from disk to the RAM using software running on a personal computer, such as MAXLOAD or other programs provided with Maxim EV kits. Programs written to execute from this RAM start at 4000 (HEX) and are typically less than 4kbytes long. The remaining RAM is available for data storage.

## Address Ranges

Logic on the module board generates various enable signals for different address ranges. The ROM and RAM enable signals are fed directly to the respective chips. Several additional signals (CS0–CS3) are available on the data connector to be used by Maxim EV kits. Table 11 outlines the address range for each of the elements found on the 80C32 module.

**Table 11. Address Ranges in Hexadecimal**

ADDRESS RANGE (HEX)	ENABLE SIGNAL
0000 → 3FFF	ROM
4000 → BFFF	RAM
C000 → CFFF	CS0
D000 → DFFF	CS1
E000 → EFFF	CS2
F000 → FFFF	CS3

## Data I/O Connector

A 40-pin connector mounted on the edge of the printed circuit board provides connection between the  $\mu$ C module and other Maxim EV kits. Both power and digital signals are transferred via the connector. To join the module board with an EV kit, carefully align and insert the pins on the connector with the mating 40-pin female connector of the kit. The pin functions are listed in Table 12.

**Table 12. I/O Connector Pin Functions**

PIN	FUNCTION	DESCRIPTION
1–4	Ground	
5, 6	Pre-regulator input	
7, 8	Regulated +5V	
9	RD	Read strobe
10	WR	Write strobe
11	CS0	Address C000–CFFF
12	CS1	Address D000–DFFF
13	CS2	Address E000–EFFF
14	CS3	Address F000–FFFF
15–18	ADDR0–ADDR3	Lowest 4 bits of address
19–26	DB0–DB7	8-bit data bus
27–34	P1.0–P1.7	8 bits of port 1
35–40	Reserved	

## Software Architecture

Software for EV kits using the Maxim 80C32 module is divided into three elements: the interface program running on an IBM-compatible PC, a module program located in EPROM, and a program supplied on disk that is transferred to the RAM located on the module.

## EPROM Resident Program

The EPROM resident program initializes the 80C32, establishes communications over the RS-232 link, verifies the static RAM, and downloads other programs. Its operation starts on power-up and whenever the reset button is pressed. After reset, the program waits indefinitely to receive a character over the RS-232 port. When the first character is received, a logon banner identifying the module and firmware revision is transmitted.

# MAX110 Evaluation System/Evaluation Kit

Evaluates: MAX110/MAX111

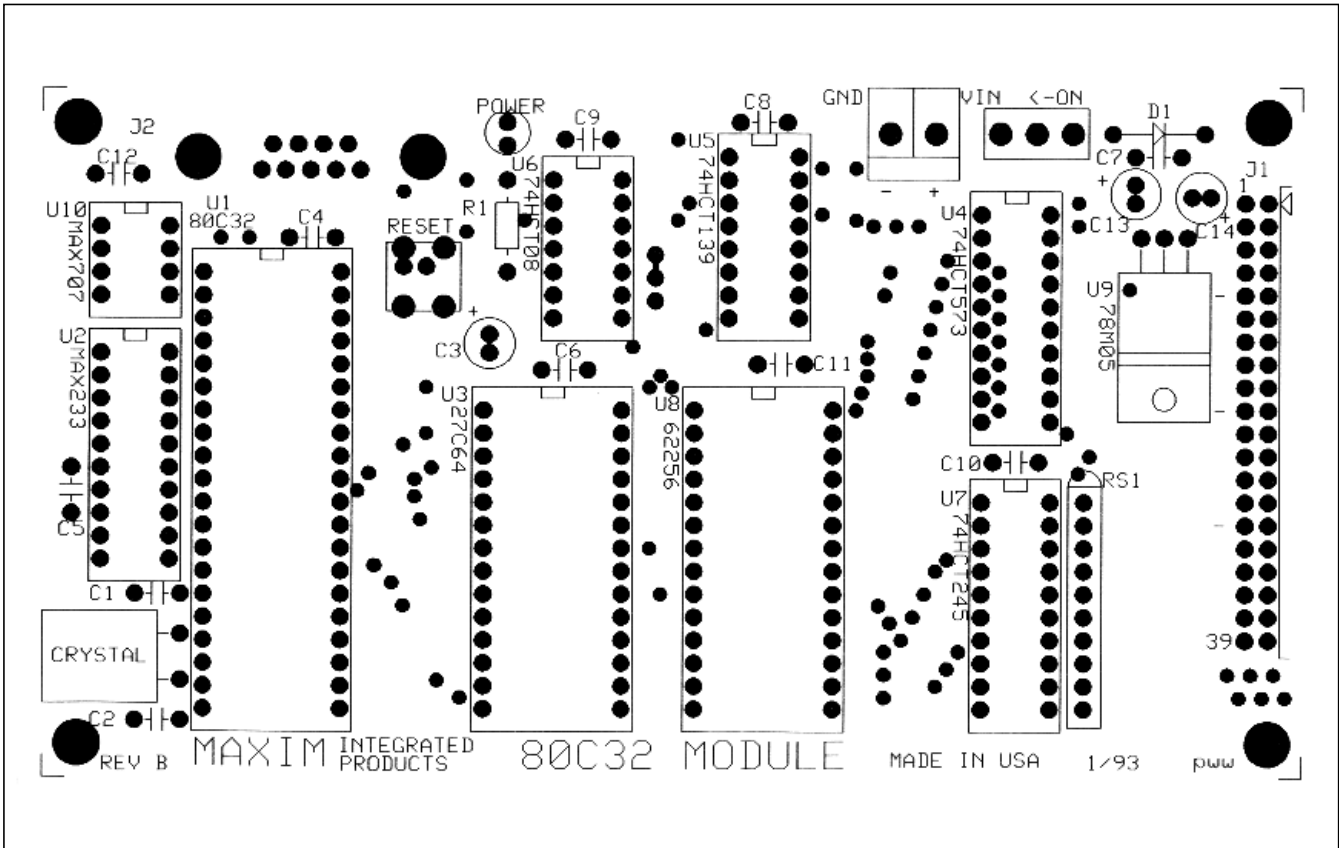


Figure 7. 80C32 Module Component Placement Guide (2x)

Immediately following transmission of the logon banner, the program runs a checker routine for the on-board 256kbit static RAM. The RAM is filled with several patterns and then read to verify that each pattern has been retained. A pass or fail indication is displayed on the personal computer after each pass. EV kit software requires proper operation of the RAM. Do not attempt to use the board if any of the RAM checks fail.

Two other programs for the EV kits are provided on a floppy disk shipped with each kit. One program acts as the user interface and transmits commands to the 80C32 module. The other is an 80C32 application program that executes from the RAM located on the module. The procedure for loading the programs varies with each kit, so follow the instructions provided.

# MAX110 Evaluation System/Evaluation Kit

**Evaluates: MAX110/MAX111**

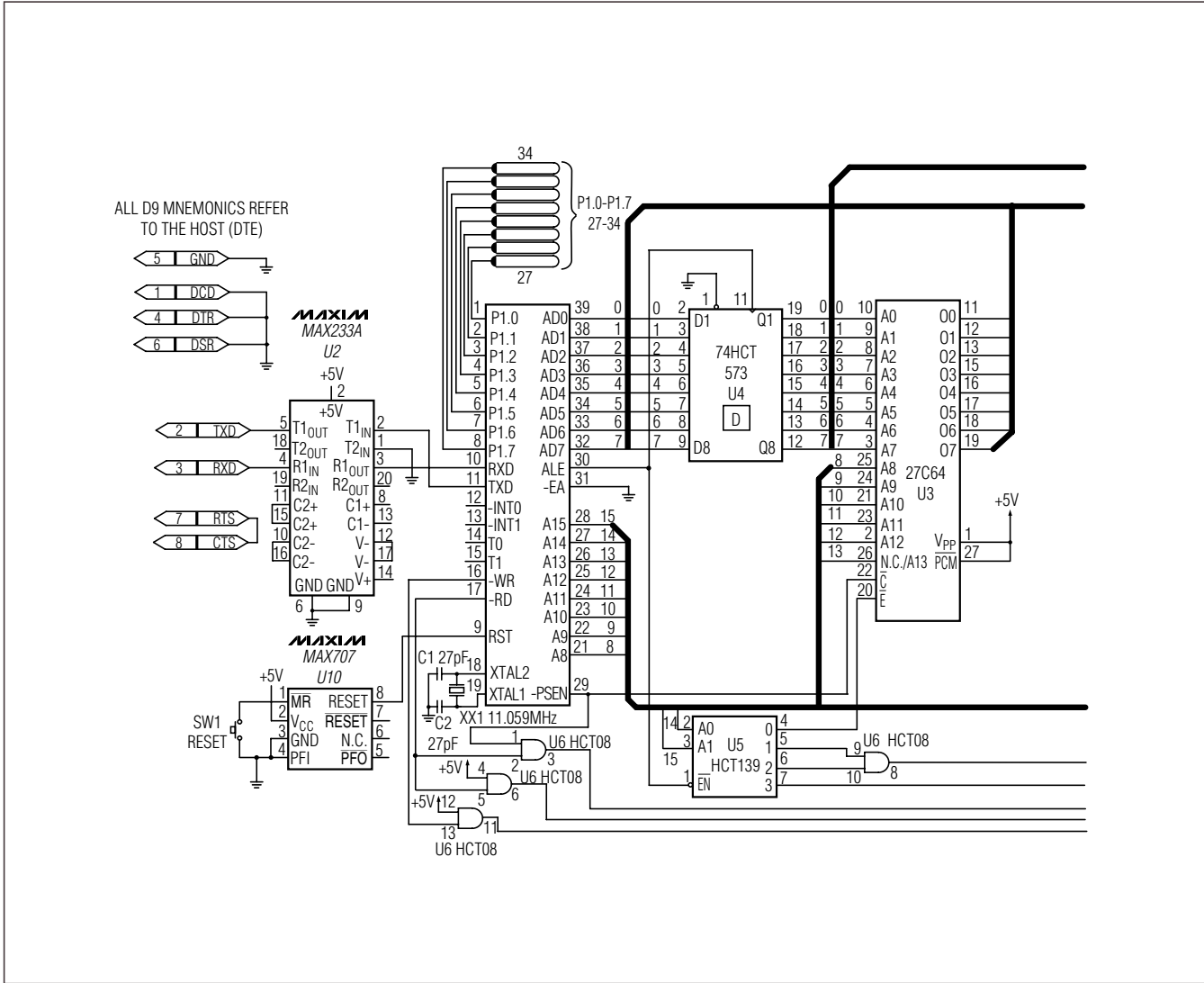


Figure 8. 80C32 Module Schematic



# MAX110 Evaluation System/Evaluation Kit

Evaluates: MAX110/MAX111

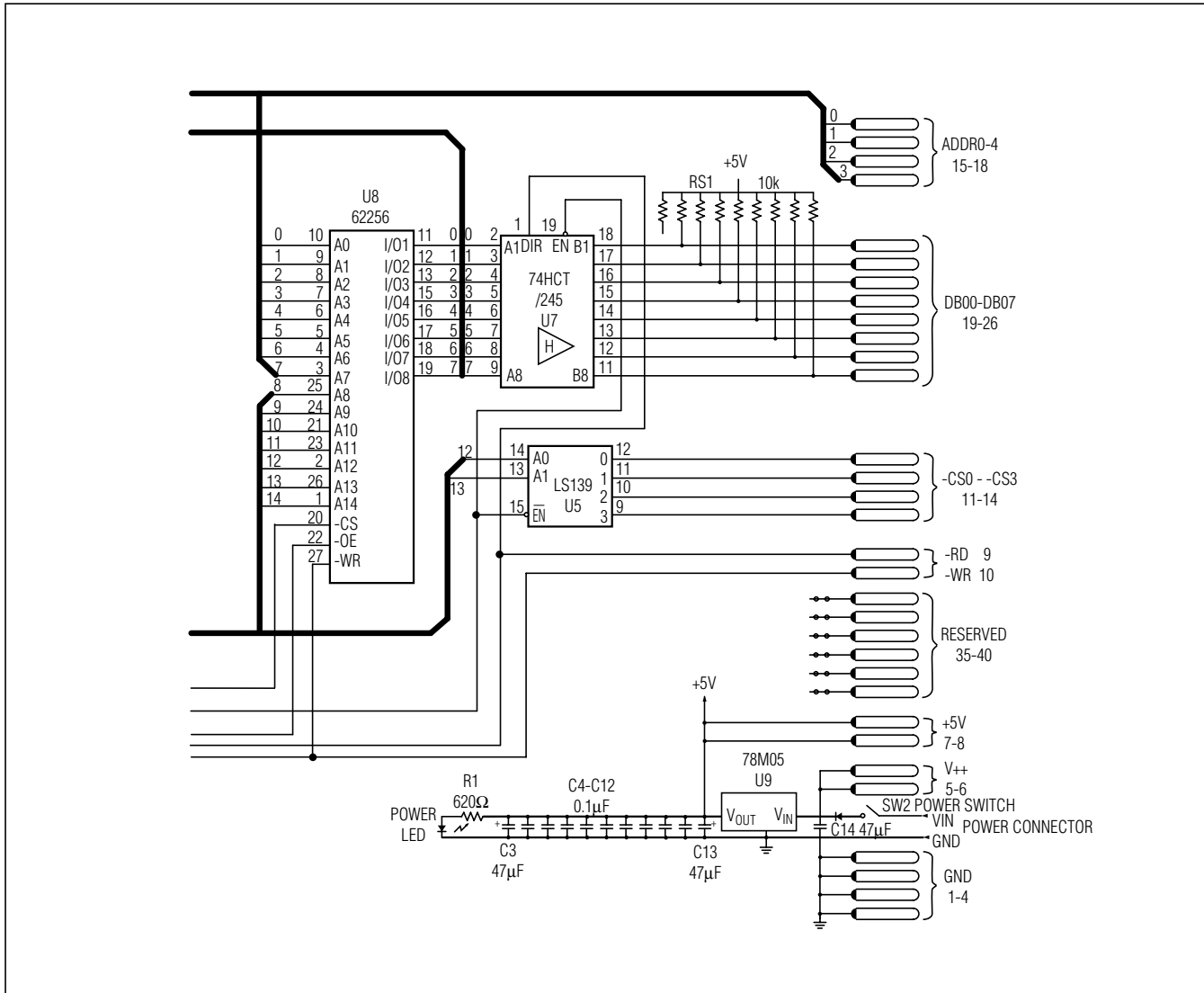


Figure 8. 80C32 Module Schematic (continued)

# MAX110 Evaluation System/Evaluation Kit

**Evaluates: MAX110/MAX111**

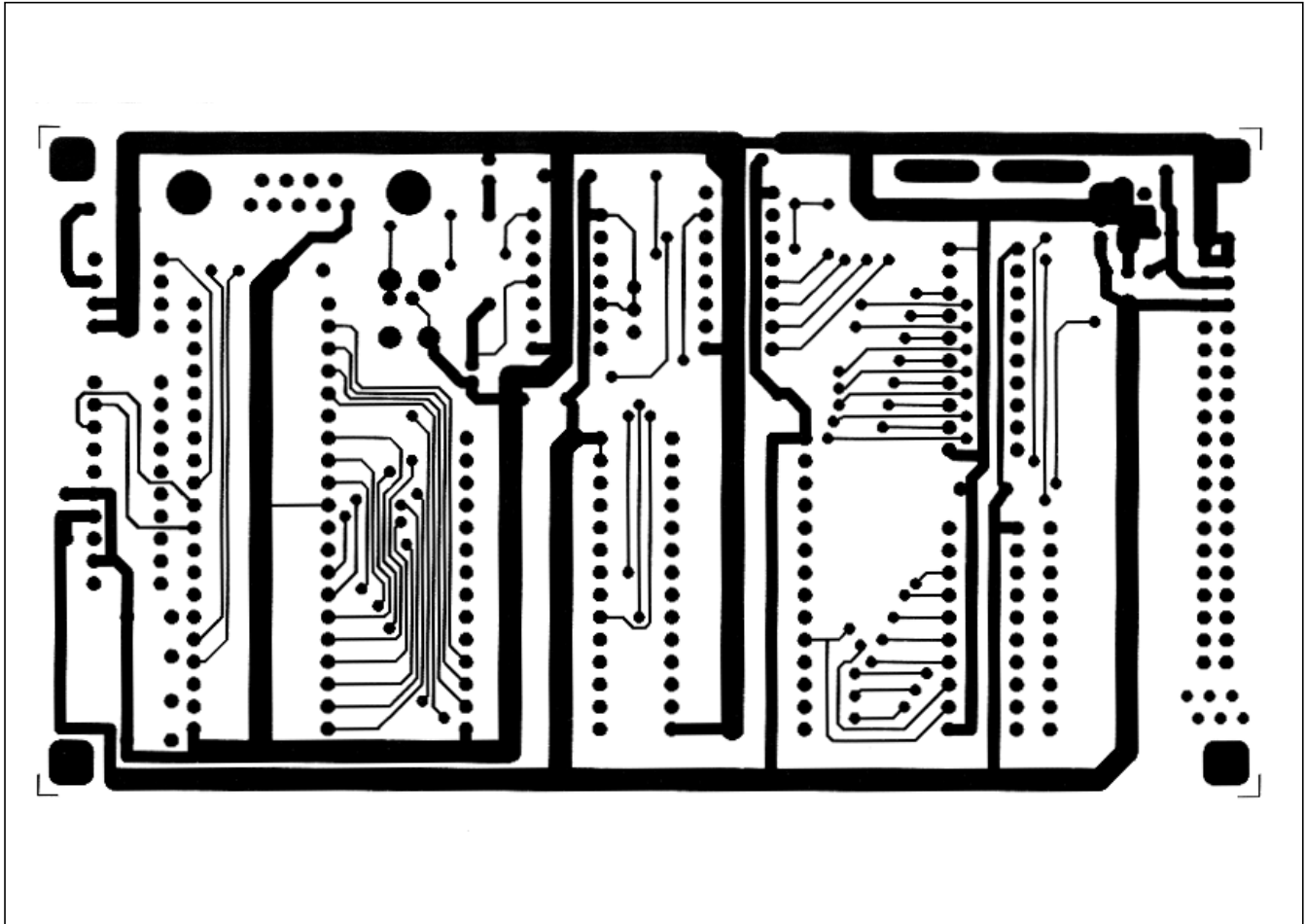


Figure 9. 80C32 Module Component-Side Layout (2x)

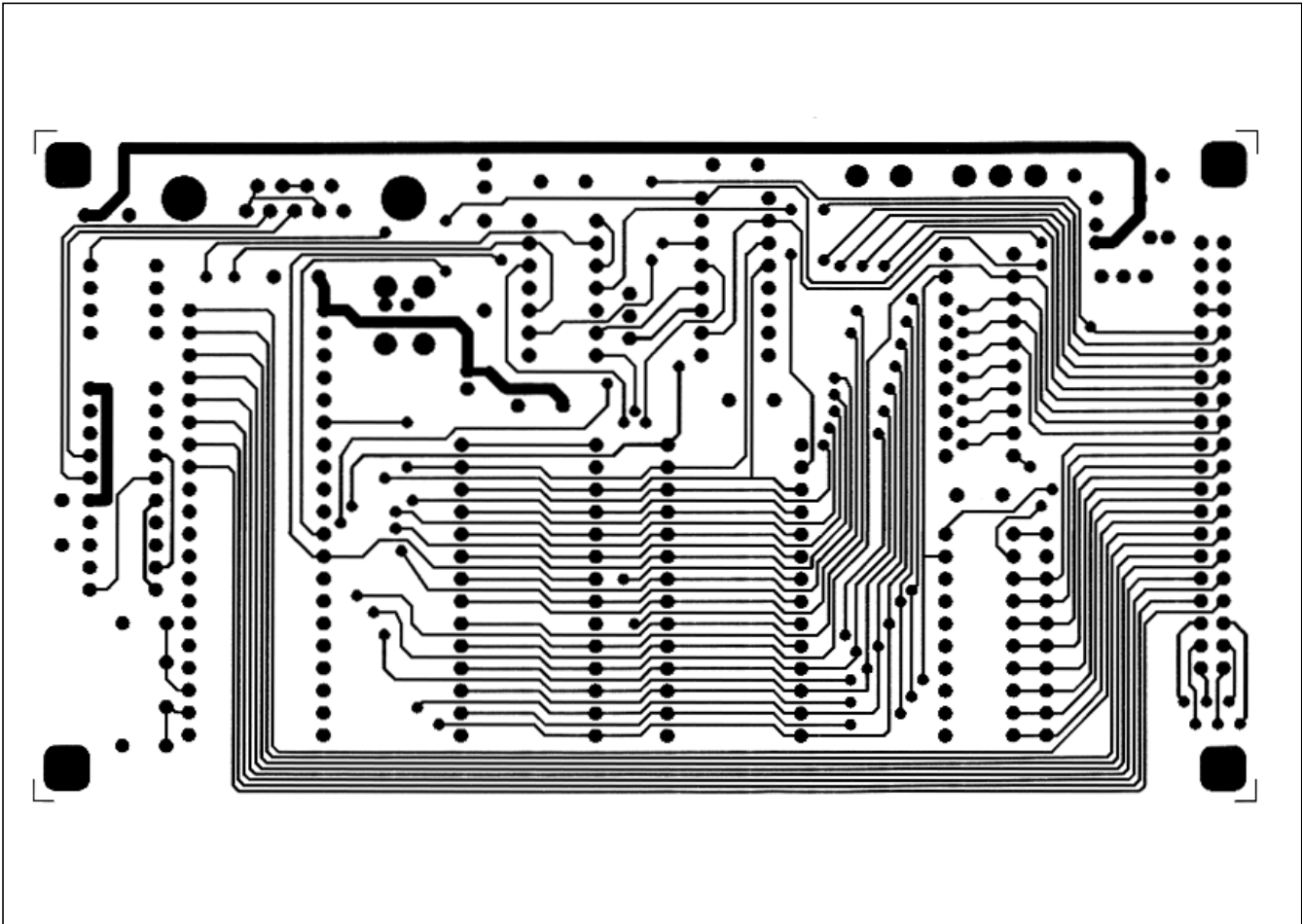


Figure 10. 80C32 Module Solder-Side Layout (2x)

Maxim cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim product. No circuit patent licenses are implied. Maxim reserves the right to change the circuitry and specifications without notice at any time.

**Maxim Integrated Products, 120 San Gabriel Drive, Sunnyvale, CA 94086 408-737-7600** \_\_\_\_\_ 19