



MAX1231 Evaluation Kit/ MAX1231 Evaluation System

General Description

The MAX1231 evaluation system (EV system) consists of a MAX1231 evaluation kit (EV kit) and a Maxim 68HC16MODULE-DIP microcontroller (μ C) module. The MAX1231 is a 12-bit, 300ksps analog-to-digital converter (ADC) with an internal temperature sensor, first-in/first-out (FIFO), and an internal reference. The evaluation software runs under Windows[®] 95/98/2000, providing a handy user interface to exercise the MAX1231 features.

Order the complete EV system (MAX1231EVC16) for comprehensive evaluation of the MAX1231 using a personal computer. Order the EV kit (MAX1231EVKIT) if the 68HC16MODULE module has already been purchased with a Maxim EV system, or for custom use in other μ C-based systems.

This system can also evaluate the MAX1030, MAX1031, and MAX1230. Contact the factory for free samples of these products. See the *Detailed Description of Hardware* section.

MAX1231 Stand-Alone EV Kit

The MAX1231 EV kit provides a proven PC board layout to facilitate evaluation of the MAX1231. It must be interfaced to appropriate timing signals for proper operation. Connect 3V and ground return to terminal block TB1 (Figure 5). Refer to the MAX1227/MAX1229/MAX1231 data sheet for timing requirements.

MAX1231 EV System

The MAX1231 EV system operates from a user-supplied 7VDC to 20VDC power supply. The evaluation software runs under Windows 95/98/2000 on an IBM PC, interfacing to the EV system board through the computer's serial communications port. See the *Quick Start* section for setup and operating instructions.

Features

- ◆ Proven PC Board Layout
- ◆ Complete Evaluation System
- ◆ Convenient On-Board Test Points
- ◆ Data-Logging Software
- ◆ Fully Assembled and Tested

Ordering Information

PART	TEMP RANGE	INTERFACE TYPE
MAX1231EVKIT	0°C to +70°C	User supplied
MAX1231EVC16	0°C to +70°C	Windows software

Note: The MAX1231 evaluation software is designed for use with the complete evaluation system MAX1231EVC16 (includes the 68HC16MODULE-DIP module together with the MAX1231EVKIT). If the MAX1231 evaluation software is not to be used, the MAX1231EVKIT can be purchased by itself, without the μ C.

MAX1231EVC16 Component List

PART	QTY	DESCRIPTION
MAX1231EVKIT	1	MAX1231 evaluation kit
68HC16MODULE-DIP	1	68HC16 μ C module

Component Suppliers

SUPPLIER	PHONE	FAX	WEBSITE
TDK	847-803-6100	847-390-4405	www.component.tdk.com

Note: Please indicate you are using the MAX1231 when contacting these component suppliers.

Windows is a registered trademark of Microsoft Corp.



Maxim Integrated Products 1

For pricing, delivery, and ordering information, please contact Maxim/Dallas Direct! at 1-888-629-4642, or visit Maxim's website at www.maxim-ic.com.

Evaluates: MAX1030/MAX1031/MAX1230/MAX1231

MAX1231 Evaluation Kit/ MAX1231 Evaluation System

MAX1231EVKIT Component List

DESIGNATION	QTY	DESCRIPTION
C1-C16	16	0.01 μ F \pm 10% ceramic capacitors (1206)
C17	1	0.1 μ F ceramic capacitor (1206)
C18	1	10 μ F \pm 20%, 6.3V X5R ceramic capacitor (0805) TDK C2012X5R0J106M
H1	1	7-pin header
J1	1	2 x 20 right-angle socket
JU1 2-3	1	Shunt
JU1	1	3-pin header
JU2	0	Shorted trace on PC board
R1-R16	16	10 Ω \pm 5% resistors (1206)
TB1	1	2-circuit terminal block, 5.08mm pitch
TB2-TB5	4	5-circuit terminal blocks, 5.08mm pitch
U1	1	MAX1231BCEG-T (24-pin QSOP)
U2	1	MAX1615EUK-T, ABZD, (5-pin SOT23)
U3, U4	2	MAX1840EUB (10-pin μ MAX) or MAX1841EUB
None	1	MAX1231 evaluation kit PC board

Quick Start

Required Equipment

Before you begin, you need the following equipment:

- MAX1231EVC16 (contains the MAX1231EVKIT and the 68HC16MODULE-DIP)
- DC power supply, +7VDC to +20VDC at 0.25A
- Windows 95/98/2000 computer with a spare serial (COM) port
- 9-pin I/O extension cable

Procedure

Do not turn on the power until all connections are made:

- 1) Ensure that JU1 is in the 2-3 position (Table 2).
- 2) Carefully connect the boards by aligning the 40-pin header of the MAX1231 EV kit with the 40-pin connector of the 68HC16MODULE-DIP module. Gently press them together. The two boards should be flush against one another.

Note: Bold items are software settings or commands.

- 3) Connect a +7VDC to +20VDC power source to the μ C module at the terminal block located next to the on/off switch, along the top edge of the μ C module. Observe the polarity marked on the board.
- 4) Connect a cable from the computer's serial port to the μ C 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 is required. The EV kit software checks the modem status lines (CTS, DSR, DCD) to confirm that the correct port has been selected.
- 5) Install the evaluation software on your computer by running the INSTALL.EXE program on the floppy disk. The program files are copied and icons are created for them in the Windows start menu.
- 6) Turn on the power supply.
- 7) Start the MAX1231 program by opening its icon in the start menu.
- 8) The program prompts you to connect the μ C module and turn its power on. Slide SW1 to the ON position. Select the correct serial port, and click OK. The program automatically downloads its software to the module.
- 9) Apply an input signal to AIN0 and click **Perform Action**. Observe the readout on the screen. To perform the action repeatedly, check the **every 200ms** box.
- 10) To view a graph of the measurements, pull down the **View** menu and click **Graph**.

Detailed Software Description

The evaluation software's main window configures the data converter and measures the analog inputs. Under **Action**, select a scanning sequence, repetitive conversions, or a single conversion. Each selected channel's measurement results are displayed in the corresponding **Measurement Results** field.

The **Action** setting **read single channel repeatedly** requires making a selection under **Repetition** to determine how many times the selected channel should be measured.

Use **Averaging** to summarize a series of measurement results on each selected channel as an arithmetic mean value. Repetition can be used together with **Averaging** to summarize a large number of measurement results as a small number of sample means.

MAX1231 Evaluation Kit/ MAX1231 Evaluation System

The **Low-Level Interface Details** panel shows the most recent low-level register write. A summary of what has been written to each register is available under the **Low-level registers** tab.

The AIN14 and AIN15 channels are automatically skipped if their alternate functions are selected in the setup register. The evaluation software updates its display to show or hide these channels whenever the alternate functions are enabled or disabled.

The **Setup** tab configures the alternate functions for the AIN14 and AIN15 pins, and also configures adjacent channels as differential input pairs.

The **Low-level registers** tab summarizes the commands that create the active configuration. The **Reset All Registers** button resets these software-shadowed register values and sends the reset command to the MAX1231.

Table 1. Graph Tool Buttons

TOOL	FUNCTION
	Show the entire available input range.
	Expand the graph data to fill the window.
	Move the view left or right.
	Move the view up or down.
	Expand or contract the x-axis.
	Expand or contract the y-axis.
	Load data from a file.
	Save data to a file.
	Option to write a header line when saving data.
	Option to write line numbers when saving data.
	View code vs. time plot.
	View histogram plot (cumulative frequency of each code).
	View table.
<input type="text" value="Min"/>	Show minimum in tabular view.
<input type="text" value="Max"/>	Show maximum in tabular view.
<input type="text" value="Span"/>	Show span in tabular view. Span = maximum - minimum
<input type="text" value="N"/>	Show number of samples in tabular view.
<input type="text" value="Sum(x)"/>	Show sum of the samples in tabular view.
<input type="text" value="Sum(x*x)"/>	Show sum of the squares of the samples in tabular view.
<input type="text" value="Mean"/>	Show arithmetic mean in tabular view: $\text{Mean} = \frac{\sum(x)}{n}$

TOOL	FUNCTION
<input type="text" value="StdDev"/>	Show standard deviation in tabular view: $\text{Standard deviation} = \sqrt{\frac{n\sum(x^2) - (\sum x)^2}{(n-1)n}}$
<input type="text" value="Rms"/>	Show root of the mean of the squares (RMS) in tabular view: $\text{RMS} = \sqrt{\frac{\sum(x^2)}{n}}$
<input type="checkbox" value="0"/>	Channel 0 enable
<input type="checkbox" value="1"/>	Channel 1 enable
<input type="checkbox" value="2"/>	Channel 2 enable
<input type="checkbox" value="3"/>	Channel 3 enable
<input type="checkbox" value="4"/>	Channel 4 enable
<input type="checkbox" value="5"/>	Channel 5 enable
<input type="checkbox" value="6"/>	Channel 6 enable
<input type="checkbox" value="7"/>	Channel 7 enable
<input type="checkbox" value="8"/>	Channel 8 enable
<input type="checkbox" value="9"/>	Channel 9 enable
<input type="checkbox" value="10"/>	Channel 10 enable
<input type="checkbox" value="11"/>	Channel 11 enable
<input type="checkbox" value="12"/>	Channel 12 enable
<input type="checkbox" value="13"/>	Channel 13 enable
<input type="checkbox" value="14"/>	Channel 14 enable

Evaluates: MAX1030/MAX1031/MAX1230/MAX1231

MAX1231 Evaluation Kit/ MAX1231 Evaluation System

Sampling

Measurement data can be sampled in external clock mode. From the **Setup** tab, set **Clock Mode** to **0111xxxx ext clock**. Then, return to the **Measurement** tab and click **Get Samples**.

Graph Window

To view recently measured data, drop down the **View** menu and choose **Graph**. Data can be viewed as a time sequence plot, a histogram plot, or as a table of raw numbers. See Table 1 for available graph commands.

Diagnostics Window

The diagnostics window is used for factory testing prior to shipping the evaluation kit. It is not meant for customer use.

Detailed Hardware Description

The MAX1231 device under test (U1) is a 12-bit, 300ksps ADC with an internal temperature sensor, FIFO, and an internal reference. Resistors R1–R16 and capacitors C1–C16 form single-pole, lowpass, anti-aliasing filters for each input. Capacitor C17 provides power-supply bypassing for U1 (Figure 5). Refer to the MAX1227/MAX1229/MAX1231 data sheet.

The EV kit includes a MAX1615 3V/5V linear regulator (U2) and a set of MAX1840/MAX1841 level shifters (U3 and U4) to support using the 3V MAX1230 with the 5V μ C.

Evaluating the MAX1030

The MAX1030 is the 5V, 10-bit version of the MAX1231. To evaluate the MAX1030, request a free sample of MAX1030BCEG. Replace U1 with a MAX1030, and move JU1 shunt to the 5V position. In the software's **Options** menu, select **Reference** = 4.096V.

The evaluation software expects 12 bits of data, but the MAX1030 only provides 10 bits of meaningful data. Because the most significant bits are aligned, the measurement code numbers reported by the software are four times the actual measurement code number. Reconstructed voltage values are unaffected. Adjust the graph window by selecting its **Options** menu; set its **Sub-LSBs** to 2.

Evaluating the MAX1031

The MAX1031 is the 10-bit version of the MAX1231. To evaluate the MAX1031, request a free sample of the MAX1031BCEG. Replace U1 with a MAX1031, and move the JU1 shunt to the 3V position.

The evaluation software expects 12 bits of data, but the MAX1031 only provides 10 bits of meaningful data. Because the most significant bits are aligned, the measurement code numbers reported by the software are

four times the actual measurement code number. Reconstructed voltage values are unaffected. Adjust the graph window by selecting its **Options** menu; set its **Sub-LSBs** to 2.

Evaluating the MAX1230

The MAX1230 is the 5V version of the MAX1231. To evaluate the MAX1230, request a free sample of the MAX1230BCEG. Replace U1 with a MAX1230, and move the shunt to the 5V position. In the software's **Options** menu, select **Reference** = 4.096V.

Using an External Reference

An external reference can be applied in single-ended or differential mode. In differential mode, analog input pin AIN14 is used as the REF(-) return connection. In single-ended mode, the reference returns to common ground.

With the power off, connect the external reference. Then power up the system and run the evaluation software. Under **Setup**, select the appropriate reference input configuration. If using a single-ended reference, set **Reference Input** to **01xx01xx pin15 = AIN14**. If using a differential reference, set **Reference Input** to **01xx11xx pin 15 = REF-**.

Table 2. Jumper JU1 (V_{DD} Voltage Selection)

SHUNT POSITION	V _{DD} (V)	FUNCTION
1–2	5V	Normal operation with U1 = MAX1230 or MAX1030.
2–3*	3V	Normal operation with U1 = MAX1231 or MAX1031.
Open	Unspecified	Do not operate kit with JU1 open.

*Indicates default configuration.

Table 3. Optional Jumper JU2 (AIN15 Alternate Function)

JU2 STATE	U1 PIN 16 CONNECTION	FUNCTION
Closed* (default trace on PC board)	Connected to μ C module J1 pin 29 (through level shifter)	U1 pin 16 = CNVST conversion start command. Leave TB5 pin 4 unconnected.
Open	Connected to TB5 pin 4	U1 pin 16 = AIN15 analog input. Connect signal source to terminal block TB5 pin 4.

*Indicates default configuration.

MAX1231 Evaluation Kit/ MAX1231 Evaluation System

Troubleshooting

Problem: No output measurement. System seems to report zero voltage or fails to make a measurement.

Solution: Check V_{DD} supply voltage. Check the reference voltage using a digital voltmeter. Use an oscilloscope to verify that the conversion-start signal is being strobed.

Problem: Measurements are erratic, unstable, or have poor accuracy.

Solution: Check the reference voltage using a digital voltmeter. Use an oscilloscope to check for noise. When probing for noise, keep the oscilloscope ground return lead as short as possible, preferably less than 0.5in (10mm).

Problem: Unacceptable errors occur when measuring a transducer.

Solution: Although most signal sources can be connected directly to the MAX1231's analog inputs, some high-impedance signal sources can require an input buffer. Check for settling errors by increasing the acquisition time (internal clock mode 01). If necessary, use a MAX4430 to buffer high-impedance signal sources.

Evaluates: MAX1030/MAX1031/MAX1230/MAX1231

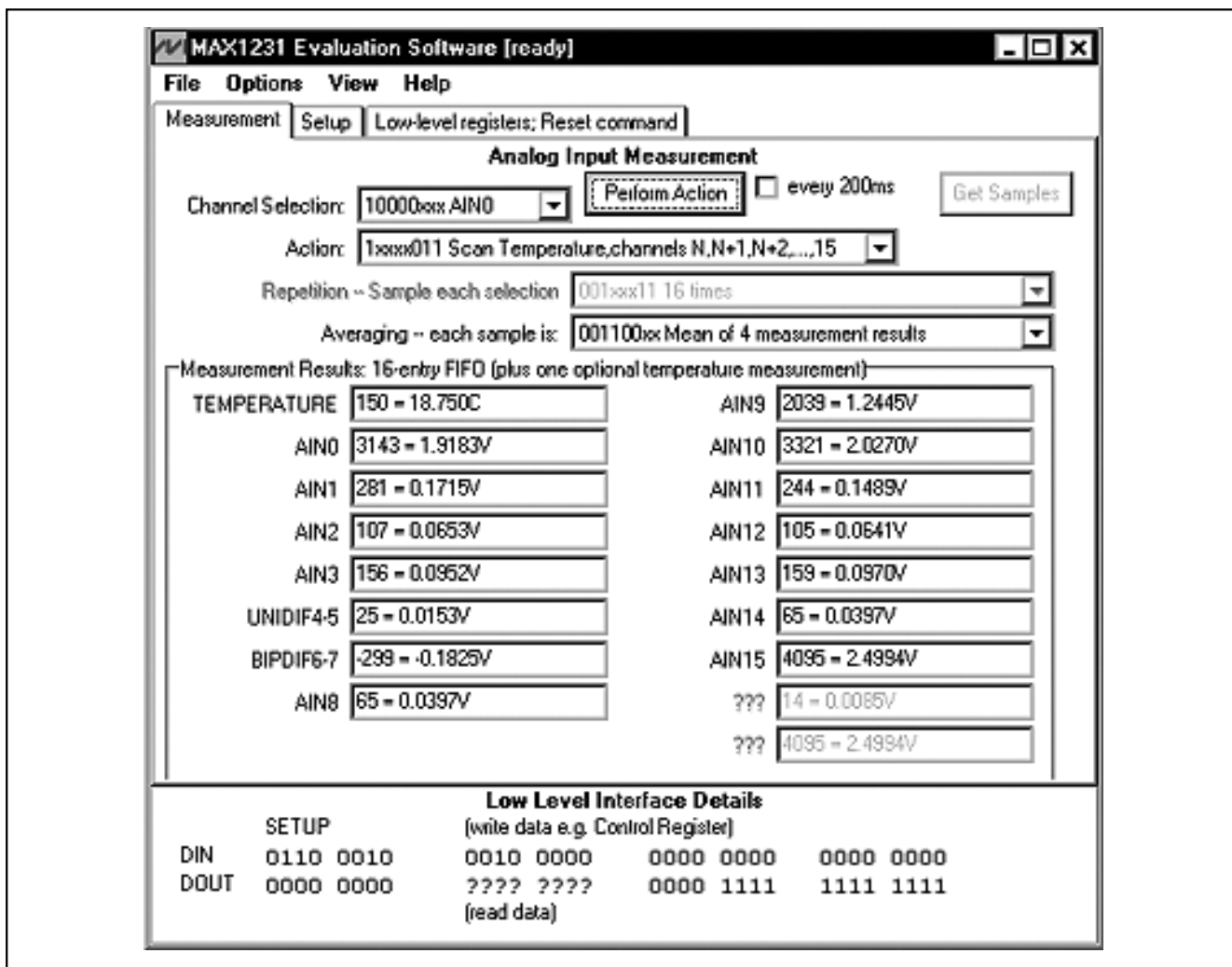


Figure 1. The MAX1231 evaluation software's main window configures the data converter and measures the analog inputs.

MAX1231 Evaluation Kit/ MAX1231 Evaluation System

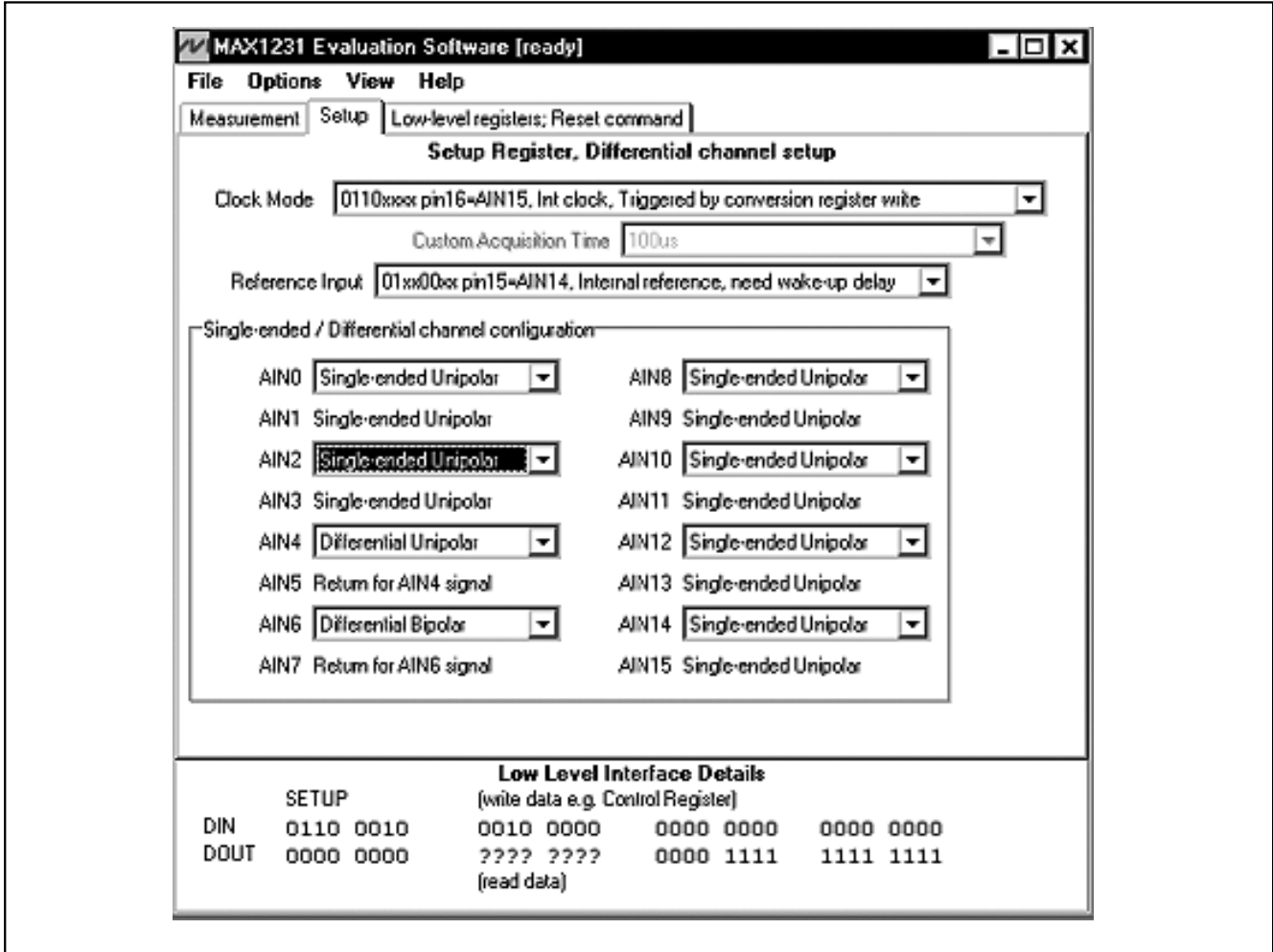


Figure 2. The main window's **Setup** tab configures the alternate functions for AIN14 and AIN15, and also configures adjacent channels as differential input pairs.

MAX1231 Evaluation Kit/ MAX1231 Evaluation System

Evaluates: MAX1030/MAX1031/MAX1230/MAX1231

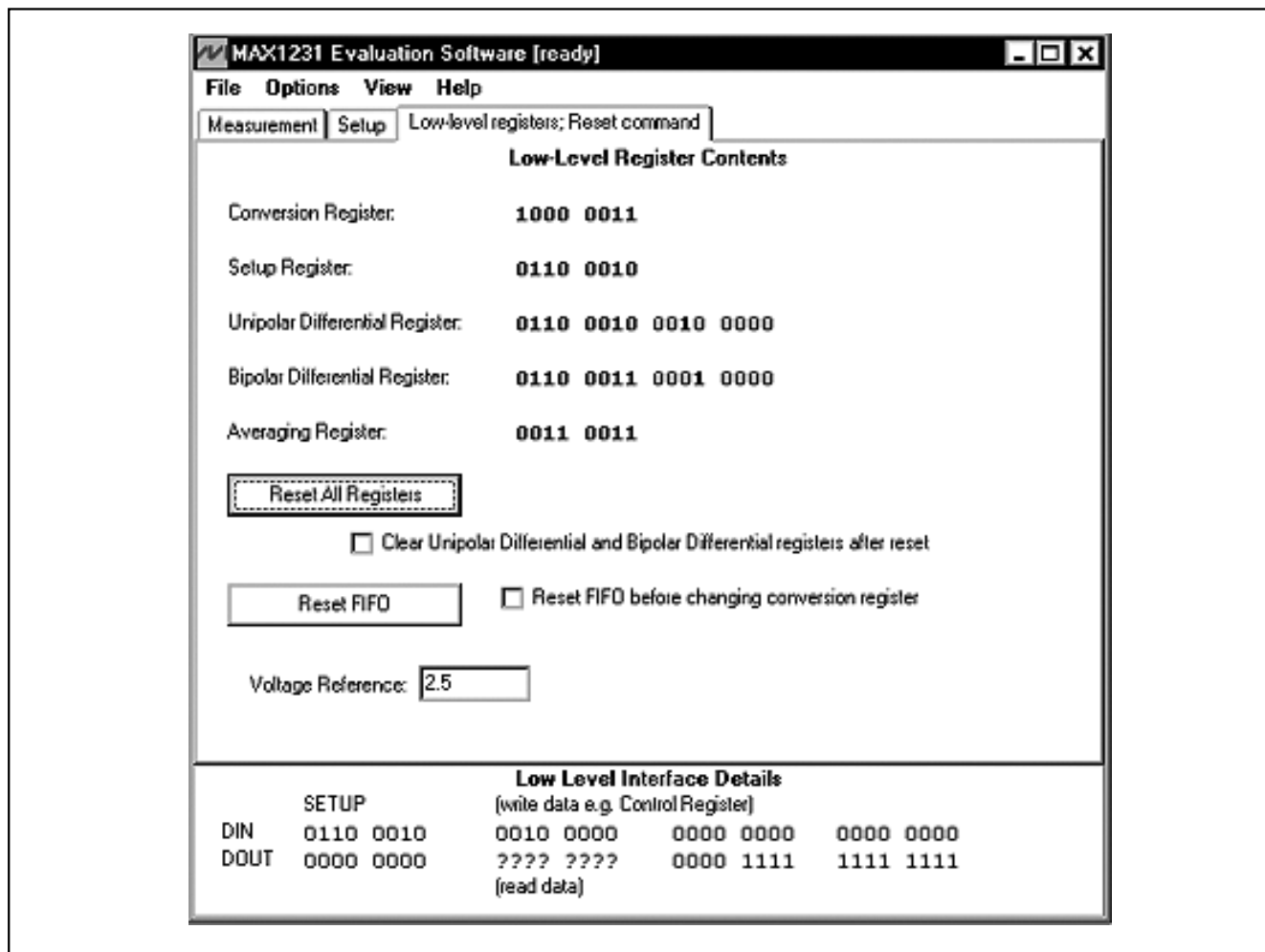


Figure 3. The main window's **Low-level registers** tab summarizes the commands that create the active configuration.

MAX1231 Evaluation Kit/ MAX1231 Evaluation System

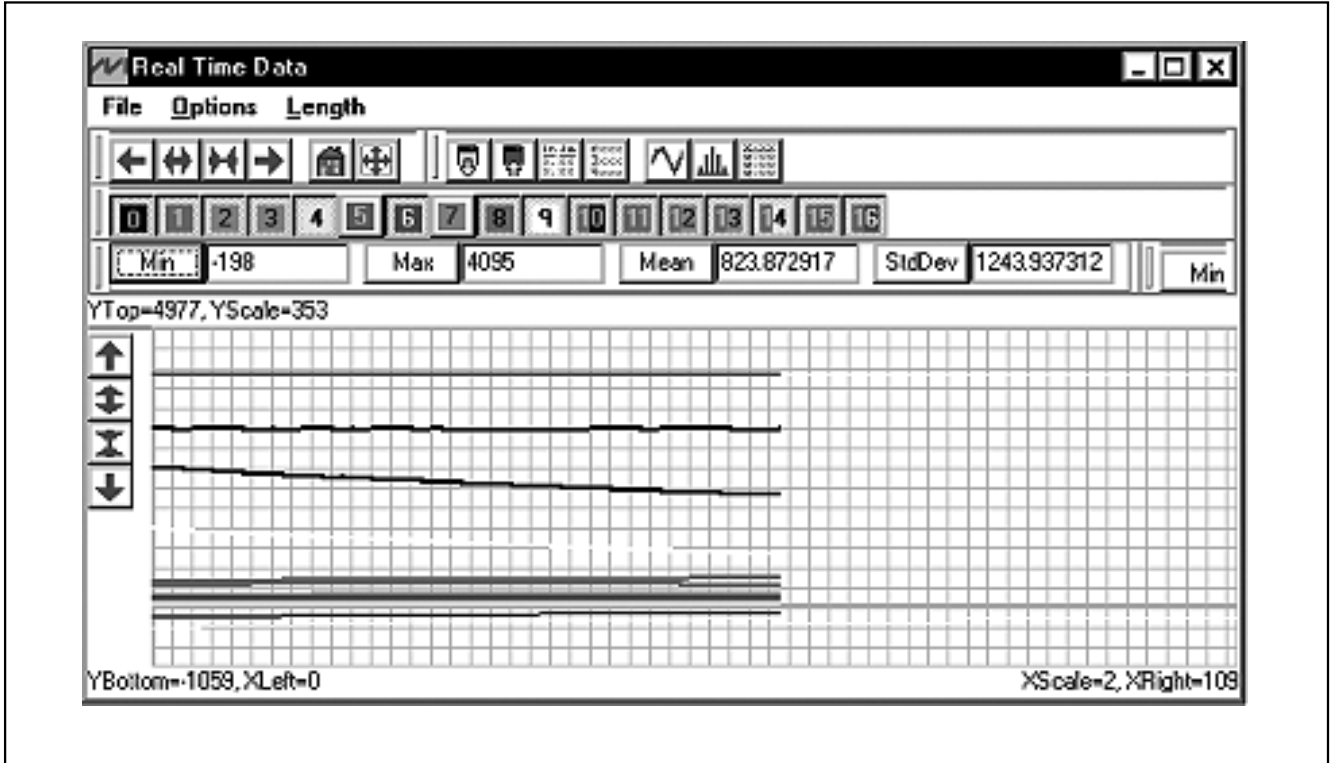


Figure 4. The real-time data and sampled data graphs display data as a time sequence, histogram, or table.

MAX1231 Evaluation Kit/ MAX1231 Evaluation System

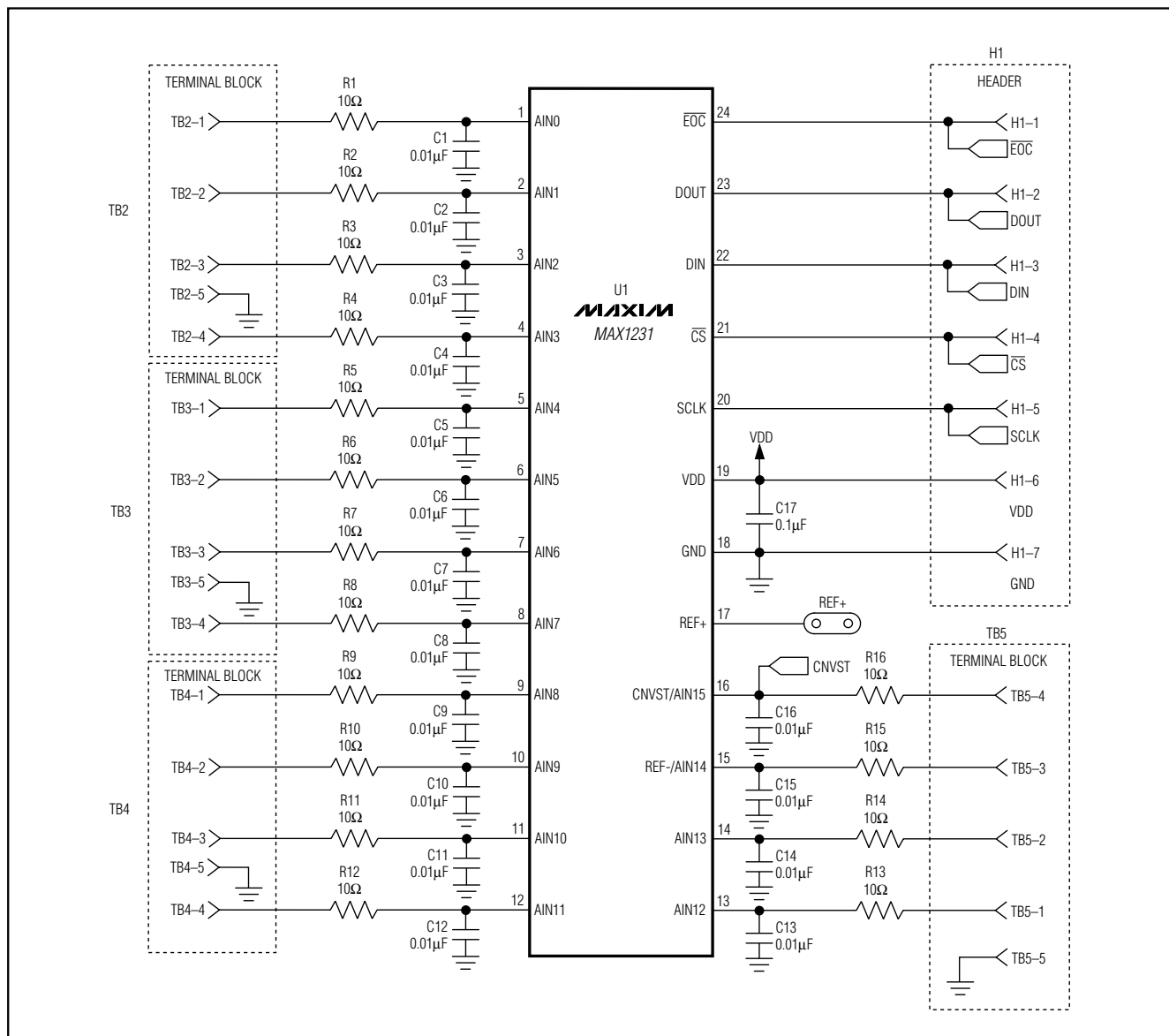


Figure 5a. MAX1231 EV Kit Schematic (Sheet 1 of 2)

Evaluates: MAX1030/MAX1031/MAX1230/MAX1231

MAX1231 Evaluation Kit/ MAX1231 Evaluation System

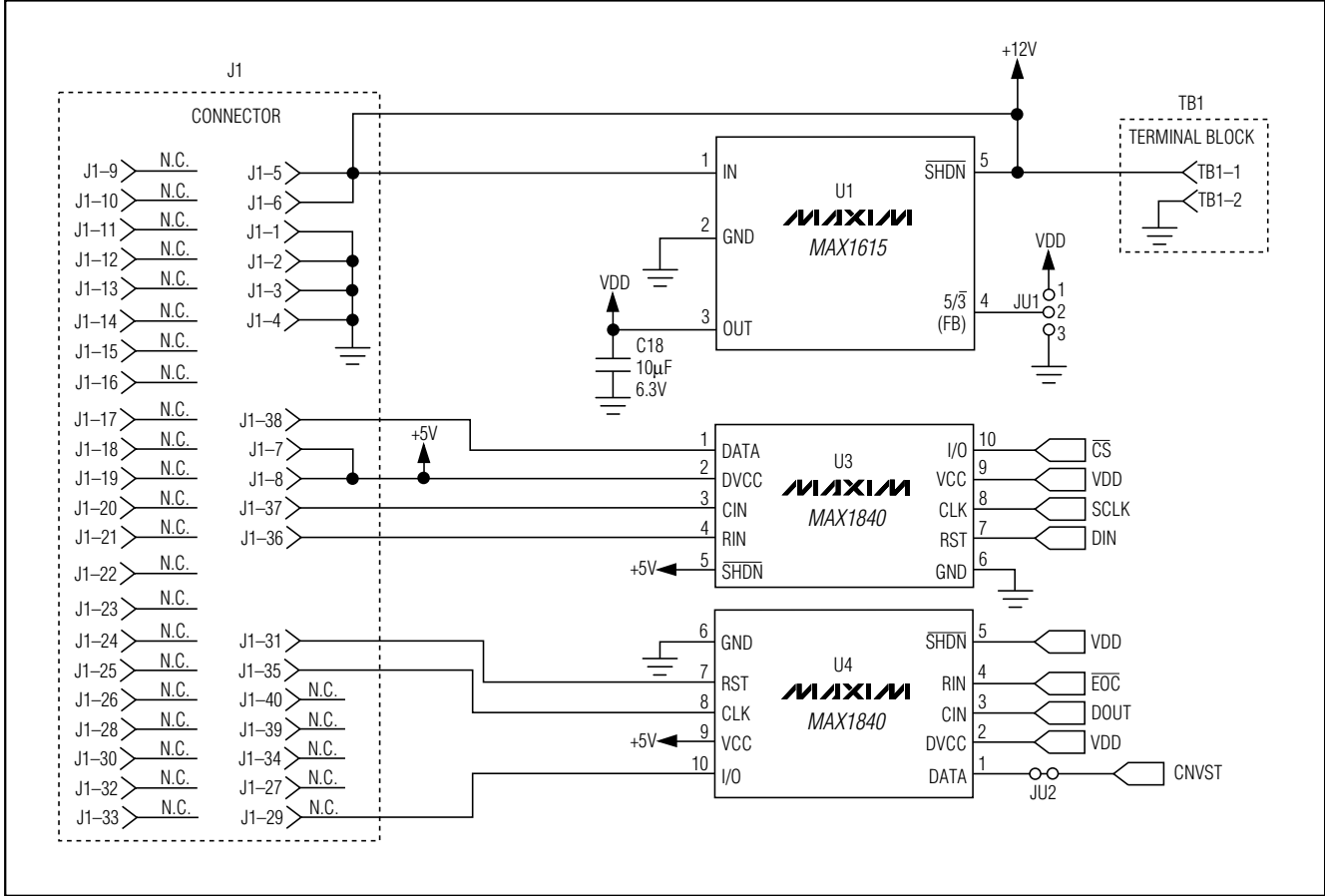


Figure 5b. MAX1231 EV Kit Schematic (Sheet 2 of 2)

MAX1231 Evaluation Kit/ MAX1231 Evaluation System

Evaluates: MAX1030/MAX1031/MAX1230/MAX1231

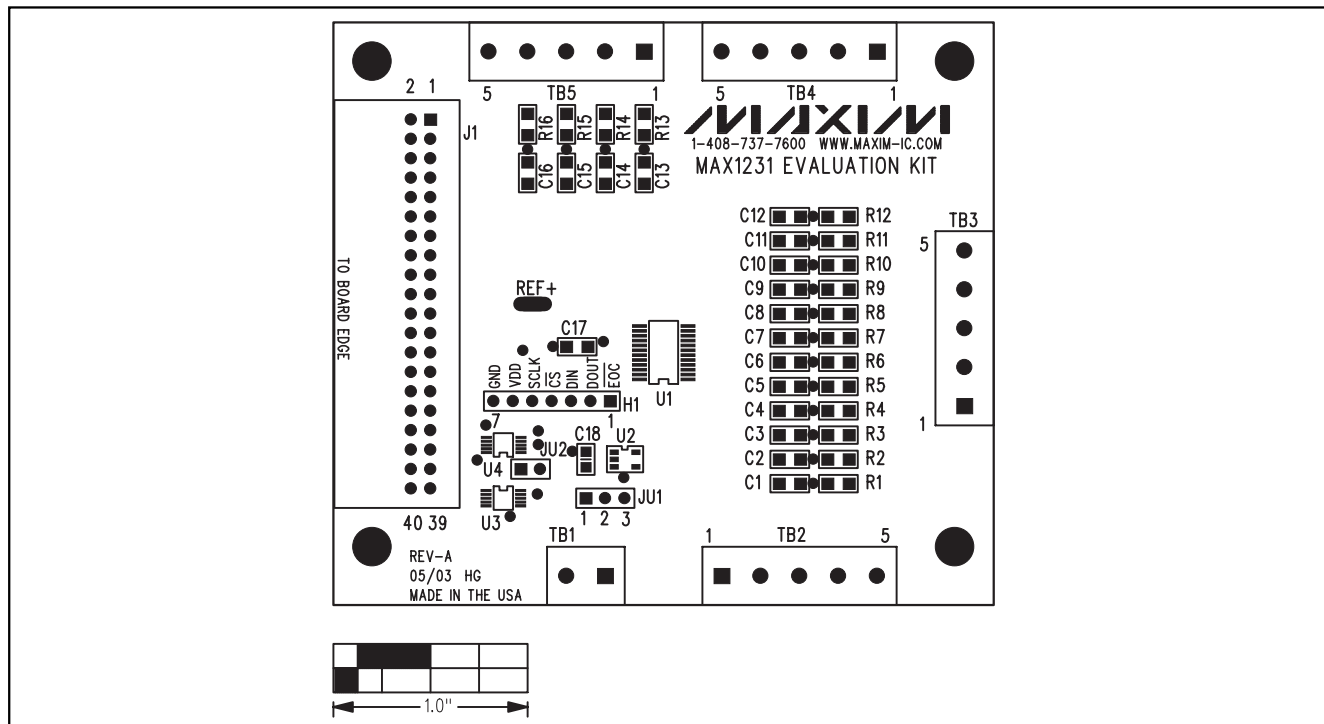


Figure 6. MAX1231 EV Kit Component Placement Guide—Component Side

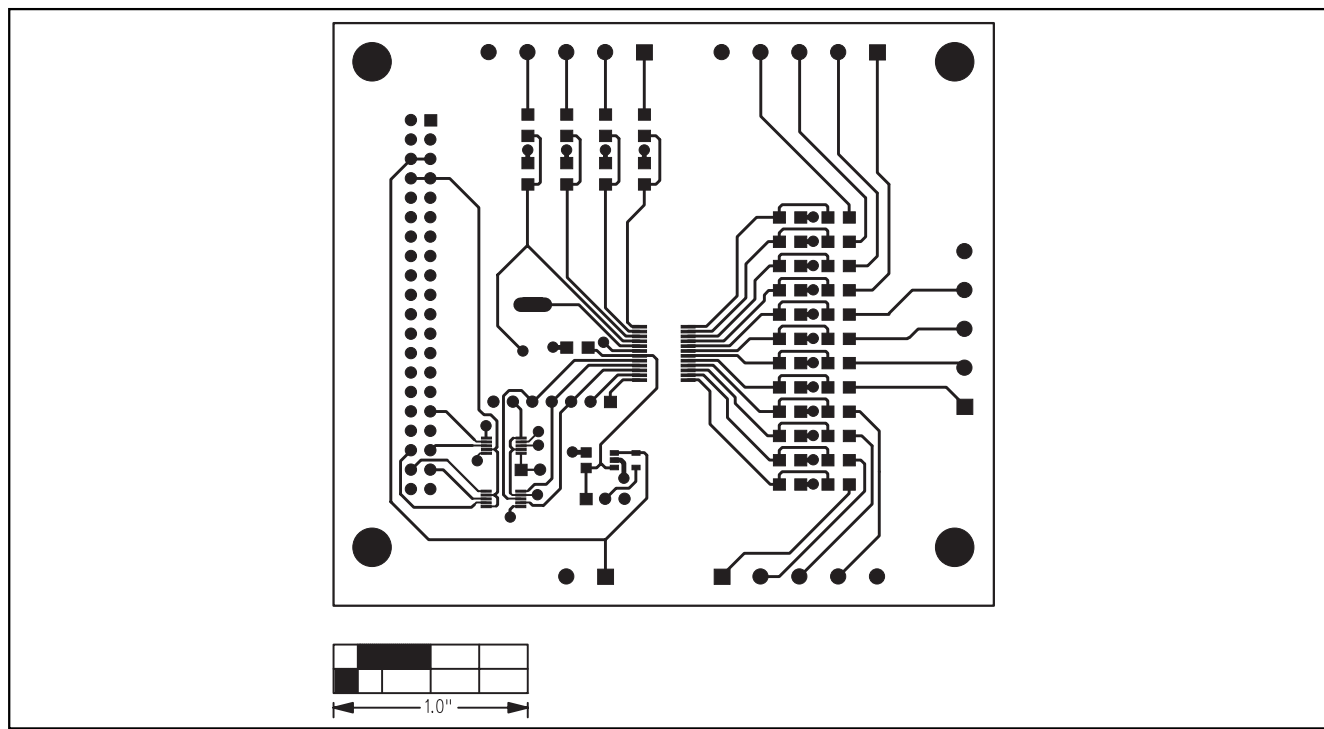


Figure 7. MAX1231 EV Kit PC Board Layout—Component Side

MAX1231 Evaluation Kit/ MAX1231 Evaluation System

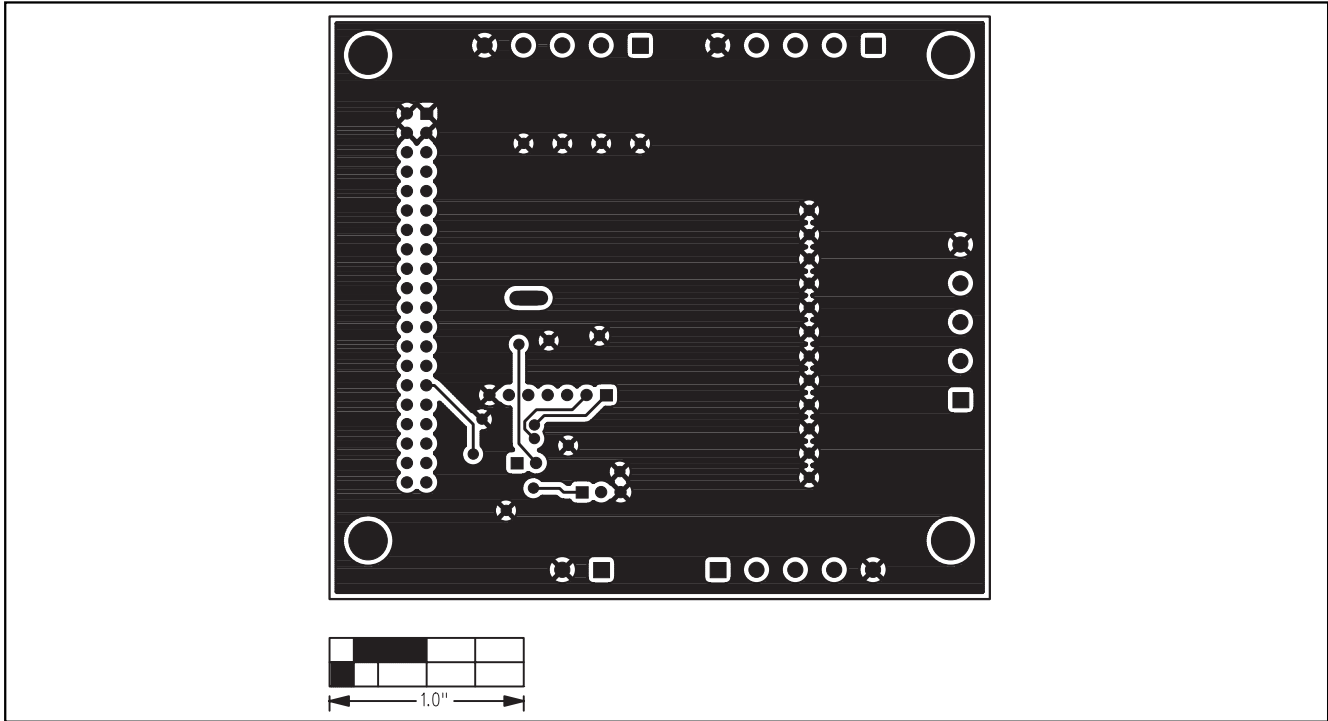


Figure 8. MAX1231 EV Kit PC Board Layout—Solder Side

MAX1231 Evaluation Kit/ MAX1231 Evaluation System

Evaluates: MAX1030/MAX1031/MAX1230/MAX1231

```

// Drv1231.h
// MAX1231-specific driver.
// (C) 2003 Maxim Integrated Products
//-----
#ifndef DRV1231H
#define DRV1231H
//-----

//-----
// MAX1231 Conversion register
// 1 x x x x x x x x
#define MAX1231_CONV          0x80
//
// Power-on state: 1000 0000
#define MAX1231_CONV_POR      0x80
//
// Channel Selection
#define MAX1231_CONV_AIN00    0x80    /* 10000xxx AIN0 */
#define MAX1231_CONV_AIN01    0x88    /* 10001xxx AIN1 */
#define MAX1231_CONV_AIN02    0x90    /* 10010xxx AIN2 */
#define MAX1231_CONV_AIN03    0x98    /* 10011xxx AIN3 */
#define MAX1231_CONV_AIN04    0xA0    /* 10100xxx AIN4 */
#define MAX1231_CONV_AIN05    0xA8    /* 10101xxx AIN5 */
#define MAX1231_CONV_AIN06    0xB0    /* 10110xxx AIN6 */
#define MAX1231_CONV_AIN07    0xB8    /* 10111xxx AIN7 */
#define MAX1231_CONV_AIN08    0xC0    /* 11000xxx AIN8 */
#define MAX1231_CONV_AIN09    0xC8    /* 11001xxx AIN9 */
#define MAX1231_CONV_AIN10    0xD0    /* 11010xxx AIN10 */
#define MAX1231_CONV_AIN11    0xD8    /* 11011xxx AIN11 */
#define MAX1231_CONV_AIN12    0xE0    /* 11100xxx AIN12 */
#define MAX1231_CONV_AIN13    0xE8    /* 11101xxx AIN13 */
#define MAX1231_CONV_AIN14    0xF0    /* 11110xxx AIN14 */
#define MAX1231_CONV_AIN15    0xF8    /* 11111xxx AIN15 */
//
// Actions
#define MAX1231_CONV_SCAN_00_N 0x80    /* 1xxxx000 Scan 0,1,2,...N */
#define MAX1231_CONV_SCAN_T_00_N 0x81    /* 1xxxx001 Scan T,0,1,2,...N */
#define MAX1231_CONV_SCAN_N_15 0x82    /* 1xxxx010 Scan N,N+1,...,15 */
#define MAX1231_CONV_SCAN_T_N_15 0x83    /* 1xxxx011 Scan T,N,N+1,...,15 */
#define MAX1231_CONV_SINGLE_REPEAT 0x84    /* 1xxxx10x Read repeatedly */
#define MAX1231_CONV_SINGLE_READ 0x86    /* 1xxxx11x Read once */
//
#define MAX1231_ACTION_MASK      0x87    /* 1xxxx111 bits to test*/
//-----
// MAX1231 Setup register
// 0 1 x x x x 0 0
//
// Setup register may optionally be followed by
// one of the the differential configuration registers.
// 01xxxx10 followed by a second byte, selecting Unipolar-Differential inputs
// 01xxxx11 followed by a second byte, selecting Bipolar-Differential inputs
#define MAX1231_SETUP            0x40    /* 01xxxx00 no additional bytes */
#define MAX1231_SETUP_UNIDIFF    0x42    /* 01xxxx10 followed by another byte */
#define MAX1231_SETUP_BIPDIFF    0x43    /* 01xxxx11 followed by another byte */
//
// Power-on state: 0110 0000
#define MAX1231_SETUP_POR        0x60
//
// Clock Mode
// 0100xxxx pin16=CNVST, Int clock, Triggered by CNVST pulse
// 0101xxxx pin16=CNVST, Int clock, Triggered by CNVST pulses, custom Tacq
// 0110xxxx pin16=AIN15, Int clock, Triggered by conversion register write
// 0111xxxx pin16=AIN15, Ext clock, Triggered by conversion register write
#define MAX1231_SETUP_INTCLK_CNVST 0x40    /* 0100xxxx CNVST */

```

Listing 1 (Sheet 1 of 4)

MAX1231 Evaluation Kit/ MAX1231 Evaluation System

```

#define MAX1231_SETUP_INTCLK_CNVST_TACQ      0x50    /* 0101xxxx CNVST */
#define MAX1231_SETUP_INTCLK                 0x60    /* 0110xxxx AIN15 */
#define MAX1231_SETUP_EXTCLK                 0x70    /* 0111xxxx AIN15 */
//
// Reference Voltage
// 01xx00xx pin15=AIN14, Internal reference, need wake-up delay
// 01xx01xx pin15=AIN14, External reference, single-ended
// 01xx10xx pin15=AIN14, Internal reference, always on
// 01xx11xx pin15=REF-, External reference, differential
#define MAX1231_SETUP_INTREF_SLEEP           0x40    /* 01xx00xx AIN14 */
#define MAX1231_SETUP_EXTREF                 0x44    /* 01xx01xx AIN14 */
#define MAX1231_SETUP_INTREF_ACTIVE          0x48    /* 01xx10xx AIN14 */
#define MAX1231_SETUP_EXTREF_DIFF           0x4C    /* 01xx11xx REF(-) */
//
// MAX1231 Unipolar-Differential input pairs
// Byte Following MAX1231_SETUP_UNIDIFF
// 01 x x x x 10 unidiff
//
// Power-on state: 0110 0010 0000 0000
#define MAX1231_SETUP_UNIDIF_POR             0x00
//
#define MAX1231_SETUP_UNIDIF0001            0x80
#define MAX1231_SETUP_UNIDIF0203            0x40
#define MAX1231_SETUP_UNIDIF0405            0x20
#define MAX1231_SETUP_UNIDIF0607            0x10
#define MAX1231_SETUP_UNIDIF0809            0x08
#define MAX1231_SETUP_UNIDIF1011            0x04
#define MAX1231_SETUP_UNIDIF1213            0x02
#define MAX1231_SETUP_UNIDIF1415            0x01
//
// MAX1231 Bipolar-Differential input pairs
// Byte Following MAX1231_SETUP_BIPDIFF
// 01 x x x x 11 bipdiff
//
// Power-on state: 0110 0011 0000 0000
#define MAX1231_SETUP_BIPDIF_POR            0x00
//
#define MAX1231_SETUP_BIPDIF0001            0x80
#define MAX1231_SETUP_BIPDIF0203            0x40
#define MAX1231_SETUP_BIPDIF0405            0x20
#define MAX1231_SETUP_BIPDIF0607            0x10
#define MAX1231_SETUP_BIPDIF0809            0x08
#define MAX1231_SETUP_BIPDIF1011            0x04
#define MAX1231_SETUP_BIPDIF1213            0x02
#define MAX1231_SETUP_BIPDIF1415            0x01

//-----
// MAX1231 Averaging register
// 001 x x x x x
//
// Power-on state: 0010 0000
#define MAX1231_AVERAGE_POR                 0x20
//
// Averaging
// 001000xx One measurement result (no averaging)
// 001100xx Mean of 4 measurement results
// 001101xx Mean of 8 measurement results
// 001110xx Mean of 16 measurement results
// 001111xx Mean of 32 measurement results
#define MAX1231_AVERAGE_1                   0x20    /* 001000xx No averaging */
#define MAX1231_AVERAGE_4                   0x30    /* 001100xx Mean of 4 measurements */
#define MAX1231_AVERAGE_8                   0x34    /* 001101xx Mean of 8 measurements */
#define MAX1231_AVERAGE_16                  0x38    /* 001110xx Mean of 16 measurements */
#define MAX1231_AVERAGE_32                  0x3C    /* 001111xx Mean of 32 measurements */

```

Listing 1 (Sheet 2 of 4)

MAX1231 Evaluation Kit/ MAX1231 Evaluation System

Evaluates: MAX1030/MAX1031/MAX1230/MAX1231

```
//
// Repeat Count
// Enabled by MAX1231_CONV_SINGLE_REPEAT 1xxxx10x
// Internal clock modes only
#define MAX1231_REPEAT_4      0x20    /* 001xxx00 4 times */
#define MAX1231_REPEAT_8      0x21    /* 001xxx01 8 times */
#define MAX1231_REPEAT_12     0x22    /* 001xxx10 12 times */
#define MAX1231_REPEAT_16     0x23    /* 001xxx11 16 times */

//-----
// MAX1231 Reset register (reset command)
// 0 0 0 1 x x x x
//
// Reset only the FIFO entries
#define MAX1231_RESET_FIFO    0x18    /* 00011xxx Reset FIFO data registers */
//
// Reset all registers to their power-on default states
#define MAX1231_RESET_ALL     0x10    /* 00010xxx Reset All Registers */

//-----
// Enumerated type defining the meaning of each of the
// MAX1231's FIFO data slots.
typedef enum {
    //
    // Unused FIFO slot; meaningless data.
    UNDEFINED = 0,
    //
    // Temperature measurement.
    // The scan modes always place temperature data
    // at the head of the FIFO.
    TEMPERATURE,
    //
    // Single-ended unipolar analog inputs.
    // Code 0x0000 = minimum voltage
    // Code 0x0FFF = maximum voltage
    // Note that AIN14 and AIN15 pins have optional alternate functions.
    UNIAIN00, UNIAIN01, UNIAIN02, UNIAIN03,
    UNIAIN04, UNIAIN05, UNIAIN06, UNIAIN07,
    UNIAIN08, UNIAIN09, UNIAIN10, UNIAIN11,
    UNIAIN12, UNIAIN13, UNIAIN14, UNIAIN15,
    //
    // Unipolar differential input pairs.
    // Code 0x0000 = minimum voltage
    // Code 0x0FFF = maximum voltage
    UNIDIF0001, UNIDIF0203, UNIDIF0405, UNIDIF0607,
    UNIDIF0809, UNIDIF1011, UNIDIF1213, UNIDIF1415,
    //
    // Bipolar differential input pairs.
    // Code 0x07FF = maximum voltage
    // Code 0x0000 = zero volts
    // Code 0x0800 = minimum voltage
    BIPDIF0001, BIPDIF0203, BIPDIF0405, BIPDIF0607,
    BIPDIF0809, BIPDIF1011, BIPDIF1213, BIPDIF1415,
    //
    NUM_FIFO_ENTRY_TYPES
} MAX1231_fifo_entry_t;

//-----
// Enumerated type defining each pair of MAX1231 inputs
// as single-ended or differential
typedef enum {
    SINGLE_ENDED = 0,
    UNIPOLAR_DIFFERENTIAL,

```

Listing 1 (Sheet 3 of 4)

MAX1231 Evaluation Kit/ MAX1231 Evaluation System

Evaluates: MAX1030/MAX1031/MAX1230/MAX1231

```
        BIPOLAR_DIFFERENTIAL
    } MAX1231_channel_config_t;

//-----
// C++ class representing the state of a MAX1231
class MAX1231
{
public:
    // MAX1231 registers cannot be read,
    // so keep track of the register values here.
    int conversion_register;
    int new_conversion_register;
    int setup_register;
    int setup_unidiff_register;
    int setup_bipdiff_register;
    int averaging_register;

    // The reference voltage is used to calculate the input voltage
    // represented by each measurement.
    double Vref;

    // Constructor for class MAX1231.
    MAX1231(void) ;

    // Input configuration
    // Array InputPairConfig[] determines whether each pair
    // of input channels is configured as two single-ended inputs,
    // a unipolar differential pair, or a bipolar differential pair.
    MAX1231_channel_config_t InputPairConfig[8];
    //
    // Member function to figure out the values of InputPairConfig
    // based on the MAX1231 register values.
    // Call this function after setting setup_unidiff_register and setup_bipdiff_register
    // before using the values of InputPairConfig[].
    void Update_InputPairConfig(void) ;

    // The MAX1231 has a FIFO data buffer with
    // 16 entries (plus an optional temperature measurement).
    // Array FIFO_meaning[] determines what to do with
    // each successive word read from the MAX1231.
    MAX1231_fifo_entry_t FIFO_meaning[17];
    //
    // Member function to figure out the values of FIFO_meaning
    // based on the MAX1231 register values.
    // Call this function after setting conversion_register, setup_register, and averaging_register
    // before using the values of FIFO_meaning[].
    void Update_FIFO_meaning(void) ;

    int channel (MAX1231_fifo_entry_t meaning) {
        if ((UNIAIN00 <= meaning) && (meaning <= UNIAIN15)) {
            return (meaning - UNIAIN00);
        } else
        if ((UNIDIF0001 <= meaning) && (meaning <= UNIDIF1415)) {
            return (meaning - UNIDIF0001) * 2;
        } else
        if ((BIPDIF0001 <= meaning) && (meaning <= BIPDIF1415)) {
            return (meaning - BIPDIF0001) * 2;
        } else
        if (meaning == TEMPERATURE) {
            return 16;
        } else {
            return 0;
        }
    }
};

//-----
#endif // DRV1231H
```

Listing 1 (Sheet 4 of 4)

MAX1231 Evaluation Kit/ MAX1231 Evaluation System

Evaluates: MAX1030/MAX1031/MAX1230/MAX1231

```
// Drv1231.cpp
// MAX1231-specific driver.
// (C) 2003 Maxim Integrated Products
//-----

#include "Drv1231.h"

//-----
MAX1231::MAX1231(void)
{
    conversion_register = MAX1231_CONV_POR;
    setup_register = MAX1231_SETUP_POR;
    setup_unidiff_register = MAX1231_SETUP_UNIDIF_POR;
    setup_bipdiff_register = MAX1231_SETUP_BIPDIF_POR;
    averaging_register = MAX1231_AVERAGE_POR;
    Vref = 2.500;
    for (int index = 0; index < 8; index++) {
        InputPairConfig[index] = SINGLE_ENDED;
    }
    for (int index = 0; index < 17; index++) {
        FIFO_meaning[index] = UNDEFINED;
    }
}
//-----
void MAX1231::Update_InputPairConfig(void)
{
    for (int index = 0; index < 8; index++) {
        InputPairConfig[index] = SINGLE_ENDED;
    }

    if (setup_bipdiff_register & MAX1231_SETUP_BIPDIF0001)
        InputPairConfig[0] = BIPOLAR_DIFFERENTIAL;
    if (setup_bipdiff_register & MAX1231_SETUP_BIPDIF0203)
        InputPairConfig[1] = BIPOLAR_DIFFERENTIAL;
    if (setup_bipdiff_register & MAX1231_SETUP_BIPDIF0405)
        InputPairConfig[2] = BIPOLAR_DIFFERENTIAL;
    if (setup_bipdiff_register & MAX1231_SETUP_BIPDIF0607)
        InputPairConfig[3] = BIPOLAR_DIFFERENTIAL;
    if (setup_bipdiff_register & MAX1231_SETUP_BIPDIF0809)
        InputPairConfig[4] = BIPOLAR_DIFFERENTIAL;
    if (setup_bipdiff_register & MAX1231_SETUP_BIPDIF1011)
        InputPairConfig[5] = BIPOLAR_DIFFERENTIAL;
    if (setup_bipdiff_register & MAX1231_SETUP_BIPDIF1213)
        InputPairConfig[6] = BIPOLAR_DIFFERENTIAL;
    if (setup_bipdiff_register & MAX1231_SETUP_BIPDIF1415)
        InputPairConfig[7] = BIPOLAR_DIFFERENTIAL;

    if (setup_unidiff_register & MAX1231_SETUP_UNIDIF0001)
        InputPairConfig[0] = UNIPOLAR_DIFFERENTIAL;
    if (setup_unidiff_register & MAX1231_SETUP_UNIDIF0203)
        InputPairConfig[1] = UNIPOLAR_DIFFERENTIAL;
    if (setup_unidiff_register & MAX1231_SETUP_UNIDIF0405)
        InputPairConfig[2] = UNIPOLAR_DIFFERENTIAL;
    if (setup_unidiff_register & MAX1231_SETUP_UNIDIF0607)
        InputPairConfig[3] = UNIPOLAR_DIFFERENTIAL;
    if (setup_unidiff_register & MAX1231_SETUP_UNIDIF0809)
        InputPairConfig[4] = UNIPOLAR_DIFFERENTIAL;
    if (setup_unidiff_register & MAX1231_SETUP_UNIDIF1011)
        InputPairConfig[5] = UNIPOLAR_DIFFERENTIAL;
    if (setup_unidiff_register & MAX1231_SETUP_UNIDIF1213)
        InputPairConfig[6] = UNIPOLAR_DIFFERENTIAL;
    if (setup_unidiff_register & MAX1231_SETUP_UNIDIF1415)
        InputPairConfig[7] = UNIPOLAR_DIFFERENTIAL;
}
//-----
void MAX1231::Update_FIFO_meaning(void)
{
    int conversion_register = new_conversion_register;
```

Listing 2 (Sheet 1 of 4)

MAX1231 Evaluation Kit/ MAX1231 Evaluation System

```

for (int index = 0; index < 17; index++) {
    FIFO_meaning[index] = UNDEFINED;
}

int channel_field = (conversion_register >> 3) & 0x0F;
int channel_index = channel_field;
int repeat_count = 4;
switch(averaging_register & 0xE3) {
case MAX1231_REPEAT_4: // 001xxx00 4 times
    repeat_count = 4;
    break;
case MAX1231_REPEAT_8: // 001xxx01 8 times
    repeat_count = 8;
    break;
case MAX1231_REPEAT_12: // 001xxx10 12 times
    repeat_count = 12;
    break;
case MAX1231_REPEAT_16: // 001xxx11 16 times
    repeat_count = 16;
    break;
}

MAX1231_fifo_entry_t meaning =
    (MAX1231_fifo_entry_t)(UNIAIN00 + channel_field);
if (InputPairConfig[channel_field / 2] == BIPOLAR_DIFFERENTIAL) {
    meaning = (MAX1231_fifo_entry_t)(BIPDIF0001 + channel_field/2);
} else if (InputPairConfig[channel_field / 2] == UNIPOLAR_DIFFERENTIAL) {
    meaning = (MAX1231_fifo_entry_t)(UNIDIF0001 + channel_field/2);
}
int index = 0;
switch(conversion_register & MAX1231_ACTION_MASK) {
case MAX1231_CONV_SCAN_00_N:
    meaning = UNIAIN00;
    channel_index = 0;
    while(channel_index <= channel_field) {
        int pair_index = channel_index / 2;
        if (InputPairConfig[pair_index] == BIPOLAR_DIFFERENTIAL) {
            meaning = (MAX1231_fifo_entry_t)(BIPDIF0001 + pair_index);
            FIFO_meaning[index++] = meaning;
            channel_index = channel_index + 2;
        } else if (InputPairConfig[pair_index] == UNIPOLAR_DIFFERENTIAL) {
            meaning = (MAX1231_fifo_entry_t)(UNIDIF0001 + pair_index);
            FIFO_meaning[index++] = meaning;
            channel_index = channel_index + 2;
        } else {
            meaning = (MAX1231_fifo_entry_t)(UNIAIN00 + channel_index);
            FIFO_meaning[index++] = meaning;
            channel_index = channel_index + 1;
        }
    }
    break;
case MAX1231_CONV_SCAN_T_00_N:
    FIFO_meaning[index++] = TEMPERATURE;
    meaning = UNIAIN00;
    channel_index = 0;
    while(channel_index <= channel_field) {
        int pair_index = channel_index / 2;
        if (InputPairConfig[pair_index] == BIPOLAR_DIFFERENTIAL) {
            meaning = (MAX1231_fifo_entry_t)(BIPDIF0001 + pair_index);
            FIFO_meaning[index++] = meaning;
            channel_index = channel_index + 2;
        } else if (InputPairConfig[pair_index] == UNIPOLAR_DIFFERENTIAL) {
            meaning = (MAX1231_fifo_entry_t)(UNIDIF0001 + pair_index);
            FIFO_meaning[index++] = meaning;
            channel_index = channel_index + 2;
        } else {
            meaning = (MAX1231_fifo_entry_t)(UNIAIN00 + channel_index);
            FIFO_meaning[index++] = meaning;
            channel_index = channel_index + 1;
        }
    }
}
}

```

Listing 2 (Sheet 2 of 4)

MAX1231 Evaluation Kit/ MAX1231 Evaluation System

Evaluates: MAX1030/MAX1031/MAX1230/MAX1231

```

        break;
    case MAX1231_CONV_SCAN_N_15:
        while (index < 17) {
            int pair_index = channel_index / 2;
            if (InputPairConfig[pair_index] == BIPOLAR_DIFFERENTIAL) {
                meaning = (MAX1231_fifo_entry_t)(BIPDIF0001 + pair_index);
                FIFO_meaning[index++] = meaning;
                channel_index = channel_index + 2;
            } else if (InputPairConfig[pair_index] == UNIPOLAR_DIFFERENTIAL) {
                meaning = (MAX1231_fifo_entry_t)(UNIDIF0001 + pair_index);
                FIFO_meaning[index++] = meaning;
                channel_index = channel_index + 2;
            } else {
                meaning = (MAX1231_fifo_entry_t)(UNIAIN00 + channel_index);
                FIFO_meaning[index++] = meaning;
                channel_index = channel_index + 1;
            }
            if (meaning == UNIAIN15) break;
            if (meaning == UNIDIF1415) break;
            if (meaning == BIPDIF1415) break;
        }
        break;
    case MAX1231_CONV_SCAN_T_N_15:
        FIFO_meaning[index++] = TEMPERATURE;
        while (index < 17) {
            int pair_index = channel_index / 2;
            if (InputPairConfig[pair_index] == BIPOLAR_DIFFERENTIAL) {
                meaning = (MAX1231_fifo_entry_t)(BIPDIF0001 + pair_index);
                FIFO_meaning[index++] = meaning;
                channel_index = channel_index + 2;
            } else if (InputPairConfig[pair_index] == UNIPOLAR_DIFFERENTIAL) {
                meaning = (MAX1231_fifo_entry_t)(UNIDIF0001 + pair_index);
                FIFO_meaning[index++] = meaning;
                channel_index = channel_index + 2;
            } else {
                meaning = (MAX1231_fifo_entry_t)(UNIAIN00 + channel_index);
                FIFO_meaning[index++] = meaning;
                channel_index = channel_index + 1;
            }
            if (meaning == UNIAIN15) break;
            if (meaning == UNIDIF1415) break;
            if (meaning == BIPDIF1415) break;
        }
        break;
    case MAX1231_CONV_SINGLE_REPEAT:
        while (index < repeat_count) {
            FIFO_meaning[index++] = meaning;
        }
        break;
    case MAX1231_CONV_SINGLE_READ:
        FIFO_meaning[index++] = meaning;
        break;
}

/* Check for setups where AIN14-AIN15 are used for an alternate function */
if ((setup_register & 0xE0) == MAX1231_SETUP_INTCLK_CNVS_T) {
    /* 0100xxxx AIN15 alternate function as CNVS_T input */
    /* 0101xxxx AIN15 alternate function as CNVS_T input */
    for (int index = 0; index < 17; index++) {
        meaning = FIFO_meaning[index];
        if (meaning == UNIAIN15)
            FIFO_meaning[index] = UNDEFINED;
        if (meaning == UNIDIF1415)
            FIFO_meaning[index] = UNDEFINED;
        if (meaning == BIPDIF1415)
            FIFO_meaning[index] = UNDEFINED;
    }
}
if ((setup_register & 0xCC) == MAX1231_SETUP_EXTREF_DIFF) {
    /* 01xx11xx AIN14 alternate function as REF- input */
    for (int index = 0; index < 17; index++) {

```

Listing 2 (Sheet 3 of 4)

MAX1231 Evaluation Kit/ MAX1231 Evaluation System

```
        meaning = FIFO_meaning[index];  
        if (meaning == UNIAIN14)  
            FIFO_meaning[index] = UNDEFINED;  
        if (meaning == UNIDIF1415)  
            FIFO_meaning[index] = UNDEFINED;  
        if (meaning == BIPDIF1415)  
            FIFO_meaning[index] = UNDEFINED;  
    }  
}  
//-----
```

Listing 2 (Sheet 4 of 4)

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.

20 _____ **Maxim Integrated Products, 120 San Gabriel Drive, Sunnyvale, CA 94086 408-737-7600**