



# MAX1499 평가 키트/MAX1499 평가 시스템

평가 대상: MAX1498/MAX1499

## 제품설명

MAX1499 평가 시스템 (EV 시스템)은 MAX1499 평가 키트 (EV kit)과 Maxim 68HC16MODULE-DIP 마이크로컨트롤러 ( $\mu$ C) 모듈로 구성된다. MAX1499는 LED 디스플레이 드라이버가 있는 저전력, 4.5 디지털 아날로그-디지털 컨버터 (ADC)이다. 평가 소프트웨어는 Windows® 95/98/2000/XP 에서 실행되므로 MAX1499의 기능을 실행하기 위한 편리한 사용자 인터페이스가 제공된다.

PC를 사용하여 MAX1499를 종합적으로 평가하려면 완벽한 EV 시스템 (MAX1499EVC16)을 주문하면 된다. 이전 Maxim EV 시스템과 함께 68HC16MODULE을 이미 구매 했거나 다른  $\mu$ C 기반 시스템에서 맞춤 사용하려면 이 EV 키트 (MAX1499EVKIT)를 주문한다.

이 시스템은 또한 MAX1498ECJ도 평가할 수 있다. 무료 샘플을 구하려면 공장에 문의한다. 자세한 내용은 *하드웨어 세부 설명*을 참조한다.

## MAX1499 독립형 EV 키트

MAX1499 EV 키트는 손쉬운 MAX1499 평가를 위한 입증된 PCB 레이아웃을 제공한다. 이 키트는 원활한 동작을 위해 해당 적합한 신호와 인터페이스되어야 한다. 6V~26VDC를 연결하고 TB1 블록 단자에 접지를 리턴한다. (그림 7 참조). 타이밍 요구사항은 MAX1499 데이터 시트를 참조한다.

## MAX1499 EV 시스템

MAX1499 EV 시스템은 사용자가 제공하는 7VDC~20VDC 전원으로 동작한다. 평가 소프트웨어는 PC에서 Windows 95/98/2000/XP 에서 실행되고 컴퓨터의 직렬 통신 포트를 통해 EV 시스템 보드에 인터페이스된다. 설정 및 동작 지침은 *퀵 스타트*를 참조한다.

## 제품특징

- ◆ 입증된 PCB 레이아웃
- ◆ 완벽한 평가 시스템
- ◆ 편리한 온 보드 테스트 포인트
- ◆ 데이터 로깅 소프트웨어
- ◆ 완벽하게 조립 및 시험된 보드

## 주문정보

PART	TEMP RANGE	INTERFACE TYPE
MAX1499EVKIT	0°C to +70°C	User supplied
MAX1499EVC16	0°C to +70°C	Windows software

**참고:** MAX1499 평가 소프트웨어는 완벽한 평가 시스템 MAX1499EVC16 (68HC16MODULE-DIP 모듈과 MAX1499EVKIT이 모두 포함됨)과 함께 사용하도록 설계되었다. MAX1499 평가 소프트웨어를 사용하지 않을 경우, MAX1499EVKIT 보드는  $\mu$ C 없이 보드 자체만 구입이 가능하다.

## MAX1499EVC16 시스템 부품 목록

PART	QTY	DESCRIPTION
MAX1499EVKIT	1	MAX1499 EV kit
68HC16MODULE-DIP	1	68HC16 $\mu$ C module

## MAX1499 EV 키트 부품 목록

DESIGNATION	QTY	DESCRIPTION
AIN+, AIN-, REF+, REF-	4	Noninsulated banana jacks Mouser 530-108-0740-1
C1, C2, C11	3	10 $\mu$ F $\pm$ 20%, 10V X7R ceramic capacitors (1210) TDK C3225X7R1C106M Taiyo Yuden LMK325BJ106MN
C3, C4, C5, C7-C10, C12	8	0.1 $\mu$ F $\pm$ 20%, 16V X7R ceramic capacitors (0603) TDK C1608X7R1C104K Taiyo Yuden EMK107BJ104MA

DESIGNATION	QTY	DESCRIPTION
C6	1	4.7 $\mu$ F $\pm$ 10%, 16V X7R ceramic capacitor (1206) TDK C3216X7R1C475K
CLK	1	BNC 50 $\Omega$ PC board vertical mount A/D Electronics 580-002-00
DIG0-DIG4	5	Bicolor seven-segment LED displays, common cathode (DIP-10-0.600in) Kingbright Corporation SBC56-21EGWA
J1	1	2 x 20 right-angle socket

Windows는 Microsoft Corp.의 등록상표이다.



## MAX1499 평가 키트/MAX1499 평가 시스템

MAX1499 EV 키트 부품 목록  
(계속)

REFERENCE	QTY	DESCRIPTION
JU10-JU14	5	3 pins
JU1-JU9	9	2 pins
R1	1	133kΩ 1% resistor (1206)
R2, R12	2	100kΩ 1% resistors (1206)
R3-R7	5	1kΩ 5% resistors (1206)
R8, R9	0	Do not install—shorted trace on PC board (1206)
R10	1	500kΩ potentiometer
R11	1	24kΩ 5% resistor (1206)
R13, R14	2	10Ω 5% resistors (1206)
TB1	1	0.200in two-circuit screw terminal block
TP1-TP4	4	8 pins
U1	1	MAX1499ECJ
U2	1	MAX1659ESA
U3, U4	2	MAX1840EUB or MAX1841EUB
U5	1	MAX6062AEUR-T, FZFY
—	1	PC board, MAX1499 EV kit
—	13	Shunts

## 퀵 스타트

## 필요 장비

시작하기 전에 다음 장비가 필요하다.

- MAX1499EVC16 (MAX1499EVKIT 보드 및 68HC16 MODULE-DIP 포함)
- DC 전원, +7VDC~+20VDC / 0.5A
- 사용 가능한 직렬 (COM) 포트가 있는 Windows 95/98/2000/XP 컴퓨터
- 9핀 I/O 확장 케이블

## 테스트 절차

모든 연결이 완료될 때까지 전원을 켜지 않도록 한다.

- 1) JU1-JU8과 JU10-JU14에 셉트가 설치되어 있고 JU9가 개방되어 있는지 확인한다. 표 2의 점퍼 설정을 참조한다.
- 2) MAX1499 EV 키트의 40핀 헤더를 68HC16MODULE-DIP 모듈의 40핀 커넥터와 맞추어 보드를 조심스럽게 연결한다. 헤더와 커넥터를 살며시 누른다. 두 보드는 서로 접해야 한다.
- 3) +7VDC~+20VDC 전원을  $\mu$ C 모듈의 상단 가장자리를 따라 온/오프 스위치 옆에 위치한 단자 블록의  $\mu$ C 모듈에 연결한다. 보드에 표시된 극성을 확인한다.
- 4) 컴퓨터의 직렬 포트에서 나온 케이블을  $\mu$ C 모듈에 연결한다. 9핀 직렬 포트를 사용하는 경우, 직선형 9핀 암-수 케이블을 사용한다. 유일하게 사용 가능한 직렬 포트가 25핀 커넥터를 사용할 경우, 표준 25핀-9핀 어댑터가 필요하다. EV 키트 소프트웨어는 모뎀 상태 라인 (CTS, DSR, DCD)을 점검하여 정확한 포트가 선택되었는지 확인한다.
- 5) INSTALL.EXE 프로그램을 디스크에 실행하여 컴퓨터에 평가 소프트웨어를 설치한다. 프로그램 파일들이 복사되고 Windows 시작 메뉴에 아이콘이 만들어진다.
- 6) 전원을 켜다.
- 7) 시작 메뉴에서 아이콘을 클릭하여 MAX1499 프로그램을 시작한다.
- 8) 프로그램은 사용자에게  $\mu$ C 모듈을 연결하고 전원을 켜도록 지시한다. SW1을 ON 위치로 민다. 정확한 직렬 포트를 선택하고 확인을 클릭한다. 프로그램은 자동으로 소프트웨어를 모듈로 다운로드한다.

## 부품 공급업체

SUPPLIER	PHONE	FAX	WEBSITE
Kingbright Corporation	909-468-0500 (ext 126)	909-468-0505	www.kingbright.com
Taiyo Yuden	800-348-2496	847-925-0899	www.t-yuden.com
TDK	847-803-6100	847-390-4405	www.component.tdk.com

참고: 상기 부품 업체를 이용할 때에는 MAX1499를 사용 중임을 알려 준다.

# MAX1499 평가 키트/MAX1499 평가 시스템

- 9) AIN+와 AIN- 사이에 -2V~+2V 범위의 입력 신호를 적용한다. 화면에 나타난 내용을 확인한다.
- 10) 측정치의 그래프를 보려면 **View** 메뉴에서 **Graph**를 클릭한다.

## 소프트웨어 세부 설명

### 측정

평가 소프트웨어의 **Measurement** 탭은 디지털 전압계 (DVM)의 동작을 모사한다. 상태 비트는 초 당 약 1회 폴된다. **Data** 상태 비트가 1이면 ADC 결과 레지스터가 읽히고 **Analog Input Code** 로서 표시된다. MAX1499는 또한 자체 LED 디스플레이에도 결과를 표시한다.

이 EV 키트는 완벽한 DVM이 아니다. 추가 입력 눈금 및 보호 회로가 필요할 수 있다.

**Measurement** 탭이 활성화될 때마다 소프트웨어는 **spi/adc** 및 **seg\_sel** 제어 비트가 아직 소거되지 않았을 경우 이들 제어 비트를 0으로 소거하려고 시도한다.

### 산술 처리

평가 소프트웨어는 물리적인 시스템에서 발견되는 몇 가지 산술 기능을 구현한다. **Math** 탭이 활성화되면 소프트웨어는 **spi/adc** 제어 비트가 이미 설정되어 있지 않을 경우 이들 제어 비트를 설정하려고 시도한다. 소프트웨어는 또한 **seg\_sel** 제어 비트가 이미 소거되지 않았을 경우 이 제어 비트를 0으로 소거하려고 시도한다.

**Measurement** 또는 **Math** 탭이 활성화되어 있고 **spi/adc** 제어 비트가 1로 설정되어 있을 경우, 평가 소프트웨어는 표시하기 전에 ADC 결과를 인터셉트하여 새로운 LED 디스플레이 값을 계산한다. 산술 결과는 채널 0 데이터로 그래프로 표시되며, 원래의 ADC 결과도 채널 1 데이터로서 함께 표시된다.

**Type K Thermocouple** 기능은 적합한 냉 접합 연결과 함께 사용되어 K 형 서모커플의 측정된 Seebeck 전압을 섭씨 온도로 변환할 수 있다. **a0** 계수 230은 냉 접합 온도가 +23°C임을 나타낸다.

### 제어 레지스터

**Control Register** 탭은 모든 제어 레지스터 비트에 대한 액세스를 제공한다. 해당 콤보 상자를 드롭 다운한 다음 **write**를 클릭한다.

### 제한 레지스터, ADC 오프셋, ADC 결과, LED 디스플레이 및 최대값

**Results, Displays, Limits** 탭은 2의 보수 데이터 레지스터에 대한 액세스를 제공한다. 읽기 전용 레지스터인 **ADC RESULT1, ADC RESULT2** 및 **PEAK RESULT**를 제외한 각 레지스터에 **read** 버튼과 **write** 버튼이 있다.

**ADC RESULT1** 또는 **ADC RESULT2** 레지스터를 읽으면 **seg\_sel** 제어 레지스터 설정과 상관없이 LED 디스플레이가 자동으로 업데이트된다.

ADC OFFSET 레지스터에 쓰면 **off-set\_cal1** 제어 레지스터 설정과 상관없이 **ADC RESULT1** 과 **ADC RESULT2** 에 영향이 미친다.

### LED 세그먼트 레지스터

**LED Segments** 탭에서는 사용자가 개별 LED 세그먼트를 마우스로 클릭하여 켜고 끌 수 있다.

**LED Segments** 탭이 활성화되면 소프트웨어는 **seg\_sel** 제어 비트가 이미 설정되어 있지 않을 경우 이 제어 비트를 1로 설정하려고 시도한다.

**Write LED Text** 버튼은 텍스트 문자열을 대략 7세그먼트 문자로 변환한 다음, 문자 패턴을 LED 디스플레이에 기록한다.

### 그래프

평가 소프트웨어는 데이터를 그래프로 표시하는 두 가지 옵션이 있다. 최근 데이터 그래프는 **View** 메뉴를 선택한 다음 **Graph**를 선택하여 표시된다. 데이터는 시간 순서 플롯, 히스토그램 플롯, 또는 원시 숫자 표로 볼 수 있다. 데이터 표시의 크기와 타이밍을 제어하려면, 메인 윈도우의 **Collect Samples** 버튼을 클릭하여 샘플링 툴을 활성화한다.

샘플된 데이터는 콤마 또는 탭으로 구분되는 형식의 파일로 저장될 수 있다. 행 번호와 설명 헤더 행은 옵션이다.

채널 0은 원시 16비트 ADC 결과 데이터를 플롯한다. 산술 처리가 인에이블된 경우 채널 1은 LED 디스플레이 데이터를 플롯한다. 확장 해상도가 인에이블된 경우, 채널 2는 원시 20비트 ADC 결과 데이터를 플롯한다.

### 진단 창

진단 창은 평가 키트를 선적하기 전에 공장에서 테스트를 위해 사용된다. 이 창은 고객용이 아니다.

### 하드웨어 세부설명

테스트 중인 MAX1499 소자 (U1)은 통합 LED 디스플레이 드라이버가 있는 저전력 4.5디지트 ADC이다. MAX6062 (U5)는 온 보드 2.048V 기준 전압을 제공한다. 그림 7의 MAX1499 EV 키트 회로도 및 MAX1499 데이터 시트를 참조한다.

EV 키트는 MAX1659 고전류 5V 선형 레귤레이터 (U2)와 3V 로직 사용을 지원하기 위한 일련의 MAX1840/MAX1841 레벨 쉬프터 (U3 및 U4)를 포함한다.

# MAX1499 평가 키트/MAX1499 평가 시스템

표 1. 그래프 툴 버튼

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.
<b>Min</b>	Show minimum in tabular view.
<b>Max</b>	Show maximum in tabular view.
<b>Span</b>	Show span in tabular view. Span = maximum - minimum.
<b>N</b>	Show number of samples in tabular view.

TOOL	FUNCTION
<b>Sum(x)</b>	Show sum of the samples in tabular view.
<b>Sum(x*x)</b>	Show sum of the squares of the samples in tabular view.
<b>Mean</b>	Show arithmetic mean in tabular view: $\text{Mean} = \frac{\sum(x)}{n}$
<b>StdDev</b>	Show standard deviation in tabular view: $\text{Standard deviation} = \sqrt{\frac{n\sum(x^2) - \left(\sum x\right)^2}{(n-1)n}}$
<b>Rms</b>	Show root of the mean of the squares (RMS) in tabular view: $\text{RMS} = \sqrt{\frac{\sum(x^2)}{n}}$
<b>0</b>	Channel 0 enable (16-bit ADC result).
<b>1</b>	Channel 1 enable (math result).
<b>2</b>	Channel 2 enable (20-bit ADC result).

# MAX1499 평가 키트/MAX1499 평가 시스템

표 2. 점퍼 기능 표

JUMPER	SHUNT POSITION	FUNCTION
JU1	Closed*	LED displays are powered by U2.
	Open	VLED must be supplied by an external power source.
JU2	Closed*	VDISP = GND.
	Open	Apply VDISP voltage at VDISP pad.
JU3	Closed*	Banana jack AIN+ connects to AIN+ input pin.
	Open	Insert custom filtering between JU3 pins 1 and 2.
JU4	Closed*	Banana jack AIN- connects to AIN- input pin.
	Open	Insert custom filtering between JU4 pins 1 and 2.
JU5	Closed*	REF- = GND.
	Open	REF- must be provided by user.
JU6	Closed*	REF+ = 2.048V from U5, MAX6062.
	Open	REF+ must be provided by user.
JU7	Closed*	REF+ is bypassed by C6.
	Open	C6 is disconnected.
JU8	Closed*	GLLED return current flows to DVDD.
	Open	GLLED return current flows to external power source.
JU9	Closed	LED_EN = low; LED displays are blanked.
	Open*	LED_EN = high; LED displays are enabled.
JU10–JU14	1-2*	Display color = red.
	2-3	Display color = green.

\* 별표는 기본 구성을 나타냄.

표 3. 독립형 인터페이스 핀 기능

U1 PIN	MAX1499 FUNCTION	MAX1498 FUNCTION
8	CLK	INTREF
9	EOC	RANGE
10	CS	DPSET1
11	DIN	DPSET2
12	SCLK	PEAK
13	DOUT	HOLD
30	LOWBATT	DPON

### MAX1498 평가

MAX1499EVKIT은 MAX1498의 독립된 동작을 지원하지 않음,  $\mu$ C 인터페이스가 없기 때문에 평가 소프트웨어는 사용할 수 없다.

MAX1498은 MAX1499의 독립형 버전이다. MAX1498/MAX1499 데이터 시트를 참조한다. MAX1498ECJ의 무료 샘플을 요청한다.

- 1) MAX1499EVKIT은 68HC16MODULE과 분리되어야 한다.
- 2) 전원이 분리된 상태에서 U1을 MAX1498로 교체한다.
- 3) TB1 블록 단자에 DC 전원을 연결한다.
- 4) 전원을 켜다. LED 디스플레이는 측정 데이터를 표시하기 시작한다.

U1을 MAX1498로 교체하면 일부 핀 기능이 달라진다. 표 3을 참조한다.

### 코드 예

목록 1은 EV 키트 소프트웨어에서 필요한 변수 지정을 나타낸 것이다. 목록 2는 EV 키트 소프트웨어에서 사용되는 기능을 보여준다.

평가 대상: MAX1498/MAX1499

## MAX1499 평가 키트/MAX1499 평가 시스템

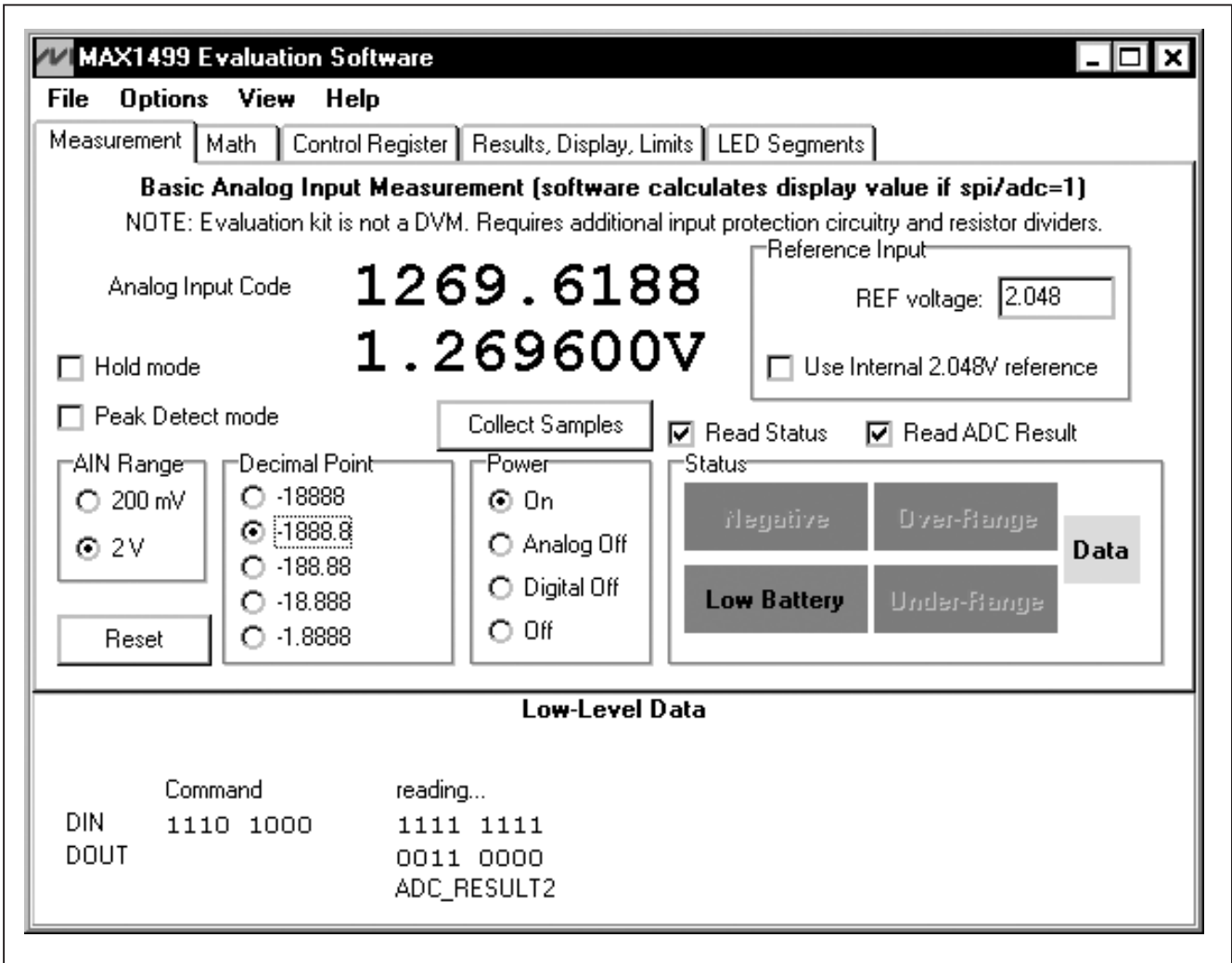


그림 1. MAX1499 평가 소프트웨어 — Measurement 탭

# MAX1499 평가 키트/MAX1499 평가 시스템

평가 대상: MAX1498/MAX1499

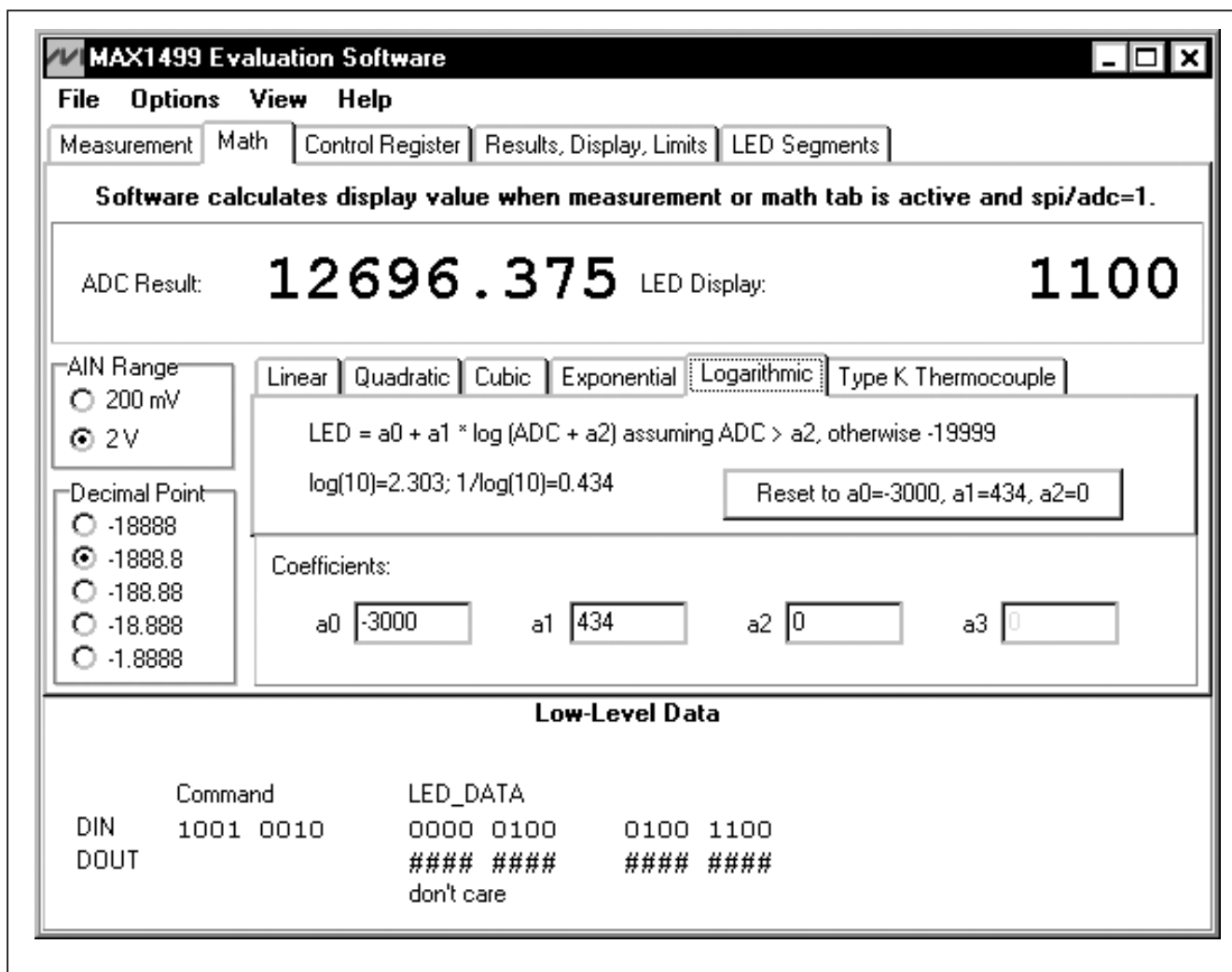


그림 2. MAX1499 평가 소프트웨어 — Math 탭

## MAX1499 평가 킷/MAX1499 평가 시스템

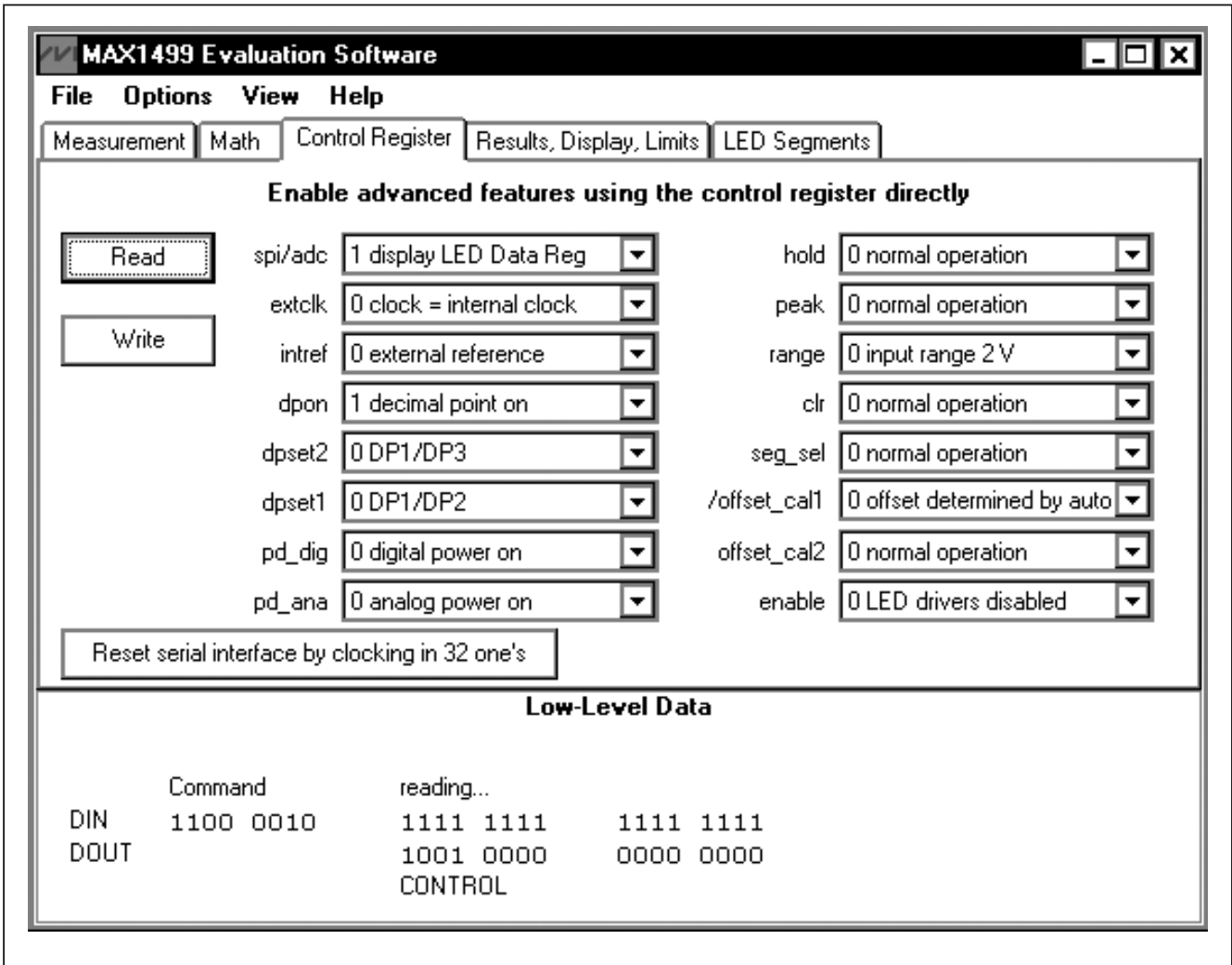


그림 3. MAX1499 평가 소프트웨어 — Control Register 탭

# MAX1499 평가 키트/MAX1499 평가 시스템

평가 대상: MAX1498/MAX1499

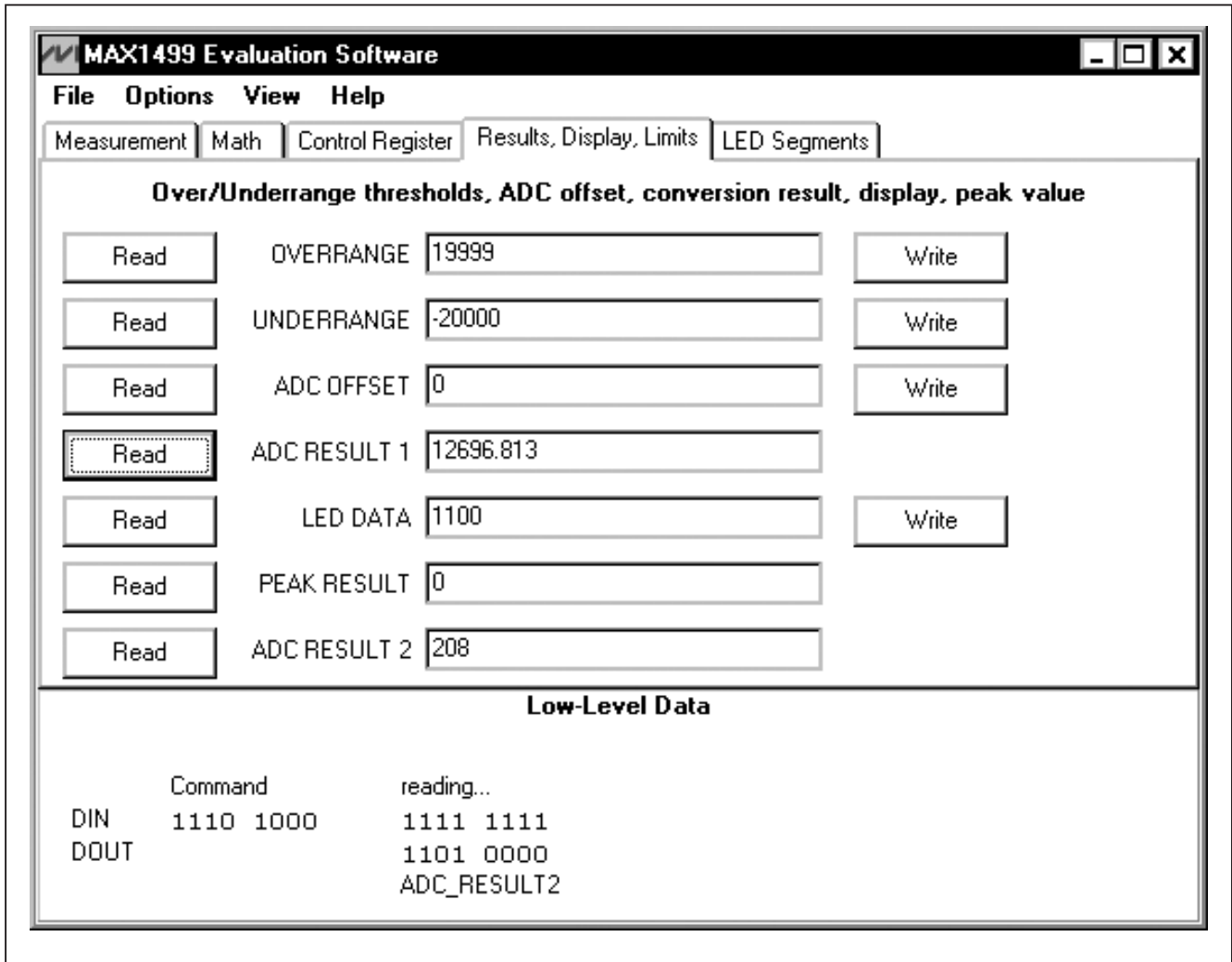


그림 4. MAX1499 평가 소프트웨어 — Results, Display, Limits 탭

# MAX1499 평가 키트/MAX1499 평가 시스템

평가 대상: MAX1498/MAX1499

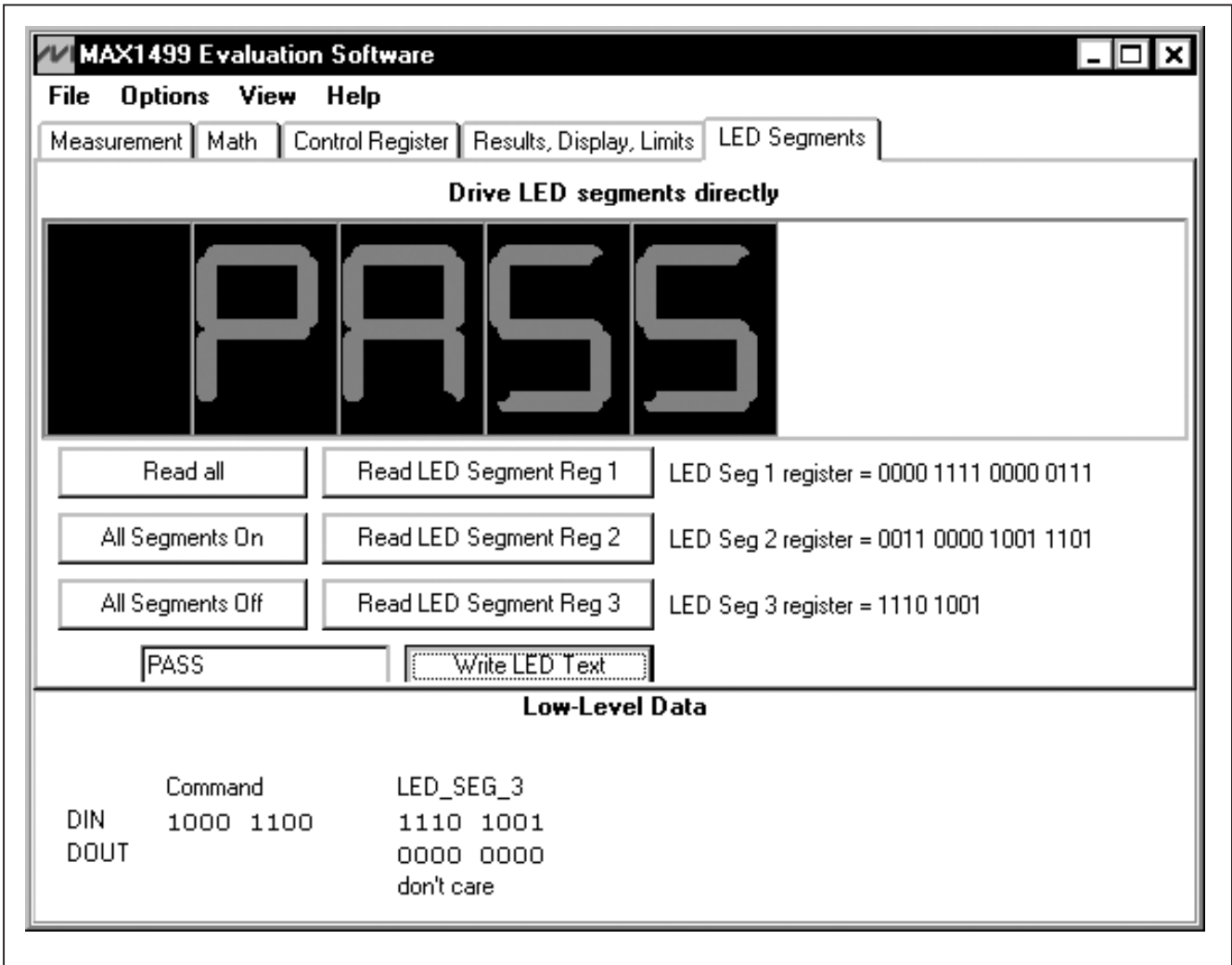


그림 5. MAX1499 평가 소프트웨어 — LED Segments 탭

# MAX1499 평가 키트/MAX1499 평가 시스템

평가 대상: MAX1498/MAX1499

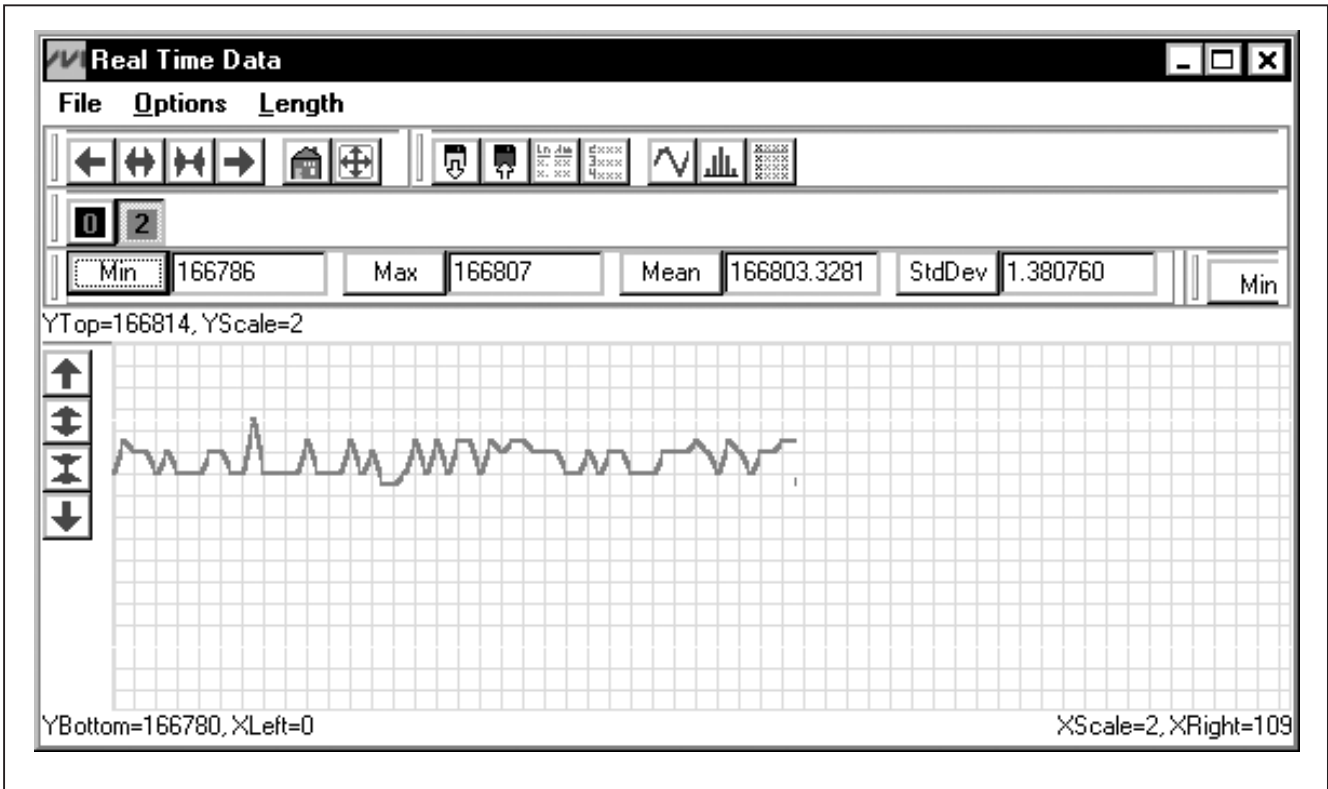


그림 6. MAX1499 평가 소프트웨어 — 그래프

# MAX1499 평가 키트/MAX1499 평가 시스템

평가 대상: MAX1498/MAX1499

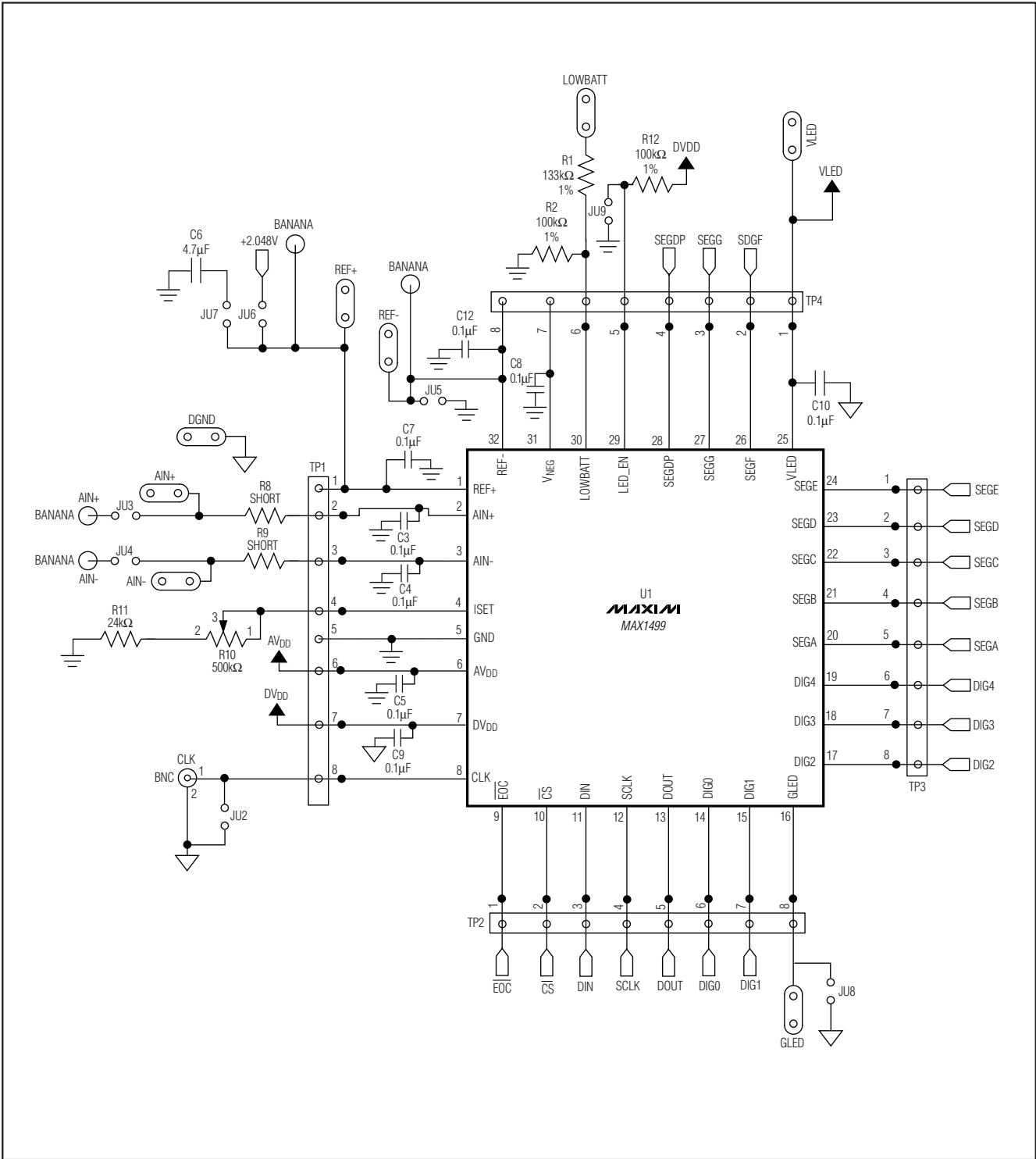


그림 7a. MAX1499 EV 키트 회로도 (시트 1/2)

# MAX1499 평가 키트/MAX1499 평가 시스템

평가 대상: MAX1498/MAX1499

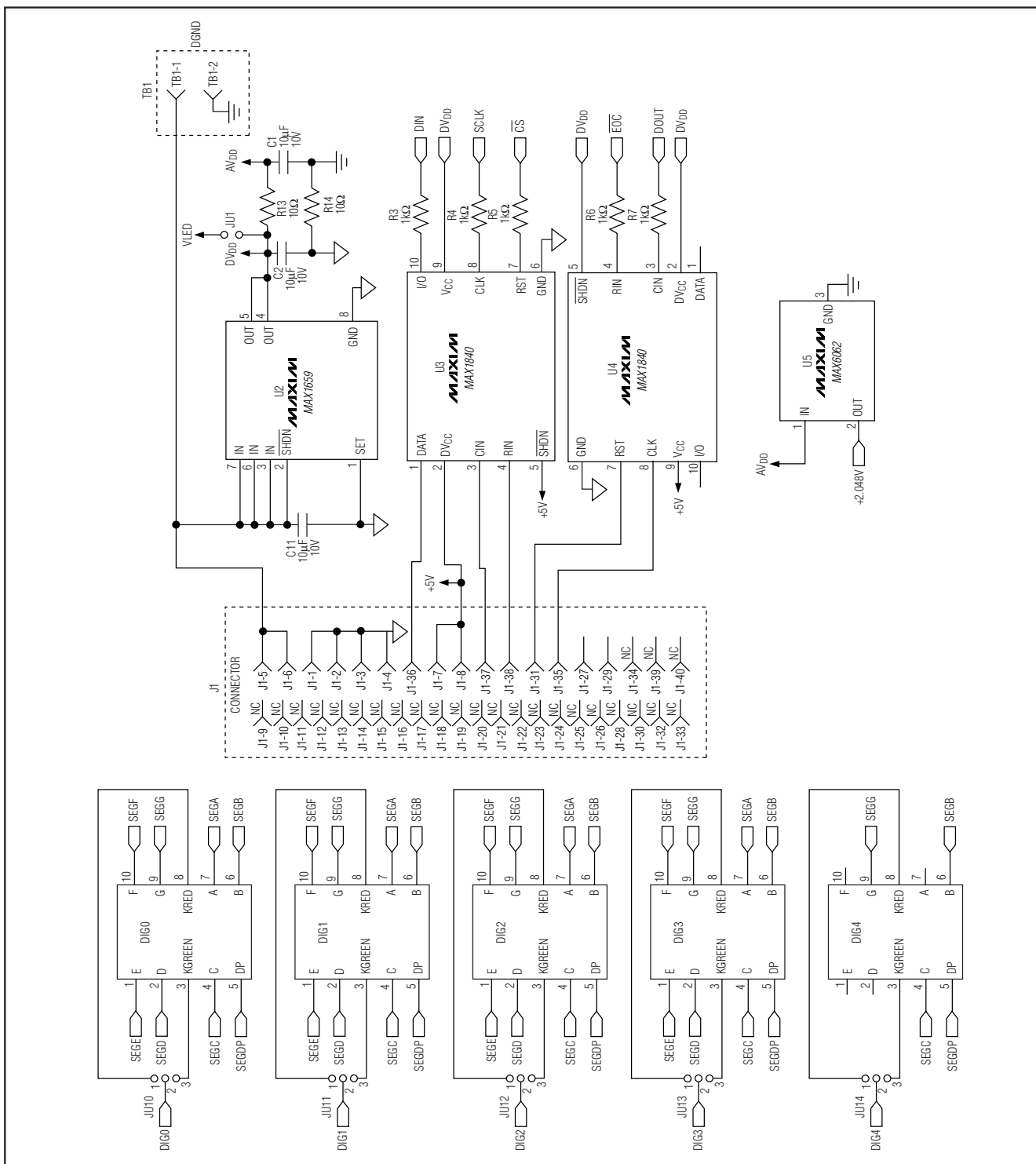


그림 7b. MAX1499 EV 키트 회로도 (시트 2/2)

# MAX1499 평가 키트/MAX1499 평가 시스템

평가 대상: MAX1498/MAX1499

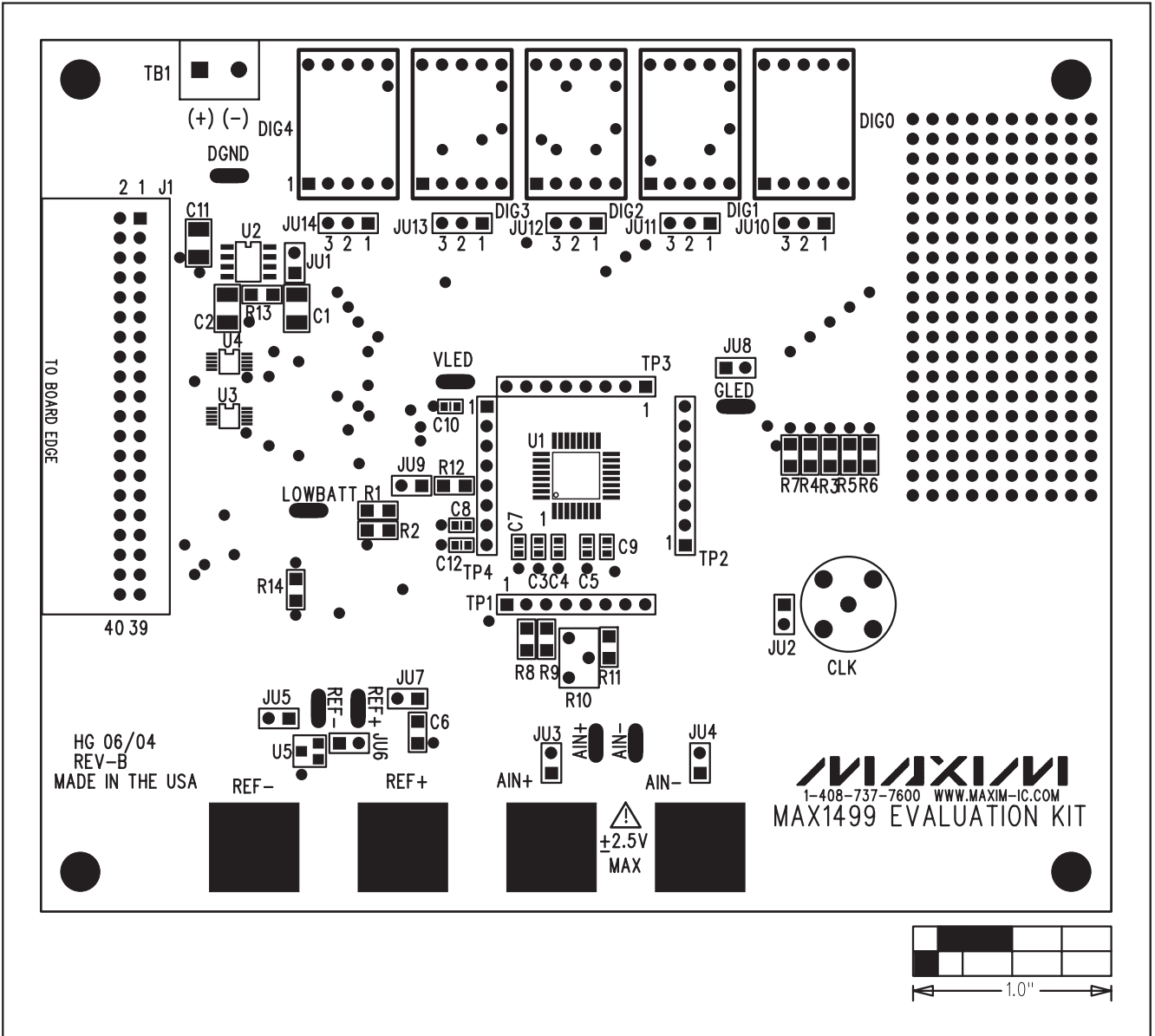


그림 8. MAX1499 EV 키트 부품 배치 가이드 — 부품 측

# MAX1499 평가 키트/MAX1499 평가 시스템

평가 대상: MAX1498/MAX1499

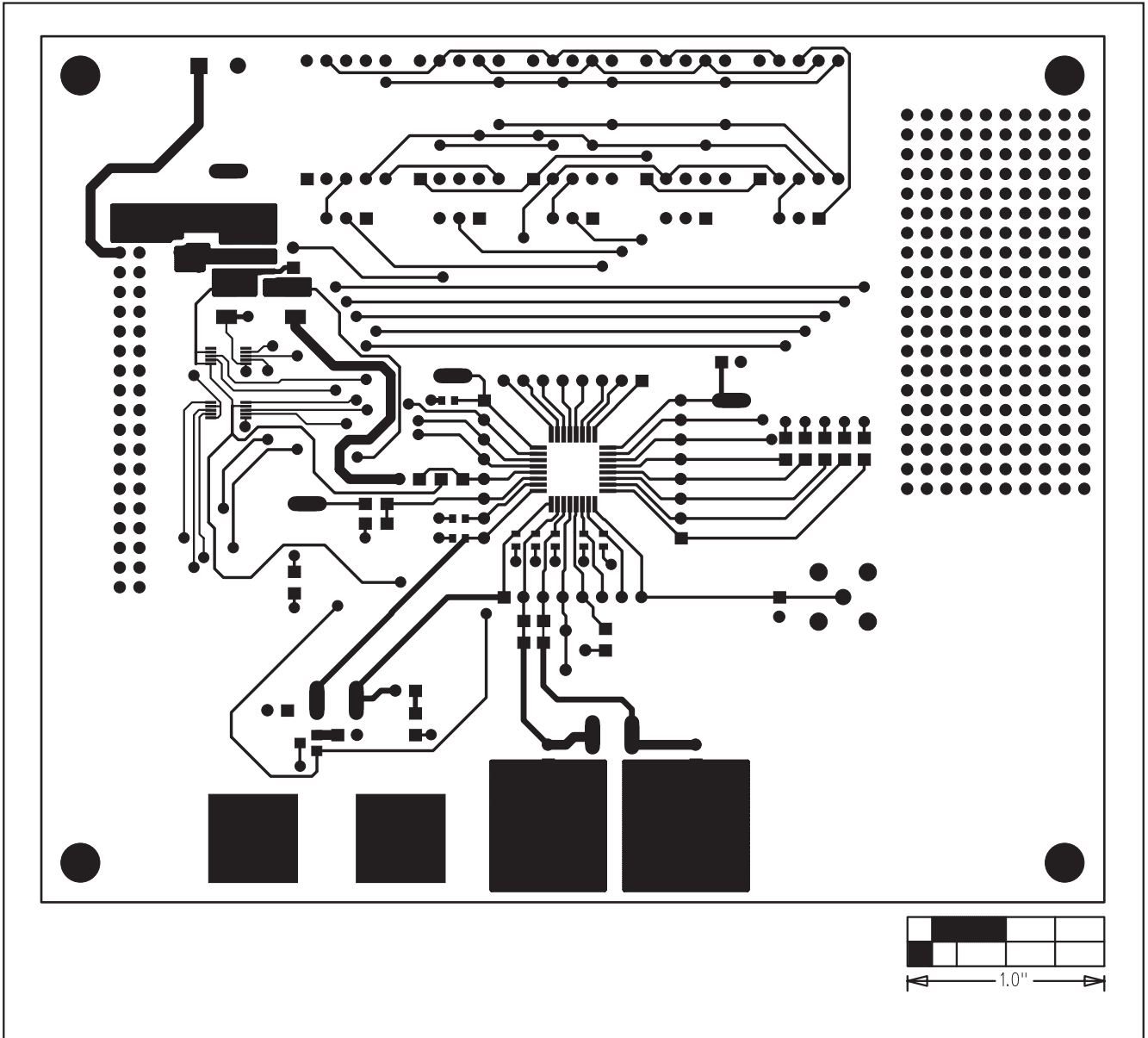


그림 9. MAX1499 EV 키트 PCB 레이아웃 — 부품 측

# MAX1499 평가 키트/MAX1499 평가 시스템

평가 대상: MAX1498/MAX1499

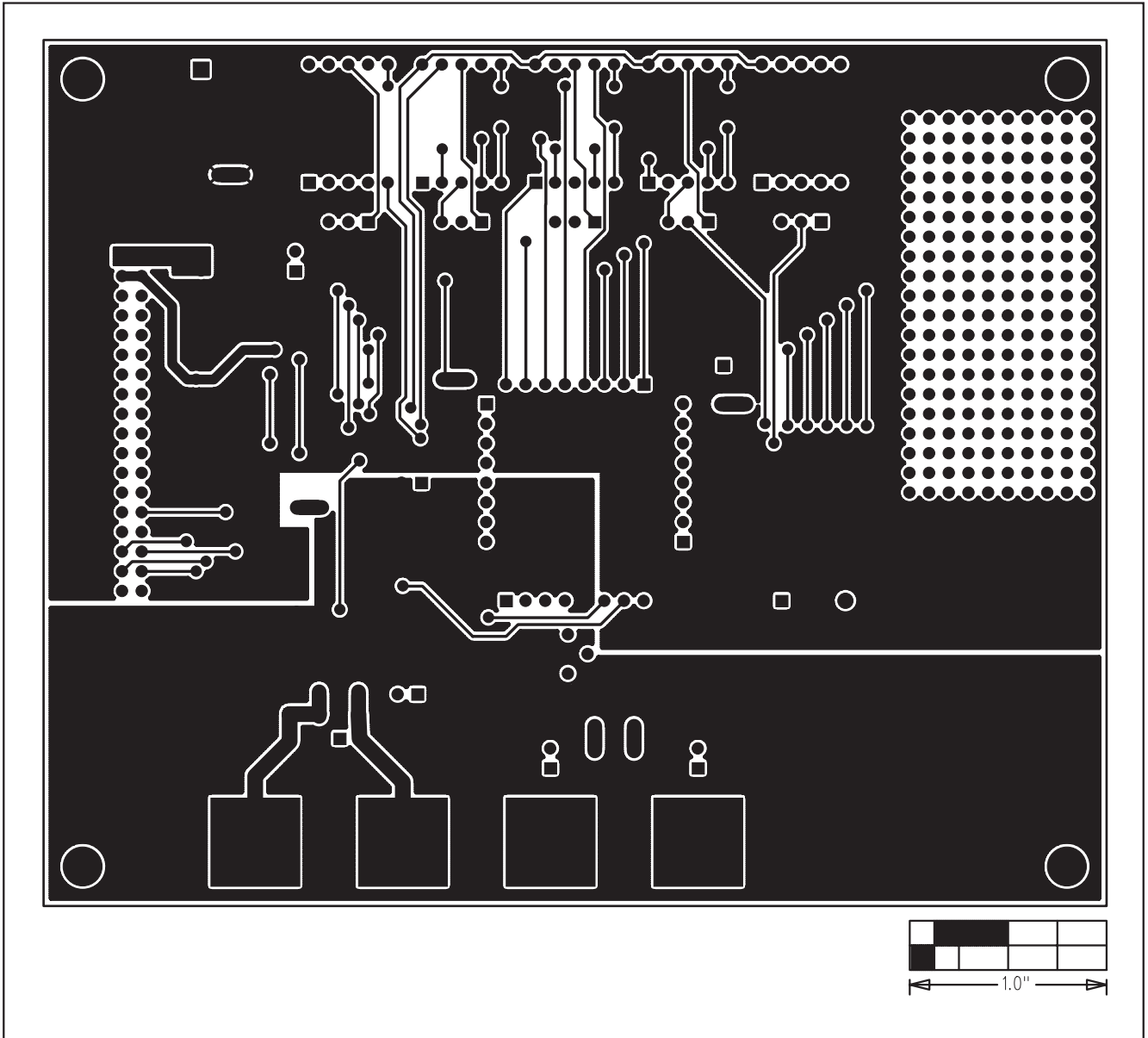


그림 10. MAX1499 EV 키트 PCB 레이아웃 — 납땜 측

# MAX1499 평가 키트/MAX1499 평가 시스템

평가 대상: MAX1498/MAX1499

```
// Drv1499.h
// MAX1499-specific driver.
// mku 01/07/2004
// (C) 2004 Maxim Integrated Products
// For use with Borland C++ Builder 3.0
//-----
// Revision history:
// 01/07/2004: modify drv 1 4 9 4 driver to become drv1499
// 12/04/2003: fix indentation
// 09/15/2003: add double Voltage(void)
// 09/12/2003: add SPI_Transfer_After_EOC()
// 09/09/2003: add class MAX1499 dependent on external SPI_Interface()
// 08/13/2003: preliminary draft of reusable code
//-----
#ifndef drv1499H
#define drv1499H
//-----

//-----
// The following interface protocols must be provided by
// the appropriate low-level interface code.
//

/* SPI interface:
** byte_count = transfer length
** mosi[] = array of master-out, slave-in data bytes
** miso_buf[] = receive buffer for master-in, slave-out data bytes
*/
extern bool SPI_Transfer(int byte_count,
    const unsigned __int8 mosi[], unsigned __int8 miso_buf[]);

/* SPI interface, with data transfer immediately after EOC is asserted:
** byte_count = transfer length
** mosi[] = array of master-out, slave-in data bytes
** miso_buf[] = receive buffer for master-in, slave-out data bytes
*/
extern bool SPI_Transfer_After_EOC(int byte_count,
    const unsigned __int8 mosi[], unsigned __int8 miso_buf[]);

//-----
// Define the bits in the COMMS register.
// START R/W RS4 RS3 RS2 RS1 RS0 0
#define MAX1499_COMMS_START 0x80
#define MAX1499_COMMS_RW_MASK 0x40
#define MAX1499_COMMS_RW_WRITE 0x00
#define MAX1499_COMMS_RW_READ 0x40
#define MAX1499_COMMS_RS_MASK 0x3E
#define MAX1499_COMMS_RS_00000 0x00
#define MAX1499_COMMS_RS_STATUS 0x00
#define MAX1499_COMMS_RS_00001 0x02
#define MAX1499_COMMS_RS_CONTROL 0x02
#define MAX1499_COMMS_RS_00010 0x04
#define MAX1499_COMMS_RS_OVERRANGE 0x04
#define MAX1499_COMMS_RS_00011 0x06
#define MAX1499_COMMS_RS_UNDERRANGE 0x06
#define MAX1499_COMMS_RS_00100 0x08
#define MAX1499_COMMS_RS_LED_SEG_1 0x08
#define MAX1499_COMMS_RS_00101 0x0A
#define MAX1499_COMMS_RS_LED_SEG_2 0x0A
#define MAX1499_COMMS_RS_00110 0x0C
#define MAX1499_COMMS_RS_LED_SEG_3 0x0C
#define MAX1499_COMMS_RS_00111 0x0E
#define MAX1499_COMMS_RS_ADC_OFFSET 0x0E
#define MAX1499_COMMS_RS_01000 0x10
#define MAX1499_COMMS_RS_ADC_RESULT1 0x10
#define MAX1499_COMMS_RS_01001 0x12
#define MAX1499_COMMS_RS_LED_DATA 0x12
#define MAX1499_COMMS_RS_01010 0x14
#define MAX1499_COMMS_RS_PEAK 0x14
#define MAX1499_COMMS_RS_10100 0x28
#define MAX1499_COMMS_RS_ADC_RESULT2 0x28

//-----
// Define the bits in the STATUS register.
// POL OVR_RNG UNDR_RNG LOW_BATT ADD(data available) 0 0 0
```

목록 1 (시트 1/4)

# MAX1499 평가 키트/MAX1499 평가 시스템

평가 대상: MAX1498/MAX1499

```

#define MAX1499_STATUS_POL_MASK          0x80
#define MAX1499_STATUS_POL_POSITIVE     0x00
#define MAX1499_STATUS_POL_NEGATIVE     0x80
#define MAX1499_STATUS_OVER_RANGE       0x40
#define MAX1499_STATUS_UNDER_RANGE      0x20
#define MAX1499_STATUS_LOW_BATTERY      0x10
#define MAX1499_STATUS_DATA_READY       0x08

//-----
// Define the bits in the CONTROL register.
// SPI_ADC EXTCLK INTREF DPON DPSET1 PD_DIG PD_ANA
// HOLD PEAK RANGE CLR LED OFFSET_CAL1 OFFSET_CAL2 ENABLE
#define MAX1499_CONTROL_SPI_ADC          0x8000
#define MAX1499_CONTROL_EXTCLK           0x4000
#define MAX1499_CONTROL_INTREF           0x2000
#define MAX1499_CONTROL_DPMASK          0x1C00
#define MAX1499_CONTROL_DPON             0x1000
#define MAX1499_CONTROL_DPSET2           0x0800
#define MAX1499_CONTROL_DPSET1           0x0400
// (DPSET2 is the LSB and DPSET1 is the MSB)
#define MAX1499_CONTROL_DP1ON            0x1000 /* -1888.8 */
#define MAX1499_CONTROL_DP2ON            0x1800 /* -188.88 */
#define MAX1499_CONTROL_DP3ON            0x1400 /* -18.888 */
#define MAX1499_CONTROL_DP4ON            0x1C00 /* -1.8888 */
#define MAX1499_CONTROL_PD_DIG           0x0200
#define MAX1499_CONTROL_PD_ANA           0x0100
#define MAX1499_CONTROL_PD_ALL           0x0300
#define MAX1499_CONTROL_HOLD              0x0080
#define MAX1499_CONTROL_PEAK              0x0040
#define MAX1499_CONTROL_RANGE_200mV      0x0020
#define MAX1499_CONTROL_CLR                0x0010
#define MAX1499_CONTROL_SEG_SEL           0x0008
#define MAX1499_CONTROL_OFFSET_CAL1       0x0004
#define MAX1499_CONTROL_OFFSET_CAL2       0x0002
#define MAX1499_CONTROL_ENABLE            0x0001

//-----
// Define the bits in the LED SEGMENT 1 register.
// A2 G2 D2 F2 E2 DP2 0 B1
// C1 A1 G1 D1 F1 E1 DP1 0
//
#define MAX1499_LED_SEG1_A1                0x8000
#define MAX1499_LED_SEG1_G1                0x4000
#define MAX1499_LED_SEG1_D1                0x2000
#define MAX1499_LED_SEG1_F1                0x1000
#define MAX1499_LED_SEG1_E1                0x0800
#define MAX1499_LED_SEG1_DP2               0x0400
#define MAX1499_LED_SEG1_B0                0x0100
#define MAX1499_LED_SEG1_C0                0x0080
#define MAX1499_LED_SEG1_A0                0x0040
#define MAX1499_LED_SEG1_G0                0x0020
#define MAX1499_LED_SEG1_D0                0x0010
#define MAX1499_LED_SEG1_F0                0x0008
#define MAX1499_LED_SEG1_E0                0x0004
#define MAX1499_LED_SEG1_DP1               0x0002

//-----
// Define the bits in the LED SEGMENT 2 register.
// F4 E4 DP4 G4 B3 C3 A3 G3
// D3 F3 E3 DP3 0 B2 C2 0
//
#define MAX1499_LED_SEG2_F3                0x8000
#define MAX1499_LED_SEG2_E3                0x4000
#define MAX1499_LED_SEG2_DP4               0x2000
#define MAX1499_LED_SEG2_MINUS              0x1000 /* TODO: is this documented? minus sign
*/
#define MAX1499_LED_SEG2_B2                0x0800
#define MAX1499_LED_SEG2_C2                0x0400
#define MAX1499_LED_SEG2_A2                0x0200
#define MAX1499_LED_SEG2_G2                0x0100
#define MAX1499_LED_SEG2_D2                0x0080
#define MAX1499_LED_SEG2_F2                0x0040
#define MAX1499_LED_SEG2_E2                0x0020

```

목록 1 (시트 2/4)

# MAX1499 평가 키트/MAX1499 평가 시스템

평가 대상: MAX1498/MAX1499

```

#define MAX1499_LED_SEG2_DP3      0x0010
#define MAX1499_LED_SEG2_B1      0x0004
#define MAX1499_LED_SEG2_C1      0x0002

//-----
// Define the bits in the LED SEGMENT 3 register.
// 0 0 BC5 B4 C4 A4 G4 D4
//
#define MAX1499_LED_SEG3_BC_      0x20
#define MAX1499_LED_SEG3_B3_      0x10
#define MAX1499_LED_SEG3_C3_      0x08
#define MAX1499_LED_SEG3_A3_      0x04
#define MAX1499_LED_SEG3_G3_      0x02
#define MAX1499_LED_SEG3_D3_      0x01

//-----
class MAX1499
{
public:
    MAX1499(void);

    // Enumerated type describing the register select bits.
    enum RegisterSelect_t {
        RS_STATUS      = MAX1499_COMMS_RS_STATUS,
        RS_CONTROL      = MAX1499_COMMS_RS_CONTROL,
        RS_OVERRANGE    = MAX1499_COMMS_RS_OVERRANGE,
        RS_UNDERRANGE   = MAX1499_COMMS_RS_UNDERRANGE,
        RS_LED_SEG_1    = MAX1499_COMMS_RS_LED_SEG_1,
        RS_LED_SEG_2    = MAX1499_COMMS_RS_LED_SEG_2,
        RS_LED_SEG_3    = MAX1499_COMMS_RS_LED_SEG_3,
        RS_ADC_OFFSET    = MAX1499_COMMS_RS_ADC_OFFSET,
        RS_ADC_RESULT1   = MAX1499_COMMS_RS_ADC_RESULT1,
        RS_LED_DATA      = MAX1499_COMMS_RS_LED_DATA,
        RS_PEAK          = MAX1499_COMMS_RS_PEAK,
        RS_ADC_RESULT2   = MAX1499_COMMS_RS_ADC_RESULT2
    };

    // Reference voltage
    //
    double vref;

    //-----
    // Status Register
    // POL OVR_RNG UNDR_RNG LOW_BATT ADD(data available) 0 0 0
    int STATUS_REG;
    //
    bool Read_STATUS(void);

    //-----
    // Control Register
    // SPI_ADC EXTCLK INTREF DPON DPSET2 DPSET1 PD_DIG PD_ANA
    // HOLD PEAK RANGE CLR LED OFFSET_CAL1 OFFSET_CAL2 ENABLE
    int CONTROL_REG;
    //
    bool Write_CONTROL(int data);
    bool Read_CONTROL(void);

    //-----
    // Data Registers
    int ADC_RESULT1;
    unsigned int ADC_RESULT2;
    //
    bool Read_ADC_RESULT1(void);
    bool Read_ADC_RESULT2(void);
    long int DATA_REG; // 16-bit or 24-bit result from A/D converter
    bool extended_resolution;
    long Read_DATA(void);
    double Voltage(void);

    //-----
    // Other registers, having 16-bit 2's complement data format
    bool Write_2s_complement(int reg, int data);
    int Read_2s_complement(int reg);

```

목록 1 (시트 3/4)

# MAX1499 평가 키트/MAX1499 평가 시스템

평가 대상: MAX1498/MAX1499

```
//-----  
// Other registers, having 8 bit data format  
bool Write_8bit_reg(int reg, int data);  
int Read_8bit_reg(int reg);  
  
};  
  
//-----  
#endif
```

# MAX1499 평가 키트/MAX1499 평가 시스템

```
// Drv1499.cpp
// MAX1499-specific driver.
// mku 09/15/2003
// (C) 2003 Maxim Integrated Products
// For use with Borland C++ Builder 3.0
//-----
// Revision history:
// 09/15/2003: add double Voltage(void)
// 09/09/2003: add class MAX1499 dependent on external SPI_Interface()
// 08/13/2003: preliminary draft of reusable code

#include "drv1499.h"
//-----
MAX1499::MAX1499(void)
{
    vref = 2.048;
    extended_resolution = false;
}
//-----
bool MAX1499::Read_STATUS(void)
{
    const unsigned __int8 mosi[] = {
        (unsigned __int8) (MAX1499_COMMS_START |
            MAX1499_COMMS_RW_READ | MAX1499_COMMS_RS_STATUS),
        (unsigned __int8) (0xFF)
    };
    unsigned __int8 miso_buf[sizeof(mosi)];
    bool result = SPI_Transfer(sizeof(mosi), mosi, miso_buf);
    if (result) {
        int data = miso_buf[1];
        STATUS_REG = data;          // remember the value we just received
    }
    return result;
}
//-----
bool MAX1499::Write_CONTROL(int data)
{
    data = data & 0xFFFF;          // validate the data
    const unsigned __int8 mosi[] = {
        (unsigned __int8) (MAX1499_COMMS_START |
            MAX1499_COMMS_RW_WRITE | MAX1499_COMMS_RS_CONTROL),
        (unsigned __int8) ((data >> 8) & 0xFF),
        (unsigned __int8) (data & 0xFF)
    };
    unsigned __int8 miso_buf[sizeof(mosi)];
    bool result = SPI_Transfer(sizeof(mosi), mosi, miso_buf);
    CONTROL_REG = data;          // remember the value we just wrote
    // The CLR bit is self-clearing, and should not be kept high.
    CONTROL_REG &=~ MAX1499_CONTROL_CLR;
    return result;
}
//-----
bool MAX1499::Read_CONTROL(void)
{
    const unsigned __int8 mosi[] = {
        (unsigned __int8) (MAX1499_COMMS_START |
            MAX1499_COMMS_RW_READ | MAX1499_COMMS_RS_CONTROL),
        (unsigned __int8) (0xFF),
        (unsigned __int8) (0xFF)
    };
    unsigned __int8 miso_buf[sizeof(mosi)];
    bool result = SPI_Transfer(sizeof(mosi), mosi, miso_buf);
    if (result) {
        int data = miso_buf[1] * 0x100 + miso_buf[2];
        CONTROL_REG = data;          // remember the value we just wrote
    }
    return result;
}
//-----
bool MAX1499::Read_ADC_RESULT1(void)
{
    const unsigned __int8 mosi[] = {
```

## MAX1499 평가 키트/MAX1499 평가 시스템

```

        (unsigned __int8) (MAX1499_COMMS_START |
        MAX1499_COMMS_RW_READ | MAX1499_COMMS_RS_ADC_RESULT1),
        (unsigned __int8) (0xFF),
        (unsigned __int8) (0xFF)
    };
    unsigned __int8 miso_buf[sizeof(mosi)];
    bool result = SPI_Transfer_After_EOC(sizeof(mosi), mosi, miso_buf);
    if (result) {
        ADC_RESULT1 = (miso_buf[1] * 0x100L) + miso_buf[2];
        long data = (miso_buf[1] * 0x100L) + miso_buf[2];
        if (data >= 32768) {
            data -= 65536;
        }
        DATA_REG = data;          // remember the value we just received
    }
    return result;
}
//-----
bool MAX1499::Read_ADC_RESULT2(void)
{
    const unsigned __int8 mosi[] = {
        (unsigned __int8) (MAX1499_COMMS_START |
        MAX1499_COMMS_RW_READ | MAX1499_COMMS_RS_ADC_RESULT2),
        (unsigned __int8) (0xFF)
    };
    unsigned __int8 miso_buf[sizeof(mosi)];
    bool result = SPI_Transfer(sizeof(mosi), mosi, miso_buf);
    if (result) {
        ADC_RESULT2 = miso_buf[1];
        long data_24 = ((long)ADC_RESULT1 * 0x100L) + ADC_RESULT2;
        DATA_REG = data_24;
    }
    return result;
}
//-----
long MAX1499::Read_DATA(void)
{
    // Read the DATA register
    const unsigned __int8 mosi[] = {
        (unsigned __int8) (MAX1499_COMMS_START |
        MAX1499_COMMS_RW_READ | MAX1499_COMMS_RS_ADC_RESULT1),
        (unsigned __int8) (0xFF),
        (unsigned __int8) (0xFF)
    };
    unsigned __int8 miso_buf[sizeof(mosi)];
    if (SPI_Transfer_After_EOC(sizeof(mosi), mosi, miso_buf) == false) {
        return 0; // failure
    }
    ADC_RESULT1 = (miso_buf[1] * 0x100L) + miso_buf[2];
    long data = (miso_buf[1] * 0x100L) + miso_buf[2];
    if (data >= 32768) {
        data -= 65536;
    }
    DATA_REG = data;          // remember the value we just received
    if (extended_resolution) {
        // Read the ADC_RESULT2 register
        const unsigned __int8 mosi[] = {
            (unsigned __int8) (MAX1499_COMMS_START |
            MAX1499_COMMS_RW_READ | MAX1499_COMMS_RS_ADC_RESULT2),
            (unsigned __int8) (0xFF)
        };
    };
    unsigned __int8 miso_buf[sizeof(mosi)];
    if (SPI_Transfer(sizeof(mosi), mosi, miso_buf) == false) {
        return 0; // failure
    }
    ADC_RESULT2 = miso_buf[1];
    long data_24 = ((long)ADC_RESULT1 * 0x100L) + ADC_RESULT2;
    double data_16 = data_24 / 256.0;
    if (data_16 >= 32768) {
        data_16 = data_16 - 65536;
    }
    DATA_REG = data_24;
}

```

목록 2 (시트 2/4)

# MAX1499 평가 키트/MAX1499 평가 시스템

```
    }
    return DATA_REG;
}
//-----
double MAX1499::Voltage(void)
{
    if ((CONTROL_REG & MAX1499_CONTROL_RANGE_200mV) == 0) {
        // Input range 2V
        return DATA_REG * (vref / 2.048) * 10e-6 * 10;
    } else {
        // Input range 200mV
        return DATA_REG * (vref / 2.048) * 10e-6;
    }
}
//-----
bool MAX1499::Write_2s_complement(int reg, int data)
{
    // Write one of the 2's complement registers
    reg = (reg & MAX1499_COMMS_RS_MASK);
    data = data & 0xFFFF; // validate the data

    const unsigned __int8 mosi[] = {
        (unsigned __int8)(MAX1499_COMMS_START | MAX1499_COMMS_RW_WRITE | reg),
        (unsigned __int8)((data >> 8) & 0xFF),
        (unsigned __int8)(data & 0xFF)
    };
    unsigned __int8 miso_buf[sizeof(mosi)];
    bool result = SPI_Transfer(sizeof(mosi), mosi, miso_buf);
    return result;
}
//-----
int MAX1499::Read_2s_complement(int reg)
{
    // Read one of the 2's complement registers
    reg = (reg & MAX1499_COMMS_RS_MASK);

    const unsigned __int8 mosi[] = {
        (unsigned __int8)(MAX1499_COMMS_START | MAX1499_COMMS_RW_READ | reg),
        (unsigned __int8)(0xFF),
        (unsigned __int8)(0xFF)
    };
    unsigned __int8 miso_buf[sizeof(mosi)];
    bool result = SPI_Transfer(sizeof(mosi), mosi, miso_buf);
    if (result == false) {
        return 0; // failure
    }
    int data = miso_buf[1] * 0x100 + miso_buf[2];
    if (data >= 32768) {
        data -= 65536;
    }
    if (data >= 32768) {
        data -= 65536;
    }
    return data;
}
//-----
bool MAX1499::Write_8bit_reg(int reg, int data)
{
    // Write one of the 8 bit registers
    reg = (reg & MAX1499_COMMS_RS_MASK);
    const unsigned __int8 mosi[] = {
        (unsigned __int8)(MAX1499_COMMS_START | MAX1499_COMMS_RW_WRITE | reg),
        (unsigned __int8)(data & 0xFF)
    };
    unsigned __int8 miso_buf[sizeof(mosi)];
    bool result = SPI_Transfer(sizeof(mosi), mosi, miso_buf);
    return result;
}
//-----
int MAX1499::Read_8bit_reg(int reg)
{
    // Read one of the 8 bit registers
    reg = (reg & MAX1499_COMMS_RS_MASK);
```

## MAX1499 평가 키트/MAX1499 평가 시스템

```
const unsigned __int8 mosi[] = {
    (unsigned __int8) (MAX1499_COMMS_START | MAX1499_COMMS_RW_READ | reg),
    (unsigned __int8) (0xFF)
};
unsigned __int8 miso_buf[sizeof(mosi)];
bool result = SPI_Transfer(sizeof(mosi), mosi, miso_buf);
if (result == false) {
    return 0; // failure
}
int data = miso_buf[1];
return data;
}
//-----
```

목록 2 (시트 4/4)

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.

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