Keywords: Temperature sensor, LED driver, micro controller

APPLICATION NOTE 6647

# HOW TO BUILD A HIGH-RESOLUTION TEMPERATURE MONITORING SYSTEM

*Abstract: This application note introduces a temperature monitoring system with LED display based on the MAXQ2000 microcontroller, the MAX31875 high-resolution local temperature sensor, and the MAX7219 eight-digit LED display drivers.*

## Introduction

The MAX31875 is a ±2°C-accurate local temperature sensor with an $I^2C$/SMBus interface. The device is available in a 4-bump, wafer-level package, uses <10µA average supply current, and can operate over -20°C to +150°C. With its tiny package, low power consumption, and excellent accuracy, the MAX31875 can be used in various temperature-measurement applications, such as portable equipment, handheld electronics, and industrial equipment. This application note presents an idea for a temperature-monitoring system. The system is based on the MAXQ2000 microcontroller, MAX31875 high-resolution local temperature sensor, and MAX7219 8-digit LED display drivers. The MAXQ2000 reads temperatures through the $I^2C$ interface of the MAX31875, and an LED display is accomplished using the MAX7219 LED driver through a serial peripheral interface (SPI).

## Overview

The temperature-monitoring system uses the MAXQ2000 as a controller. It works as an interface between the temperature sensor and LED display. The MAXQ2000 receives temperature information through the $I^2C$ interface from the MAX31875 and creates an SPI to the MAX7219, which drives the LED display. The $I^2C$ interfaces that the MAXQ2000 offers run at 400kHz or 100kHz, and SPI commands can be sent at up to 8MHz. The MAXQ2000 keeps checking environment temperature through the $I^2C$ interface from the MAX31875. An eight-digit LED displays the MAX31875's local temperature in Celsius. The MAX7219 is a compact, serial input/output common-cathode display driver. Simply connect it to the eight-digit, seven-segment numeric LED displays and send temperature information through the SPI from the microcontroller.

## Hardware Setup

The test program runs on the Maxim Command Module (CMAXQUSB). It works as an interface to receive commands from the PC and creates an SPI or SMBus/$I^2C$-compatible interface to send commands to the MAX31875 and MAX7219. An on-board level translator converts the MAXQ2000 2.5V logic signals to the external VDD logic level. The VDD logic level is jumper-selectable for 2.5V, 3.3V, or 5V, provided by an on board ultra low-noise LDO MAX8511. The MAX3373 is a bidirectional level translator specially designed for the $I^2C$ bus. Optional user-switchable SCL/SDA pullup resistors are provided on-board. The MAX31875 EV kit with VDD, GND, SCL, and SDA test point header is connected to connector P7 on the CMAXQUSB. The MAX7219 is used to drive the 8-digit, 7-segment LED display. The MAX7219's SPI is connected to connector P8 on the CMAXQUSB. The firmware is loaded into the MAXQ2000 through connector P5. The hardware configuration is shown as in Figure 1. An actual setup is shown in Figure 2.

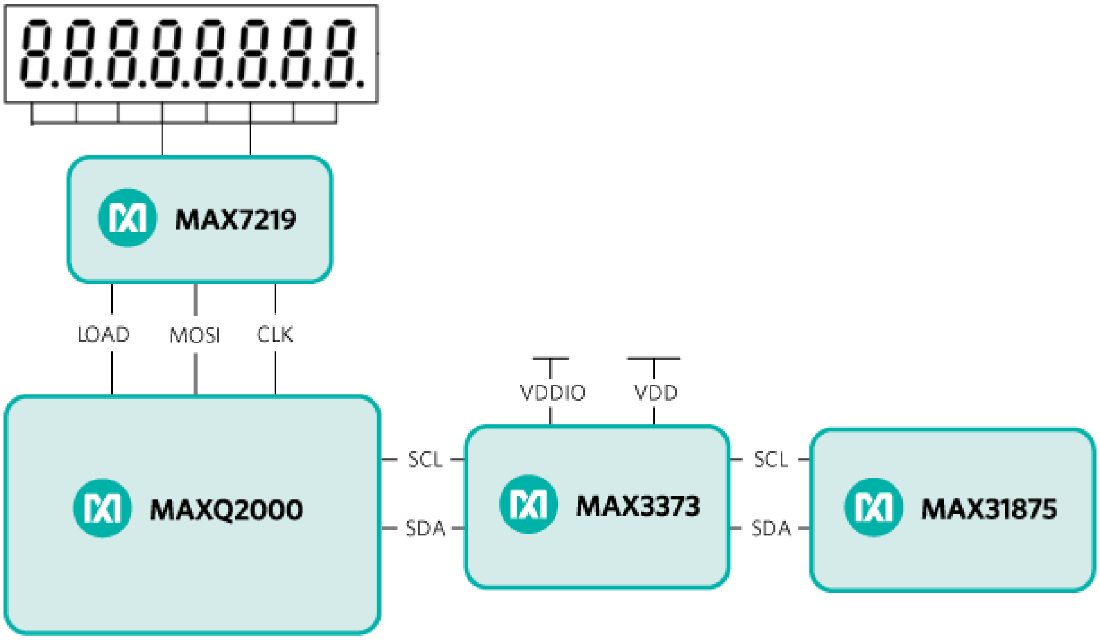

*Figure 1. Temperature monitoring system.*

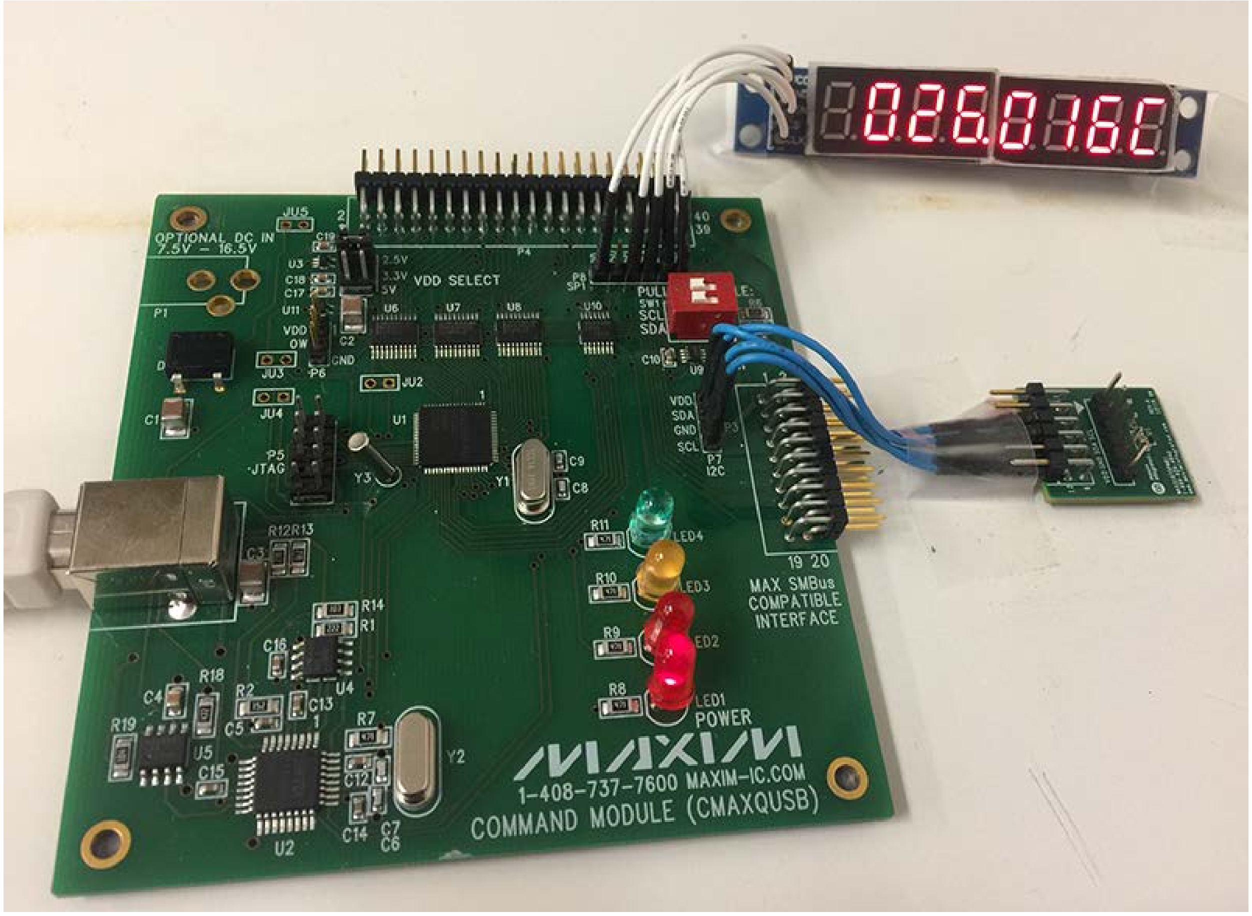
026.016C
VDD SELECT
MAX SMBus COMPATIBLE INTERFACE
1-408-737-7600 MAXIM-IC.COM
COMMAND MODULE (CMAXQUSB)

*Figure 2. Temperature monitoring setup.*

## Software Function

The temperature-monitoring program consists of three main functions: getting temperature readings from the MAX31875 through the I$^2$C interface, sending SPI commands to the MAX7219, and outputting results on the LED display.

The CMAXQUSB module offers a "bit banged" I$^2$C using the MAXQ2000's GPIO pins. Microcontrollers from the MAXQ family are well-suited for such bit-banging applications because of their high-speed, flexible GPIO modules, and separate I/O supply voltage.

Two GPIO pins are selected for use as SCL and SDA. Include the MAXQ2000 I$^2$C library files (maxq2000_i2c.h and maxq2000_i2c.c) in the project directory and configure the desired I/O for the I$^2$C interface.

The code to choose the GPIO pin for the SCL and SDA signals is as follows:

```
 #define SCL_ENABLE PD6 |= 0x01;
#define SCL_DISABLE PD6 &= 0xFE;
#define SDA_ENABLE PD6 |= 0x02;
#define SDA_DISABLE PD6 &= 0xFD;

#define SCL_HIGH  PO6 |= 0x01;
#define SCL_LOW  PO6 &= 0xFE;
#define SDA_HIGH  PO6 |= 0x02;
#define SDA_LOW  PO6 &= 0xFD;
```

To read the temperature from the MAX31875, send a read command to IC address 0x90 with temperature register address 0x00. The MAX31875's temperature register is displayed in a two's complement format; MSB is sign bit.

The code to get temperature is shown below. TxByte in the maxq2000_i2c.c file is used to transmit an 8-bit byte on the I$^2$C bus, and I$^2$C_RxByte is used to receive an 8-bit byte from the slave. To read temperature from float GetTEMP():

```
 {
int temp;
float number=0;
int temp_reg;
// first set the address
I2C_Start();
I2C_TxByte(0x90);
if (!checkForValidAck()) return -1;
I2C_TxByte(0x00);
if (!checkForValidAck()) return -1;
I2C_Stop();
// now read the data
I2C_Start();
I2C_TxByte(0x91);
if (!checkForValidAck()) return -1;
temp=I2C_RxByte();
I2C_TxBit(0);
temp=temp<<8;
temp|=I2C_RxByte();
I2C_TxBit(1);
I2C_Stop();
temp_reg=temp&0X7FFF;//get unsigned bit
temp=temp>>15;//get sign
if(temp)
sign=1;
else
sign=0;
number=(temp_reg-sign*32768)*0.0625/8 ;
return number;
}
```

The temperature reading received from the MAX31875 is sent to the MAX7219 using the SPI and the MAX7219's register. spi_transmit() performs an SPI transfer as shown below.

```
 unsigned int spi_transmit(const unsigned int ch)
{
unsigned int spib;

while(SPICN_bit.STBY);       /* Wait until the character can be sent. */
SPICN_bit.SPIC = 0;         /* In case it is already set, clear the */
         /* transfer complete flag. */
SPIB = ch;         /* Send the data. */
while(!SPICN_bit.SPIC);         /* Wait for the transfer to complete. */
spib = SPIB;         /* Get the data received. */
SPICN_bit.SPIC = 0;         /* Clear the tansfer complete flag. */
return spib;
}
```

The MAX7219 drives an 8-digit, 7-segment LED display. The MAX7219 offers a BCD code-B decoder. However, to display the Celsius sign, we use no-decode mode. A digit pattern is assigned to the MAX7219's register data, so, each LED can display the corresponsive number.

```
 char digit_pattern[] =
{ 0x7E, // 0
0x30, // 1
0x6D, // 2
0x79, // 3
0x33, // 4
0x5b, // 5
0x5F, // 6
0x70, // 7
0x7F, // 8
0x7B, // 9
0x77, // A
0x1F, // b
0x4E, // C
0x3d, // d
0x4f, // E
0x47 // F
};
max7219_writeregister(MAX7219REG_Degree,digit_pattern[12],LENGTH_16);// Display Celsius sign
```

## Test Result

Put this temperature monitor inside the Fluke® oil bath calibrator and collect temperature readings from -20°C to

+150°C. The accuracy of the monitor over temperature (Figure 3) is below 1°C.

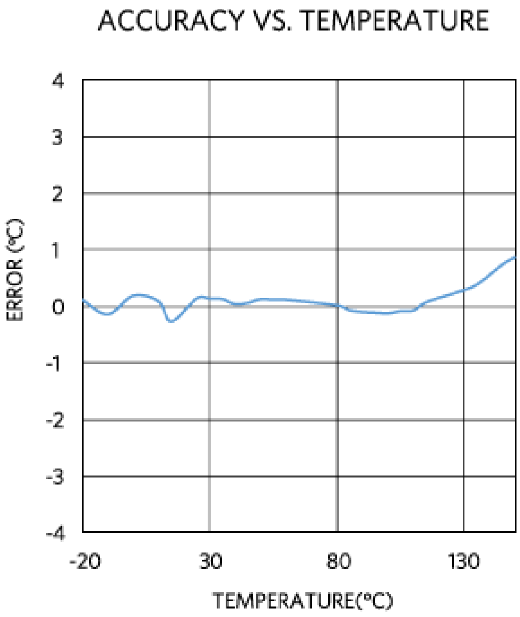

*Figure 3. Accuracy vs. temperature.*

## Conclusion

The MAX31875 is an accurate local temperature sensor. With the MAXQ2000's SPI and $I^2C$ interfaces, this temperature-monitoring system can work as a low-cost temperature monitor by displaying real-time temperatures through the MAX7219.

Fluke is a registered trademark of Fluke Corporation.

| Related Parts | | |
|---|---|---|
| MAX31875 | Low-Power $I^2C$ Temperature Sensor in WLP Package | Free Samples |
| MAX7219 | Serially Interfaced, 8-Digit, LED Display Drivers | Free Samples |
| MAX8511 | Ultra-Low-Noise, High PSRR, Low-Dropout, 120mA Linear Regulators | Free Samples |
| MAXQ2000 | Low-Power LCD Microcontroller | |

**More Information**
For Technical Support: https://www.maximintegrated.com/en/support
For Samples: https://www.maximintegrated.com/en/samples
Other Questions and Comments: https://www.maximintegrated.com/en/contact

Application Note 6647: https://www.maximintegrated.com/en/an6647
APPLICATION NOTE 6647, AN6647, AN 6647, APP6647, Appnote6647, Appnote 6647