

APPLICATION NOTE 3688

Using the MAXQ2000 with the MAX6960 to Create a Stock Quote Display System

Abstract: This application note provides everything you need to implement a stock quote display system based around the MAX6960 display driver and the MAXQ2000 microcontroller (MAXQ20 core). Included are schematics, Windows software, and microcontroller firmware. The example firmware is an assembly program written and assembled specifically for the MAXQ2000 using the free MAX-IDE. (Version 1.0, Build Date: Nov 18 2004). Download the free [MAX-IDE](#) from Maxim's website.



The stock quote display system.

MAX6960 Overview

The [MAX6960](#) is a graphic LED, tile display driver capable of driving 8x8 LED tiles. This application note uses twenty MAX6960 devices to drive twenty 8x8 LED tiles to display the Maxim stock price. The Windows software and microcontroller firmware can be easily modified to display your company's stock price and symbol.

Hardware Overview

The application circuit utilizes five cascaded [MAX6960EVKITS](#) and one [MAXQ2000-KIT](#). **Figure 1** shows the hardware system block diagram for this setup.

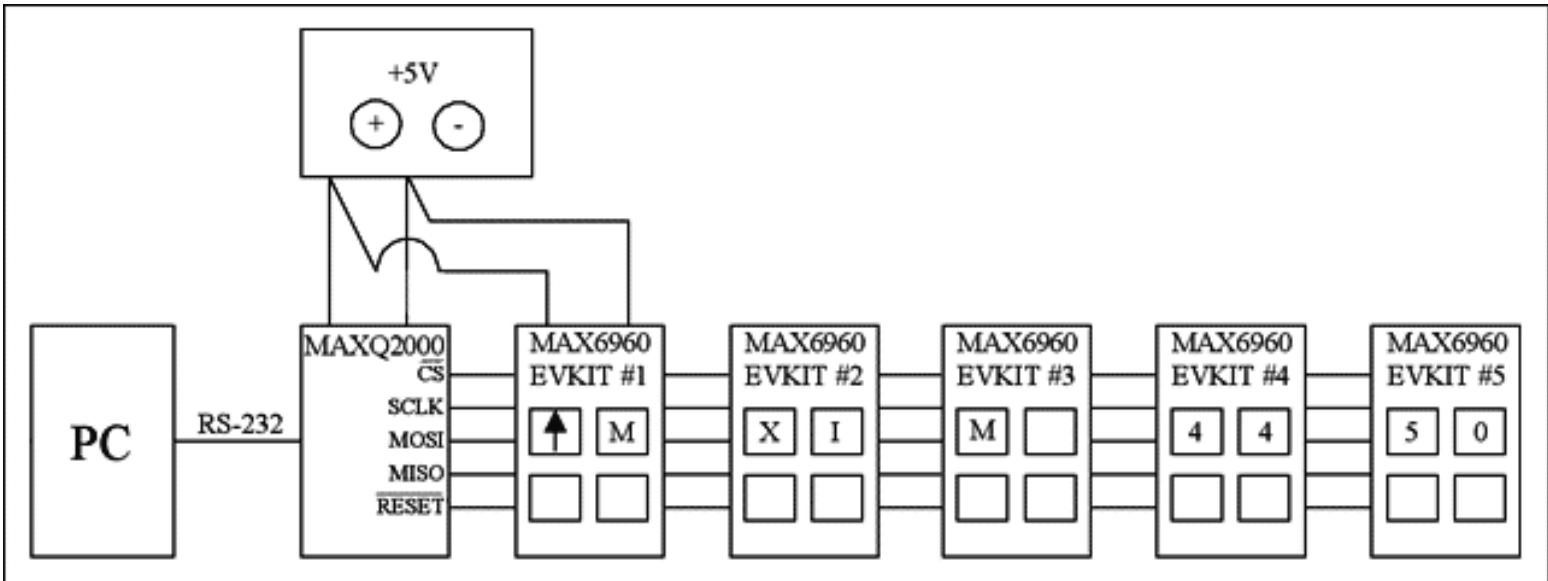
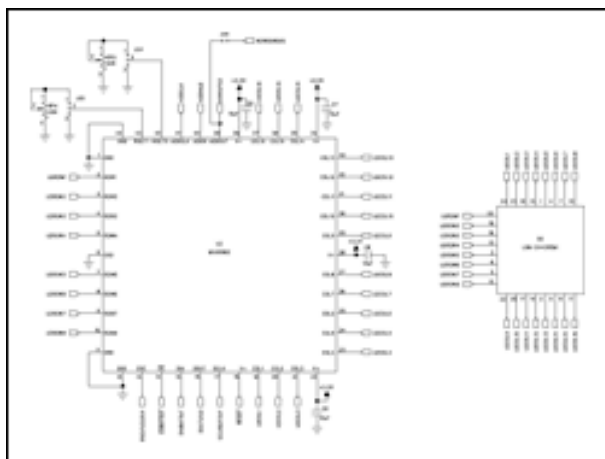


Figure 1. Hardware system block diagram for the stock quote display.

The MAX6960 EV kit consists of four MAX6960 devices, four 8x8 LED tiles, all required passive components, and a proven PCB

layout. The MAX6960 EV kit schematic is shown in **Figure 2**. The MAXQ2000 is not present on the MAX6960 EV kit board; it was, however, added to the system by connecting active-low CS, SCLK, DIN, DOUT, active-low RESET and GND from the MAXQ2000 EV kit to the first MAX6960 EV kit. The remaining four MAX6960 EV kits are cascaded to the first MAX6960 EV kit. In addition, a +5V power supply capable of providing 5A must be applied to the +5V pad of the first MAX6960 EV kit. **Table 1** shows the required jumper settings when using five cascaded MAX6960 EV kits.



[More Detailed Image](#) (PDF, 495kb)

Figure 2. MAX6960 EV kit schematic (sheets 1 - 6).

Table 1. Cascading Five MAX6960 EV Kit Boards

MAX6960 EV kit #1	MAX6960 EV kit #2	MAX6960 EV kit #3	MAX6960 EV kit #4	MAX6960 EV kit #5
JU1: (2-3)	JU1: (2-3)	JU1: (2-3)	JU1: (2-3)	JU1: (2-3)
JU2: (2-3)	JU2: (2-3)	JU2: (2-3)	JU2: (2-3)	JU2: (2-3)
JU3: (1-2)	JU3: (2-3)	JU3: (2-3)	JU3: (2-3)	JU3: (2-3)
JU4: (2-3)	JU4: (2-3)	JU4: (2-3)	JU4: (2-3)	JU4: (2-3)
JU5: (2-3)	JU5: (2-3)	JU5: (2-3)	JU5: (2-3)	JU5: (2-3)
JU6: (1 only)	JU6: (1 only)	JU6: (1 only)	JU6: (1 only)	JU6: (1-2)
JU7: (2-3)	JU7: (2-3)	JU7: (2-3)	JU7: (2-3)	JU7: (2-3)
JU8: (2-3)	JU8: (2-3)	JU8: (2-3)	JU8: (2-3)	JU8: (2-3)
JU9: (1-2)	JU9: (2-3)	JU9: (2-3)	JU9: (2-3)	JU9: (2-3)
JU10: (2-3)	JU10: (2-3)	JU10: (2-3)	JU10: (2-3)	JU10: (2-3)
JU11: (2-3)	JU11: (2-3)	JU11: (2-3)	JU11: (2-3)	JU11: (2-3)
JU12: (OPEN)	JU12: (SHORT)	JU12: (SHORT)	JU12: (SHORT)	JU12: (SHORT)
JU13: (1-2)	JU13: (2-3)	JU13: (2-3)	JU13: (2-3)	JU13: (2-3)
JU14: (1 only)	JU14: (1-2)	JU14: (1-2)	JU14: (1-2)	JU14: (1-2)
JU15: (1-2)	JU15: (1 only)	JU15: (1 only)	JU15: (1 only)	JU15: (1 only)
JU16: (1-2)	JU16: (1-2)	JU16: (1-2)	JU16: (1-2)	JU16: (1-2)
JU17: (1 only)	JU17: (1-2)	JU17: (1-2)	JU17: (1-2)	JU17: (1-2)

Note: Bold indicates changes from the default settings.

The MAXQ2000 EV kit contains a complete set of software, hardware, sample code, and documentation needed to start designing with the MAXQ2000 microcontroller. Schematics for the [MAXQ2000-KIT](#) are available in the data sheet.

Firmware Overview

The example [MAX-IDE](#) assembly program files in [Firmware.Zip](#) allow the MAXQ2000's SPI peripheral to communicate with the MAX6960. The SPI serial clock is 2.5MHz when a 16MHz system clock is used for the MAXQ2000. **Table 2** shows the SPI/ GPIO operations used to initialize all the MAX6960 devices after power-up.

Table 2. SPI/GPIO Operations to Initialize All the MAX6960 Devices after Power-Up

Step	Operation	Register Address	Description
1	GPIO: active-low RESET	N/A	active-low RESET was held low for 2.5ms after power-up to properly apply power to all five MAX6960 devices while power is stable. active-low RESET was then driven high for 100ms.
2	SPI: 16-bit address mode write	Panel Configuration Register (0x0D)	Sets: <ul style="list-style-type: none"> • iColor to bicolor, 1-bit per pixel • Take all MAX6960 devices out of shutdown
3	SPI: 16-bit address mode write	Global Driver Devices Register (0x0E)	Sets: <ul style="list-style-type: none"> • iNumber of devices to 20.
4	SPI: 16-bit address mode write	Global Driver Rows Register (0x0F)	Sets: <ul style="list-style-type: none"> • iNumber of rows to 2.
5	Delay	N/A	Adds a required 450ms delay after steps 1 through 4 above. All the MAX6960 devices require this time to properly configure themselves.
6	RS-232 serial COM port	N/A	Waits for stock quote bytes to be sent from the Windows program to the MAXQ2000 through the RS-232 serial COM port.

Windows Program Overview

Once the MAX6960 devices are initialized, the firmware waits for the [Windows program](#), shown in **Figure 3**, to obtain Maxim's stock price from Yahoo's website using the /n software™ IP*Works™ internet development components for Borland C++ Builder. A free trial version with several demo programs can be downloaded from the [/n Software](#) website. An active Internet connection is required for the Windows program to run correctly. If you are accessing the Internet through a proxy server, then you will need to provide the name of the proxy server and the proxy port number. Once executed, the Windows program automatically sends the most recent quote for MXIM (or your monitored company) to the MAXQ2000-KIT through the RS-232 serial COM port every 10 seconds. The user does not need to press any buttons on the Windows program unless proxy server information is required to establish the Internet connection.

The RS-232 serial COM port data format is:
 XXXU567.89X

X = Byte1 (Factory use only)
 X = Byte2 (Factory use only)
 X = Byte3 (Factory use only)
 U = Byte4 (Stock Quote: Up, Down, or No Change Indicator)
 5 = Byte5 (Hundreds Digit: 1xx.xx)
 6 = Byte6 (Tens Digit: x1x.xx)
 7 = Byte7 (Ones Digit: xx1.xx)
 8 = Byte8 (Tenths Digit: xxx.1x)
 9 = Byte9 (Hundredths Digit: xxx.x1)
 X = Byte10 (Factory use only)

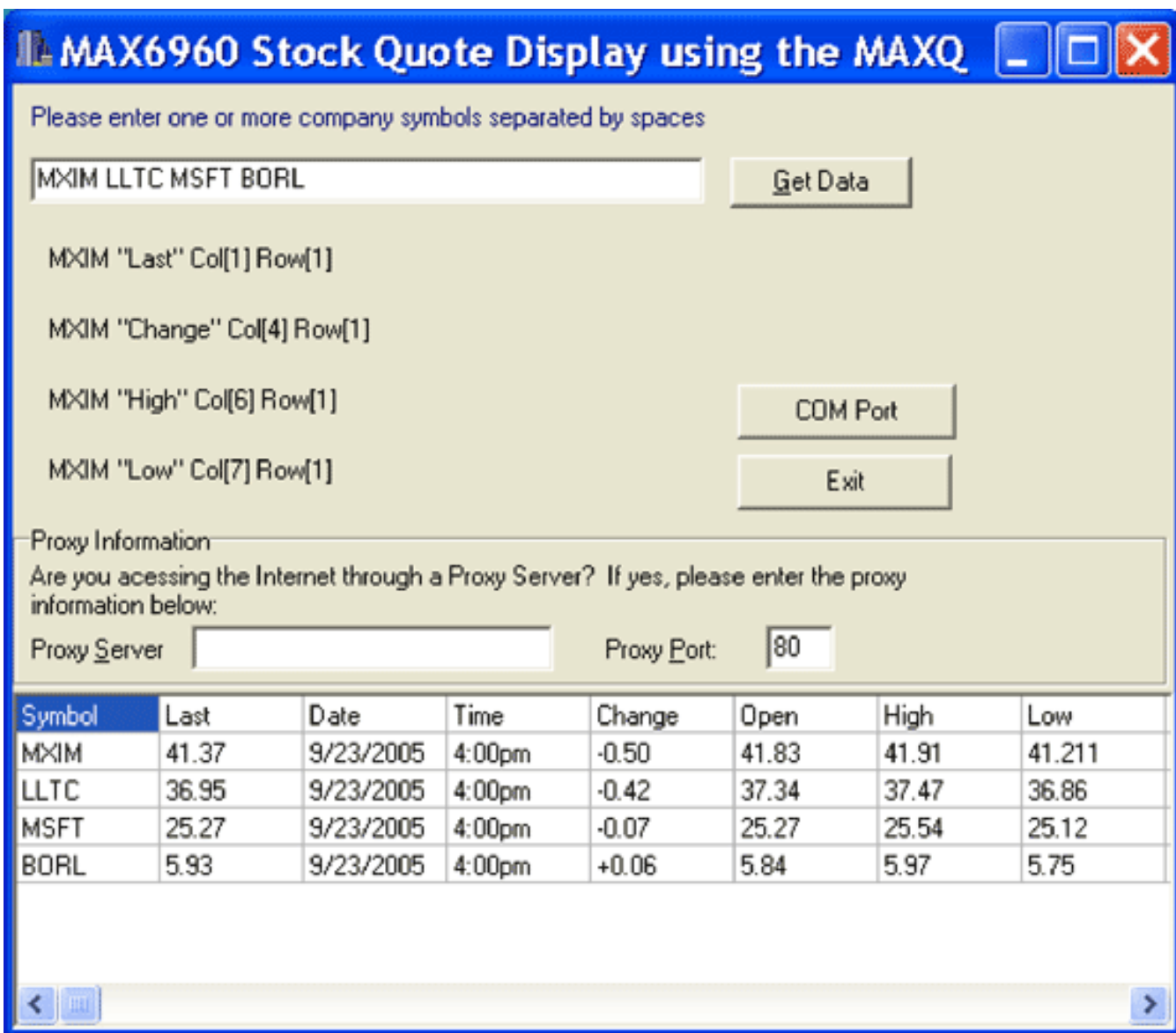


Figure 3. Windows program for obtaining the MXIM stock quote from the Internet.

/n software and IP*Works! are trademarks of /n software Inc.

Application Note 3688: www.maxim-ic.com/an3688

More Information

For technical support: www.maxim-ic.com/support

For samples: www.maxim-ic.com/samples

Other questions and comments: www.maxim-ic.com/contact

Automatic Updates

Would you like to be automatically notified when new application notes are published in your areas of interest? [Sign up for EE-Mail™.](#)

Related Parts

MAX6960: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)

MAX6961: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)

MAX6962: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)

MAX6963: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)

MAXQ2000: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

AN3688, AN 3688, APP3688, Appnote3688, Appnote 3688

Copyright © by Maxim Integrated Products

Additional legal notices: www.maxim-ic.com/legal