



Maxim > Design Support > Technical Documents > Application Notes > Automotive > APP 5280

Maxim > Design Support > Technical Documents > Application Notes > Energy Measurement & Metering > APP 5280

Maxim > Design Support > Technical Documents > Application Notes > Sensors > APP 5280

Keywords: passive keyless entry, PKE, remote keyless entry, RKE, signal conditioner, low power, proximity sensor, touch sensor, RF ID readers, meter tampering prevention

## APPLICATION NOTE 5280

# MAX1441 Proximity and Touch Sensor Programmer User Manual

By: Youssef Fathi

Dec 21, 2011

*Abstract: The MAX1441 proximity and touch sensor is designed for capacitive proximity sensing applications, including automotive passive remote keyless entry (PRKE). The sensor has 2-kwords of embedded flash that must be programmed with the user's application firmware. This programmer was developed to facilitate programming of the MAX1441 flash memory in a production environment. The programmer continuously polls the USB-to-JTAG interface for a communicating MAX1441. Once the MAX1441 device is found, it will erase, program, and verify the embedded flash memory in approximately 3 seconds. It will then start polling for the next MAX1441 to be programmed. Multiple instances of the application can be launched on the same computer.*

## Introduction

For any manufacturing line, minimizing cost and maximizing efficiency are main concerns. To facilitate achieving this goal, Maxim has developed [application software](#) to program the [MAX1441](#) automotive proximity and touch sensor's embedded flash memory in a production environment. The software will continuously poll the JTAG interface through the USB port for a communicating MAX1441. Once the MAX1441 device is found, it will perform a master erase of the flash memory, write the specified firmware into the flash memory, and verify its content. All three operations will be completed in approximately 3 seconds. The [MAXQUSBTAG-KIT](#) device-driver must be properly installed before launching the software.

The MAX1441 Programmer Software verifies completion of all the necessary operations before moving forward, guaranteeing that no extra time is spent and that the operation has been completed. Also, no user interaction is necessary with mouse or keyboard for the whole programming sequence. The user only inserts the MAX1441 and removes it after programming is complete, as the software automatically detects the presence or absence of the MAX1441. To avoid hot-plugging the MAX1441, an on/off power switch may be required. It is possible to run multiple instances of the application on the same computer. An available USB port and a MAXQUSBTAG-KIT is required for each instance. When multiple instances of the application are running, there is no need to know which communication (COM) port the operating system has assigned to a particular socket. The preferred sequence of starting multiple instances of the application is to connect hardware of one instance to a USB port and start the programming for one device before starting the next one.

## Requirements

The MAX1441 Programmer software requires the following:

- a. Windows XP®/Windows Vista®/7 (32 or 64-bit) computer.
- b. One available USB port for each instance of application running.
- c. One Maxim USB-to-JTAG board (MAXQUSBTAG-KIT) for each instance of application running.

- d. Required power supply for MAX1441 VBAT terminal.
- e. Firmware file with the HEX extension, in the same directory as the application.
- f. Test socket to receive the MAX1441.

## Software Installation

Download and extract the [MAX1441\\_programmer.zip](#) in a temporary folder. Run `Install_MAX1441_Programmer.exe` and follow instructions. The MAX1441\_programmer application and supporting files will be downloaded in the folder specified during installation. Default directory is **C:\MAX1441**. At the completion of installation, the content of the folder should look similar to **Figure 1**.

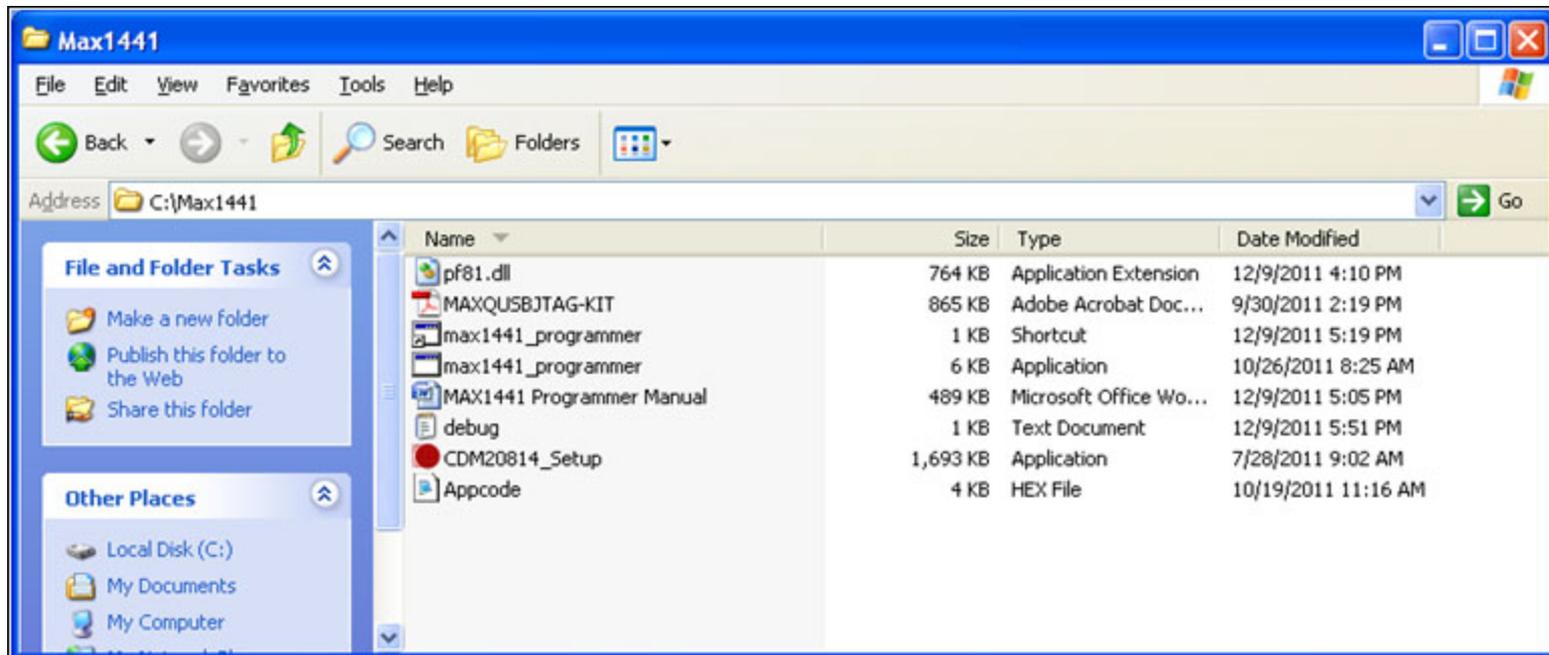


Figure 1. File contents of the **c:\MAX1441** folder.

## Driver Installation

Double-click on the **CDM20814\_Setup** file to install the FTDI Virtual Com-Port Driver. A screen similar to the one in **Figure 2** will appear. Driver installation will start and, at the completion, the window automatically closes.

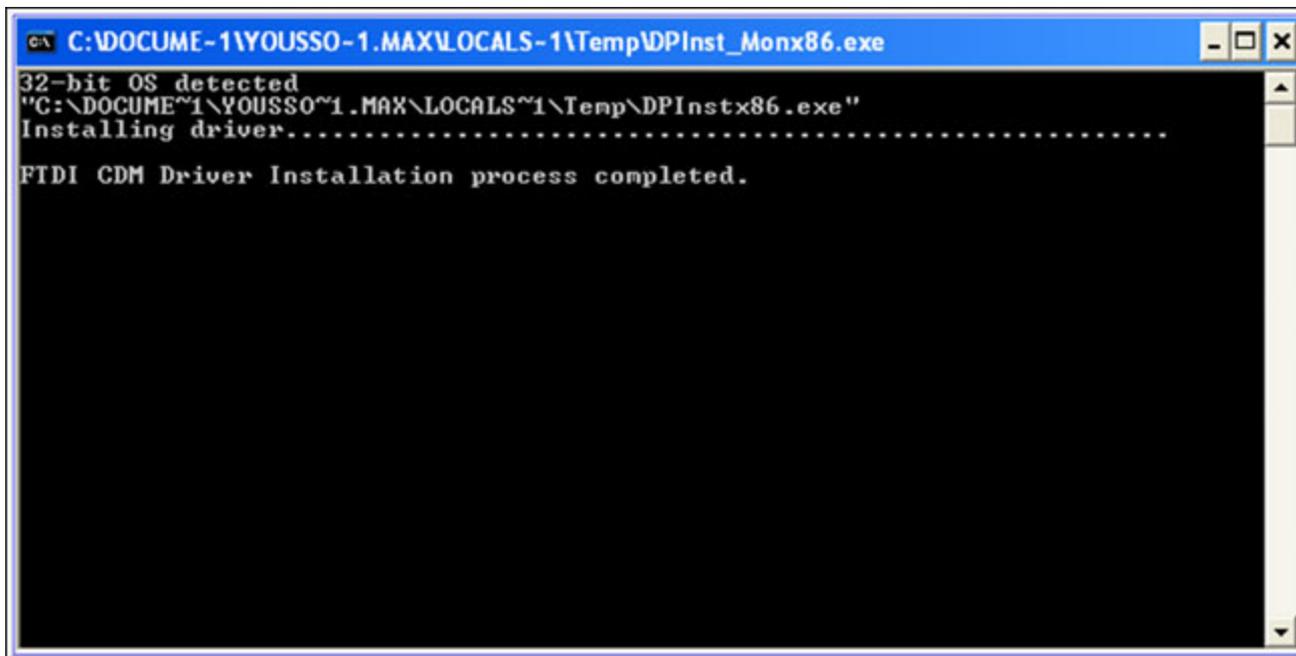


Figure 2. Virtual Communication Port Driver installation.

### USB-JTAG Adapter

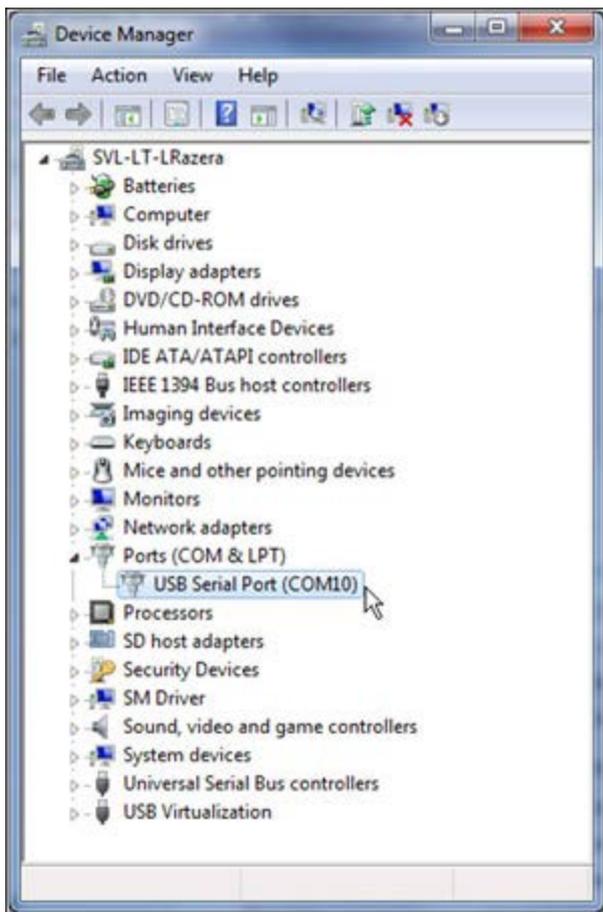
After the FTDI Virtual Communication Port Driver is installed, connect a MAXQUSBJTAG-KIT (Figure 3) to an available USB port, using a mini-USB connector.



Figure 3. The Maxim USB-JTAG adapter (MAXQUSBJTAG-KIT).

Once the adapter is found as a new hardware by the operating system, proper drivers installed in the previous step should be loaded. To

verify, open the **Device Manager**, and check for the device under **Ports (COM & LPT)**. See **Figure 4**.



*Figure 4. A Maxim USB-to-JTAG converter identified as USB Serial Port.*

For detailed information about the Maxim USB-to-JTAG adapter, including further details about driver installation, electrical characteristics, and features, refer to the [MAXQUSBJTAG-KIT](#) PDF file, also present in the installation directory.

## Programming Procedure

There are two ways to start the MAX1441 Programmer application: directly and via a shortcut. Starting the application through a shortcut is the preferred way.

### Starting the Programmer Directly

To start the MAX1441 Programmer directly, double-click on the **MAX1441 Programmer** application file. On detailed file view, the programmer folder should look similar to one in **Figure 5**.

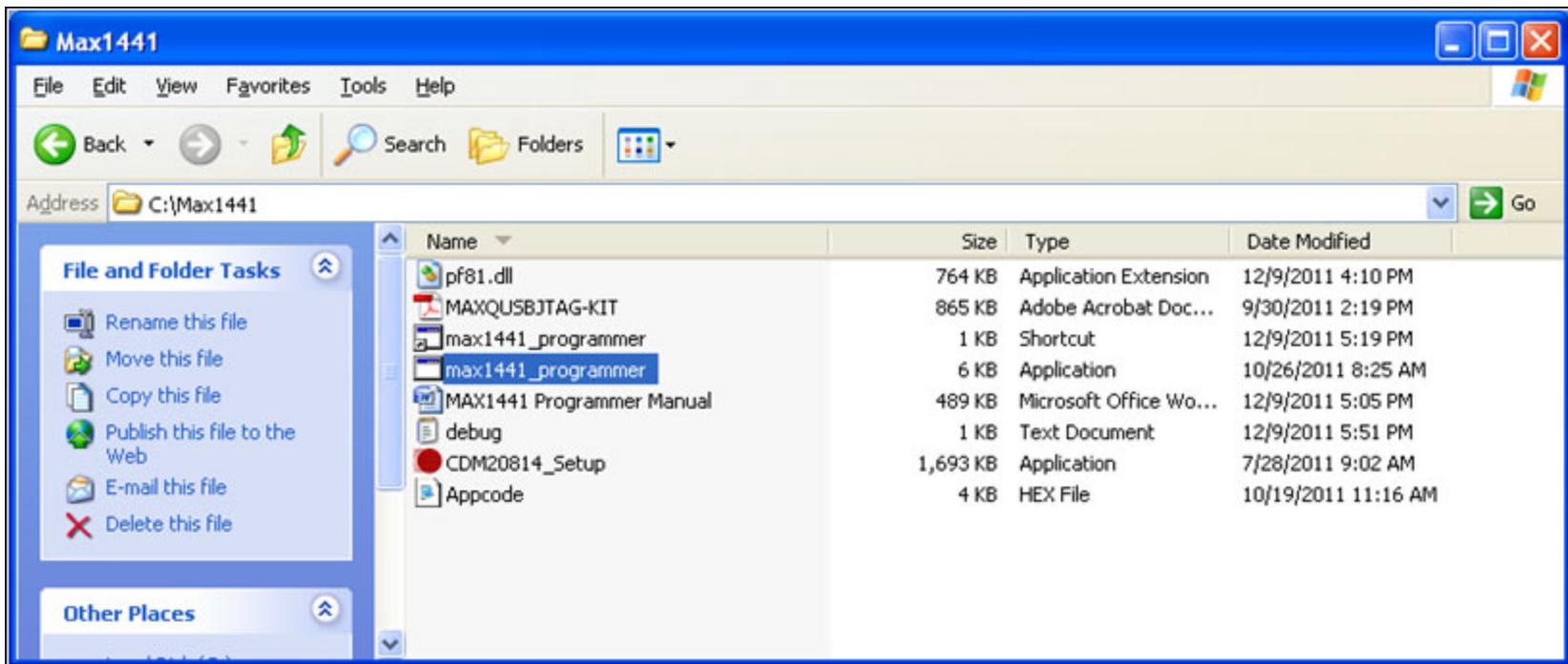


Figure 5. Detailed View of **C:\MAX1441** folder, indicating the application for direct launch.

After the Programmer starts, it will scan the system for available COM ports and search for the Maxim USB-to-JTAG interface board. Once a USB-to-JTAG interface is identified, a screen will appear (**Figure 6**). All HEX files (firmware) in the folder from which the application was launched will be listed. Note that the identified COM port is listed on the window-header.

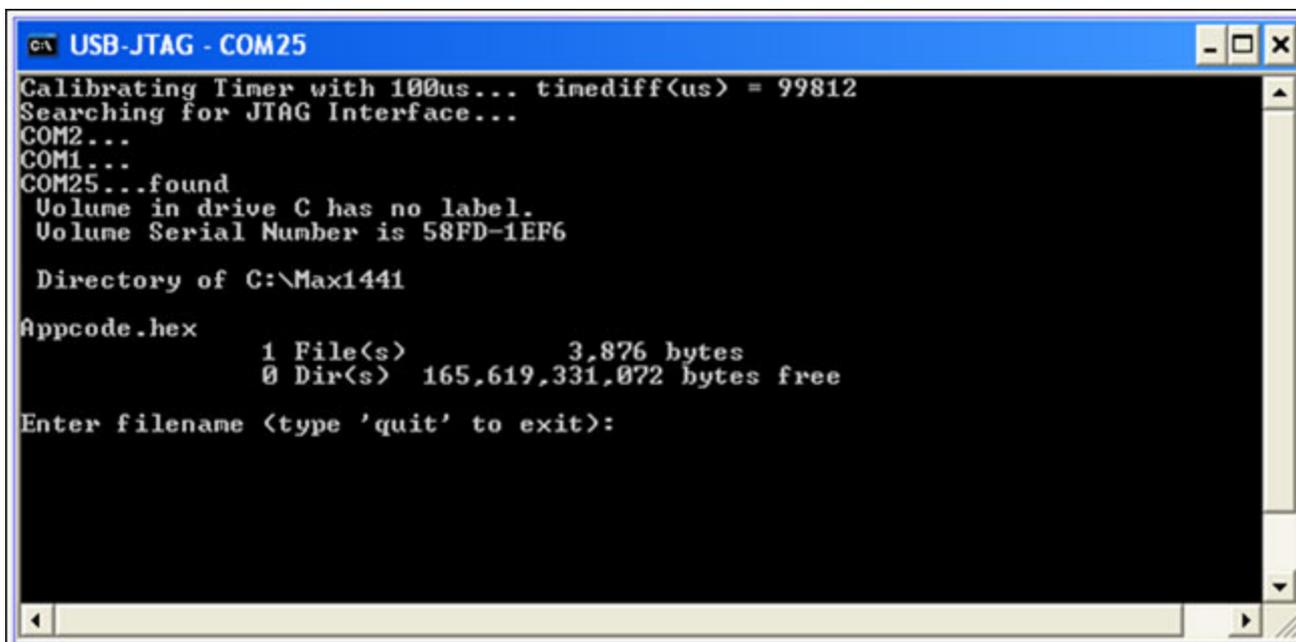


Figure 6. Screen shown when the MAX1441 Programmer is started directly.

Type in the complete file name (including the file extension) of the desired firmware to be used in the programming session.

Notes:

- Do not create filenames with spaces; use underscores, "\_", instead.
- Keep firmware filenames short if possible.

## Starting the Programmer from a Shortcut

To start the Programmer from a shortcut, first create a shortcut to the MAX1441 Programmer application. In the shortcut's properties, specify the firmware filename.

Using a shortcut eliminates the need to type in a firmware file name every time application is started. And because the shortcut already has the file name information, it eliminates operator error and thus makes it an ideal choice in a production environment. To verify and/or modify information in the shortcut, right-click on the shortcut and select **Properties** (Figure 7). The default Target in the **Properties | Shortcut** tab is **C:\Max1441\max1441\_programmer.exe appcode.hex**, where "appcode.hex" is the firmware name to be written to the MAX1441 devices. For ease of use, copy the shortcut on the PC desktop and launch it from there.

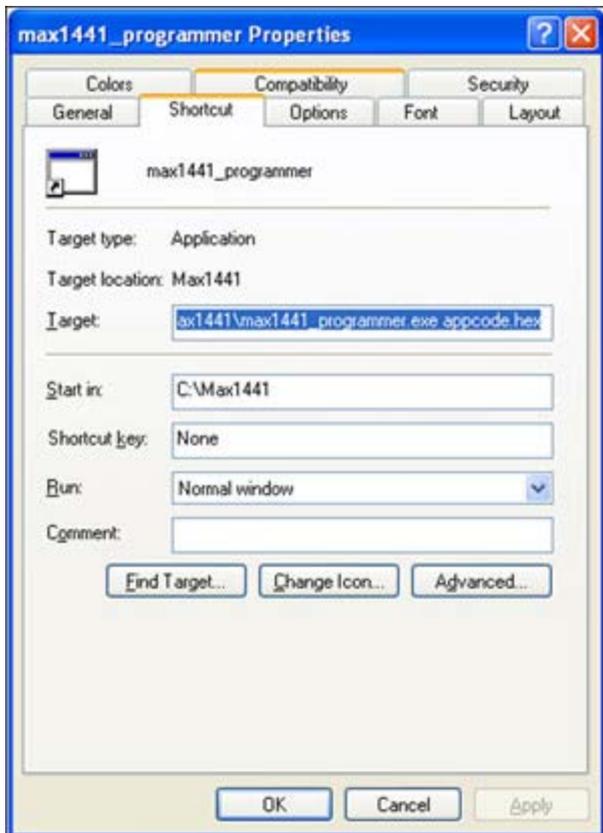


Figure 7. Properties window of "MAX1441\_Programmer" shortcut.

Once the MAX1441 Programmer has started, through a direct launch or shortcut, a screen will appear where the firmware file name and the COM port are shown in the title bar (Figure 8).

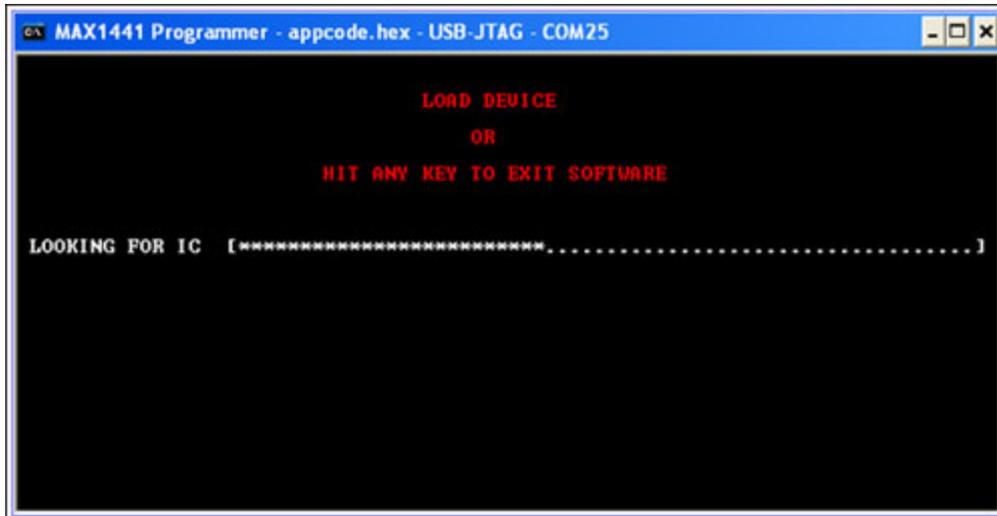


Figure 8. MAX1441 Programmer Search Mode.

At this point, the software is actively looking for a MAX1441. The progress bar completes in approximately 13 seconds, and it returns to the beginning if a MAX1441 is not found. **The Programmer will terminate and close if any key on the keyboard is pressed.**

Once a MAX1441 is found, flash memory programming starts and a window similar to the one in **Figure 9** will be display.

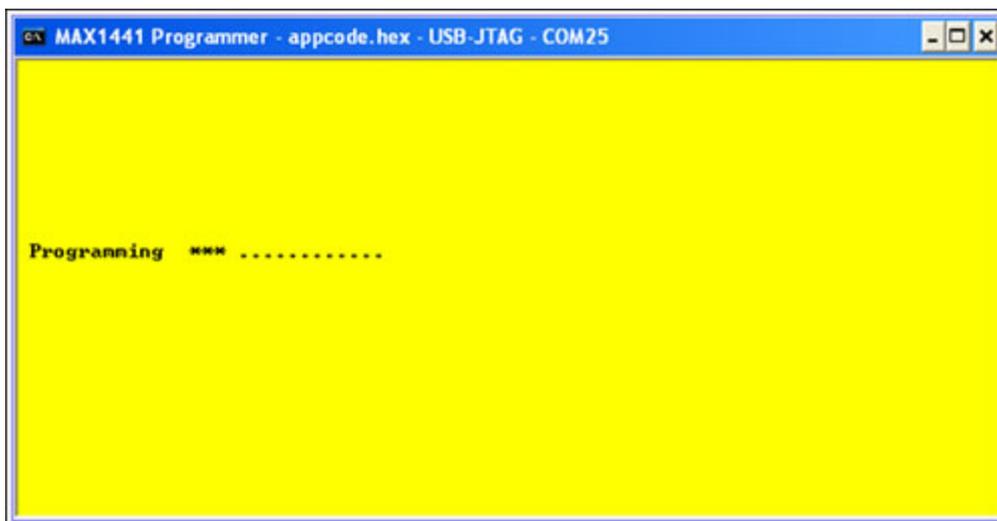


Figure 9. Programming Mode of the MAX1441 Production Programmer.

**Caution:** The MAX1441 interface must **NOT** be disturbed while programming is in progress (while the yellow window is showing). Any interruption in communication between the programmer and the MAX1441 will cause programming to fail.

The Programming line in Figure 9 shows the process progress. The "\*" symbols indicate memory erasing, and the periods indicate progress in writing to flash memory. Each period indicates programming of 32 words (64 bytes) of memory. Total number of periods displayed will depend on the size of the application firmware. Once the programming session is completed, a green or a red window will show. If the device programming was successful, a green window will display (**Figure 10**). The programmer will immediately start searching for the next device to program. If the setup is not disturbed and same device is left in the setup, it will be reprogrammed after approximately 10 seconds. To terminate the application, press the ESC key, and to start reprogramming the same device without further delay, press any other key. If the device is removed from the setup, a window similar to that in Figure 8 will appear and programming will

start as soon as a communicating MAX1441 is reloaded.



Figure 10. Pass Status screen of MAX1441 Programmer.

If programming of the MAX1441 is not successful, a red window will display (**Figure 11**). Remove the device and load another one, press any key (but the ESC key) to reprogram the same device, or press the ESC key to terminate the application.



Figure 11. Fail Status screen of the MAX1441 Programmer.

In addition to displaying the green and red windows to indicate pass or fail programming of flash memory, the programmer will also toggle the OUT1 and OUT2 pins of the MAX1441 to reflect the programming operation progress and results:

- At the completion of flash-erase operation, both OUT1 and OUT2 pins will be pulled low and released. On the MAX1441 evaluation board, this will cause LED1 and LED2 to blink.
- If programming the flash is successful, OUT1 pin will be pulled low. On the MAX1441 evaluation board, this will cause LED1 to turn on. OUT1 will be kept low until MAX1441 is removed or next programming cycle is started.
- If programming the flash is unsuccessful, OUT2 pin will be pulled low. On the MAX1441 evaluation board, this will cause LED2 to turn on. OUT2 will be kept low until MAX1441 is removed or next programming cycle is started.

When the programming of the current MAX1441 is complete, remove the device from the socket and insert the next one to be programmed.

This application can be utilized in an automated manufacturing environment by using state of OUT1 and OUT2 pins to sort out (bin) the good and bad devices.

**Caution:** While the MAX1441 Programmer is running, do not disconnect the USB cable connecting the Maxim USB-to-JTAG interface board to the computer. Any interruption will cause application to crash.

Windows Vista is a registered trademark and registered service mark of Microsoft Corporation.

Windows XP is a registered trademark and registered service mark of Microsoft Corporation.

#### Related Parts

<a href="#">MAX1441</a>	Automotive, Two-Channel Proximity and Touch Sensor
<a href="#">MAX1441EVSYS</a>	Evaluation System for the <a href="#">MAX1441</a>
<a href="#">MAXQUSBTAG-KIT</a>	Evaluation Kit for the MAXQ USB-to-JTAG Board

#### More Information

For Technical Support: <http://www.maximintegrated.com/support>

For Samples: <http://www.maximintegrated.com/samples>

Other Questions and Comments: <http://www.maximintegrated.com/contact>

Application Note 5280: <http://www.maximintegrated.com/an5280>

APPLICATION NOTE 5280, AN5280, AN 5280, APP5280, Appnote5280, Appnote 5280

Copyright © by Maxim Integrated Products

Additional Legal Notices: <http://www.maximintegrated.com/legal>