



[Maxim](#) > [Design Support](#) > [Technical Documents](#) > [Application Notes](#) > [Interface Circuits](#) > APP 3937

[Maxim](#) > [Design Support](#) > [Technical Documents](#) > [Application Notes](#) > [Power-Supply Circuits](#) > APP 3937

Keywords: MAXIM,USB,lab,laboratory,MAX3421E,MAX3420E,host,peripheral

APPLICATION NOTE 3937

Setting Up the Maxim USB Laboratory

Nov 02, 2006

Abstract: This application note is a companion to application note 3936, "The Maxim USB Laboratory," which describes a hardware and software combination that simplifies development of USB embedded host and peripheral applications. The MAX3421E is the host in this system, and the MAX3420E is the peripheral. This application note explains how to set up the hardware using an evaluation (EV) kit and then shows three ways to evaluate the software.

Introduction

A companion application note 3936, "[The Maxim USB Laboratory](#)," describes the C code and a Keil™ project for implementing both a USB host (MAX3421E) and peripheral (MAX3420E) on the same ARM®-based hardware. The hardware for this kit comprises:

- [MAX3421E EVKIT-1 board](#)
- A Keil MCB2130 board

The software is available for [download](#).

This application note explains how to assemble and configure the various components for this kit.

Other Reading and Contact Information

Consult the Maxim website for the [MAX3420E](#) and the [MAX3421E](#). You will find data sheets, another application note describing the Maxim USB Laboratory code, many application notes relating to the MAX3420E and MAX3421E, and a way to order samples.

If you have questions about the MCB2130 board or the Keil Software® and ARM software development tools, please contact Keil at www.keil.com.

The Hardware Configuration

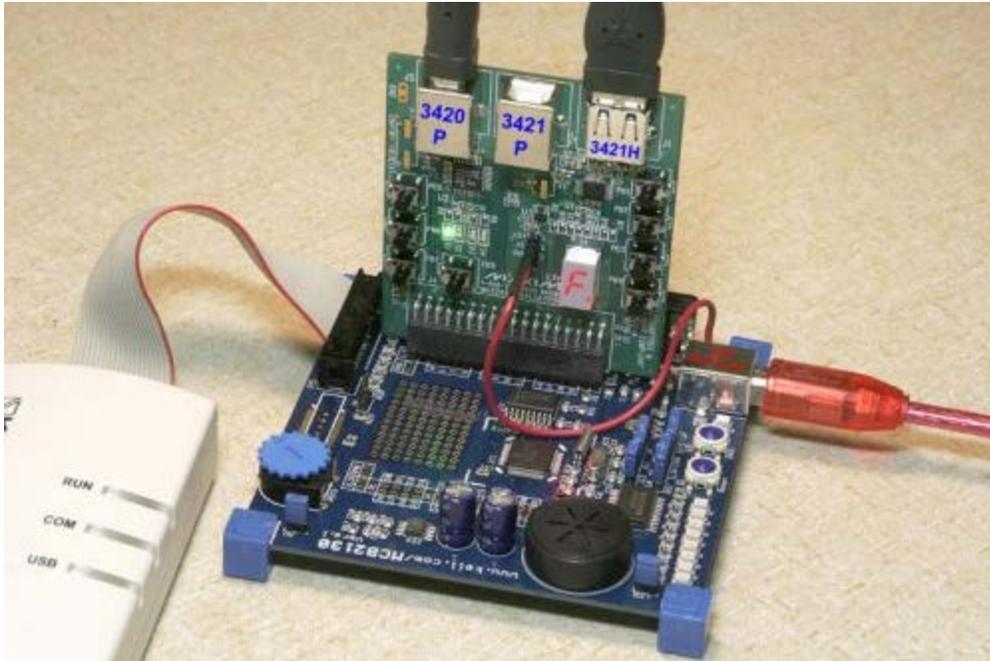


Figure 1. The Maxim MAX3421E EVKIT-1 plugs into the Keil MCB2130 board.

Figure 1 shows the two-board set. The blue board is a Keil MCB2130 which contains the Philips® LPC2138, an ARM7™ microcontroller. The LPC2138 has two SPI™ hardware units, each of which connects to one of the two USB controller chips.

The vertical board is the MAX3421E EVKIT-1. A MAX3420E peripheral controller connects to one of the ARM SPI ports, and is wired to the USB "B" connector (J5) labeled "3420P" (P = peripheral) in Figure 1. A MAX3421E host/peripheral controller connects to the other ARM SPI port, and is wired to the "3421P" (J2) and "3421H" (J1) connectors (H = host). The USB Laboratory code uses the MAX3421E as a host, so the middle USB connector (J2) on the EV kit board is not used.

The host software uses one of the two serial ports on the MCB2130 board (P1) to send USB descriptive information to a PC running a terminal emulation program. Terminal programs such as Tera Term Pro can emulate terminals (VT100) that recognize special "escape code" sequences sent by the program to clear the screen and home the cursor. The settings for the terminal program are 38400, N, 8, 1 with no flow control.

The beige box and ribbon cable in Figure 1 is a Keil ULINK® JTAG loader-debugger. This unit is supported by the Keil µVision®3 development environment. The MCB2130 board comes with an evaluation version of µVision3, a fully functional version of the Keil toolset. This evaluation version limits code size to 16kB.

There are four USB connectors on this board set. The USB connector on the Keil board has only V_{BUS} and ground connected. This connector supplies 5V power to the board set by plugging into any USB host. Some USB hubs may not supply enough current to power the boards, so it is best to get power directly from a spare PC USB port.

The Maxim board has three USB connectors. As explained above, a B connector is wired to the MAX3420E, shown in Figure 1 as 3420E P. The MAX3421E, which can function either as USB peripheral or host, is wired to two USB connectors, 3421P and 3421H. These connectors have their D+

and D- pins wired together, and are intended to be connected only one at a time. The USB Laboratory code does not use the middle USB connector, 3421P.

The MAX3420E and MAX3421E connect to two separate SPI ports on the ARM microcontroller. This configuration makes it easy to write both host and peripheral applications that run concurrently in the same code. This also provides an ideal USB training and debugging system. The host can dispatch USB requests to the peripheral, the peripheral can respond, and the host can evaluate the results, all in the same C code. The black USB cable in Figure 1 shows the USB cable that connects the MAX3420E and MAX3421E together for such a system.

The Keil MCB2130 board is available from Keil and other sources. The Maxim webpages noted above will have current information on Keil product availability.

MAX3421E EVKIT-1 Block Diagram

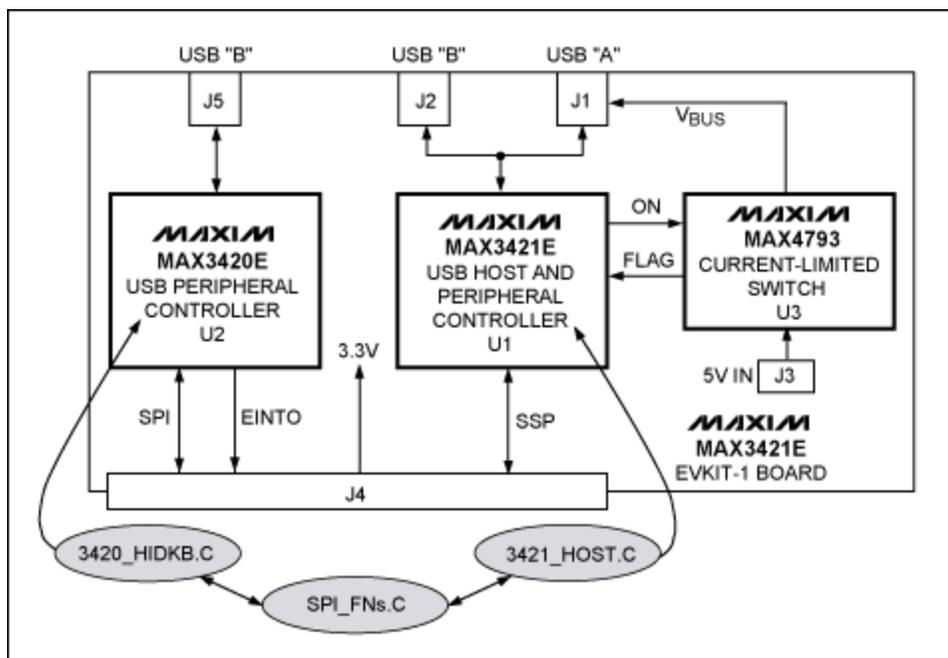


Figure 2. Block diagram of the MAX3421E EV kit. The shaded ovals are the software modules that run the application.

Figure 2 shows how the two USB controllers connect to the USB connectors J1, J2, and J5. A 36-pin connector (J4) is wired to plug into the Keil MCB2130 development board. The design uses two separate SPI ports in the ARM processor to provide independent operation of the two USB controllers (U1 and U2). The host software runs in the background (`main{ }` is here), while the peripheral code is invoked using LPC2138 interrupt EINT0.

Because U1 operates as a host, the system must supply V_{BUS} power to USB A connector J1. A 5V power source is supplied to the EV kit board using a flying lead attached to J3-3. This 5V power can originate from a test pad on the Keil board "5V IN," or from a laboratory supply. U3, the MAX4793, switches and current-limits the V_{BUS} voltage. U3 controls V_{BUS} power to J1 using its GPO-7 pin, and it can detect an overcurrent (over 300mA) condition on its GPIN-0 pin.

Buttons and lights on the EV kit board connect to both controllers. U1 drives a 7-segment readout and

connects to four pushbuttons (PB1–PB4). U2 drives four LEDs and connects to pushbuttons PB5 through PB8.

Preparing the Keil MCB2130 Board

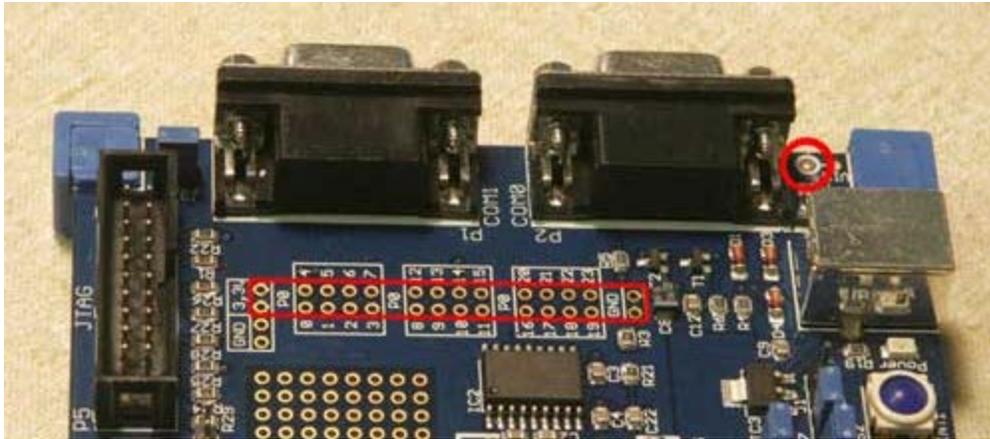


Figure 3. Solder a 36-pin male header into the site shown in the red rectangle.

The Keil MCB2310 board has sites designed to accept multiple 2-pin and 8-pin headers (**Figure 3**). The Maxim EV kit board uses a single 36-pin header (included with the board) to attach to the Keil board by spanning the 3.3V power pins, the three 8-pin headers, and the two GND pins (red outline in Figure 3). Spaces exist between the various Keil connector sites, so certain pin pairs must be removed from the 36-pin header before installing it. Make sure that the "missing" pins are removed from the header, then solder it onto the Keil board.

Figure 3 shows a test point pad (red circle) in the upper right corner of the board. This is where 5V power can be tapped for use by the Maxim board. This 5V power originates from the V_{BUS} pin of the USB power connector on the Keil board (**Figure 4**).

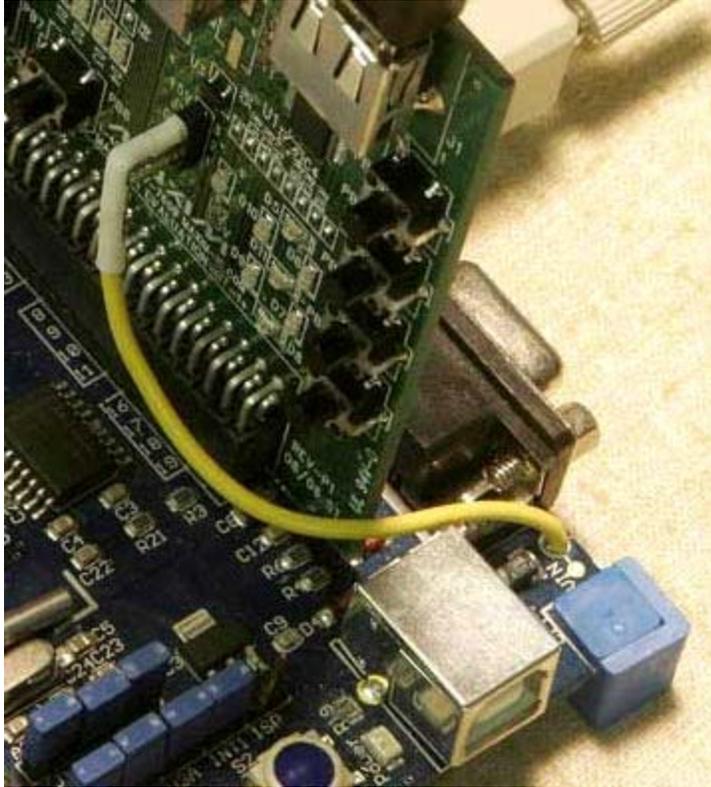


Figure 4. A flying lead supplies V_{BUS} power to the Maxim board.

Installing the Keil Software

A demonstration version of the Keil μ Vision3 ARM development software is available at www.keil.com. The μ Vision3 project file supplied with the Maxim software package was built using the downloaded Keil file `mdk303a.exe`. It is important to use this version because two of the project files are provided by Keil in their development environment. (The files are not included in the `MaximUSBLab10.zip` file.) The `MaximUSBLab` project locates these files using specific path names under `C:\Keil`. When installing the Keil demo tools, be sure to accept the default install directory as `C:\Keil` so the pathnames to the Keil-provided files are correct.

For the "`mdk303a.exe`" toolset from Keil, the two Keil-provided files are located at:

```
C:\Keil\ARM\RV30\Boards\Keil\MCB2130\Blinky\startup.s  
C:\Keil\ARM\RV30\Boards\Keil\MCB2130\Blinky\retarget.c
```

The Keil demo projects include many versions of "Blinky." It is important to use the files corresponding to the MCB2310 board, which is the board used by the Maxim software.

If you have a Keil ULINK JTAG debug unit, you also should download the USB driver from the Keil website. The Maxim code has been tested using the file `3071.zip` from the Keil website. After installing the demo μ Vision development software, double-click on this file to install the USB driver that connects μ Vision3 to the ULINK box.

Once the Keil demo software is installed, navigate to the `MaxUSBLab10` folder and double-click on the file, `MaximUSBLab.Uv2`. This opens the project with all the correct project settings. Click the "Rebuild All

Target Files" button and the project should compile. If you have a ULINK box and have installed the USB driver, you can click on "Start/Stop Debug Session" to run, modify, and debug the Maxim example code.

Three Ways to Run the Firmware

1. Load and Observe MaximUSBLab.hex

The Maxim USB Laboratory load (hex) file is available on the Maxim website as part of the zipped Keil project included with application note 3936, "[The Maxim USB Laboratory](#)." You can load the hex file into the LPC2138 flash memory using a free utility called Flash Magic (**Figure 5**), available at www.esacademy.com. Before using this utility, check the indicated box (**Figure 6**) under Options/Advanced Options/Hardware Config.

On the Keil board, use the P2 serial connector and move jumper J7 to the ISP position to use the Flash Magic utility.

Note: Moving the J7 jumper deactivates pushbutton S2 as an INT1 interrupt source. You can leave J7 in the ISP position since the Maxim application code uses INT1 for other purposes.

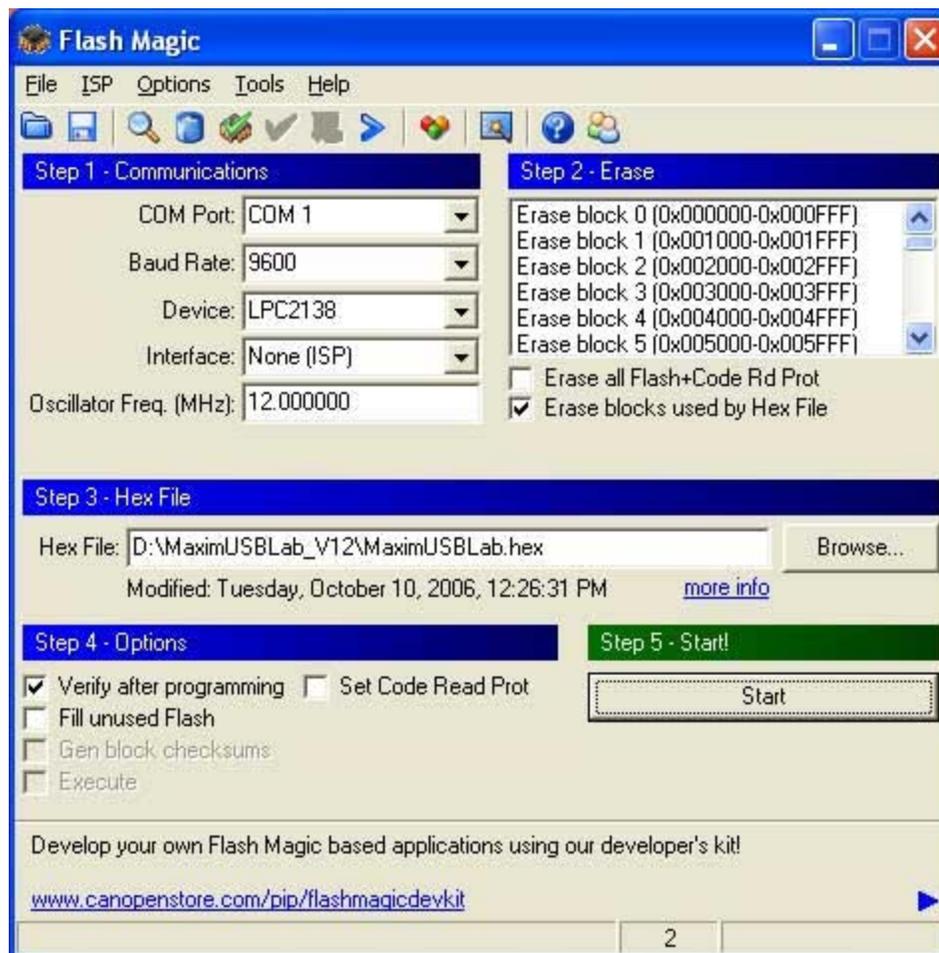


Figure 5. Typical settings in the Flash Magic screen for the MCB2130 board.



Figure 6. Set this option first.

2. Debug and Modify (Limited)

If you have the Keil ULINK JTAG unit, you can compile the Maxim demo code, load it, and debug it through the JTAG port. You can also set breakpoints and single-step through the code. This is an excellent way to get up to speed on USB—study and modify working host and/or peripheral code to suit your purposes.

The demonstration version of the Keil μ Vision3 for ARM has the full version of μ Vision3, code-size limited to 16kB. The Maxim USB Laboratory project (MaximUSBLab10.Uv2) was compiled in "Thumb only" mode to make the code size for this project under the limit.

3. Full Development

For serious development work it is highly recommended that you purchase the full Keil μ Vision3 ARM toolset. This removes the 16kB limit and allows much larger applications, up to the full 512kB of flash in the LPC2138.

μ Vision is a registered trademark of ARM, Inc.

ARM is a registered trademark and registered service mark of ARM Limited.

ARM7 is a trademark of ARM Limited.

Keil is a registered trademark and registered service mark of ARM Limited.

Keil Software is a trademark of ARM Limited.

Philips is a registered trademark of Koninklijke Philips Electronics N.V. Ltd.

ULINK is a registered trademark of ARM, Inc.

Related Parts

MAX3420E	USB Peripheral Controller with SPI Interface	Free Samples
MAX3421E	USB Peripheral/Host Controller with SPI Interface	Free Samples
MAX3421EVKIT-1	Evaluation Kit for the MAX3421E/MAX3420E	

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 3937: <http://www.maximintegrated.com/an3937>

APPLICATION NOTE 3937, AN3937, AN 3937, APP3937, Appnote3937, Appnote 3937

Copyright © by Maxim Integrated Products

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