

## MUSIC KEYBOARD EVALUATION KIT USER'S GUIDE

### 1. Kit Contents

The Music Keyboard Evaluation Kit contains the following items:

- C8051F700 Music Keyboard
- Toolstick Base Adapter
- Human Interface Kit CD
- USB Extender Cable
- Music Keyboard Quickstart Guide

The music keyboard EK installation package can be found at [www.silabs.com/keyboardEK](http://www.silabs.com/keyboardEK).

### 2. Hardware Setup

Refer to Figure 1 for a diagram of the hardware configuration.

1. Connect the Toolstick Base Adapter to the Music Keyboard's card edge connector.
2. Plug the Base Adapter into a PC's USB port. The red LED on the Music Keyboard should light up.

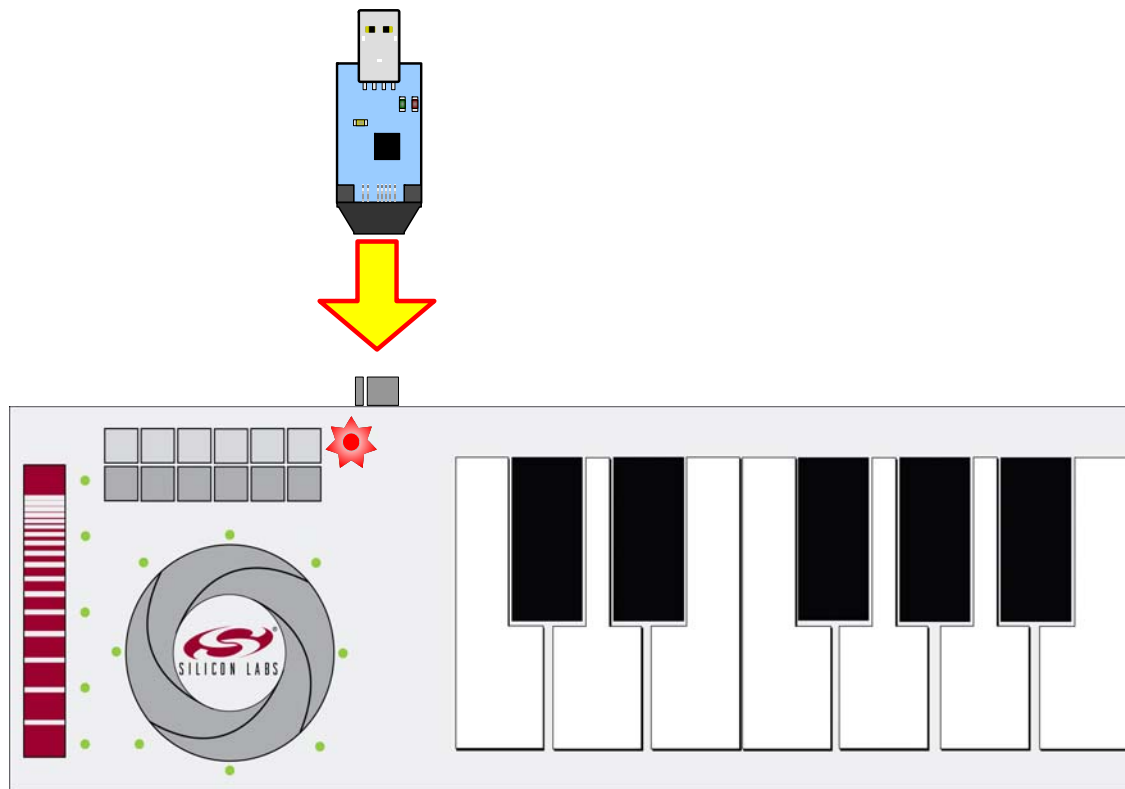


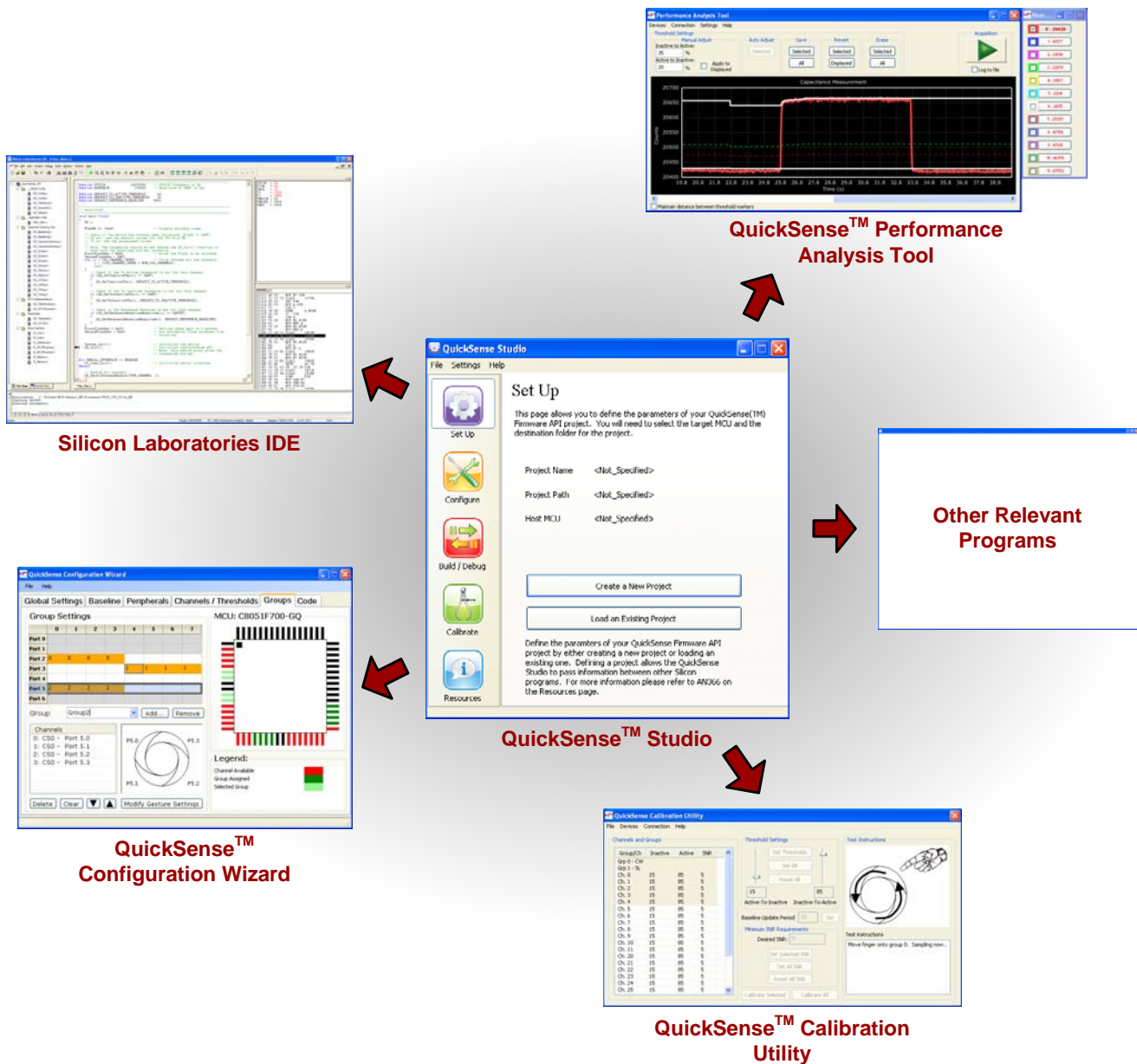
Figure 1. Hardware Setup

# MusicKeyboard-EK

## 3. PC Software Overview

### 3.1. QuickSense™ Studio

The QuickSense Studio software facilitates rapid code development and analysis for applications using the Silicon Labs family of QuickSense products. The Studio is comprised of multiple programs that guide users through an intuitive development flow including graphical configuration wizards, firmware templates, and performance monitoring tools. These programs interface with the QuickSense Firmware API, a highly configurable open-source firmware library that provides support for many different applications, from simple buttons to complex algorithms like gesture recognition.



**Figure 2. Silicon Labs QuickSense Studio Software**

For detailed information on the QuickSense Studio software, please see the QuickSense Studio User's Guide available on the Silicon Labs QuickSense webpage ([www.silabs.com/quicksense](http://www.silabs.com/quicksense)) in the QuickSense Studio section.

## 3.2. Silicon Laboratories IDE

The Silicon Laboratories IDE integrates a source-code editor, a source-level debugger, and an in-system Flash programmer.

### 3.2.1. IDE System Requirements

The Silicon Laboratories IDE requirements:

- Pentium-class host PC running Microsoft Windows 2000 or newer.
- One available USB port.

## 3.3. Performance Analysis Tool

The Performance Analysis Tool application allows users to see real-time measured capacitance values for the Music Keyboard's keys and other buttons. Figure 3 shows an example of the Performance Analysis Tool when connected to a Music Keyboard EK. To connect the Music Keyboard to the Performance Analysis Tool, follow these instructions.

1. Run the Performance Analysis Tool application, which is found by default by clicking Start→Silicon Laboratories→Performance Analysis Tool.
2. Click “Devices” and select the Toolstick Base Adapter from the pull down menu. The Base Adapter will have a name that begins with the letters “TS”.
3. Click “Connection” and then “Connect” to initialize communication with the Music Keyboard.
4. Click “Start acquisition” to begin receiving real-time data from the Music Keyboard.
5. Click on the on-screen keyboard keys to add channels to the graphing window.
6. For detailed instructions describing how to use Performance Analysis Tool, please see the application’s Help documentation.

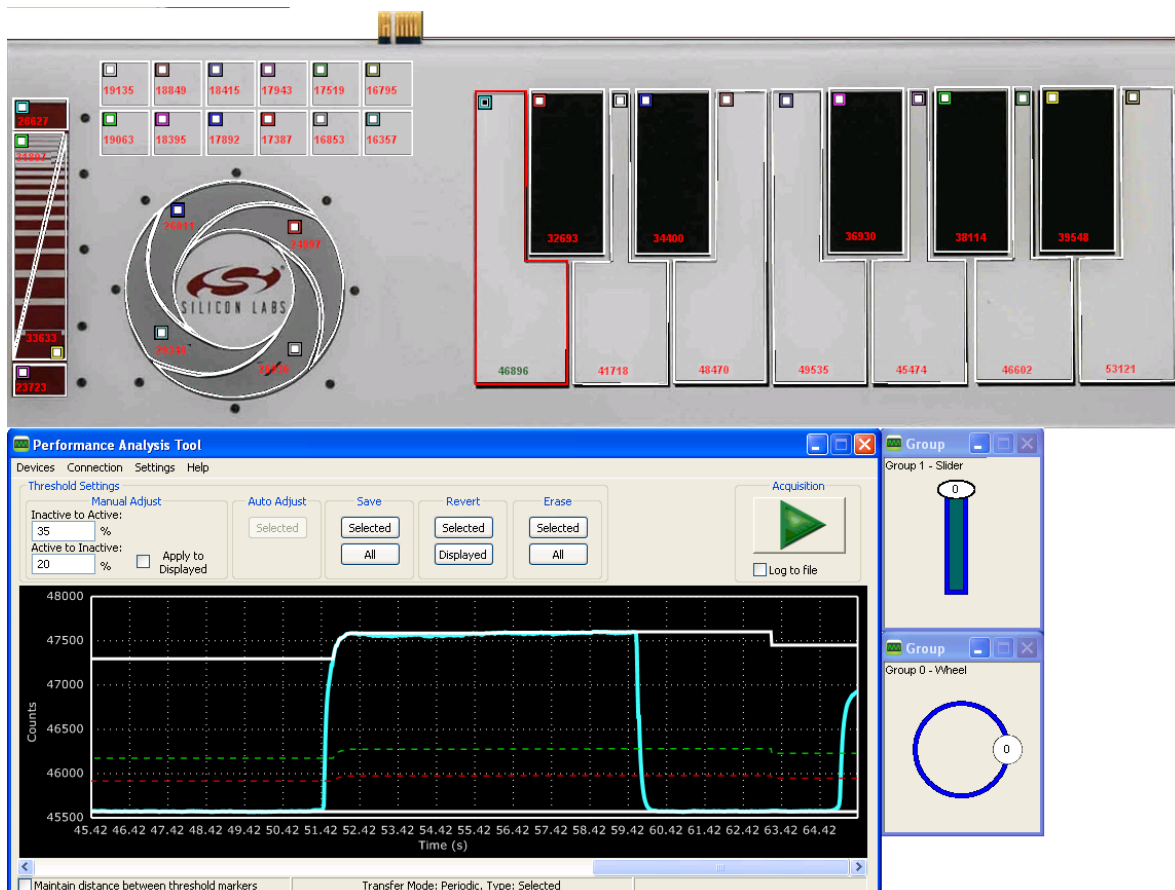


Figure 3. Performance Analysis Tool

# MusicKeyboard-EK

## 3.4. Music Keyboard Application

The Music Keyboard application plays musical notes and other sounds whenever a user presses a capacitive sensing key on the Music Keyboard. Figure 4 shows an example of the Music Keyboard application when connected to a Music Keyboard EK. To connect the Music Keyboard to the application, follow these instructions.

1. Run the Music Keyboard application, which is found by clicking Start→Silicon Laboratories→Music Keyboard App.
2. Click Options→Keyboard Input... and select the connected Toolstick Base Adapter from the list. The ToolStick Base Adapter will be a device on the list that begins with the letters "TS".

For detailed instructions describing how to use Music Keyboard Application, see the application's Help documentation

**Note:** The Music Keyboard Application uses MIDI sound information that is included with a PC's sound card. To hear notes when keys are pressed, the PC being used must have the capability of playing MIDI sounds, and the audio must be enabled and audible.

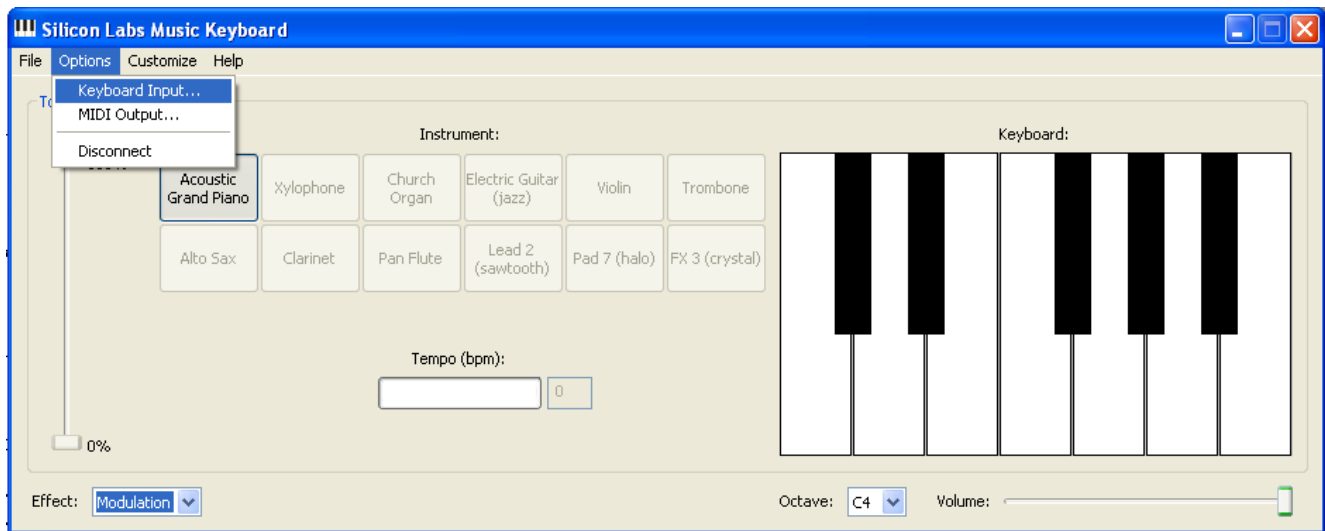


Figure 4. Music Keyboard Application

## 4. Silicon Laboratories Integrated Development Environment

The Silicon Laboratories IDE integrates a source-code editor, source-level debugger and in-system programmer. The use of third-party compilers, assemblers, and linkers is also supported. This development kit includes evaluation versions of commercial C compilers and assemblers which can be used from within the Silicon Laboratories IDE.

### 4.1. System Requirements

The Silicon Laboratories IDE requirements:

- Pentium-class host PC running Windows 2000 or later.
- One available USB port.

### 4.2. Third-Party Toolsets

The Silicon Laboratories IDE has native support for many 8051 compilers. Natively-supported tools are as follows:

- Keil
- IAR
- Raisonance
- Tasking
- SDCC

Specific instructions for integrating each of the supported tools can be found in the application notes section of the CD, or on the Silicon Labs web site (<http://www.silabs.com>).

### 4.3. Getting Started with the Silicon Labs IDE

The following sections discuss how to create a new project with the IDE, build the source code, and download it to the target device.

#### 4.3.1. Creating a New Project

1. Select **Project**→**New Project** to open a new project and reset all configuration settings to default.
2. Select **File**→**New File** to open an editor window. Create your source file(s) and save the file(s) with a recognized extension, such as .c, .h, or .asm, to enable color syntax highlighting.
3. Right-click on “New Project” in the **Project Window**. Select **Add files to project**. Select files in the file browser and click Open. Continue adding files until all project files have been added.
4. For each of the files in the **Project Window** that you want assembled, compiled and linked into the target build, right-click on the file name and select **Add file to build**. Each file will be assembled or compiled as appropriate (based on file extension) and linked into the build of the absolute object file.

**Note:** If a project contains a large number of files, the “Group” feature of the IDE can be used to organize. Right-click on “New Project” in the **Project Window**. Select **Add Groups to project**. Add pre-defined groups or add customized groups. Right-click on the group name and choose **Add file to group**. Select files to be added. Continue adding files until all project files have been added.

# MusicKeyboard-EK

---

## 4.3.2. Building and Downloading the Program for Debugging

1. Once all source files have been added to the target build, build the project by clicking on the **Build/Make Project** button in the toolbar or selecting **Project**→**Build/Make Project** from the menu.

**Note:** After the project has been built the first time, the **Build/Make Project** command will only build the files that have been changed since the previous build. To rebuild all files and project dependencies, click on the **Rebuild All** button in the toolbar or select **Project**→**Rebuild All** from the menu.

2. Before connecting to the target device, several connection options may need to be set. Open the **Connection Options** window by selecting **Options**→**Connection Options...** in the IDE menu. First, select the “USB Debug Adapter” option. Next, the correct “Debug Interface” must be selected. C8051F700 devices use Silicon Labs “C2” 2-wire debug interface. Once all the selections are made, click the **OK** button to close the window.
3. Click the **Connect** button in the toolbar or select **Debug**→**Connect** from the menu to connect to the device.
4. Download the project to the target by clicking the **Download Code** button in the toolbar.

**Note:** To enable automatic downloading if the program build is successful select **Enable automatic connect/download after build** in the **Project**→**Target Build Configuration** dialog. If errors occur during the build process, the IDE will not attempt the download.

5. Save the project when finished with the debug session to preserve the current target build configuration, editor settings and the location of all open debug views. To save the project, select **Project**→**Save Project As...** from the menu. Create a new name for the project and click on **Save**.

## 5. Example Source Code

Example source code and register definition files are provided in the “*SiLabsMCU\Examples\C8051F70x\_71x*” default directory during IDE installation. These files may be used as a template for code development. The comments in each example file indicate which development tool chains were used when testing the code.

Capacitive Sensing example code can be found by default in “*SiLabs\MCU\QuickSense\_Studio*”. The Music Keyboard firmware can be found in the “*SiLabs\MCU\QuickSense\_Studio\Kits\MusicKeyboardEK*” directory.

## 6. Schematics

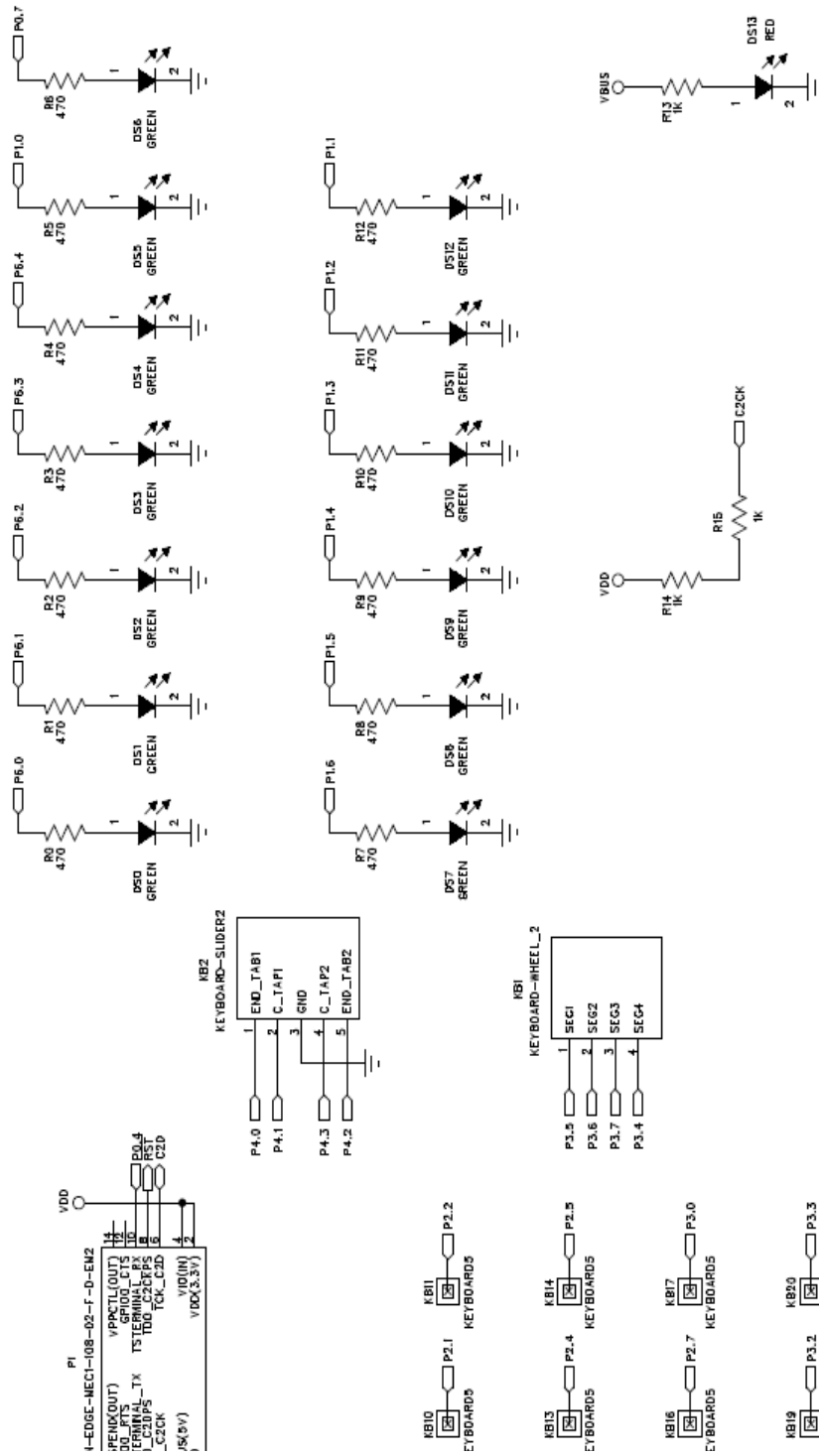


Figure 5. Music Keyboard EK Schematic (1 of 2)

# MusicKeyboard-EK

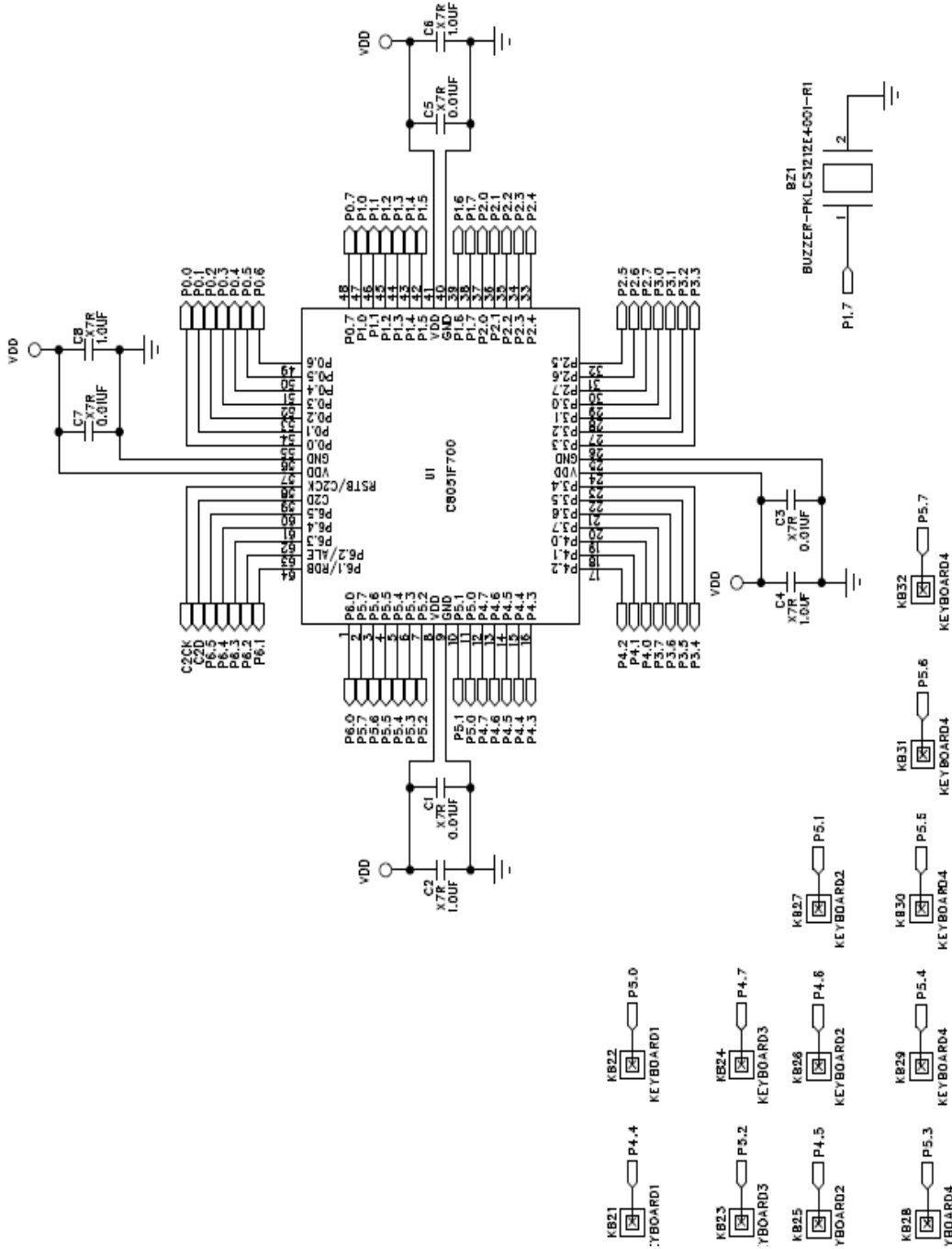


Figure 6. Music Keyboard EK Schematic (2 of 2)



**NOTES:**

# MusicKeyboard-EK

---

## CONTACT INFORMATION

Silicon Laboratories Inc.  
4635 Boston Lane  
Austin, TX 78735  
Tel: 1+(512) 416-8500  
Fax: 1+(512) 416-9669  
Toll Free: 1+(877) 444-3032

Please visit the Silicon Labs Technical Support web page:  
<https://www.silabs.com/support/pages/contacttechnicalsupport.aspx>  
and register to submit a technical support request.

The information in this document is believed to be accurate in all respects at the time of publication but is subject to change without notice. Silicon Laboratories assumes no responsibility for errors and omissions, and disclaims responsibility for any consequences resulting from the use of information included herein. Additionally, Silicon Laboratories assumes no responsibility for the functioning of undescribed features or parameters. Silicon Laboratories reserves the right to make changes without further notice. Silicon Laboratories makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Silicon Laboratories assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Silicon Laboratories products are not designed, intended, or authorized for use in applications intended to support or sustain life, or for any other application in which the failure of the Silicon Laboratories product could create a situation where personal injury or death may occur. Should Buyer purchase or use Silicon Laboratories products for any such unintended or unauthorized application, Buyer shall indemnify and hold Silicon Laboratories harmless against all claims and damages.

Silicon Laboratories and Silicon Labs are trademarks of Silicon Laboratories Inc.

Other products or brandnames mentioned herein are trademarks or registered trademarks of their respective holders