

# MAX32600 USB HID Keyboard Demonstration

---

*April 2015*

©2015 Maxim Integrated Products, Inc.  
All rights reserved.

No part of this documentation may be reproduced nor distributed in any form or by any means, graphic, electronic, or mechanical, including but not limited to photocopying, scanning, recording, taping, e-mailing, or storing in information storage and retrieval systems without the written permission of Maxim Integrated Products, Inc. (hereafter, "Maxim"). Products that are referenced in this document such as Microsoft Windows® may be trademarks and/or registered trademarks of their respective owners. Maxim makes no claim to these trademarks. While every precaution has been taken in the preparation of this document, individually, as a series, in whole, or in part, Maxim, the publisher, and the author assume no responsibility for errors or omissions, including any damages resulting from the express or implied application of information contained in this document or from the use of products, services, or programs that may accompany it. In no event shall Maxim, publishers, authors, or editors of this guide be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

April 2015

# 1 Abstract

This document describes the USB HID Keyboard sample application provided for the MAX32600. This application demonstrates how to configure a the USB device controller as a HID keyboard class device.

# 2 Requirements

- MAX32600 EvKit
- Sample code for this application located in `Firmware/Applications/USBHIDKeyboardDemo`
- Olimex JTAG ARM-USB-TINY-H
- GNU ARM toolchain

# 3 Setup

- Load the compiled `max32600.elf` file onto the MAX32600 EvKit.
- Connect a USB cable from CN1 to a host PC.

# 4 Observation

- The EvKit should enumerate as a HID Keyboard device.
- The yellow LED will illuminate once enumeration and configuration is complete.
- Pressing pushbutton SW1 will cause a message to be typed in on a virtual keyboard.

## **5 Source Code Overview**

### **5.1 Drivers In Use**

- MAXUSB
- Instruction Cache
- Clock Manager
- Power Manager
- UART
- GPIO

### **5.2 Interrupts Enabled**

- USB
- UART

### **5.3 Code Operation**

- Enable Instruction Cache
- Setup Clocks
- Enable the RTC clock in 'run' mode and drive systick
- Enable USB
- Wait for interrupts
- Send characters received from UART to USB
- Send characters received from USB to UART