17741 MITCHELL NORTH, IRVINE CALIFORNIA 92614 USA
PHONE: (949) 833-3355 FAX: (949) 833-0322
INTERNET: http://www.genovation.com
Email: support@genovation.com

ControlPad 682/683 LED Programming



Developer Application Note

Copyright © 2008, Genovation, Inc.

Manual Last Updated February 25, 2008

NOTICE

Copyright © 2008, Genovation, Inc. All rights reserved. No part of this document may be photocopied, transmitted or reproduced in any form without prior written consent from Genovation, Inc.

Genovation, Inc. reserves the right to revise this documentation and to make changes in content from time to time without obligation on the part of Genovation, Inc. to provide notification of such revision or changes.

GENOVATION, INC. SHALL NOT BE LIABLE FOR TECHNICAL OR EDITORIAL ERRORS OR OMISSIONS HEREIN; NOR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL OR ACCOMPANYING SOFTWARE.

Product names mentioned in this document may be trademarks and/or registered trademarks of other companies.
“Windows” is a registered trademark of Microsoft Corporation.

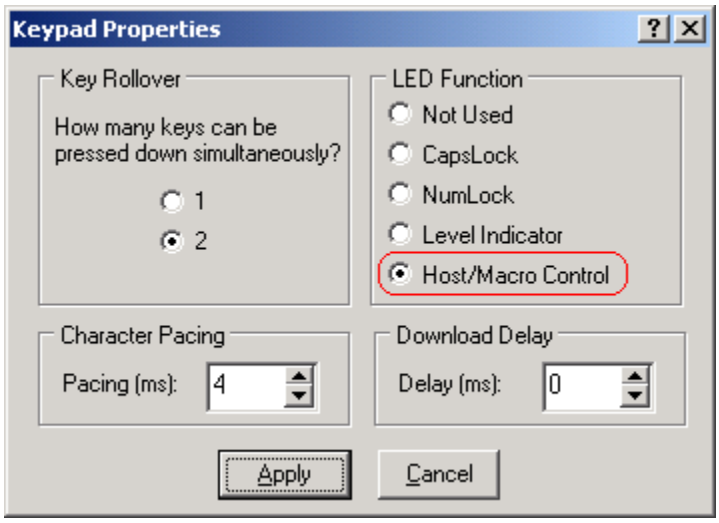
For the latest version of this document, visit
http://www.genovation.com

The sample code is provided on an "as is" basis, and Genovation, Inc. expressly disclaims any or all warranties expressed or implied, including without limitation warranties of fitness for a particular purpose. In no event shall Genovation, Inc. be liable for any direct, indirect, incidental consequential damages of any kind whatsoever with respect to the use of the code.

Genovation, Inc. reserves the right, in its sole discretion and without any obligation, to make improvements to, or correct any error in any portion of the code.

Introduction

Starting with firmware version 5.0 of the ControlPad series, the LED state can be controlled from within a macro and/or from the Host PC. This allows third party applications to use the indicator for their own purposes. For example, a CNC Pendant controller could show a state on the LED or an email monitoring application could flash the LED on new email arriving.



Command Set

The command set is very simple for the device and is supported over PS/2 or USB. Since the Windows OS makes very it difficult to access the keyboard port directly, this document focuses mainly on a USB implementation. If you are able to access the keyboard port with your software then you may also control the LED over PS/2. All values in the table below are hexadecimal.

Command Byte	Parameter Byte	Comment
E4	04	Turn the LED off
E4	05	Turn the LED on
E4	06	Flash the LED

The LED will stay in the state it is set to until it is changed by another command or by a macro containing an LED code.

Included Examples

1. ControlPadLed Tray Icon application. This application places an LED icon in the system tray and allows the user to manually set the LED state. This application also accepts Windows messages to control the LED.

2. ControlPadLed Tools. These three small applications send Window messages to the ControlPadLED tray icon application.

3. ControlPadLED.dll. Dynamic Link Library that exports one function to control the LED.

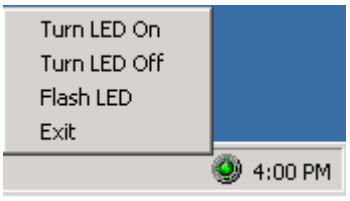
4. TestControlPadLED.exe application. This is a console application that uses the DLL to control the LED.

In all cases the project files used to build the applications is included.

Sample Application Capabilities

The archive file set, `ControlPadLED.rar` contains an application that accesses the LED directly over USB. This application allows a person to set the LED using a menu from the system tray **and** this application accepts inter-process communications (messages) from other applications to also control the LED.

o Using the menu in the system tray:



o Using messages from another application:

```
// Lparam: 06 = flash the led, 05 = turn it on, 04 = turn it off.

HANDLE Wnd;

Wnd = FindWindow(NULL, "Genovation ControlPadLED");
if(Wnd)
   PostMessage(Wnd, WM_APP+0x0683, (WPARAM)0x00E4, (LPARAM)0x0006);
```

Accessing the device over USB

The included module USB.cpp provides a basic framework to sending bytes to the ControlPad LED. It contains the following functions:

```
bool UsbOpenKeypad(void);
void UsbCloseKeypad(void);
bool UsbWriteChar(unsigned char byte);
```

Therefore, to send a command to the LED, the calling code looks like:

```
if( UsbOpenKeypad() )
   {
   UsbWriteChar(0xE4);
   UsbWriteChar(0x05);
   UsbCloseKeypad();
   }
```

The rest of the application consists of looking for right mouse clicks on the system tray icon (a green LED bitmap) and looking for custom messages that are sent from another application.

Sample Tools

The sample package also includes three simple applications that utilize the inter-process communications feature of ControlPadLED.exe. These applications are:

- Flash.exe
- LedOff.exe
- LedOn.exe

The operation of these programs works as follows:

Flash.exe >> ControlPadLED.exe >> OS:USB >> Controlpad 68x hardware

Sample DLL

The function exported by the DLL:

```
extern "C" __declspec(dllexport) unsigned short
ControlPadLED_Cmd(unsigned char *pCommand, unsigned short length);
```

Using the DLL:

```
unsigned char LedFlash[2] = { 0xE4, 0x06 };
ControlPadLED_Cmd(LedFlash, 2);
```