APPLICATION NOTE

USING THE ST7263 KIT TO IMPLEMENT A USB GAME PAD

by Microcontroller Division Applications

1 INTRODUCTION

The game pad described in this application note is a low speed, self powered device. It has digital and analog capabilities for the X and Y axes, and includes 10 buttons and two motors for vibration.

In order to demonstrate the use of the ST7263 microcontroller, we adapted this game pad to the ST7263 demo kit to make a USB game pad, with USB mouse and hotkey functions.

This application uses the key features of the ST7263 USB microcontroller, the analog converter, the 3 USB Endpoints (control Endpoint, interrupt IN, interrupt OUT) and PWM capabilities with the two output compare waveforms on Port A.

The source code of the software described on this application note is available from ST.

As with other USB devices, Endpoint 0 is the control Endpoint used for device enumeration. Endpoint 1 is an interrupt IN Endpoint, this allows the device to send data to the PC. This Endpoint is shared using the reportID capabilities specific to the HID class. In this case, the first byte sent to the PC is the reportID number. In this demo application we use three reportIDs:

– reportID number 1 is used for the game pad data

– reportID number 2 is used for the mouse data

– reportID number 3 for the hotkey function

Endpoint 2 is an interrupt OUT Endpoint to send data from the PC to the device. This Endpoint is used to control the motors for the vibration feature.

2 GAME PAD HARDWARE

Figure 1. Game Pad Controls

L2 BUTTON

ANALOG BUTTON

R2 BUTTON

L1 BUTTON

R1 BUTTON

B1 BUTTON

DIGITAL AXES

B2 BUTTON

B3 BUTTON

SELECT BUTTON

B4 BUTTON

ANALOG X AXIS

LED FOR ANALOG STATE

START BUTTON

ANALOG Y AXIS
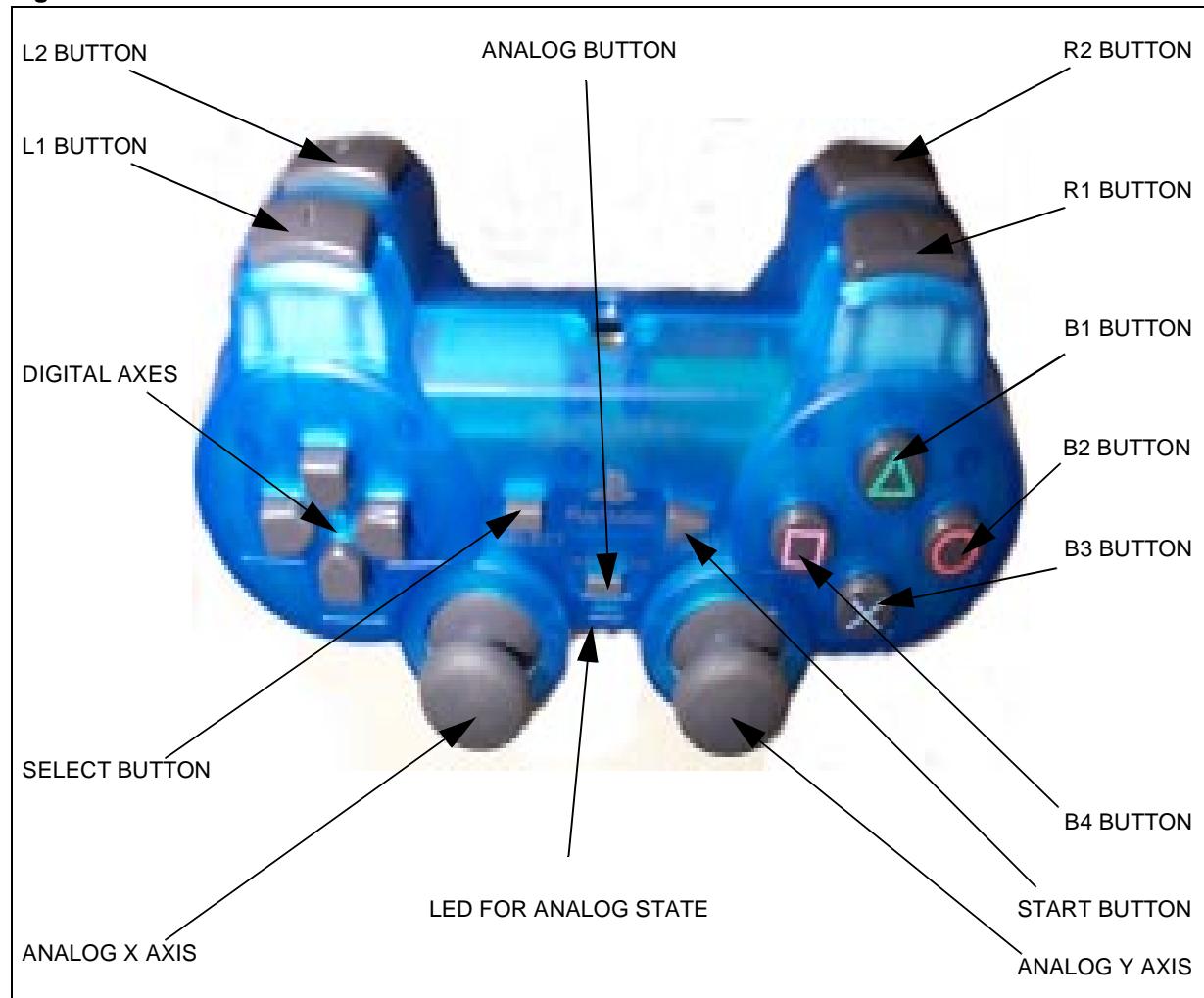
2.1 SELECT BUTTON

This button is used to toggle between game pad mode and hotkey mode. You can not use it as a game pad button for games in game pad mode, and you can not use it as a hotkey function in hotkey mode.

Note: Toggling between the two modes can also be controlled from the Windows application on the PC (see Figure 5).

2.2 ANALOG BUTTON

This button can be used only in game pad mode. It has no effect in hotkey mode. By pushing the analog button, the analog axes are enabled, the analog LED is switched on and the digital

axes are switched off. To return to digital mode, press the analog button again and check that the LED is switched off.

2.3 GAME PAD MODE

At start-up, game pad mode is enabled (hotkey mode disabled) as are digital axes (analog axes disabled). You can open the game controller properties dialog box to test the operation of the different buttons, and check the X and Y axes.

In game pad mode, you have 9 buttons: L1, L2, R1, R2, B1, B2, B3, B4 and Start.

Figure 2. Windows 98 Control panel
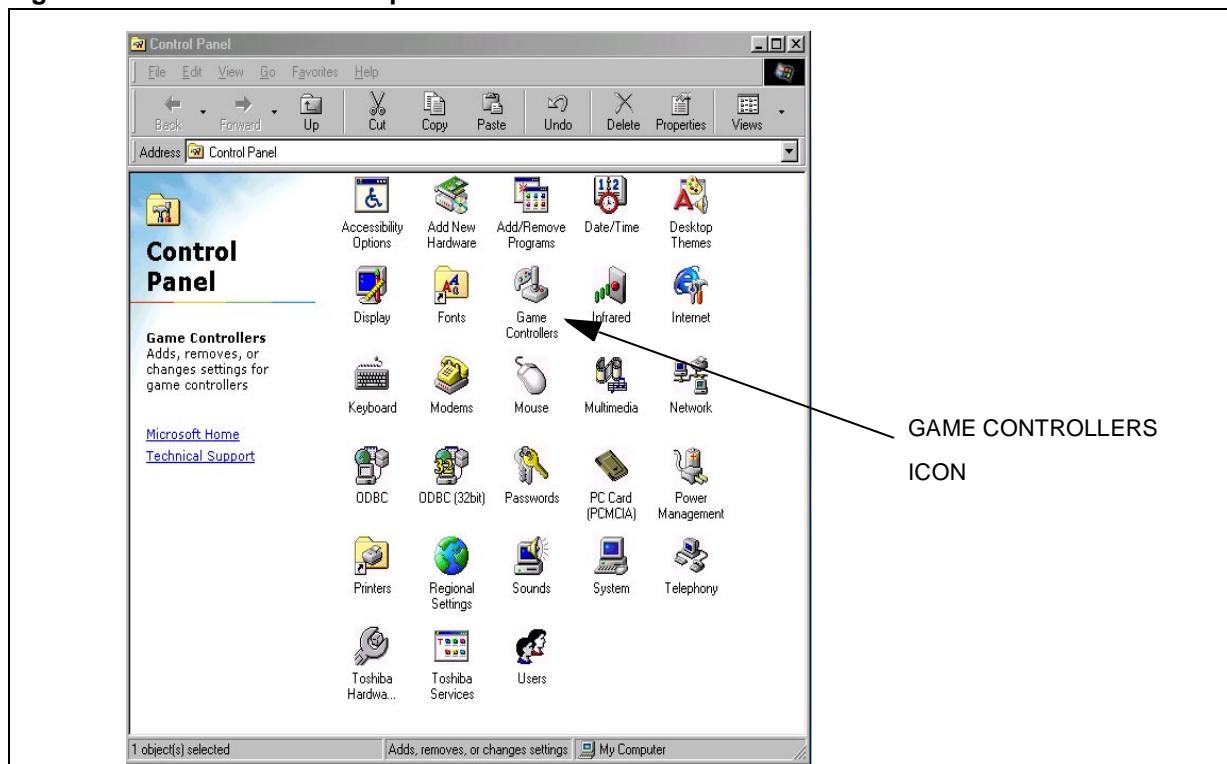
### 2.3.1 How to open the game controller properties dialog box

In the control panel window, click on the game controller icon, select the USB game pad and click on the Properties button, then click on the Test tab to see the dialog box shown in Figure 3.

**Figure 3. Game Controller Properties Dialog Box**
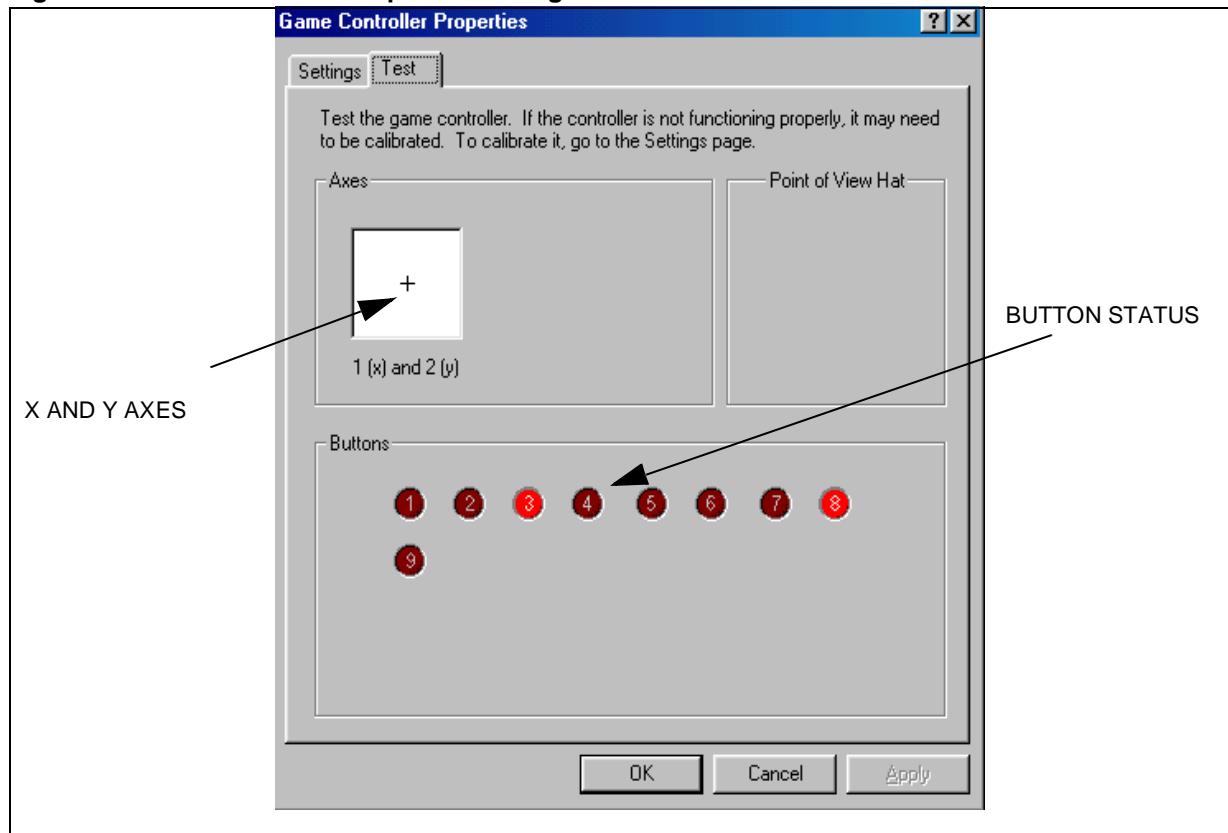
With the demo there is an associated windows application: hotkey.exe. This application is displayed in the Windows 98 taskbar (see Figure 4)

Figure 4. Hotkey Application Icon
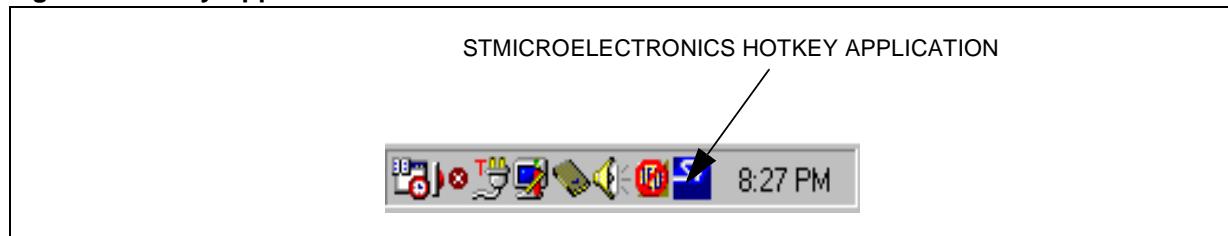
When you plug the demo to the USB port, the hotkey application will open the Dual-shock game pad dialog box (see Figure 5).

Figure 5. USB game pad dialog box
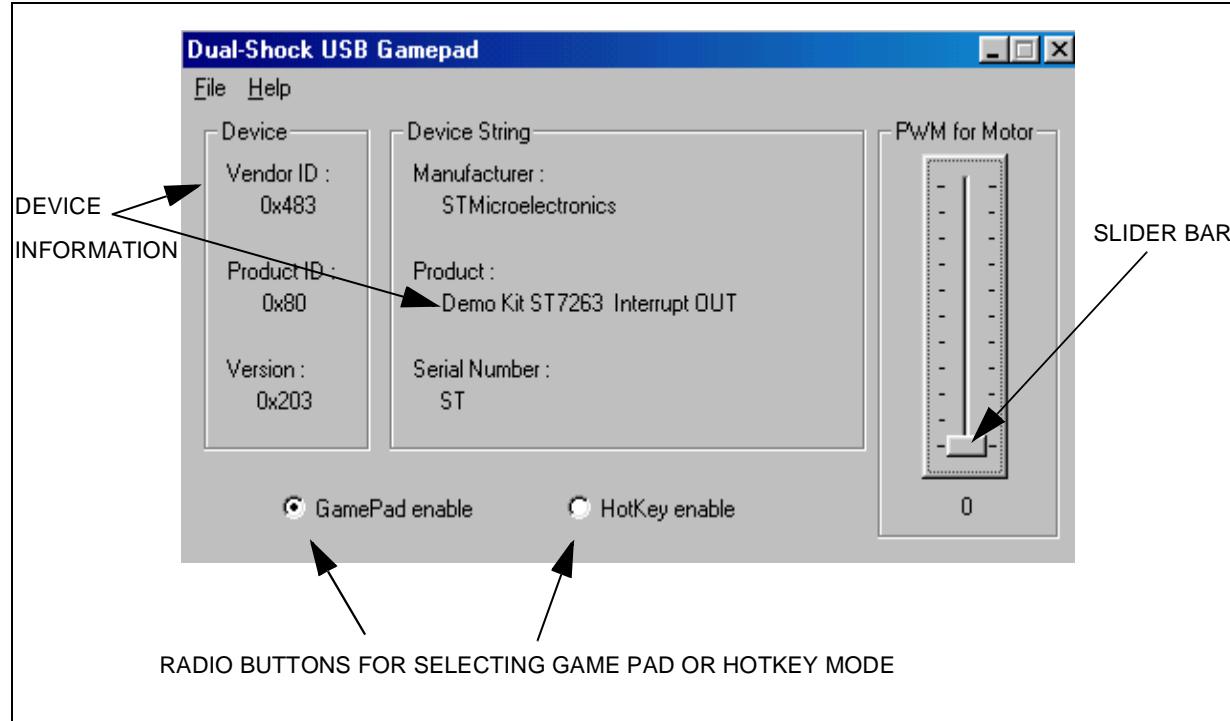
To close the dialog box, choose Close in the File menu. This will close the dialog box but the hotkey application is still running and the ST icon remains displayed in the taskbar. To close the application, choose Exit from the File menu, and the ST icon will disappear.

Once the game pad is plugged-in, a left click to the ST icon in the taskbar will open the dual-shock dialog box, a right click opens the about window.

The slider bar allows you to generate a PWM waveform to drive the vibration motors if the game pad radio button is selected.

With the radio button you can select game pad mode or hotkey mode. If you press the (physical) select button on the game pad, the select status displayed in the radio buttons is automatically updated via the USB bus.

Hotkey Mode:

Hotkey mode is enabled by clicking the hotkey radio button in the dialog box, or by pressing the select button on the game pad. In this case, the game pad function is disabled and the arrows for the digital X and Y axes are used for the USB mouse. The L1 and L2 buttons are used for the left and right click of the mouse. The analog mode is not functional and the B1, B2, B3 and Start buttons are used for the hotkey functions described below.

The reportID number 1 reports the data for the game pad. Using the Endpoint 1 for X, Y axes and the 9 buttons. Using the Endpoint 2 for the vibration of the motor.

**Figure 6. Example of data sent through Endpoint 1 (IN Endpoint)**

| 01 | 81 | 81 | 00 | 00 |

ReportID NUMBER   X AXIS   Y AXIS   BUTTON 9

BUTTONS 1 TO 8
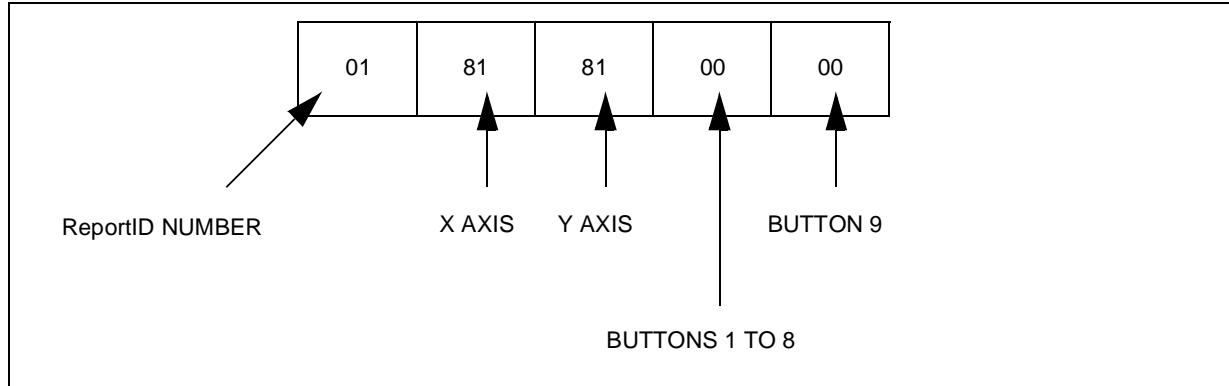
Figure 7. Example of data sent through Endpoint 2 (OUT Endpoint)

| 01 | 20 |

ReportID NUMBER   PWM VALUE
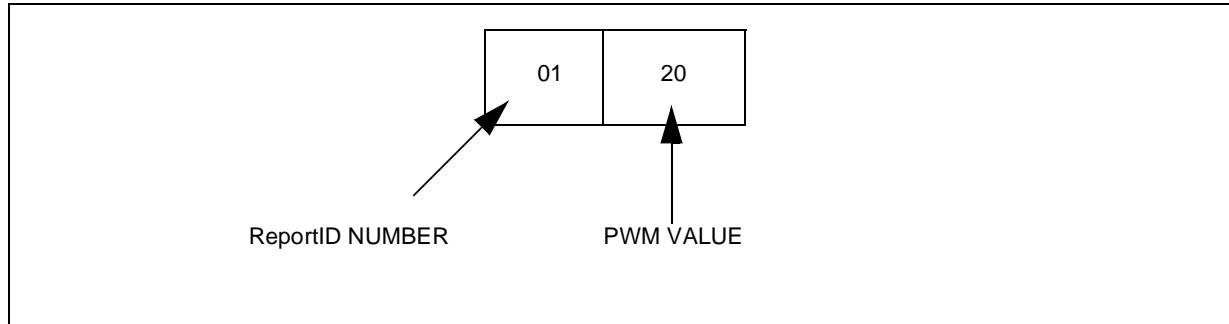
The interrupt OUT for low speed device is only supported from Windows 98 Second Edition (build 4.10.2222 A) or later: Windows 2000.

To check the version of Windows you have click on the system icon in the control panel (see Figure 8).

The other buttons have no effect.

Table 1. Hotkey Assignments

Button	Path
Start	Internet browser (Netscape or Internet explorer):
	Netscape: C:/Program Files/Netscape/Communicator/Program/netscape.exe
	Internet Explorer: C:/Program Files/Internet Explorer/iexplore.exe
B1	Windows Explorer: C:/Windows/explorer.exe
B2	Solitaire game: C:/Windows/sol.exe
B3	Notepad: C:/Windows/notepad.exe

Note: In the current version of the demo software, the paths assigned to the hotkey functions are not programmable.

2.4 REPORT DESCRIPTOR USING REPORTID

```
0x05, 0x01,              // USAGE_PAGE (Generic Desktop)
0x09, 0x05,              // USAGE (Game Pad)
  0xa1, 0x01,            //  COLLECTION (Application)
  0x85, 0x01,            //   REPORT_ID (1)
  0x09, 0x01,            //   USAGE (Pointer)
  0xa1, 0x00,            //   COLLECTION (Physical)
  0x09, 0x30,            //    USAGE (X)
  0x09, 0x31,            //    USAGE (Y)
  0x15, 0x00,            //    LOGICAL_MINIMUM (0)
  0x26, 0xff, 0x00,      //    LOGICAL_MAXIMUM (255)
  0x35, 0x00,            //    PHYSICAL_MINIMUM (0)
  0x46, 0xff, 0x00,      //    PHYSICAL_MAXIMUM (255)
  0x75, 0x08,            //    REPORT_SIZE (8)
  0x95, 0x02,            //    REPORT_COUNT (2)
  0x81, 0x02,            //    INPUT (Data,Var,Abs)
  0xc0,                  //   END_COLLECTION
  0x05, 0x09,            //   USAGE_PAGE (Button)
  0x15, 0x00,            //   LOGICAL_MINIMUM (0)
  0x25, 0x01,            //   LOGICAL_MAXIMUM (1)
  0x35, 0x00,            //   PHYSICAL_MINIMUM (0)
  0x45, 0x01,            //   PHYSICAL_MAXIMUM (1)
  0x75, 0x01,            //   REPORT_SIZE (1)
  0x95, 0x0a,            //   REPORT_COUNT (10)
  0x19, 0x01,            //   USAGE_MINIMUM (Button 1)
  0x29, 0x0a,            //   USAGE_MAXIMUM (Button 10)
  0x81, 0x02,            //   INPUT (Data,Var,Abs)
  0x75, 0x01,            //   REPORT_SIZE (1)
  0x95, 0x06,            //   REPORT_COUNT (6)
  0x81, 0x01,            //   INPUT (Cnst,Ary,Abs)
  0x05, 0x8c,            //   USAGE_PAGE (ST Page)
  0x09, 0x05,            //   USAGE (PWM out)
  0x95, 0x01,            //   REPORT_COUNT (1)
  0x75, 0x08,            //   REPORT_SIZE (8)
  0x15, 0x00,            //   LOGICAL_MINIMUM (0)
  0x25, 0x64,            //   LOGICAL_MAXIMUM (100)
  0x91, 0x02,            //   OUTPUT (Data,Var,Abs)
  0xc0,                  // END_COLLECTION
  0x05, 0x01,            // USAGE_PAGE (Generic Desktop)
  0x09, 0x02,            // USAGE (Mouse)
  0xa1, 0x01,            // COLLECTION (Application)
  0x85, 0x02,            //  REPORT_ID (2)
  0x09, 0x01,            //  USAGE (Pointer)
```

```
0xa1, 0x00,           //  COLLECTION (Physical)
0x05, 0x09,           //   USAGE_PAGE (Button)
0x19, 0x01,           //   USAGE_MINIMUM (Button 1)
0x29, 0x03,           //   USAGE_MAXIMUM (Button 3)
0x15, 0x00,           //   LOGICAL_MINIMUM (0)
0x25, 0x01,           //   LOGICAL_MAXIMUM (1)
0x95, 0x03,           //   REPORT_COUNT (3)
0x75, 0x01,           //   REPORT_SIZE (1)
0x81, 0x02,           //   INPUT (Data,Var,Abs)
0x95, 0x01,           //   REPORT_COUNT (1)
0x75, 0x05,           //   REPORT_SIZE (5)
0x81, 0x03,           //   INPUT (Cnst,Var,Abs)
0x05, 0x01,           //   USAGE_PAGE (Generic Desktop)
0x09, 0x30,           //   USAGE (X)
0x09, 0x31,           //   USAGE (Y)
0x15, 0x81,           //   LOGICAL_MINIMUM (-127)
0x25, 0x7f,           //   LOGICAL_MAXIMUM (127)
0x35, 0x81,           //   PHYSICAL_MINIMUM (-127)
0x45, 0x7f,           //   PHYSICAL_MAXIMUM (127)
0x75, 0x08,           //   REPORT_SIZE (8)
0x95, 0x02,           //   REPORT_COUNT (2)
0x81, 0x07,           //   INPUT (Cnst,Var,Rel)
0xc0,                 //  END_COLLECTION
0xc0,                 // END_COLLECTION
0x05, 0x8c,           // USAGE_PAGE (ST Page)
0x09, 0x01,           // USAGE (Demo Kit)
0xa1, 0x01,           // COLLECTION (Application)
0x85, 0x03,           //  REPORT_ID (3)
0x09, 0x0e,           //  USAGE (HotKey)
0x15, 0x00,           //  LOGICAL_MINIMUM (0)
0x26, 0xff, 0x00,      //  LOGICAL_MAXIMUM (255)
0x35, 0x00,           //  PHYSICAL_MINIMUM (0)
0x46, 0xff, 0x00,      //  PHYSICAL_MAXIMUM (255)
0x75, 0x08,           //  REPORT_SIZE (8)
0x95, 0x01,           //  REPORT_COUNT (1)
0x81, 0x02,           //  INPUT (Data,Var,Abs)
0x85, 0x03,           //  REPORT_ID (3)
0x09, 0x0e,           //  USAGE (HotKey)
0xb1, 0x02,           //  FEATURE (Data,Var,Abs)
0xc0                  // END_COLLECTION
```

Figure 8. Windows System Properties
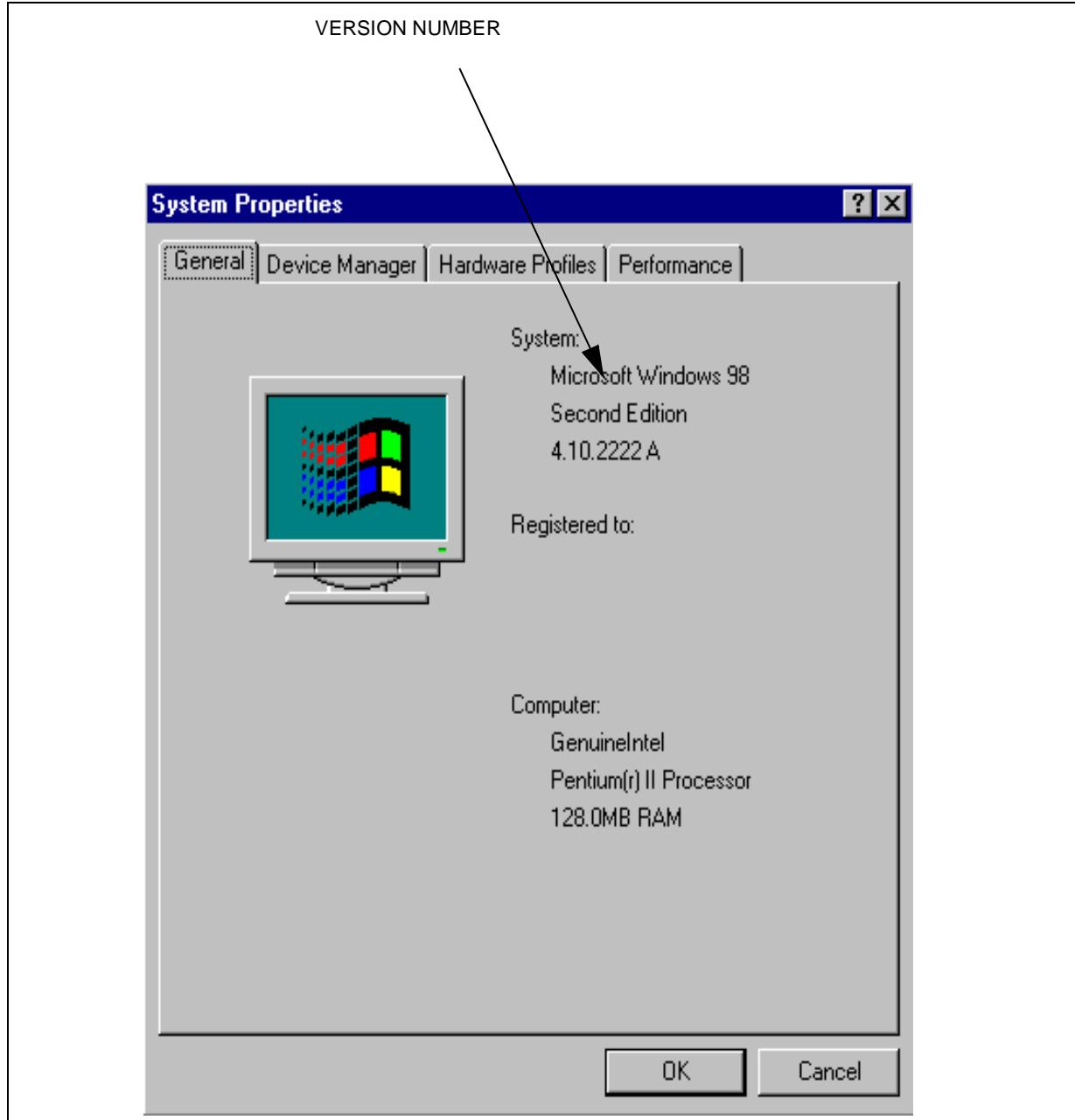
To send information to the device, the windows application uses a function defined in the HID driver. If the use of the OUT Endpoint is allowed, (this is the case for Windows Second Edition and later) the application will send data through the interrupt OUT Endpoint. If the use of the OUT Endpoint is not allowed, (in the case of Windows first edition: build 4.10.1998), the application will use the SET_OUTPUT specific request through Endpoint 0.

Figure 9. Data sent through Endpoint 2 (in Windows 98 Second Edition)
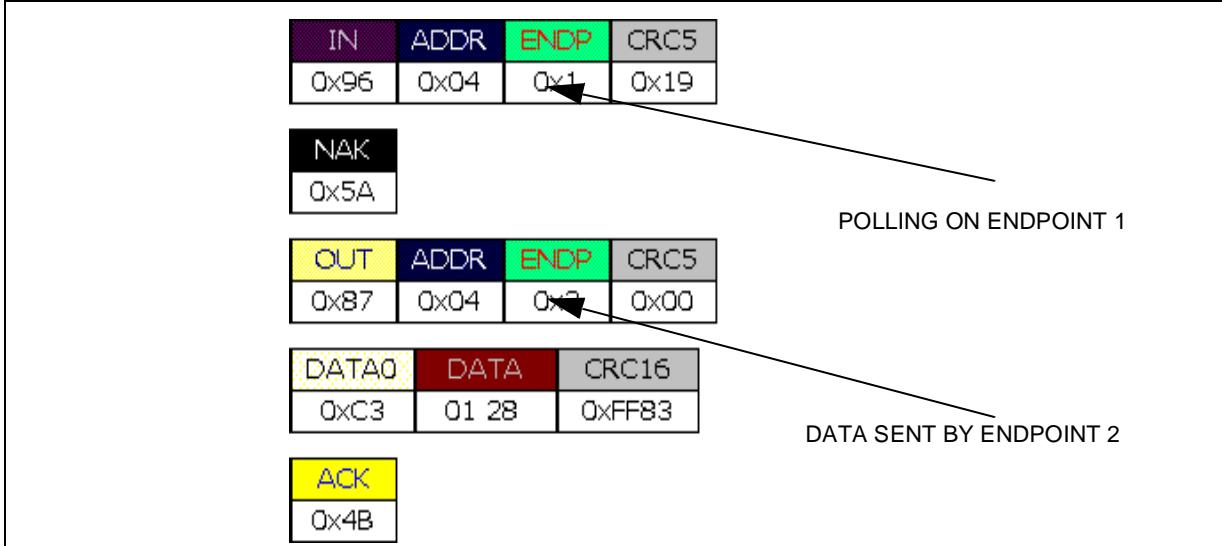
Figure 10. Data sent through Endpoint 0 (in Windows 98 version 4.10.1998)
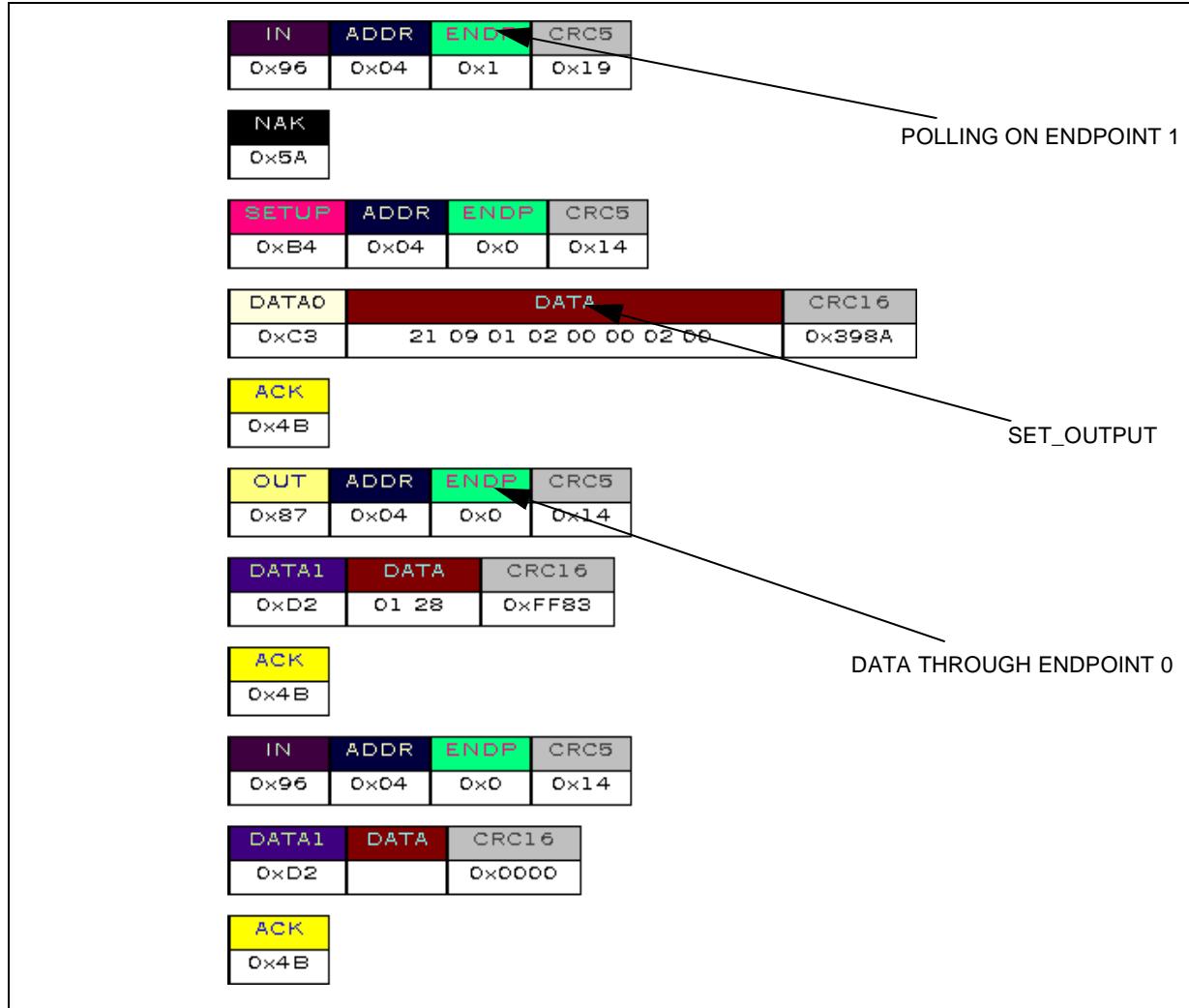
ReportID 2 is used for the mouse data. The first byte is the reportID number, the second one is the X axis, the third one is the Y axis and the fourth one is the button status

Figure 11. Data for the mouse
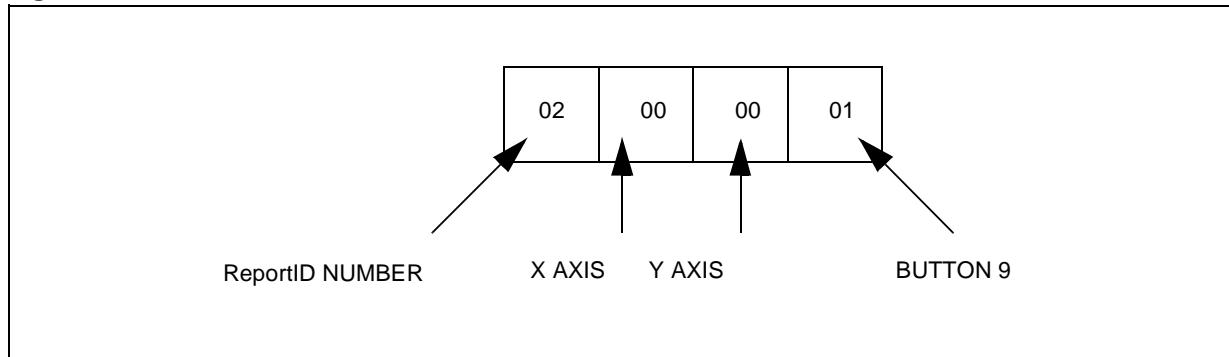
ReportID 3 is for the hotkey function. We use only one byte to send the hotkey value. To switch from game pad mode to hotkey mode, we send 0xFD. To switch from hotkey mode to game pad mode we send 0xFF.

Figure 12. Hotkey mode enable
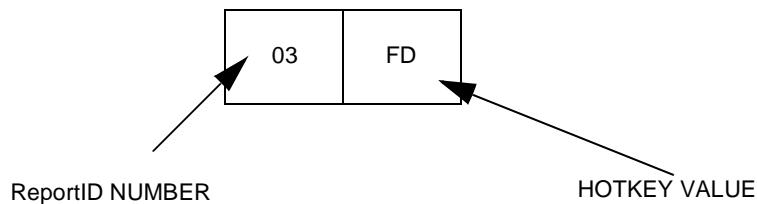
Figure 13. Game Pad mode enable
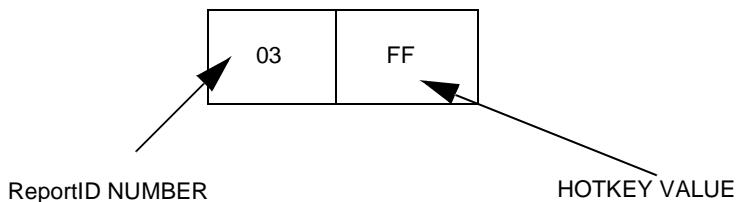
Figure 14. Interface to the ST7263 demokit
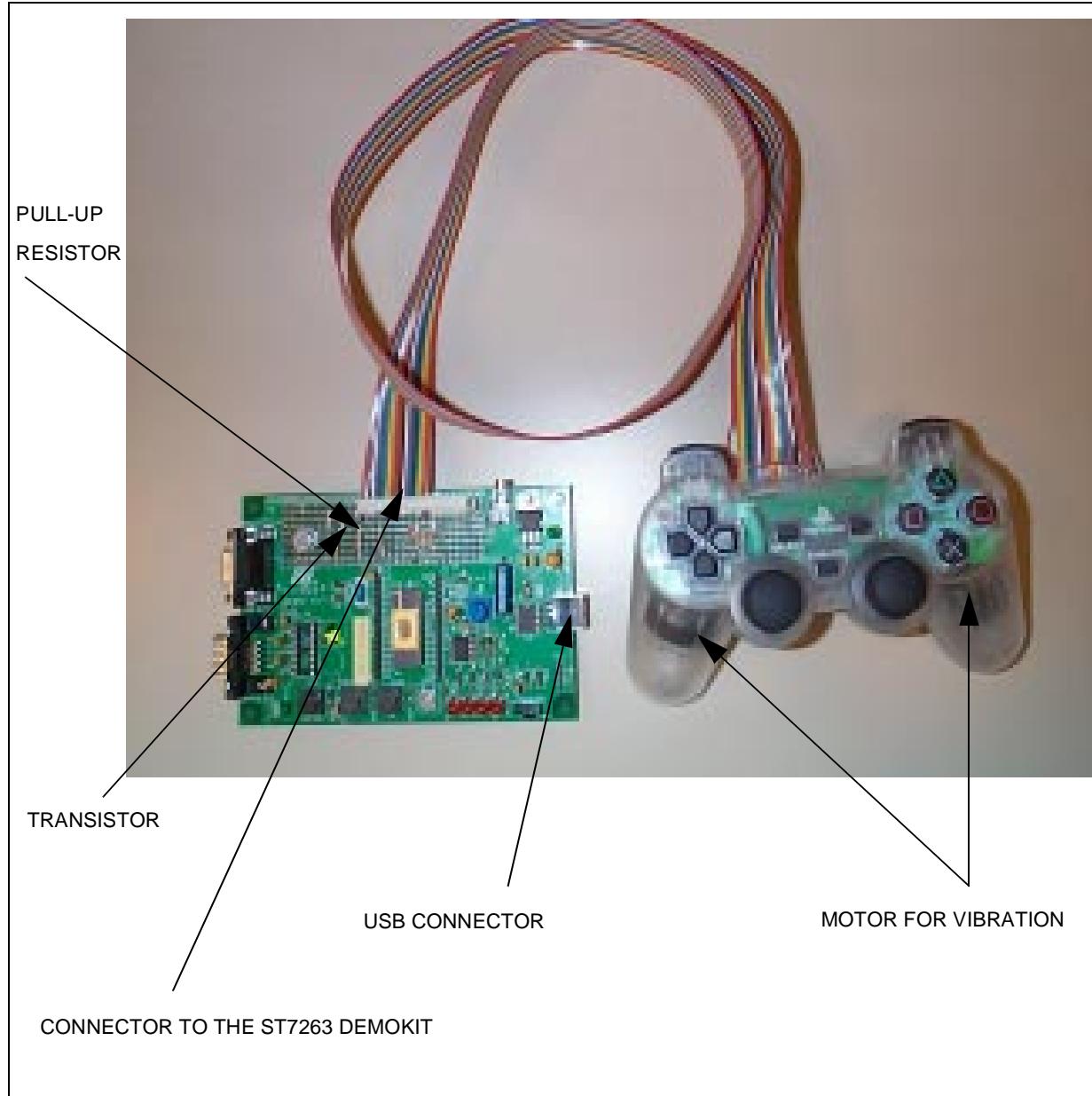
Figure 15. Board schematics and hardware description
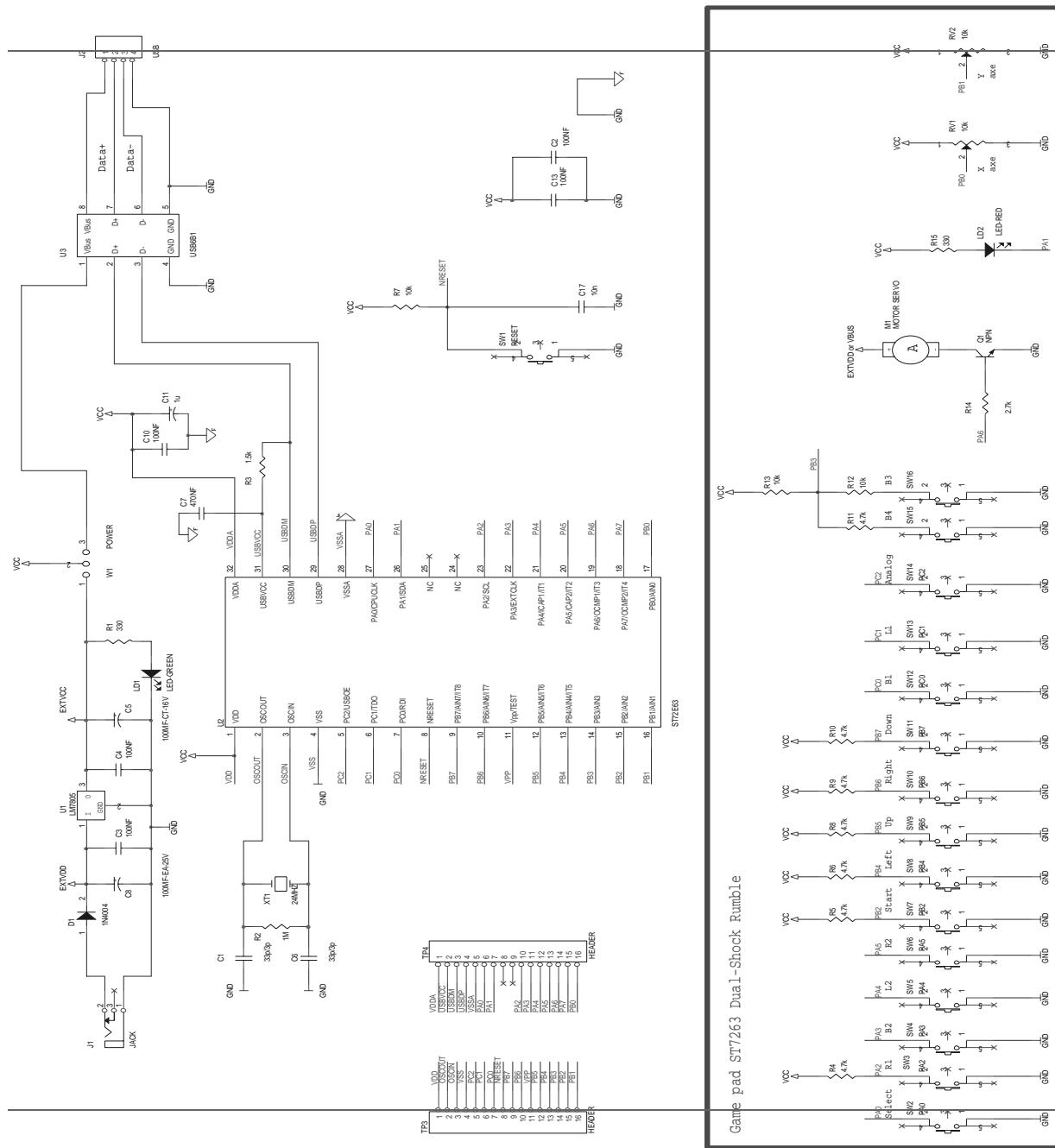
There are two types of PCB used to manufacture the game pad. Depending on the PCB type, the pinout of the connection to the demokit is not the same.

Figure 16. PCB 1 (2 lines)

CONNECTOR FOR BUTTONS
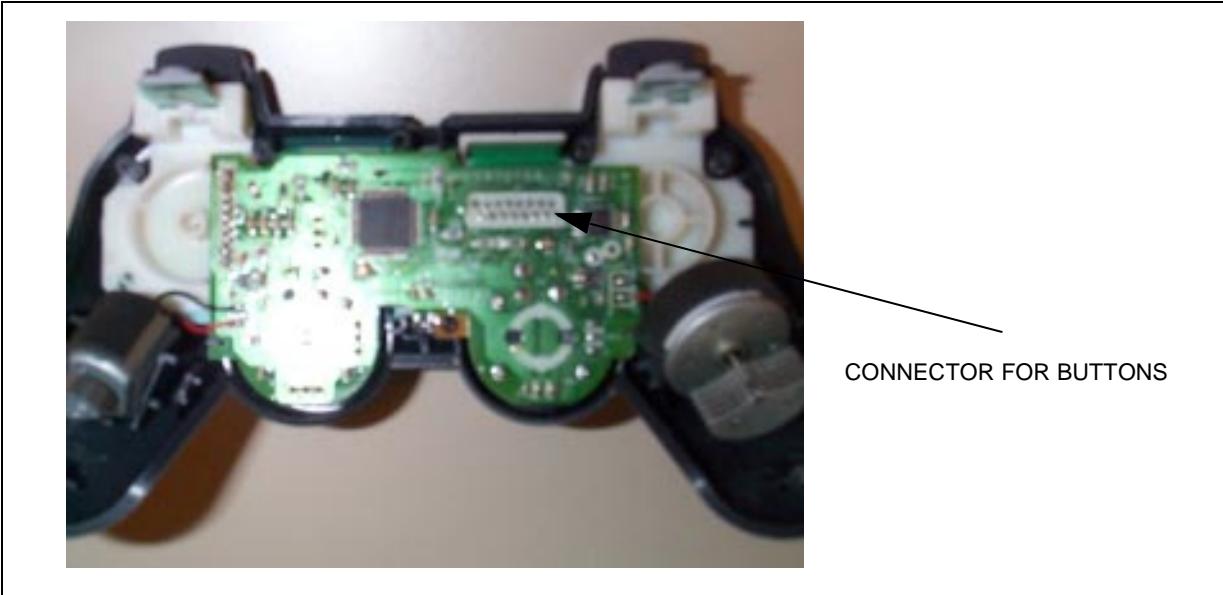
For this PCB, the pinout is the following:

1	3	5	7	9	11	13	15
2	4	6	8	10	12	14	16

```
1 : R2 (-> PA5)
2 : R1 (-> PA2)
3 : B1 (-> PC0)
4 : B2 (-> PA3)
5 : B3 (-> PB3 through 10K)
6 : B4 (-> PB3 through 4.7K)
7 : Start (-> PB2)
8 : Select (-> PA0)
9 : Analog (-> PC2)
10 : GND
11 : Left (-> PB4)
12 : Down (-> PB7)
13 : Right (-> PB6)
14 : Up (-> PB5)
15 : L1(-> PC1)
16 : L2 (-> PA4)
```

Figure 17. PCB 2 (1 line)

CONNECTOR FOR BUTTONS
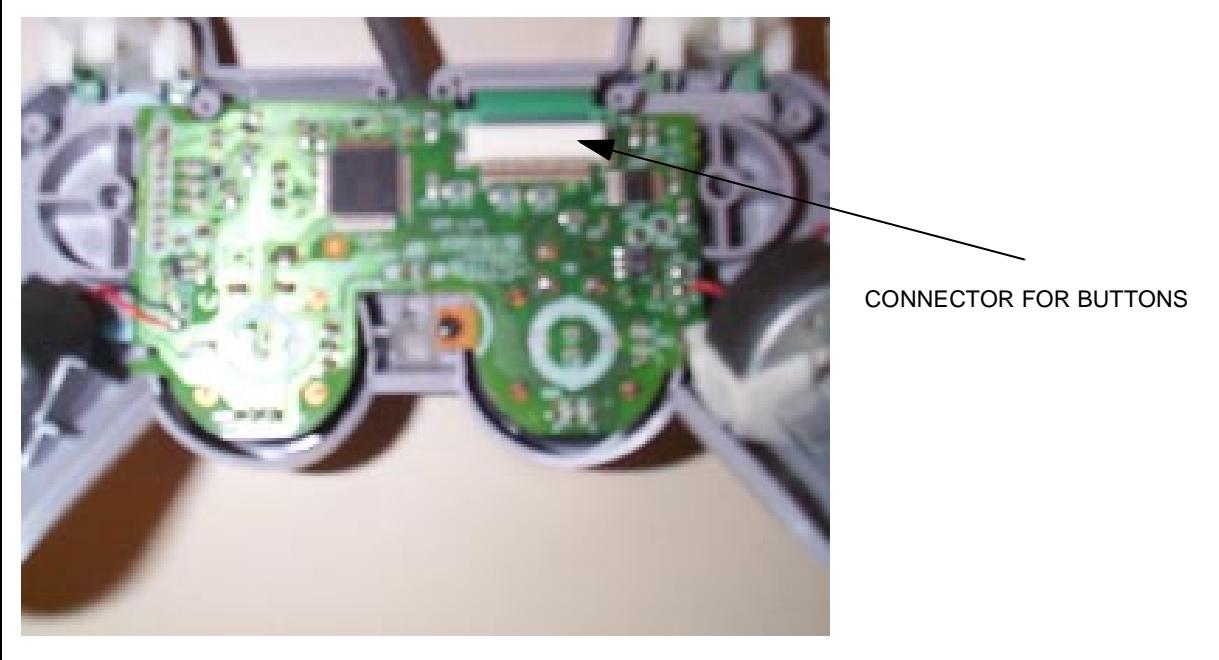
For this PCB the pinout is the following:

```
            1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
```

```
1 : R2 (-> PA5)
2 : R1 (-> PA2)
3 : B1 (-> PC0)
4 : B2 (-> PA3)
5 : B3 (-> PB3 through 10K)
6 : B4 (-> PB3 through 4.7K)
7 : Start (-> PB2)
8 : Select (-> PA0)
9 : Analog (-> PC2)
10 : GND
11 : Left (-> PB4)
12 : Down (-> PB7)
13 : Right (-> PB6)
14 : Up (-> PB5)
15 : L1(-> PC1)
16 : L2 (-> PA4)
```

Notes: