INTRODUCTION

With the introduction of Microchip's microcontrollers with the USB OTG peripheral, microcontroller applications can easily support USB embedded host functionality. Many devices fall under one of the defined USB classes, such as Mass Storage Device (thumb drives and external hard drives), Human Interface Device (or HID, such as computer mice), Printer and so on. Some devices, however, do not fit well within the restrictions of these classes. Instead, they utilize the fundamental USB transfers without any additional protocols.

Microchip provides a USB Generic Client Driver to facilitate USB communication with these unclassified or vendor-specific devices. This application note provides a brief overview on its configuration and use.

USB TRANSFERS

The USB Specification defines four different types of transfers:

• Control Transfers: Used to configure a device at attach time and can be used for other device-specific purposes, including control of other pipes on the device.
• Bulk Data Transfers: Generated or consumed in relatively large and “bursty” quantities and have wide dynamic latitude in transmission constraints (e.g., printers and mass storage devices).
• Interrupt Data Transfers: Used for timely but reliable delivery of data. For example, characters or coordinates with human perceptible echo or feedback response characteristics (e.g., keyboards, mice and joysticks).
• Isochronous Data Transfers: Occupy a pre-negotiated amount of USB bandwidth with a pre-negotiated delivery latency (e.g., audio data). These are also called streaming real-time transfers.

Control transfers are typically performed on Endpoint 0, and are performed automatically by the USB Embedded Host Stack. Application level transfers are usually on other endpoints, and are bulk, interrupt or isochronous, depending on the type of application.

ABOUT THE DRIVER

The Generic Client Driver provided with the Microchip USB Embedded Host Stack provides support for a device with one IN endpoint (i.e., data transfer to the host) and one OUT endpoint (data transfer from the host), in addition to Endpoint 0. By default, Endpoint 1 is used. The transfer type for these endpoints can be any of the four possible transfer types: control, bulk, interrupt or isochronous. The USB Embedded Host Driver will manage the specific timing and packet size requirements of whatever transfer type is used.

The type, length and format of the data transferred is totally application dependent. The only requirement is that both the USB peripheral device and the USB embedded host are aware of the data format.

USING THE GENERIC CLIENT DRIVER

Application Architecture

Functionally, the Generic Client Driver is a layer between the application software and the embedded host USB driver. The general architecture is shown in Figure 1.

FIGURE 1: APPLICATION ARCHITECTURE
Configuring the Generic Client Driver

Use the USB configuration tool, USBConfig.exe, or the USB library configuration tool provided in the MPLAB® IDE VDI to configure the Generic Client Driver. The tool generates two files which must be included in the project: usb_config.c and usb_config.h.

There are items to configure in four of the tabs, in the following order:

1. From the Main tab (Figure 2):
   a) Select the Target Device Family from the drop-down combo box.
   b) Select the Device Type. To use the Generic Client Driver, the device must be an embedded host. The USB Embedded Host, USB Dual Role and USB OTG options all provide embedded host functionality.
   c) Select the Ping-Pong Buffer mode, if applicable. Using Ping-Pong mode requires more data memory, but results in higher data throughput.

FIGURE 2: CONFIGURATION TOOL, MAIN TAB
2. From the **Host** tab (Figure 3):
   a) Enable the transfer type that the USB peripheral device uses. The transfer type used by the USB peripheral is transparent to the application. As long as the application provides support for the transfer type, the application can communicate with the peripheral.
   
   ![Figure 3: Configuration Tool, Host Tab](image)

   b) If the host application will utilize transfer events, check **Generate Transfer Events**. Otherwise, the application must poll for transfer completion. Refer to the “**Generic Client Driver Events**” section for more information on transfer events.
3. From the **TPL** tab (Figure 4), add support for each of the required devices. All devices supported by the Generic Client Driver must be supported by VID and PID, since there are no class identifiers.

For each device to be supported:

a) Enter a brief text description of the peripheral device in the **Description** field.

b) Select **Support via VID/PID**, then enter the hexadecimal values for the USB Vendor ID and Product ID in the appropriate fields. These values must be entered in hexadecimal, in the format ‘0xnnnn’, as shown in the figure.

c) At the **Client Driver** drop-down combo box, select **Generic**.

d) Fill in the **Initial Configuration** and **Initialization Flags** fields, as needed. Currently, the Generic Client Driver does not use these fields; leave them as 0.

e) Click **Add to TPL** to update the Targeted Peripheral List.

![FIGURE 4: CONFIGURATION TOOL, TPL TAB](image)

**Microchip USB OTG Configuration Tool v2.0.0.0**

A Targeted Peripheral List is required for this type of application.

<table>
<thead>
<tr>
<th>Description</th>
<th>VID</th>
<th>PID</th>
<th>Class</th>
<th>SubClass</th>
<th>Protocol</th>
<th>Client Driver</th>
<th>Config Flags</th>
<th>HNP?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microchip TS USB</td>
<td>0x04D0</td>
<td>0x001C</td>
<td></td>
<td></td>
<td></td>
<td>Generic</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
4. From the **Generic** tab (Figure 5):
   a) If you want the Generic Client Driver to utilize device serial numbers, check **Include Serial Number Support** and indicate the maximum length of the serial number.
   b) If **Generate Transfer Events** is checked on the **Host** tab, then the Generic Client Driver will also generate transfer events. Otherwise, the application must poll for transfer completion.
   c) The USB specification requires that a transfer should not be terminated as long as a device returns a NAK. In an embedded application, transfers are often terminated after a certain time-out duration. If you would like the stack to automatically terminate a transfer if no response is received within a certain time, check **Enable NAK Timeout** and adjust the **Number of NAK's Allowed** for the transfer type specified on the **Host** tab.

**FIGURE 5: CONFIGURATION TOOL, GENERIC TAB**
Generic Client Driver Events

The Generic Client Driver generates four events:

- **EVENT_GENERIC_ATTACH**: This event indicates that a generic device has been attached. The address of the attached device is sent with the event, as well as the VID, PID and serial number (if supported).
- **EVENT_GENERIC_DETACH**: This event indicates that a generic device has detached from the bus. The address previously used by the device is sent with the event.
- **EVENT_GENERIC_TX_DONE**: This event indicates that a write request has completed. This event will be sent only if transfer events are enabled. Otherwise, the application must poll for transfer completion status.
- **EVENT_GENERIC_RX_DONE**: This event indicates that a read request has completed. This event will be sent only if transfer events are enabled. Otherwise, the application must poll for transfer completion status.

These events should be handled in the event handler specified in the Name of Application Event Handler field on the Host tab (Figure 3). For more information about the function prototype and structure of the event handler, refer to Microchip Application Note AN1140, “USB Embedded Host Stack”.

Communicating with the USB Peripheral

After the application receives the **EVENT_GENERIC_ATTACH** event, it can communicate with the USB peripheral. The available function calls are summarized in Example 1.

**Note:** Refer to the Help file documentation installed with the USB Embedded Host Stack for the complete list of API calls.

Depending on its structure, the application may use four of these function calls. All applications use the function calls, **USBHostGenericRead()** and **USBHostGenericWrite()**, to communicate with the device after it is attached.

If the application does not use transfer events, the application will use **USBHostGenericRxIsComplete()** and **USBHostGenericTxIsComplete()** to determine if the transfer is complete. An example of their use is shown in Example 2.

If the application uses transfer events, the application will receive the **EVENT GENERIC_RX_DONE** event when a read terminates and the **EVENT GENERIC_TX_DONE** event when a write terminates. Before initiating a transfer, the application can see if a transfer is already in progress by using **USBHostGenericRxIsBusy()** and **USBHostGenericTxIsBusy()**. An example of their use is shown in Example 3.

**EXAMPLE 1: GENERIC CLIENT DRIVER FUNCTION PROTOTYPES**

For Communications:

```
BYTE USBHostGenericRead( BYTE deviceAddress, void *buffer, DWORD length);
BYTE USBHostGenericWrite( BYTE deviceAddress, void *buffer, DWORD length);
```

For Applications Not Using Transfer Events:

```
BOOL USBHostGenericRxIsComplete( BYTE deviceAddress, BYTE *errorCode, DWORD *byteCount );
BOOL USBHostGenericTxIsComplete( BYTE deviceAddress, BYTE *errorCode );
```

For Applications Using Transfer Events:

```
BOOL USBHostGenericRxIsBusy( BYTE deviceAddress );
BOOL USBHostGenericTxIsBusy( BYTE deviceAddress );
```

**EXAMPLE 2: STARTING A READ USING POLLING**

```
if (USBHostGenericRxIsComplete( deviceAddress,&errorCode,&byteCount ))
{
    USBHostGenericRead( deviceAddress, &buffer, sizeof( buffer ) );
}
```

**EXAMPLE 3: STARTING A READ USING TRANSFER EVENTS**

```
if (!USBHostGenericRxIsBusy( deviceAddress ))
{
    USBHostGenericRead( deviceAddress, &buffer, sizeof( buffer ) );
}
```
DEMONSTRATION PROGRAM

The USB Embedded Host Generic Client Driver is installed with the USB software support packages available for download from the Microchip web site, at www.microchip.com/USB. Refer to the Release Notes for the installation location of the Generic Client Driver files.

One demonstration project is provided for the Generic Client Driver. This project shows how to use the driver to communicate with the USB Peripheral Generic Device Driver. Refer to the Release Notes for the installation location of the demonstration.

To use the projects:

1. Program one Explorer 16 Demonstration Board with the USB embedded host example.
2. Program another demonstration board (either a second Explorer 16 board or a PICDEM™ FS USB Demonstration Board) with the USB peripheral example.
3. Connect the two boards using a USB cable and allow them to execute.

The USB peripheral device reads the temperature and potentiometer value on its demo board. The USB embedded host example requests this information from the USB peripheral and displays the information on the LCD. For more information about these projects, refer to the additional documentation installed with each of them.

CONCLUSION

Many USB peripheral devices utilize one of the standard USB classes, but some do not. The USB Embedded Host Generic Client Driver provides a simple mechanism to interface with a USB peripheral that does not fall under a standard USB class.

REFERENCES

The following application notes pertain specifically to the Generic Client Driver:

• Microchip Application Note AN1140, "USB Embedded Host Stack" (DS01140)
• Microchip Application Note AN1141, "USB Embedded Host Stack Programmer’s Guide" (DS01141)

Additional detailed information is available in the Help file installed with the Microchip USB Embedded Host Stack. This file is located in the directory path: \Microchip\Help

For more information on components of the Microchip USB solutions, as well as more USB Embedded Host Client Drivers and Peripheral Function Drivers, visit the Microchip USB Design Center at: www.microchip.com/usb

For more information on USB in general:

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip’s Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks
The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, KEELOQ logo, MPLAB, PIC, PICmicro, PICSTART, rFIC, SmartShunt and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

FilterLab, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, PICkit, PICDEM, PICDEM.net, PICtail, PIC® logo, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rFLAB, Select Mode, Total Endurance, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2009, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company’s quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip’s quality system for the design and manufacture of development systems is ISO 9001:2000 certified.
<table>
<thead>
<tr>
<th>Region</th>
<th>Country</th>
<th>City</th>
<th>Phone Number</th>
<th>Fax Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMERICAS</td>
<td>Atlanta</td>
<td>Duluth, GA</td>
<td>678-957-9614</td>
<td>678-957-1455</td>
</tr>
<tr>
<td></td>
<td>Boston</td>
<td>Westborough, MA</td>
<td>774-760-0087</td>
<td>774-760-0088</td>
</tr>
<tr>
<td></td>
<td>Chicago</td>
<td>Itasca, IL</td>
<td>630-285-0071</td>
<td>630-285-0075</td>
</tr>
<tr>
<td></td>
<td>Cleveland</td>
<td>Independence, OH</td>
<td>216-447-0464</td>
<td>216-447-0043</td>
</tr>
<tr>
<td></td>
<td>Dallas</td>
<td>Addison, TX</td>
<td>972-818-7423</td>
<td>972-818-2924</td>
</tr>
<tr>
<td></td>
<td>Detroit</td>
<td>Farmington Hills, MI</td>
<td>248-538-2250</td>
<td>248-538-2260</td>
</tr>
<tr>
<td></td>
<td>Kokomo</td>
<td>Kokomo, IN</td>
<td>765-864-8360</td>
<td>765-864-8387</td>
</tr>
<tr>
<td></td>
<td>Los Angeles</td>
<td>Mission Viejo, CA</td>
<td>949-462-9523</td>
<td>949-462-9608</td>
</tr>
<tr>
<td></td>
<td>Santa Clara</td>
<td>Santa Clara, CA</td>
<td>408-961-6444</td>
<td>408-961-6445</td>
</tr>
<tr>
<td></td>
<td>Toronto</td>
<td>Mississauga, Ontario, Canada</td>
<td>905-673-0699</td>
<td>905-673-6509</td>
</tr>
<tr>
<td>ASIA/PACIFIC</td>
<td>Asia Pacific Office</td>
<td>Suites 3707-14, 37th Floor</td>
<td>61-2-9868-6733</td>
<td>61-2-9868-6755</td>
</tr>
<tr>
<td></td>
<td>China - Beijing</td>
<td>Tel: 86-10-8528-2100</td>
<td>Fax: 86-10-8528-2104</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Chengdu</td>
<td>Tel: 86-28-8665-5511</td>
<td>Fax: 86-28-8665-7889</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Hong Kong SAR</td>
<td>Tel: 852-2401-1200</td>
<td>Fax: 852-2401-3431</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Nanjing</td>
<td>Tel: 86-25-8473-2460</td>
<td>Fax: 86-25-8473-2470</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Qingdao</td>
<td>Tel: 86-532-8502-7355</td>
<td>Fax: 86-532-8502-7205</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Shanghai</td>
<td>Tel: 86-21-5407-5533</td>
<td>Fax: 86-21-5407-5066</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Shenyang</td>
<td>Tel: 86-24-2334-2829</td>
<td>Fax: 86-24-2334-2393</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Shenzhen</td>
<td>Tel: 86-755-8203-2660</td>
<td>Fax: 86-755-8203-1760</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Wuhan</td>
<td>Tel: 86-27-5980-5300</td>
<td>Fax: 86-27-5980-5118</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Xiamen</td>
<td>Tel: 86-592-2388138</td>
<td>Fax: 86-592-2388130</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Xian</td>
<td>Tel: 86-29-8833-7252</td>
<td>Fax: 86-29-8833-7256</td>
<td></td>
</tr>
<tr>
<td></td>
<td>China - Zhuhai</td>
<td>Tel: 86-756-3210040</td>
<td>Fax: 86-756-3210049</td>
<td></td>
</tr>
<tr>
<td>ASIA/PACIFIC</td>
<td>India - Bangalore</td>
<td>Tel: 91-80-3090-4444</td>
<td>Fax: 91-80-3090-4080</td>
<td></td>
</tr>
<tr>
<td></td>
<td>India - New Delhi</td>
<td>Tel: 91-11-4160-8631</td>
<td>Fax: 91-11-4160-8632</td>
<td></td>
</tr>
<tr>
<td></td>
<td>India - Pune</td>
<td>Tel: 91-20-2566-1512</td>
<td>Fax: 91-20-2566-1513</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Japan - Yokohama</td>
<td>Tel: 81-45-471-6166</td>
<td>Fax: 81-45-471-6122</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Korea - Daegu</td>
<td>Tel: 82-53-744-4301</td>
<td>Fax: 82-53-744-4302</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Korea - Seoul</td>
<td>Tel: 82-2-554-7200</td>
<td>Fax: 82-2-558-5932 or 82-2-558-5934</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Malaysia - Kuala Lumpur</td>
<td>Tel: 60-3-6201-9857</td>
<td>Fax: 60-3-6201-9859</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Malaysia - Penang</td>
<td>Tel: 60-4-227-8870</td>
<td>Fax: 60-4-227-4068</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Philippines - Manila</td>
<td>Tel: 63-2-634-9065</td>
<td>Fax: 63-2-634-9069</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Singapore</td>
<td>Tel: 65-6334-8870</td>
<td>Fax: 65-6334-8850</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Taiwan - Hsin Chu</td>
<td>Tel: 886-3-572-9526</td>
<td>Fax: 886-3-572-6459</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Taiwan - Kaohsiung</td>
<td>Tel: 886-7-536-4818</td>
<td>Fax: 886-7-536-4803</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Taiwan - Taipei</td>
<td>Tel: 886-2-2500-6610</td>
<td>Fax: 886-2-2508-0102</td>
<td></td>
</tr>
<tr>
<td>MEurope</td>
<td>Austria - Wels</td>
<td>Tel: 43-7242-2244-39</td>
<td>Fax: 43-7242-2244-393</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Denmark - Copenhagen</td>
<td>Tel: 45-4450-2828</td>
<td>Fax: 45-4485-2829</td>
<td></td>
</tr>
<tr>
<td></td>
<td>France - Paris</td>
<td>Tel: 33-1-69-53-63-20</td>
<td>Fax: 33-1-69-30-90-79</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Germany - Munich</td>
<td>Tel: 49-89-627-144-0</td>
<td>Fax: 49-89-627-144-44</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Italy - Milan</td>
<td>Tel: 39-0331-742611</td>
<td>Fax: 39-0331-466781</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Netherlands - Drunen</td>
<td>Tel: 31-416-690399</td>
<td>Fax: 31-416-690340</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Spain - Madrid</td>
<td>Tel: 34-91-708-08-90</td>
<td>Fax: 34-91-708-08-91</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UK - Wokingham</td>
<td>Tel: 44-118-921-5869</td>
<td>Fax: 44-118-921-5820</td>
<td></td>
</tr>
</tbody>
</table>

© 2009 Microchip Technology Inc.