# USB Demonstrations Help

MPLAB Harmony Integrated Software Framework

# Volume I: Getting Started With MPLAB Harmony Libraries and Applications

This volume introduces the MPLAB® Harmony Integrated Software Framework.

## Description

MPLAB Harmony is a layered framework of modular libraries that provide flexible and interoperable software "building blocks" for developing embedded PIC32 applications. MPLAB Harmony is also part of a broad and expandable ecosystem, providing demonstration applications, third-party offerings, and convenient development tools, such as the MPLAB Harmony Configurator (MHC), which integrate with the MPLAB X IDE and MPLAB XC32 language tools.


## Legal Notices

Please review the *Software License Agreement* prior to using MPLAB Harmony. It is the responsibility of the end-user to know and understand the software license agreement terms regarding the Microchip and third-party software that is provided in this installation. A copy of the agreement is available in the `<install-dir>/doc` folder of your MPLAB Harmony installation.

The OPENRTOS® demonstrations provided in MPLAB Harmony use the OPENRTOS evaluation license, which is meant for demonstration purposes only. Customers desiring development and production on OPENRTOS must procure a suitable license. Please refer to one of the following documents, which are located in the `<install-dir>/third_party/rtos/OPENRTOS/Documents` folder of your MPLAB Harmony installation, for information on obtaining an evaluation license for your device:

- `OpenRTOS Click Thru Eval License PIC32MXxx.pdf`
- `OpenRTOS Click Thru Eval License PIC32MZxx.pdf`

**TIP!**

Throughout this documentation, occurrences of `<install-dir>` refer to the default MPLAB Harmony installation path:

- Windows: `C:/microchip/harmony/<version>`
- Mac OS/Linux: `~/microchip/harmony/<version>`

# Applications Help

This section provides information on the various application demonstrations that are included in MPLAB Harmony.

## Description

Applications determine how MPLAB Harmony libraries (device drivers, middleware, and system services) are used to do something useful. In a MPLAB Harmony system, there may be one main application, there may be multiple independent applications or there may be one or more Operating System (OS) specific applications. Applications interact with MPLAB Harmony libraries through well defined interfaces. Applications may operate in a strictly polling environment, they may be interrupt driven, they may be executed in OS-specific threads, or they may be written so as to be flexible and easily configured for any of these environments. Applications generally fit into one of the following categories.

## Demonstration Applications

Demonstration applications are provided (with MPLAB Harmony or in separate installations) to demonstrate typical or interesting usage models of one or more MPLAB Harmony libraries. Demonstration applications can demonstrate realistic solutions to real-life problems.

## Sample Applications

Sample applications are extremely simple applications provided with MPLAB Harmony as examples of how to use individual features of a library. They will not normally accomplish anything useful on their own. They are provided primarily as documentation to show how to use a library.

# USB Demonstrations

This section provides descriptions of the USB demonstrations.

MPLAB Harmony is available for download from the Microchip website by visiting: http://www.microchip.com/mplabharmony. Once you are on the site, click the Downloads tab to access the appropriate download for your operating system. For additional information on this demonstration, refer to the “Applications Help” section in the MPLAB Harmony Help.

## Introduction

USB Library Demonstration Applications Help

## Description

This distribution package contains a variety of USB-related firmware projects that demonstrate the capabilities of the MPLAB Harmony USB stack. This section describes the hardware requirement and procedures to run these firmware projects on Microchip demonstration and development boards.

To know more about the MPLAB Harmony USB stack and configuring the USB stack and the APIs provided by the USB stack, refer to the USB Library documentation.

## Program, Data Memory, and Stack Component Memory

Refer to USB Device Stack Demonstration Application Program and Data Memory Requirements and USB Device Stack Component Memory Requirements for important memory information.

## Pen Drive Tests

Refer to USB MSD Host USB Pen Drive Tests for information on the tests conducted on USB Flash devices.

## USB Device Stack Demonstration Application Program and Data Memory Requirements

Provides information on program and data memory requirements, as well as pen drive test specifications.

## Description

## Program Memory and Data Memory Requirements with -O1 Optimization

The following table shows the program memory and data memory requirements of the USB Device Stack demonstration applications. All size figures are in bytes. Demonstration applications were compiled with the MPLAB XC32 C/C++ Compiler, v1.40, with –O1 optimization.

| Demonstration Name | | Program Memory Components | | | Data Memory Components | |
|---|---|---|---|---|---|---|
| -O1 Optimization | | USB Stack | Other Drivers | System and Application | USB Stack | Others |
| cdc_com_port_single | PIC32MX | 14048 | 0 | 5880 | 262 | 718 |
| | PIC32MZ | 18924 | 4328 | 11632 | 318 | 1410 |
| cdc_com_port_dual | PIC32MX | 13980 | 0 | 5776 | 262 | 1250 |
| | PIC32MZ | 18856 | 4328 | 10908 | 318 | 2166 |
| cdc_serial_emulator | PIC32MX | 13976 | 9100 | 5916 | 262 | 858 |
| | PIC32MZ | N/A | N/A | N/A | N/A | N/A |
| cdc_serial_emulator_msd | PIC32MX | 20380 | 12560 | 39540 | 438 | 1950 |
| | PIC32MZ | NA | NA | NA | NA | NA |
| cdc_msd_basic | PIC32MX | 20392 | 3460 | 39444 | 494 | 1810 |
| | PIC32MZ | 25264 | 7788 | 45404 | 494 | 20438 |
| hid_basic | PIC32MX | 13988 | 0 | 5440 | 263 | 601 |
| | PIC32MZ | 18764 | 4328 | 10460 | 319 | 893 |
| hid_joystick | PIC32MX | 13432 | 0 | 5332 | 263 | 485 |
| | PIC32MZ | 18208 | 4328 | 10368 | 319 | 777 |
| hid_keyboard | PIC32MX | 14016 | 0 | 6196 | 263 | 581 |
| | PIC32MZ | 18792 | 4328 | 11048 | 319 | 873 |
| hid_mouse | PIC32MX | 13460 | 0 | 5972 | 263 | 497 |
| | PIC32MZ | 18236 | 4328 | 10964 | 319 | 793 |
| hid_msd_basic | PIC32MX | 20352 | 3460 | 39172 | 495 | 1861 |
| | PIC32MZ | 25232 | 7788 | 44216 | 495 | 20445 |
| msd_basic | PIC32MX | 17848 | 3460 | 38108 | 492 | 1348 |
| | PIC32MZ | 22632 | 7788 | 43996 | 492 | 20068 |
| vendor | PIC32MX | 12560 | 0 | 5660 | 324 | 560 |
| | PIC32MZ | 17356 | 4328 | 12080 | 324 | 1748 |

**Note:** The msd_basic, cdc_msd_basic, and the hid_msd_basic demonstrations use the PIC32 program Flash memory as the MSD storage media. The difference in Data Memory requirements between the PIC32MX and PIC32MZ microcontrollers for these demonstration examples, is due to an application demonstration buffer whose size is equal to the erase page size of the PIC32 microcontroller. On the PIC32MX795F512L, this size is 4096 bytes. On the PIC32MZ2048ECH144, the erase page size is 16 KB.

## Program Memory and Data Memory Requirements with -Os Optimization

The following table shows the program memory and data memory requirements of the USB Device Stack demonstration applications. All size figures are in bytes. Demonstration applications were compiled with the MPLAB XC32 C/C++ Compiler, v1.40, with –Os optimization.

| Demonstration Name | | Program Memory Components | | | Data Memory Components | |
|---|---|---|---|---|---|---|
| Optimization -Os | | USB Stack | Other Drivers | System and Application | USB Stack | Others |
| cdc_com_port_single | PIC32MX | 12556 | 0 | 5656 | 262 | 718 |
| | PIC32MZ | 16840 | 4144 | 11076 | 318 | 1410 |
| cdc_com_port_dual | PIC32MX | 12548 | 0 | 5620 | 262 | 1250 |
| | PIC32MZ | 16832 | 4144 | 10448 | 318 | 2166 |
| cdc_serial_emulator | PIC32MX | 12504 | 7744 | 5880 | 262 | 858 |
| | PIC32MZ | N/A | N/A | N/A | N/A | N/A |
| cdc_serial_emulator_msd | PIC32MX | 20380 | 12560 | 35080 | 438 | 1950 |
| | PIC32MZ | N/A | N/A | N/A | N/A | N/A |
| cdc_msd_basic | PIC32MX | 17896 | 2884 | 39232 | 494 | 1810 |
| | PIC32MZ | 22172 | 7012 | 44868 | 494 | 20438 |
| hid_basic | PIC32MX | 12656 | 0 | 5344 | 263 | 601 |
| | PIC32MZ | 16784 | 4144 | 10060 | 319 | 893 |
| hid_joystick | PIC32MX | 12144 | 0 | 5304 | 263 | 485 |
| | PIC32MZ | 16272 | 4144 | 10040 | 319 | 777 |
| hid_keyboard | PIC32MX | 12664 | 0 | 6060 | 263 | 581 |
| | PIC32MZ | 16792 | 4144 | 10604 | 319 | 873 |
| hid_mouse | PIC32MX | 12152 | 0 | 5820 | 263 | 497 |
| | PIC32MZ | 16280 | 4144 | 10512 | 319 | 793 |
| hid_msd_basic | PIC32MX | 18060 | 2884 | 39068 | 495 | 1861 |
| | PIC32MZ | 20352 | 7012 | 45788 | 495 | 20445 |
| msd_basic | PIC32MX | 15776 | 2884 | 38044 | 492 | 1348 |
| | PIC32MZ | 19090 | 7012 | 44426 | 492 | 20068 |
| vendor | PIC32MX | 11452 | 0 | 5540 | 324 | 560 |
| | PIC32MZ | 15584 | 4144 | 10948 | 324 | 1748 |

## *USB Device Stack Component Memory Requirements*

Provides memory requirements.

### Description

The following table shows the Program and Data Memory requirements for individual components in the MPLAB Harmony USB Device Stack.

| Device Stack Component | Program Memory | Data Memory |
|---|---|---|
| Device Layer | 5688 | 184 |
| CDC Function Driver | 2420 | 64 + (36 * Queue Size) |
| MSD Function Driver | 5352 | 217 |
| HID Function Driver | 2376 | 40 + (36 * Queue Size) |
| Vendor | 912 | 8 + (36 * Queue Size) |
| PIC32MX USB Driver | 5636 | 144 + (32 * Number of Endpoints) |
| PIC32MZ USB Driver | 10244 | 192 + (32 * Number of Endpoints) |

**Notes:**

1. Memory requirements (in bytes) for a single instance.
2. Size measured for USB Device Stack Components in MPLAB Harmony.
3. Data Memory does not include function call stack memory size.

## *USB MSD Host USB Pen Drive Tests*

Provides pen drive test specifications.

### Description

**USB MSD Host USB Pen Drive Tests**

The following table lists the commercially available USB pen drives, which have been tested to successfully enumerate with the MSD Host Driver

in the MPLAB Harmony USB Host. Note that if the USB pen drive you are using in not included in the table, this indicates that this USB pen drive has not been tested with the MSD Host Driver. However, the USB pen drive could still potentially work with MSD Host Driver. Some USB pen drives in this table did not have their manufacturer or model data available. The USB Pen drives were tested with the msd_basic USB Host demonstration in the latest version of the MPLAB Harmony USB Host Stack.

| VID | PID | Manufacturer | Model/Drive Capacity |
|---|---|---|---|
| 0x1B1C | 0x1A0F | Corsair Components | Flash Voyager Go 8 GB |
| 0x03F0 | 0x0AB7 | Hewlett-Packard | 64 GB |
| 0xABCD | 0x1234 | Microchip Technology Inc. | 4 GB |
| 0x125F | 0xCB10 | Adata | Dashdrive UV100 8 GB |
| 0x8644 | 0x8003 | Verico | T Series 16 GB |
| 0x8564 | 0x1000 | Transcend | USB 3.0 32 GB |
| 0x0951 | 0x16A7 | Dell | Kingston Technology 16 GB |
| 0x0718 | 0x0704 | Imation | 16 GB Pen Drive |
| 0x048D | 0x1168 | iBall | Jaldi 16 GB Pen Drive |
| 0x058F | 0x6366 | Alcor | Micro AXL 32 GB |
| 0x154B | 0x005B | PNY | Cube 16 GB |
| 0x0930 | 0x6544 | Toshiba | Hatabusa Pen Drive 8 GB |
| 0x058F | 0x6387 | Alcor | ZipMem 16 GB |
| 0x090C | 0x1000 | Silicon Motion Inc. | Axl 8GB |
| 0x18A5 | 0x0245 | Verbatim | Store N Go Audio USB 8 GB |
| 0x05DC | 0xC75C | Lexar | USB Pen Drive 8 GB |
| 0x1005 | 0xb113 | Apacer | 8 GB (AH233) |
| 0x054C | 0x06B0 | Sony | 8 GB |
| 0x054C | 0x0862 | Sony | Micro Vault USM-V 8 GB |
| 0x0781 | 0x557c | SanDisk | 8 GB |
| 0x1E4E | 0x3257 | Etron | iBall 16 GB |
| 0x1EC9 | 0x0101 | Moserbaer | Swivel 16 GB Pen Drive |
| 0x0BDA | 0x0109 | SanDisk | Standard A and Mini-B connector 16 GB |
| 0x1908 | 0x1320 | ZBEL | Wrist Band Flash Drive 4 GB |
| 0x0951 | 0x1665 | Kingston | Data Traveler SE9 16 GB |

### *USB HID Host Keyboard and Mouse Tests*

Provides information on tested USB keyboard and mouse devices.

## Description

The following table lists the commercially available USB keyboard and mouse devices, which have been tested to successfully enumerate with the HID Host Driver in the MPLAB Harmony USB Host. Note that if the USB HID device you are using in not included in the table, this indicates that this USB HID device has not been tested, but could still potentially work with the HID Host Driver.

| Type | Manufacturer | VID | PID | Device Details |
|---|---|---|---|---|
| Keyboard | Microsoft | 0x045E | 0x07B9 | Microsoft wired 200. |
| | ProHT | 0x1A2C | 0x0C21 | 110 keys. |
| | HP | 0x04D9 | 0x1702 | K1500 standard keyboard. |
| | Zebronics | 0x1A2C | 0x0027 | Standard keyboard. |
| | Logitech | 0x046D | 0xC31D | 112 keys. |
| | Gear Head | 0x04D9 | 0x2BA0 | 82 key, mini-USB keyboard plus integrated mouse. |
| | Intex | 0x1C4F | 0x0002 | 122 keys. |
| Mouse | HP | 0x0461 | 0x4D0F | Three button mouse. |
| | Intex | 0x1BCF | 0x0007 | Three button mouse. |
| | Anker | 0x1BCF | 0x0005 | Five button mouse. |
| | TeraByte | 0x10C4 | 0x8103 | Three button mouse. |
| | Kensington | 0x1BCF | 0x0002 | Five button mouse. |
| | Logitech | 0x046D | 0xC069 | Five button mouse. |
| | Microsoft | 0x045E | 0x0797 | Optical mouse 200, three buttons. |
| | Logitech | 0x046D | 0xC246 | Nine button mouse. |
| | HP | 0x03F0 | 0x0941 | Three button mouse. |
| | Lenovo | 0x1BCF | 0x000A | Five button mouse. |

**Note:** The above tests have been performed only on the PIC32M family of devices.

## *Demonstration Application Configurations*

This topic provides information on the available USB demonstration project configurations.

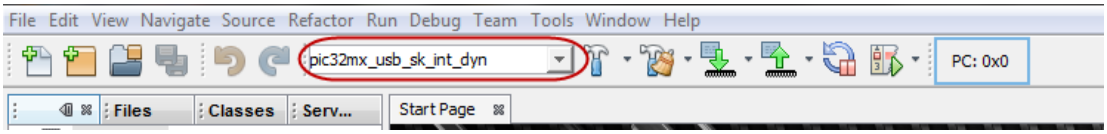
### Description

The available USB Demonstration application MPLAB X IDE projects feature support for multiple configurations. Selecting these configurations allow for the demonstration projects to run across different PIC32 microcontrollers and development boards. The following project configurations are available:

| Configuration name | Description |
|---|---|
| pic32mx_usb_sk2_int_dyn | Selecting this configuration will set up the demonstration application to run on the PIC32 USB Starter Kit II development board, with the PIC32MX795F512L microcontroller. The USB Stack will be configured for Interrupt mode operation and the USB Driver will be configured for Dynamic operation mode. |
| pic32mx_usb_sk2_poll_dyn | Selecting this configuration will set up the demonstration application to run on the PIC32 USB Starter Kit II development board, with the PIC32MX795F512L microcontroller. The USB Stack will be configured for Polled mode operation and the USB driver will be configured for Dynamic operation mode. |
| pic32mx_usb_sk3_int_dyn | Selecting this configuration will set up the demonstration application to run on the PIC32 USB Starter Kit III development board, with the PIC32MX470F512L microcontroller. The USB Stack will be configured for Interrupt mode operation and the USB Driver will be configured for Dynamic operation mode. |
| pic32mx_bt_sk_int_dyn | Selecting this configuration will set up the demonstration application to run on the PIC32 Bluetooth Starter Kit development board, with the PIC32MX270F256D microcontroller. The USB Stack will be configured for Interrupt mode operation and the USB Driver will be configured for dynamic operation mode. |
| pic32mz_da_sk_intddr_int_dyn | Selecting this configuration will set up the demonstration application to run on the PIC32MZ Embedded Graphics with Internal DRAM (DA) Starter Kit development board, with the PIC32MZ2064DAH169 microcontroller. The USB Stack will be configured for Interrupt mode operation and the USB Driver will be configured for Dynamic operation mode. |
| pic32mz_ec_sk_int_dyn | Selecting this configuration will set up the demonstration application to run on the PIC32MZ EC Starter Kit development board, with the PIC32MZ2048ECH144 microcontroller. The USB Stack will be configured for Interrupt mode operation and the USB Driver will be configured for Dynamic operation mode. |
| pic32mz_ec_sk_poll_dyn | Selecting this configuration will set up the demonstration application to run on the PIC32MZ EC Starter Kit development board, with the PIC32MC2048ECH144 microcontroller. The USB Stack will be configured for Polled mode operation and the USB Driver will be configured for Dynamic operation mode. |
| pic32mz_ec_sk_meb2_int_dyn | Selecting this configuration will set up the demonstration application to run on the PIC32MZ EC Starter Kit, with the PIC32MZ2048ECH144 microcontroller board attached to the MEB II. The USB Stack will be configured for Interrupt mode operation and the USB Driver will be configured for Dynamic operation mode. |
| pic32mz_ef_sk_int_dyn | Selecting this configuration will set up the demonstration application to run on the PIC32MZ EF Starter Kit, with the PIC32MZ2048EFM144 microcontroller. The USB Stack will be configured for Interrupt mode operation and the USB Driver will be configured for Dynamic operation mode. |

| | |
|---|---|
| pic32mz_ef_sk_poll_dyn | Selecting this configuration will set up the demonstration application to run on the PIC32MZ EF Starter Kit development board, with the PIC32MZ2048EFM144 microcontroller. The USB Stack will be configured for Polled mode operation and the USB Driver will be configured for Dynamic operation mode. |
| pic32mx795_pim_e16_int_dyn | Selecting this configuration will set up the demonstration application to run on the Explorer 16 Development Board along with the PIC32MX795F512L microcontroller Plug In Module and USB PICtail Plus Daughter Board. The USB Stack will be configured for Interrupt mode operation and the USB Driver will be configured for Dynamic operation mode. |
| pic32mx460_pim_e16_int_dyn | Selecting this configuration will set up the demonstration application to run on the Explorer 16 Development Board along with the PIC32MX460F512L microcontroller Plug In Module and USB PICtail Plus Daughter Board. The USB Stack will be configured for Interrupt mode operation and the USB Driver will be configured for Dynamic operation mode. |
| pic32mx470_curiosity | Selecting this configuration will set up the demonstration application to run on the PIC32MX470 Curiosity Development Board, with the PIC32MX470F512H microcontroller. The USB Stack will be configured for Interrupt mode operation and the USB Driver will be configured for Dynamic operation mode. |
| pic32mz_ef_curiosity | Selecting this configuration will set up the demonstration application to run on the PIC32MZ EF Curiosity Development Board, with the PIC32MZ2048EFM100 microcontroller. The USB Stack will be configured for Interrupt mode operation and the USB Driver will be configured for Dynamic operation mode. |
| pic32mk_evk_int_dyn | Selecting this configuration will set up the demonstration application to run on the PIC32MK GP Development Board, with the PIC32MK1024GPE100 microcontroller. The USB Stack will be configured for Interrupt mode operation and the USB Driver will be configured for Dynamic operation mode. |
| pic32mx_xlp_sk_int_dyn | Selecting this configuration will set up the demonstration application to run on the PIC32MX XLP Starter Kit, with the PIC32MX274F256D microcontroller. The USB Stack will be configured for Interrupt mode operation and the USB Driver will be configured for Dynamic operation mode. |
| chipkit_wf32 | Selecting this configuration will set up the demonstration application to run on the chipKIT WF32 Wi-Fi Development Board, with the PIC32MZ2048EFG100 microcontroller. The USB Stack will be configured for Interrupt mode operation and the USB Driver will be configured for Dynamic operation mode. |
| chipkit_wifire | Selecting this configuration will set up the demonstration application to run on the chipKIT Wi-FIRE Development Board, with the PIC32MX275F256D microcontroller. The USB Stack will be configured for Interrupt mode operation and the USB Driver will be configured for Dynamic operation mode. |

The following figure shows how a configuration can be selected in MPLAB X IDE.




Alternatively, the active configuration can be selected in the Project Properties.

## USB Device Demonstrations Matrix

The following table shows the availability of a configuration across available USB Device demonstration applications. **Green** indicates support. Red indicates no support.

| Demo Apps \ Configurations | pic32mx_usb_sk2_int_dyn | pic32mz_ec_sk_int_dyn | pic32mx795_pim_e16_int_dyn | pic32mx_bt_sk_int_dyn | pic32mx_usb_sk2_poll_dyn | pic32mx460_pim_e16_int_dyn | pic32mx_usb_sk2_int_sta | pic32mx_usb_sk3_int_dyn | pic32mz_bt_audio_int_dyn | pic32mz_ef_sk_poll_dyn | pic32mz_ef_sk_int_dyn | pic32mx_125_sk_int_dyn | chipkit_wf32 | chipkit_wifire | pic32mx470_curiosity | pic32mz_ef_curiosity | pic32mx_xlp_sk_int_dyn | pic32mz_ef_sk_int_dyn_micromips | pic32mz_da_sk_int_dyn |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| cdc_com_port_single | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| cdc_com_port_dual | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| cdc_serial_emulator | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| hid_basic | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| hid_mouse | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| hid_keyboard | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| hid_joystick | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| msd_basic | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| vendor | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| hid_msd_basic | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| cdc_serial_emulator_msd | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| cdc_msd_basic | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| msd_sdcard | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| msd_fs_spiflash | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| msd_multiple_luns | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |

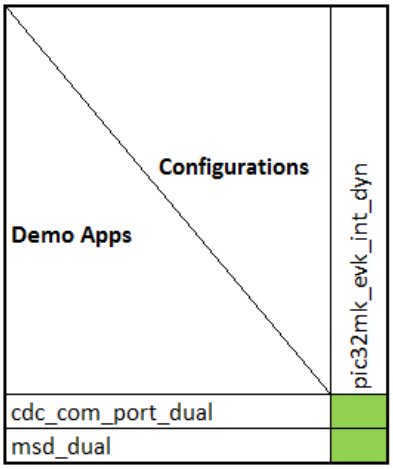
## USB Host Demonstration Matrix

The following table shows the availability of a configuration across available USB Host demonstration applications. Green indicates support. Red indicates no support.

| Demo Apps \ Configurations | pic32mx_usb_sk2_int_dyn | pic32mz_ec_sk_int_dyn | pic32mz_ec_sk_meb2_int_dyn | pic32mz_ef_sk_int_dyn | pic32mz_ef_sk_meb2_int_dyn | pic32mx795_pim_e16_int_dyn | chipkit_wf32 | chipkit_wifire | pic32mz_ef_curiosity | pic32mz_ef_curiosity | pic32mx_xlp_sk_int_dyn |
|---|---|---|---|---|---|---|---|---|---|---|---|
| cdc_basic | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| msd_basic | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| cdc_msd | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| hid_basic_mouse | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| audio_speaker | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| hub_msd | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| hub_cdc_hid | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| hid_basic_keyboard | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |

## USB Multiple Controller Demonstration Matrix

The following table shows the availability of a configuration across available USB Multiple Controller Demonstration applications. Green indicates support. Red indicates no support.




## *Demonstrations*

The USB Demonstrations are grouped into USB Device Stack, USB Host Stack, USB Dual Role, and USB demonstrations that make use of multiple USB controllers on certain PIC32 family devices.

## Device

This section describes the USB Device demonstrations.

### Description

The MPLAB Harmony USB Device Stack demonstration applications uses LEDs on the development board to indicate the USB state of the device. The following table provides details on the development board specific LEDs and the USB Device State these indicate when active. This indication scheme is implemented by all USB Device Stack Demonstration applications.

**USB Device State and LED Indication**

| Demonstration Board | Reset State | Configured State | Suspended State |
|---|---|---|---|
| Explorer 16 Development Board and PIM | D3, D4 | D5 | D4, D5 |
| PIC32 USB Starter Kit II | LED1, LED2 | LED3 | LED2, LED3 |
| PIC32MZ Embedded Connectivity (EC) Starter Kit | LED1, LED2 | LED3 | LED2, LED3 |
| PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit | LED1, LED2 | LED3 | LED2, LED3 |
| PIC32 USB Starter Kit III | LED1, LED2 | LED3 | LED2, LED3 |
| PIC32 Bluetooth Starter Kit | Red LED, Green LED | Blue LED | Green LED, Blue LED |
| PIC32MX470 Curiosity Development Board | LED1, LED2 | LED3 | LED2, LED3 |
| PIC32MZ EF Curiosity Development Board | LED1, LED2 | LED3 | LED2, LED3 |

## *cdc_com_port_dual*

Demonstrates a USB CDC device, emulating dual serial COM ports - one looping back into the other.

### Description

This demonstration application creates a USB CDC Device that enumerates as two serial ports on the USB Host personal computer. This application demonstrates the ability of the MPLAB Harmony USB Device Stack to support multiple instances of the same Device class.

## Building the Application

This section identifies the MPLAB X IDE project name and location and lists and describes the available configurations for the USB CDC Device Dual COM Port Demonstration.

### Description

To build this project, you must open the `cdc_com_port_dual.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/device/cdc_com_port_dual`.

## MPLAB X IDE Project

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `cdc_com_port_dual.X` | `<install-dir>/apps/usb/device/cdc_com_port_dual/firmware` |

## MPLAB X IDE Project Configurations

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mx460_pim_e16_int_dyn | pic32mx460_pim+e16 | Select this MPLAB X IDE project configuration to run the demonstration on the Explorer 16 Development Board configured for Interrupt mode and dynamic operation. This configuration also requires PIC32MX460F512L Plug-In Module (PIM) and the USB PICtail Plus Daughter Board. |
| pic32mx_bt_sk_int_dyn | pic32mx_bt_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 Bluetooth Starter Kit configured for Interrupt mode and dynamic operation. |
| pic32mx_usb_sk3_int_dyn | pic32mx_usb_sk3 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit III configured for Interrupt mode and dynamic operation. |
| pic32mx_usb_sk2_int_dyn | pic32mx_usb_sk2 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit II configured for Interrupt mode and dynamic operation. |
| pic32mx_xlp_sk_int_dyn | pic32mx_xlp_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MX XLP Starter Kit configured for Interrupt mode and dynamic operation. |
| pic32mz_ef_sk_int_dyn | pic32mz_ef_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit configured for Interrupt mode and dynamic operation. |
| pic32mx470_curiosity | pic32mx470_curiosity | Select this MPLAB X IDE project configuration to run the demonstration application to run on the PIC32MX470 Curiosity Development Board, with the PIC32MX470F512H microcontroller. The USB Stack will be configured for Interrupt mode operation and the USB Driver will be configured for Dynamic operation mode. |
| pic32mz_ef_curiosity | pic32mz_ef_curiosity | Select this MPLAB X IDE project configuration to run the demonstration application to run on the PIC32MZ EF Curiosity Development Board, with the PIC32MZ2048EFM100 microcontroller. The USB Stack will be configured for Interrupt mode operation and the USB Driver will be configured for Dynamic operation mode. |

## Configuring the Hardware

Describes how to configure the supported hardware.

## Description

PIC32 USB Starter Kit II

Remove jumper JP2.

PIC32MZ EC Starter Kit

Remove jumper JP1.

PIC32MZ EF Starter Kit

No hardware related configuration or jumper setting changes are necessary.

PIC32MX XLP Starter Kit

No hardware related configuration or jumper setting changes are necessary.

PIC32 USB Starter Kit III

Remove jumper JP1.

PIC32 Bluetooth Starter Kit

Jumper J8 should either be shorted between pins 2 and 3 or should be completely open.

PIC32MX460F512L PIM

Jumper J10 should be removed. This plug-in module should be used along with the Explorer 16 Development Board and the USB PICtail Plus daughter board. The microcontroller PIM should be plugged into the PIM socket on the board. The USB PICtail Plus daughter board should be connected to the edge connector J9.

On the Explorer 16 Development Board:

- Switch S2 should be set to PIM
- Jumper JP2 should be in place

On the USB PICtail Plus Daughter Board:

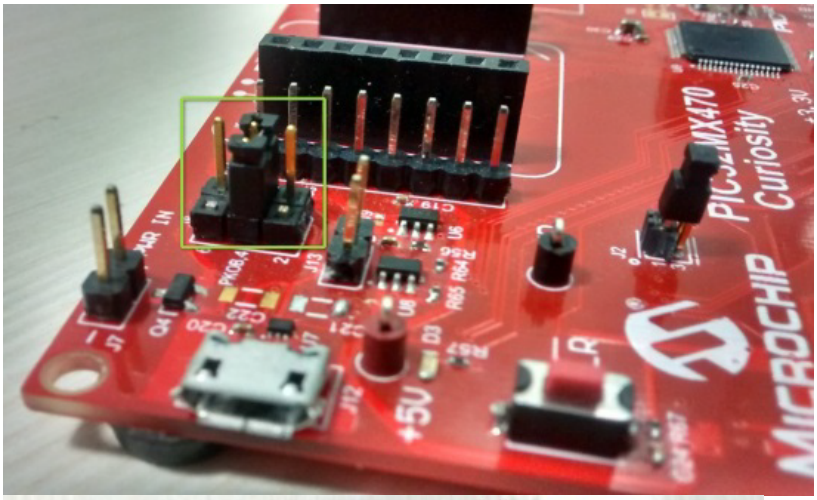
- Jumper JP1 should be in place
- Jumper JP2 and JP4 should be removed

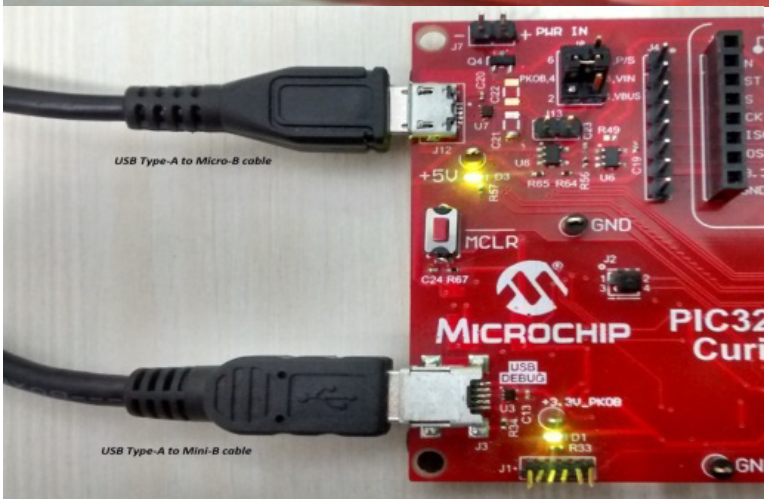
On the PIC32MX460F512L PIM:

- Keep jumper J10 open
- Keep all jumpers in J9 open

PIC32MX470 Curiosity Development Board

- Ensure that a jumper is placed at 4-3 on J8, to select supply from debug USB connector.
- Power the PIC32MX470 Curiosity Development Board from a Host PC through a Type-A male to mini-B USB cable connected to Mini-B port (J3).
- Ensure that jumper is not present in the J13 header to use the Curiosity board in device mode.
- Plug in a USB cable with a micro-B type connector to Micro-B port (J12), and plug the other end into your computer.

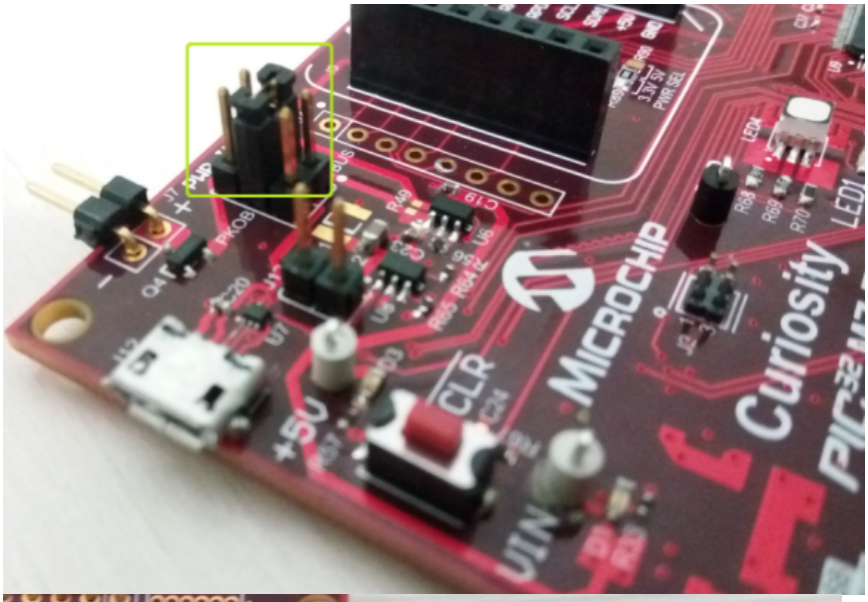


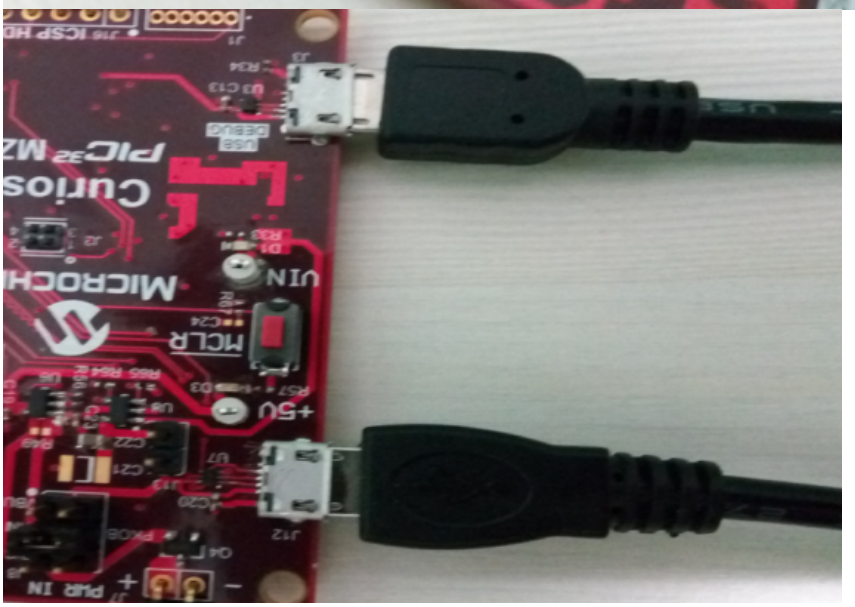




PIC32MZ EF Curiosity Development Board

- Ensure that a jumper is placed at 4-3 on J8, to select supply from debug USB connector.
- Power the PIC32MZ EF Curiosity Development Board from a Host PC through a Type-A male to micro-B USB cable connected to Micro-B port (J3).

- Ensure that jumper is not present in the J13 header to use the Curiosity board in device mode.
- Plug in a USB cable with a micro-B type connector to Micro-B port (J12), and plug the other end into your computer.



## Running the Demonstration

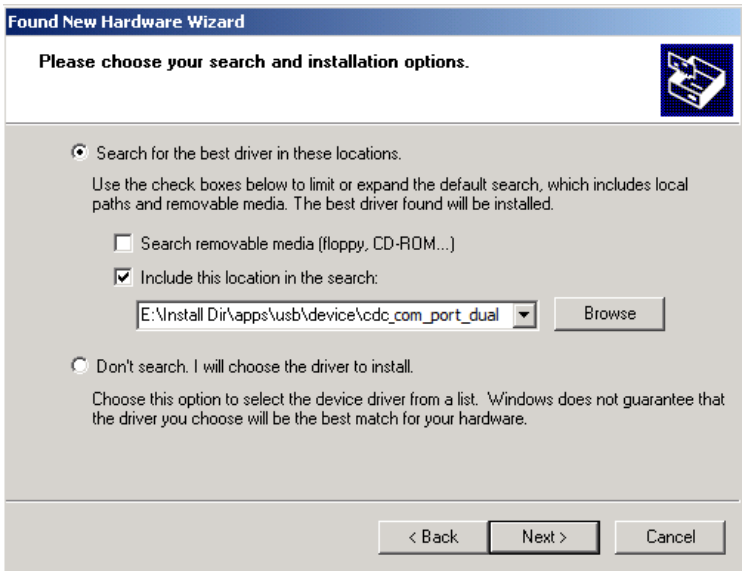
Provides instructions on how to build and run the CDC Dual COM Port demonstration.

## Description

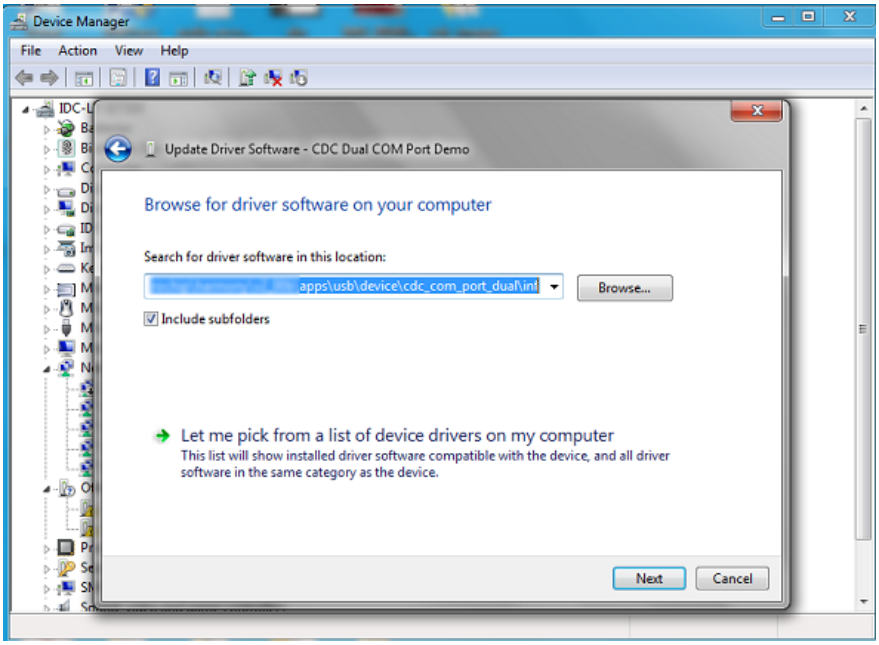
This demonstration allows the device to appear like dual serial (COM) ports to the host. Do the following to run this demonstration:

1. First compile and program the target device. While compiling, select the appropriate MPLAB X IDE project configuration based on the demonstration board. Refer to Building the Application for details.
2. Attach the device to the host. If the host is a personal computer and this is the first time you have plugged this device into the computer you may be prompted for a `.inf` file.

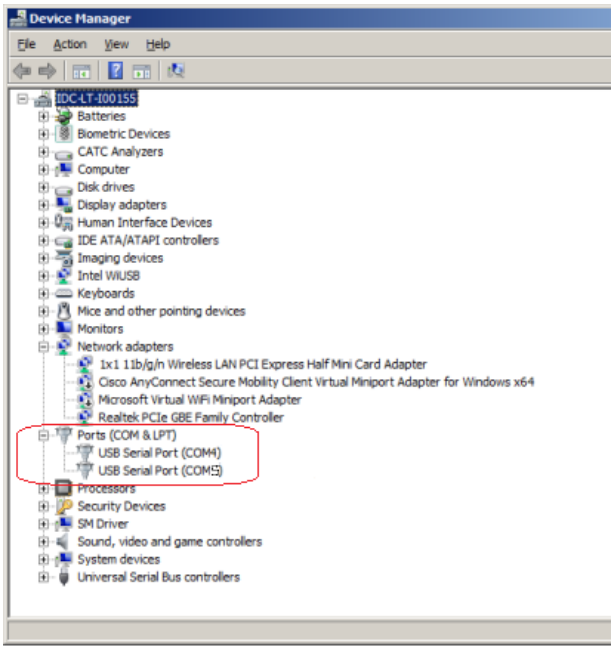

3. Select the "Install from a list or specific location (Advanced)" option. Specify the `<install-dir>/apps/usb/device/cdc_com_port_dual/inf` directory.



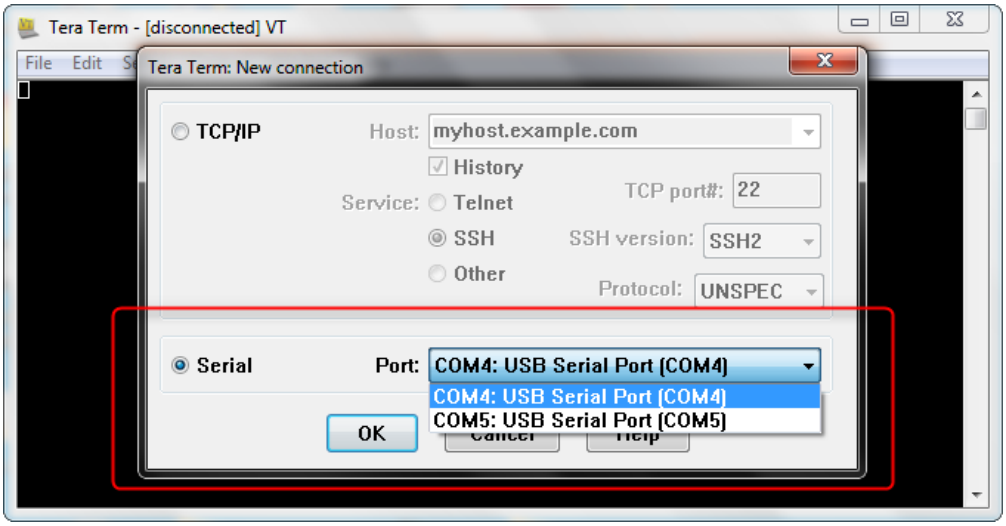
**Note:** As an option, to specify the driver, you may open the device manager and expand the Ports (COM & LPT) tab, and right click on "Update Driver Software…"



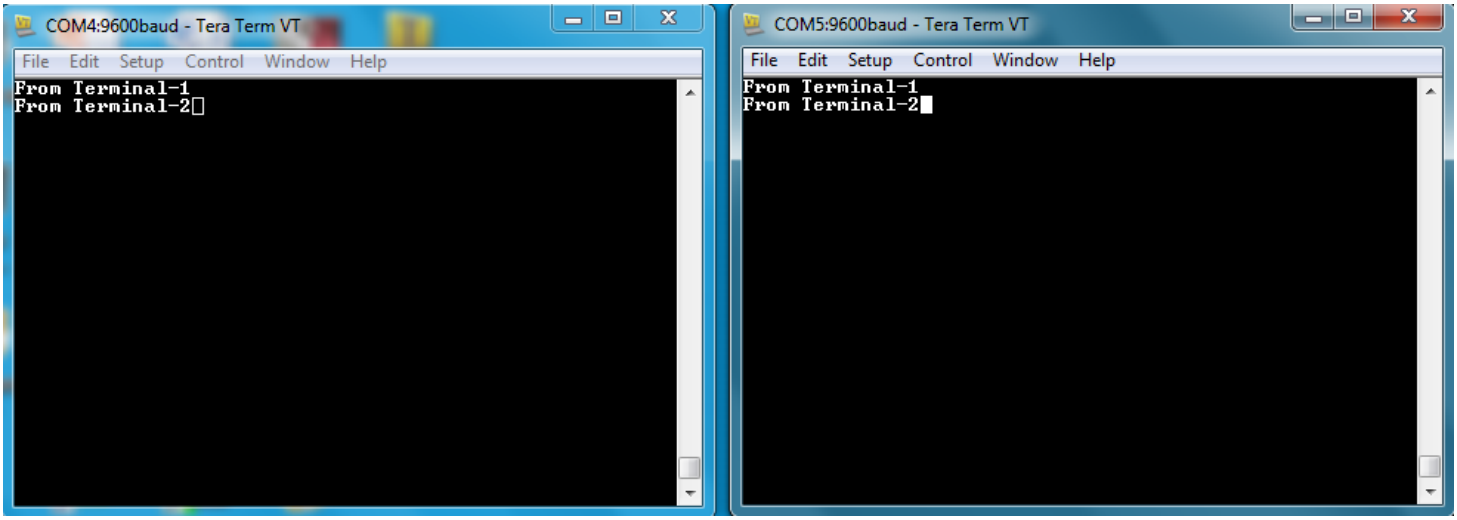

Verify that the enumerated USB device is seen as a virtual USB serial comport in Device Manager.




4. Once the device is successfully installed, open up two instances of a terminal program, such as HyperTerminal. Select the appropriate COM port for each of these terminal instances. The following screen shot shows the COM port selection for the Tera Term terminal program.




5. The LEDs on the demonstration board will indicate the USB state of the device, as described in the USB Device State and LED Indication Table in the Device section.
6. To run the demonstration, turn on local echo on both the terminals. For Tera Term terminal application, navigate to Setup->Terminal to turn on local echo. Type a character or string in one terminal window. The same character or string appears on the second terminal window. Similarly, any character typed in the second window appears in the first window. The following screen shot shows two instances of Tera Term.




**Note:** Some terminal programs, like HyperTerminal, require users to click the disconnect button before removing the device from the computer. Failing to do so may result in having to close and open the program again to reconnect to the device.

### *cdc_com_port_single*

Demonstrates a USB CDC device, emulating a serial COM port.

## Description

This demonstration application creates a USB CDC Device that enumerates as a single COM port on the host personal computer. The application demonstrates two-way communication between the USB device and the personal computer host.

### Building the Application

This section identifies the MPLAB X IDE project name and location and lists and describes the available configurations for the USB CDC Device Single COM Port Demonstration.

## Description

To build this project, you must open the `cdc_com_port_single.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/device/cdc_com_port_single`.

## MPLAB X IDE Project

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `cdc_com_port_single.X` | `<install-dir>/apps/usb/device/cdc_com_port_single/firmware` |

## MPLAB X IDE Project Configurations

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mx460_pim_e16_int_dyn | pic32mx460_pim+e16 | Select this MPLAB X IDE project configuration to run the demonstration on the Explorer 16 Development Board configured for Interrupt mode and dynamic operation. This configuration also requires PIC32MX460F512L Plug-In Module (PIM) and the USB PICtail Plus Daughter Board. |
| pic32mx_usb_sk2_poll_dyn | pic32mx_bt_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit II with the USB Device Stack configured for Polled mode and dynamic operation. |
| pic32mx_usb_sk3_int_dyn | pic32mx_usb_sk3 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit III configured for Interrupt mode and dynamic operation. |
| pic32mx_usb_sk2_int_dyn | pic32mx_usb_sk2 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit II configured for Interrupt mode and dynamic operation. |
| pic32mz_da_sk_intddr_int_dyn | pic32mz_da_sk_intddr | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ Embedded Graphics with Internal DRAM (DA) Starter Kit configured for Interrupt mode and dynamic operation. |
| pic32mz_ef_sk_int_dyn_micromips | pic32mz_ef_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit configured in microMIPS mode for Interrupt mode and dynamic operation. |
| pic32mz_ef_sk_int_dyn | pic32mz_ef_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit configured for Interrupt mode and dynamic operation. |
| pic32mz_ef_sk_poll_dyn | pic32mz_ef_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit with the USB Device Stack configured for Polled mode and dynamic operation. |
| pic32mx_125_sk_int_dyn | pic32mx_125_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MX1/2/5 Starter Kit with the USB Device Stack configured for Interrupt mode and dynamic operation. |

### Configuring the Hardware

Describes how to configure the supported hardware.

## Description

PIC32 USB Starter Kit II

Remove jumper JP2.

PIC32MZ Embedded Graphics with Internal DRAM (DA) Starter Kit

No hardware related configuration or jumper setting changes are necessary.

PIC32MZ EF Starter Kit

No hardware related configuration or jumper setting changes are necessary.

PIC32 USB Starter Kit III

Remove jumper JP1.

PIC32MX460F512L PIM

Jumper J10 should be removed. This plug-in module should be used along with the Explorer 16 Development Board and the USB PICtail Plus daughter board. The microcontroller PIM should be plugged into the PIM socket on the board. The USB PICtail Plus daughter board should be connected to the edge connector J9.

On the Explorer 16 Development Board:

- Switch S2 should be set to PIM
- Jumper JP2 should be in place

On the USB PICtail Plus Daughter Board:

- Jumper JP1 should be in place
- Jumper JP2 and JP4 should be removed

On the PIC32MX460F512L PIM:

- Keep jumper J10 open
- Keep all jumpers in J9 open

PIC32WK Wi-Fi Starter Kit

No hardware related configuration or jumper setting changes are necessary.

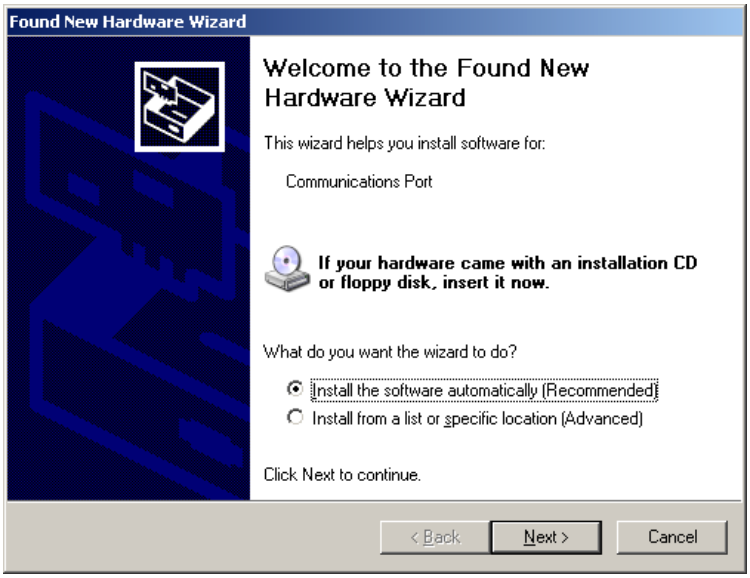
## Running the Demonstration

Provides instructions on how to build and run the CDC Single COM Port demonstration.

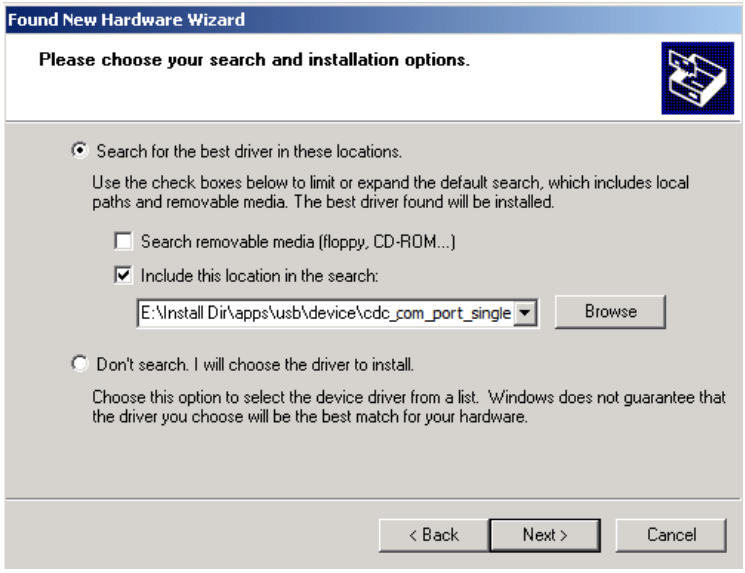
## Description

This demonstration allows the device to appear like a serial (COM) port to the host. Do the following to run this demonstration:

1. First compile and program the target device. While compiling, select the appropriate MPLAB X IDE project configuration based on the demonstration board. Refer to Building the Application for details.
2. Attach the device to the host. If the host is a personal computer and this is the first time you have plugged this device into the computer, you may be prompted for a `.inf` file.




3. Select the "Install from a list or specific location (Advanced)" option. Specify the `<install-dir>/apps/usb/device/cdc_com_port_single/inf` directory.




4. Once the device is successfully installed, open up a terminal program, such as HyperTerminal and select the appropriate COM port. On most machines this will be COM5 or higher. Set the communication properties to 9600 baud, 1 Stop bit and No parity, with Flow Control set to None.
5. The LEDs on the demonstration board will indicate the USB state of the device, as described in the USB Device State and LED Indication Table in the Device section.
6. Once connected to the device, there are two ways to run this example project:
   - a) Typing a key in the terminal window will result in the attached device echoing the next letter. Therefore, if the letter 'b' is pressed, the device will echo 'c'.
   - b) If the push button is pressed, the device will echo "PUSH BUTTON PRESSED" to the terminal window.

The following table shows the switch buttons to be pressed for different demonstration boards.

| Demonstration Board | Button |
|---|---|
| PIC32 USB Starter Kit II<br>PIC32 USB Starter Kit III<br>PIC32MZ Embedded Graphics with Internal DRAM (DA) Starter Kit<br>PIC32MZ Embedded Connectivity (EC) Starter Kit<br>PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit<br>PIC32WK Wi-Fi Starter Kit | SW1 |
| Explorer 16 Development Board | S3 |

**Note:** Some terminal programs, like HyperTerminal, require users to click the disconnect button before removing the device from the computer. Failing to do so may result in having to close and open the program again to reconnect to the device.

## *cdc_msd_basic*

Demonstrates a composite USB device emulating a COM port and Flash drive.

### Description

This demonstration application creates a composite USB Device that enumerates as a COM port and as Flash drive simultaneously.

## Building the Application

This section identifies the MPLAB X IDE project name and location and lists and describes the available configurations for the USB CDC MSD Composite Device Demonstration.

### Description

To build this project, you must open the `cdc_msd_basic.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/device/cdc_msd_basic`.

### MPLAB X IDE Project

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| cdc_msd_basic.X | <install-dir>/apps/usb/device/cdc_msd_basic/firmware |

## MPLAB X IDE Project Configurations

This table lists and describes the supported configurations of the demonstration, which are located within ./firmware/src/system_config.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mx_usb_sk2_int_dyn | pic32mx_usb_sk2 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit II configured for Interrupt mode and dynamic operation. |
| pic32mz_ef_sk_int_dyn | pic32mz_ef_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit configured for Interrupt mode and dynamic operation. |

### Configuring the Hardware

## Description

PIC32 USB Starter Kit II

Remove jumper JP2.

PIC32MZ EF Starter Kit

No hardware related configuration or jumper setting changes are necessary.

### Running the Demonstration

Provides instructions on how to build and run the USB CDC MSD Composite Device demonstration.

## Description

This demonstration application creates a composite USB Device that works simultaneously as a CDC and as a MSD device. This application combines the functionality of the cdc_com_port_single and msd_basic demonstration applications into one device.

Refer to Running the Demonstration section of the cdc_com_port_single demonstration and the Running the Demonstration section of the msd_basic demonstration for details on exercising the CDC and MSD device features, respectively.

The LEDs on the demonstration board will indicate the USB state of the device, as described in the USB Device State and LED Indication Table in the Device section.

### *cdc_serial_emulator*

This application demonstrates the use of the CDC device class in implementing a USB-to-Serial Dongle.

## Description

This application demonstrates the use of the CDC device class in implementing a USB-to-Serial Dongle. The application enumerates a COM port on the personal computer. Data received through the CDC USB interface is forwarded to a UART. Data received on the UART is forwarded to the CDC USB interface. This emulates a USB-to-Serial Dongle.

### Building the Application

This section identifies the MPLAB X IDE project name and location and lists and describes the available configurations for the USB CDC Device USB-to-Serial Demonstration.

## Description

To build this project, you must open the cdc_serial_emulator.X project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is <install-dir>/apps/usb/device/cdc_serial_emulator.

## MPLAB X IDE Project

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| cdc_serial_emulator.X | <install-dir>/apps/usb/device/cdc_serial_emulator/firmware |

## MPLAB X IDE Project Configurations

This table lists and describes the supported configurations of the demonstration, which are located within ./firmware/src/system_config.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mx795_pim_e16_int_dyn | pic32mx795_pim+e16 | Select this MPLAB X IDE project configuration to run the demonstration on the Explorer 16 Development Board configured for Interrupt mode and dynamic operation. This configuration also requires the PIC32MX795F512L Plug-In Module (PIM) and the USB PICtail Plus Daughter Board. |
| pic32mz_ef_sk_int_dyn | pic32mz_ef_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ EF Starter Kit configured for Interrupt mode and dynamic operation. |

## Configuring the Hardware

Describes how to configure the supported hardware.

## Description

PIC32MZ EF Starter Kit

No hardware related configuration or jumper setting changes are necessary.

PIC32MX795F512L CAN-USB PIM

Jumper J10 should be removed. Jumper J1 and J2 should connect to positions 1 and 2. This PIM should be used along with the Explorer 16 Development Board and the USB PICtail Plus daughter board. The microcontroller PIM should be plugged into the PIM socket on the board. The USB PICtail Plus daughter board should be connected to the edge connector J9.

On the Explorer 16 Development Board:

- Switch S2 should be set to PIM
- Jumper JP2 should be in place

On the USB PICtail Plus Daughter Board:

- Jumper JP1 should be in place
- Jumper JP2 and JP4 should be removed

On the PIC32MX795F512L PIM:

- Keep jumper J10 open
- Keep all jumpers in J9 open
- Jumper J1 should be shorted between positions 1 and 2. This configuration is only applicable for the PIC32MX795F512L USB CAN PIM (MA320003), and not the PIC32MX795F512L USB PIM (MA320002).
- Jumper J2 should be shorted between positions 1 and 2. This configuration is only applicable for the PIC32MX795F512L USB CAN PIM (MA320003) and not the PIC32MX795F512L USB PIM (MA320002).

## Running the Demonstration

Provides instructions on how to build and run the CDC Serial Emulator Demonstration.

Prior to using this demonstration, it is recommended to review the MPLAB Harmony Release Notes for any known issues. A PDF copy of the release notes is provided in the <install-dir>/doc folder of your installation.

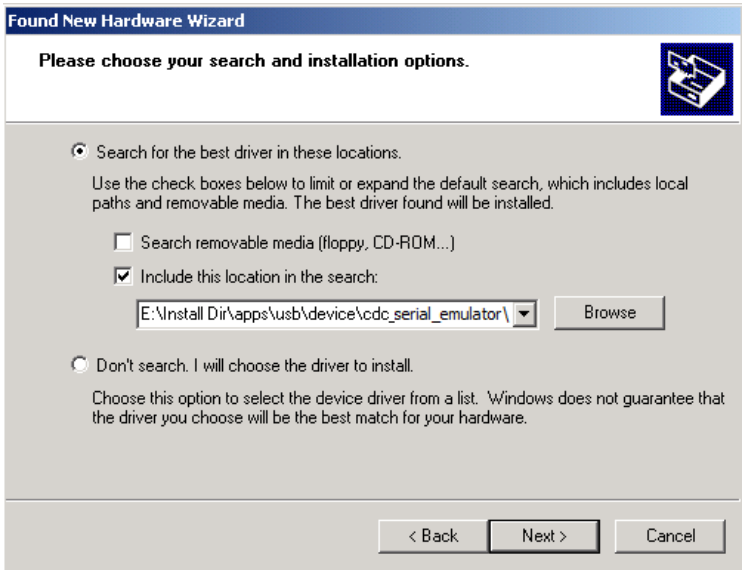
## Description

This application demonstrates the use of the CDC Device class in implementing a USB-to-Serial Dongle. The application enumerates a COM port on the personal computer. Data received through the CDC USB interface is forwarded to a UART. Data received on the UART is forwarded to the CDC USB interface. This emulates a USB-to-Serial Dongle.

1. Open the project in MPLAB X IDE and select the desired configuration.
2. Build the code and program the device.
3. Depending on the hardware in use, do one of the following:

- If you are using the Explorer 16 board, connect the mini-B device connector on the USB PICtail Plus Daughter Board to the personal computer
- If you a are using the PIC32MZ EF starter kit, connect the micro-USB device connector to the personal computer


7. Select the "Install from a list or specific location (Advanced)" option. Specify the `<install-dir>/apps/usb/device/cdc_serial_emulator/inf` directory.




The LEDs on the demonstration board will indicate the USB state of the device, as described in the USB Device State and LED Indication Table in the Device section.

8. Open a terminal emulation program of your choice and select the enumerated USB COM port.

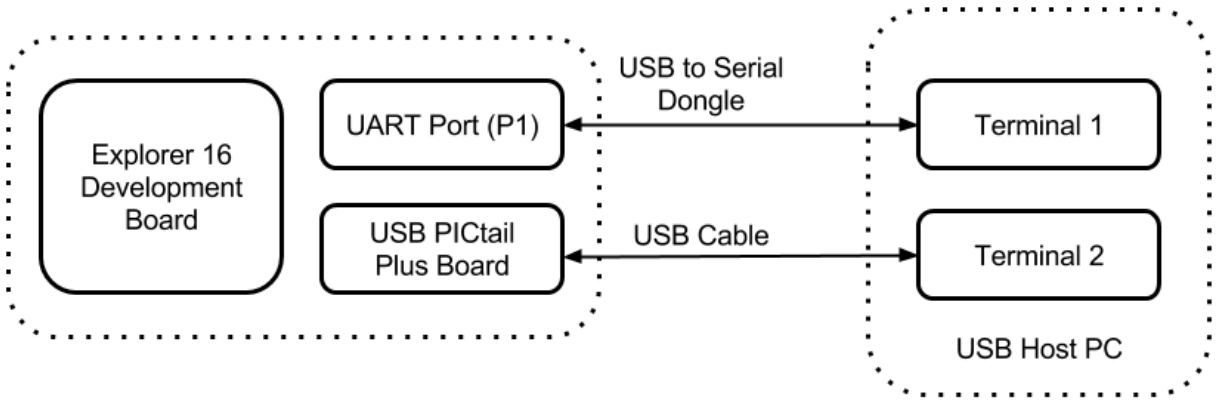
9. Connect the USB-to-Serial Dongle to the same personal computer.

10. Open another instance of the terminal emulation program and select the USB-to-Serial Dongle.

11. Connect the serial connector of the USB-to-Serial Dongle to the UART connector (P1) on the Explorer 16 Development Board.

12. Choose a baud rate of 9600, 1 Stop bit and no parity while opening both of the terminal emulation programs.

The setup should be similar to the following diagram.




Any text entered into the terminal 1 program will be echoed on terminal 2 and vice versa.

## *cdc_serial_emulator_msd*

Demonstrates a USB to Serial Dongle combined with a MSD class.

### Description

This demonstration application creates a USB Device that combines the functionality of the cdc_serial_emulator and msd_basic demonstration applications.

## Building the Application

This section identifies the MPLAB X IDE project name and location and lists and describes the available configurations for the this demonstration application.

### Description

To build this project, you must open the `cdc_serial_emulator_msd.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/cdc_serial_emulator_msd`.

**MPLAB X IDE Project**

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `cdc_serial_emulator_msd.X` | `<install-dir>/apps/usb/device/cdc_serial_emulator_msd/firmware` |

**MPLAB X IDE Project Configurations**

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mx795_pim_e16_int_dyn | pic32mx795_pim+e16 | Select this MPLAB X IDE project configuration to run the demonstration on the Explorer 16 Development Board configured for Interrupt mode and dynamic operation. This configuration also requires the PIC32MX795F512L Plug-In Module (PIM) and the USB PICtail Plus Daughter Board. |

## Configuring the Hardware

Describes how to configure the supported hardware.

### Description

PIC32MX795F512L CAN-USB PIM

Jumper J10 should be removed. Jumper J1 and J2 should connect to positions 1 and 2. This PIM should be used along with the Explorer 16 Development Board and the USB PICtail Plus daughter board. The microcontroller PIM should be plugged into the PIM socket on the board. The USB PICtail Plus daughter board should be connected to the edge connector J9.

On the Explorer 16 Development Board:

- Switch S2 should be set to PIM
- Jumper JP2 should be in place

On the USB PICtail Plus Daughter Board:

- Jumper JP1 should be in place
- Jumper JP2 and JP4 should be removed

On the PIC32MX795F512L PIM:

- Keep jumper J10 open.
- Keep all jumpers in J9 open
- Jumper J1 should be shorted between positions 1 and 2
- Jumper J2 should be shorted between positions 1 and 2

### Running the Demonstration

Provides instructions on how to build and run the demonstration.

#### Description

This demonstration functions as a composite USB Device that combines the features of the devices created by the cdc_serial_emulator and the msd_basic demonstration applications. Refer to Running the Demonstration section of the cdc_serial_emulator demonstration and Running the Demonstration section of the msd_basic demonstration for details on exercising the CDC and MSD functions, respectively.

The LEDs on the demonstration board will indicate the USB state of the device, as described in the USB Device State and LED Indication Table in the Device section.

## *hid_basic*

This demonstration application creates a custom HID device that can be controlled by a personal computer-based utility.

#### Description

This application creates a custom HID device that can be controlled by a personal computer-based utility. The device allows the USB Host utility to control the LEDs on the board and query the status of a switch.

### Building the Application

This section identifies the MPLAB X IDE project name and location and lists and describes the available configurations for the USB HID Basic Demonstration.

#### Description

To build this project, you must open the `hid_basic.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/device/hid_basic`.

#### MPLAB X IDE Project

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `hid_basic.X` | `<install-dir>/apps/usb/device/hid_basic/firmware` |

#### MPLAB X IDE Project Configurations

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mx460_pim_e16_int_dyn | pic32mx460_pim+e16 | Select this MPLAB X IDE project configuration to run the demonstration on the Explorer 16 Development Board configured for Interrupt mode and dynamic operation. This configuration also requires the PIC32MX460F512L Plug-In Module (PIM) and the USB PICtail Plus Daughter Board. |
| pic32mx_usb_sk3_int_dyn | pic32mx_usb_sk3 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit III configured for Interrupt mode and dynamic operation. |
| pic32mx_usb_sk2_int_dyn | pic32mx_usb_sk2 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit II configured for Interrupt mode and dynamic operation. |

| | | |
|---|---|---|
| pic32mz_ef_sk_int_dyn | pic32mz_ef_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit configured for Interrupt mode and dynamic operation. |

## Configuring the Hardware

Describes how to configure the supported hardware.

### Description

PIC32 USB Starter Kit II

Remove jumper JP2.

PIC32MZ EF Starter Kit

No hardware related configuration or jumper setting changes are necessary.

PIC32 USB Starter Kit III

Remove jumper JP1.

PIC32MX460F512L PIM

Jumper J10 should be removed. This plug-in module should be used along with the Explorer 16 Development Board and the USB PICtail Plus daughter board. The microcontroller PIM should be plugged into the PIM socket on the board. The USB PICtail Plus daughter board should be connected to the edge connector J9.

On the Explorer 16 Development Board:

- Switch S2 should be set to PIM
- Jumper JP2 should be in place

On the USB PICtail Plus Daughter Board:

- Jumper JP1 should be in place
- Jumper JP2 and JP4 should be removed

On the PIC32MX460F512L PIM:

- Keep jumper J10 open
- Keep all jumpers in J9 open

## Running the Demonstration

Provides instructions on how to build and run the HID Basic demonstration.

### Description

This demonstration uses the selected hardware platform as a HID class USB device, but uses the HID class for general purpose I/O operations. While compiling, select the appropriate MPLAB X IDE project configuration based on the demonstration board. Refer to Building the Application for details.

Typically, the HID class is used to implement human interface products, such as mice and keyboards. The HID protocol, is however, quite flexible, and can be adapted and used to send/receive general purpose data to/from a USB device. Using the HID class for general purpose I/O operations is quite advantageous, in that it does not require any kind of custom driver installation process. HID class drivers are already provided by and are distributed with common operating systems. Therefore, upon plugging in a HID class device into a typical computer system, no user installation of drivers is required, the installation is fully automatic.

The LEDs on the demonstration board will indicate the USB state of the device, as described in the USB Device State and LED Indication Table in the Device section.

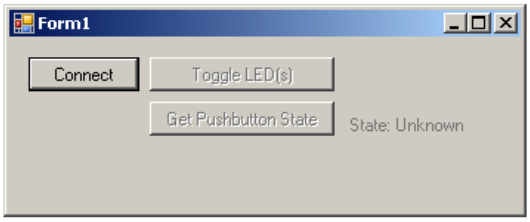
HID devices primarily communicate through one interrupt IN endpoint and one interrupt OUT endpoint. In most applications, this effectively limits the maximum achievable bandwidth for full speed HID devices to 64 kBytes/s of IN traffic, and 64 kBytes/s of OUT traffic (64 kB/s, but effectively "full duplex").

The `GenericHIDSimpleDemo.exe` program, and the associated firmware demonstrate how to use the HID protocol for basic general purpose USB data transfer.

Before you can run the `GenericHIDSimpleDemo.exe` executable, you will need to have the Microsoft® .NET Framework Version 2.0 Redistributable Package (later versions are probably acceptable, but have not been tested) installed on your computer. Programs that were built in the Visual Studio® .NET languages require the .NET redistributable package. The redistributable package can be freely downloaded from Microsoft's website. Users of Windows Vista® operating systems will not need to install the .NET framework, as it comes preinstalled as part of the operating system.

## Launching the Application

To launch the application, simply double click the executable `GenericHIDSimpleDemo.exe` in the `<install-dir>\apps\usb\device\hid_basic\bin` directory. A property sheet similar to the following should appear:




**Note:** If instead of this window, an error message appears while trying to launch the application, it is likely the Microsoft .NET Framework Version 2.0 Redistributable Package has not yet been installed. Please install it and try again.

## Send/Receive Packets

To begin sending/receiving packets to the device, you must first find and connect to the device. As configured by default, the application is looking for HID class USB devices with VID = 0x04D8 and PID = 0x003F. The device descriptor in the firmware project meant to be used with this demonstration uses the same VID/PID. If you plug in a USB device programmed with the correct precompiled `.hex` file, and click **Connect**, the other push buttons should become enabled. If clicking **Connect** has no effect, it is likely the USB device is either not connected, or has not been programmed with the correct firmware.

Clicking **Toggle LED(s)** should send a single packet of general purpose generic data to the HID class USB peripheral device. The data will arrive on the interrupt OUT endpoint. The firmware has been configured to receive this generic data packet, parse the packet looking for the Toggle LED(s) command, and should respond appropriately by controlling the LED(s) on the demonstration board.

The Get Pushbutton State option will send one packet of data over the USB to the peripheral device (to the interrupt OUT endpoint) requesting the current push button state. The firmware will process the received Get Pushbutton State command, and will prepare an appropriate response packet depending upon the pushbutton state.

The following table shows the button that has to be pressed on the demonstration board to see the change in the push button state.

| Demonstration Board | Button |
|---|---|
| PIC32 USB Starter Kit II<br>PIC32 USB Starter Kit III<br>PIC32MZ Embedded Connectivity (EC) Starter Kit<br>PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit | SW1 |
| Explorer 16 Development Board | S3 |

# *hid_joystick*

Demonstrates a USB HID device emulating a joystick.

## Description

This demonstration application creates a custom HID joystick. This application is only intended to demonstrate creation of Joystick HID Report descriptors and may not be a definite end solution. The end application requirements may need the report descriptor to be modified.

### Building the Application

This section identifies the MPLAB X IDE project name and location and lists and describes the available configurations for the USB HID Joystick Demonstration.

## Description

To build this project, you must open the `hid_joystick.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/device/hid_joystick`.

## MPLAB X IDE Project

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `hid_joystick.X` | `<install-dir>/apps/usb/device/hid_joystick/firmware` |

## MPLAB X IDE Project Configurations

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mx460_pim_e16_int_dyn | pic32mx460_pim+e16 | Select this MPLAB X IDE project configuration to run the demonstration on the Explorer 16 Development Board configured for Interrupt mode and dynamic operation. This configuration also requires the PIC32MX460F512L Plug-In Module (PIM) and the USB PICtail Plus Daughter Board. |
| pic32mx_usb_sk3_int_dyn | pic32mx_usb_sk3 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit III configured for Interrupt mode and dynamic operation. |
| pic32mx_usb_sk2_int_dyn | pic32mx_usb_sk2 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit II configured for Interrupt mode and dynamic operation. |
| pic32mz_ef_sk_int_dyn | pic32mz_ef_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit configured for Interrupt mode and dynamic operation. |

## Configuring the Hardware

Describes how to configure the supported hardware.

### Description

PIC32 USB Starter Kit II

Remove jumper JP2.

PIC32MZ EF Starter Kit

No hardware related configuration or jumper setting changes are necessary.

PIC32 USB Starter Kit III

Remove jumper JP1.

PIC32MX460F512L PIM

Jumper J10 should be removed. This plug-in module should be used along with the Explorer 16 Development Board and the USB PICtail Plus daughter board. The microcontroller PIM should be plugged into the PIM socket on the board. The USB PICtail Plus daughter board should be connected to the edge connector J9.

On the Explorer 16 Development Board:

- Switch S2 should be set to PIM
- Jumper JP2 should be in place

On the USB PICtail Plus Daughter Board:

- Jumper JP1 should be in place
- Jumper JP2 and JP4 should be removed

On the PIC32MX460F512L PIM:

- Keep jumper J10 open
- Keep all jumpers in J9 open

## Running the Demonstration

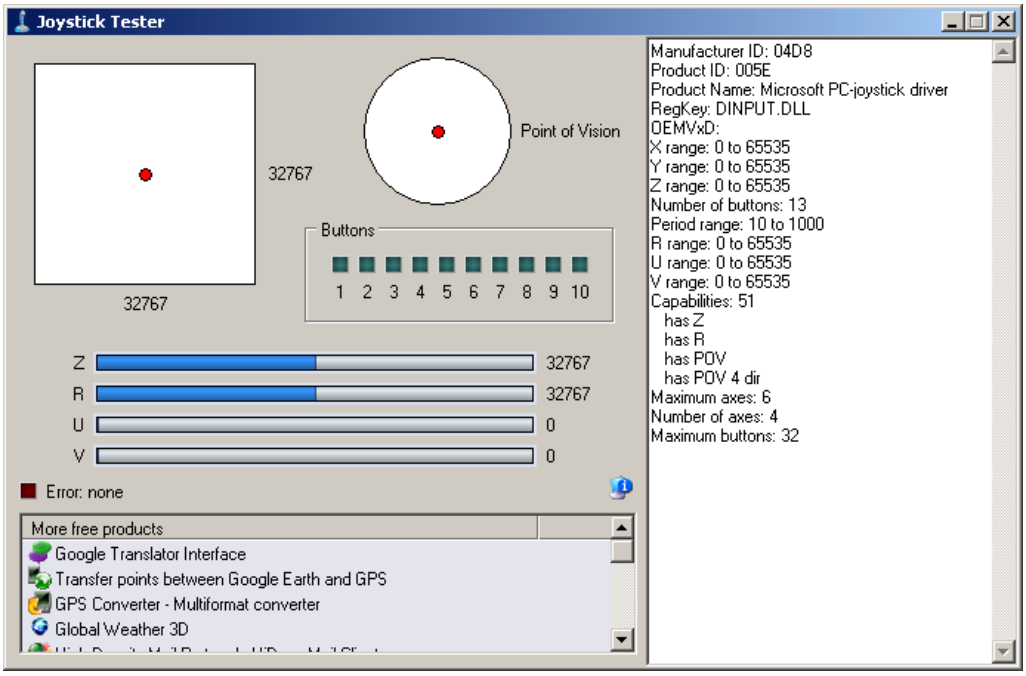
Provides instructions on how to build and run the USB HID Joystick demonstration.

### Description

This demonstration uses the selected hardware platform as a USB Joystick. Select the appropriate MPLAB X IDE project configuration based on the demonstration board. Refer to Building the Application for details.

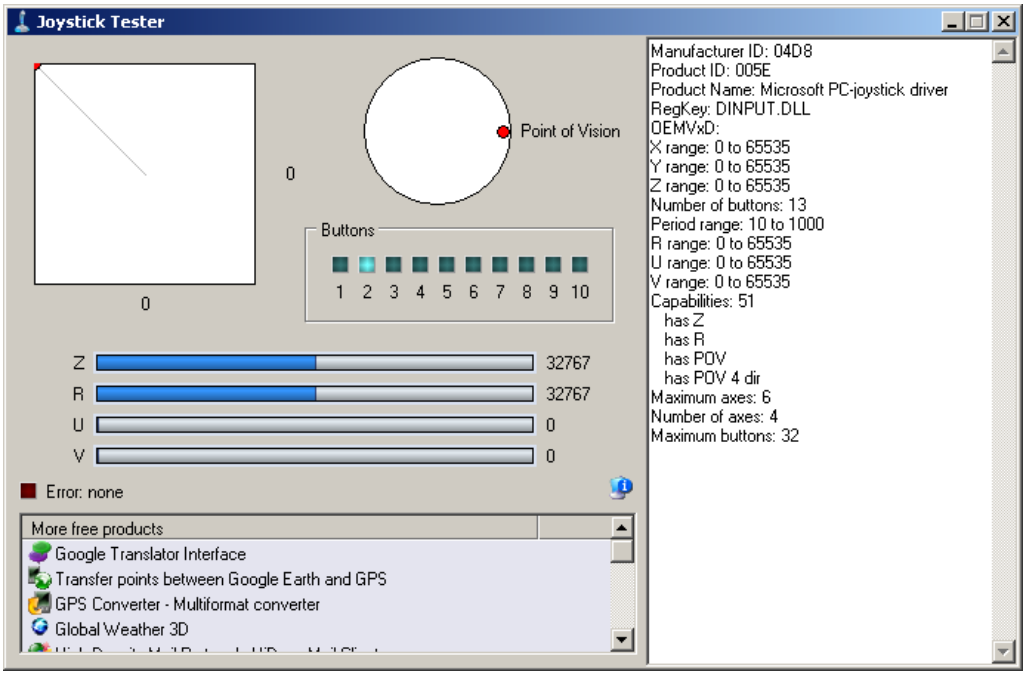
The LEDs on the demonstration board will indicate the USB state of the device, as described in the USB Device State and LED Indication Table in the Device section.

To test the joystick feature, navigate to the `<install-dir>/apps/usb/device/hid_joystick/bin` directory and open `JoystickTester.exe`:




Pressing the button will cause the device to:

- Indicate that the "x" button is pressed, but no others
- Move the hat switch to the "east" position
- Move the X and Y coordinates to their extreme values




The Following table shows the button that has to be pressed on the demonstration board to emulate the joystick.

| Demonstration Board | Button |
|---|---|
| PIC32 USB Starter Kit II<br>PIC32 USB Starter Kit III<br>PIC32MZ Embedded Connectivity (EC) Starter Kit<br>PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit | SW1 |
| Explorer 16 Development Board | S3 |

### *hid_keyboard*

Demonstrates a USB HID device, emulating a keyboard.

## Description

This demonstration application creates a Generic HID keyboard. Pressing a key on the board emulates a keyboard key press.

### Building the Application

This section identifies the MPLAB X IDE project name and location and lists and describes the available configurations for the USB HID Keyboard Demonstration.

## Description

To build this project, you must open the `hid_keyboard.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/device/hid_keyboard`.

## MPLAB X IDE Project

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `hid_keyboard.X` | `<install-dir>/apps/usb/device/hid_keyboard/firmware` |

## MPLAB X IDE Project Configurations

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mx460_pim_e16_int_dyn | pic32mx460_pim+e16 | Select this MPLAB X IDE project configuration to run the demonstration on the Explorer 16 Development Board configured for Interrupt mode and dynamic operation. This configuration also requires the PIC32MX460F512L Plug-In Module (PIM) and the USB PICtail Plus Daughter Board. |
| pic32mx_usb_sk3_int_dyn | pic32mx_usb_sk3 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit III configured for Interrupt mode and dynamic operation. |
| pic32mx_usb_sk2_int_dyn | pic32mx_usb_sk2 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit II configured for Interrupt mode and dynamic operation. |
| pic32mz_ef_sk_int_dyn | pic32mz_ef_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit configured for Interrupt mode and dynamic operation. |

### Configuring the Hardware

Describes how to configure the supported hardware.

## Description

PIC32 USB Starter Kit II

Remove jumper JP2.

PIC32MZ EF Starter Kit

No hardware related configuration or jumper setting changes are necessary.

PIC32 USB Starter Kit III

Remove jumper JP1.

PIC32MX460F512L PIM

Jumper J10 should be removed. This plug-in module should be used along with the Explorer 16 Development Board and the USB PICtail Plus daughter board. The microcontroller PIM should be plugged into the PIM socket on the board. The USB PICtail Plus daughter board should be

connected to the edge connector J9.

On the Explorer 16 Development Board:

- Switch S2 should be set to PIM
- Jumper JP2 should be in place

On the USB PICtail Plus Daughter Board:

- Jumper JP1 should be in place
- Jumper JP2 and JP4 should be removed

On the PIC32MX460F512L PIM:

- Keep jumper J10 open
- Keep all jumpers in J9 open

### Running the Demonstration

Provides instructions on how to build and run the USB HID Keyboard demonstration.

#### Description

This demonstration uses the selected hardware platform as a USB keyboard. While compiling, select the appropriate MPLAB X IDE project configuration based on the demonstration board. Refer to Building the Application for details.

The LEDs on the demonstration board will indicate the USB state of the device, as described in the USB Device State and LED Indication Table in the Device section.

Before pressing the button, select a window in which it is safe to type text freely. Pressing the button on the demonstration board will cause the device to print a character on the screen.

The following table shows the button that has to be pressed on the demonstration board to print a character.

| Demonstration Board | Button |
|---|---|
| PIC32 USB Starter Kit II<br>PIC32 USB Starter Kit III<br>PIC32MZ Embedded Connectivity (EC) Starter Kit<br>PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit | SW1 |
| Explorer 16 Development Board | S3 |

## *hid_mouse*

Demonstrates a USB HID device, emulating a mouse pointing device.

#### Description

This demonstration application creates a USB HID based two-button mouse device. When connected, the device emulates mouse operation by moving the cursor in a circular pattern.

### Building the Application

This section identifies the MPLAB X IDE project name and location and lists and describes the available configurations for the USB HID Mouse Demonstration.

#### Description

To build this project, you must open the `hid_mouse.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/device/hid_mouse`.

#### MPLAB X IDE Project

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `hid_mouse.X` | `<install-dir>/apps/usb/device/hid_mouse/firmware` |

#### MPLAB X IDE Project Configurations

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mx460_pim_e16_int_dyn | pic32mx460_pim+e16 | Select this MPLAB X IDE project configuration to run the demonstration on the Explorer 16 Development Board configured for Interrupt mode and dynamic operation. This configuration also requires the PIC32MX460F512L Plug-In Module (PIM) and the USB PICtail Plus Daughter Board. |
| pic32mx_usb_sk3_int_dyn | pic32mx_usb_sk3 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit III configured for Interrupt mode and dynamic operation. |
| pic32mx_usb_sk2_int_dyn | pic32mx_usb_sk2 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit II configured for Interrupt mode and dynamic operation. |
| pic32mz_ef_sk_int_dyn | pic32mz_ef_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit configured for Interrupt mode and dynamic operation. |

## Configuring the Hardware

Describes how to configure the supported hardware.

### Description

PIC32 USB Starter Kit II

Remove jumper JP2.

PIC32MZ EF Starter Kit

No hardware related configuration or jumper setting changes are necessary.

PIC32 USB Starter Kit III

Remove jumper JP1.

PIC32MX460F512L PIM

Jumper J10 should be removed. This plug-in module should be used along with the Explorer 16 Development Board and the USB PICtail Plus daughter board. The microcontroller PIM should be plugged into the PIM socket on the board. The USB PICtail Plus daughter board should be connected to the edge connector J9.

On the Explorer 16 Development Board:

- Switch S2 should be set to PIM
- Jumper JP2 should be in place

On the USB PICtail Plus Daughter Board:

- Jumper JP1 should be in place
- Jumper JP2 and JP4 should be removed

On the PIC32MX460F512L PIM:

- Keep jumper J10 open
- Keep all jumpers in J9 open

## Running the Demonstration

Provides instructions on how to build and run the HID Mouse Demonstration.

### Description

This demonstration uses the selected hardware platform as a USB mouse. While compiling, select the appropriate MPLAB X IDE project configuration based on the demonstration board. Refer to Building the Application for details.

The LEDs on the demonstration board will indicate the USB state of the device, as described in the USB Device State and LED Indication Table in the Device section.

Before connecting the board to the computer through the USB cable please be aware that the device will begin moving the mouse cursor on the computer. There are two ways to stop the device from allowing the cursor to continue to move. The first way is to disconnect the device from the computer. The second is to press the correct button on the hardware platform. Pressing the button again will cause the mouse cursor to start moving in a circle again.

The following table shows the button that has to be pressed on the demonstration board to stop the circular motion:

| Demonstration Board | Button |
|---|---|
| PIC32 USB Starter Kit II<br>PIC32 USB Starter Kit III<br>PIC32MZ Embedded Connectivity (EC) Starter Kit<br>PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit | SW1 |
| Explorer 16 Development Board | S3 |

### *hid_msd_basic*

Demonstrates a HID Device Class and MSD class composite USB Device.

## Description

This demonstration application creates a USB Device that combines the functionality of the hid_basic and msd_basic demonstration applications.

### Building the Application

This section identifies the MPLAB X IDE project name and location and lists and describes the available configurations for the this demonstration application.

## Description

To build this project, you must open the `hid_msd_basic.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/device/hid_msd_basic`.

## MPLAB X IDE Project

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `hid_msd_basic.X` | `<install-dir>/apps/usb/device/hid_msd_basic/firmware` |

## MPLAB X IDE Project Configurations

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mx_usb_sk2_int_dyn | pic32mx_usb_sk2 | Select this configuration to run the demonstration application on the PIC32 USB Starter Kit II configured for Interrupt mode and dynamic operation. |
| pic32mz_ef_sk_int_dyn | pic32mz_ef_sk | Select this configuration to run the demonstration application on the PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit configured for Interrupt mode and dynamic operation. |

### Configuring the Hardware

Describes how to configure the supported hardware.

## Description

PIC32 USB Starter Kit II

Remove jumper JP2.

PIC32MZ EF Starter Kit

No hardware related configuration or jumper setting changes are necessary.

### Running the Demonstration

Provides instructions on how to build and run the demonstration.

## Description

This demonstration functions as composite USB Device that combines the features of the devices created by the hid_basic and the msd_basic demonstration applications. Refer to Running the Demonstration section of the hid_basic demonstration and Running the Demonstration section of the msd_basic demonstration for details on exercising the HID and MSD functions, respectively.

The LEDs on the demonstration board will indicate the USB state of the device, as described in the USB Device State and LED Indication Table in the Device section.

### *msd_basic*

Demonstrates a USB MSD Device emulating a Flash Drive.

## Description

This demonstration application creates a Flash drive using the Mass Storage Device Class.

### Building the Application

This section identifies the MPLAB X IDE project name and location and lists and describes the available configurations for the USB MSD Basic Demonstration.

## Description

To build this project, you must open the `msd_basic.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/device/msd_basic`.

## MPLAB X IDE Project

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `msd_basic.X` | `<install-dir>/apps/usb/device/msd_basic/firmware` |

## MPLAB X IDE Project Configurations

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mx_bt_sk_int_dyn | pic32mx_bt_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 Bluetooth Starter Kit configured for Interrupt mode and dynamic operation. |
| pic32mx_usb_sk2_poll_dyn | pic32mx_usb_sk2 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit II with the USB Device Stack configured for Polled mode and dynamic operation.. |
| pic32mx_usb_sk2_int_dyn | pic32mx_usb_sk2 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit II configured for Interrupt mode and dynamic operation. |
| pic32mx_usb_sk3_int_dyn | pic32mx_usb_sk3 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit III configured for Interrupt mode and dynamic operation. |
| pic32mz_ef_sk_int_dyn | pic32mz_ef_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit configured for Interrupt mode and dynamic operation. |
| pic32mz_ef_sk_poll_dyn | pic32mz_ef_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit configured for Polled mode and dynamic operation. |

### Configuring the Hardware

Describes how to configure the supported hardware.

## Description

PIC32 USB Starter Kit II

Remove jumper JP2.

PIC32MZ EF Starter Kit

No hardware related configuration or jumper setting changes are necessary.

PIC32 USB Starter Kit III

Remove jumper JP1.

PIC32 Bluetooth Starter Kit

No hardware related configuration or jumper settings required.

PIC32MX460F512L PIM

Jumper J10 should be removed. This plug-in module should be used along with the Explorer 16 Development Board and the USB PICtail Plus daughter board. The microcontroller PIM should be plugged into the PIM socket on the board. The USB PICtail Plus daughter board should be connected to the edge connector J9.

On the Explorer 16 Development Board:

- Switch S2 should be set to PIM
- Jumper JP2 should be in place

On the USB PICtail Plus Daughter Board:

- Jumper JP1 should be in place
- Jumper JP2 and JP4 should be removed

### Running the Demonstration

Provides instructions on how to build and run the USB MSD Basic demonstration.

## Description

This demonstration uses the selected hardware platform as a logical drive on the computer using the internal Flash of the device as the drive storage media. Connect the hardware platform to a computer through a USB cable. The device should appear as a new drive on the computer named "Drive Name". The drive can used to store files.

The LEDs on the demonstration board will indicate the USB state of the device, as described in the USB Device State and LED Indication Table in the Device section.

**Note:** Reprogramming the development board will cause any stored files to be erased.

### *msd_fs_spiflash*

This application demonstrates accessing the SPI Flash connected to the PIC32 device as a media by multiple clients.

## Description

This application demonstrates accessing the SPI Flash connected to the PIC32 device as a media by multiple clients. When connected via USB to the Host Computer, the SPI Flash is shown as the storage media. The Host writes files to the media, which is later accessed by the application running on the PIC32 device using the File System.

### Building the Application

This section identifies the MPLAB X IDE project name and location and lists and describes the available configurations for the USB MSD File System SPI Flash Demonstration.

## Description

To build this project, you must open the `msd_fs_spiflash.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/device/msd_fs_spiflash`.

## MPLAB X IDE Project

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `msd_fs_spiflash.X` | `<install-dir>/apps/usb/device/msd_fs_spiflash/firmware` |

## MPLAB X IDE Project Configurations

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| bt_audio_dk_int_dyn | bt_audio_dk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 Bluetooth Audio Development Kit. |

### Configuring the Hardware

Describes how to configure the supported hardware.

## Description

PIC32 Bluetooth Audio Develoment Kit

Ensure that switch S1 is set to PIC32_MCLR.

### Running the Demonstration

Provides instructions on how to build and run the USB MSD File System SPI Flash demonstration.

## Description

This demonstration shows an example of:

- Accessing the media attached to PIC32 by multiple clients
- Application running on the PIC32 firmware accesses the media using the MPLAB Harmony File System

When connected to the USB Host the very first time, the user is expected to format the media and create a file named `FILE.TXT` in the root directory of the media. The user can update the file to provide input for the application to glow the LEDs present on the development kit. The application running on the PIC32 reads and interprets the data present in the file and accordingly turns ON or OFF the LEDs LED8 and LED9 of the development kit. The format of input in the file `FILE.TXT` should be as follows:

- For turning ON an LED:
  - LED8:1
  - LED9:1
- For turning OFF an LED:
  - LED8:0
  - LED9:0

After having set the appropriate values in the file, the user can then press and release the wwitch SW1 located on the development kit for the MPLAB Harmony File System running on the PIC32 to act upon the contents of the file.

The FS state machine of the demonstration is only triggered by the switch SW1. When the user presses and releases SW1 the following occurs:

- LED5 is turned ON to indicate that the FS state machine is running
- The USB is detached
- The file system on the SPI Flash is mounted
- The contents of `FILE.TXT` is read and acted upon. Depending on the values set in the file, the LEDs are either turned ON or OFF.
- Next, the file system is unmounted and the USB is reattached
- LED5 is turned OFF to indicate that FS state machine is no longer running
- If LED6 is turned ON during any part of the demonstration, this indicates the demonstration has failed

### *msd_multiple_luns*

This topic demonstrates data transfer between two storage media - SD card and non-volatile memory (NVM) - and a computer through USB Mass Storage Device (MSD).

## Description

This application demonstrates the creation of a USB device with multiple logical units. The storage media, SD Card, acts as one logical unit, and the NVM acts as the second logical unit. Data transfer between a computer and the logical units (SD Card / NVM) takes place through USB MSD.

### Building the Application

This section identifies the MPLAB X IDE project name and location, and then lists and describes the available configurations for the USB MSD multiple LUNs demonstration.

## Description

To build this project, you must open the `msd_multiple_luns.X` project in MPLAB X IDE, and then select the desired configuration. The following tables lists and describes the project and the supported configurations. The parent folder for these files is

`<install-dir>/apps/usb/device/msd_multiple_luns`

## MPLAB X IDE Project

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `msd_multiple_luns.X` | `<install-dir>/apps/usb/device/msd_multiple_luns/firmware` |

## MPLAB X IDE Project Configurations

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mx470_curiosity | pic32mx470_curiosity | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MX470 Curiosity board with the USB device stack configured for Interrupt mode and full speed operation. The LUN0 media type is configured as SD Card and LUN1 media type is configured as NVM. |
| pic32mz_ef_curiosity | pic32mz_ef_curiosity | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ EF Curiosity board with the USB device stack configured for Interrupt mode and high speed operation. The LUN0 media type is configured as SD Card and LUN1 media type is configured as NVM. |

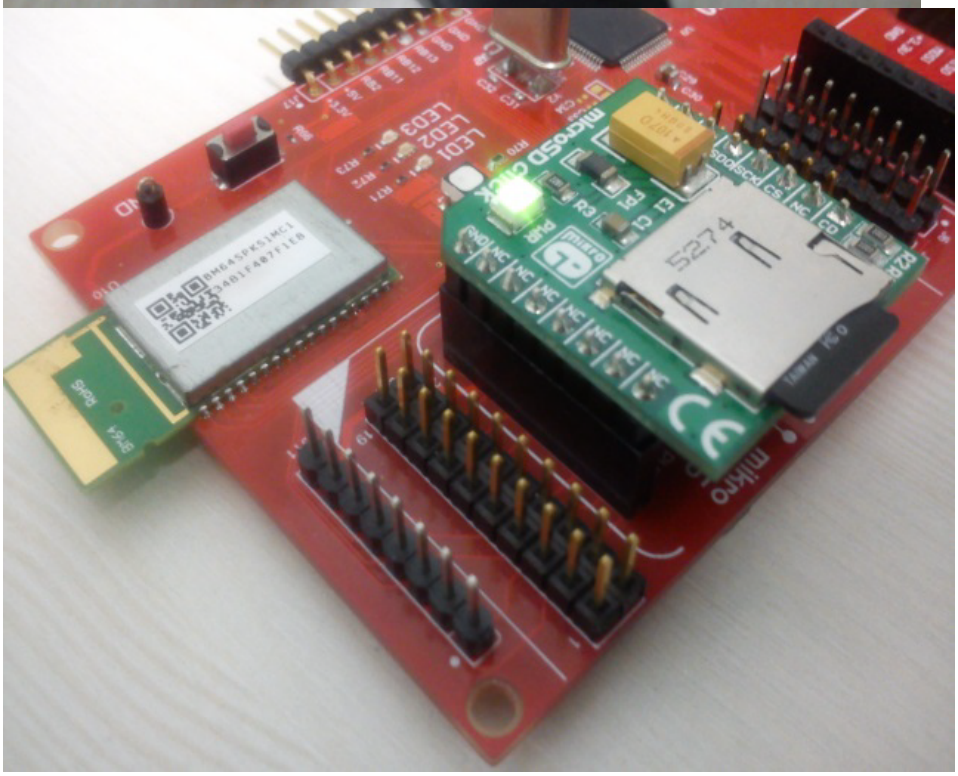
## Configuring the Hardware

This section describes how to configure the supported hardware.

## Description

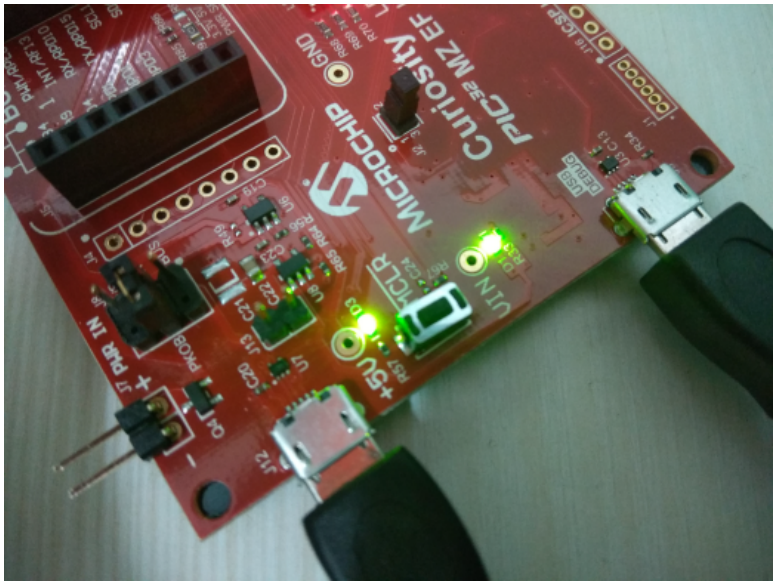
**PIC32MX470 Curiosity Development Board**

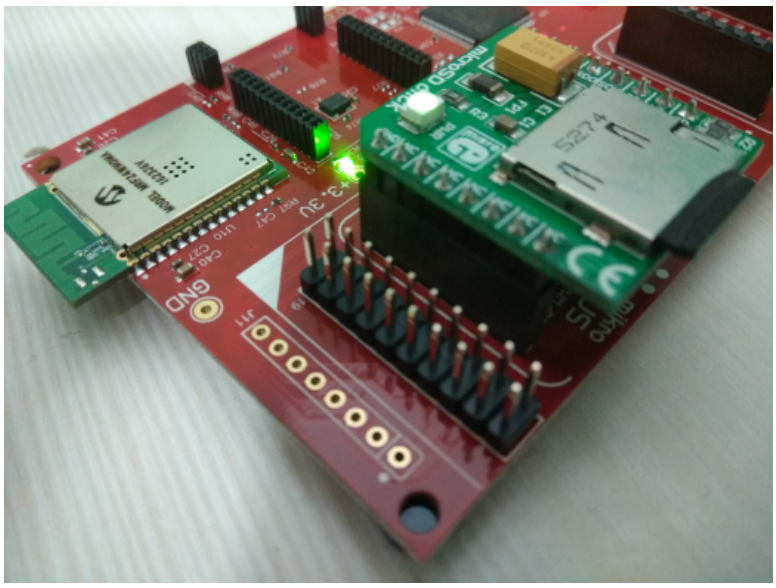
1. Ensure that a jumper is placed at 4-3 on J8, to select supply from debug USB connector.
2. Mount the SD Click board, "microSD click" from MikroElektronika (http://www.mikroe.com/click/microsd/) on the mikro bus interface J10.
3. Plug a micro SD card into the microSD click board card slot.
4. Power the PIC32MX470 Curiosity Development Board from a Host PC through a Type-A male to mini-B USB cable connected to Mini-B port (J3).
5. Connect a Type-A male to micro USB cable to the micro USB port (J12) on PIC32MX470 Curiosity Development Board.

**PIC32MZ EF Curiosity Development Board**

1. Ensure that a jumper is placed at 4-3 on J8, to select supply from debug USB connector.
2. Mount the SD Click board, "microSD click" from MikroElektronika (http://www.mikroe.com/click/microsd/) on the mikro bus interface J10.
3. Plug a micro SD card into the microSD click board card slot.
4. Power the PIC32MZ EF Curiosity Development Board from a Host PC through a Type-A male to micro USB cable connected to micro USB port (J3).
5. Connect a Type-A male to micro USB cable to the micro USB port (J12) on PIC32MZ EF Curiosity Development Board.




## Running the Demonstration

This section provides instructions about how to build and run the USB MSD Multiple LUNs demonstration.

## Description

This demonstration uses SD card and NVM as drive storage media and shows them as two logical drives on the computer.

- Connect the hardware platform to a computer through a USB cable.
  - The device should appear as two new drives on the computer.
  - The NVM media should appear as "Drive Name" and should have a sample “FILE.txt” file. The drive name for the SD card media depends on the micro SD card vendor. The drives can then be used to store files.
  - The LEDs on the demonstration board will indicate the USB state of the device, as described in the USB Device State and LED Indication Table in the Device section.

**Note:** Reprogramming the development board will cause any stored files in the NVM media to be erased.

## *msd_sdcard*

Demonstrates data transfer from a SD card and a computer through USB MSD.

## Description

This application demonstrates the usage of a SD card reader through the USB Mass Storage Device (MSD) class to transfer data between a computer and SD card. High-Speed USB is used for communication between the Host computer and the PIC32 device, while a SD card is used as the storage medium.

### Building the Application

This section identifies the MPLAB X IDE project name and location and lists and describes the available configurations for the USB MSD SD Card Demonstration.

## Description

To build this project, you must open the `msd_sdcard.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/device/msd_sdcard`.

## MPLAB X IDE Project

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `msd_sdcard.X` | `<install-dir>/apps/usb/device/msd_sdcard/firmware` |

## MPLAB X IDE Project Configurations

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mz_ec_sk_int_dyn | pic32mz_ec_sk+meb2 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ EC Starter Kit connected to the MEB II. The media drivers are configured for Interrupt mode and dynamic operation. |

### Configuring the Hardware

Describes how to configure the supported hardware.

## Description

PIC32MZ Embedded Connectivity (EC) Starter Kit and Multimedia Expansion Board II (MEB II)

No hardware related configuration or jumper settings required.

### Running the Demonstration

Provides instructions on how to build and run the USB MSD SD Card demonstration.

## Description

This demonstration uses the selected hardware platform as a logical drive on the computer using the SD card as the drive storage media. Connect the hardware platform to a computer through a USB cable. The device should appear as a new drive on the computer named "Drive Name". The drive can then be used to store files.

The LEDs on the demonstration board will indicate the USB state of the device, as described in the USB Device State and LED Indication Table in the Device section.

### *vendor*

Demonstrates a custom USB Device created by using the USB Device Layer Endpoint functions.

## Description

This demonstration application creates a custom USB device using the USB Device Layer Endpoint functions.

## Building the Application

This section identifies the MPLAB X IDE project name and location and lists and describes the available configurations for the Vendor USB Device Demonstration.

### Description

To build this project, you must open the `vendor.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/device/vendor`.

### MPLAB X IDE Project

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `vendor.X` | `<install-dir>/apps/usb/device/vendor/firmware` |

### MPLAB X IDE Project Configurations

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mx460_pim_e16_int_dyn | pic32mx460_pim+e16 | Select this MPLAB X IDE project configuration to run the demonstration on the Explorer 16 Development Board configured for Interrupt mode and dynamic operation. This configuration also requires the PIC32MX460F512L Plug-In Module (PIM) and the USB PICtail Plus Daughter Board. |
| pic32mx_usb_sk3_int_dyn | pic32mx_usb_sk3 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit III configured for Interrupt mode and dynamic operation. |
| pic32mx_usb_sk2_int_dyn | pic32mx_usb_sk2 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit II configured for Interrupt mode and dynamic operation. |
| pic32mz_ef_sk_int_dyn | pic32mz_ef_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit configured for Interrupt mode and dynamic operation. |

## Configuring the Hardware

Describes how to configure the supported hardware.

### Description

PIC32 USB Starter Kit II

Remove jumper JP2.

PIC32MZ EF Starter Kit

No hardware related configuration or jumper setting changes are necessary.

PIC32 USB Starter Kit III

Remove jumper JP1.

PIC32MX460F512L PIM

Jumper J10 should be removed. This plug-in module should be used along with the Explorer 16 Development Board and the USB PICtail Plus daughter board. The microcontroller PIM should be plugged into the PIM socket on the board. The USB PICtail Plus daughter board should be connected to the edge connector J9.

On the Explorer 16 Development Board:

- Switch S2 should be set to PIM
- Jumper JP2 should be in place

On the USB PICtail Plus Daughter Board:

- Jumper JP1 should be in place

- Jumper JP2 and JP4 should be removed

On the PIC32MX460F512L PIM:

- Keep jumper J10 open
- Keep all jumpers in J9 open

### Running the Demonstration

Provides instructions on how to build and run the Vendor USB Device demonstration.

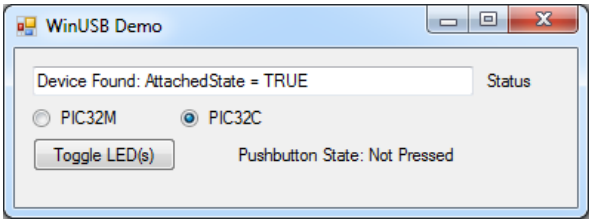
## Description

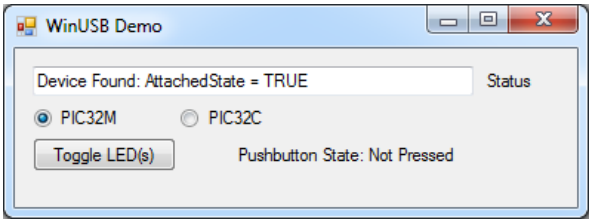
The Vendor device can be exercised by using the WinUSB PnP Demonstration application, which is provided in your installation of MPLAB Harmony.

The LEDs on the demonstration board will indicate the USB state of the device, as described in the USB Device State and LED Indication Table in the Device section.

This application allows the state of the LEDs on the board to be toggled and indicates the state of a switch (pressed/released) on the board.

To launch the application, double click `WinUSB PnP Demo.exe` located in `<install dir>/apps/usb/device/vendor/bin`. A dialog box similar to the following should appear:




The appropriate device family that is under testing should be selected in the utility. Pressing the Toggle LED button will cause the LED on the board to toggle. The Pushbutton State field in the application indicates the state of a button on connected USB Device. Pressing the switch on the development board will update the Pressed/Not Pressed status of the Pushbutton State field.

| Demonstration Board | Button |
|---|---|
| PIC32 USB Starter Kit II<br>PIC32 USB Starter Kit III<br>PIC32MZ Embedded Connectivity (EC) Starter Kit<br>PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit | SW1 |
| Explorer 16 Development Board | S3 |

**Note:** The device family under test should be selected appropriately. An incorrect selection will result in an invalid push button status.

### Host

This section describes the USB Host demonstrations.

### *audio_speaker*

This application demonstrates the use of the Audio v1.0 Host Class Driver to enumerate and operate an audio speaker device.

## Description

This application demonstrates the use of the Audio v1.0 Host Class Driver to enumerate and an audio speaker device. The application uses the USB Host Layer and Audio 1.0 class driver to enumerate an Audio v1.0 USB device. The demonstration host application then operates and uses the functionality of the attached audio speaker device.

## Building the Application

This section identifies the MPLAB X IDE project name and location and lists and describes the available configurations for the USB Host Audio Speaker Demonstration.

### Description

To build this project, you must open the `audio_speaker.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/host/audio_speaker`.

### MPLAB X IDE Project

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `audio_speaker.X` | `<install-dir>/apps/usb/host/audio_speaker/firmware` |

### MPLAB X IDE Project Configurations

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mx_usb_sk2_int_dyn | pic32mx_usb_sk2 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit II configured for Interrupt mode and dynamic operation. |
| pic32mz_ef_sk_int_dyn | pic32mz_ef_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit configured for Interrupt mode and dynamic operation. |

## Configuring the Hardware

Describes how to configure the supported hardware.

### Description

PIC32 USB Starter Kit II

JP2 should be in place.

PIC32MZ EF Starter Kit

No hardware related configuration or jumper setting changes are necessary.

## Running the Demonstration

Provides instructions on how to build and run the USB Host Audio v1.0 Basic Demo.

### Description

This application demonstrates the use of the Audio v1.0 Host Class Driver to enumerate and operate an Audio v1.0 Device. The application uses the USB Host layer and Audio v1.0 class driver to enumerate a Audio v1.0 USB device. The demonstration host application then operates and uses the functionality of the attached Audio v1.0 Device.

Prior to using this demonstration, it is recommended to review the MPLAB Harmony Release Notes for any known issues. A PDF copy of the release notes is provided in the `<install-dir>/doc` folder of your installation.

1. Open the project in MPLAB X IDE and select the desired project configuration.
2. Build the code and program the device.
3. Attach a commercially available USB speaker to the board.
4. LED1 is turned ON if the attached device is accepted by the Audio 1.0 class driver.
5. The speaker should produce a 1 kHz sine wave.
6. LED2 will continue blinking if the demonstration application cannot accept the device.
7. Press switch SW1 to mute the audio.
8. Press switch SW2 to unmute the audio

### *cdc_basic*

This application demonstrates the use of the CDC Host Class Driver to enumerate and operate a CDC Device.

## Description

This application demonstrates the use of the CDC Host Class Driver to enumerate and operate a CDC Device. The application uses the USB Host  layer and CDC class driver to enumerate a CDC USB device. The demonstration host application then operates and uses the functionality of the attached CDC Device.

### Building the Application

This section identifies the MPLAB X IDE project name and location and lists and describes the available configurations for the USB CDC Host Basic Demonstration.

## Description

To build this project, you must open the `cdc_basic.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/host/cdc_basic`.

## MPLAB X IDE Project

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `cdc_basic.X` | `<install-dir>/apps/usb/host/cdc_basic/firmware` |

## MPLAB X IDE Project Configurations

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mx_usb_sk2_int_dyn | pic32mx_usb_sk2 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit II configured for Interrupt mode and dynamic operation. |
| pic32mx_usb_sk2_poll_dyn | pic32mx_usb_sk2 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit II configured for Polled mode and dynamic operation. |
| pic32mz_ef_sk_int_dyn | pic32mz_ef_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit configured for Interrupt mode and dynamic operation. |

### Configuring the Hardware

Describes how to configure the supported hardware.

## Description

PIC32 USB Starter Kit II

JP2 should be in place.

PIC32MZ EF Starter Kit

No hardware related configuration or jumper setting changes are necessary.

### Running the Demonstration

Provides instructions on how to build and run the USB Host CDC Basic Demo.

## Description

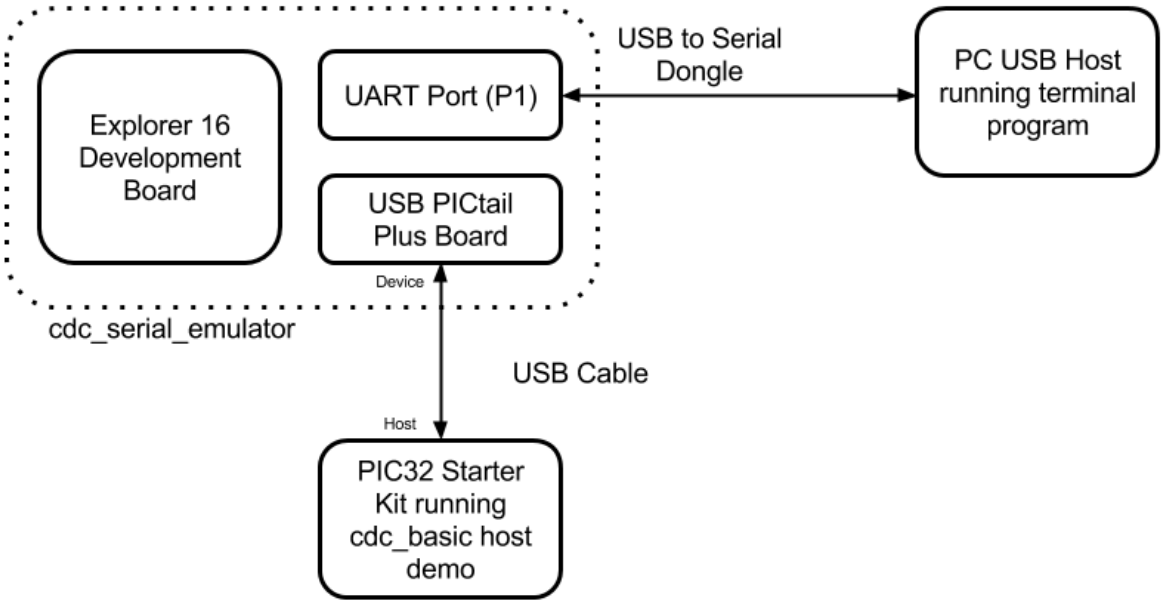
This application demonstrates the use of the CDC Host Class Driver to enumerate and operate a CDC Device. The application uses the USB Host  layer and CDC class driver to enumerate a CDC USB device. The demonstration host application then operates and uses the functionality of the attached CDC Device.

Prior to using this demonstration, it is recommended to review the MPLAB Harmony Release Notes for any known issues. A PDF copy of the

release notes is provided in the `<install-dir>/doc` folder of your installation.

1. Open the project in MPLAB X IDE and select the desired project configuration.
2. Build the code and program the device.
3. Follow the directions for setting up and running the cdc_serial_emulator USB device demonstration.
4. Connect the UART (P1) port on the Explorer 16 Development Board (running the cdc_serial_emulator demonstration) to a USB Host personal computer via a commercially available Serial-to-USB Dongle.
5. Start a terminal program on the USB Host personal computer and select the Serial-to-USB Dongle as the communication port. Select the baud rate as 9600, no parity, 1 Stop bit and no flow control.
6. Connect the mini – B connector on the USB PICtail Plus Daughter Board, of the cdc_serial_emulator demonstration setup, to the USB host connector on the starter kit. For PIC32M-based starter kits, connect to the on-board Type-A connector.
7. A prompt (LED :) will be displayed immediately on the terminal emulation program.
8. Pressing either the 1, 2, or 3 key on the USB Host keyboard will cause LEDs on the PIC32 starter kit (running the USB CDC Host application) to switch on, respectively. On PIC32M-based starter kits, the LEDs are LED1, LED2, and LED3.
9. The prompt will again be displayed on terminal emulation program, and step 8 can be repeated.

The setup should be similar to the following diagram.




The cdc_serial_emulator demonstration emulates a USB-to-Serial Dongle. The CDC Host (running the cdc_basic demonstration application) sends the prompt message to the CDC device. The CDC device forwards the prompt to the UART port from where it is transmitted to the personal computer USB Host through the USB-to-Serial Dongle. A key press on the personal computer USB Host is transmitted to the CDC device, which in turn presents the key press data to the CDC host. The cdc_basic demonstration then analyzes the key press data and switches on the respective LED.

### *cdc_msd*

Demonstrates host support for multiple device classes.

## Description

This demonstration application creates a USB Host that can support different device classes in one application.

### Building the Application

This section identifies the MPLAB X IDE project name and location and lists and describes the available configurations for this USB CDC MSD Host Demonstration.

## Description

To build this project, you must open the `cdc_msd.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/host/cdc_msd`.

## MPLAB X IDE Project

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| cdc_msd.X | <install-dir>/apps/usb/host/cdc_msd/firmware |

## MPLAB X IDE Project Configurations

This table lists and describes the supported configurations of the demonstration, which are located within ./firmware/src/system_config.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mx_usb_sk2_int_dyn | pic32mx_usb_sk2 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit II configured for Interrupt mode and dynamic operation. |
| pic32mz_ef_sk_int_dyn | pic32mz_ef_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit configured for Interrupt mode and dynamic operation. |

### Configuring the Hardware

Describes how to configure the supported hardware.

## Description

PIC32 USB Starter Kit II

JP2 should be in place.

PIC32MZ EF Starter Kit

No hardware related configuration or jumper setting changes are necessary.

### Running the Demonstration

Provides instructions on how to build and run the USB CDC MSD demonstration.

## Description

This demonstration application creates a USB Host application that enumerates a CDC and a MSD device. This application combines the functionality of the Host cdc_basic and msd_basic demonstration applications into one application. If a CDC device is connected, the demonstration application behaves like the cdc_basic host application. If a MSD device is connected, the demonstration application behaves like the msd_basic host application.

Refer to Running the Demonstration section of the host cdc_basic demonstration and the Running the Demonstration section of the host msd_basic demonstration for details on exercising the CDC and MSD host aspects of the demonstration.

### *hid_basic_keyboard*

Demonstrates using the USB HID Host Client driver with the Keyboard Usage driver to facilitate the use of a USB HID Keyboard with a PIC32 USB Host.

## Description

This application demonstrates the use of the USB HID Host Client Driver to enumerate and operate a HID keyboard device. The application uses the USB Host layer, HID Client driver and HID Keyboard Usage driver to enumerates a USB keyboard and understand keyboard press release events.

The keyboard events are displayed using a terminal emulator on a personal computer.

### Building the Application

This section identifies the MPLAB X IDE project name and location and lists and describes the available configurations for the USB HID Basic Keyboard Demonstration.

## Description

To build this project, you must open the hid_basic_keyboard.X project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/host/hid_basic_keyboard`.

**MPLAB X IDE Project**

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `hid_basic_keyboard.X` | `<install-dir>/apps/usb/host/hid_basic_keyboard/firmware` |

**MPLAB X IDE Project Configurations**

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mx795_pim_e16_int_dyn | pic32mx795_pim+e16 | Select this MPLAB X IDE project configuration to run the demonstration configured for Interrupt mode and dynamic operation on the PIC32MX795F512L PIM connected to the Explorer 16 Development Board with the USB PICtail Plus Daughter Board attached. |
| pic32mz_ef_sk_int_dyn | pic32mz_ef_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit configured for Interrupt mode and dynamic operation. |

## Configuring the Hardware

Describes how to configure the supported hardware.

### Description

PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit

No hardware related configuration or jumper setting changes are necessary.

Explorer 16 Development Board

- Switch S2 should be set to PIM

USB PICtail Plus Daughter Board

- Jumper the Host Enable pins
- Device Enable and OTG Enable should be open

PIC32MX795F512L CAN-USB PIM

- Keep jumper J10 open
- Keep all jumpers in J9 open
- Jumper J1 should be shorted between positions 1 and 2. This configuration is only applicable for the PIC32MX795F512L USB CAN PIM (MA320003), and not the PIC32MX795F512L USB PIM (MA320002).
- Jumper J2 should be shorted between positions 1 and 2. This configuration is only applicable for the PIC32MX795F512L USB CAN PIM (MA320003) and not the PIC32MX795F512L USB PIM (MA320002).

**For the pic32mx795_pim_e16_int_dyn configuration:**

1. Ensure that the PIC32MX795F512L PIM is connected properly to the PIM socket on the Explorer 16 Development Board.
2. Connect the Serial Port connector on the Explorer 16 Development Board to a PC using a Serial-to-USB converter cable.
3. Connect the USB PICtail Plus Daughter Board to the horizontal edge connector (J9) of the Explorer 16 Development Board.

**For the pic32mz_ef_sk_int_dyn configuration:**

Connect the USB to the UART connector (J11) on the PIC32MZ EF Starter Kit to a PC using a USB micro cable.

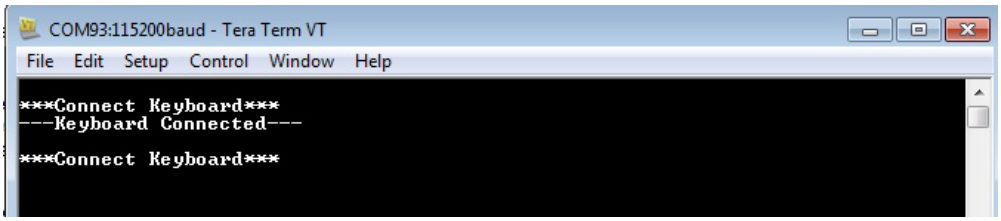
## Running the Demonstration

Provides instructions on how to build and run the USB HID Basic Keyboard demonstration.

### Description

1. Open the project in MPLAB X IDE and select the project configuration.
2. Build the code and program the device.
3. Launch a terminal emulator, such as Tera Term, and select the appropriate COM port and set the serial port settings to 115200-N-1.
4. If a USB keyboard is not connected to the PIC32 USB Host, the terminal emulator window will show the *Connect Keyboard* prompt.
5. Attach a USB keyboard to the Host connector of the target hardware. The message, *Keyboard Connected*, will appear in the terminal emulator window.

6. Begin typing on the keyboard and the appropriate keys should be displayed on the serial terminal. Subsequent press and release of modifier keys (i.e., CAPS LOCK, NUM LOCK, etc.) will result in the appropriate keyboard LEDs to turning ON and OFF.
7. Disconnecting the keyboard will result in the message, *Connect Keyboard*.




## *hid_basic_mouse_usart*

This topic demonstrates USB Host support for a USB HID Mouse.

### Description

This application demonstrates the use of the USB HID Host Client Driver to enumerate and operate a HID mouse device. The application uses the USB Host layer, HID Client driver and HID Mouse Usage driver to enumerate USB mouse and decode mouse-generated data.

Mouse-specific movements events are demonstrated by displaying relative coordinate changes using a serial terminal emulator on a personal computer. Mouse button clicks are indicated by LEDs.

## Building the Application

This section does the following:

- Identifies the MPLAB X IDE project name and location.
- Lists and describes the available configurations for the USB HID Basic Mouse USART demonstration.

### Description

To build this project, you must open the `hid_basic_mouse_usart.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/host/hid_basic_mouse_usart`.

**MPLAB X IDE Project**

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `hid_basic_mouse_usart.X` | `<install-dir>/apps/usb/host/hid_basic_mouse_usart/firmware` |

**MPLAB X IDE Project Configurations**

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mz_ef_sk_meb2 | pic32mz_ef_sk+meb2 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit attached to Multimedia Expansion Board II (MEB II) board. |

## Configuring the Hardware

This section describes how to configure the supported hardware.

### Description

1. Ensure that the PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit is securely fastened into the MEB II expansion board.
2. Connect the USB to the UART connector (J11) on the PIC32MZ EF Starter Kit to a PC using a USB micro cable.

**Note:** No hardware related configuration or jumper setting changes are necessary.

## Running the Demonstration

This section provides instructions about how to build and run the USB HID Mouse USART demonstration.

### Description

1. Open the project in MPLAB X IDE and select the project configuration.
2. Build the code and program the device.
3. Launch a terminal emulator, such as Tera Term. Select the appropriate COM port and set the serial port settings to 115200-N-1.
   - If a USB mouse is not connected to the Host connector by using J5 on the PIC32 MZ EF Starter Kit, the serial terminal emulator window will show the "Connect Mouse" prompt.
4. Attach a USB mouse to the Host connector of the target hardware. The message, "Mouse Connected", will display in the serial terminal emulator window.
5. Begin moving the mouse and the appropriate relative coordinate changes for X,Y, and Z axes should be displayed in the serial terminal window.
6. Click the mouse button to toggle LEDs on the MEB II board as shown in the following table.

| Mouse Click | MEB II LED |
|---|---|
| Left | D3 |
| Right | D4 |
| Middle | D5 |
| Lower Left | D6 |
| Lower Right | D7 |

- Disconnecting the mouse will result in the message, "Connect Mouse", to reappear on the serial console.


## *hub_cdc_hid*

Demonstrates the enumeration of a HID mouse and CDC emulator device via an external hub.

### Description

This application demonstrates the capability of the USB Host Stack to access and manage multiple USB Devices through a Hub. The demonstration application enumerates a HID mouse and CDC emulator device via an external hub. The host will demonstrate the communication from the CDC emulator device and the HID mouse.

## Building the Application

This topic identifies the MPLAB X IDE project name and location and lists and describes the available configurations for the USB Host HUB CDC HID Demonstration.

### Description

To build this project, you must open the `hub_cdc_hid.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/host/hub_cdc_hid`.

**MPLAB X IDE Project**

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `hub_cdc_hid.X` | `<install-dir>/apps/usb/host/hub_cdc_hid/firmware` |

**MPLAB X IDE Project Configurations**

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mx_usb_sk2_int_dyn | pic32mx_usb_sk2 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit II configured for Interrupt mode and dynamic operation. |
| pic32mz_ef_sk_int_dyn | pic32mz_ef_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ EF Starter Kit configured for Interrupt mode and dynamic operation. |

## Configuring the Hardware

Describes how to configure the supported hardware.

### Description

PIC32 USB Starter Kit II

JP2 should be in place.

PIC32MZ EF Starter Kit

No hardware related configuration or jumper setting changes are necessary.

## Running the Demonstration

Provides instructions on how to build and run the USB Host HUB CDC HID demonstration.

### Description

This application demonstrates the capability of the USB Host Stack to access and manage multiple USB Devices through a Hub. The demonstration application enumerates a HID mouse and CDC emulator device via an external hub. The host will demonstrate the communication from the CDC emulator device and the HID mouse.

1. Open the project in MPLAB X IDE and select the desired project configuration.
2. Build the code and program the device.
3. Connect a hub to the Type A Host connector on the desired board.
4. Connect a mouse to a spare port on the hub.
5. Connect the CDC emulator device to another spare port on the hub.
6. Click the mouse to toggle LEDs on the starter kit.
7. On the personal computer, open a terminal emulator. At the prompt, (LED:), enter 1, 2, or 3 to toggle the LEDs on the starter kit.

## *hub_msd*

This application demonstrates the capability of the USB Host stack to support multiple MSD device through a hub.

### Description

This application demonstrates the use of the Hub Driver and the MSD Host Client Driver, with File System, to support multiple MSD devices and Hub. The demonstration application copies a file from one pen driver into another pen drive.

## Building the Application

This section identifies the MPLAB X IDE project name and location and lists and describes the available configurations for the USB Host Hub MSD Demonstration.

### Description

To build this project, you must open the `hub_msd.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/host/hub_msd`.

**MPLAB X IDE Project**

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `hub_msd.X` | `<install-dir>/apps/usb/host/hub_msd/firmware` |

**MPLAB X IDE Project Configurations**

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mx_usb_sk2_int_dyn | pic32mx_usb_sk2 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit II configured for Interrupt mode and dynamic operation. |
| pic32mz_ef_sk_int_dyn | pic32mz_ef_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit configured for Interrupt mode and dynamic operation. |

## Configuring the Hardware

Describes how to configure the supported hardware.

### Description

PIC32 USB Starter Kit II

JP2 should be in place.

PIC32MZ EF Starter Kit

No hardware related configuration or jumper setting changes are necessary.

## Running the Demonstration

Provides instructions on how to build and run the USB Host Hub MSD demonstration.

### Description

This application demonstrates the capability of the USB Host Stack to access and manage multiple USB Devices through a Hub. The demonstration application copies a file from one USB pen drive (i.e., a USB Flash storage device) to another USB pen drive, where these pen drives are attached to a hub.

**Note:** The demonstration will search for a file named `file.txt` on any of the connected pen drives. Such a file should be created on one of the pen drives through any suitable method.

1. Open the project in MPLAB X IDE and select the desired project configuration.
2. Build the code and program the device.
3. Connect a hub to the Type A Host connector on the desired board.
4. Connect a USB Pen drive containing an arbitrary file named `file.txt` to a spare port on the hub.
5. Connect another USB pen drive to another spare port on the hub.
6. The application will copy the file `file.txt` from the drive containing this file to the other drive. The copied file will be renamed as `newfile.txt`. LED 2 on the demonstration board will illuminate to indicate completion of the file transfer.
7. Disconnect the drives and confirm demonstration success by inserting them into a personal computer and verifying the file transfer completed as expected.

The demonstration application will always be in state where it waits for two pen drives to be connected to the hub and at least one of these pen drives contains a file named `file.txt`.

## *msd_basic*

This application demonstrates the use of the MSD Host Class Driver to write a file to USB Flash Drive.

### Description

This application demonstrates the use of the MSD Host Class Driver to write a file to a USB Flash drive. The application uses the USB Host layer , MSD class driver and the MPLAB Harmony File System Framework to enumerate a USB Flash drive and to write a file to it.

## Building the Application

This section identifies the MPLAB X IDE project name and location, and lists and describes the available configurations for the USB MSD Host Class Driver Demonstration.

## Description

To build this project, you must open the `msd_basic.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/host/msd_basic`.

## MPLAB X IDE Project

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `msd_basic.X` | `<install-dir>/apps/usb/host/msd_basic/firmware` |

## MPLAB X IDE Project Configurations

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| chipkit_wf32 | chipkit_wf32 | Demonstration running on the chipKIT WF32 Development Board. |
| chipkit_wifire | chipkit_wifire | Demonstration running on the chipKIT Wi-FIRE Development Board. |
| pic32mx_usb_sk2_int_dyn | pic32mx_usb_sk2 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit II configured for Interrupt mode and dynamic operation. |
| pic32mx_usb_sk3_int_dyn | pic32mx_usb_sk3 | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32 USB Starter Kit III with the PIC32MX470F512L microcontroller configured for Interrupt mode and dynamic operation. |
| pic32mz_da_sk_intddr_int_dyn | pic32mz_da_sk_intddr | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ Embedded Graphics with Internal DRAM (DA) Starter Kit configured for Interrupt mode and dynamic operation. |
| pic32mz_ef_sk_int_dyn | pic32mz_ef_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit configured for Interrupt mode and dynamic operation. |
| pic32mx_xlp_sk_int_dyn | pic32mx_xlp_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MX XLP Starter Kit configured for Interrupt mode and dynamic operation. |
| pic32wk_sk_int_dyn | pic32wk_gbp_gpd_sk+module | Select this MPLAB X IDE project configuration to run the demonstration application to run on the PIC32WK Wi-Fi Starter Kit, with the WM32 Wi-Fi module. The USB Stack will be configured for Interrupt mode operation and the USB Driver will be configured for Dynamic operation mode. |
| pic32mx470_curiosity | pic32mx470_curiosity | Select this MPLAB X IDE project configuration to run the demonstration application to run on the PIC32MX470 Curiosity Development Board, with the PIC32MX470F512H microcontroller. The USB Stack will be configured for Interrupt mode operation and the USB Driver will be configured for Dynamic operation mode. |
| pic32mz_ef_curiosity | pic32mz_ef_curiosity | Select this MPLAB X IDE project configuration to run the demonstration application to run on the PIC32MZ EF Curiosity Development Board, with the PIC32MZ2048EFM100 microcontroller. The USB Stack will be configured for Interrupt mode operation and the USB Driver will be configured for Dynamic operation mode. |

### Configuring the Hardware

Describes how to configure the supported hardware.

## Description

PIC32 USB Starter Kit II

JP2 should be in place.

PIC32 USB Starter Kit III

JP1 should be in place.

PIC32MZ Embedded Graphics with Internal DRAM (DA) Starter Kit

No hardware related configuration or jumper setting changes are necessary.

PIC32MZ EC Starter Kit

JP1 should be in place and the Ethernet plug-in board should be removed.

PIC32MZ EF Starter Kit

No hardware related configuration or jumper setting changes are necessary.

chipKIT WF32 Wi-Fi Development Board

No hardware related configuration or jumper setting changes are necessary.

chipKIT Wi-FIRE Development Board

No hardware related configuration or jumper setting changes are necessary.

PIC32MX470 Curiosity Development Board

- Ensure that a jumper is placed at 4-3 on J8, to select supply from debug USB connector.
- Power the PIC32MX470 Curiosity Development Board from a Host PC through a Type-A male to mini-B USB cable connected to Mini-B port (J3)
- Place a jumper on J13 to drive VBUS in Host mode
- Plug in a USB peripheral with a micro-A USB connector, or use a micro USB OTG to USB adapter.

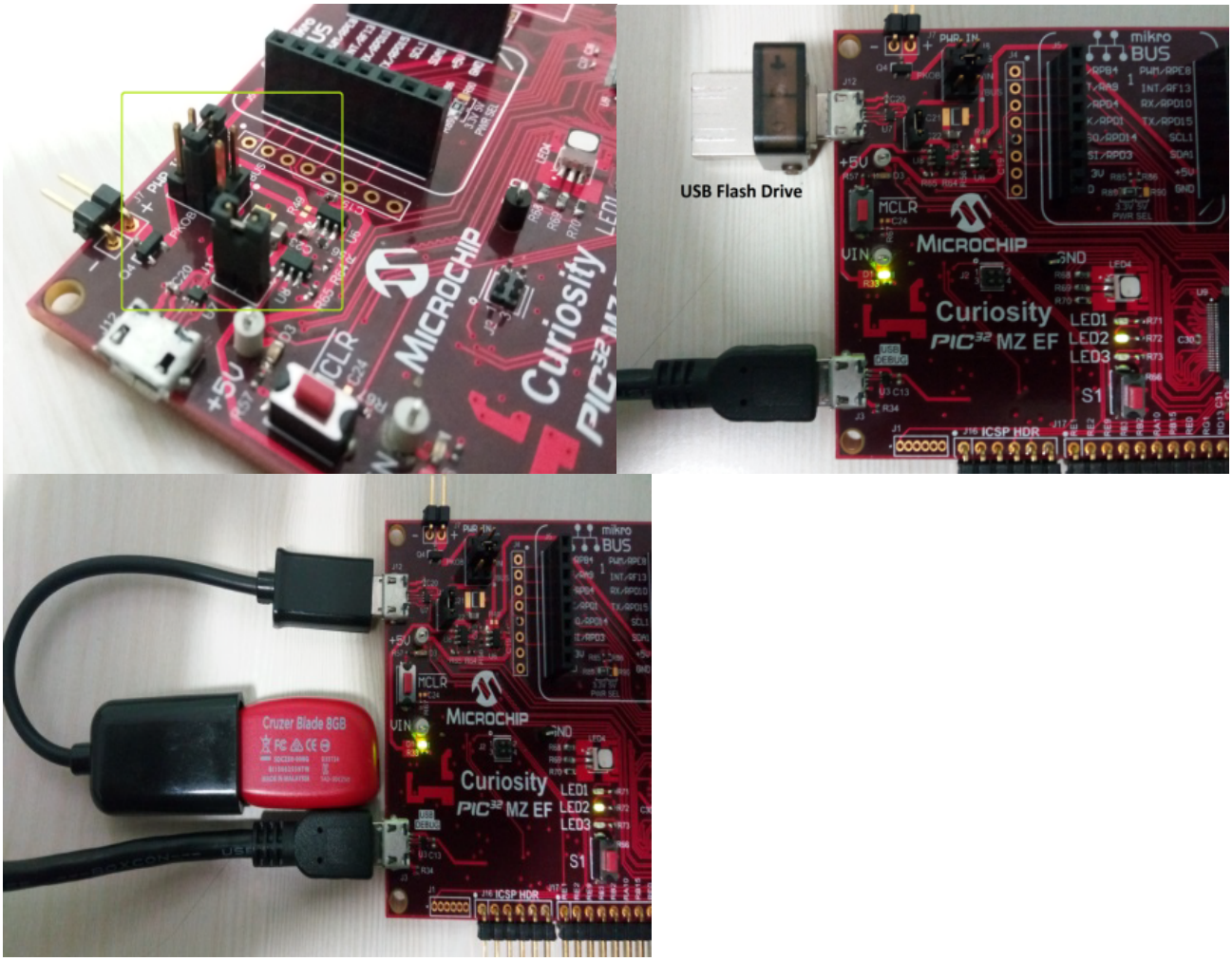


PIC32MZ EF Curiosity Development Board

- Ensure that a jumper is placed at 4-3 on J8, to select supply from debug USB connector.
- Power the PIC32MZ EF Curiosity Development Board from a Host PC through a Type-A male to micro-B USB cable connected to Micro-B port

(J3).

- Place a jumper on J13 to drive VBUS in Host mode.
- Plug in a USB peripheral with a micro-A USB connector, or use a micro USB OTG to USB adapter.




PIC32WK Wi-Fi Starter Kit

No hardware related configuration or jumper setting changes are necessary.

## Running the Demonstration

Provides instructions on how to build and run the USB Host MSD Basic demonstration.

## Description

This application demonstrates the use of the MSD Host Class Driver to write a file to USB Flash drive. The application uses the USB Host layer, MSD class driver and the MPLAB Harmony File System Framework to enumerate a USB Flash drive and to write a file to it.

Prior to using this demonstration, it is recommended to review the MPLAB Harmony Release Notes for any known issues. A PDF copy of the release notes is provided in the `<install-dir>/doc` folder of your installation.

1. Open the project in MPLAB X IDE and select the desired project configuration.
2. Build the code and program the device.
3. With the code running, attach a USB Flash drive to the Host connector on the desired starter kit.
4. The demonstration application will then create a file named `file.txt`. It will then write the text "Hello World" to this file, and then close the file.
5. The demonstration will then move to Idle mode, which is indicated when the LED on the starter kit illuminates. On PIC32M-based starter kits the LED is LED2.
6. The USB Flash drive can then be attached to a USB Host personal computer to verify the demonstration application operation.
7. Steps 3 through 6 can be repeated.
8. If the USB Flash drive already contains a file with the name `file.txt`, the demonstration application will append the text "Hello World" to the end of the file contents.
9. The LED on the starter kit illuminates if the file creation or write failed. On PIC32M-based starter kits, the LED is LED1.

## Multiple USB Controller

This section describes the demonstrations that make use of multiple USB controllers on certain PIC32 microcontrollers.

### *cdc_com_port_dual*

This application demonstrates dual USB Device operation on a PIC32 microcontroller with two USB Controllers.

#### Description

This application demonstrates dual USB Device operation on a PIC32 microcontroller with Two USB Controllers. In this demonstration both of the USB controllers act as CDC devices.

### Building the Application

This section identifies the MPLAB X IDE project name and location and lists and describes the available configurations for the Multiple USB CDC Device Dual COM Port Demonstration.

#### Description

To build this project, you must open the `cdc_com_port_dual.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/multi_usb/cdc_com_port_dual`.

#### MPLAB X IDE Project

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `cdc_com_port_dual.X` | `<install-dir>/apps/usb/multi_usb/cdc_com_port_dual/firmware` |

#### MPLAB X IDE Project Configurations

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mk_gp_db_int_dyn | pic32mk_gp_db | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MK Evaluation Kit configured for interrupt mode and dynamic operation. |

### Configuring the Hardware

Describes how to configure the supported hardware.

#### Description

PIC32MK General Purpose (GP) Development Board

Switch S4 should be set to the Device position.

### Running the Demonstration

This section provides instructions on how to build and run the USB Multiple Controller CDC Com Port Dual demonstration.

#### Description

This application demonstrates dual USB Device operation on a PIC32 microcontroller with two USB Controllers. The MPLAB Harmony USB Stack is capable of handling multiple USB controllers. In this demonstration, both of the USB controllers act as CDC devices.

This demonstration allows the each controller on the PIC32 to appear like a serial (COM) port to the host. Do the following to run this demonstration:

1. First compile and program the target device. Refer to Building the Application for details.
2. Attach both USB connectors J15 and J13 to the host.
3. Refer to the Running the Demonstration section of the USB Device cdc_com_port_single demonstration for details on exercising the CDC device features.

### *msd_dual*

This application demonstrates the capability of a PIC32 microcontroller and MPLAB Harmony USB Host stack to work with two USB Controllers in an application.

## Description

This application demonstrates the capability of a PIC32 microcontroller and the MPLAB Harmony USB Host stack to work with two USB Controllers in an application. The MPLAB Harmony USB Stack is capable of handling multiple USB controllers.

### Building the Application

This section identifies the MPLAB X IDE project name and location and lists and describes the available configurations for the Dual MSD demonstration.

## Description

To build this project, you must open the `msd_dual.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/multi_usb/dual_msd`.

## MPLAB X IDE Project

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `msd_dual.X` | `<install-dir>/apps/usb/multi_usb/msd_dual/firmware` |

## MPLAB X IDE Project Configurations

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mk_gp_deb_int_dyn | pic32mk_gp_db | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MK Evaluation Kit configured for interrupt mode and dynamic operation. |

### Configuring the Hardware

Describes how to configure the supported hardware.

## Description

PIC32MK General Purpose (GP) Development Board

- Switch S4 should be set to Host position
- Jumper J28 must be installed
- USB Connector J12 must be connected to a USB Host for powering the board
- USB Flash drive should be attached to Connector J15 and J14 after programming the microcontroller

### Running the Demonstration

This section provides instructions on how to build and run the USB Multiple Controller Dual MSD demonstration.

## Description

This application demonstrates the capability of a PIC32 microcontroller and the MPLAB Harmony USB Host stack to work with two USB Controllers in an application. The MPLAB Harmony USB Stack is capable of handling multiple USB controllers. The application uses the USB Host_layer, MSD class driver, and the MPLAB Harmony File System Framework to enumerate a USB Flash drive and to write a file to it.

Prior to using this demonstration, it is recommended to review the MPLAB Harmony Release Notes for any known issues. A PDF copy of the release notes is provided in the `<install-dir>/doc` folder of your installation. Do the following to run this demonstration:

1. Open the project in MPLAB X IDE and select the desired project configuration.
2. Build the code and program the device.
3. With the code running, attach a USB Flash drive with a file “file.txt” in it to one of the Host connector on the board.

4. Connect another USB Flash drive to other Host connector on the board. Ensure this flash drive does not contain any file named `newfile.txt`.
5. The application will copy the file file.txt from the drive containing this file to the other drive. The copied file will be renamed as `newfile.txt`. LED2 on the demonstration board will illuminate to indicate completion of the file transfer.
6. Disconnect the drives and confirm demonstration success by inserting them into a personal computer and verifying the file transfer completed as expected.

## Dual Role

This section describes the USB Dual Role Demonstrations. These demonstrations project demonstrate operation of the USB Host and the USB Device stack in the same project.

### *host_msd_device_hid*

This application demonstrates role switching between USB Host MSD Stack and USB Device HID function. The role switch is trigger by a switch press.

#### Description

This application demonstrates role switching between USB Host MSD Stack and USB Device HID function. The role switch is trigger by a switch press. In the USB Host mode, the application performs read and write operations to a USB pen drive. In the USB Device mode, the application emulates a HID mouse.

#### Building the Application

This section identifies the MPLAB X IDE project name and location and lists and describes the available configurations for the USB Host MSD and USB HID Mouse Device Dual Role application.

#### Description

To build this project, you must open the `host_msd_device_hid.X` project in MPLAB X IDE, and then select the desired configuration.

The following tables list and describe the project and supported configurations. The parent folder for these files is `<install-dir>/apps/usb/dual_role/host_msd_device_hid`.

#### MPLAB X IDE Project

This table lists the name and location of the MPLAB X IDE project folder for the demonstration.

| Project Name | Location |
|---|---|
| `host_msd_device_hid.X` | `<install-dir>/apps/usb/dual_role/host_msd_device_hid/firmware` |

#### MPLAB X IDE Project Configurations

This table lists and describes the supported configurations of the demonstration, which are located within `./firmware/src/system_config`.

| Project Configuration Name | BSP Used | Description |
|---|---|---|
| pic32mz_ef_sk_int_dyn | pic32mz_ef_sk | Select this MPLAB X IDE project configuration to run the demonstration on the PIC32MZ Embedded Connectivity with Floating Point Unit (EF) Starter Kit configured for Interrupt mode and dynamic operation. |

#### Configuring the Hardware

Describes how to configure the supported hardware.

#### Description

PIC32MZ EF Starter Kit

No hardware related configuration or jumper setting changes are necessary.

#### Running the Demonstration

Provides instructions no how to build and run the USB Host MSD and USB HID Mouse Device Dual Role application.

## Description

This application demonstrates the Dual Role capability of the MPLAB Harmony USB Stack. The application project includes both, the USB Host and Device Stacks. Both the stacks are initialized during application initialization. During operation, the application polls the switch SW2 on the starter kit to trigger a USB role switch. Note that the application cannot simultaneously operate as a host and device. The one USB role is exclusive of the other.

Prior to using this demonstration, it is recommended to review the MPLAB Harmony Release Notes for any known issues. A PDF copy of the release notes is provided in the `<install-dir>/doc` folder of your installation.

1. Open the project and in MPLAB X IDE and select the desired project operation
2. Build the code and program the device. The application initially will not operate in any USB role.
3. Press SW2 on the starter kit. This places the application in a USB Device mode.
4. Connect a USB cable between micro USB connector (J4) on the starter kit and a PC USB host. The application will emulate a USB HID mouse function. The cursor on the PC will rotate. Pressing SW1 will enable and disable the cursor movements. Exercise device plug-n-play operation to confirm USB Device operation
5. Now try switching the USB role. Disconnect the USB cable between micro USB connector (J4) on the starter kit and a PC USB host. Press SW2 on the starter kit.
6. The application now will be in USB Host role. Connect a USB pen drive to the Type-A USB Host connector (J5) on the starter kit. The application will create a file (`file.txt`) on the pen drive. The completion of the operation is indicated by LED2 on the starter kit. Disconnect the pen driver and connect it to a PC to verify the contents of the file.
7. Repeat steps 3 through 6 to exercise the role switching capability.

# Index