Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.



THIS SPEC IS OBSOLETE

Spec No: 001-614665

Spec Title: WORKING WITH INF FILE OF A DEVICE USING CYUSB.SYS - AN61465

Sunset Owner: Nikhil Naik (nikl)

Replaced by: NONE

Working With inf File of a Device Using CyUSB.sys

AN61465

Author: Anand Srinivasan
Associated Project: Yes
Associated Part Family: EZ-USB (AN21xx/FX/FX1/FX2/FX2LP)
Associated Application Notes: None

OBSOLETE

Application Note Abstract

Microsoft Windows uses inf file to bind a device to its appropriate driver. This application note describes the sections and the guidelines for modifying an inf file of a USB device.

Introduction

Microsoft windows uses inf file to bind a device to its appropriate driver. If a window has a native driver for that USB device class then an inf file is not required and the device is bound to the native driver. However if we want the device to bind to a custom driver we need a custom inf file. This application note describes the sections of this inf file and guidelines to modify them based on the requirement.

Overview

The inf file can be divided into sections. Each section has its own purpose and modifying them for a particular end use requires a proper understanding of purpose of the section. The modifications are discussed based on three inf files that comes with this document. These files were developed just for demonstrating the modifications discussed in this application note. These inf files cannot be used “as is” and will need proper modification to have an end use.

A word of caution, though the modifications are simple, proper understanding of the different sections is required to properly modify an inf file. Modification of inf file are recommended only for advanced users. Since the modification can be complex based on the OS and/or hardware platform supported by the inf file.

Terminology

The following section describes the terminologies/notations that are used in this application note and the assumptions for them if any.

1. The demonstration inf files are based on the usage case of *CyUSB.sys* being the driver and the driver files being in the same folder. Only the essential parts of the inf file for this use case are being discussed in this application note.
2. The term Operating System (OS) refers to any version of Microsoft Windows.
3. *XXXX* refers to Vendor ID (VID) and *YYYY* refers to product ID (PID).
4. The sections, entries and their values of an inf file are defined by Microsoft and are not Cypress-specific unless specified so.
5. Section in an inf file holds entries of similar purpose together.
6. Entry – The inf file uses entry to hold the information for the operating system. Values are assigned to these entries through the statement: *Entry* = *Value*. The type of these values varies from entry to entry.
7. HKLM refers to HKEY_LOCAL_MACHINE.
8. %**string**% refers.- This notation is used to represent a string. The value for the **string** is usually defined in the Strings section of the inf file. Various names based on the context will be used for **string** throughout this application note. This notation is used because it would be easier to edit the strings because as they are grouped together. Also in cases where the string is used more than once, editing the string becomes easier. Instead of this notation, the string can be used directly.
9. Decoration – Decoration is used to specify the OS and/or hardware platform for which a particular section/entry of the inf file is intended for. This is of the form *NT[Architecture].[OSMajorVersion].[OSMinorVersion]* and follows the section name with a dot in between. The different parts that comprise decoration are optional in some OS/hardware platform and mandatory in others. Table 1. Details of Parts of Decorationdescribes each of these parts. Table 2 describes the values that *[OSMajorVersion]* and *[OSMinorVersion]* can take. The decoration that matches the OS closely is used by the OS to load the driver. For example, if an inf file with NT, NTx86 and NTx86.5.1 is used in an XP 32 system, then the OS will use the sections with NTx86.5.1 decoration. If only NTx86 and NT are used, then the OS will go with NTx86. Starting with Windows 2003 SP1, *[Architecture]* is mandatory in 64-bit OS. The decoration is useful in cases where the inf file has to handle driver loading for more than one OS and/or hardware platform. Here decoration points out which particular section/entry is to be used in OS.

Table 1. Details of Parts of Decoration

Part	Description
NT	Identifies the target OS to be NT-based. Windows 2000 and OS later than that are all NT-based.
[Architecture]	Identifies the hardware platform (CPU architecture). Following are the values ▪ x86 is for 32-bit processor architecture. ▪ ia64 is for Intel itanium architecture processors. ▪ amd64 is for 64-bit processors with architecture other than intel itanium architecture (IEM64T and AMD 64bit architecture).
[OSMajorVersion]	A number that represents the OS's major version number.
[OSMinorVersion]	A number that represents the OS's minor version number.

Table 2. Values of *[OSMajorVersion]* and *[OSMinorVersion]* Based on OS

Windows Version	*[OSMajorVersion]*	*[OSMinorVersion]*
Windows 7	6	1
Windows vista	6	0
Windows XP	5	1
Windows 2000	5	0

OBSOLETE

Details

The sections of an inf file are listed in the order in which they appear in an inf file. User-defined sections are described as sub-sections and their position in the inf file may vary.

1. Version
2. SourceDisksNames
3. SourceDisksFiles
4. DestinationDirs
5. Manufacturer
6. Models
7. DDInstall
8. DDInstall.HW
9. DDInstall.Services
10. Strings

There are differences in the name used for certain sections/entries (which can have a custom name) between this application note and the demonstration inf file. In such cases, the name used in the inf file follows the section heading. Following is a detailed description of each of these sections.

1. Version

Version section contains entries which store the system files version details and details of the OS for which the inf file is intended for. Following are the essential entries of this section

1.1 *Signature*

Signature is used to specify the OS for which the inf file is intended for. Its value must be *"$WINDOWS NT$"* or *"$Chicago$"* for the inf file to be considered valid. The string inside the $ is case-insensitive.

1.2 *Class*

Specifies the class of the device. In this case, it takes the value of *USB.*

1.3 *ClassGUID*

GUID stands for Global Unique Identifier. ClassGUID holds the class-specific GUID. Here it is of the form {*nnnnnnnn-nnnn-nnnn-nnnn-nnnnnnnnnnnn*} where n is a hexadecimal number. Here since the class is USB, the ClassGUID takes the value {36FC9E60-C465-11CF-8056-444553540000}.

1.4 *Provider*

Specifies the provider of the inf file. It holds the value of %Provider_name% in this case.

1.5 *CatalogFile*

Specifies the catalog file (.cat) that is obtained when the driver is signed. Here it takes the value *Provider.cat*. Here Provider stands for the name for your *.cat* file. The system assumes that the catalog file is in the same folder as the inf file. This entry can use decoration.

1.6 *DriverVer*

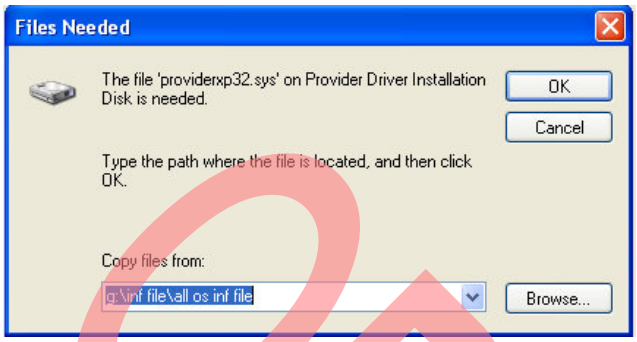
Specifies the version information of the drivers. Its value is of the form mm/dd/yyyy[,w.x.y.z]. Here mm is the month, dd the date and yyyy the year of the latest files in the driver package. [.w.x.y.z] is an optional version number for the driver package.

2. SourceDisksNames

This section identifies the medium that contains the driver files. The entries are numbers that do not exceed 4 bytes. It takes a string value that indicates the medium with the driver files to the user during installation. In this case, the value is %Installation_medium%. If the OS is not able to locate the driver files, it will ask for the location of the driver files. %Installation_medium% can be used in these cases to provide an indication to the end user as to where the file might be located. Figure 1. Files Needed Window

shows how this works in Windows XP 32-bit system. Not all OSs use this. This section can use decoration.

Figure 1. Files Needed Window



3. SourceDisksFiles

This section identifies the files that are copied to the system through the inf file. Here the entries are the names of the files to be copied and take the entries of SourceDisksNames section (numbers) as values. The OS uses to select the appropriate string to indicate the medium in which the particular file is present. This section can use decoration. Unlike other sections/entries, this section does not use NT in its decoration.

4. DestinationDirs

This section specifies the destination to which the driver files are to be copied. Here the entry is a user defined section name *filelistsectionname*. This user defined section contains the driver files to be copied. The value is of the *directoryid,subdirectory. directoryid* has system-defined values from -1 to 32767. 10 is used to refer to %systemroot% environment variable *(C:\WINNT* in *Windows 2000* and *C:\Windows in Windows XP* and later). Here *subdirectory* is assigned the value System32/Drivers. So in the inf files 10, System32\Drivers is used to copy the driver files to drivers system folder (*C:\WINNT\System32\Drivers* in Windows 2000 and *C:\Windows\System32\Drivers* in the case of Windows XP and later).

5. Manufacturer

This section identifies the manufacturer of the devices that use this inf file. The entry is a string that identifies the manufacturer. Here it is %Provider_name%. The value is of the form *modelname,decorations*. *modelname* is the user defined name of the models section. *decorations* comprises of decorations separated by comma (,). These decorations specify the OS and/or hardware platform for which the inf file is intended for.

6. Models (Provider_Device)

This section identifies the devices (VID/PID) that are supported by the inf file. It takes the *modelname* specified in Manufacturer section as its name. Here the entry is a string that describes the device.

Here it is *%VID_XXXX&PID_YYYY_device_description%.* This format is used for easier differentiation when the string is defined in Strings section. The value is of the form *installsectionname,hardwareid.* Here *installsectionname* is the name of the DDInstall section. The *hardwareid* is of the form *enumerator/enumeratorid*. Here *enumerator* is the device type and in this case it is USB. Here *enumeratorid* is the ID of the device by which the device is identified by the OS. Here it is *VID_XXXX&PID_YYYY.* The number of entries in this section is based on the number of VID/PID supported by the inf file. This section can use decoration.

Note The inf files that come with this application note have *XXXX* in places where VID is required and *YYYY* in places where PID is required. This has to be modified with your VID/PID for testing the working of the inf files.

7. DDInstall (provider)

This section contains instructions for installing the driver files and device-specific software information that are to be written to the registry.

It takes the name specified for *installsectionname* in the Models section as its name. This section can use decoration. Following are the essential entries of this section:

7.1 *CopyFiles*

CopyFiles specify the files that are to be copied to the destination directory. The value contains the files that are to be copied. Here the value is a user-defined section name *filelistsectionname.*

7.1.1 filelistsectionname (provider.Files.Ext)

This section specifies the files to be copied to the location specified in DestinationDirs. It takes the name specified for *filelistsectionname* in CopyFiles entry of DDInstall section as its name. Here the entries are the files.

7.2 *AddReg*

AddReg specifies the device-specific information that is to be written to the registry. Here the value is a user-defined section name used to write the device's software keys to the registry. Here it is *softwareaddreg.*

7.2.1 softwareaddreg (provider.AddReg)

This section adds the device-specific software keys to the registry. It takes the name specified for *softwareaddreg* in AddReg entry of DDInstall section as its name. Here the entry is of the form *registryrootname,subkey,valueentryname,flags,value.* Here *registryrootname* is the path in the registry to be written to. Here **HKR** is used for *registryrootname* to relatively specify the registry location. Because it is used in DDInstall section HKR refers to HKLM\SYSTEM\CurrentControlSet\Control\Class*ClassGuid\instanceid*. *ClassGuid* is the class GUID of the device. *instanceid* is assigned in ascending order and starts from 0. Each time a unique device is connected an *instanceid* is assigned to it. Hence this number will vary from system to system. *subkey* is the registry path/name of a new registry key. *Valueentryname* is the name of the registry key to be created. *flags* is a system-defined bit mask to represent the type of value/add registry operation. *value* is the value assigned to the *valueentryname*. The essential entries of this section, and they are:

OBSOLETE

7.2.1.1 HKR,,DevLoader,,*ntkern

This specifies to the host that the ntkern method is not used by the driver. *ntkern* is a method used in older versions of windows.

7.2.1.2 HKR,,NTMPDriver,,provider.sys

This specifies to the host the drivers that are to be loaded at the same time for the device. In this case *provider.sys* is the name of the driver.

7.2.1.3 HKR,"Parameters",valueentryname,flags,value

This specifies parameters that can be used to specify how the driver and or or application interact. Here *valueentryname*,*flags* and *value* are user-defined and can be extracted by the application to decide parameters (if any) of how the driver and application interact.

8. DDInstall.HW (provider.HW)

This section contains instruction for installing device-specific hardware information that is to be written to the registry.

It takes *installsectionname* specified in Models section instead of DDInstall in its name. This section can use decoration and if used, uses it between DDInstall and HW. The essential entry of this section is:

8.1 AddReg

AddReg specifies the device-specific information that is to be written to the registry. Here the value is a user-defined section name used to write the device's hardware keys to the registry. Here it is *hardwareaddreg*.

8.1.1 hardwareaddreg (provider.AddReg.Guid)

This section adds the device-specific hardware keys to the registry. It takes the name specified for *hardwareaddreg* in AddReg entry of DDInstall.HW section as its name. Here the entries take the same form as the entries in *softwareaddreg* section. Here **HKR** refers to HKLM\SYSTEM\CurrentControlSet\Enum\USB\Vid_*XXXX* &Pid_*YYYY*\PortID\Device Parameters. Here PortID is a value used to depict the USB port in the system. This varies from system to system.

The essential entries of this section are:

8.1.1.1 HKR,,DriverGUID,,%Provider_GUID%

This entry is a Cypress-specific entry. This entry is used to specify the GUID, used to access the driver and the devices connected to it. Here %Provider_GUID% is the GUID used.

8.1.1.2 HKR,,DriverEXECSCRIPT,,\systemroot\system32\Drivers\%Provider_sptfile%

This entry is a Cypress-specific entry. If the device uses a script file, then this entry is used to specify the script file to be used. %Provider_scriptfile% points to the script file.

9. DDInstall.Services (provider.Services)

This section contains instructions for installing device-specific services. It takes *installsectionname* specified in the Models section instead of DDInstall in its name. This section can use decoration. If used, it is used between DDInstall and HW. The essential entry of this section is:

9.1 AddService

AddService is used to specify the device-specific characteristics of the services associated with the driver. Here the value is of the form

servicename,flags,serviceinstallsectionname
Here *servicename* is user-defined name of the service associated with the device. *flags* are used to specify characteristics of the service. Here 2 are used for *flags* to assign this service name as the PnP function driver. *serviceinstallsectionname* is a user-defined section name. Add Service section installs the required services for the device.

9.1.1 serviceinstallsectionname (provider.Addservice)

This section adds the device-specific characteristics of the services associated with the driver. It takes the name *serviceinstallsectionname* specified in DDInstall.Services section as its name. Here the essential entries are:

9.1.1.1 DisplayName

This entry is used to specify a friendly name for the service. Here it takes the value %Providerservice_name%.

9.1.1.2 ServiceType

This entry is used to specify the service type. *CyUSB.sys* is a kernel-mode device driver. Hence it takes the value 0x00000001 (SERVICE_KERNEL_DRIVER).

9.1.1.3 StartType

This entry is used to specify when to start the driver. Here we want *CyUSB.sys* to be started when a device that uses *CyUSB.sys* is enumeration. So we set the value to 0x3 (SERVICE_DEMAND_START).

9.1.1.4 ErrorControl

This entry is used to specify the error control needed. Here if *CyUSB.sys* does not load properly during system startup then the system has to display an error message and proceed. So we set the value to 0x1 (SERVICE_ERROR_NORMAL). To

9.1.1.5 ServiceBinary

This entry is used to specify the path of the binary for the service. Here value is of the form *%directoryid%\filepath.* Here *directoryid* is the same as in DestinationDirs section. *filepath* is *subdirectoryid\drivername. subdirectoryid* is the same as in DestinationDirs section and *drivername* is the driver being used. In this case it is *CyUSB.sys*.

9.1.1.6 AddReg

This entry is used to specify the required registry keys. Here it takes the value *softwareaddreg.*

9.1.1.7 LoadOrderGroup

This entry is used to specify the load order group: of which the driver is a member. The system controls the driver loading through this. The value of this is Base for USB drivers. Hence it takes the value Base here.

OBSOLETE

10. Strings

This section specifies values to the strings used in the inf file. Here the entry is the **string** of %**string**%. The value is the corresponding string enclosed in double quotes (“ “).

Modification of demonstration inf file

This section describes the modifications required to make the demonstration inf file a proper working inf file. The modifications required are

1. Replace XXXX and YYYY in model section with the VID and PID of the device.
2. Remove the semi-colon (;) in front of *HKR,,DriverGUID,,%Provider_GUID%* entry of hardwareaddreg (here provider.AddReg.Guid) section. Replace the value of provider.GUID in strings section with your GUID. This modification is a recommended change but is optional.
3. If a script file is used then remove the semi-colon (;) in front of script file name (%Provider_sptfile% here) entry of SourceDisksFiles , filelistsectionname sections and also the one in front of *HKR,,DriverEXECSCRIPT,,%Provider_sptfile%* entry of hardwareaddreg (here provider.AddReg.Guid) section. Replace the value of Provider_sptfile in strings section with the script file name.

When going for production, it is recommended to rename the driver files and version details as per your requirement and make corresponding changes to the inf file.

Demonstration inf Files

This section describes the demonstration inf files that come with this document. Renamed *CyUSB.sys* version 3.4.1.20 is used for demonstration. Here the driver names in the folders are of the form provider*OSArchitecture*.sys. *OS* depicts the OS with which the driver is compatible and *Architecture* the CPU architecture with which the driver file is compatible. i.e. providerxp32 is renamed XP 32-bit version of *CyUSB.sys.* These inf files were developed using XP and Vista as the test platform. For other Windows versions these files can be used as reference to write an inf file.

Vista and XP.inf

This inf file takes care of loading the appropriate driver in Windows XP and Vista across hardware platforms. This way the end user does not have to write an inf for every OS and hardware platform combination. The *provider.sys* is same as *providerxp32.sys.* In Windows XP-32 bit machine, remove the value NTx86.5.1 from manufacturer section. Now when the modified inf file is used, it can be seen that *provider.sys* is used by the OS. Because it is the driver used by sections with NT decoration which is the closest match after NTx86.5.1.

32-bit.inf

This inf file can be used in 32-bit OS for loading driver. Two versions of this inf file are available in separate folder for Vista and XP respectively.

64-bit.inf

This inf file can be used in 64-bit OS for loading driver. Two versions of this inf file are available in separate folder for Vista and XP respectively.

Summary

This document explains the use of different sections of an inf file. It provides information on the modifications and their effects on the way the OS sees the inf file. It also demonstrates the use of different sections and the modifications through demonstration inf files.

About the Author

Name: Anand Srinivasan

Title: Applications engineer Sr

Contact: aasi@cypress.com

OBSOLETE

Document History

Document Title: Working With inf File of a Device Using CyUSB.sys - AN61465

Document Number: 001-61465

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	2936342	AASI	05/24/2010	New application note
*A	3089956	AASI	01/20/2011	Emphasized the decoration requirement in 64-bit OS.
*B	4011817	NIKL	05/27/2013	Obsolete document.

OBSOLETE

EZ-USB FX, EZ-USB FX2, and EZ-USB FX2LP are trademarks of Cypress Semiconductor Corporation. All product and company names mentioned in this document are the trademarks of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
http://www.cypress.com/

© Cypress Semiconductor Corporation, 2008-2013. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.