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THIS SPEC IS OBSOLETE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: 002-05440

Spec Title: AN205440 - F2MC-USB FAMILIES 16/32-BIT MICROCONTROLLER FUJITSU USB SERIES USING THE USB WIZARD / ASSISTANT

Replaced by: NONE

AN205440

F²MC-USB FAMILIES

16/32-BIT MICROCONTROLLER

FUJITSU USB SERIES

USING THE USB WIZARD / ASSISTANT

APPLICATION NOTE

OBSOLETE

Revision History

Document Title: AN205440 - F2MC-USB FAMILIES 16/32-BIT MICROCONTROLLER FUJITSU USB SERIES USING THE USB WIZARD / ASSISTANT

Document Number: 002-05440

Rev	ECN	Date	Issue
**	–	2010-09-24	V1.0, MSc, First version
–	–	2011-06-27	V1.1, MSc, Program Specific Update – New Features Description
–	–	2011-07-21	V1.2, MSc, USB Assistant V1.6 Update
–	–	2012-08-24	V1.3, MSc, USB Assistant V1.9, USB Wizard V2.0
*A	5392058	2016-08-05	Obsoleting the document.

This document contains 71 pages.

OBSOLETE

 © Cypress Semiconductor

Warranty and Disclaimer

© Cypress Semiconductor Corporation, 2016. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC (“Cypress”). This document, including any software or firmware included or referenced in this document (“Software”), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress’s patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage (“Unintended Uses”). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.

Contents

OBSOLETE

OBSOLETE

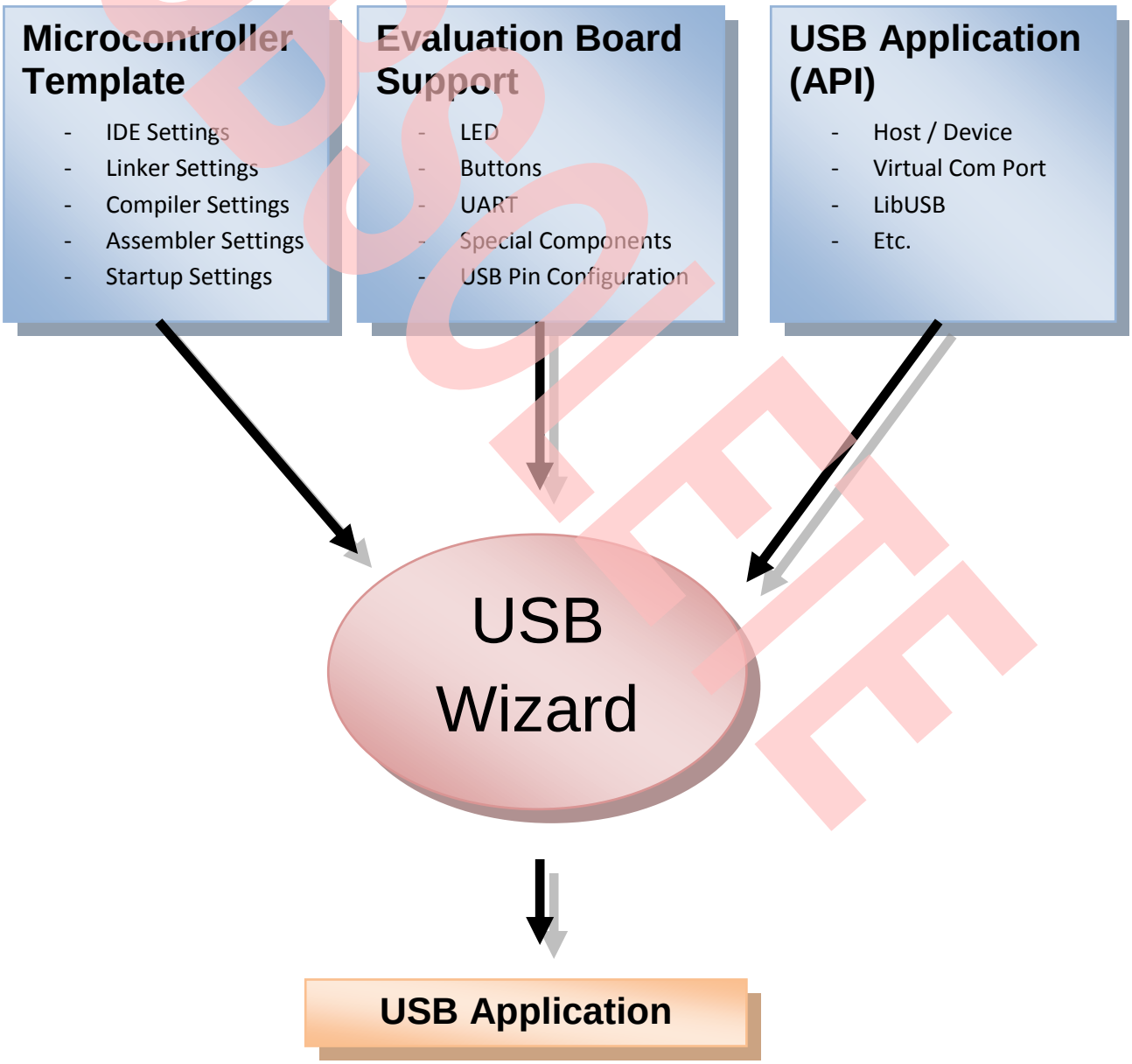
OBSOLETE

1 Introduction

INTRODUCTION

This Application Note gives an introduction in using the Fujitsu USB Wizard / Assistant. The USB Wizard / Assistant can be used to configure the USB functionality for the USB Device hardware abstraction layer used in different applications.

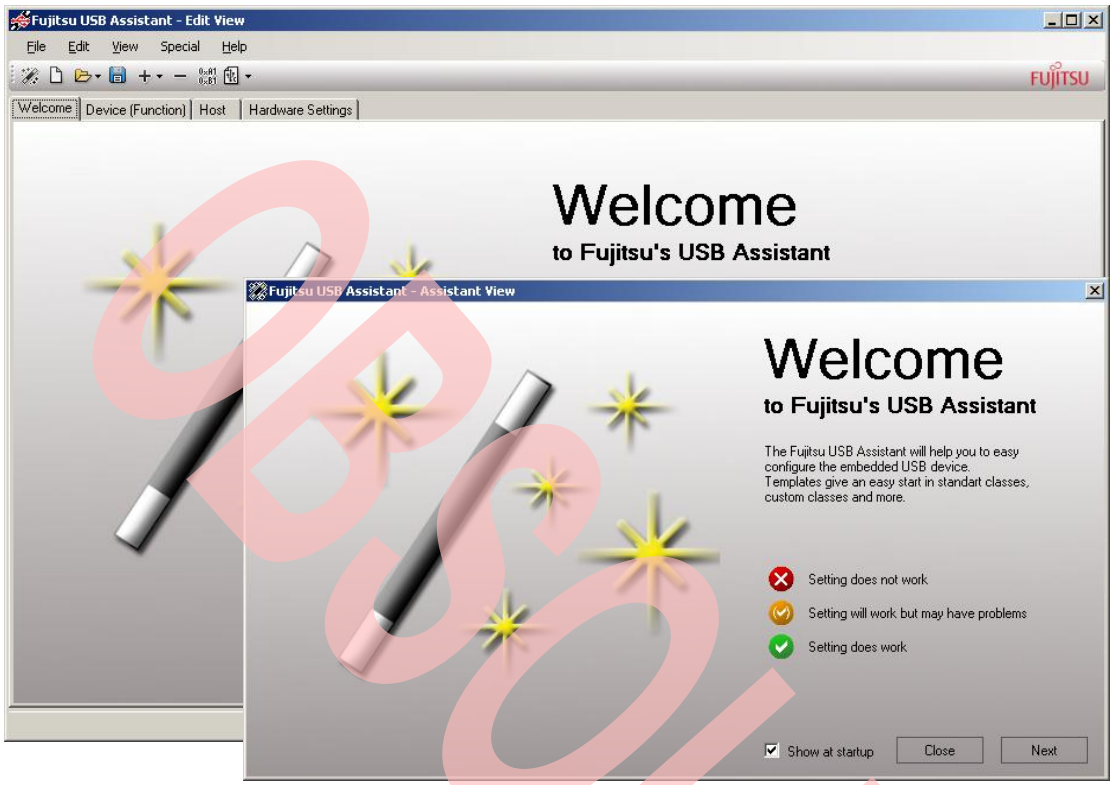
The Fujitsu USB Wizard / Assistant combines...



OBSOLETE

1.1 USB Assistant for USB stack < 2.0, supporting 16FX, FR80 and FM3

For automatically configuration of the USB stack, the Fujitsu USB Assistant can be used for 16FX, FR80 and FM3 with use of USB stack version < 2.0. **For further FM3 development please use the Fujitsu USB Wizard**.

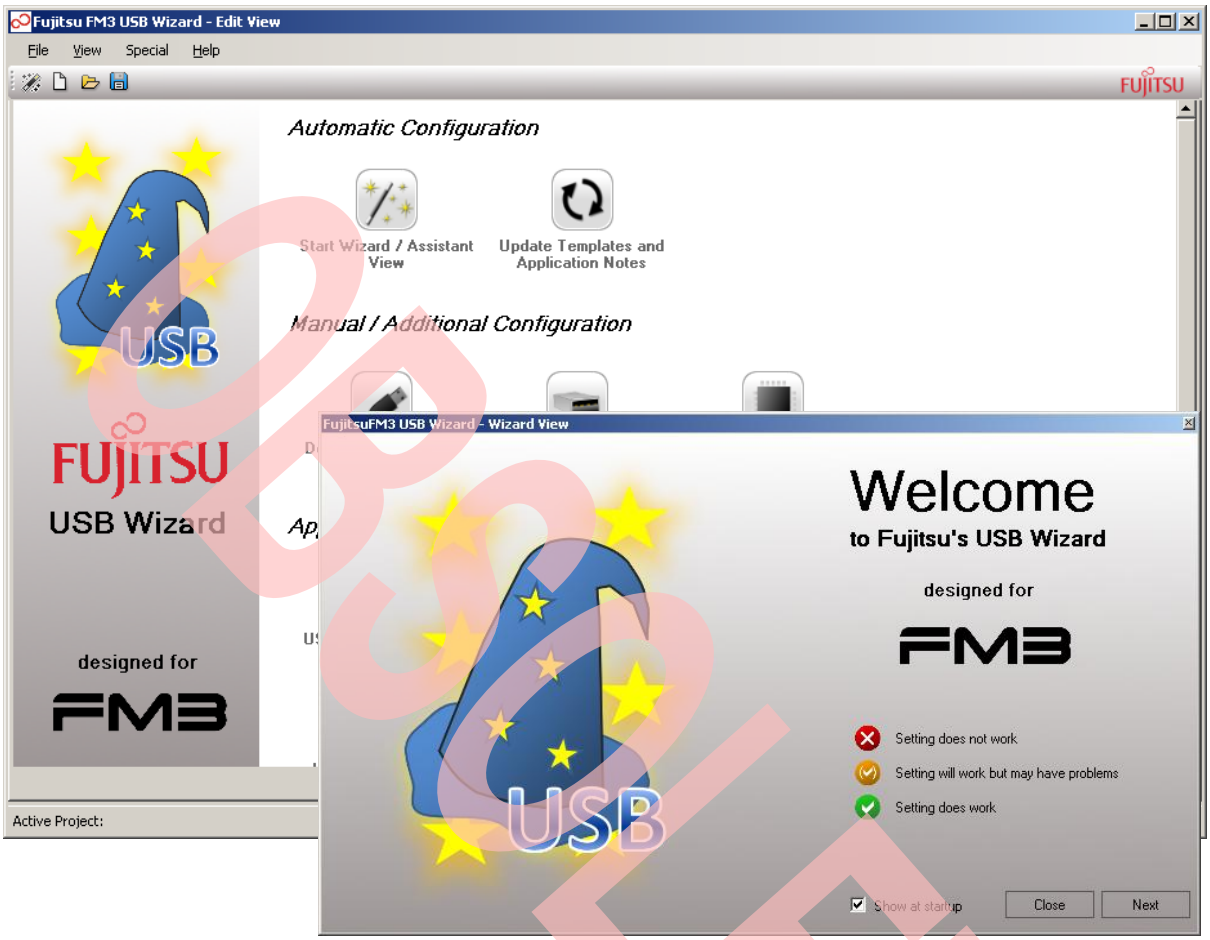


http://mcu.emea.fujitsu.com/mcu_tool/detail/FUJITSU_USB_ASSISTANT.htm

OBSOLETE

1.2 USB Wizard for USB stack >= 2.0, supporting FM3 and higher

For automatically configuration of the USB stack, the Fujitsu USB Wizard can be used for FM3 and higher with use of USB stack version >= 2.0.



http://mcu.emea.fujitsu.com/mcu_tool/detail/FUJITSU_USB_WIZARD.htm

1.3 Program Versions

Date	Issue
2010-09-24	V1.0.0.0, MSc, First Version, Name: “Fujitsu USB Assistant”
2010-11-22	V1.1.0.0, MSc, FR80 MCUs / Host functionality added
2011-02-08	V1.2.0.0, MSc, Software Templates Updates
2011-03-31	V1.3.0.0, MSc, FM3 MCUs added
2011-04-28	V1.4.0.0, MSc, several bug fixes
2011-06-27	V1.5.0.0, MSc, board support, hardware management added
2011-07-21	V1.6.0.0, MSc, IAR Workbench >= 6.20 fix
2012-08-15	V1.9.0.0, MSc, Fujitsu USB Assistant for 16FX, FR80, FM3
2012-08-15	V2.0.0.0, MSc, Fujitsu USB Assistant is now Fujitsu USB Wizard for FM3 or newer

OBSOLETE

2 Usage (USB Wizard)

HOW TO INSTALL AND USE THE FUJITSU USB WIZARD

In this chapter different use cases will be described. This can be simple creating of embedded applications but also manipulating different USB descriptors manually.

2.1 Installation

The installation of the Fujitsu USB Wizard is quite simple.

Figure 2-1: Step 1: Starting installation

While clicking on *Next* and selecting “I accept the agreement” the disclaimer will be accepted.

Figure 2-2: Accept Disclaimer

OBSOLETE

The Fujitsu USB Wizard can be placed on every directory instead of network drives! The USB Wizard requires the .NET Framework (>= 2.0) needed to be configured to run on network devices for security reasons (setting trusted zones).

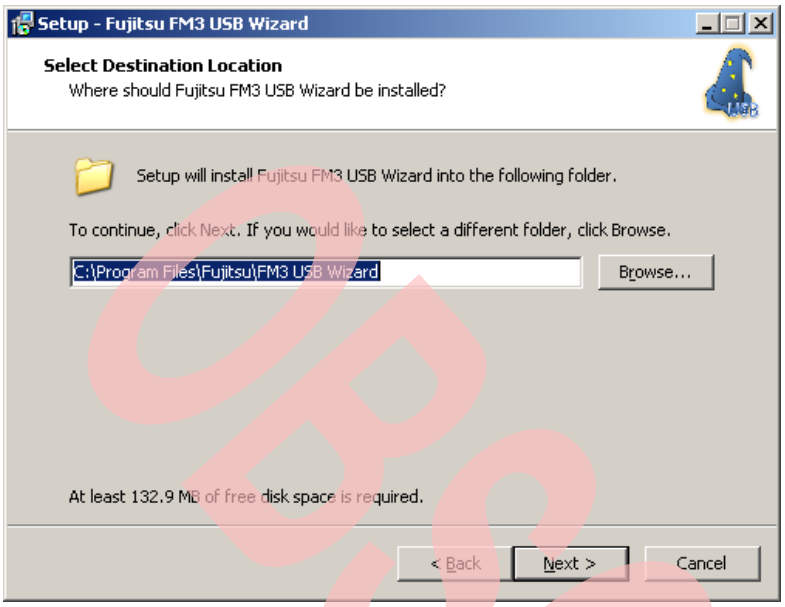


Figure 2-3: Choosing the installation directory

The installation also supports to install the old USB Assistant to run at the same time with the Fujitsu USB Wizard. To install as well the old version, this can be selected here:

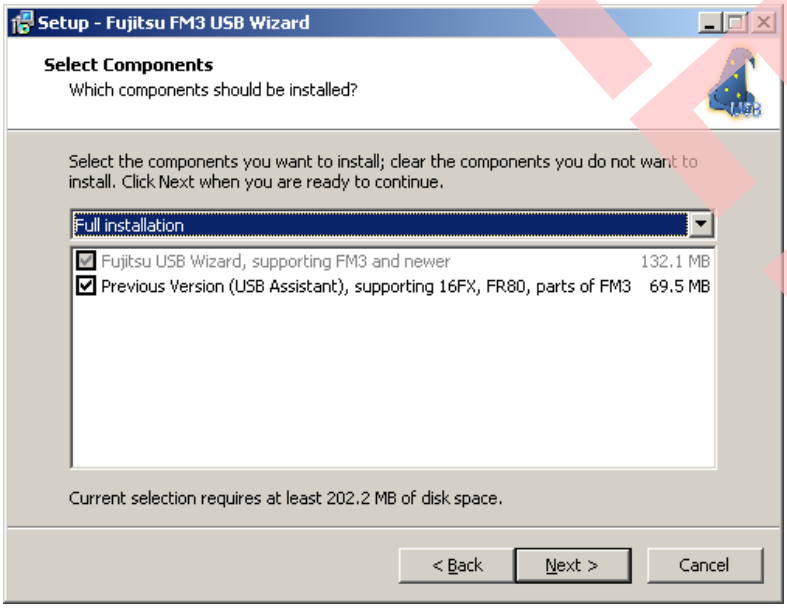


Figure 2-4: Choosing the installation parts

OBSOLETE

At this step the directory in the start menu can be defined.

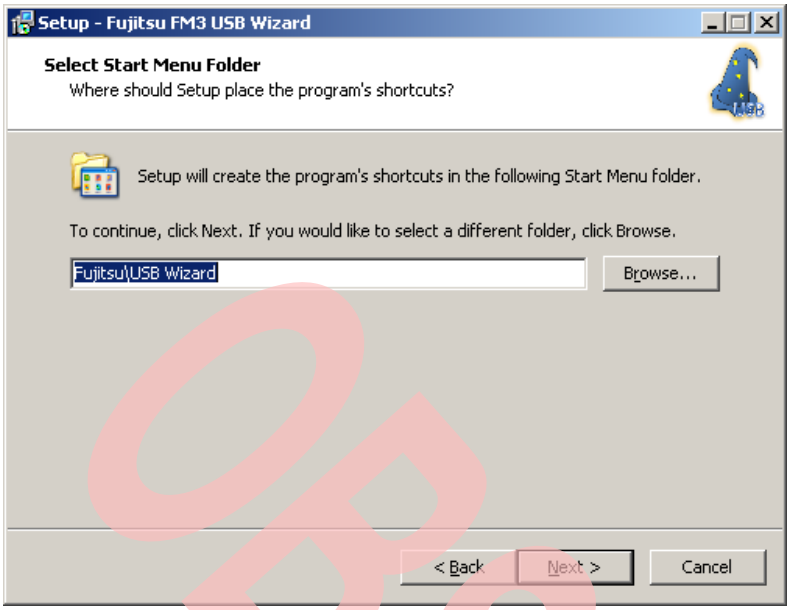


Figure 2-5: Creating start menu entry

Additional links on desktop or the Quick Launch can be also defined during the installation process.

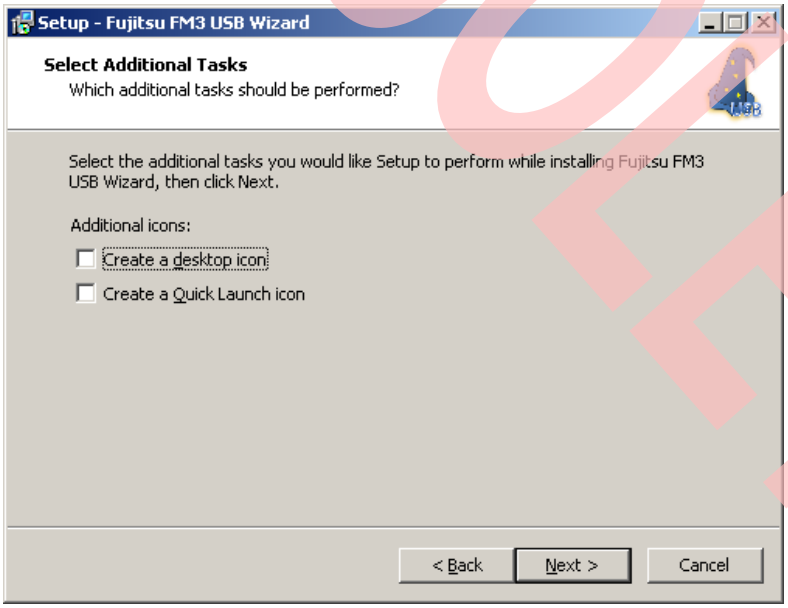


Figure 2-6: Adding additional links on desktop and Quick Launch

OBSOLETE

 © Cypress Semiconductor

At the end a summery will be displayed and after clicking on *Install* the software will be installed.

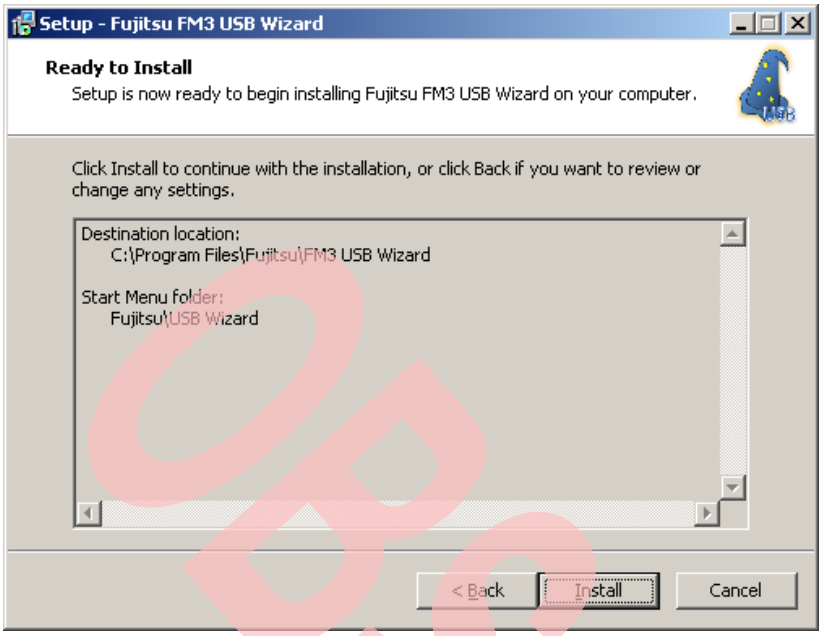


Figure 2-7: Installation Overview

OBSOLETE

2.2 Internet Updates

The USB Wizard is able to do Internet Updates. At first start the USB Wizard is asking for using Internet updates

:

Figure 2-8: Internet Updates

Normally the USB Assistant is NOT using any internet connection. Only after selecting “Yes, keep me informed”, internet updates will be enabled.

If a new Update is available, following dialog will appear:

Figure 2-9: Internet Updates - Download

After pressing “Update” the new installation will be downloaded and started.

OBSOLETE

2.3 First Start – Wizard View

At the first start the Fujitsu USB Wizard will show the Assistant View. The Wizard View can be used to create first time USB embedded applications or in later projects to add different USB functionalities to existing embedded applications.

Figure 2-10: Fujitsu USB Assistant - Assistant View

The other view is the Edit View. With this view the complete USB descriptors can be edited in detail. After using the Wizard View all descriptors can be edited here, too.

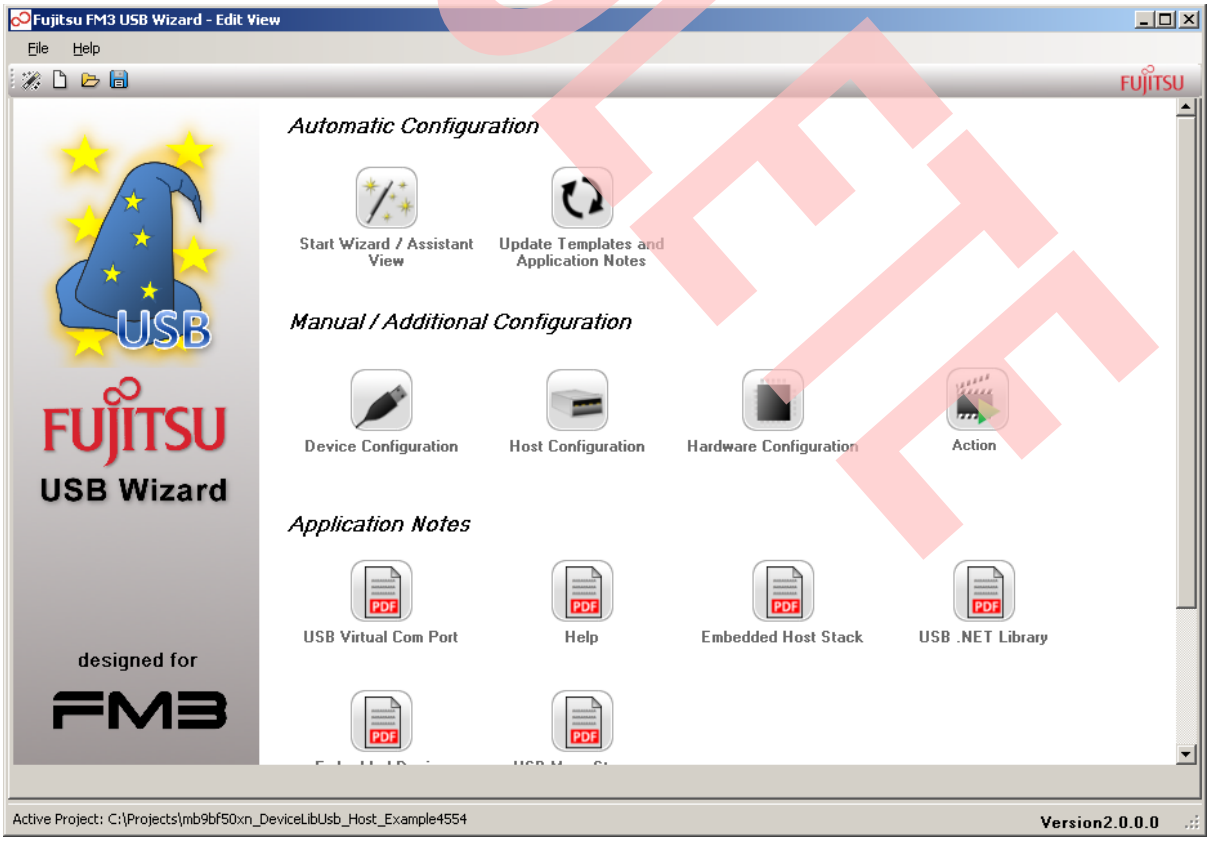


Figure 2-11: Fujitsu USB Wizard – Edit View

OBSOLETE

2.3.1 General: Choose Target Initial Template

In this step the type of environment is selected. This can depend on the MCU but can also depend on an evaluation board (Starterkit). If a starterkit is selected, the assistant will automatically choose hardware settings for the USB peripheral. If the “Add Board Support” option was chosen, the assistant will also add some human interface to act with (buttons, LEDs, UART).

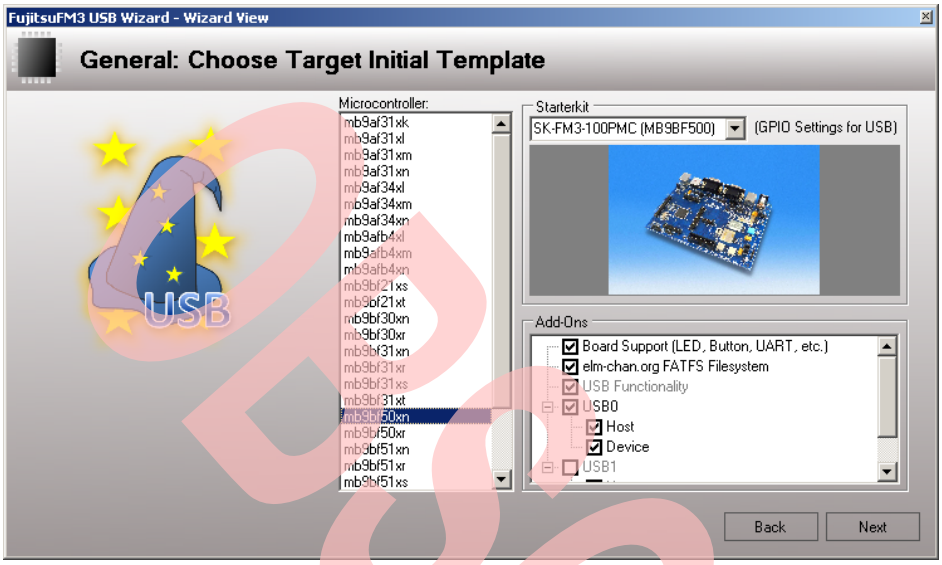


Figure 2-12: Wizard View – MCU Template

2.3.2 USB Device: Choose USB Application Type

The Program starts normally in Wizard View. The Wizard implements two different templates combined in one: An application template and an MCU template. The application template represents the USB application, for example: Virtual Com Port. The MCU template represents the workspace settings for the specified MCU. In the first step the template selector will open:

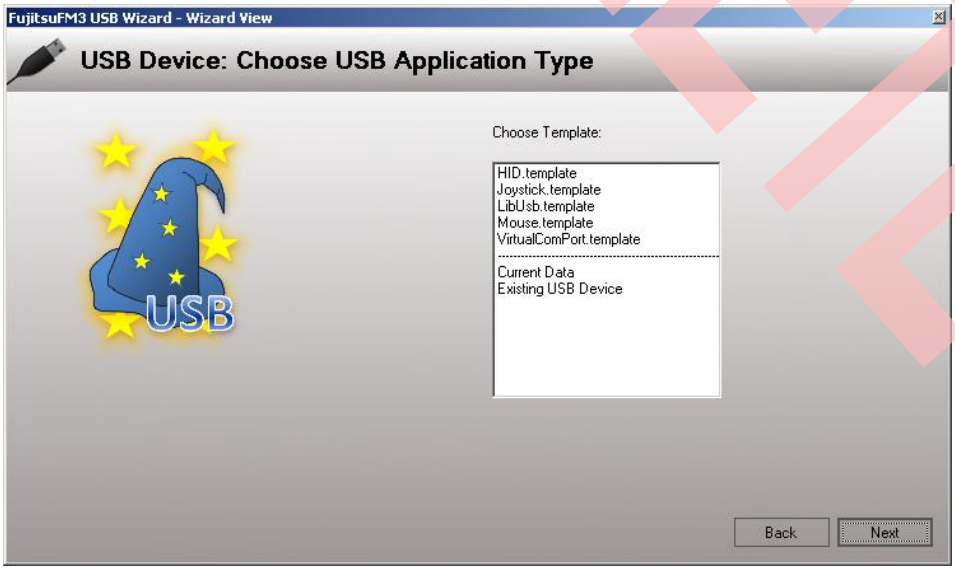


Figure 2-13: Wizard View - USB Application Template

At new template different existing templates can be chosen, but also an existing device.

OBSOLETE

Existing USB Device: This option can be used to read in all descriptor files from an existing device. For starting with developing USB devices this helps to adapt settings. The protocol part is left open! The user should also change the Vendor ID and Product ID to his specific IDs.

Current Data: This option is used if the USB Wizard view is called manually after setting all descriptors in Edit View. This can be used to create an initial USB template with own settings.

2.3.3 USB Device: Edit device descriptor settings

The next step is used to set general settings like Vendor ID, Product ID, etc. The wizard will help to configure well known settings. Warnings are orange and errors are red.

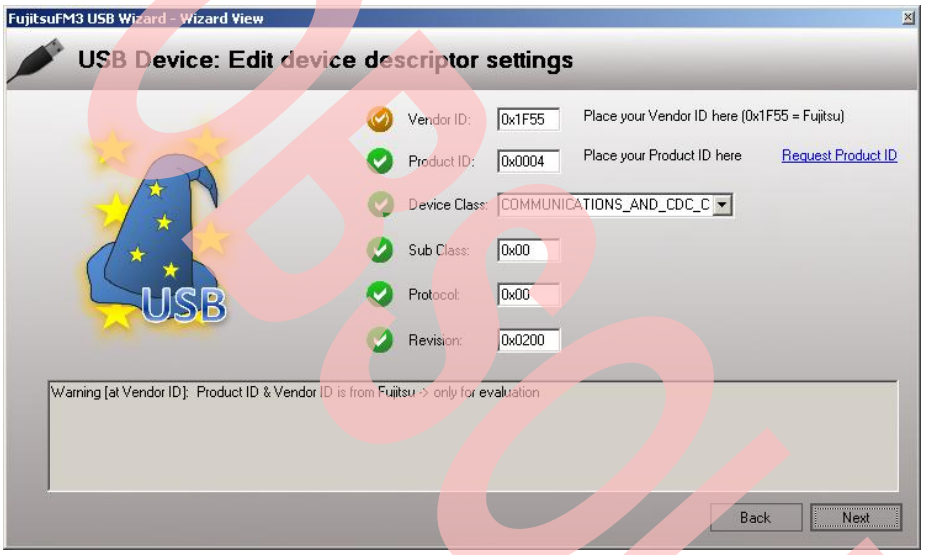


Figure 2-14: Wizard View - Device Descriptor

2.3.4 USB Device: Power Management

Step 3 helps configurating the power management and remote wakeup features.

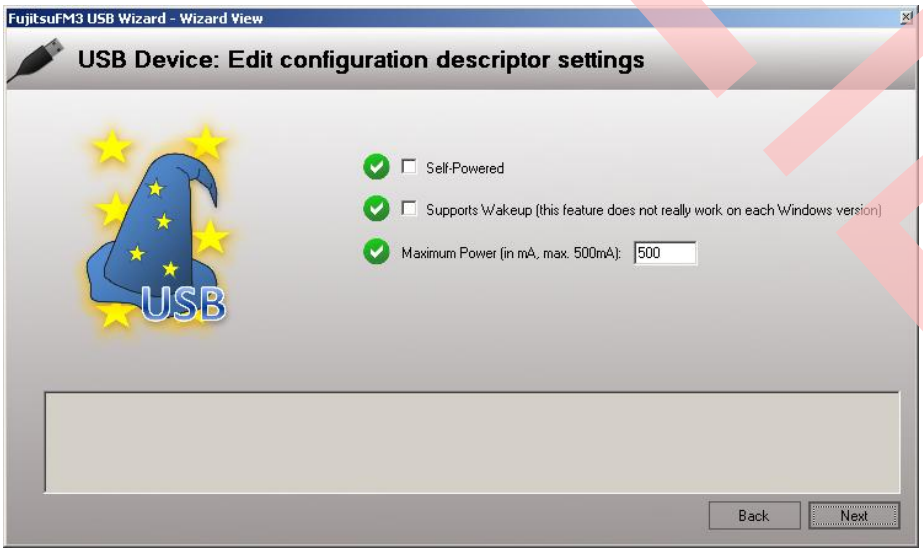


Figure 2-15: Wizard View – Configuration Descriptor

OBSOLETE

2.3.5 USB Host: Select Driver Support

For Host different driver support can be enabled or disabled.

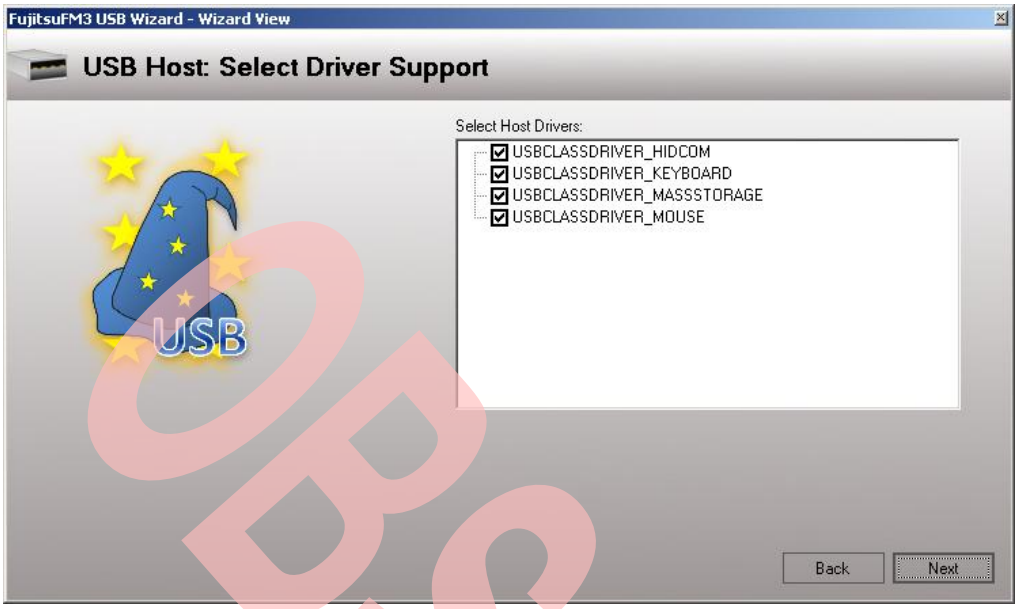


Figure 2-16: Wizard View – Host Settings

2.3.6 Code Generation: Project Naming

In this window the project name and location is set.. After clicking on *Next* all files will be copied and patched.

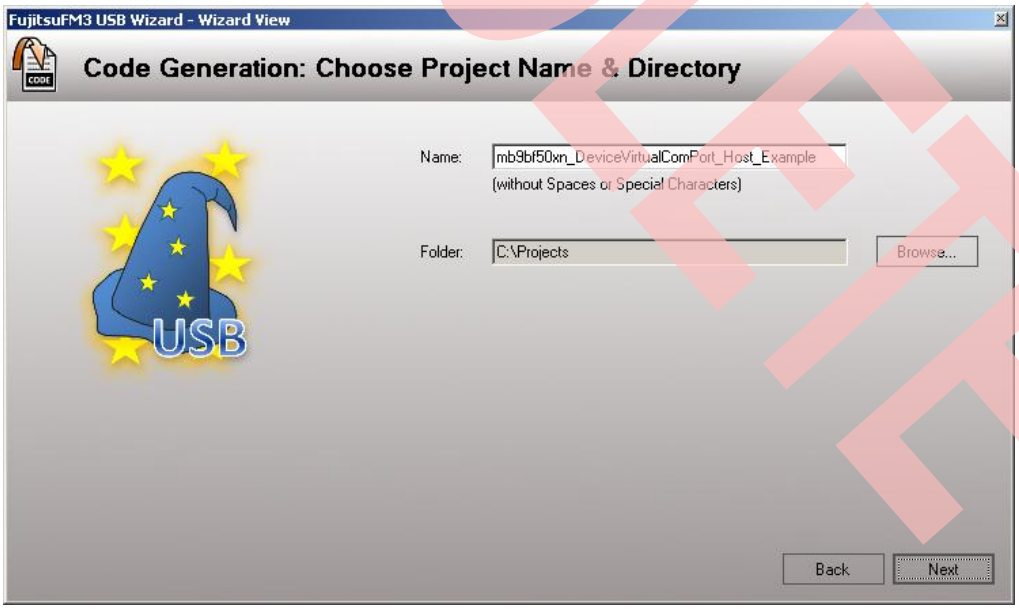


Figure 2-17: Wizard View – Project Name

OBSOLETE

2.3.7 Last Steps

Now the configuration process is finished and after clicking *Finish* the program will show the Edit View. The complete project was now created on the specified location. The Edit View will show all configured options. To update the descriptors, the project only has to be saved. To recreate and overwrite the USB class file template, the Code Creation button in Edit View can be used (see also chapter 3.4, button (4)).

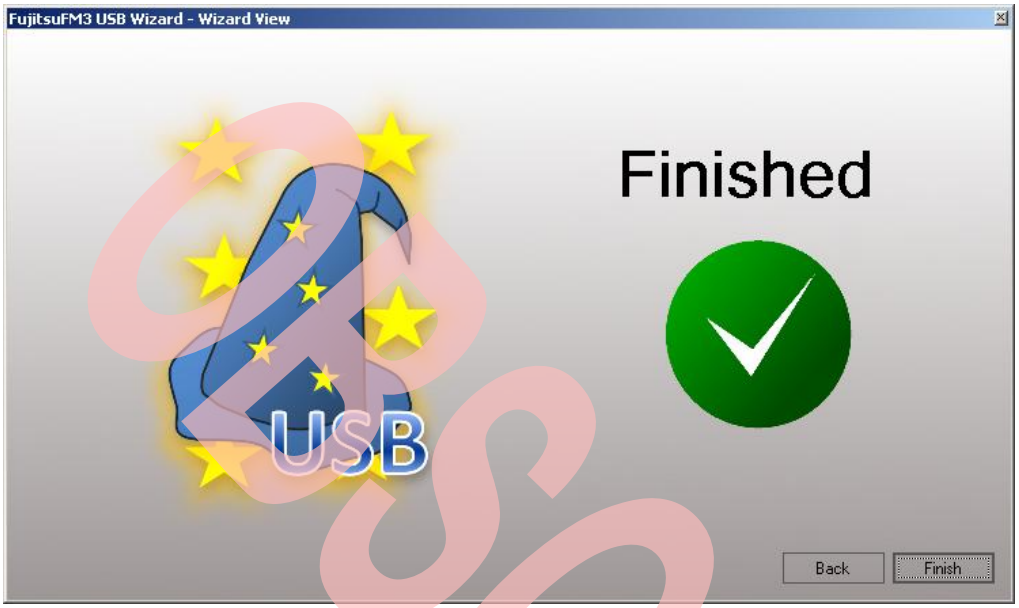


Figure 2-18: Wizard View – Finished

OBSOLETE

2.3.8 View Result

The generated result described before can be now used in several formats.

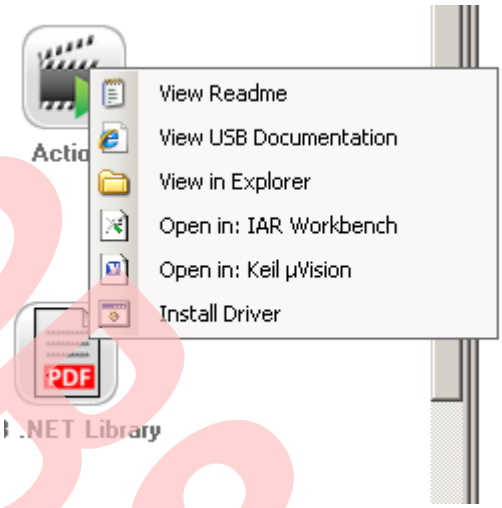


Figure 2-19: Open several options of created code

OBSOLETE

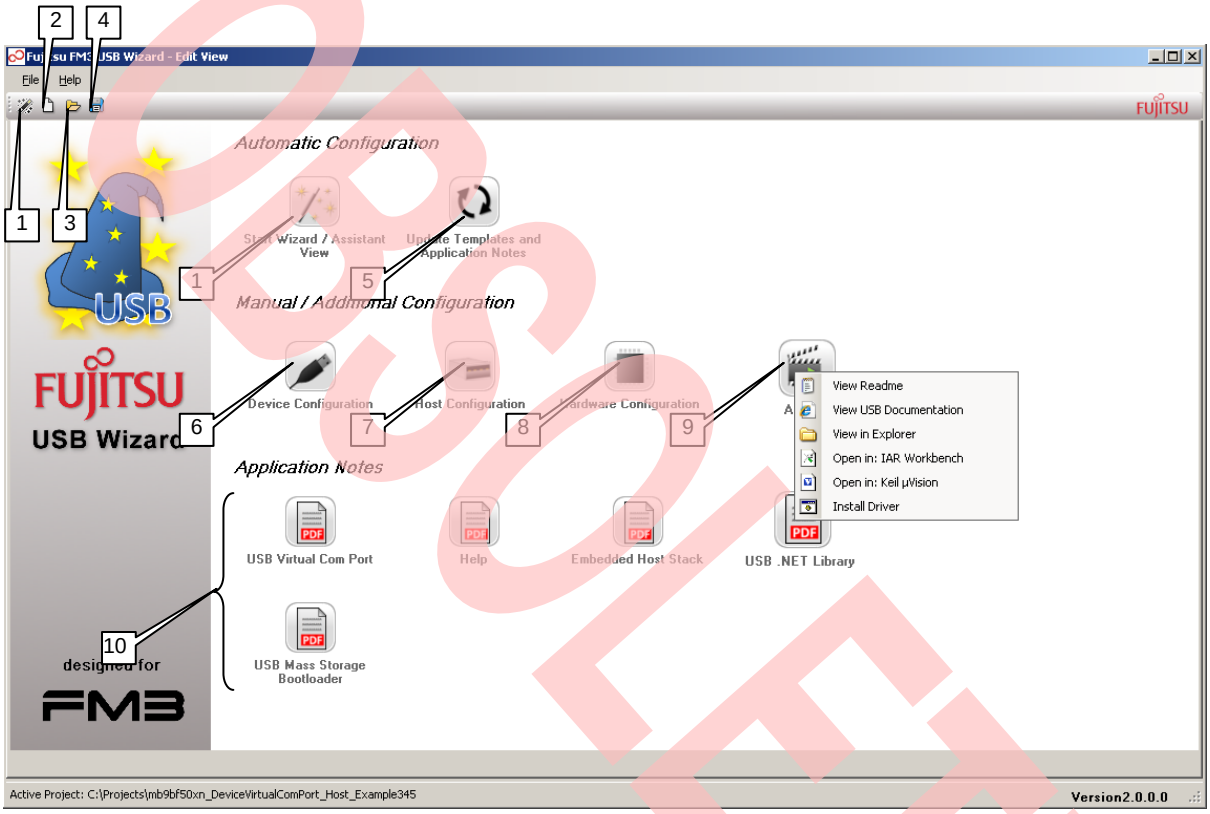
 © Cypress Semiconductor

2.4 Edit View

The Edit View is used to create descriptors manually. In this mode the most complex descriptors can be created. If there is some information not configurable or missing, contact the programmer via mcu_ticket.fseu@de.fujitsu.com.

2.4.1 Welcome Screen

As first start, the Assistant View should be used. The Assistant View can be started via pressing “Start Assistant View”.

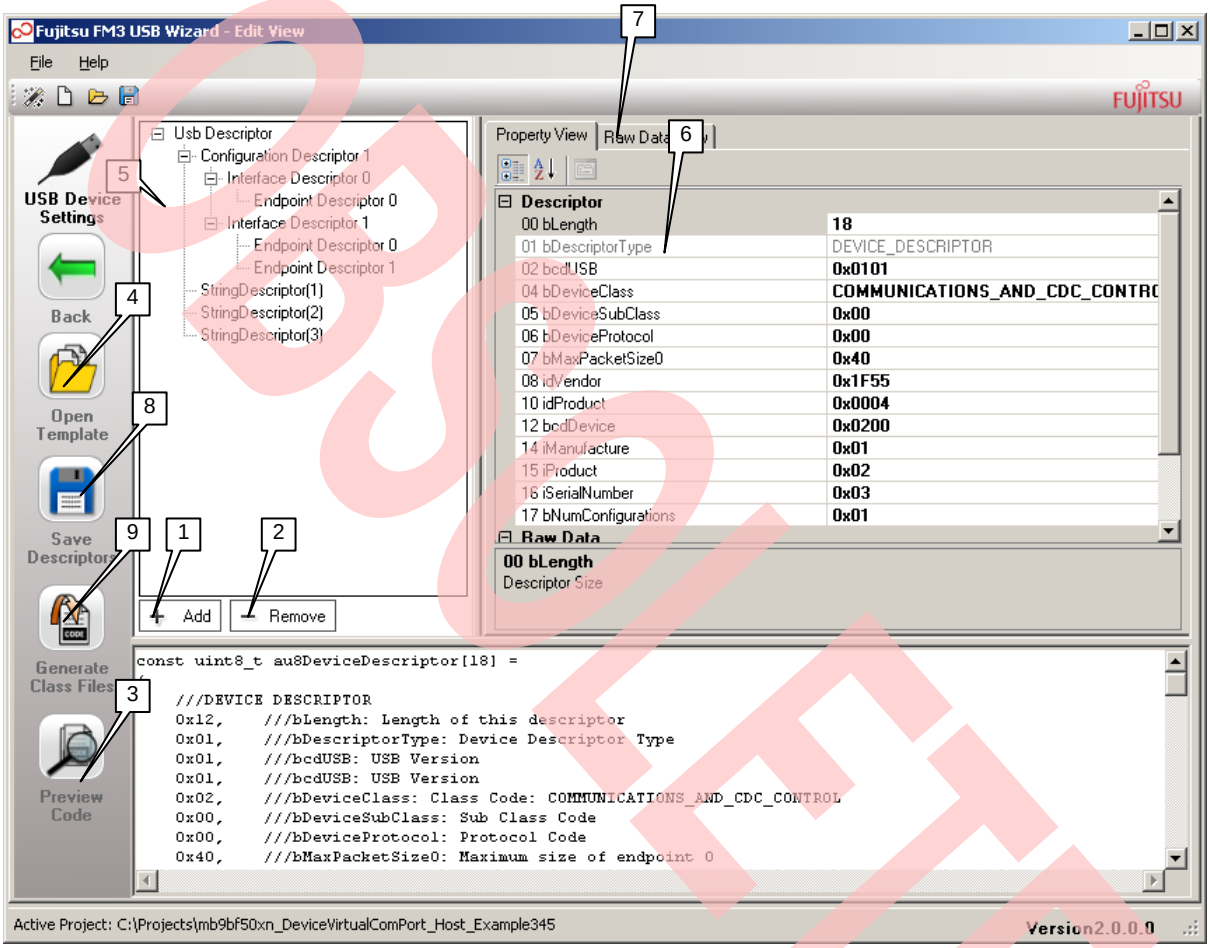


1) Start Wizard
2) New empty configuration
3) Open Workspace
4) Save
5) Update MCU Templates & Application Notes via internet
6) Switch to manual device configuration
7) Switch to manual host configuration
8) Hardware specific settings
9) Execute different actions
10) View USB application notes

OBSOLETE

2.4.2 Device Mode Settings

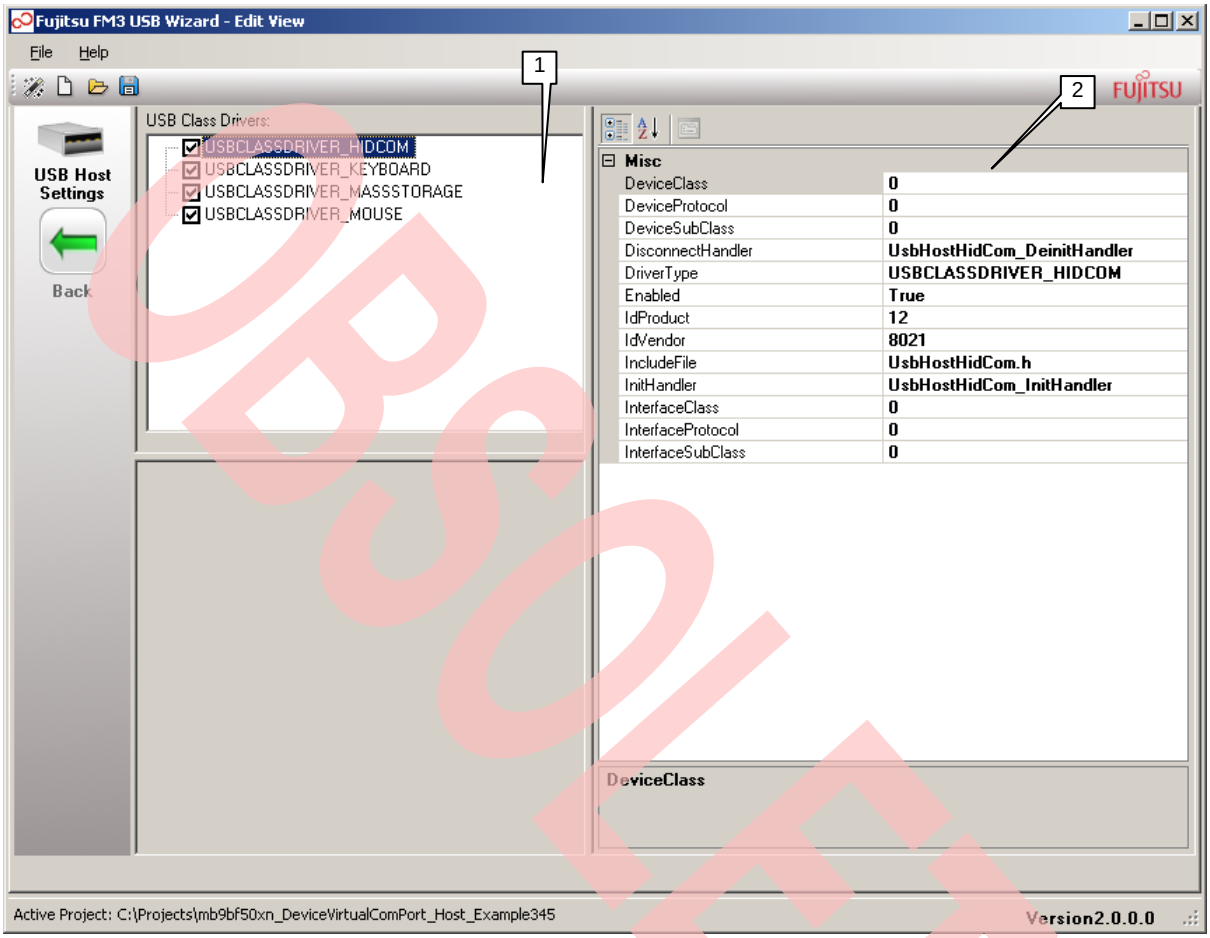
This view can be used to manually changing USB descriptor data. (1) is used to add a descriptor. For adding a descriptor the parent descriptor has to be selected in which the new descriptor shall be created. By clicking on “Usb Descriptor” the root descriptor will be selected and the descriptor data will be shown in (6). With (2) a selected descriptor can be removed. (3) shows the files *UsbDescriptors.h*, *UsbClass.c* and *UsbClass.h*. (5) shows the USB descriptor tree. (6) shows the USB descriptor data. With (7) it can be switched in a RAW data mode.



OBSOLETE

2.4.3 Host Mode Settings

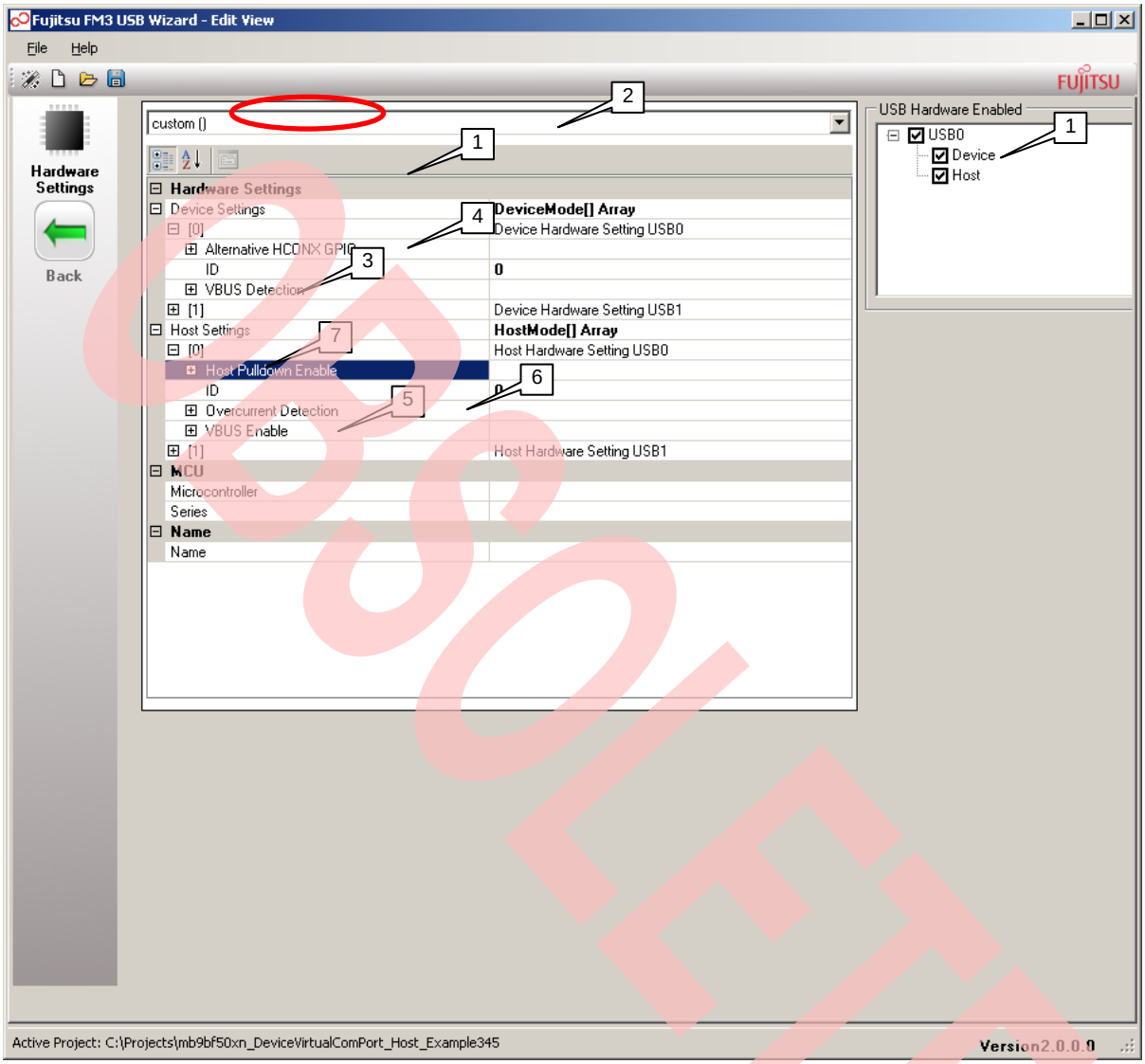
For USB Host the different virtual driver modules can be configured. (1) describes the different devices which can be recognized. (2) is used to change matching features of the selected device driver (1).



OBSOLETE

2.4.4 Hardware Settings

In this panel different hardware settings can be configured. USB Device / USB Host can be enabled or disabled (1). For different starterkits / evaluation boards a preselection of hardware settings can be done (2).



3) The VBUS signal detection settings can be chosen.

4) If devices do not have a special port for rising the Full-Speed pull-up, a port-pin can be specified for this.

5) In Host Mode often ICs are used to switch the VBUS line on or off and recognize overcurrent. For overcurrent detection the signal detection settings can be chosen.

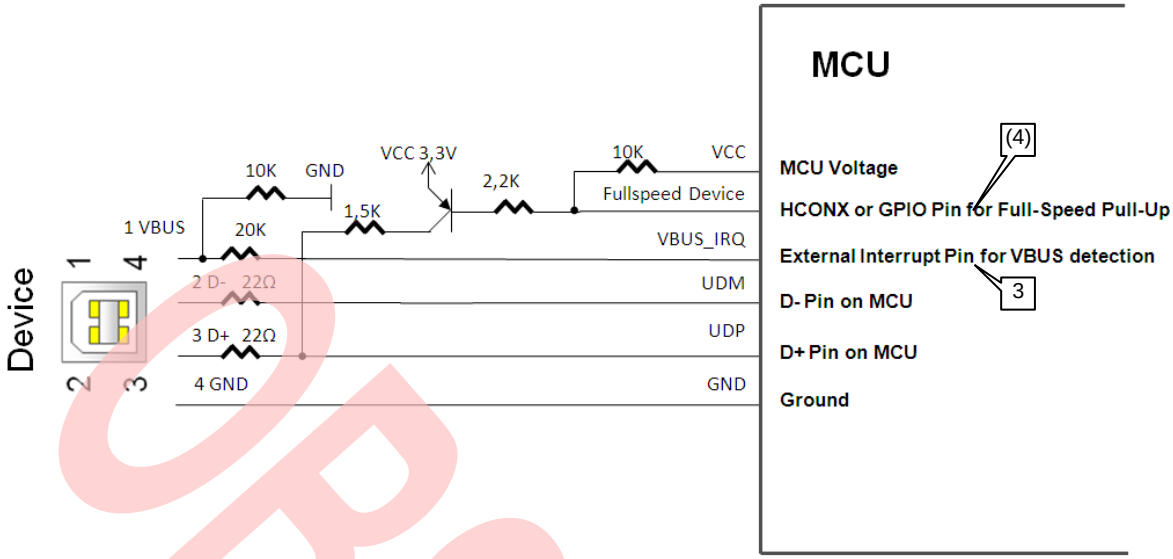
6) In Host Mode often ICs are used to switch the VBUS line on or off and recognize overcurrent. To enable VBUS line settings can be configured here.

7) If the Host supports USB OTG or switches between Host / Device mode, this setting can be used.

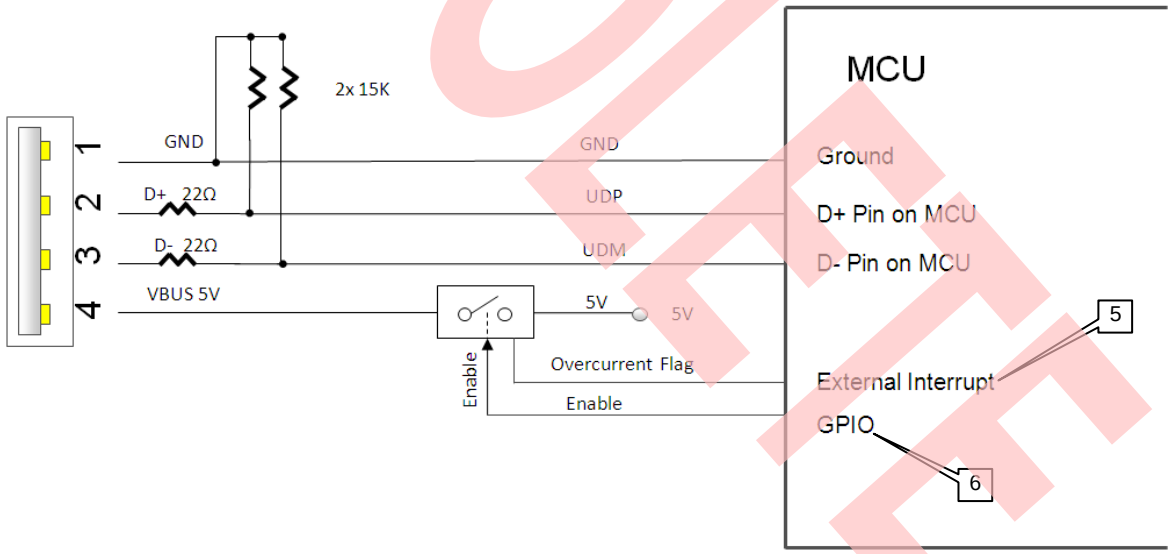
OBSOLETE

 © Cypress Semiconductor

2.4.4.1 Example: USB Device

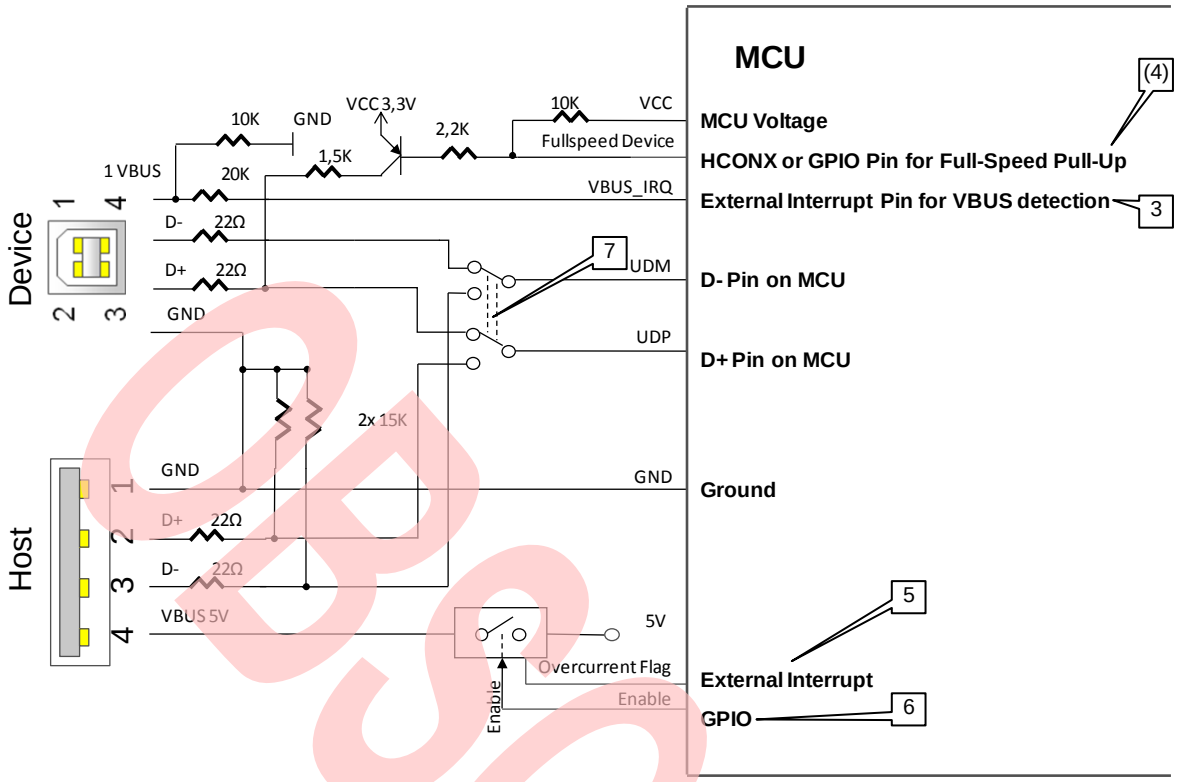


2.4.4.2 Example: USB Host

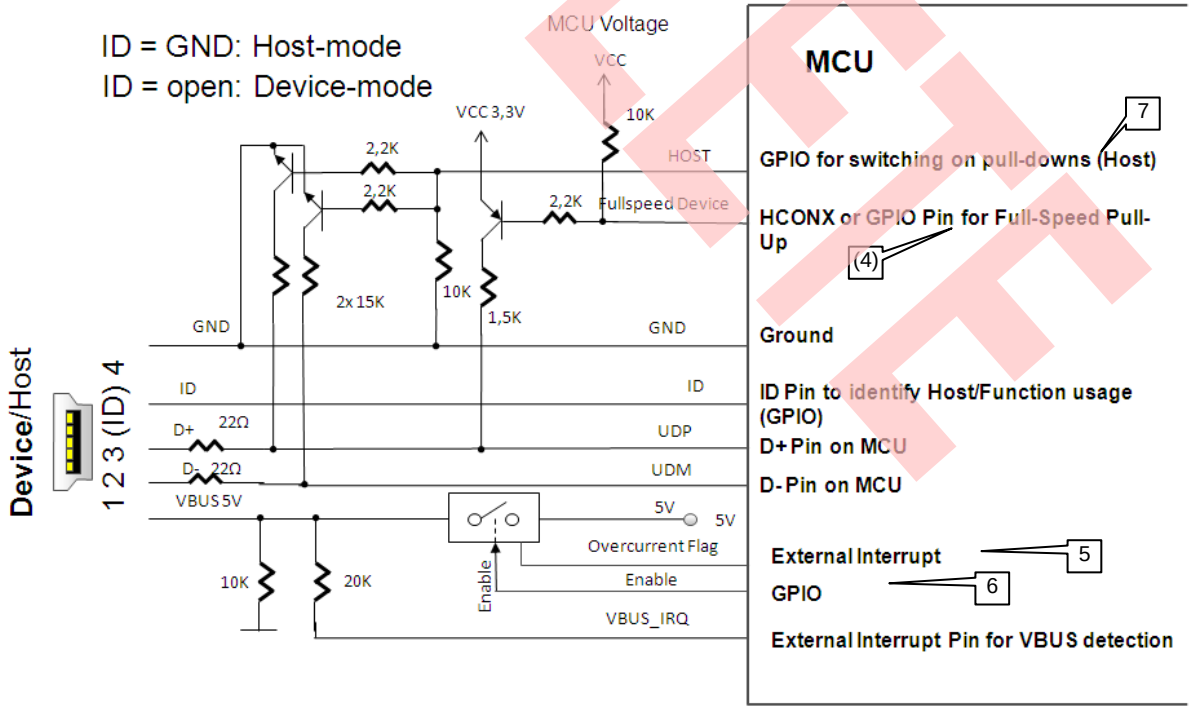


OBSOLETE

2.4.4.3 Example: USB On-Demand

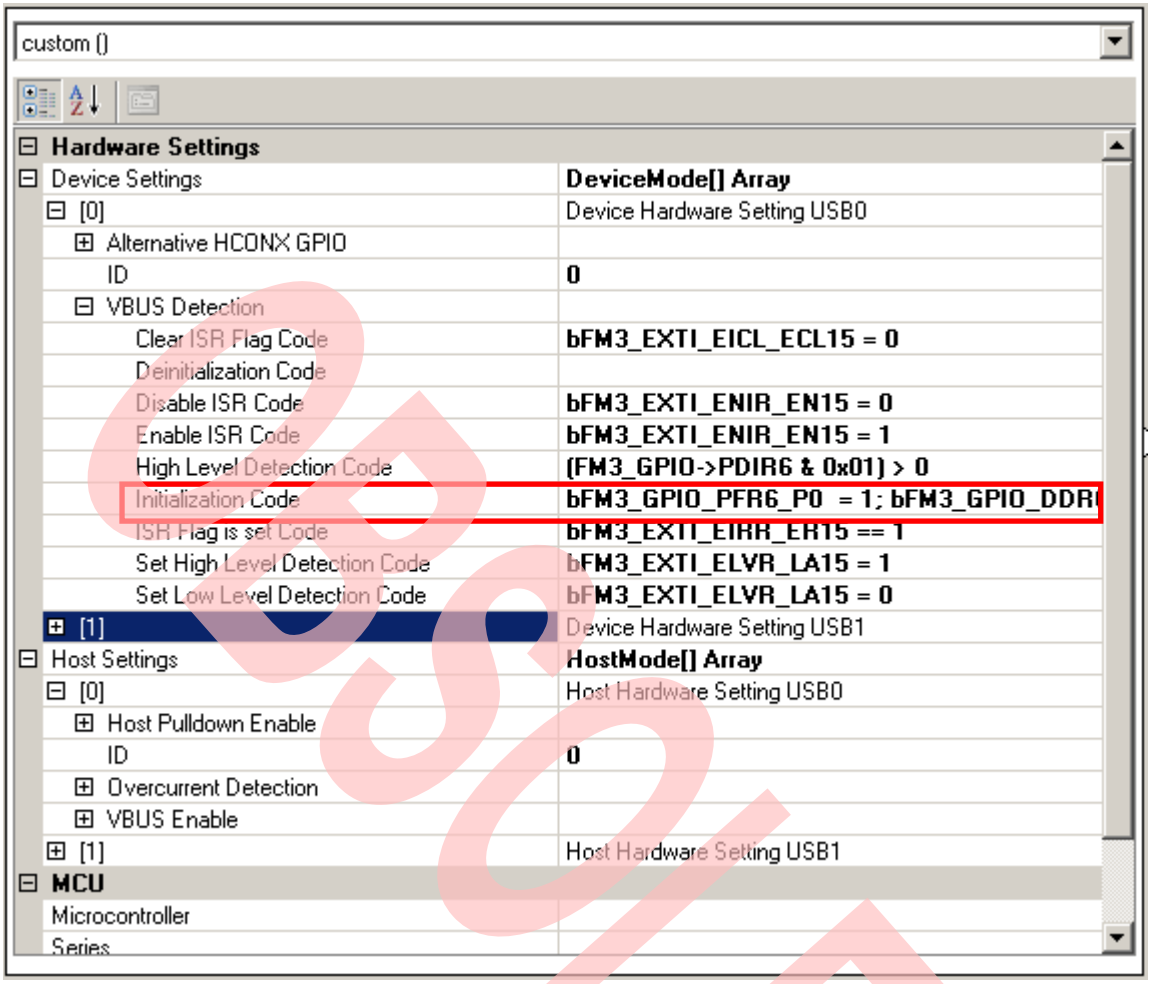


2.4.4.4 Example: USB On-The-Go



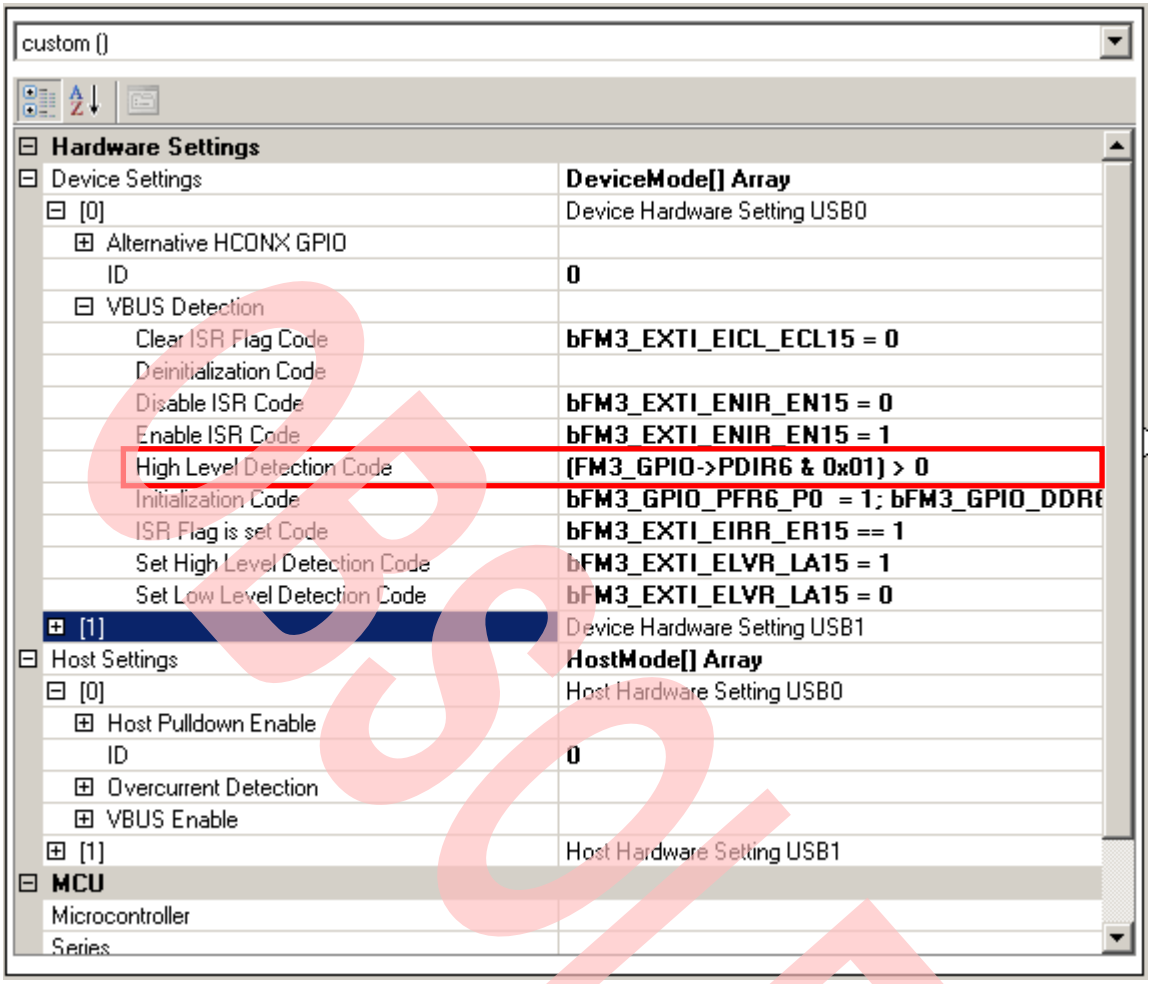
OBSOLETE

2.4.4.5 Example VBUS Detection (Device) - Initialization Routine (for FM3)



OBSOLETE

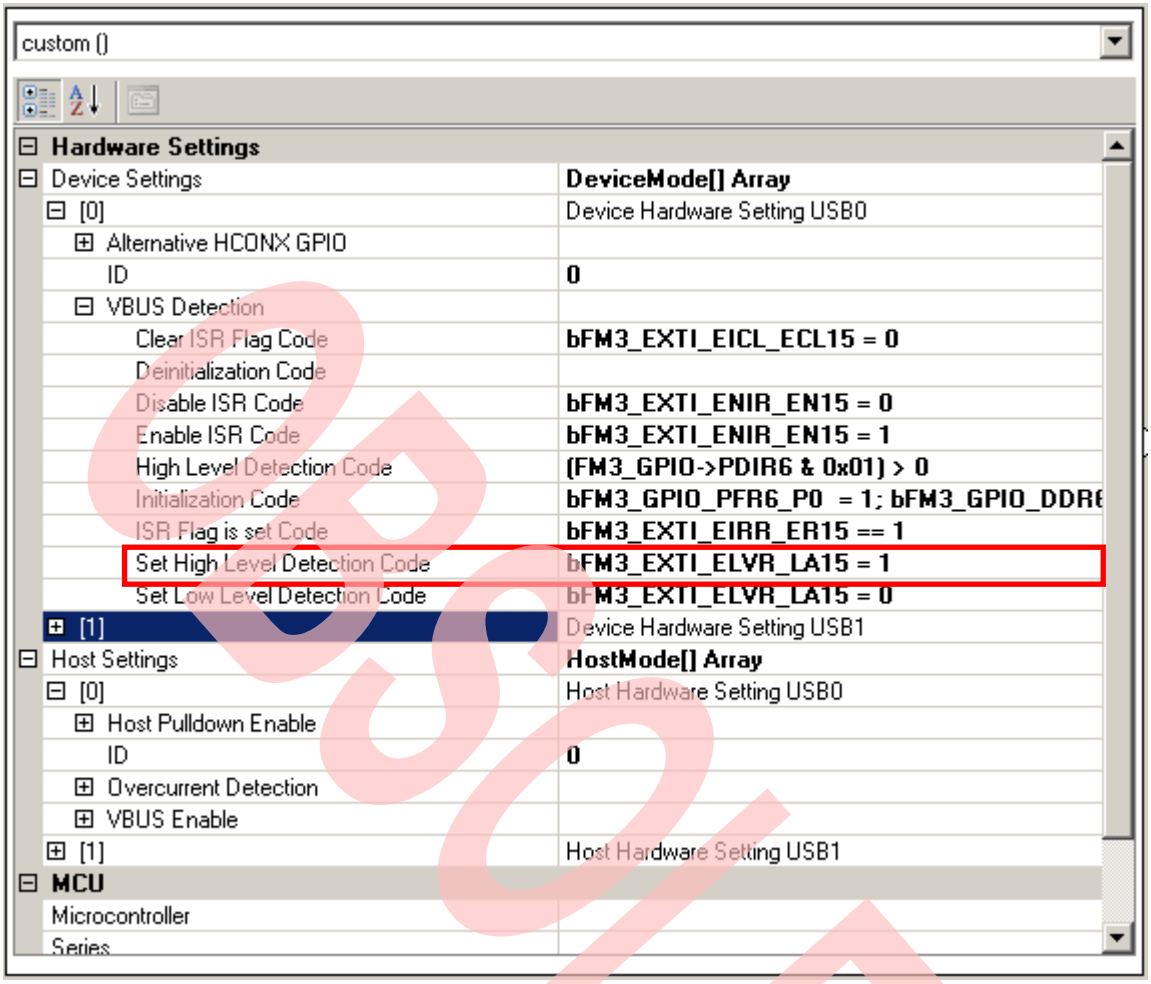
2.4.4.6 Example VBUS Detection (Device) - Detect H level (for FM3)



OBSOLETE

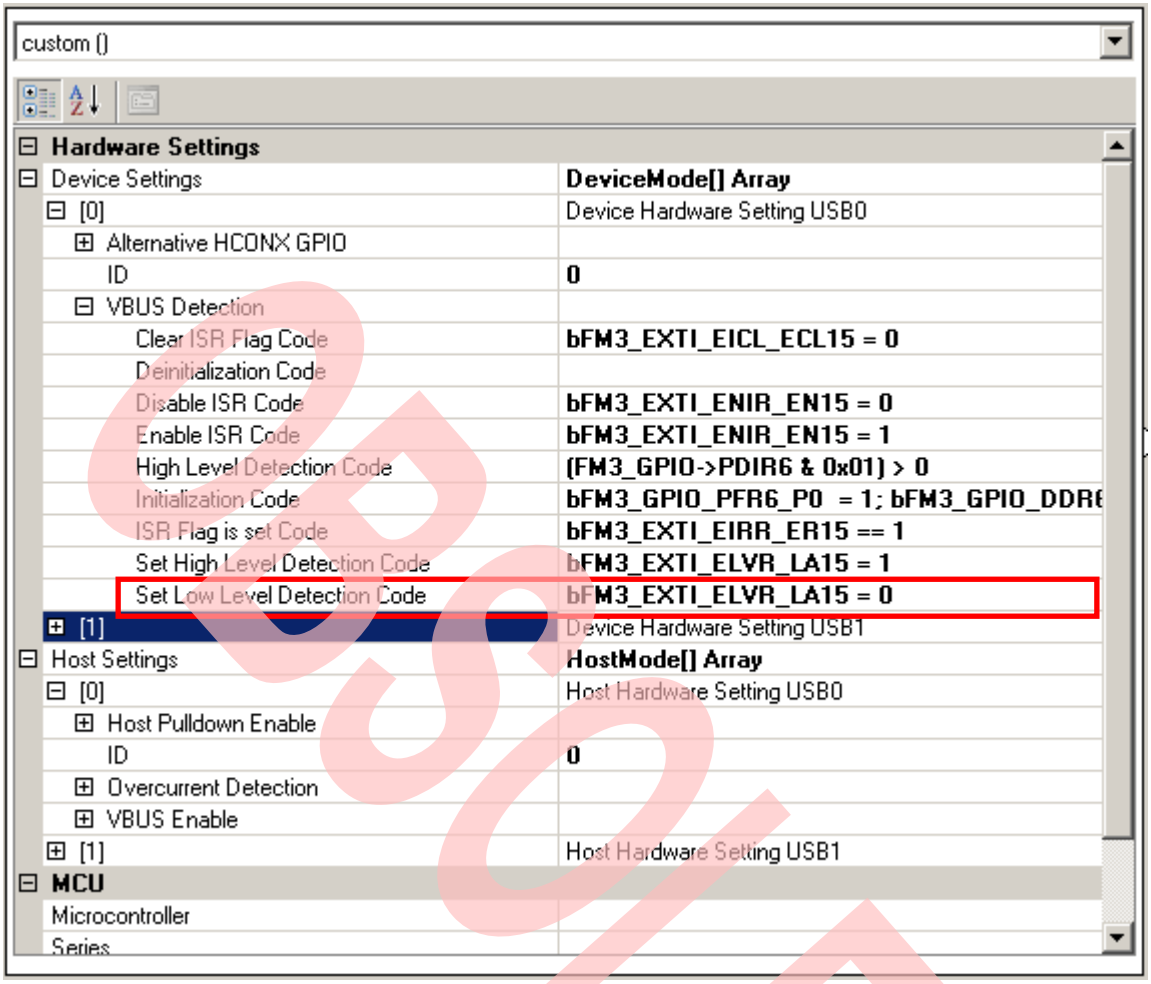
 © Cypress Semiconductor

2.4.4.7 Example VBUS Detection (Device) - Set H level ISR (for FM3)



OBSOLETE

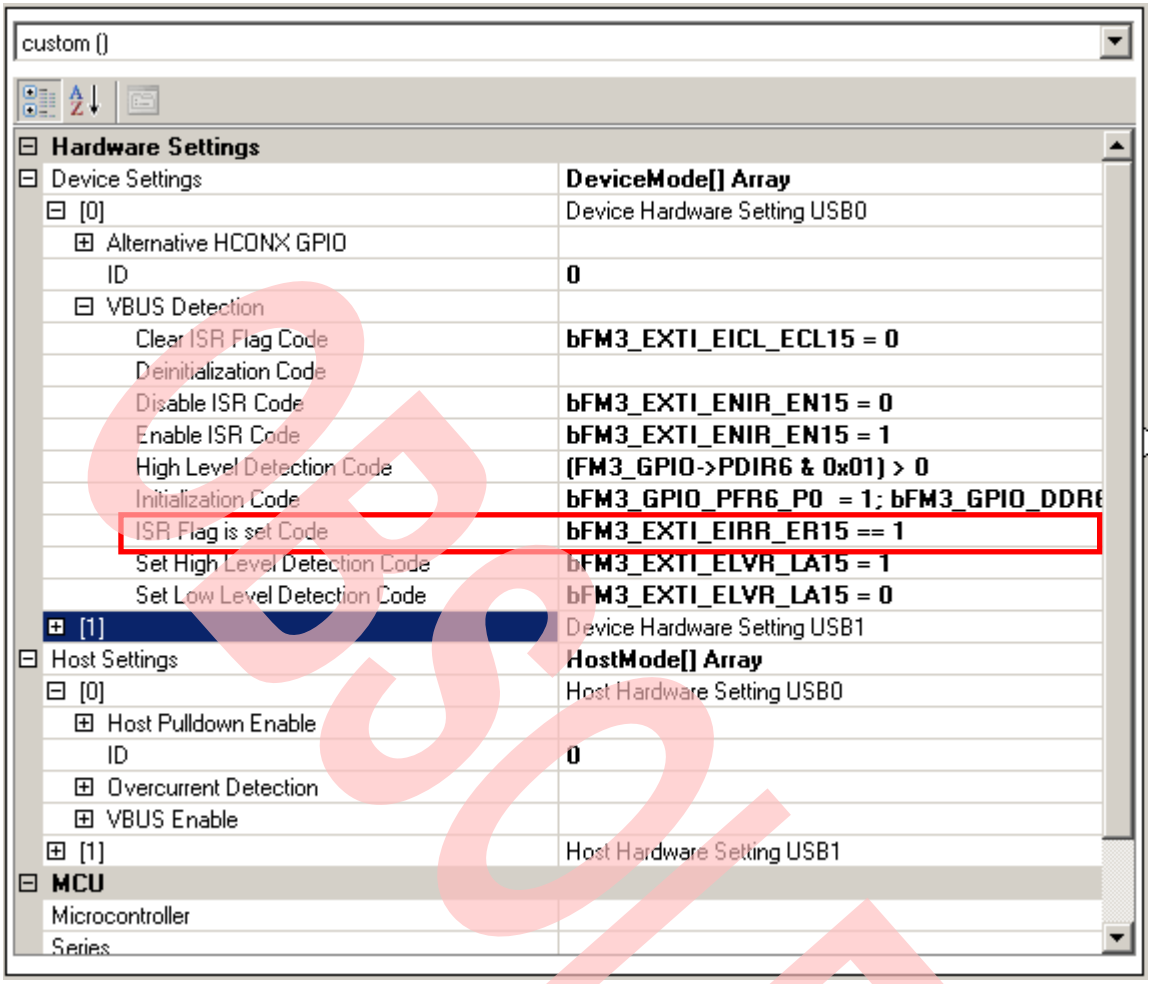
2.4.4.8 Example VBUS Detection (Device) - Set L level ISR (for FM3)



OBSOLETE

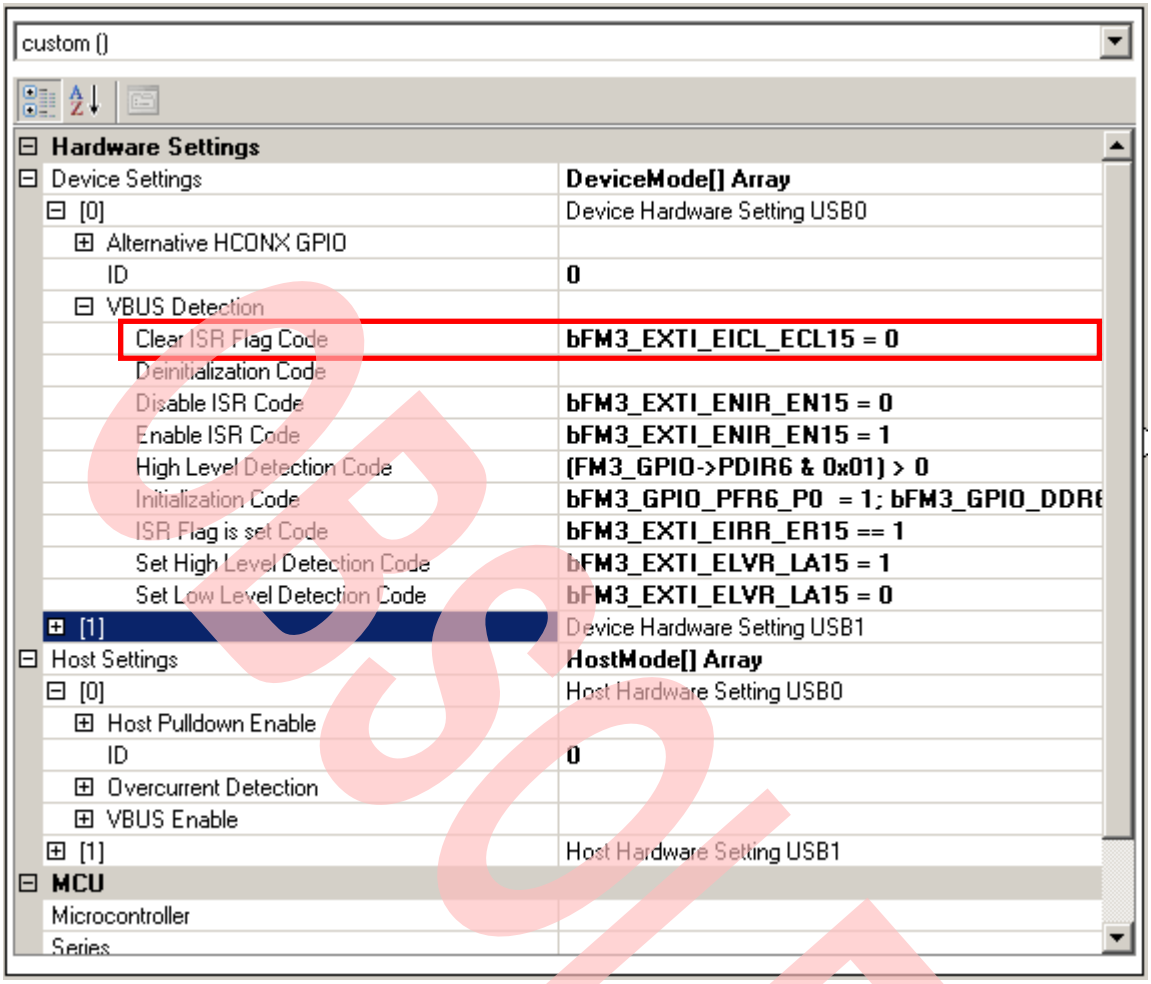
 © Cypress Semiconductor

2.4.4.9 Example VBUS Detection (Device) – ISR flag is set (for FM3)



OBSOLETE

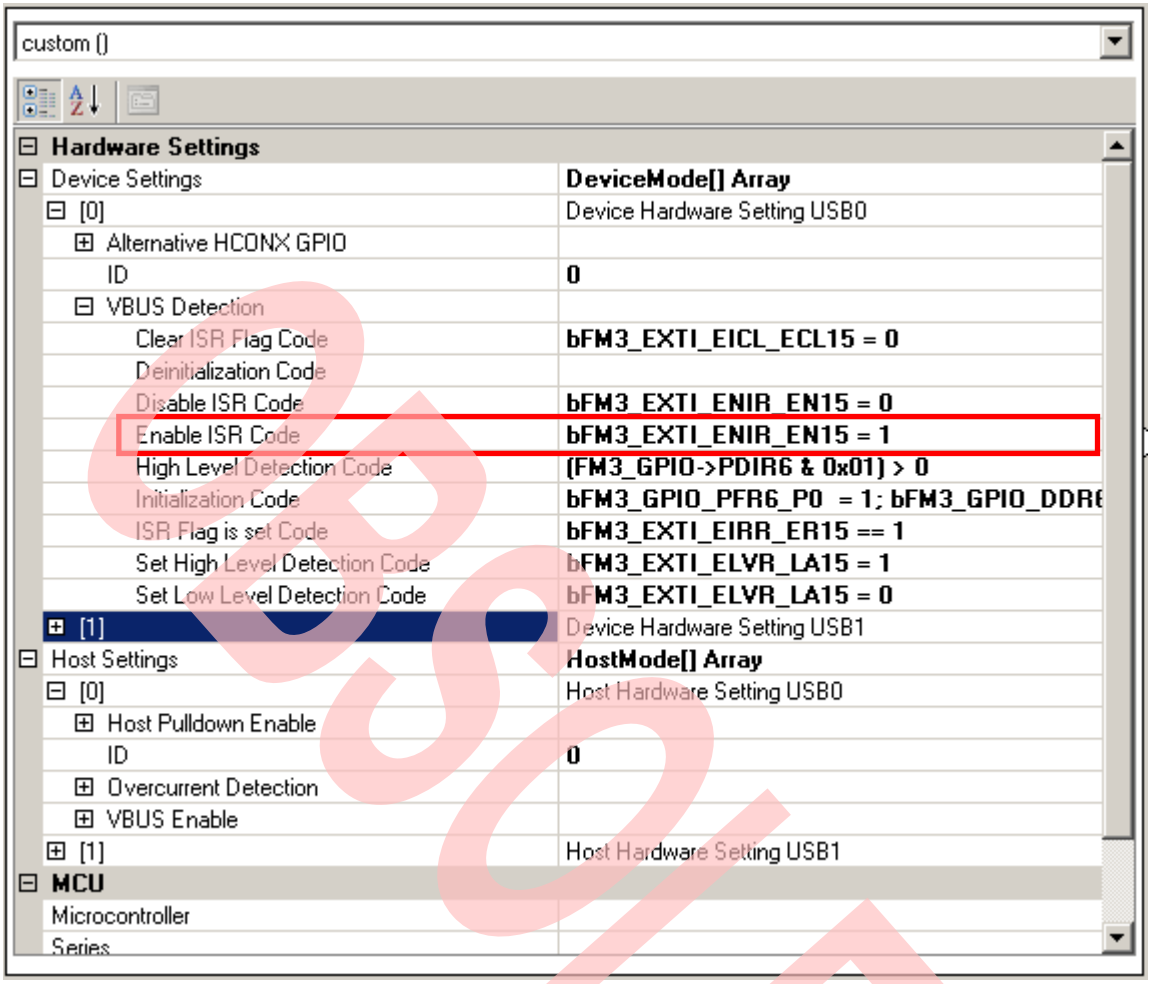
2.4.4.10 Example VBUS Detection (Device) – Clear ISR flag (for FM3)



OBSOLETE

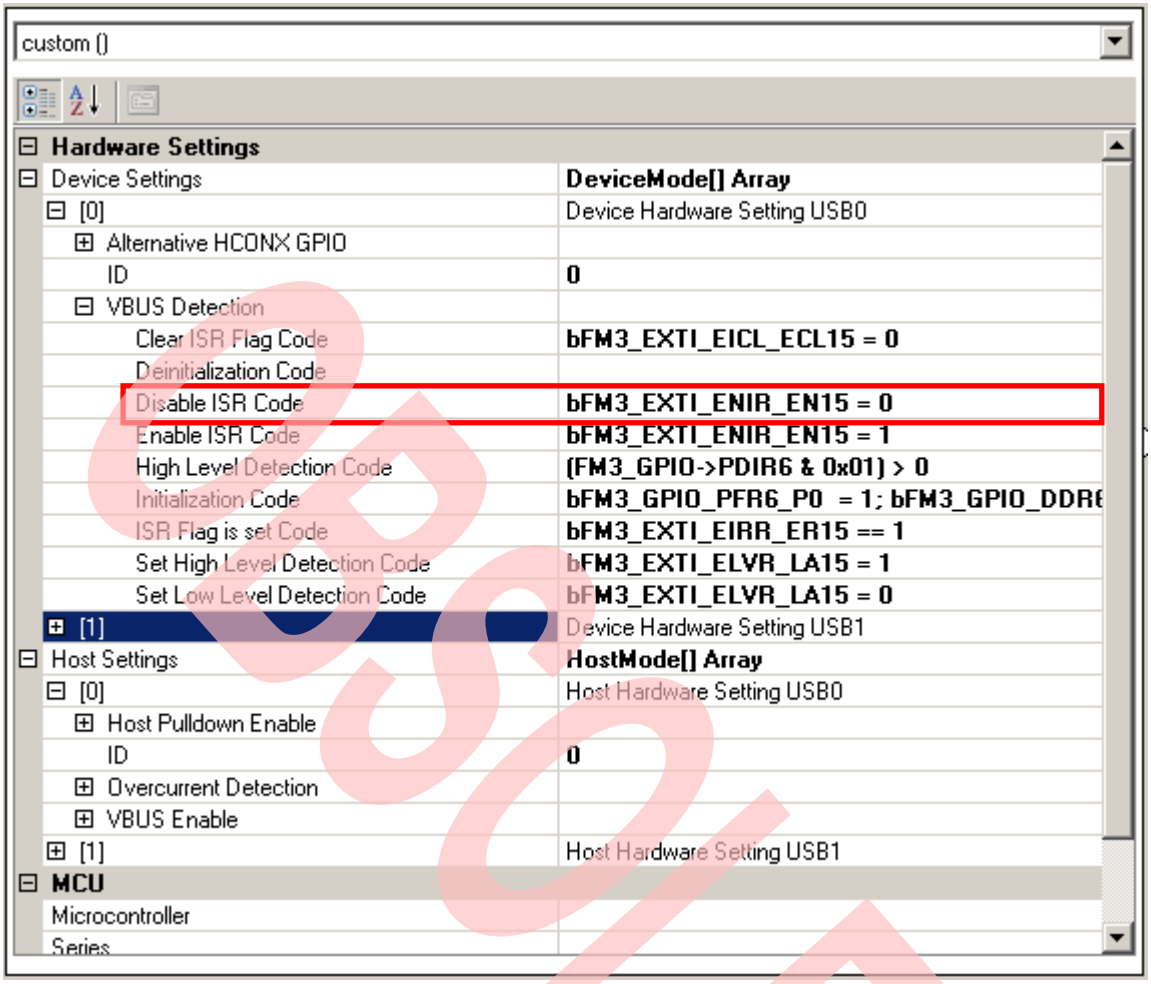
 © Cypress Semiconductor

2.4.4.11 Example VBUS Detection (Device) – Enable ISR (for FM3)



OBSOLETE

2.4.4.12 Example VBUS Detection (Device) – Disable ISR (for FM3)



OBSOLETE

 © Cypress Semiconductor

2.4.5 Open / Load USB Configurations

The Fujitsu USB Assistant can load an IAR or µVision Workbench Project.

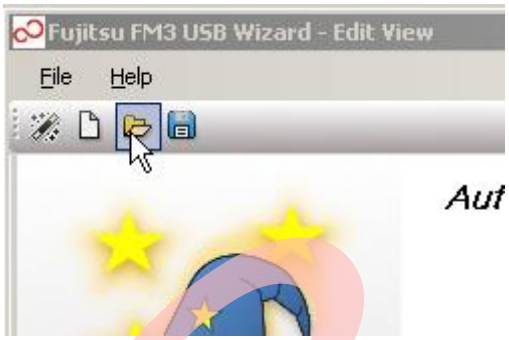


Figure 2-20: Edit View - Open File

OBSOLETE

2.4.6 Save USB Configurations

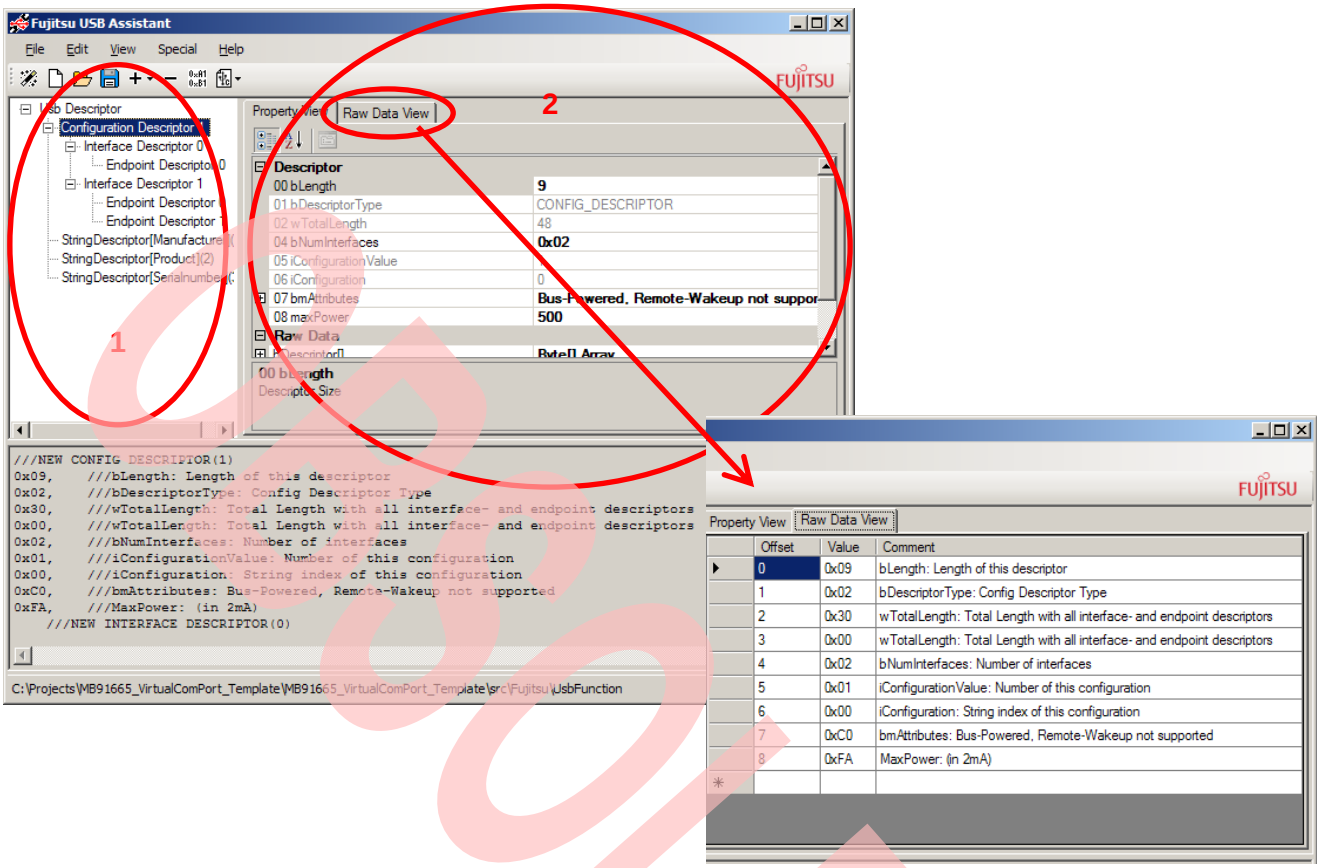
The only file which will be saved as configuration is the *UsbDescriptors.h* file. Before a configuration can be saved, at least a configuration descriptor containing at minimum one interface descriptor has to exist in the configuration.

OBSOLETE

 © Cypress Semiconductor

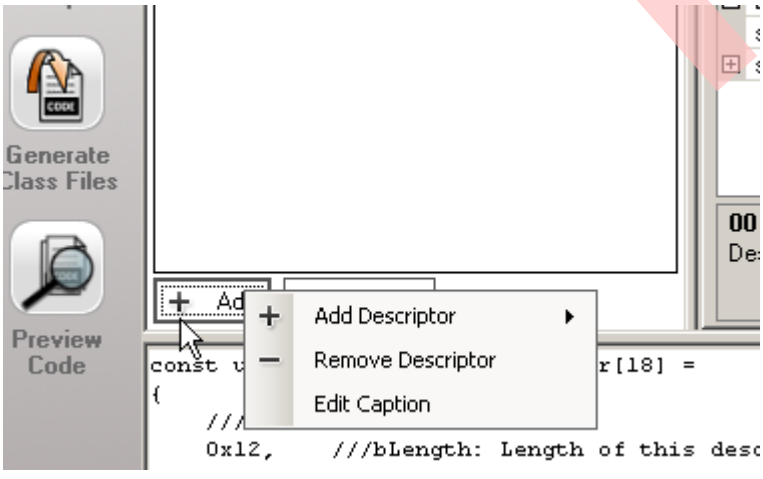
2.4.7 Edit Descriptors

First a descriptor must be selected (1), before the different values can be edited in the Property View (2). It is possible to edit the descriptor also in raw data view (3).



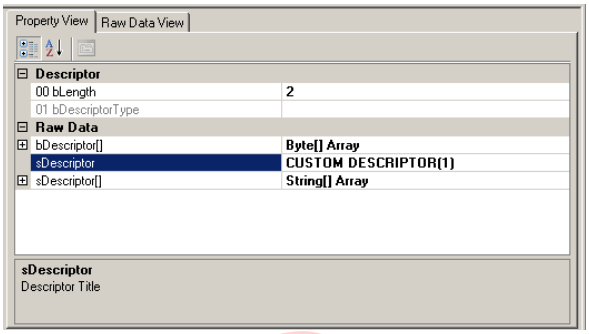
2.4.8 Creating Custom Descriptors

First select the descriptor in which the new descriptor shall be created. After clicking on “Add Custom Descriptor” via + in the icon menu bar, a customized descriptor will be created in the selected descriptor.

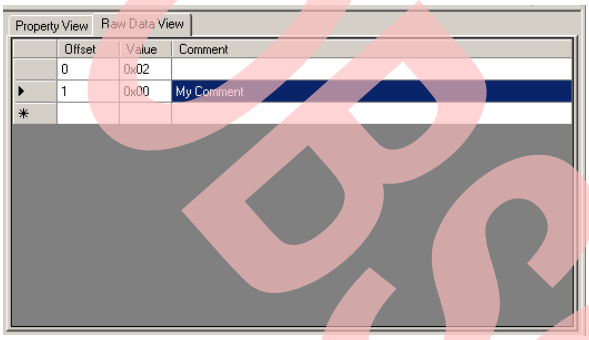


In the *Property View* the descriptor name can be specified in the field *sDescriptor*.

OBSOLETE



The descriptor values and comments can be set via *Raw Data View*.

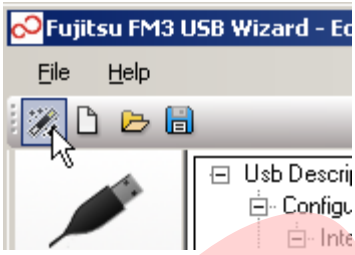


2.4.9 Create initial project of manually created settings

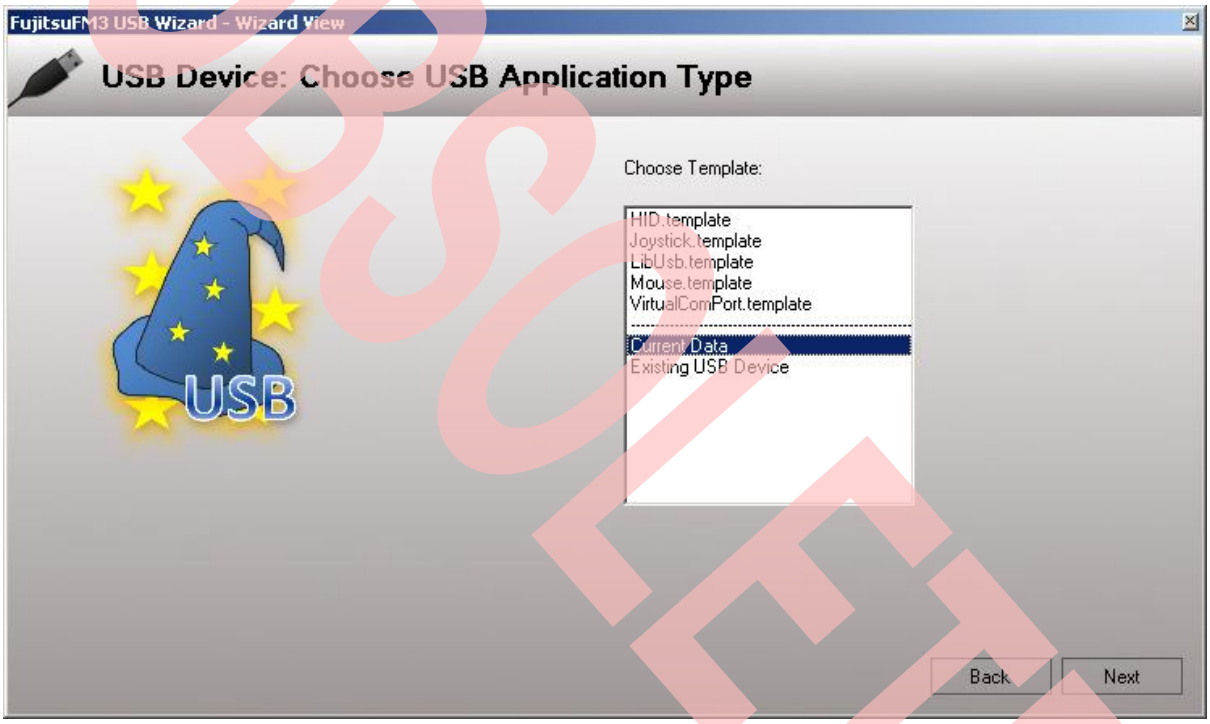
If the Assistant View was skipped and the Edit View was used to configure the USB device, normally only the files *UsbDescriptors.h*, *UsbClass.h* and *UsbClass.c* can be created.

OBSOLETE

These files normally will be placed into the UsbMiddleWare files directory. The second way is using the Wizard-View to create an initial project for a special platform. To use the current settings for the initial project, the Wizard View can be opened manually:



In Step 1 the entry Current Data has to be selected, to use the current settings from Edit View.

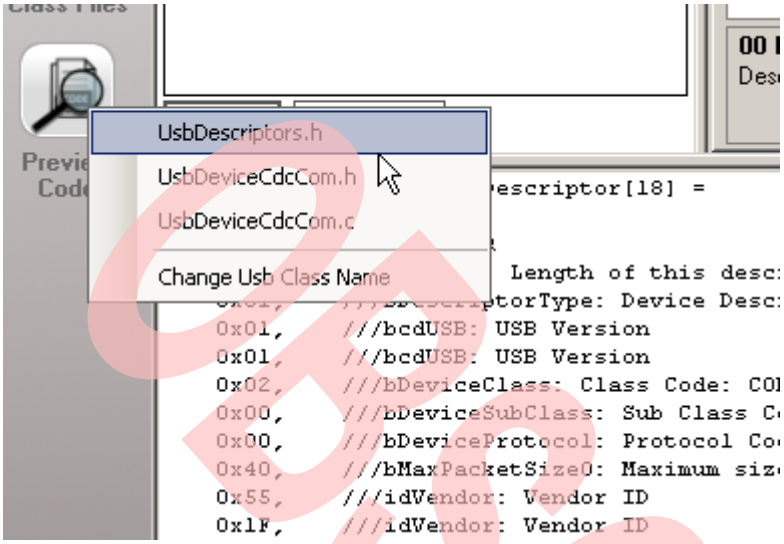


For all other steps see chapter 3.3.2.

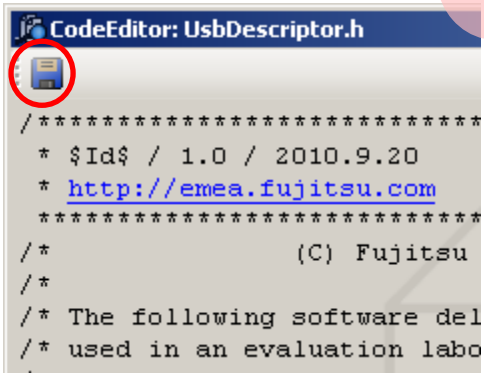
OBSOLETE

2.4.10 Code View

In the USB Descriptors the different features of the USB device are defined. With this information the *UsbDescriptors.h* file and the USB Class files will be created. To view them, they can be opened via Device Config->Preview Code-><Filetype>.



These files can also be saved manually. This can be done by the save button in the code viewer.



OBSOLETE

3 Usage (USB Assistant)

USAGE OF THE USB ASSISTANT

In this chapter different use cases will be described. This can be simple creating of embedded applications but also manipulating different USB descriptors manually.

3.1 Installation

The installation of the Fujitsu USB Assistant is quite simple.

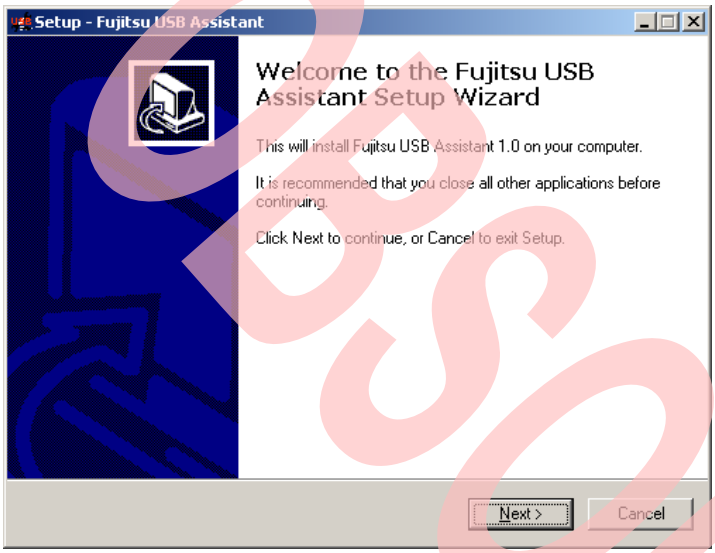


Figure 3-1: Step 1: Starting installation

While clicking on *Next* and selecting “I accept the agreement” the disclaimer will be accepted.

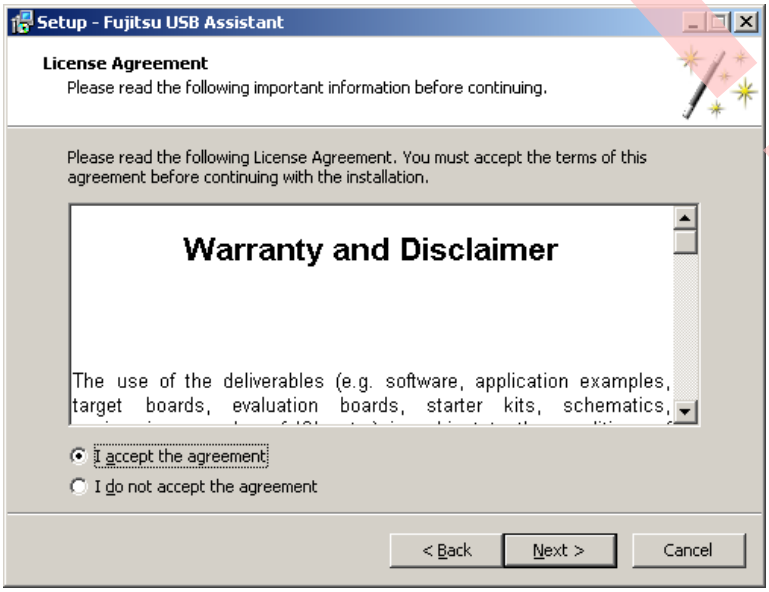


Figure 3-2: Accept Disclaimer

OBSOLETE

The Fujitsu USB Assistant can be placed on every directory instead of network drives! The USB Assistant requires the .NET Framework (>= 2.0) needed to be configured to run on network devices for security reasons (setting trusted zones).

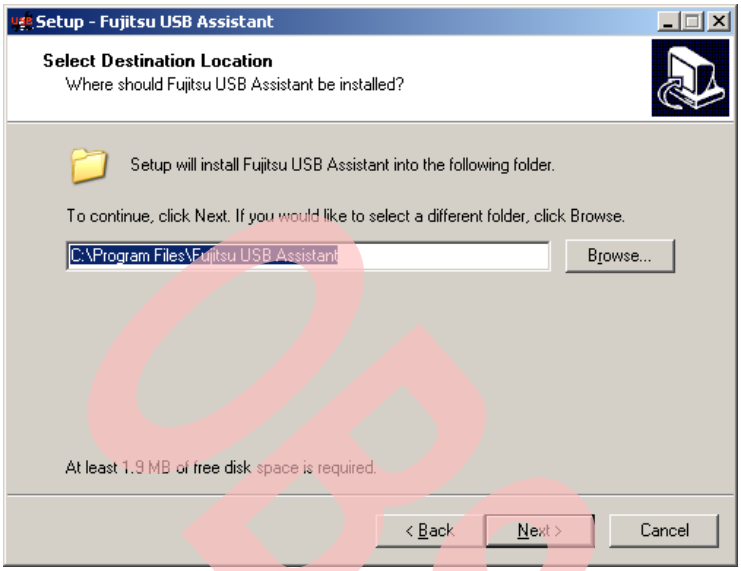


Figure 3-3: Choosing the installation directory

OBSOLETE

At this step the directory in the start menu can be defined.

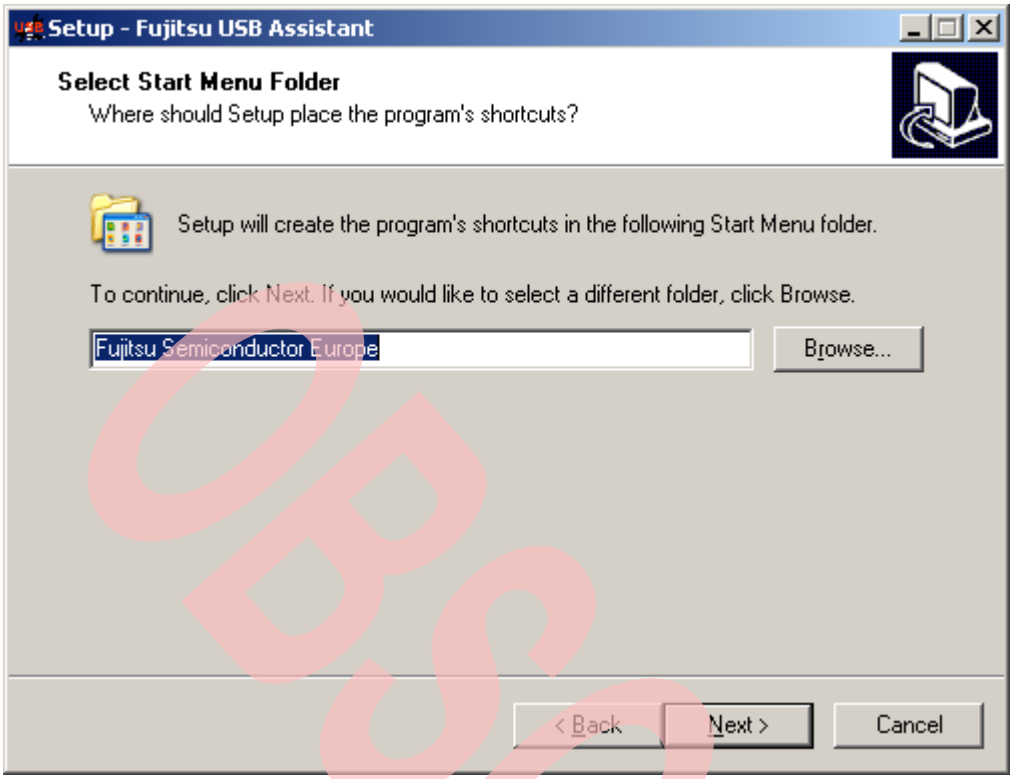


Figure 3-4: Creating start menu entry

Additional links on desktop or the Quick Launch can be also defined during the installation process.

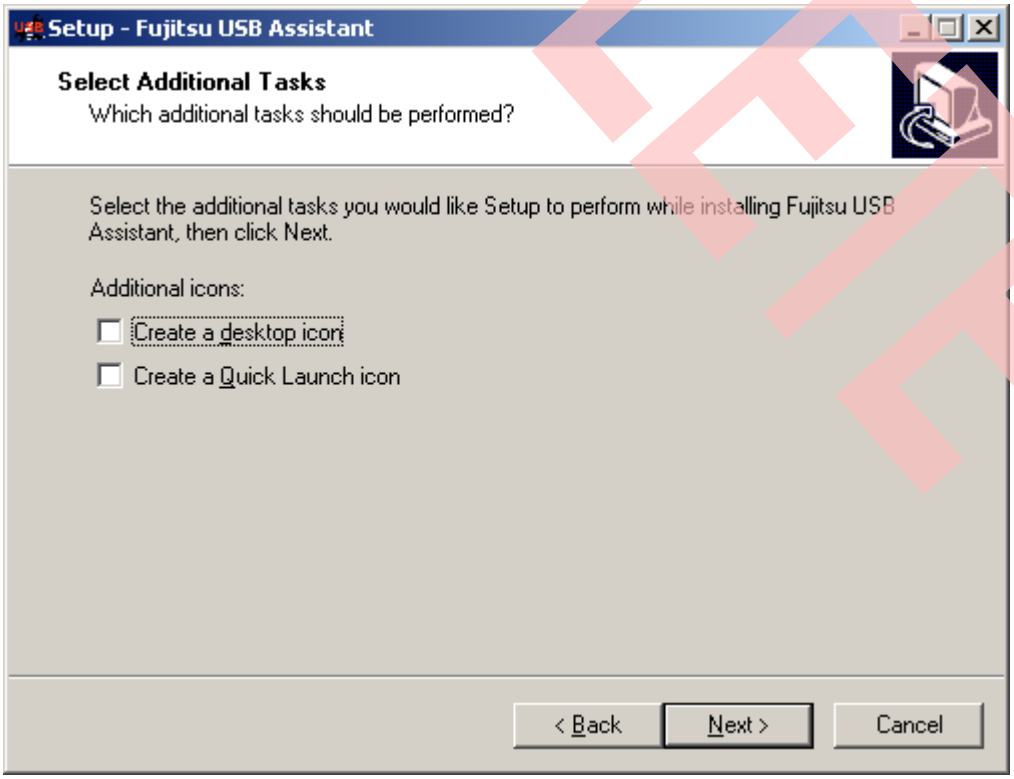


Figure 3-5: Adding additional links on desktop and Quick Launch

OBSOLETE

At the end a summery will be displayed and after clicking on *Install* the software will be installed.

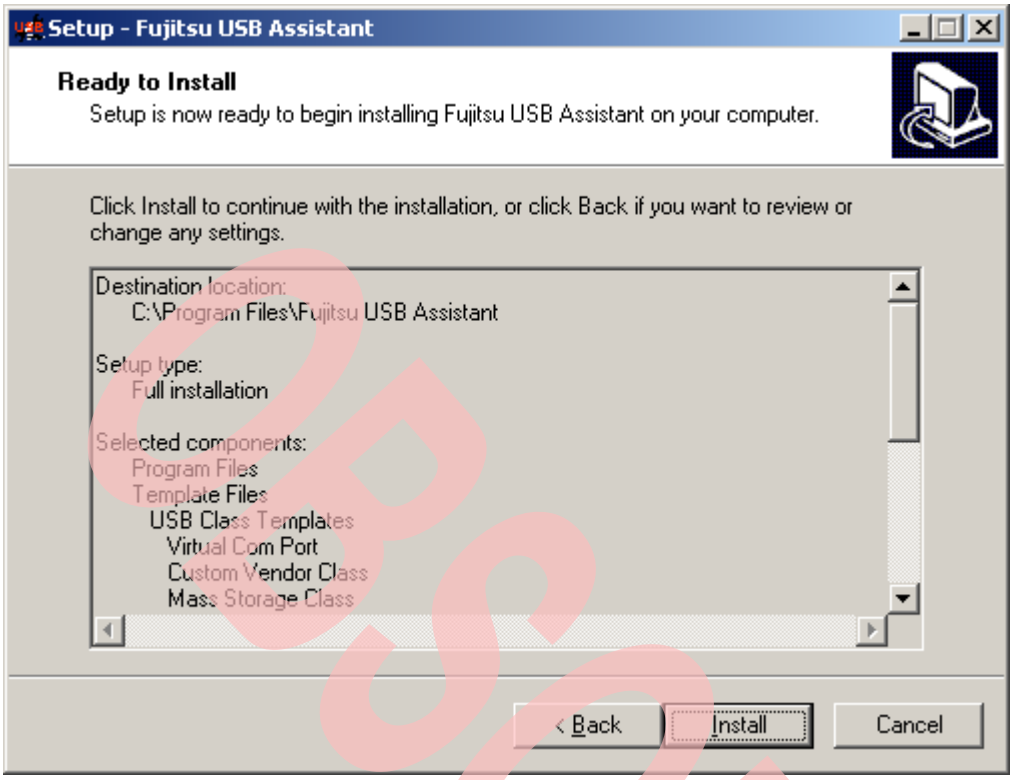


Figure 3-6: Installation Overview

OBSOLETE

3.2 Internet Updates

The USB Assistant is able to do Internet Updates. At first start the USB Assistant is asking for using Internet updates
:

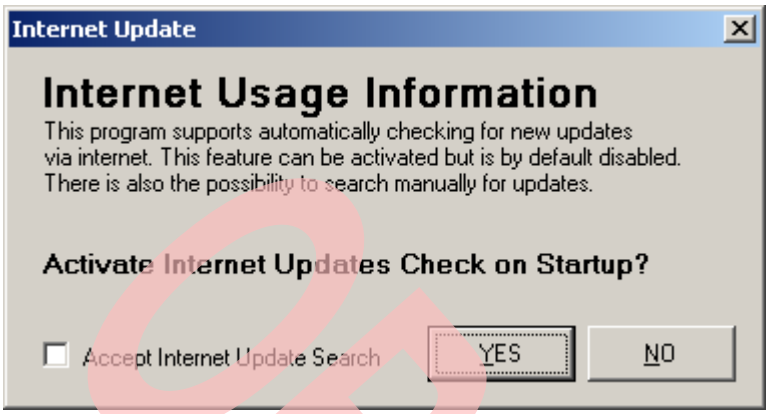


Figure 3-7: Internet Updates

Normally the USB Assistant is NOT using any internet connection. Only after selecting “Accept Internet Update Search” and pressing “YES” at the “Internet Usage Information” dialog, internet updates will be enabled.

If a new Update is available, following dialog will appear:

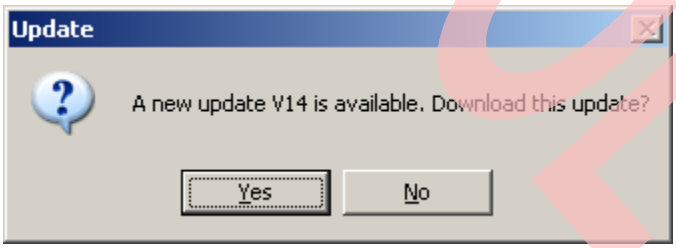


Figure 3-8: Internet Updates - Download

After pressing “Yes” the new installation will be downloaded and started.

OBSOLETE

3.3 First Start – Assistant View

At the first start the Fujitsu USB Assistant will show the Assistant View. The Assistant View can be used to create first time USB embedded applications or in later projects to add different USB functionalities to existing embedded applications.

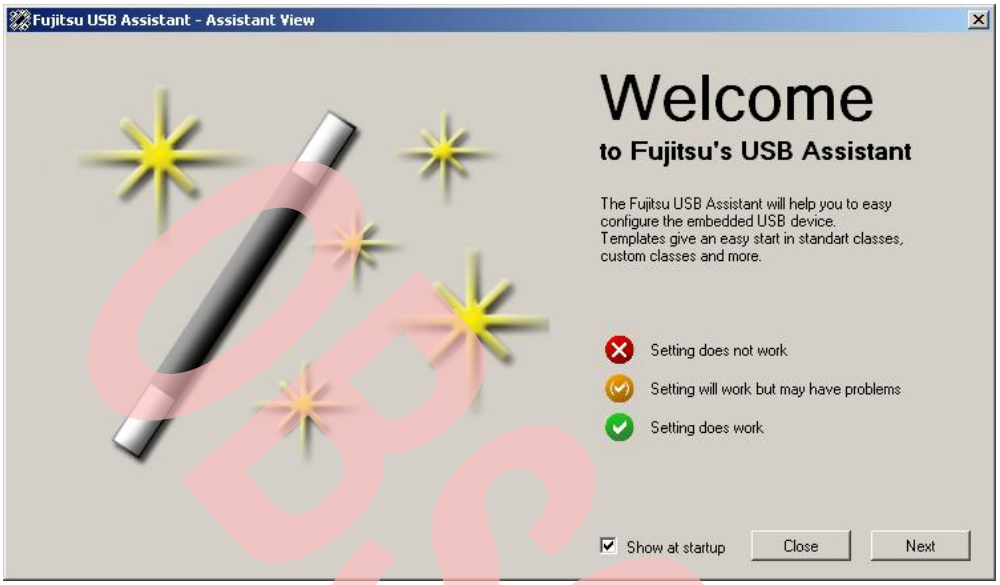


Figure 3-9: Fujitsu USB Assistant - Assistant View

The other view is the Edit View. With this view the complete USB descriptors can be edited in detail. After using the Assistant View all descriptors can be edited here, too.

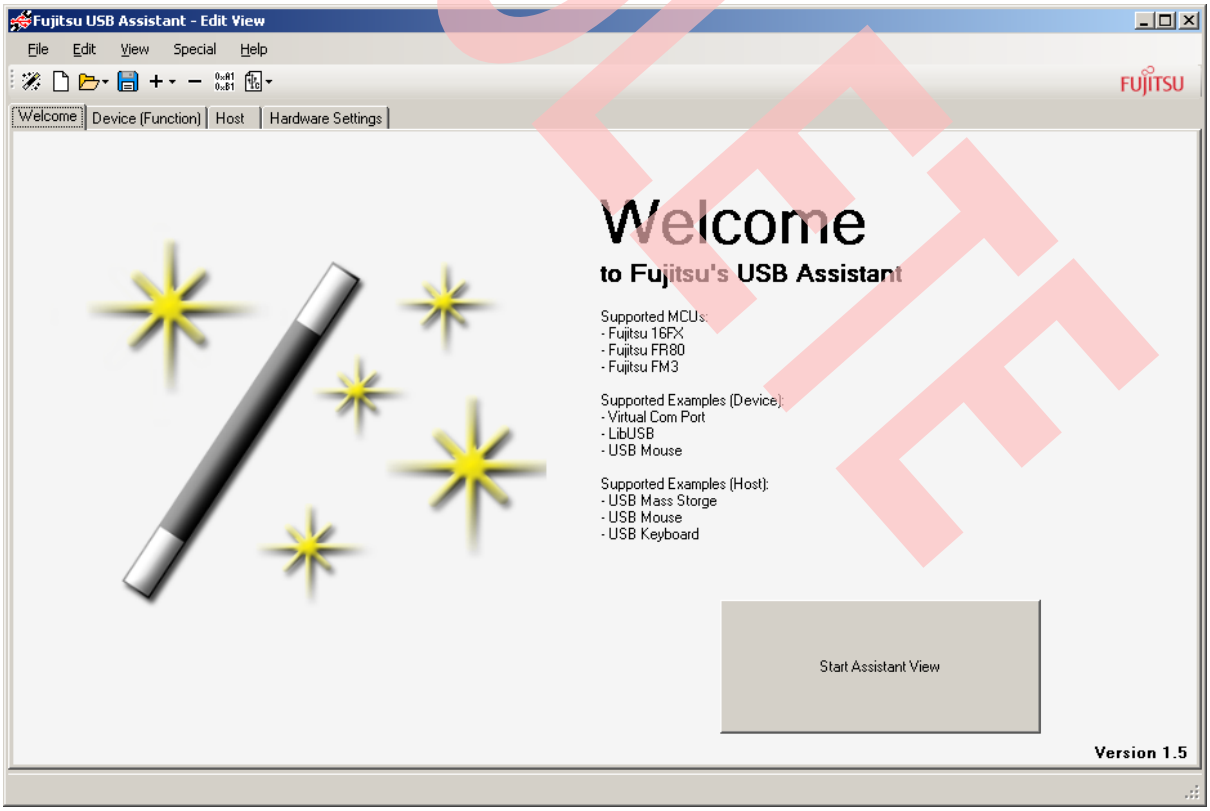


Figure 3-10: Fujitsu USB Assistant – Edit View

OBSOLETE

3.3.1 Step 1: Application Template

The Program starts normally in Assistant View. The Assistant implements two different templates combined in one: An application template and an MCU template. The application template represents the USB application, for example: Virtual Com Port. The MCU template represents the workspace settings for the specified MCU. In the first step the template selector will open:

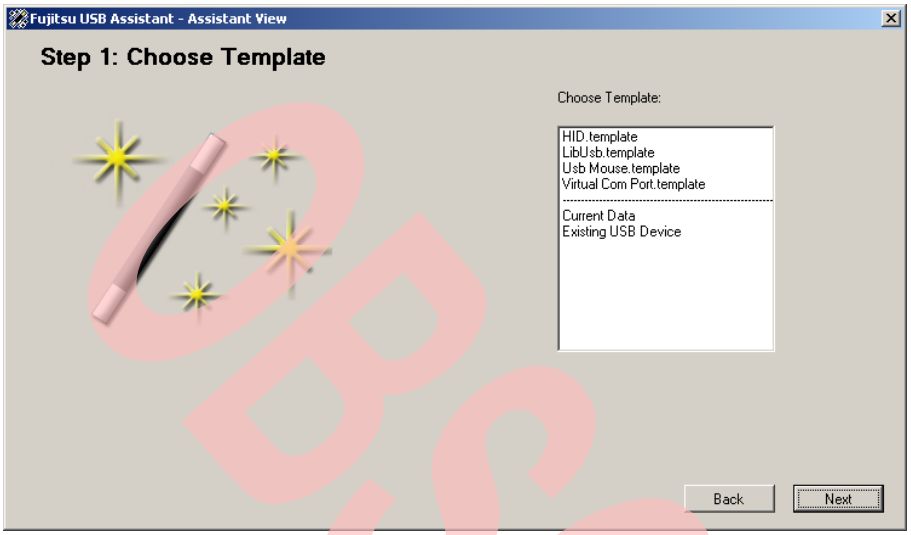


Figure 3-11: Assistant View - USB Application Template

At new template different existing templates can be chosen, but also an existing device.

Existing USB Device: This option can be used to read in all descriptor files from an existing device. For starting with developing USB devices this helps to adapt settings. The protocol part is left open! The user should also change the Vendor ID and Product ID to his specific IDs.

Current Data: This option is used if the USB Assistant is called manually after setting all descriptors in Edit View. This can be used to create an initial USB template with own settings.

3.3.2 Step 2: Device Descriptor Settings

The next step is used to set general settings like Vendor ID, Product ID, etc. The assistant will help to configure well known settings. Warnings are orange and errors are red.

OBSOLETE

Figure 3-12: Assistant View - Device Descriptor

3.3.3 Step 3: Power Management

Step 3 helps configurating the power management and remote wakeup features.

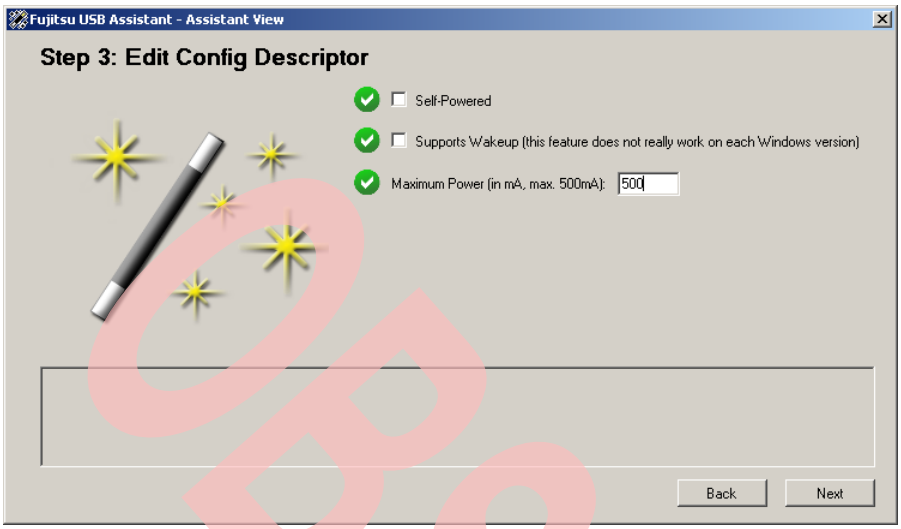


Figure 3-13: Assistant View – Configuration Descriptor

3.3.4 Step 4: MCU Template

In this step the type of environment is selected. This can depend on the MCU but can also depend on an evaluation board (Starterkit). If a starterkit is selected, the assistant will automatically choose hardware settings for the USB peripheral. If the “Add Board Support” option was chosen, the assistant will also add some human interface to act with (buttons, LEDs, UART).

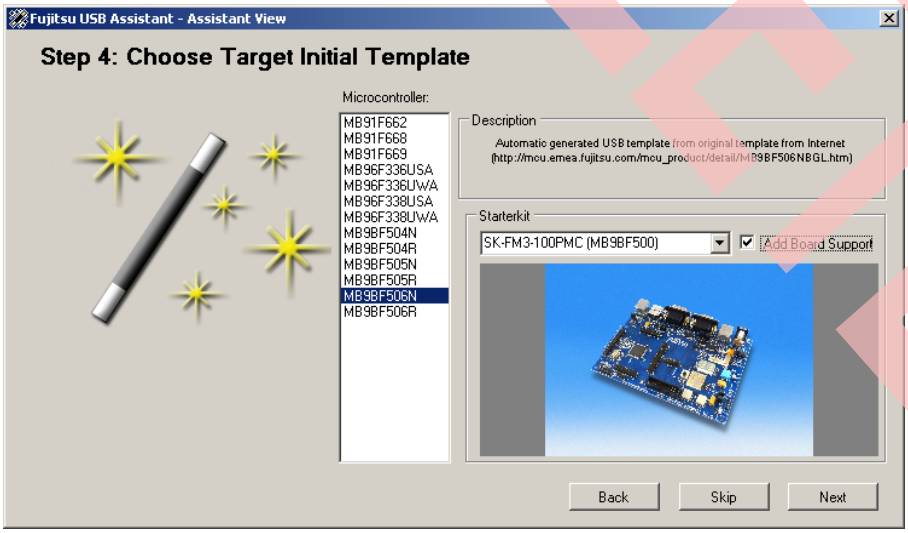


Figure 3-14: Assistant View – MCU Template

OBSOLETE

Step 5: Project Naming

In this window the project name and location is set. For 16FX and FR80 MCUs always a Softune Workbench workspace and project is created with the given project and workspace name. After clicking on *Next* all files will be copied and patched.

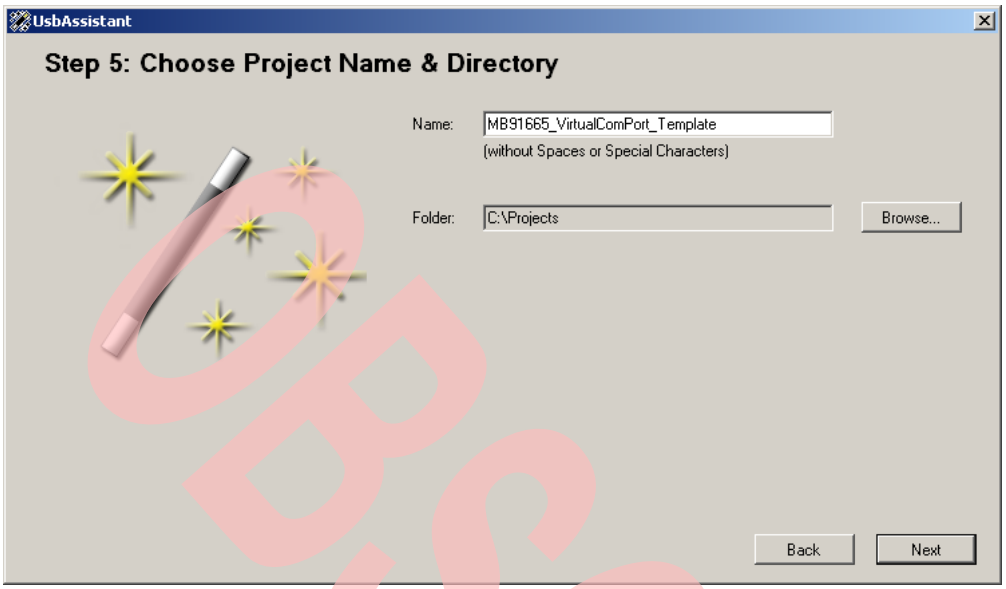


Figure 3-15: Assistant View – Project Name

OBSOLETE

3.3.5 Last Steps

Now the configuration process is finished and after clicking *Finish* the program will show the Edit View. The complete project was now created on the specified location. The Edit View will show all configured options. To update the descriptors, the project only has to be saved. To recreate and overwrite the USB class file template, the Code Creation button in Edit View can be used (see also chapter 3.4, button (4)).

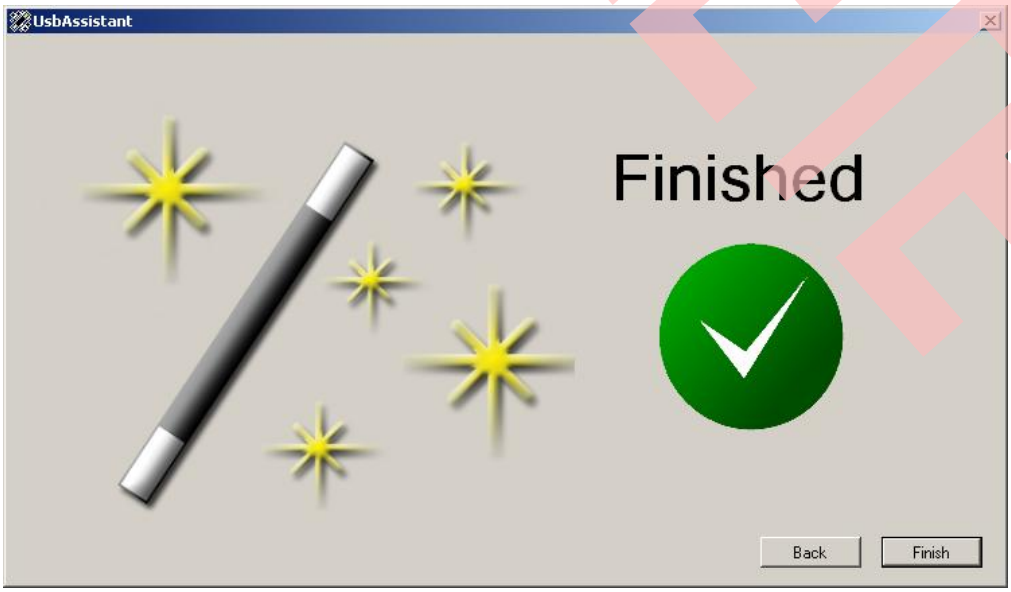
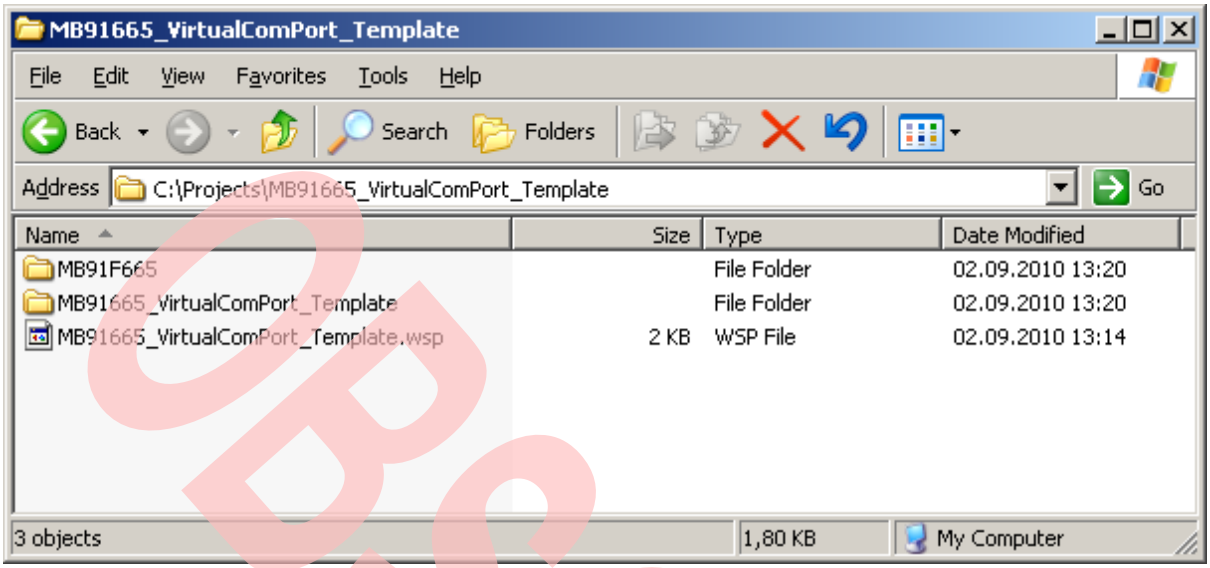


Figure 3-16: Assistant View – Finished

3.3.6 Open Result in Softune Workbench

The generated result described before can be now used in Softune Workbench. The example here was created as a Virtual Com Port with a MB91F665 series MCU.



The *.wsp file represents the Softune Workbench workspace file.

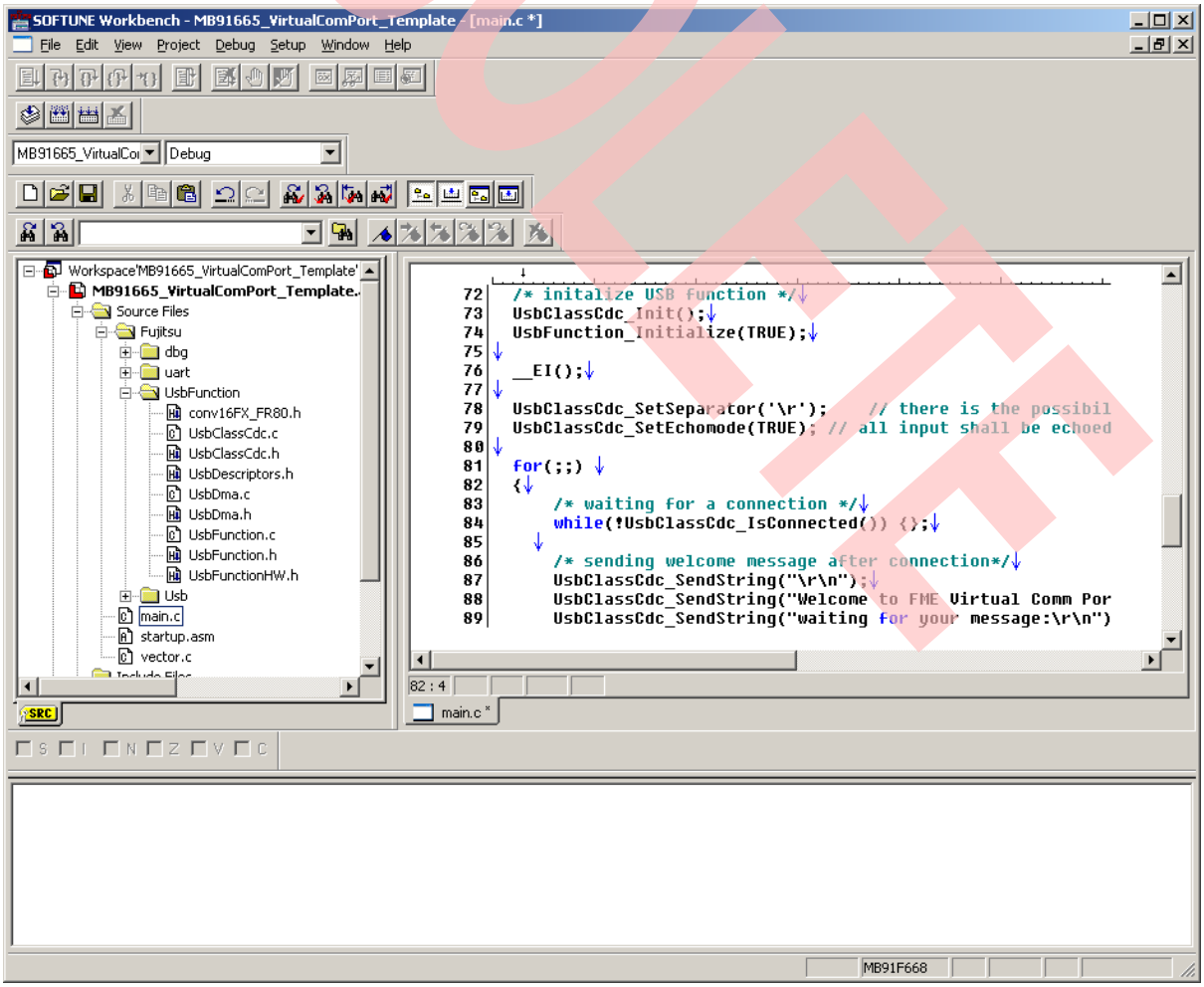


Figure 3-17: Softune Workbench

OBSOLETE

Following code can be added in the main module, to realize a simple loopback interface:

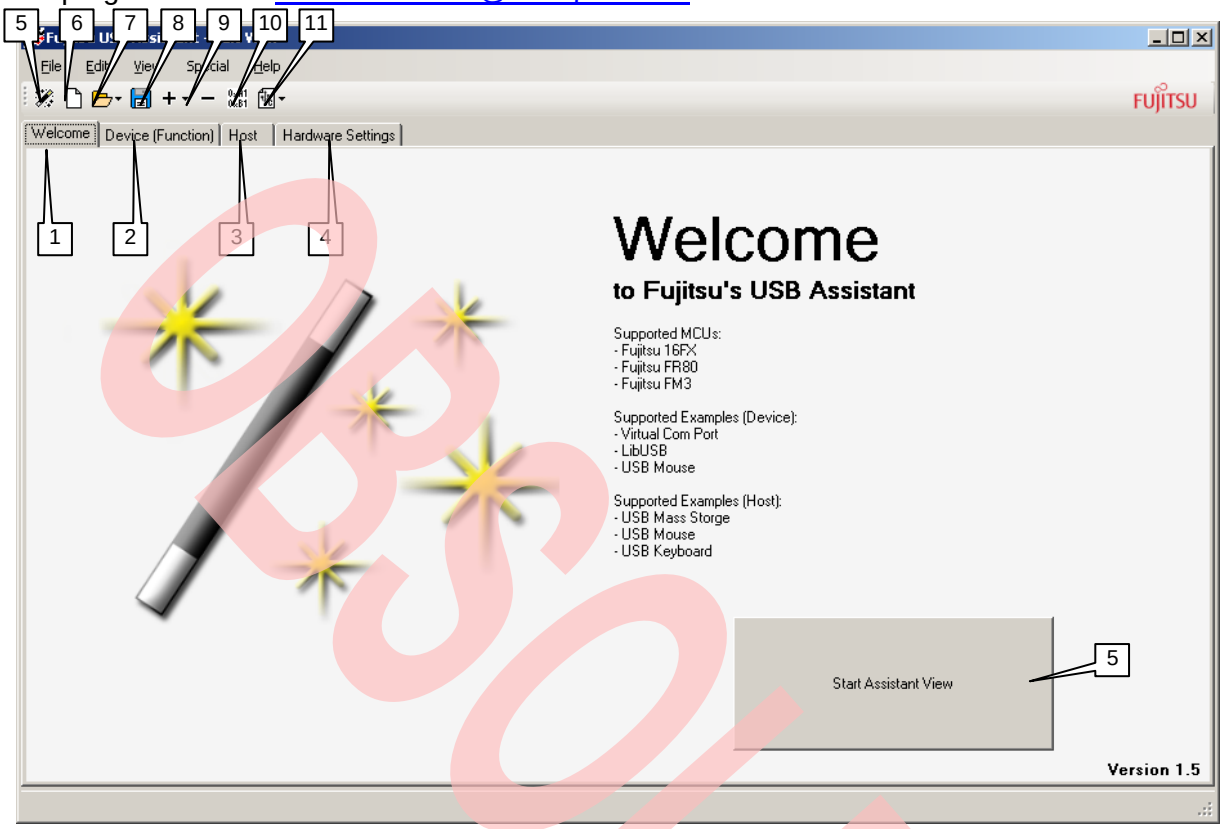
```
UsbClassCdc_SetSeparator('\r'); // set seperator
UsbClassCdc_SetEchomode(TRUE);  // all input shall be echoed

for(;;)
{
  /* waiting for a connection */
  while(!UsbClassCdc_IsConnected()) {};

  /* sending welcome message after connection*/
  UsbClassCdc SendString("\r\n");
  UsbClassCdc SendString("Welcome to FME Virtual Comm Port Example!\r\n");
  UsbClassCdc_SendString("waiting for your message:\r\n");

  while(UsbClassCdc IsConnected())
  {
      if (UsbClassCdc ReceivedLength() > 0) {
          /* receive data, clears also the receive buffer */
          receiveSize = UsbClassCdc_ReceiveBuffer((uint8_t *)receiveBuffer);
          receiveBuffer[receiveSize] = '\0';    //adding zero termination to string

          /* print out receiveBuffer through Virtual Comm Port */
          UsbClassCdc SendByte('\n');
          UsbClassCdc_SendString("Received String: ");
          UsbClassCdc_SendString(receiveBuffer);
          UsbClassCdc_SendString("\r\n");
      }
  }
}
```

Figure 3-18: Virtual Com Port Loop Device

OBSOLETE

3.4 Edit View

The Edit View is used to create descriptors manually. In this mode the most complex descriptors can be created. If there is some information not configurable or missing, contact the programmer via mcu_ticket.fseu@de.fujitsu.com.

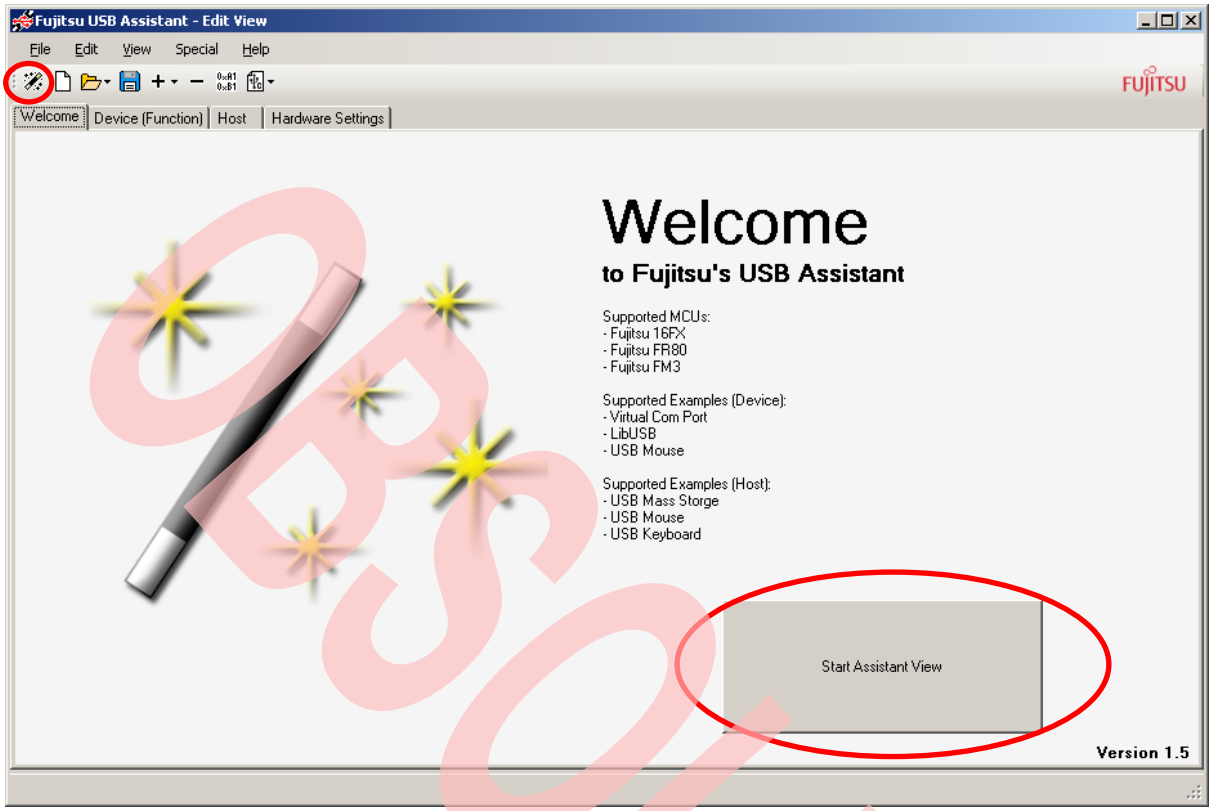


1) Welcome Screen
2) Device Mode Settings
3) Host Mode Settings
4) Hardware Specific Settings
5) Open Assistant View
6) Create Empty Configuration
7) Open Configuration
8) Save Configuration
9) Add Remove Descriptor in Device Mode
10) View RAW Descriptor in Device Mode
11) View Usb Device Files

OBSOLETE

3.4.1 Welcome Screen

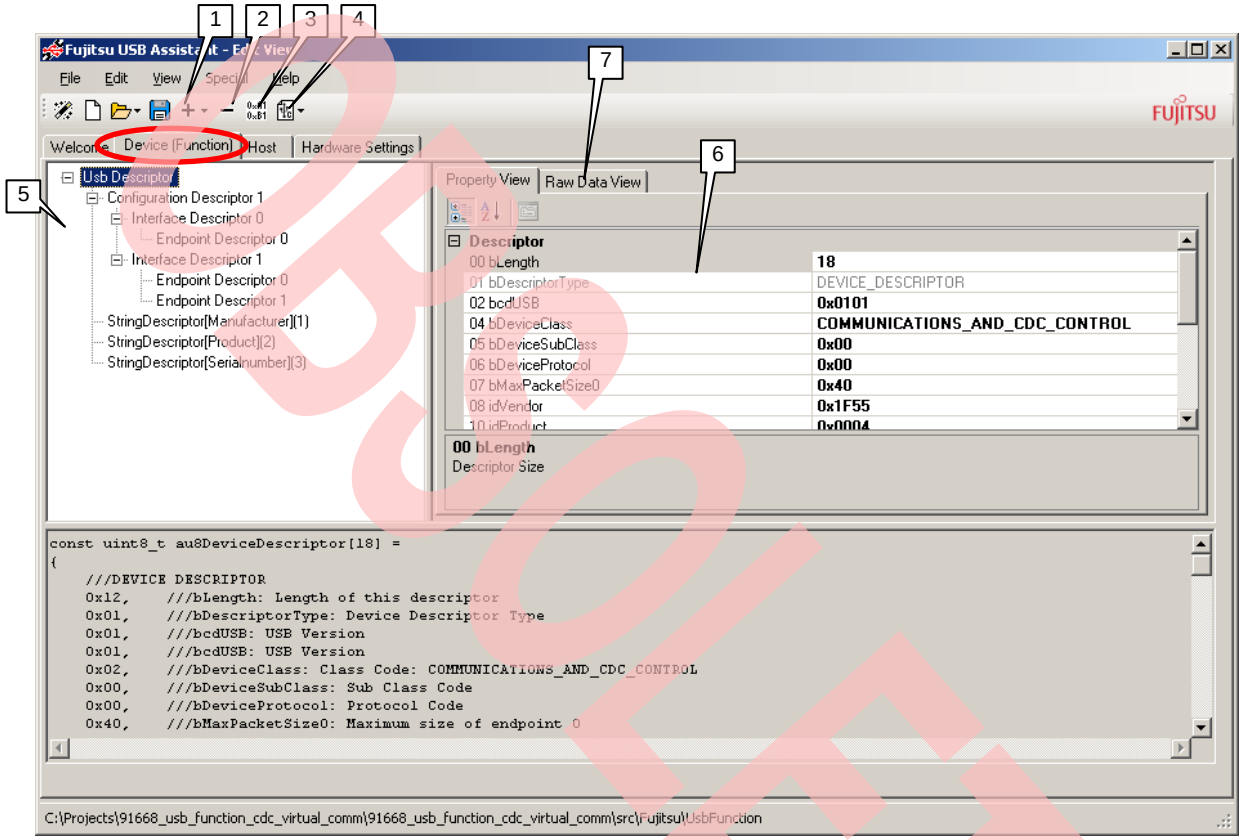
As first start, the Assistant View should be used. The Assistant View can be started via pressing “Start Assistant View”.



OBSOLETE

3.4.2 Device Mode Settings

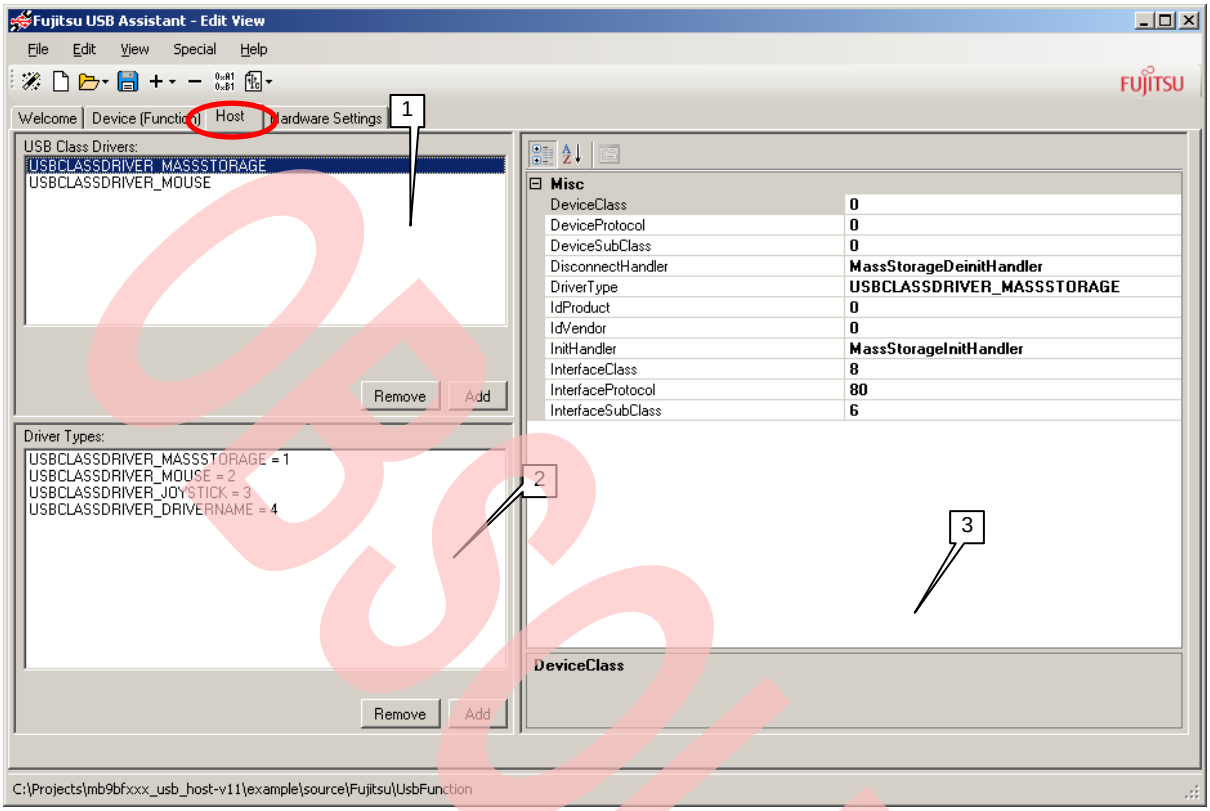
This view can be used to manually changing USB descriptor data. (1) is used to add a descriptor. For adding a descriptor the parent descriptor has to be selected in which the new descriptor shall be created. By clicking on “Usb Descriptor” the root descriptor will be selected and the descriptor data will be shown in (6). With (2) a selected descriptor can be removed. (3) shows the descriptors in RAW data. (4) shows the files *UsbDescriptors.h*, *UsbClass.c* and *UsbClass.h*. (5) shows the USB descriptor tree. (6) shows the USB descriptor data. With (7) it can be switched in a RAW data mode.



OBSOLETE

3.4.3 Host Mode Settings

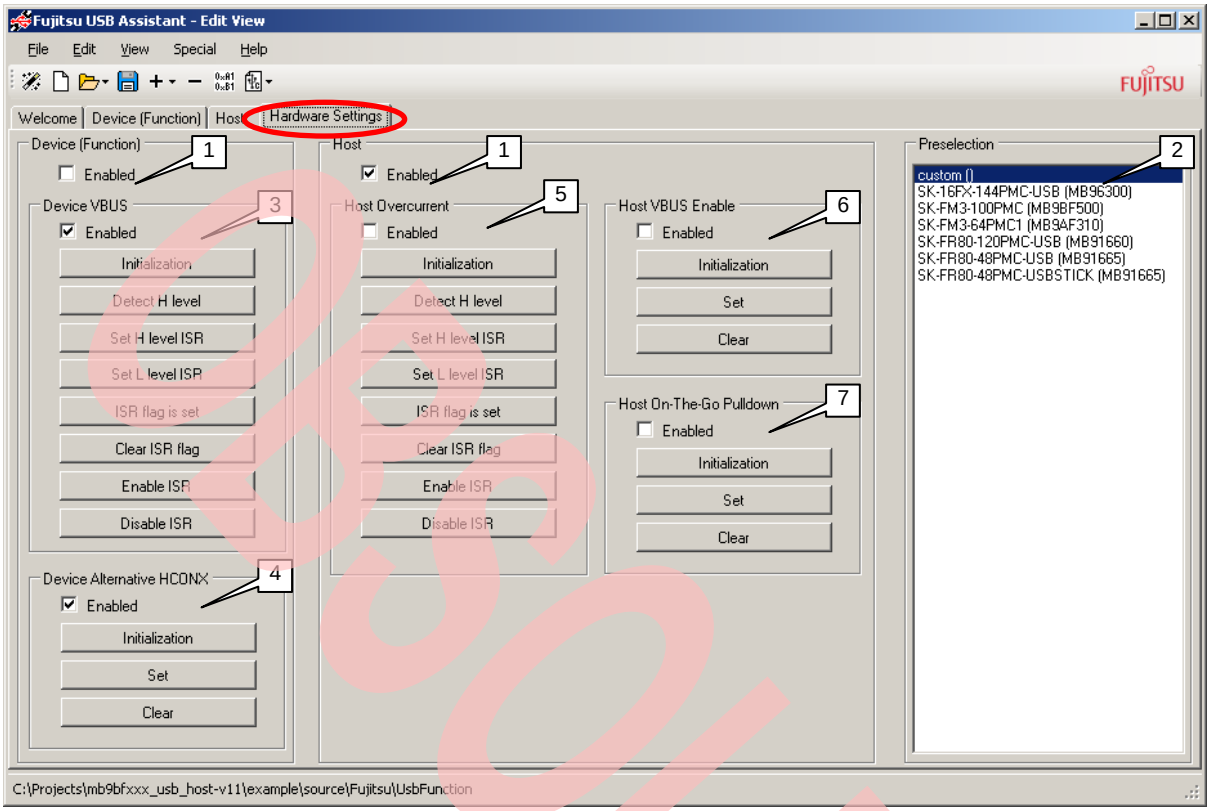
For USB Host the different virtual driver modules can be configured. (1) describes the different devices which can be recognized. (2) describes the different driver types. (3) is used to change matching features of the selected device driver (1).



OBSOLETE

3.4.4 Hardware Settings

In this panel different hardware settings can be configured. USB Device / USB Host can be enabled or disabled (1). For different starterkits / evaluation boards a preselection of hardware settings can be done (2).



3) The VBUS signal detection settings can be chosen.

4) If devices do not have a special port for rising the Full-Speed pull-up, a port-pin can be specified for this.

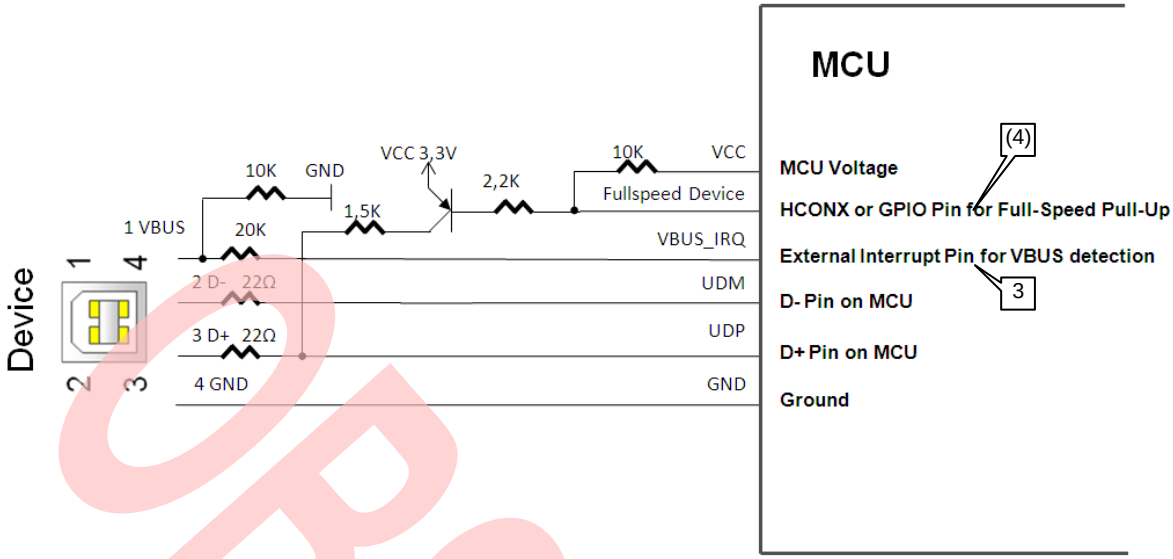
5) In Host Mode often ICs are used to switch the VBUS line on or off and recognize overcurrent. For overcurrent detection the signal detection settings can be chosen.

6) In Host Mode often ICs are used to switch the VBUS line on or off and recognize overcurrent. To enable VBUS line settings can be configured here.

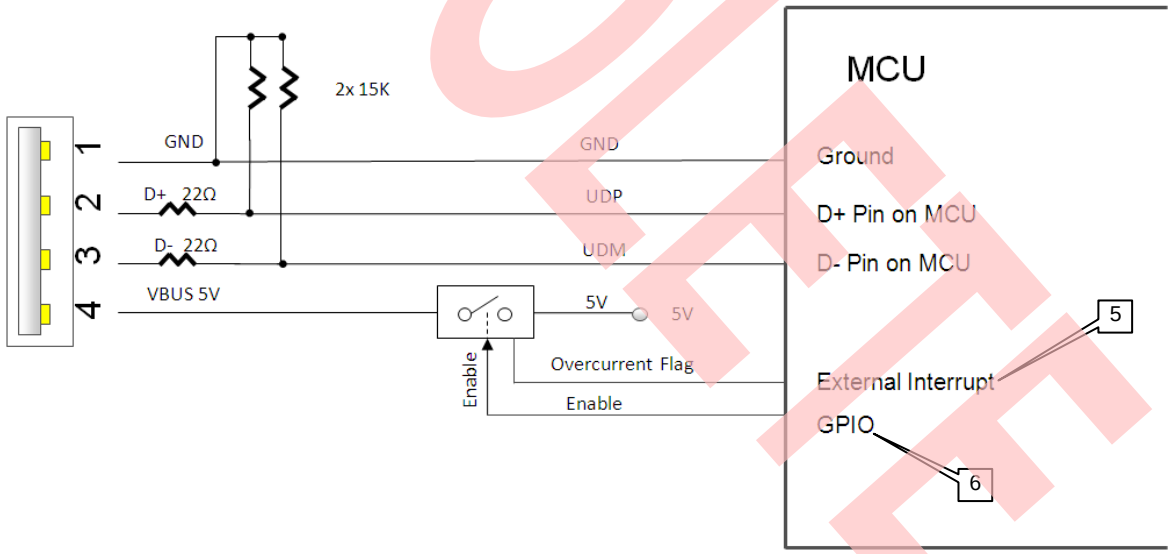
7) If the Host supports USB OTG or switches between Host / Device mode, this setting can be used.

preliminary

3.4.4.1 Example: USB Device

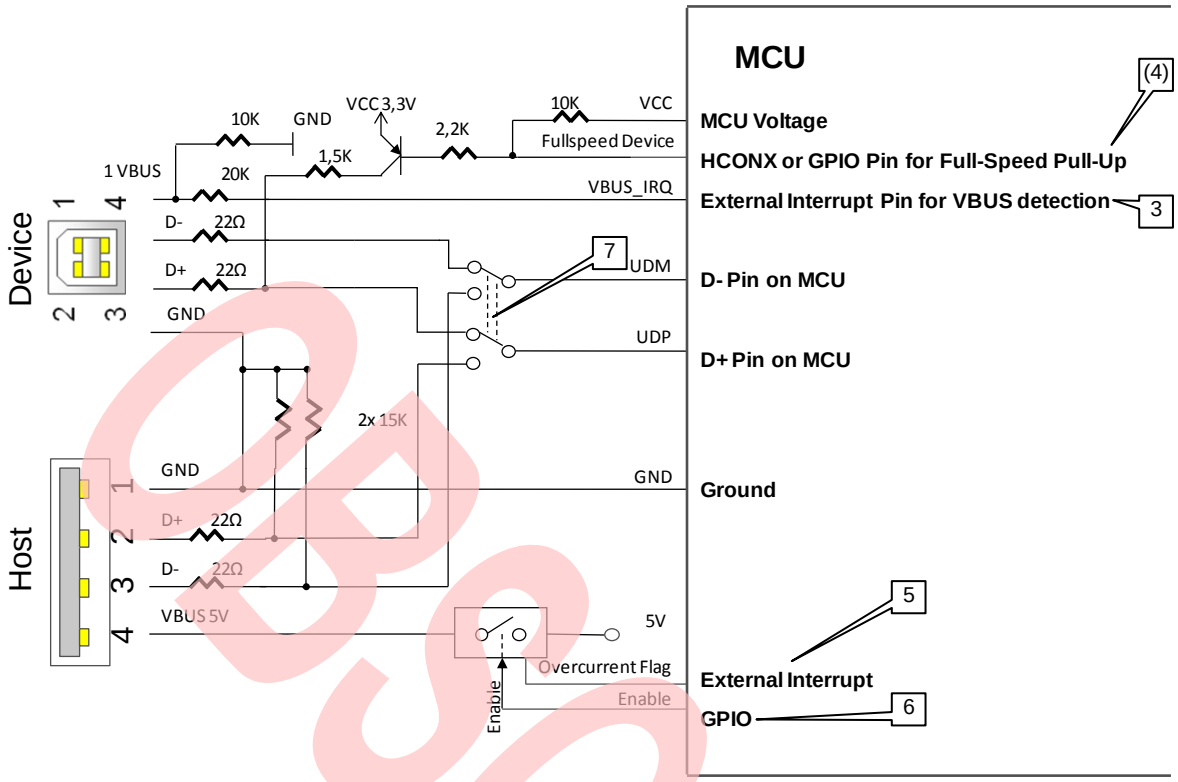


3.4.4.2 Example: USB Host

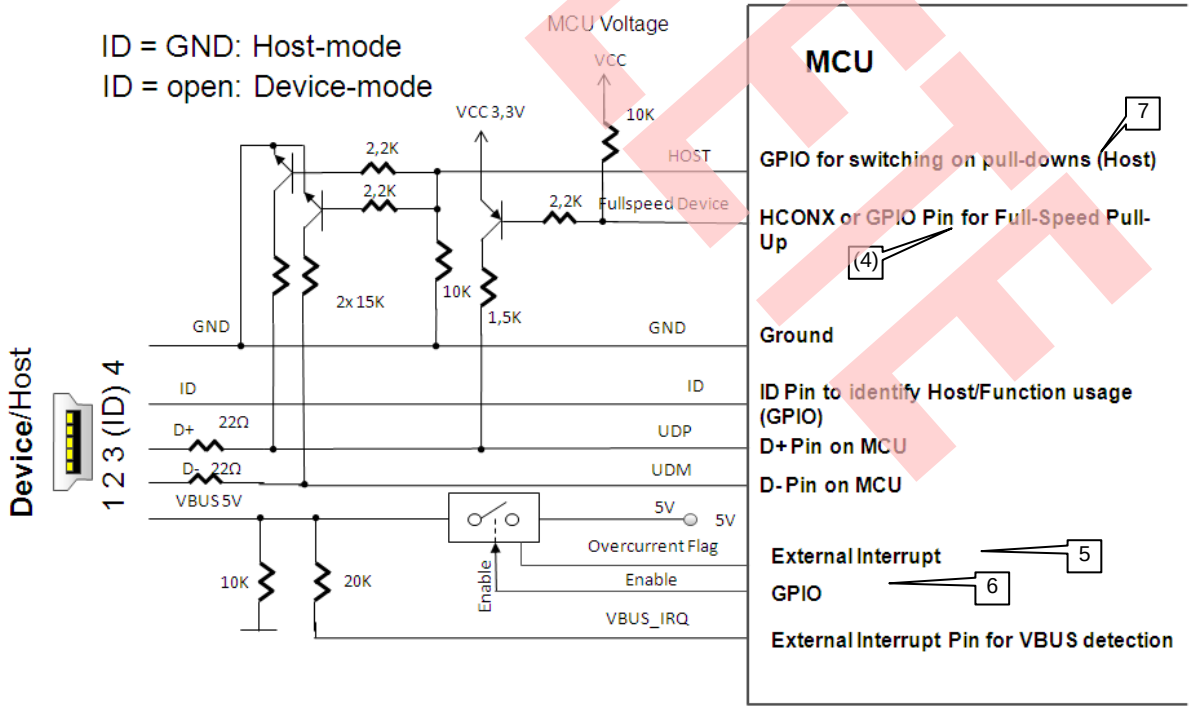


OBSOLETE

3.4.4.3 Example: USB On-Demand

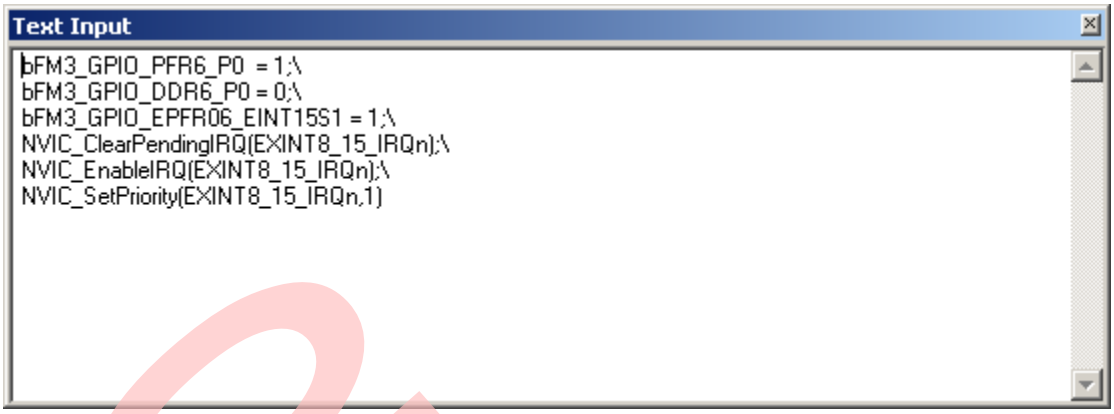


3.4.4.4 Example: USB On-The-Go



OBSOLETE

3.4.4.5 Example VBUS Detection (Device) - Initialization Routine (for FM3)



3.4.4.6 Example VBUS Detection (Device) - Detect H level (for FM3)

3.4.4.7 Example VBUS Detection (Device) - Set H level ISR (for FM3)

OBSOLETE

3.4.4.8 Example VBUS Detection (Device) - Set L level ISR (for FM3)

3.4.4.9 Example VBUS Detection (Device) – ISR flag is set (for FM3)

3.4.4.10 Example VBUS Detection (Device) – Clear ISR flag (for FM3)

OBSOLETE

3.4.4.11 Example VBUS Detection (Device) – Enable ISR (for FM3)

3.4.4.12 Example VBUS Detection (Device) – Disable ISR (for FM3)

3.4.5 Open / Load USB Configurations

The Fujitsu USB Assistant can load the *UsbDescriptors.h* file located in the UsbFunction directory, from a Softune Workbench Project or an IAR Workbench Project.

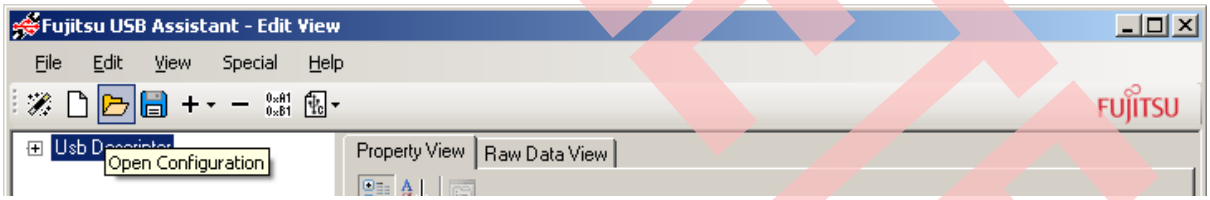


Figure 3-19: Edit View - Open File

Following screenshots are showing where to find the *UsbDescriptors.h*:

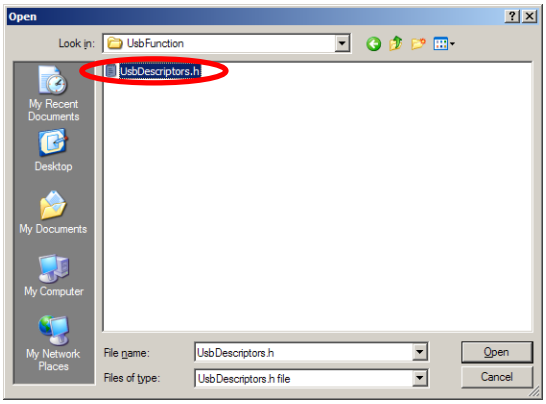


Figure 3-20: Open File Dialog

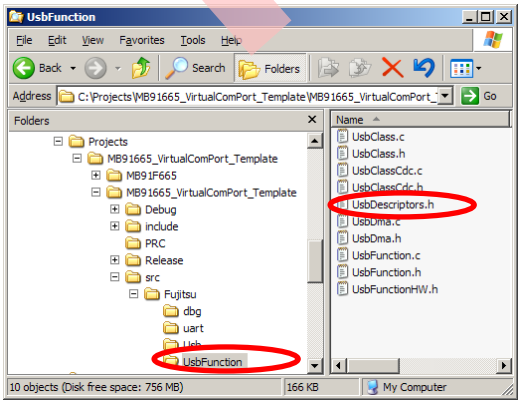
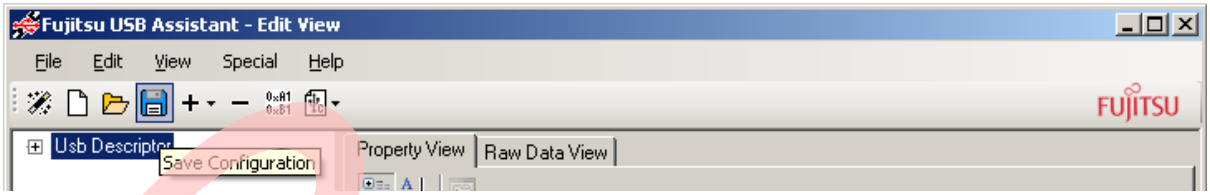


Figure 3-21: Location of UsbDescriptors.h

OBSOLETE

3.4.6 Save USB Configurations

The only file which will be saved as configuration is the *UsbDescriptors.h* file. Before a configuration can be saved, at least a configuration descriptor containing at minimum one interface descriptor has to exist in the configuration.



Following screenshot is showing where to find the *UsbDescriptors.h*:

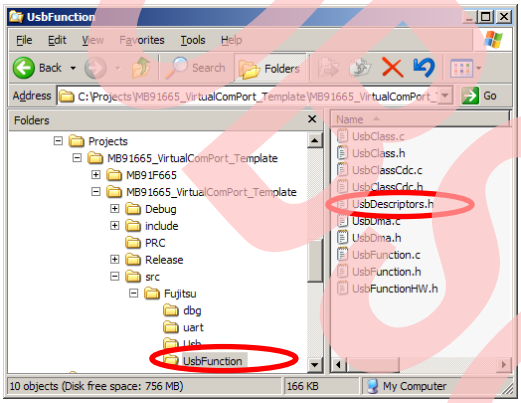
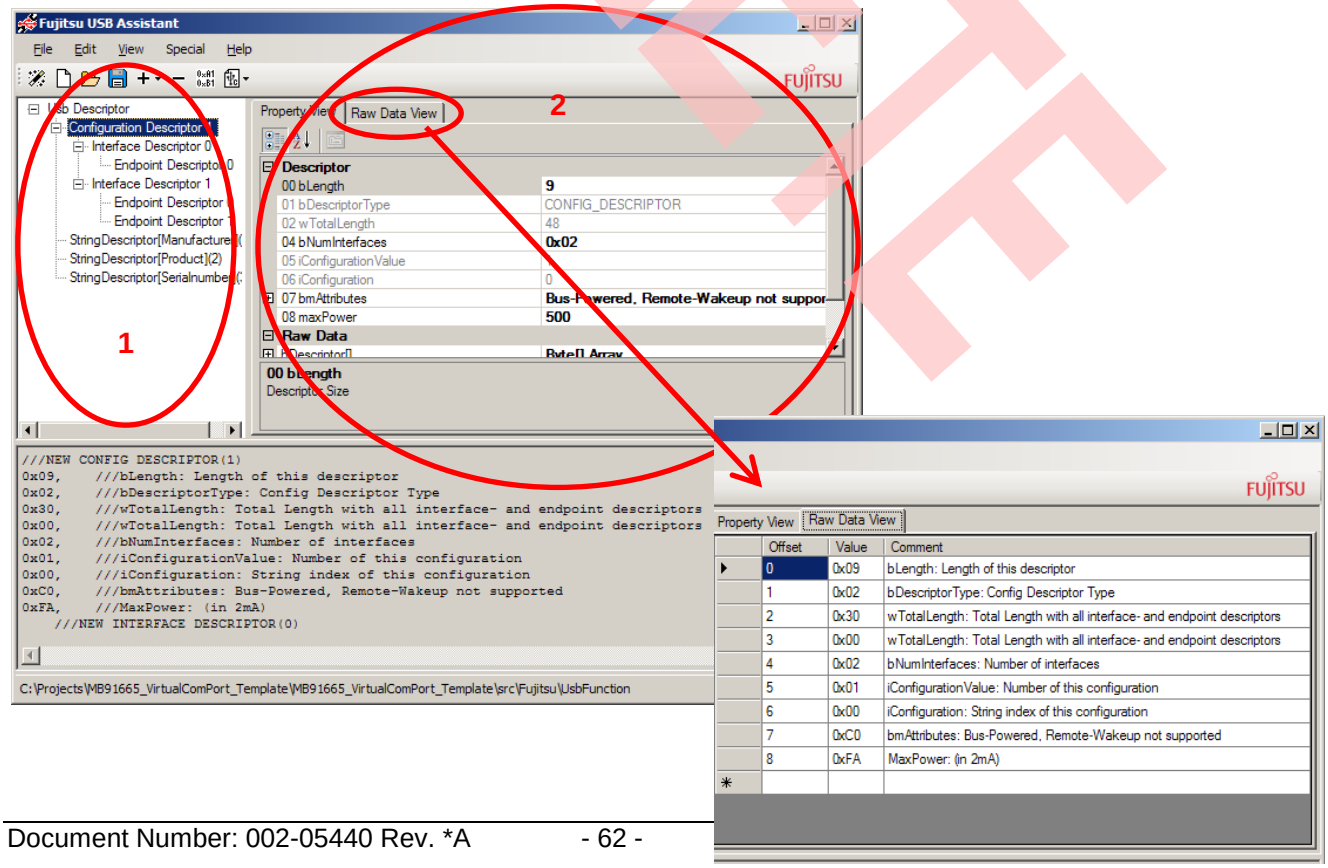


Figure 3-22: Location of UsbDescriptors.h

3.4.7 Edit Descriptors

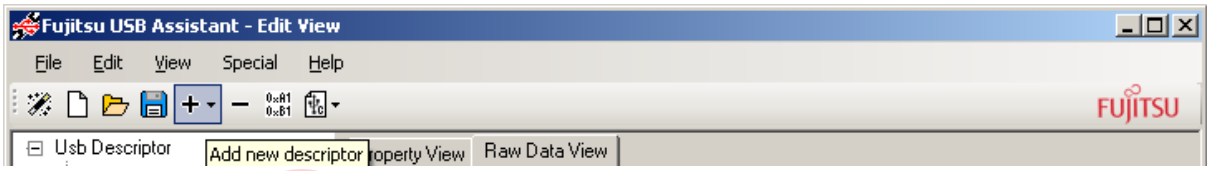
First a descriptor must be selected (1), before the different values can be edited in the Property View (2). It is possible to edit the descriptor also in raw data view (3).



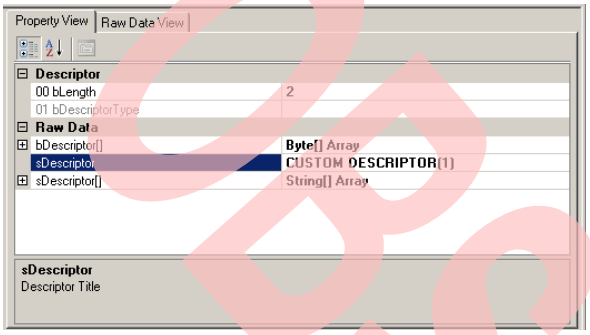
OBSOLETE

3.4.8 Creating Custom Descriptors

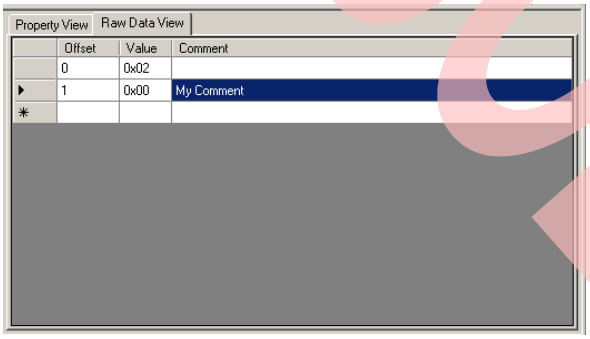
First select the descriptor in which the new descriptor shall be created. After clicking on “Add Custom Descriptor” via + in the icon menu bar, a customized descriptor will be created in the selected descriptor.



In the *Property View* the descriptor name can be specified in the field *sDescriptor*.

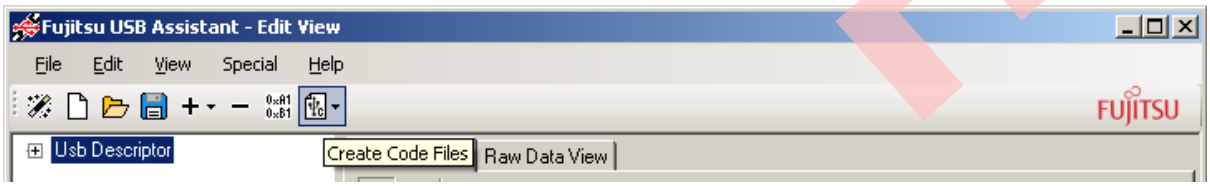


The descriptor values and comments can be set via *Raw Data View*.



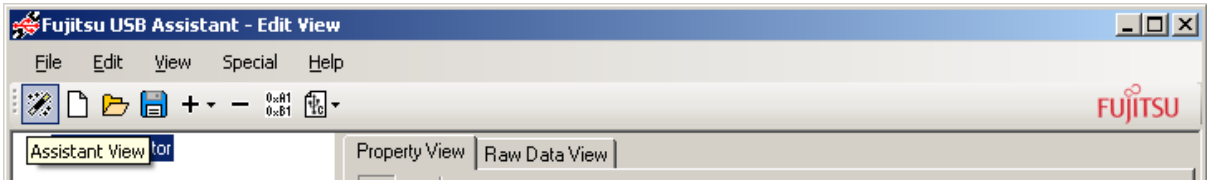
3.4.9 Create initial project of manually created settings

If the Assistant View was skipped and the Edit View was used to configure the USB device, normally only the files *UsbDescriptors.h*, *UsbClass.h* and *UsbClass.c* can be created.

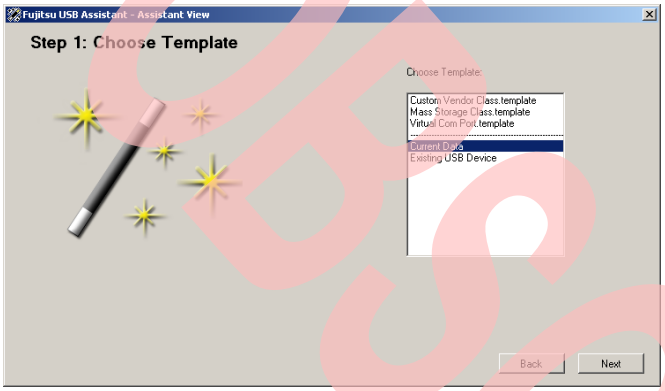


OBSOLETE

These files normally will be placed into the UsbFunction files directory. The second way is using the Assistant-View to create an initial project for a special platform. To use the current settings for the initial project, the Assistant View can be opened manually:



In Step 1 the entry Current Data has to be selected, to use the current settings from Edit View.

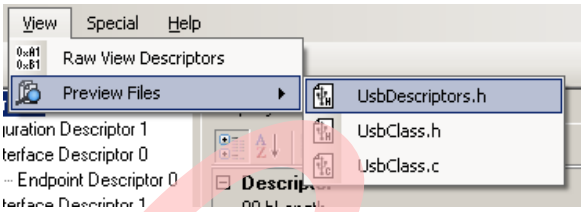


For all other steps see chapter 3.3.2.

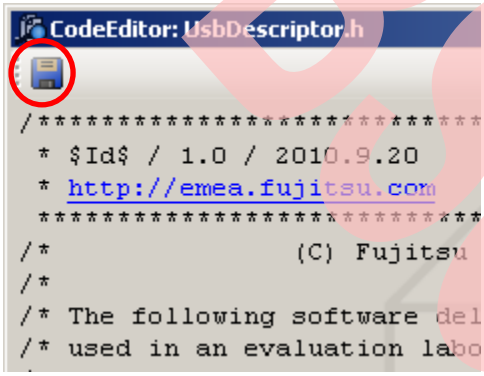
OBSOLETE

3.4.10 Code View

In the USB Descriptors the different features of the USB device are defined. With this information the *UsbDescriptors.h* file and the USB Class files will be created. To view them, they can be opened via View->Preview Files-><Filetype>.



These files can also be saved manually. This can be done by the save button in the code viewer.



OBSOLETE

4 Detailed Information

DETAILED INFORMATION OF FILEFORMATS AND PLUG-INS

This section will describe settings, plug-ins and file formats the Fujitsu USB Assistant is supporting.

4.1 Configuration File

The configuration file is the *UsbDescriptors.h*. Following settings will be read in (stored):

- **USB Function Files Directory**: The directory is the same of the UsbDescriptors.h file. So the *UsbDescriptors.h* file has to be placed in the USB Function files directory.

- **USB Class Name**: This information is stored in a define named `CLASSNAME`:

```
#define CLASSNAME "UsbClassCdc"
```

- **USB Device Descriptor**: The Descriptor is stored in a const `unsigned char` array:

```
#define USB_DEVDESC_SIZE 18
const uint8_t au8DeviceDescriptor[18] = {};
```

- **USB Configuration Descriptor**: This Descriptor contains also interface and endpoint descriptors. This file will be stored in following `unsigned char` array:

```
#define USB_CNFGDESC_SIZE 48
      const uint8_t au8ConfigDescriptor[48] = {};
```

- **String Descriptors**: These descriptors are saved as `pcManufacturerStringDescriptor`, `pcProductStringDescriptor`, and `pcSerialnumberStringDescriptor`:

```
const char_t* pcManufacturerStringDescriptor = "Fujitsu";
const char_t* pcProductStringDescriptor = "Virtual Comm Port";
const char_t* pcSerialnumberStringDescriptor = "1.0";

const char_t** ppStringDescriptors[3] =
{
      &pcManufacturerStringDescriptor,
      &pcProductStringDescriptor,
      &pcSerialnumberStringDescriptor
};
```

OBSOLETE

4.2 USB Application Templates

These templates can be found in {appdir}\Templates\Assistant. Templates are *UsbDescriptors.h* files renamed to {templatename}.template. If additional files are needed they can be placed in an additional directory named {templatename}. In this directory an *info.ini* file has to be placed. In this current assistant version only class files are allowed. The ini file in following example is used for the virtual com port template:

```
[Information]
description=
classfiles=UsbClassCdc.h,UsbClassCdc.c
```

4.3 USB Descriptors and USB Class Templates

For all code files generated with the Fujitsu USB Assistant, there are template files which specify the structure and the data/code inside of the structure.

4.3.1 File Templates

Following files are file structure specific templates (located in {appdir}\Templates):

- **Empty_UsbDescriptors.h:**

  Template for the UsbDescriptors.h file
- **Empty_UsbClass.c:**

  Template for the USB Class code file.
- **Empty_UsbClass.h:**

  Template for the USB Class header file.

4.3.1.1 Keywords

Rev	Version
x.yy	Version
$Date$	Date
YYYY-MM-DD	Date
%INCLUDES%	Include files section
%PREPROCESSOR%	Global pre-processor symbols/macros ('#define') section
%TYPEDEF%	Global type definitions ('typedef') section
%DECLARATIONS%	Global variable declarations ('extern', definition in C source)
%PROTOTYPES%	Global function prototypes ('extern', definition in C source)
%FILENAME%	Filename (for USB Class files)
%CLASSINIT%	USB Class initializations
%FUNCTIONS%	Functions and procedures section

OBSOLETE

4.3.2 Code Templates

Code template are used to add features to the different files. Features are currently Class Events, Receiving Routines and Sending Routines. For each feature, the different sections can be written.

4.3.2.1 Code Sections

Sections are:

- **PROTOTYPE**:
  This section is used to define prototypes
- **DEFINE**:
  All pre-processor defines are included in this section
- **VAR**:
  All global variables are included here
- **INIT**:
  Here the init code is written (as content from a procedure)
- **CODE**:
  Here all code is written. This have to be procedures and functions
-

After the upper cased section name, the keywords [START] or [STOP] are added without space to open or close a section. Example:

```
CODE[START]
/* some code */
CODE[STOP]
```

4.3.2.2 Template Files

Following files are code specific templates (located in {appdir}\Templates):

- **UsbClass_ClassEvent.res:**

  Template for code implementing the class event handling in the initial USB class.
- **UsbClass_ReceivingRoutine.res:**

  Template for code implementing receiving routines in the initial USB class.
- **UsbClass_SendingRoutine.res:**

  Template for code implementing sending routines in the initial USB class.

4.3.2.3 Keywords

Rev	Version
x.yy	Version
$Date$	Date
YYYY-MM-DD	Date
%FILENAME%	Filename (for USB Class files)
%CLASSINIT%	USB Class initializations
%ENDPOINT%	Endpointnumber used with this procedure/function

OBSOLETE

5 Appendix

5.1 Figures

OBSOLETE

5.2 Information in the WWW

Information about FUJITSU MICROELECTRONICS products can be found here:

http://emea.fujitsu.com/microelectronics

Dedicated information on FUJITSU Microcontroller products including datasheets and manuals, software examples, application notes and tools can be found here:

http://mcu.emea.fujitsu.com/

OBSOLETE

OBSOLETE