

TASKING[®]

***Using the TASKING Pin
Mapper for AURIX***

Copyright © 2019 TASKING BV.

All rights reserved. You are permitted to print this document provided that (1) the use of such is for personal use only and will not be copied or posted on any network computer or broadcast in any media, and (2) no modifications of the document is made. Unauthorized duplication, in whole or part, of this document by any means, mechanical or electronic, including translation into another language, except for brief excerpts in published reviews, is prohibited without the express written permission of TASKING BV. Unauthorized duplication of this work may also be prohibited by local statute. Violators may be subject to both criminal and civil penalties, including fines and/or imprisonment. Altium[®], TASKING[®], and their respective logos are registered trademarks of Altium Limited or its subsidiaries. All other registered or unregistered trademarks referenced herein are the property of their respective owners and no trademark rights to the same are claimed.

Table of Contents

1. Preparing for First Use	1
1.1. Installing the Software	1
1.1.1. Installation for Windows	1
1.1.2. Licensing	2
1.2. Starting / Closing the TASKING Pin Mapper	6
1.3. How to Use the Documentation	7
1.4. Related Publications	8
2. Introduction	9
2.1. Port Pin	9
2.2. Peripheral	9
2.3. Connection	9
2.4. Package	10
2.5. Device Definition File	10
3. Getting Started with the TASKING Pin Mapper	11
3.1. Creating a Pin Mapper Project Containing a Pin Mapper Document	11
3.2. Adding a Pin Mapper Document to an Existing Project	12
3.3. TASKING Pin Mapper Perspective	13
3.3.1. TASKING Pin Mapper Editor View	14
3.3.2. Package View	18
3.3.3. Pin Conflicts View	21
3.3.4. Project Explorer View	22
3.4. Configuring the Pins	22
3.5. Solving Pin Conflicts	23
3.6. Generating and Using the Source Code	25
3.7. Generate CSV File	26
3.8. Launch Altium Designer	27
3.9. Pin Mapper Preferences	27
3.10. Errors and Warnings	27

Chapter 1. Preparing for First Use

This chapter guides you through the installation process of the TASKING® Pin Mapper for AURIX™. It also describes which documentation is available and how you best can use it.

In this manual, **TASKING Pin Mapper for AURIX** and **TASKING Pin Mapper** are used as synonyms.

1.1. Installing the Software

This section describes the installation of the embedded software for Windows. It also describes how to license the software.

1.1.1. Installation for Windows

System Requirements

Before installing, make sure the following minimum system requirements are met:

- 64-bit version of Windows 7 or higher
- 2 GHz Pentium class processor
- 4 GB memory
- 3 GB free hard disk space
- Screen resolution: 1024 x 768 or higher

Installation

1. If you received a download link, download the software and extract its contents.

- or -

If you received an USB flash drive, insert it into a free USB port on your computer.

2. Run the installation program (**setup.exe**).

The TASKING Setup dialog box appears.

3. Select a product and click on the **Install** button. If there is only one product, you can directly click on the **Install** button.
4. Follow the instructions that appear on your screen. During the installation you need to enter a license key, this is described in [Section 1.1.2, Licensing](#).

1.1.2. Licensing

TASKING products are protected with TASKING license management software (TLM). To use a TASKING product, you must install that product and install a license.

The following license types can be ordered from Altium.

Node-locked license

A node-locked license locks the software to one specific computer so you can use the product on that particular computer only.

For information about installing a node-locked license see [Section 1.1.2.3.2, *Installing Server Based Licenses \(Floating or Node-Locked\)*](#) and [Section 1.1.2.3.3, *Installing Client Based Licenses \(Node-Locked\)*](#).

Floating license

A floating license is a license located on a license server and can be used by multiple users on the network. Floating licenses allow you to share licenses among a group of users up to the number of users (seats) specified in the license.

For example, suppose 50 developers may use a client but only ten clients are running at any given time. In this scenario, you only require a ten seats floating license. When all ten licenses are in use, no other client instance can be used.

For information about installing a floating license see [Section 1.1.2.3.2, *Installing Server Based Licenses \(Floating or Node-Locked\)*](#).

License service types

The license service type specifies the process used to validate the license. The following types are possible:

- **Client based** (also known as 'standalone'). The license is serviced by the client. All information necessary to service the license is available on the computer that executes the TASKING product. This license service type is available for node-locked licenses only.
- **Server based** (also known as 'network based'). The license is serviced by a separate license server program that runs either on your companies' network or runs in the cloud. This license service type is available for both node-locked licenses and floating licenses.

Licenses can be serviced by a cloud based license server called "**Remote TASKING License Server**". This is a license server that is operated by TASKING. Alternatively, you can install a license server program on your local network. Such a server is called a "**Local TASKING License Server**". You have to configure such a license server yourself. The installation of a local TASKING license server is not part of this manual. You can order it as a separate product (SW000089).

The benefit of using the Remote TASKING License Server is that product installation and configuration is simplified.

Unless you have an IT department that is proficient with the setup and configuration of licensing systems we recommend to use the facilities offered by the Remote TASKING License Server.

1.1.2.1. Obtaining a License

You need a license key when you install a TASKING product on a computer. If you have not received such a license key follow the steps below to obtain one. Otherwise, you cannot install the software.

Obtaining a server based license (floating or node-locked)

- Order a TASKING product from Altium or one of its distributors.

A license key will be sent to you by email or on paper.

If your node-locked server based license is not yet bound to a specific computer ID, the license server binds the license to the computer that first uses the license.

Obtaining a client based license (node-locked)

To use a TASKING product on one particular computer with a license file, Altium needs to know the computer ID that uniquely identifies your computer. You can do this with the **getcid** program that is available on the TASKING website. The detailed steps are explained below.

1. Download the **getcid** program from <http://www.tasking.com/support/tlm/download.shtml>.
2. Execute the **getcid** program on the computer on which you want to use a TASKING product. The tool has no options. For example,

```
C:\Tasking\getcid  
Computer ID: 5Dzm-L9+Z-WFbO-aMkU-5Dzm-L9+Z-WFbO-aMkU-MDAy-Y2Zm
```

The computer ID is displayed on your screen.

3. Order a TASKING product from Altium or one of its distributors and supply the computer ID.

A license key and a license file will be sent to you by email or on paper.

When you have received your TASKING product, you are now ready to install it.

1.1.2.2. Frequently Asked Questions (FAQ)

If you have questions or encounter problems you can check the support page on the TASKING website.

<http://www.tasking.com/support/tlm/faq.shtml>

This page contains answers to questions for the TASKING license management system TLM.

If your question is not there, please contact your nearest Altium Sales & Support Center or Value Added Reseller.

1.1.2.3. Installing a License

The license setup procedure is done by the installation program.

If the installation program can access the internet then you only need the licence key. Given the license key the installation program retrieves all required information from the remote TASKING license server.

Using the TASKING Pin Mapper for AURIX

The install program sends the license key and the computer ID of the computer on which the installation program is running to the remote TASKING license server, no other data is transmitted.

If the installation program cannot access the internet the installation program asks you to enter the required information by hand. If you install a node-locked client based license you should have the license file at hand (see [Section 1.1.2.1, Obtaining a License](#)).

Floating licenses are always server based and node-locked licenses can be server based. All server based licenses are installed using the same procedure.

1.1.2.3.1. Configure the Firewall in your Network

For using the TASKING license servers the TASKING license manager tries to connect to the Remote TASKING servers `lic1.tasking.com` .. `lic4.tasking.com` at the TCP ports 8080, 8936 or 80. Make sure that the firewall in your network is transparently enabled for one of these ports.

1.1.2.3.2. Installing Server Based Licenses (Floating or Node-Locked)

If you do not have received your license key, read [Section 1.1.2.1, Obtaining a License](#) before you continue.

1. If you want to use a local license server, first install and run the local license server before you continue with step 2. You can order a local license server as a separate product (SW000089).
2. Install the TASKING product and follow the instruction that appear on your screen.

The installation program asks you to enter the license information.

TASKING Pin Mapper for AURIX vx.yrz - InstallShield Wizard

License key Information
Specify your license key

TASKING

Please enter the license key that you have received from TASKING. The key has the format like aaa-bbbb-cccc-dddd. If you do not have a key, please contact TASKING through licensing@tasking.com, or contact your TASKING representative.

License Key:

InstallShield

Licensing Support < Back Next > Cancel

3. In the **License key** field enter the license key you have received from Altium and click **Next** to continue.

The installation program tries to retrieve the license information from a remote TASKING license server. Wait until the license information is retrieved. If the license information is retrieved successfully subsequent dialogs are already filled-in and you only have to confirm the contents of the dialogs by clicking the **Next** button. If the license information is not retrieved successfully you have to enter the information by hand.

4. Select your **License type** and click **Next** to continue.

You can find the license type in the email or paper that contains the license key.

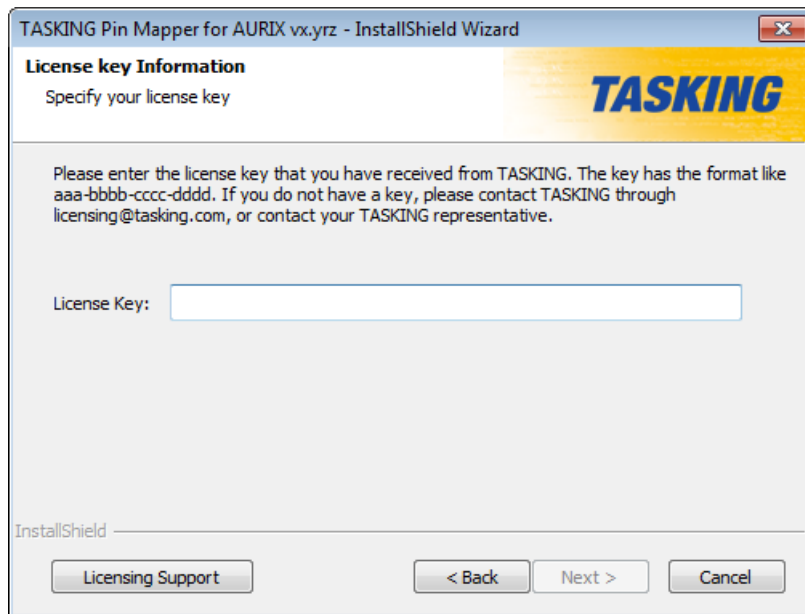
5. Select **Remote TASKING license server** to use one of the remote TASKING license servers, or select **Local TASKING license server** for a local license server. The latter requires optional software.
6. (For local license server only) specify the **Server name** and **Port number** of the local license server.
7. Click **Finish** to complete the installation.

1.1.2.3.3. Installing Client Based Licenses (Node-Locked)

If you do not have received your license key and license file, read [Section 1.1.2.1, Obtaining a License](#) before continuing.

1. Install the TASKING product and follow the instruction that appear on your screen.

The installation program asks you to enter the license information.



2. In the **License key** field enter the license key you have received from Altium and click **Next** to continue.

Using the TASKING Pin Mapper for AURIX

*The installation program tries to retrieve the license information from a remote TASKING license server. Wait until the license information is retrieved. If the license information is retrieved successfully subsequent dialogs are already filled-in and you only have to confirm the contents of the dialogs by clicking the **Next** button. If the license information is not retrieved successfully you have to enter the information by hand.*

3. Select **Node-locked client based license** and click **Next** to continue.
4. In the **License file content** field enter the contents of the license file you have received from Altium.
The license data is stored in the file licfile.txt in the etc directory of the product.
5. Click **Finish** to complete the installation.

1.2. Starting / Closing the TASKING Pin Mapper

The TASKING Pin Mapper uses Eclipse as the Integrated Development environment (IDE).

Starting the TASKING Pin Mapper

To start the TASKING Pin Mapper:

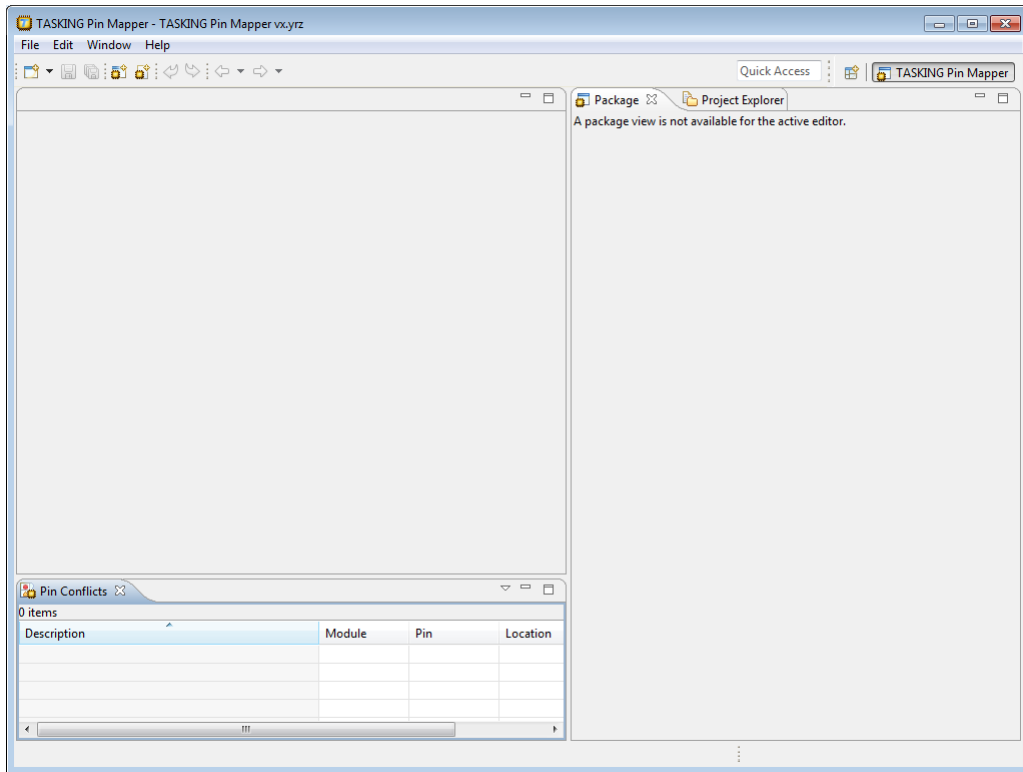
1. On Windows PCs, from the **Start** menu, select **Programs » TASKING Pin Mapper for AURIX vx.yrz » TASKING Pin Mapper**.

On UNIX/Linux workstations, type **eclipse** at a shell command prompt, assuming that the TASKING Pin Mapper program directory is in the search path.

The Workspace Launcher dialog appears.

2. Enter the path to the workspace.
In the remainder of this manual, we assume you use the default.
3. Enable the option **Use this as the default and do not ask again**.
4. Click **OK** to proceed.

Initially, the TASKING Pin Mapper opens with a workbench displaying the TASKING Pin Mapper perspective with several views. Your workbench should look similar to this:



Closing the TASKING Pin Mapper

To close TASKING Pin Mapper:

- From the **File** menu, select **Exit**.

Upon exit, the TASKING Pin Mapper saves the current workbench layout. The next time you start the TASKING Pin Mapper, the last saved workbench layout is used.

1.3. How to Use the Documentation

The documentation for the TASKING Pin Mapper for AURIX consists of:

- online documentation for Eclipse
- this Using the TASKING Pin Mapper for AURIX manual

It is strongly recommended to read the documentation in this order.

Getting acquainted with Eclipse

If you are new to Eclipse, start familiarizing with Eclipse. Eclipse comes with a document that describes how Eclipse is organized as a Workbench, with Perspectives that contain Views.

To start with this documentation:

1. Start Eclipse.
2. From the **Help** menu, select **Help Contents**.

The help screen overlays the Eclipse Workbench.

3. In the left pane, select **Workbench User Guide** to learn more about working in Eclipse.

Using the TASKING Pin Mapper for AURIX (this manual)

This manual contains specific information for the TASKING Pin Mapper for AURIX. Its content overrides any information found in the Eclipse documentation.

The next chapters of this manual explain how to setup and start working with the TASKING Pin Mapper for AURIX.

This manual is also available in PDF format via the Windows Start menu:

- Browse to **Start » Programs » TASKING Pin Mapper for AURIX vx.y.rz » Manuals** and select the **Pin Mapper User Guide**.

1.4. Related Publications

AURIX

- TC2xx User's Manuals [Infineon]
- TC3xx User's Manuals [Infineon]

Chapter 2. Introduction

Nowadays many microcontrollers are equipped with a large number of on-chip peripheral modules. These microcontrollers are made available in surface mount packages with various numbers of physical pins. The number of pins usually does not allow all peripheral modules to be used simultaneously. Hardware registers on the microcontroller allow for configurable assignment of peripheral module signals to physical pins. This means that you have to select the proper device for your application and properly initialize hardware registers from software. The purpose of the TASKING Pin Mapper is to assist you in performing those tasks.

2.1. Port Pin

The basic entity of the TASKING Pin Mapper is the port pin. Port pins are physical pins of the chip. A port pin name typically consists of a port identifier and, if the port identifier is shared by multiple pins, a sequential pin number, for example TMS, AN4 or P10.2. A port pin name is not package specific. Each package has its own specific assignment of port pin names to package pin numbers.

The basic configuration item of a port pin is its function (which in most cases is determined by the signal assigned to it). Some ports have dedicated functions, mostly related to power supply or chip infrastructure. Other ports have configurable functionality (ultimately defined by run-time register settings). The required function is application specific.

A port function definition consists of a symbolic name, a direction and a description. The TASKING Pin Mapper distinguishes between input functions and output functions. Functions are listed by their symbolic names (which are mostly signal names, for example ASCLK3).

Port pins can have a number of other chip specific properties. Such properties can be fixed (for example the type of pad the port is connected to) or configurable (for example the pad state). Port pins are visible in the Package view. You can assign a user-defined symbol name to the port pin and you can make the symbol names visible in the Package view. You can use filters in pin selection views or you can use property values as background color codes in the Package view. You can change the configurable property values.

2.2. Peripheral

AURIX devices, such as the TC27xC, have support for several on-chip peripherals, such as serial controllers, timer units, and analog-to-digital converters.


2.3. Connection

The TASKING Pin Mapper allows you to define connections between modules, where a module is either a port pin or a peripheral.

You can connect port pin modules to peripheral modules by selecting the pin's Direction and its Chip input function or its Chip output function. The peripheral module side of a connection is referred to as a virtual pin. Some peripheral modules can also be connected to other peripheral modules. Such a connection

Using the TASKING Pin Mapper for AURIX

is made between two virtual pins. Virtual pins are internal to the chip and are not visible in the Package view. However, virtual pin conflicts are reported in the Pin Conflicts view, for example when you try to connect a GTM virtual output pin to a physical port pin and at the same time to a virtual input pin of an ADC peripheral module.

If you have defined a connection (either valid or invalid) the "follow the connection" button () becomes available and you can navigate from one side of the connection to the other side. This applies to virtual pins as well as physical pins. Virtual pins usually do not have a configurable direction. They are either Input or Output (and some virtual pins are bidirectional).

See [Section 3.4, Configuring the Pins](#) for a step-by-step example.

2.4. Package

When you create a new pin mapper document, as explained in [Section 3.2, Adding a Pin Mapper Document to an Existing Project](#), you first select a processor (family), for example TC27xC, and a package, for example BGA292. This selection defines the package pin numbers and the linking of package pins to port pins. It is possible that a port pin is linked to a single package pin multiple times by means of port pin aliases, for example AN39 and P40_9.

For QFP (Quad Flat Package) type packages the pin number consists of the package identifier and a sequential pin number. For BGA (Ball Grid Array) type packages the pin number consists of the package identifier and a matrix cell number.

2.5. Device Definition File

The characteristics of the device, such as the available packages, the number of pins and the peripherals present are defined in a device definition file. The device definition file has the XML format. For example, the device definitions for the TC27xC and TC27xD are combined in the file `TC27x_C-D_pinmappings.xml`. You can choose to add this file to your project. This way you are sure to always use the same device definitions, independent of any software updates of the TASKING Pin Mapper.

Chapter 3. Getting Started with the TASKING Pin Mapper

The TASKING Pin Mapper is the graphical interface to configure the pins and modules in a package.

3.1. Creating a Pin Mapper Project Containing a Pin Mapper Document

To configure the pins and modules of a package you need to create a TASKING Pin Mapper project and add a Pin Mapper document. If you have no existing TASKING Pin Mapper project, you need to create one first.

1. From the **File** menu, select **New » TASKING Pin Mapper Project**.

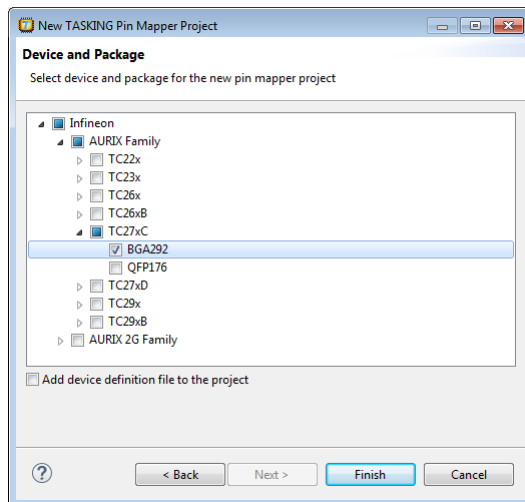
The New TASKING Pin Mapper Project wizard appears.

2. Enter a name for your project, for example `myproject`.

*In the **Location** field you will see the location where the new project will be stored. To change the default location, you can uncheck the **Use default location** check box and browse for an alternative location. However, use the default location for now.*

3. Click **Next**.

The Device and Package page appears.



4. Select the processor, for example TC27xC, and package, for example BGA292, for which you want to make a pin configuration.

Using the TASKING Pin Mapper for AURIX

5. Select **Add device definition file to the project** if you want to make sure that the device definition is stored with the project. This can be useful when the device definition has to stay the same, independent from any software updates of the TASKING Pin Mapper.
6. Click **Finish**.

A new project `myproject` is created that contains a pin configuration file `myproject.pincfg` and optionally a device definition file `TC27x_C-D_pinmappings.xml`. If the TASKING Pin Mapper perspective is not opened yet, Eclipse asks to open it. In that case click **Yes**.

3.2. Adding a Pin Mapper Document to an Existing Project

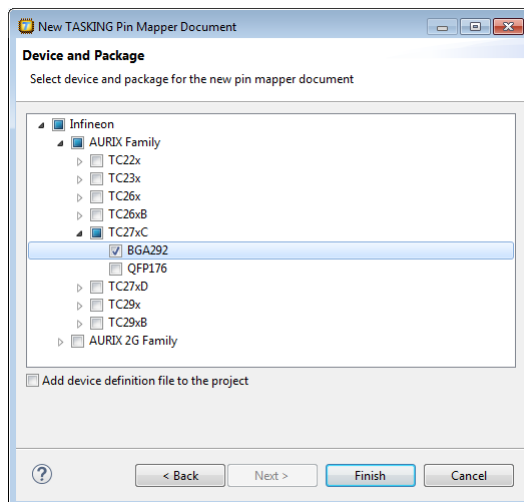
To configure the pins and modules of a package you need to create a TASKING Pin Mapper project and add a Pin Mapper document. If you have no existing TASKING Pin Mapper project, you need to create one first. See [Section 3.1, Creating a Pin Mapper Project Containing a Pin Mapper Document](#). You can add multiple Pin Mapper documents to a project.

1. From the **File** menu, select **New » TASKING Pin Mapper Document**.

The New TASKING Pin Mapper Document wizard appears.

2. Select the **Project** folder for the Pin Mapper document: type the name of your project (`myproject`) or click the **Browse** button to select a project.
3. In the **File name** field, enter a name for the Pin Mapper document, for example `myproject_TC27xC.pincfg` and click **Next**.

The Device and Package page appears.



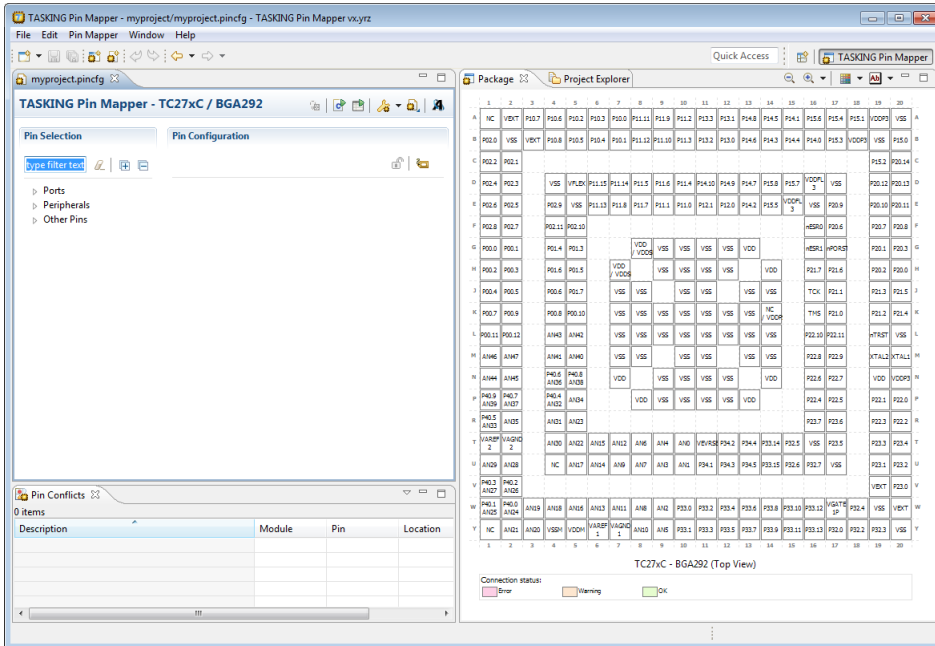
4. Select the processor, for example TC27xC, and package, for example BGA292, for which you want to make a pin configuration.

5. Select **Add device definition file to the project** if you want to make sure that the device definition is stored with the project. This can be useful when the device definition has to stay the same, independent from any software updates of the TASKING Pin Mapper.
6. Click **Finish**.

A pin configuration file `myproject_TC27xC.pincfg`, and optionally a device definition file `TC27x_C-D_pinmappings.xml`, is added to the existing project. If the TASKING Pin Mapper perspective is not opened yet, Eclipse asks to open it. In that case click **Yes**.

3.3. TASKING Pin Mapper Perspective

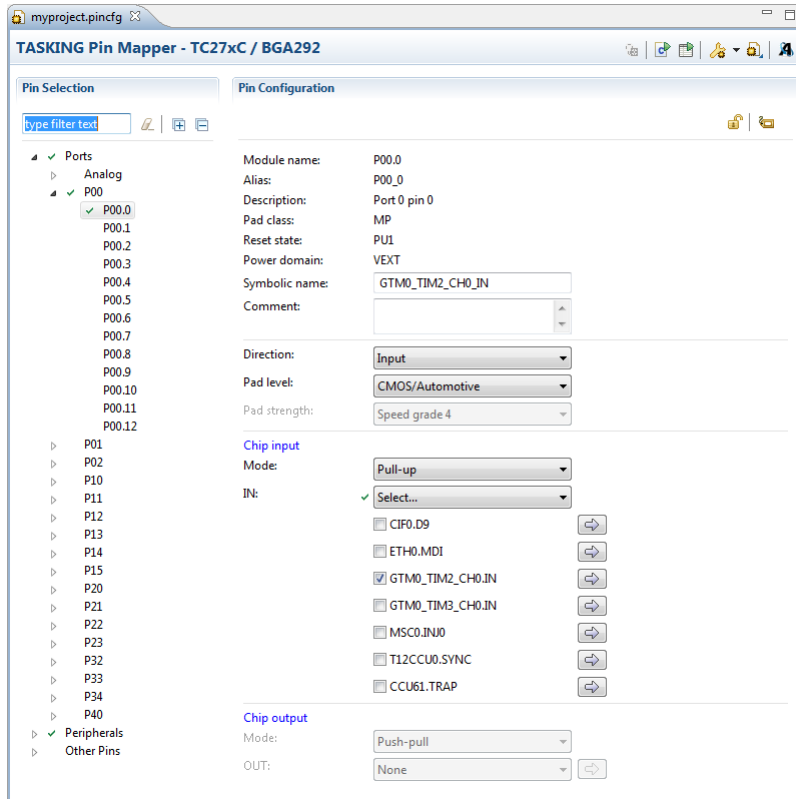
The TASKING Pin Mapper perspective contains several views. Each of the views are discussed in the following sections.



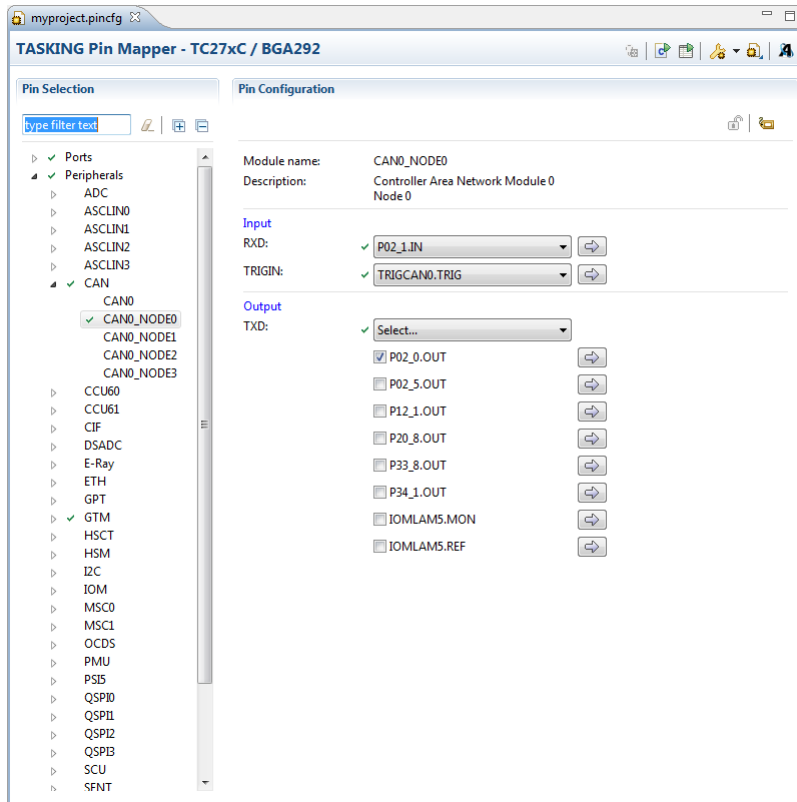
3.3.1. TASKING Pin Mapper Editor View

The TASKING Pin Mapper Editor view is the main area where you can make changes to your configuration.

Port example



Peripheral example



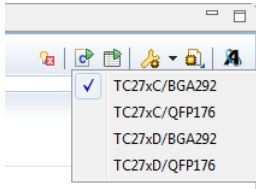
The following toolbar icons are available:

Icon	Action	Description
	Solve Pin Conflicts	Solve conflicts with pin mappings. See Section 3.3.3, Pin Conflicts View .
	Generate Code	Generate the source code and add it to your project. See Section 3.6, Generating and Using the Source Code .
	Generate CSV File	Save the pin mapper configuration in a file with comma-separated values (CSV). See Section 3.7, Generate CSV File .
	Select Device/Package	Select another package for your device. For example, from BGA to QFP or vice versa.
	Import Pin Configuration	Import pin configurations from a file and merge it with the current configuration.
	Launch Altium Designer	Launch Altium Designer with the pin configurations. See Section 3.8, Launch Altium Designer .

Using the TASKING Pin Mapper for AURIX

Select Device/Package

With the Select Device/Package drop-down menu (🔑) you change the package for your device. For example:



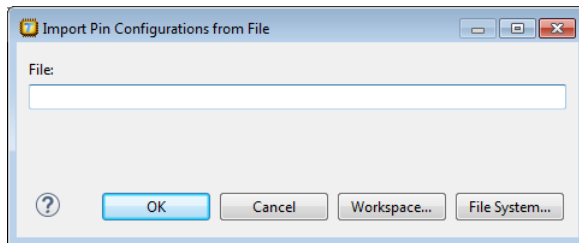
After the package switch you still keep the current configuration settings. If the same pin mapping can be used on the new device/package, the pin mapper automatically applies the mapping. For mapping parts which cannot be applied, the pin mapper issues a warning message.

Import Pin Configuration

You can import saved pin configurations and merge it with the current configuration. This can be useful, for example, when you have a set of configurations for ports and peripherals that you want to use in several projects.

1. Click the Import Pin Configuration button (📁).

The Import Pin Configurations from File dialog appears.



2. Type the full path name of the configuration file (`.pincfg`), or use the **Workspace** button to select a configuration from one of your project directories, or use the **File System** button to select one from any directory.

The imported configuration will be merged with the current configuration.

If the current configuration already has settings for a pin or peripheral and the imported configuration has other settings, pin conflicts errors or warnings may occur.

Pin Selection

In the left pane the port pins, peripherals and other pins are listed. Port pins can be logically grouped in different ways. Each of these groupings defines a tree view. Groups can be logically grouped into higher

level groups as long as the overall structure can be represented as a mathematical tree. The leaves of the tree are references to port pins, peripherals or peripheral sub-modules.

The tree structure of the grouping allows you to expand and collapse the view.

In the `type filter text` edit field, you can add a port pin selection filter in order to reduce the number of visible port pins. The filter is case insensitive. For example, type `an` to only show pins that have "an" in their name. Wildcards are allowed.



Pin Configuration

When you click on a port or module in the left pane, the configuration appears in the right pane. The following information can be present:

- **Module name** - The name of the selected module or port pin.
- **Alias** - An alias for the module name.
- **Description** - The description of the selected module or port pin.
- **Pad class** - The assignment of a port pin to one of the pad classes.
- **Reset state** - The state of the pin after reset. For example, PU (Pull-up) or PD (Pull-down).
- **Power domain** - The power domain the pin uses.
- **Symbolic name** - You can assign a user-defined symbol name to the port pin. You can make the symbol names visible in the Package view.
- **Comment** - Any user comments you can add here.
- **Pin function** - You can configure some pins as digital or analog.
- **Direction** - You can specify if a pin must be configured as an input pin or an output pin.
- **Pad level** - Here you can select CMOS/Automotive or TTL.
- **Pad strength** - For an output pin you can specify the Speed grade.
- **Mode** - The chip input or output mode. For chip input this can be Pull-up, Pull-down or Tri-state. For chip output this can be Push-pull or Open drain.
- Other properties allow you to make a pin connection. See [Section 3.4, Configuring the Pins](#) for more information.



When you make a selection a '*' can appear in front of other selectable pin names. When you select a pin name with a '*', this results in an error.

Lock/Unlock

When you want to be certain that the port settings cannot be changed anymore, you can click the  button to lock the current settings. You can always click the  button to unlock the settings again.

Using the TASKING Pin Mapper for AURIX


Tags

You can assign a tag name to a connection. When you click the  (Edit tags) button, you can edit the tag field to the right of the  (Follow connection) button. You can use it the way you want. For example, to tag connections of the same type.


Undo/Redo

You can undo (Ctrl+Z) or redo (Ctrl+Y) one or more actions.

Back/Forward

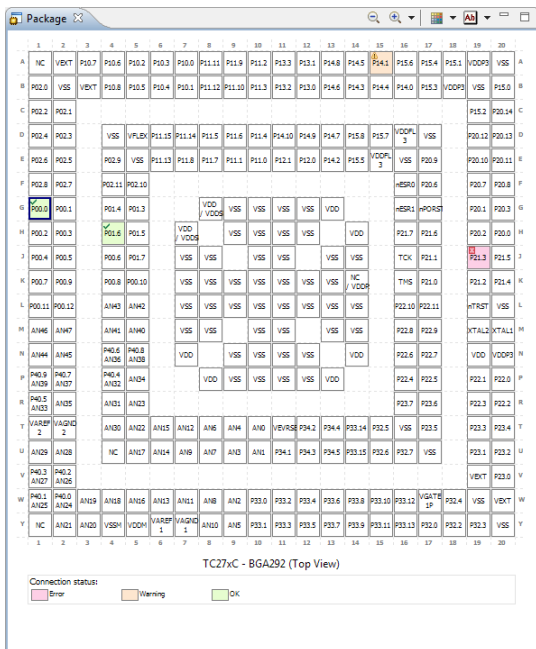
With the  (Back) and  (Forward) buttons you can navigate to a previously opened pin configuration.

Save

From the **File** menu, select **Save** (Ctrl+S) or click  to save the configuration.

3.3.2. Package View




The Package view shows a graphical representation of the package. For example, it shows the individual pins in a Ball Grid Array.




A square around a pin marks the selected pin. In the package above P00.0 is selected. A green check mark indicates that the pin has a valid connection; P00.0 and P01.6 in the package above. A red cross

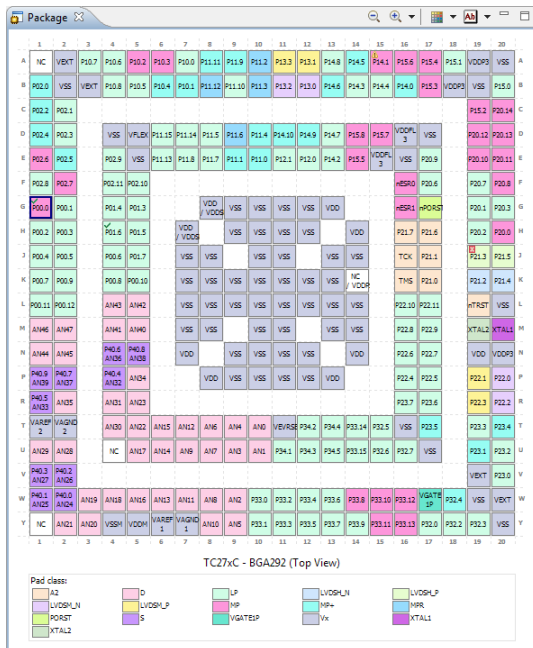
indicates an error; P21.3 in the package above. A triangle with exclamation mark indicates a warning; P14.4 in the package above. When you click on a pin, the pin appears in the editor. If the pin contains an error, you can see what is wrong in the Pin Conflicts view.

Zoom in / Zoom out

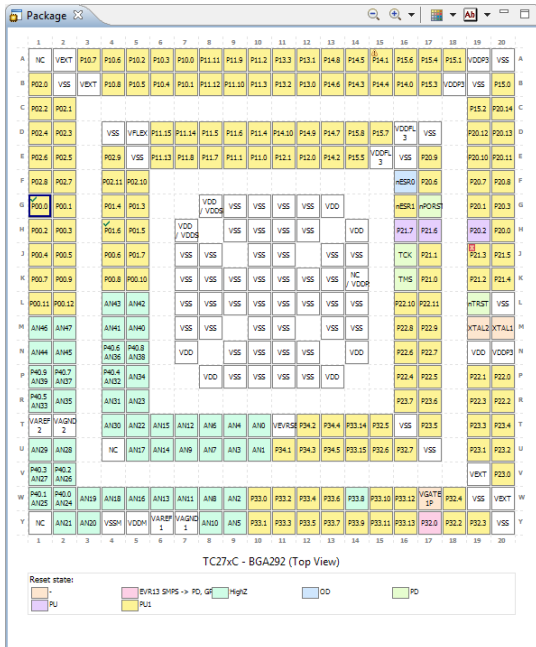
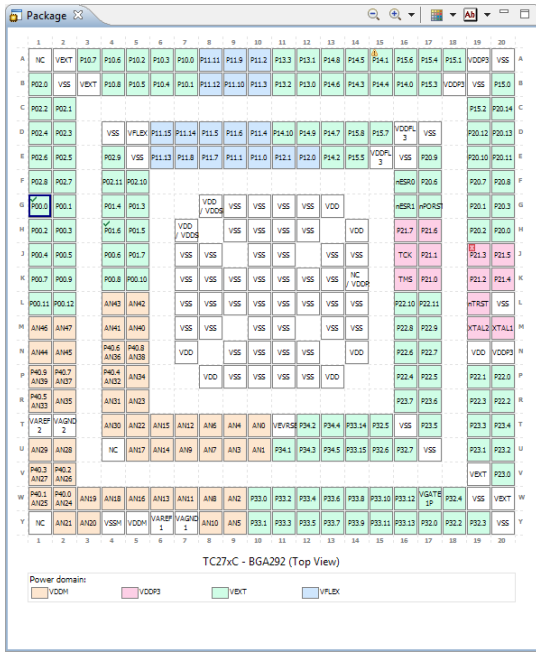
With  and  you can zoom in or zoom out on the package view. Use the drop-down menu next to  to select a zoom factor directly.

Background Color

By default, the background colors of the pins indicate the connection status. But you can change the color properties of the view to show static configuration items (Pad class, Power domain or Reset state), or dynamic configuration items (Direction, Pad level, Pad strength, Mode or Pin function). Use the drop-down menu next to  to switch color properties. Some examples:



Using the TASKING Pin Mapper for AURIX

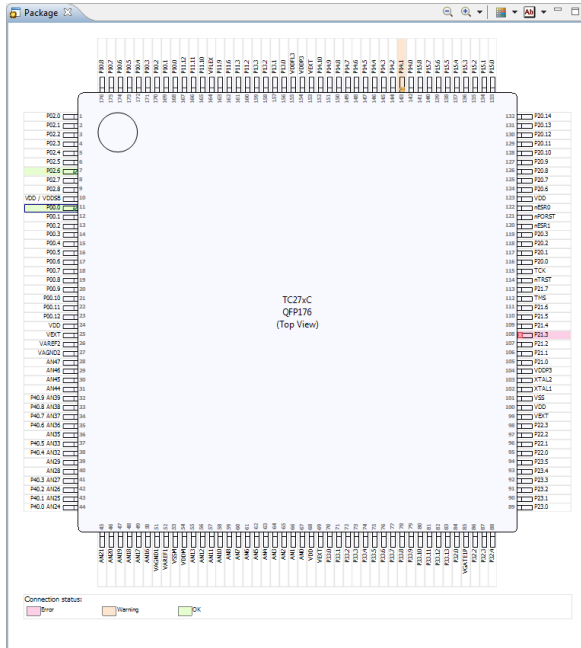


Labels

With the drop-down menu next to  you can switch between module names and symbol names.



Quad Flat Package

If you have selected a Quad Flat Package, the Package view may look similar to:



3.3.3. Pin Conflicts View

Potentially conflicting pin assignments are reported in the Pin Conflicts view. For example, if multiple virtual pins are assigned to the same port pin then there is a potential conflict. Also, if the same virtual pin is assigned to multiple port pins then there is a potential conflict.

Pin Conflicts				
1 error, 1 warning, 0 others				
Description	Module	Pin	Location	Resource
 Dangling connection	P00.11	IN	N.A.	myproject.pi...
 Multiple selections for a 1-to-N connection	P14.1	IN	N.A.	myproject.pi...

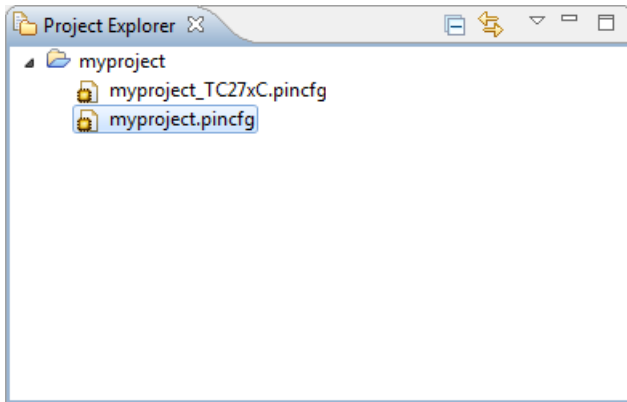
When you double-click on a conflict the corresponding pin configuration will be visible in the Editor view.

See [Section 3.5, Solving Pin Conflicts](#) for information how to solve pin conflicts.

See [Section 3.10, Errors and Warnings](#) for a list of errors and warnings.

3.3.4. Project Explorer View

The Project Explorer allows you to navigate the contents of your workspace. It contains a list of the available projects and Pin Configuration files. This is a standard Eclipse view.



3.4. Configuring the Pins

Basically you can connect pins in two different ways. The first way is to select a port pin and then assign a peripheral virtual pin (function) to it. The second way is to select a peripheral virtual pin (function) and then assign a port pin to it.

You can choose between input or output mode. If an application is going to use a port pin in both directions then you should select the initial run-time direction of the port. This is necessary because the pin mapper will generate port pin initialization code and needs to know whether to configure for input or for output.




To connect pins from the port pin side


1. In the Project Explorer view double-click on a configuration file (`myproject.pincfg`) to view its contents.
2. In the Pin selection pane, select a port pin. For example, **Ports » P00 » P00.0**.
3. In the Pin configuration pane select the **Direction: Input** or **Output**. In this example we select **Input**.
4. Select the pin **Mode**.
5. Connect the input pin to a peripheral virtual pin:
 - From the **IN** drop-down box choose **GPIO** or **Select...**

With Select... a virtual pin appears, or with a 1-to-N connection (one source, many destinations) a list of virtual pins appears.





- Select one (or more) pins.

A green check mark ✓ appears if the connection is successful.

- (Optional) When you want to prevent that these settings can be changed, click  to lock the settings.
A little lock  appears next to the port pin.
- From the **File** menu, select **Save** (Ctrl+S) or click  to save the configuration.

When you click  you follow the connection from source to destination or vice versa. The corresponding selection is shown in a light blue color.


To connect pins from the peripheral side

- In the Project Explorer view double-click on a configuration file (`myproject.pincfg`) to view its contents.
- In the Pin selection pane, select a module. For example, **Peripherals » ETH » ETH0**.
- In the Pin configuration pane select the pin you want to assign to the virtual pin.
A green check mark ✓ appears if the connection is successful.
- Click  to follow the connection to the port pin side. The corresponding selection is shown in a light blue color.
- (Optional) When you want to prevent that these settings can be changed, click  to lock the settings.
A little lock  appears next to the port pin.
- From the **File** menu, select **Save** (Ctrl+S) or click  to save the configuration.

3.5. Solving Pin Conflicts

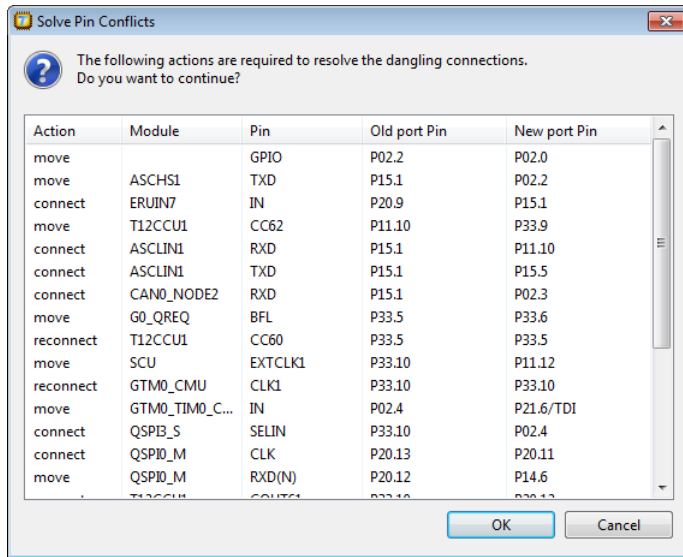
It is your responsibility to solve any pin conflicts. It is important that there are no conflicts when you use the tools to generate the code. You can solve the conflicts by hand by making other connections, but when there are many conflicts or in situations where most port pins are in use, it can be quite cumbersome and complex to solve conflicts. The Pin Conflicts Solver can automate this process and solve most conflicts for you.

To solve pin conflicts

- From the **Pin Mapper** menu, select **Solve Pin Conflicts** ()

The Solve Pin Conflicts dialog appears with a list of actions that will be taken when you click OK.

Using the TASKING Pin Mapper for AURIX



2. Click **OK** to accept the actions.

The configuration is updated accordingly.

3. Check the pin configuration settings of the new connections and make changes if needed.

The meaning of the columns in the Solve Pin Conflicts dialog is:

Column	Description
Action	<p>The action that takes place to solve the pin conflict.</p> <p><code>move</code> - move the connection of the module pin from the Old port Pin to the New port Pin including all configuration settings. The Old port Pin is reset to the default settings.</p> <p><code>connect</code> - connect the module pin from the Old port Pin to the New port Pin using the default configuration settings. You still need to check and configure the pin settings.</p> <p><code>reconnect</code> - after a move of the original connection to another pin, reconnect to the same port pin. You still need to check and configure the pin settings.</p> <p><code>reset</code> - reset the port pin to the default settings.</p>
Module	The name of the module the action is applicable to.
Pin	The name of the pin the action is applicable to.
Old port Pin	The name of the port pin the module pin was connected to.
New port Pin	The name of the port pin the module pin will be connected to.

Note that the solver only tries to solve dangling connections between virtual module pins and port pins, not between virtual pins (internal connections). Also, the solver does not change peripheral instances. If, for example, a connection error with peripheral ASCLIN0 exists, the solver will not try to use ASCLIN1 instead. Try to use a different instance of the same peripheral type manually. If the solver does not find a solution for all pin conflicts, then no port pins will be swapped at all. You can try to move some connections manually or unlock some pin settings (if applicable) or, as a final resort, try to switch to another device/package.


Depending on the projected calculation time, the solver either performs an exhaustive search or applies a heuristic partial search algorithm to find a solution. In case of a partial search it is possible that a solution does exist while the solver does not find it. The solver error dialog distinguishes these situations.

Locked connections are not touched. If you have locked a port pin, its configuration stays the same. The standard undo/redo functionality is available. The menu entry and button are disabled (grayed) if there are no Pin Conflicts to resolve.

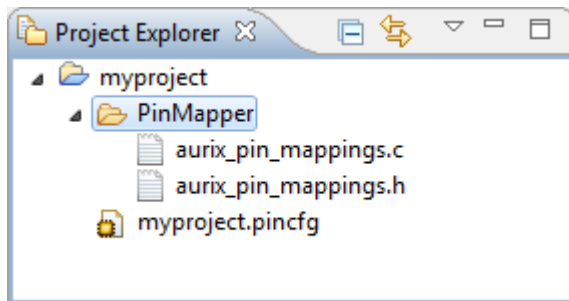
3.6. Generating and Using the Source Code

When you have configured the pins you are ready to add the Pin Mapper sources to your project.

Generate source code

- From the **Pin Mapper** menu, select **Generate Code** or click .

The Pin Mapper sources are generated and are added to your project in the folder `PinMapper`.



Using the Pin Mapper sources

The Pin Mapper sources initialize port pins configured as GPIO as well as port pins connected to peripherals. For port pins the following line is present in `aurix_pin_mappings.h`.

```
#include <Port/Io/IfxPort_Io.h>
```

The generated code is meant to work together with the Infineon iLLD library. This library must be available to your project.

Using the TASKING Pin Mapper for AURIX

The Infineon iLLD library is not part of the product. You have to obtain it from Infineon.

You can access the Pin Mapper sources from your application. To use the generated files, you must add a `#include` statement into your top-level source file (typically `main.c`):

```
#include "PinMapper/aurix_pin_mappings.h"
```

You can now use the initialization functions of the Pin Mapper in your sources for initialization of the port pins. The initialization functions have the following name convention:

```
pinname_init_pins()
```

For example,

```
#include "PinMapper/aurix_pin_mappings.h"

int main( void )
{
    /* initialize GPIO */
    gpio_init_pins();


    /* initialize port pin gtm0_tim2_ch0 */
    gtm0_tim2_ch0_init_pins();

    /* ... */

    return 0;
}
```

3.7. Generate CSV File

You can save the pin mapper configuration in a file with comma-separated values (CSV).

- From the **Pin Mapper** menu, select **Generate CSV File** .

The CSV file (.csv) is generated in the folder `PinMapper` of your project.*

The following is an example of such a file:

```
"Name","Pin","Function","Direction","Mode","Pad level","Pad class","Power domain","Reset state"
"P00.0","G1","GTM0_TIM2_CH0_IN","Input","Pull-up","CMOS/Automotive","MP","VEXT","PU1"
"P10.5","B5","SCU_HWCFG4","Input","Pull-up","CMOS/Automotive","LP","VEXT","PU1"
```

You can open the file with any text editor or in a spreadsheet program.

3.8. Launch Altium Designer

When you have Altium Designer installed with the TASKING Pin Mapper software extension, you can start Altium Designer from the TASKING Pin Mapper Editor view.

- From the **Pin Mapper** menu, select **Launch Altium Designer** (4).

Altium Designer launches and a wizard guides you to import the pin configurations. This results in a schematic document with the selected device, peripherals and pins.

3.9. Pin Mapper Preferences

You can use the Preferences dialog in Eclipse to specify how the Pin Mapper should operate.

To set preferences

1. From the **Window** menu, select **Preferences**.

The Preferences dialog appears.

2. Select **TASKING Pin Mapper**.

The Pin Mapper page appears.

3. Set your preferences and click **OK**.

You can set the following preferences:

Generate code on save

By default the TASKING Pin Mapper asks if you want to generate code when you save a document (**Prompt**). You can choose to do this automatically (**Always**) or **Never**.

3.10. Errors and Warnings

The TASKING pin mapper reports errors and warnings in the [Pin Conflicts view](#). When you encounter an error, you can double-click on the error message to go to the conflicting situation.

Error: Dangling connection

A connection consists of two ends; a source and a destination. If the source is connected to a destination while the destination is not connected to the source then the connection is reported to be dangling. There are two situations in which this error can appear.

- Situation 1. When dealing with a 1-to-N connection (one source, many destinations), on the right-hand side of the drop-down box a list of destinations is shown. Each destination has a check box and you can make one or more selections. Initially nothing is selected and at that point the error appears.

Using the TASKING Pin Mapper for AURIX

For example, select **Ports » P00 » P00.0** and change the **Chip input** configuration **IN** from **None** to **Select**. Several destinations appear, all unselected. The error appears to notify you to make a selection. Once you made a selection, the error disappears.

- Situation 2. When you attempt to connect a source to a destination that already has a connection. In this case the source is connected to the destination, while the destination keeps its old connection. For example,
 1. Select **Ports » P00 » P00.0** and change the **Direction** configuration from **Input** to **Output**
 2. Change the configuration **OUT** from **None** to **GPIO**.
 3. Select **Peripherals » GTM » TOUT9** and change the output **OUT** from **None** to **P00_0.OUT**.

The error appears since TOUT9.OUT is now connected to P00_0.OUT, while P00_0.OUT is still connected to GPIO.

Error: Incorrect settings for *name*, expected pins *name1* and *name2* to be connected

This error appears when you connect one pin, where multiple pins need to be connected.

For example,

1. Select **Ports » P21 » P21.3** and change the **Direction** configuration from **Input** to **Output**.
2. Change the configuration **OUT** from **None** to **ETH0.MDO3**.

The error will disappear when you follow the connection and also connect input MDI from ETH0 to P21_3.IN.

Error: Modules *name1* and *name2* cannot be used at the same time, since they are mutually exclusive

This error appears when you try to use two modules of which only one module can be used at a time.

For example,

1. Select **Peripherals » ASCLIN0 » SPI3W0** and change the input **RXD(N)** from **None** to **P14_1.IN**.
2. Select **Peripherals » ASCLIN0 » SPI4W0** and change the output **TXD** from **None** to **P14_1.OUT**.

Modules SPI4W0 and SPI3W0 cannot be used at the same time.

Warning: Multiple selections for a 1-to-N connection

This warning appears when you make more than one selection for a 1-to-N connection.

For example, select **Ports » P00 » P00.0** and change the **Chip input** configuration **IN** from **None** to **Select**. Several destinations appear, all unselected. Now select more than one destination. The warning disappears if you only select one destination.

