

MA160-870 April 05, 2019

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. Introduction	
1.1. Port Pin	1
1.2. Peripheral	1
1.3. Connection	
1.4. Package	2
1.5. Device Definition File	
2. Getting Started with the TASKING Pin Mapper	
2.1. Adding a Pin Mapper Document to an Existing Project	
2.2. TASKING Pin Mapper Perspective	
2.2.1. TASKING Pin Mapper Editor View	5
2.2.2. Package View	
2.2.3. Pin Conflicts View	12
2.3. Configuring the Pins	13
2.4. Solving Pin Conflicts	14
2.5. Generating and Using the Source Code	16
2.6. Generate CSV File	
2.7. Launch Altium Designer	17
2.8. Pin Mapper Preferences	
2.9. Errors and Warnings	

Chapter 1. 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.

1.1. 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.

1.2. 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.

1.3. 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

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 2.3, Configuring the Pins for a step-by-step example.

1.4. Package

When you create a new pin mapper document, as explained in Section 2.1, 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.

1.5. 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 2. Getting Started with the TASKING Pin Mapper

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

2.1. Adding a Pin Mapper Document to an Existing Project

To configure the pins and modules of a package you need to create a project and add a Pin Mapper document.

1. If you have no existing C project, follow the steps to create a new C project, as explained in the *Getting Started with the TASKING VX-toolset for TriCore*.

In the following steps we assume you have an existing TriCore TC27xC C project named myproject.

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

The New TASKING Pin Mapper Document wizard appears.

- 3. 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.
- 4. In the File name field, enter a name for the Pin Mapper document, for example myproject.pincfg and click Next.

The Device and Package page appears.

🖸 New TASKING Pin Mapper Document
Device and Package
Select device and package for the new pin mapper document
Processor Settings
Only show items for TC27xC Show all Show
▲ Infineon ▲ URIX Family ▲ IT C27xC ✓ BGA292 ✓ QFP176
Add device definition file to the project
Cancel

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

- 6. 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.
- 7. Click Finish.

A pin configuration file <code>myproject.pincfg</code>, and optionally a device definition file, is added to the existing project. Because the TASKING C/C++ perspective is still active, Eclipse asks to open the TASKING Pin Mapper perspective. In that case click **Yes**.

2.2. TASKING Pin Mapper Perspective

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

D TASKING Pin Mapper - myproject/myproject.pincfg	TriCore Eclipse	IDE vx.yrz																				-			×
File Edit Navigate Search Project Debug Pin																									
📬 🕶 🖪 🔛 📾 🕸 🕶 🛷 🕶 💱 🖓	- 🎨 🔶 - 🛛	Ş ▼ 🖻 —							Q	uick A	Acces	s			EÎ.	NX.	TAS	KING	C/C	•• (5 T/	ASKIN	G Pin	Map	per
nyproject.pincfg 🛛			- 6		Pac	kage	×												€	٠	-	•	Ab ·	, –	
TASKING Pin Mapper - TC27xC / BGA2	92 🤉	s 🖻 🖻 a	<u> </u> a = 🗟 A					1		1		8		10	11	12	13	14	15	16	17	18	19	20	
Pin Selection Pin Configuration					A NC		-		<u> </u>	<u> </u>	<u> </u>	P11.11					P54.8	<u> </u>	<u> </u>	P15.6			(00P3	_	^
					5 902.0			P10.8	P10.5	P10.4	P10.1	P11.12	P11.10	P11.3	P13.2	P13.0	P14.6	P14.3	P14.4	P14.0	P15.3		_	P15.0	
type filter text 🖉 🕀 🕀			6 🔁			2 902.5		-	_			_												P20.14	
> Ports						4 P02.3		VSS		P11.15					P54.50				P15.7	V00PL 3			_	P20.13	
Peripherals Other Pins					-	6 902.5		902.9	<u> </u>	P11.13	P11.8	P11.7	P11.1	P11.0	P12.1	P12.0	P14.2	P15.5	3	<u> </u>	P20.9		_	P20.11	E
					-	8 902.7		-	P02.10				_	_	_					nESRO			_	P20.8	
						D P00.5		1	P01.3		-	V00 / V005	VSS	_	VSS	VSS	VDD			<u> </u>	NPORS				G
					900.3	2 900.3		P01.6	P01.5		V00 / V005		VSS	VSS	VSS	VSS		V00		P21.7	P21.6		P20.2	P20.0	н
					900.	4 P00.5		P00.6	P01.7	ļ	VSS	VSS		VSS	VSS		VSS	VSS		тск	P21.1		P21.3	P21.5	2
					K 900.	7 900.9		P00.8	P00.10		VSS	VSS	VSS	VSS	VSS	VSS	VSS	VDOP		THS	P21.0		P21.2	F21.4	ĸ
					P00.1	1 900.1	2	AN43	A1H2		VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS		P22.10	922.11		TRST	VSS	L .
					ANH	5 ANH7	<u> </u>		ANHO		VSS	VSS		VSS	VSS		VSS	VSS		P22.8	P22.9		(TAL2	CTAL1	м
					AN			AN06	P40.8 A1438		VDD		VSS	VSS	VSS	VSS		VDD		P22.6	P22.7		V00	V00P3	N
					ANG	_		P40.4 AN32	A184			V00	VSS	VSS	VSS	VSS	V00			P22.4	P22.5		P22.1	P22.0	
					PH0.5 ANG			ANG1	A103											P23.7	P23.6		P22.3	P22.2	R
					T VARE	VAGN 2	9	AN30	A1422	AN15	AN12	ANS	A114	AND	veves	P34.2	P34.4	P33.14	P32.5	VSS	P23.5		P23.3	P23.4	T
						9 A1128	1	NC	A3127	AN14	AN9	AN7	NВ	ANI	P34.1	P34.3	P34.5	P33.15	P32.6	P32.7	VSS		P23.1	P23.2	U
Pin Conflicts			V		V PH0.3 ANZ	1 P40.2 7 A1126																	VERT	P23.0	v
0 items					v P40.1 AN2	1 P40.0 5 AN24	AN19	AN1S	AN16	AN13	AN11	ANS	AM2	P33.0	P33.2	P33.4	P33.6	P33.8	P33.10	P33.12	VGATE 1P	P32.4	VSS	VEXT	w
Description	Module	Pin	Location			A/Q1			VDDM	1	VAGNE 1	ANILO	ANS	P33.1					_		P32.0		P32.3	VSS	۲
				_ -	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
					Con	nection	status					TC27	xC -	BGA2	292 (T	op V	iew)								
				_		Error			Wa	erning			ок												
•				F																					

2.2.1. 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

👩 myproject.pincfg 🛛			- 0
TASKING Pin Mapper - TC27	7xC / BGA292		1a 🕑 🖆 🋵 - 🚉 🤱
Pin Selection	Pin Configuration		
type filter text 🖉 🕀 🕀			a a
 ✓ Ports Analog ✓ P00 ✓ P00.0 P00.1 P00.2 P00.3 P00.4 P00.5 P00.6 P00.7 P00.8 P00.9 P00.10 P00.11 P00.12 P01 P00.12 P01 P01 P01 P01 P11 P12 P13 P14 P15 P20 P21 P21 P22 P23 P32 P33 	Module name: Alias: Description: Pad class: Reset state: Power domain: Symbolic name: Comment: Direction: Pad level: Pad level: Pad strength: Chip input Mode: IN:	P00.0 P00_0 Port 0 pin 0 MP PUL VEXT GTM0_TIM2_CH0_IN	•
 P34 P40 Peripherals Other Pins 	Chip output Mode: OUT:	CCU61.TRAP	

Peripheral example

🔒 myproject.pincfg 🛛			- 8
TASKING Pin Mapper - TC27	xC / BGA292		12 🖻 🖻 🌽 🗕 🛍 🖄
Pin Selection	Pin Configuration		
type filter text 🖉 🕀 🕀			1
 ▷ ✓ Ports ▲ ✓ Peripherals ▷ ADC 	Description: Cor	N0_NODE0 htroller Area Network Module 0	
ASCLIN0 ASCLIN1	Input RXD: v P0.	de 0 2_1.IN ⊂>	
 ASCLIN3 ✓ CAN 	•	2_1.IN ➡ ↔	
CAN0 ✓ CAN0_NODE0 CAN0_NODE1 GPT PAU PAU PAU PAU PAU PAU		ect	
QSPI1 QSPI2 QSPI3 QSPI3 SCU SFNT			

The following toolbar icons are available:

lcon	Action	Description
%	Solve Pin Conflicts	Solve conflicts with pin mappings. See Section 2.2.3, <i>Pin Conflicts View</i> .
c	Generate Code	Generate the source code and add it to your project. See Section 2.5, <i>Generating and Using the Source Code</i> .
	Generate CSV File	Save the pin mapper configuration in a file with comma-separated values (CSV). See Section 2.6, <i>Generate CSV File</i> .
28	Select Device/Package	Select another package for your device. For example, from BGA to QFP or vice versa.
<mark>a</mark>],	Import Pin Configuration	Import pin configurations from a file and merge it with the current configuration.
A	Launch Altium Designer	Launch Altium Designer with the pin configurations. See Section 2.7, <i>Launch Altium Designer</i> .

Select Device/Package

With the Select Device/Package drop-down menu (2) you change the package for your device. For example:

- 8
9a 🖻 🖆 🋵 🗝 <table-cell> 🦓</table-cell>
✓ TC27xC/BGA292
TC27xC/QFP176
TC27xD/BGA292
TC27xD/QFP176

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.

辺 Impor	t Pin Configurations from File	
File:		
?	OK Cancel Workspace	File System

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 leafs 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 2.3, *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 in button to lock the current settings. You can always click the abutton to unlock the settings again.

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.

2.2.2. 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.

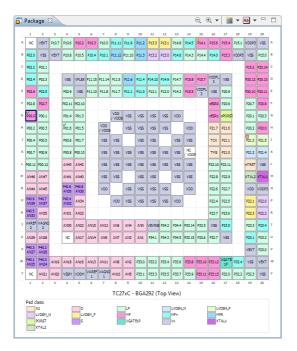
Zoom in / Zoom out

With e and e you can zoom in or zoom out on the package view. Use the drop-down menu next to e 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 I to switch color properties. Some examples:



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
•	NC	VEXT	P10.7	P10.6	P10.2	P10.3	P10.0	P11.11	P11.9	P11.2	P13.3	P13.1	P14.8	P14.5	P14.1	P15.6	P15.4	P15.1	VDDP3	VSS
3	P02.0	VSS	VEXT	P10.8	P10.5	P10.4	P10.1	P11.12	P11.10	P11.3	P13.2	P13.0	P14.6	P14.3	P14.4	P14.0	P15.3	VDDP3	VSS	P15.0
	P02.2	P02.1																	P15.2	P20.14
	P02.4	P02.3		VSS	VFLEX	P11.15	P11.14	P11.5	P11.6	P11.4	P14.10	P14.9	P14.7	P15.8	P15.7	VDDFL 3	VSS		P20.12	P20.13
	P02.6	P02.5		P02.9	VSS	P11.13	P11.8	P11.7	P11.1	P11.0	P12.1	P12.0	P14.2	P15.5	VDDFL 3	VSS	P20.9		P20.10	P20.11
	P02.8	P02.7		P02.11	P02.10											nESR0	P20.6		P20.7	P20.8
	¥00.0	P00.1		P01.4	P01.3			VDD / VDDS	VSS	VSS	VSS	VSS	VDD			nESR1	nPORS		P20.1	P20.3
•	P00.2	P00.3		¥01.6	P01.5		VDD / VDDS		VSS	VSS	VSS	VSS		VDD		P21.7	P21.6		P20.2	P20.0
,	P00.4	P00.5		P00.6	P01.7		VSS	VSS		VSS	VSS		VSS	VSS		тск	P21.1		P21.3	P21.5
¢	P00.7	P00.9		P00.8	P00.10		VSS	VSS	VSS	VSS	VSS	VSS	VSS	NC VOOP		TMS	P21.0		P21.2	P21.4
	P00.11	P00.12		ANH3	AN42		VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS		P22.10	P22.11		nTRST	VSS
i	AN46	AN47		AN41	AN40		VSS	VSS		VSS	VSS		VSS	VSS		P22.8	P22.9		XTAL2	XTAL
•	AN44	ANHS		P40.6 AN36	P40.8 AN38		VDD		VSS	VSS	VSS	VSS		VDD		P22.6	P22.7		VDD	VOOPS
,	P40.9 AN39	P40.7 AN37		P40.4 AN32	AN34			VDD	VSS	VSS	VSS	VSS	VDD			P22.4	P22.5		P22.1	P22.0
ł	P40.5 AN33	AN35		ANB1	AN23					_						P23.7	P23.6		P22.3	P22.2
r	VAREF	VAGNO		ANBO	AN22	AN15	AN12	ANS	ANH	AND	VEVRS	P34.2	P34.4	P33.14	P32.5	VSS	P23.5		P23.3	P23.4
,	A1(29	AN28		NC	AN17	AN14	AN9	AN7	ANB	AN1	P34.1	P34.3	P34.5	P33.15	P32.6	P32.7	VSS		P23.1	P23.2
v	P40.3 ANZ7	P40.2 AN26					_							_			_		VEXT	P23.0
ł	P40.1 AN25	P40.0 AN24	AN19	AN18	AN16	AN13	AN11	ANS	AN2	P33.0	P33.2	P33.4	P33.6	P33.8	P33.10	P33.12	VGATE 1P	P32.4	VSS	VEXT
Y	NC	AN21	AN20	VSSM	VDDM	VAREF	VAGNO	AN10	ANS	P33.1	P33.3	P33.5	P33.7	P33.9	P33.11	P33.13	P32.0	P32.2	P32.3	VSS
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
								TC2	7xC -	BGA	292 (Гор V	iew)							
		r doma /DDM	in:	ſ	VD	093			VEXT				VFLEX							

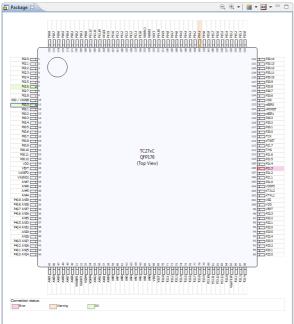
	JUNI	ige S	~ ``							_	_				୍	₽, -			Ab 🔻	
r	1 NC	2 VEXT	3 P10.7	4 P10.6	5 P10.2	6 P10.3	7	8	9 P11.9	10	11 P13.3	12 P13.1	13 P148	14 P14.5	15 0 14 1	16 P15.6	17 P15.4	18 P15.1	19 VDDP3	20 VSS
Ļ		_						P11.11		P11.2							_	_		
1	02.0	VSS	VEXT	P10.8	P10.5	P10.4	P10.1	P11.12	P11.10	P11.3	P13.2	P13.0	P14.6	P14.3	P14.4	P14.0	P15.3	VDDP3	VSS	P15.0
•	02.2	P02.1																	P15.2	P20.14
8	02.4	P02.3		VSS	VFLEX	P11.15	P11.14	P11.5	P11.6	P11.4	P14.10	P14.9	P14.7	P15.8	P15.7	VDDFL 3	VSS		P20.12	P20.13
5	02.6	P02.5		P02.9	VSS	P11.13	P11.8	P11.7	P11.1	P11.0	P12.1	P12.0	P14.2	P15.5	VDDFL 3	VSS	P20.9		P20.10	P20.11
F	02.8	P02.7		P02.11	P02.10											nESRO	P20.6		P20.7	P20.8
F	00.0	P00.1		P01.4	P01.3			VDD	VSS	VSS	VSS	VSS	VDD	1		nESR1	nPORS		P20.1	P20.3
F	00.2	P00.3		P01.6	P01.5		VDD	<u> </u>	VSS	VSS	VSS	VSS		VDD		P21.7	P21.6		P20.2	P20.0
•	00.4	P00.5		P00.6	P01.7		VSS	VSS		VSS	VSS		VSS	VSS		тск	P21.1		P21.3	P21.5
•	00.7	P00.9		P00.8	P00.10		VSS	VSS	VSS	VSS	VSS	VSS	VSS	NC / VOOF		TMS	P21.0		P21.2	P21.4
P	00.11	P00.12		ANH3	A1142		VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS		P22.10	P22.11		nTRST	vss
4	0146	AN47		AN41	AN40		VSS	VSS		VSS	VSS		VSS	VSS		P22.8	P22.9		XTAL2	XTAL1
4	0144	ANHS		P40.6 AN36	P40.8 AN38		VDD		VSS	VSS	VSS	VSS	<u> </u>	VDD		P22.6	P22.7		VDD	VDDP3
	40.9 N(39	P40.7 AN37		P40.4 AN32	AN34		_	VDD	VSS	VSS	VSS	VSS	VDD	1		P22.4	P22.5		P22.1	P22.0
P	40.5	ANBS		ANB1	AN23			_								P23.7	P23.6		P22.3	P22.2
		VAGND		AN30	AN22	AN15	AN12	ANS	AN4	AND	VEVRS	P34.2	P34.4	P33.14	P32.5	VSS	P23.5		P23.3	P23.4
4	1129	AN28		NC	AN17	AN14	AN9	AN7	ANB	ANI	P34.1	P34.3	P34.5	P33.15	P32.6	P32.7	VSS		P23.1	P23.2
	40.3 //27	P40.2 AN26					_			_									VEXT	P23.0
P	40.1 N25	P40.0 4N24	AN19	AN18	AN16	AN13	AN11	ANS	AN2	P33.0	P33.2	P33.4	P33.6	P33.8	P33.10	P33.12	VGATE 1P	P32.4	vss	VEXT
F	NC	AN21	AN20	VSSM	VDDM	VAREF	VAGNO	AN10	ANS	P33.1	P33.3	P33.5	P33.7	P33.9	P33.11	P33.13	P32.0	P32.2	P32.3	VSS
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
								TC2	7xC -	BGA	292 (Гор V	iew)							
R		state: U			EV PU	R13 SMF	is -> PC	o, GF	High				op			PC)			

Labels

With the drop-down menu next to M 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:



2.2.3. 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 🛛				~ - 8
1 error, 1 warning, 0 others				
Description	Module	Pin	Location	Resource
Ø 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 2.4, Solving Pin Conflicts for information how to solve pin conflicts.

See Section 2.9, Errors and Warnings for a list of errors and warnings.

2.3. 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 C/C++ Projects 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.

6. (Optional) When you want to prevent that these settings can be changed, click if to lock the settings.

A little lock appears next to the port pin.

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

When you click vous a light blue 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

- 1. In the C/C++ Projects view double-click on a configuration file (myproject.pincfg) to view its contents.
- 2. In the Pin selection pane, select a module. For example, Peripherals » ETH » ETH0.
- 3. In the Pin configuration pane select the pin you want to assign to the virtual pin.

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

- 4. Click to follow the connection to the port pin side. The corresponding selection is shown in a light blue color.
- 5. (Optional) When you want to prevent that these settings can be changed, click if to lock the settings.

A little lock appears next to the port pin.

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

2.4. 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

1. From the **Pin Mapper** menu, select **Solve Pin Conflicts** (¹/₂₀).

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

	u want to continue?		olve the dangling conn		
Action	Module	Pin	Old port Pin	New port Pin	
move		GPIO	P02.2	P02.0	
move	ASCHS1	TXD	P15.1	P02.2	
connect	ERUIN7	IN	P20.9	P15.1	
move	T12CCU1	CC62	P11.10	P33.9	
connect	ASCLIN1	RXD	P15.1	P11.10	
connect	ASCLIN1	TXD	P15.1	P15.5	
connect	CAN0_NODE2	RXD	P15.1	P02.3	
move	G0_QREQ	BFL	P33.5	P33.6	
reconnect	T12CCU1	CC60	P33.5	P33.5	
move	SCU	EXTCLK1	P33.10	P11.12	
reconnect	GTM0_CMU	CLK1	P33.10	P33.10	
move	GTM0_TIM0_C	IN	P02.4	P21.6/TDI	
connect	QSPI3_S	SELIN	P33.10	P02.4	
connect	QSPI0_M	CLK	P20.13	P20.11	
move	QSPI0_M	RXD(N)	P20.12	P14.6	
	TIRCOUR	COUTCI	000.10	000.10	

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	The action that takes place to solve the pin conflict. move - 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. connect - 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.
	reconnect - 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. reset - reset the port pin to the default settings.
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.

2.5. 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.

- myproject [Active Debug]
 - Includes
 - 🔺 📂 PinMapper
 - b aurix_pin_mappings.c
 - In aurix_pin_mappings.h
 - cstart_tc1.c
 - h cstart_tc1.h
 - b cstart_tc2.c
 - b h cstart_tc2.h
 - cstart.c
 - b h cstart.h
 - b ic myproject.c
 - b sync_on_halt.c
 - DConfig
 - MConfig
 - myproject.launch
 - myproject.lsl
 - myproject.pincfg

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.

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;
}
```

2.6. 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 (1).

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.

2.7. 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.

2.8. 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**.

2.9. 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.

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 two use two modules of which only one module can be used at a time.

For example,

- 1. Select Peripherals » ASCLINO » SPI3WO and change the input RXD(N) from None to P14_1.IN.
- 2. Select Peripherals » ASCLINO » 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.