

SCU_Clock_1 for KIT_AURIX_TC297_TFT

Clock configuration via SCU

AURIX™ TC2xx Microcontroller Training
V1.0.0



[Please read the Important Notice and Warnings at the end of this document](#)

Scope of work

The clock system is configured based on a PLL frequency of 200 MHz and the clock signal is provided at an external port pin.

The Clock Control Unit, which is part of the System Control Units (SCU), is used to configure the PLL clock. This clock signal is routed to an external clock output pin, which can be observed with an oscilloscope.

Introduction

- › The Clock Control Unit (CCU) controls the clock system and contains different blocks:
 - Basic clock generation
 - Clock speed upscaling
 - Clock distribution
 - Individual clock configuration

- › The Clock Generation Unit (CGU) is part of the CCU and allows a flexible clock generation.

- › Phase Lock Loops (PLLs) are provided for upscaling the clock frequency from an internal or external oscillator.

- › The System Peripheral Bus (SPB) is used to enable the Fractional Divider (FDR) to divide the source clock.

Hardware setup

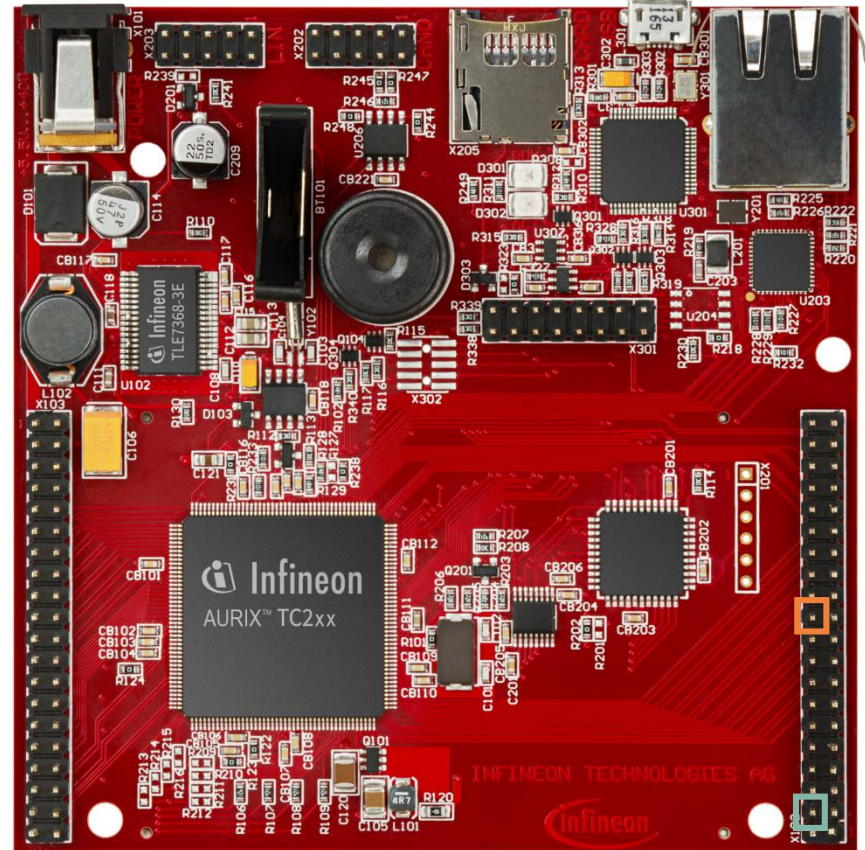
This code example has been developed for the board KIT_AURIX_TC297_TFT_BC-Step.

An oscilloscope is needed to observe the clock at the pin P23.1. Connect the clock and the ground to the oscilloscope.

	X102		
P14.5	40	39	P14.4
P20.10	38	37	P20.9
P15.7	36	35	P15.6
P15.5	34	33	P15.4
P15.3	32	31	P15.2
P22.3	30	29	P22.2
P22.1	28	27	P22.0
P33.11	26	25	P23.4
P23.3	24	23	P23.2
P23.1	22	21	P23.0
P33.6	20	19	P33.8
P33.12	18	17	P33.1
P33.2	16	15	P33.3
P33.4	14	13	P33.5
AN0	12	11	AN8
AN2	10	9	AN3
AN32	8	7	AN33
AN20	6	5	AN21
GND	4	3	GND
V_UC(+5V)	2	1	VCC_IN

Clock

Ground



Implementation

Configuring System Control Units

Configuration of the System Control Unit (SCU) is done once in the setup phase by calling the initialization function ***initScuClock()***, which contains the following steps:

- › Create an instance of the ***IfxScuCcu_Config*** structure which contains the CCU configuration.
- › Initialize the configuration by calling the iLLD function ***IfxScuCcu_initConfig()***.
- › Calculate the PLL dividers through the iLLD function ***IfxScuCcu_calculateSysPlldDividers()*** and set the desired frequency as parameter (PLL frequency is 200 MHz for this example).
- › Set SPB frequency to the desired value by calling the function ***IfxScuCcu_setSpbFrequency()*** (SPB frequency is 100 MHz for this example).
- › Initialize the CCU with the iLLD function ***IfxScuCcu_init()***.

The ***initScuClock()*** function is contained in the ***SCU_Clock.h***, while the other functions are part of the iLLD header ***IfxScuCcu.h***.

Implementation

Configuring SCU Clock output

Configuration of the SCU clock output is done once in the setup phase by calling the initialization function ***outputConfigScuClock()***, which contains the following steps:

- › Call the iLLD function ***IfxScuWdt_clearSafetyEndinitInline()*** to disable the Safety Endinit protection in order to modify the SCU register.
- › Set ***SCU_CCUCON0.B.CLKSEL*** to 0x1 to select PLL as clock source.
- › Set ***SCU_EXTCON.B.EN0*** and ***SCU_EXTCON.B.SELO*** to enable and select output frequency (f_{OUT}) as external source (***SCU_EXTCON.B.EN0*** = 0x1, ***SCU_EXTCON.B.SELO*** = 0x0).
- › Set ***SCU_FDR.B.STEP*** to the desired reload value ([see page 8](#)) and ***SCU_FDR.B.DM*** to choose the normal divider mode.
- › Call the iLLD function ***IfxScuWdt_setSafetyEndinitInline()*** to reenale the Safety Endinit protection.

The functions ***IfxScuWdt_clearSafetyEndinitInline()*** and ***IfxScuWdt_setSafetyEndinitInline()*** are contained in iLLD header ***IfxScuWdt.h***.

Implementation

Configuring port pin

Configuration of the port pin is done as well in the function ***outputConfigScuClock()*** with the following steps:

- › Call the iLLD function ***IfxPort_setPinMode()*** with ***IfxPort_Mode_outputPushPullAlt6*** as parameter to select SCU as output.
- › Set pin pad driver to increase pin's speed by calling the iLLD function ***IfxPort_setPinPadDriver()*** and setting ***IfxPort_PadDriver_cmosAutomotiveSpeed1*** as parameter.

The functions ***IfxPort_setPinMode()*** and ***IfxPort_setPinPadDriver()*** are contained in iLLD header ***IfxPort.h***.

Implementation

Step Value calculation example

The Step value in normal divider mode is defined according to the following formula:

$$f_{OUT} = \frac{f_{SPB} * \frac{1}{n}}{2}, \text{ with } n = 1024 - Step$$

$$\text{Therefore, } Step = 1024 - \frac{f_{SPB}}{2 * f_{OUT}}$$

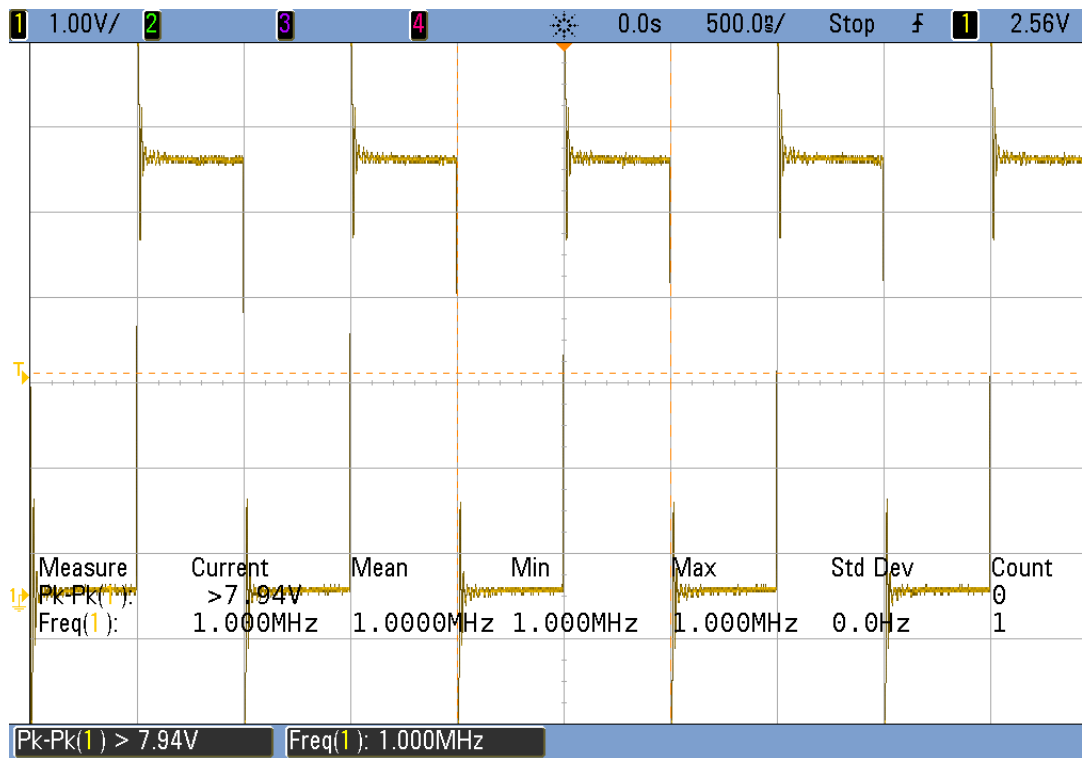
In this example, the desired external frequency value (f_{OUT}) is 1 MHz, and f_{SPB} is 100 MHz.

After calculation, the Step Value is 974 (0x3CE).

Run and Test

After code compilation and flashing the device, perform the following steps:

- › Connect the oscilloscope to the board via port pin P23.1 and ground.
- › Observe the desired clock on the oscilloscope's screen.



References



- > AURIX™ Development Studio is available online:
- > <https://www.infineon.com/aurixdevelopmentstudio>
- > Use the „*Import...*“ function to get access to more code examples.



- > More code examples can be found on the GIT repository:
- > https://github.com/Infineon/AURIX_code_examples



- > For additional trainings, visit our webpage:
- > <https://www.infineon.com/aurix-expert-training>



- > For questions and support, use the AURIX™ Forum:
- > <https://www.infineonforums.com/forums/13-Aurix-Forum>

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2020-02

Published by

Infineon Technologies AG
81726 Munich, Germany

© 2020 Infineon Technologies AG.
All Rights Reserved.

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

SCU_Clock_1_KIT_TC297_TFT

IMPORTANT NOTICE

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics (“Beschaffenheitsgarantie”).

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer’s compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer’s products and any use of the product of Infineon Technologies in customer’s applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer’s technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies office (www.infineon.com).

WARNINGS

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies’ products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.