

ADC_Group_Scan_1 for KIT_AURIX_TC297_TFT

ADC group scan source

AURIX™ TC2xx Microcontroller Training
V1.0.3



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

Scope of work

The Versatile Analog-to-Digital Converter (VADC) is configured to measure multiple analog signals in a sequence using scan request.

The Auto Scan mode of the Versatile Analog to Digital Converter (VADC) module is enabled over the channels 0 to 4 of the Group 0. Additionally, the converted results are checked with the Valid Flag (VF) of the Result Register, and then sent to the PC via UART.

Introduction

- › An analog to digital converter is a module that converts an analog signal, such as a voltage sent by an analog sensor, into a digital signal
- › The Versatile Analog to Digital Converter module (VADC) of the AURIX™ TC29x comprises 11 independent analog to digital converters. Each of them can do conversions with a resolution of up to 12-bits
- › Analog/Digital conversions can be requested by several request sources:
 - **Queued scan** request source, specific to a single group
 - Channel scan request source, which comprises:
 - **Group scan** source, specific to a single group
 - **Background scan** source, which can request all channels of all groups
- › Each request source can trigger up to three conversion modes:
 - **Fixed Channel Conversion**
 - **Channel Sequence Conversion**
 - **Auto Scan Conversion**
- › The **Auto Scan Conversion** mode enables the conversion of a configurable linear sequence of all available channels and it can only be triggered by the **Group Scan Request Source** or by the **Background Scan Request Source**

Hardware setup

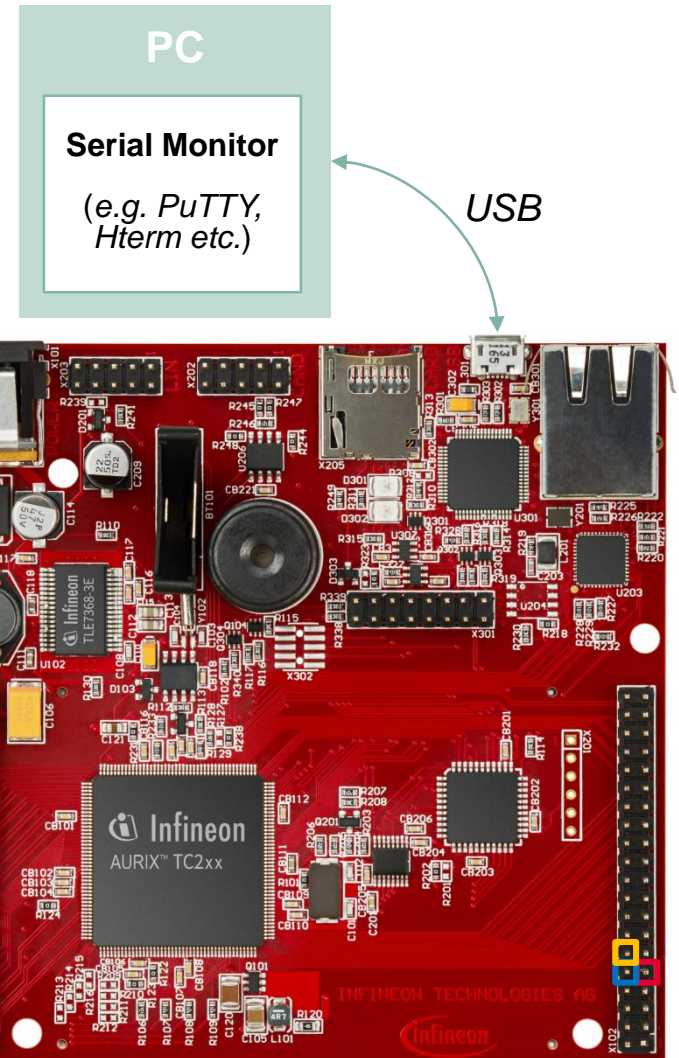
This code example has been developed for the board KIT_AURIX_TC297_TFT_BC-Step.

The board should be connected to the PC via USB, in order to allow UART communication.

	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

Group 0 - Channel 0
Group 0 - Channel 2

Group 0 - Channel 3



Note: For this Application Kit, not all pins are available. Therefore, the AN1 and AN4 pins of this example cannot be used.

Implementation

Configuration of the VADC module

This example performs Auto Scan conversions over the channels 0 to 4 of the group 0.

› *init_vadc()*

In this function, the VADC module is initialized and configured. Then, Group4 is selected and configured. For the conversions to be done repeatedly, the **autoscanEnabled** parameter of the **adcGroupConfig** has to be set.

The following iLLD functions are used in the initialization:

- ***IfxVadc_Adc_initModuleConfig()*** initializes the module structure with the default values
- ***IfxVadc_Adc_initModule()*** initializes the VADC to run with the expected frequency and calibration
- ***IfxVadc_Adc_initGroupConfig()*** initializes the group buffer with the default configuration
- ***IfxVadc_Adc_initGroup()*** initializes the VADC group specified in the parameters

A loop over the channels is needed to configure, initialize and set them up. Then, the ADC can be started. The initialization and configuration of each channel is done via the two iLLD functions:

- › ***IfxVadc_Adc_initChannelConfig()***, which initializes the buffer with the default channel configuration
 - › ***IfxVadc_Adc_initChannel()***, which initializes the specified channel with the user configuration
- Therefore, the channels are added to the scan sequence via ***IfxVadc_Adc_setScan()*** and the conversions start by calling the function ***IfxVadc_Adc_startScan()***.

The iLLD functions mentioned above can be found in the header file ***IfxVadc_Adc.h***.

Implementation

Configuration of the VADC module

› *run_vadc()*

In this function, a loop is performed in order to check that new results are available and that they have not been read before.

The iLLD functions mentioned above can be found in the header file ***IfxVadc_Adc.h***.

Finally, the ***send_vadc()*** function sends the result of the conversions of channels 4 to 7 to the serial monitor via UART.

Implementation

Configuration of the UART

In this tutorial, the UART connection is used to make the debugging more convenient and easier to understand. Since Auto Scan mode of the ADC is enabled, a new conversion is started as soon as the previous one is completed. It is then easier to debug and visualize the results if they are sent only when the user requests them.

› *init_uart()*

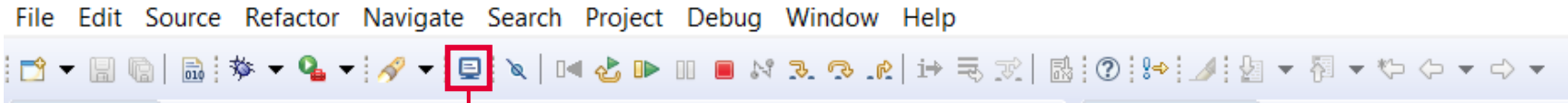
The iLLD function ***IfxAsclin_Asc_initModuleConfig()*** fills the configuration **ascConf** structure with the default values. Then, the parameters are set, depending on the needed connection: baudrate, Tx and Rx buffer size, Tx and Rx pin configuration etc.

Finally, ***IfxAsclin_Asc_initModule()*** initializes the module with the user configuration previously done.

The iLLD functions mentioned above can be found in the files ***IfxAsclin_Asc.h***.

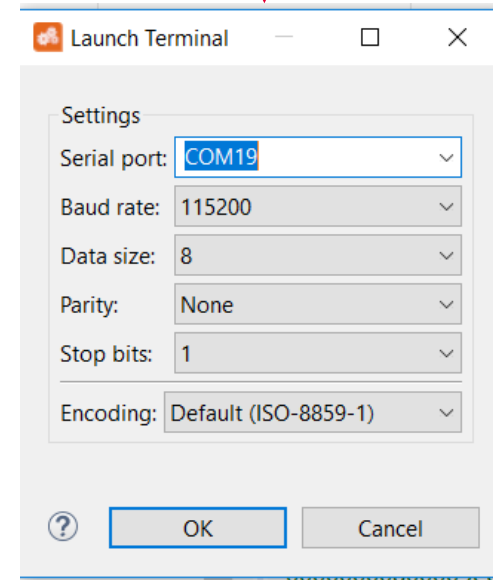
Run and Test

- > For this training, a serial monitor is required for visualizing the values. The monitor can be opened inside the AURIX™ Development Studio using the following icon:



- > The serial monitor must be configured with the following parameters to enable the communication between the board and the PC:

- Speed (baud): 115200
- Data bits: 8
- Stop bit: 1



Run and Test

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

- › Open the serial monitor and start the serial communication, linked with the appropriate COMx port (this can be checked in the Device Manager)
- › In order not to overload the readout due to the endless repeated conversions, results are sent out by the AURIX™ only when the user sends the character “1”



- › The 12-bits integer value, in decimal format (0 - 4095 range), of each channel is then displayed, right after conversion

Note: If no signal is connected to the pins, the shown values represent random analog noise.

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>

Revision history

Revision	Description of change
V1.0.3	Restructured init_vadc() and run_vadc() description
V1.0.2	Update of version to be in line with the code example's version
V1.0.1	Run and Test section updated with the new terminal feature and a different baud rate
V1.0.0	Initial version

Trademarks

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

Edition 2021-06

Published by

**Infineon Technologies AG
81726 Munich, Germany**

**© 2021 Infineon Technologies AG.
All Rights Reserved.**

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

**ADC_Group_Scan_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.