ADC_Queued_Scan_1 for KIT_AURIX_TC275_LK

ADC queued source

AURIX™ TC2xx Microcontroller Training
V1.0.0

Scope of work

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

The Queued Request of the Versatile Analog-to-Digital Converter (VADC) module is used to continuously scan the analog inputs channels 5, 6 and 7 of group 4.

Introduction

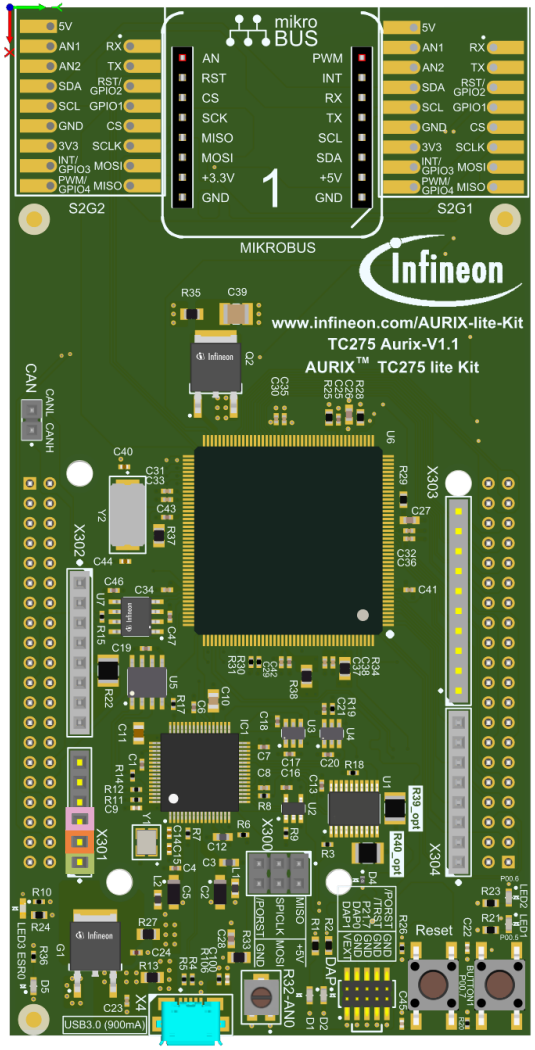
- The Versatile Analog-to-Digital Converter module (VADC) of the AURIX™ TC27x comprises 8 independent analog to digital converters (VADC groups) with up to 8 analog input channels each
- Each channel can convert analog inputs with a resolution of up to 12-bit.
- Analog/Digital conversions can be requested by several request sources:
  - **Queued 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
- A queued source can issue conversion requests for an arbitrary sequence of input channels. The channel numbers for this sequence can be freely programmed
- A queued source converts a series of input channels permanently (using the refill option) or on a regular time base

Hardware setup

This code example has been developed for the board KIT_AURIX_TC275_LITE.

The signals to be measured have to be connected to channels 5, 6 and 7 of the group 4 of the VADC (pins AN37, AN38, AN39).

	X301
(P40.1) AN25	6
(P40.0) AN24	5
(P40.6) AN36	4
(P40.7) AN37	3
(P40.8) AN38	2
(P40.9) AN39	1



Note: The reference voltage (VAREF) of the EVADC on the board KIT_AURIX_TC275_LITE is 3.3 V.

Implementation

Configuration of the VADC

The configuration of the VADC is done in the ***initVADC()*** function in four different steps:

- Configuration of the **VADC module**
- Configuration of the **VADC group**
- Configuration of the **VADC channels**
- Filling the queue

Configuration of the VADC module with the function *initVADCModule()*

The default configuration of the VADC module, given by the iLLDs, can be used for this example.
This is done by initializing an instance of the ***IfxVadc_Adc_Config*** structure and applying default values to its fields through the function ***IfxVadc_Adc_initModuleConfig()***.
Then, the configuration can be applied to the VADC module with the function ***IfxVadc_Adc_initModule()***.

Implementation

Configuration of the VADC group with the function *initVADCGroup()*

The configuration of the VADC group is done by initializing an instance of the ***IfxVadc_Adc_GroupConfig*** structure with default values through the function ***IfxVadc_Adc_initGroupConfig()*** and modifying the following fields:

- ***groupId*** – to select which converters to configure
- ***master*** – to indicate which converter is the master. In this example, only one converter is used, therefore it is also the master
- ***arbiter*** – a structure that represents the enabled request sources, which can be Group scan, Queue and/or Background sources. In this example, it is set to ***arbiter.requestSlotQueueEnabled***
- ***triggerConfig*** – a parameter that specify the trigger configuration

Then, the user configuration is applied through the function ***IfxVadc_Adc_initGroup()***.

Implementation

Configuration of the VADC channels with the function *initVADCChannels()*

The configuration of each channel is done by initializing a separate instance of the ***IfxVadc_Adc_ChannelConfig*** structure with default values through the function ***IfxVadc_Adc_initChannelConfig()*** and modifying the following fields:

- ***channelId*** – to select the channel to configure
- ***resultRegister*** – to indicate the register where the A/D conversion value is stored

Then, the configuration is applied to the channel with the function ***IfxVadc_Adc_initChannel()***.

Filling the queue

Each channel is added to the queue through the function ***IfxVadc_Adc_addToQueue()***.

When the VADC configuration is done and the queue is filled, the conversion is started with the function ***IfxVadc_Adc_startQueue()***.

Finally, to read a conversion, the function ***IfxVadc_Adc_getResult()*** from iLLDs is used inside the function ***readVADC()***.

All the functions used to get a conversion and configuring the VADC module, its group and channels can be found in the iLLD header ***IfxVadc_Adc.h***.

Run and Test

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

- Run the code and then pause it
- Repeat the above step to see that the result is changing accordingly to the signal you measure:
  - AN37 is g_results[0]
  - AN38 is g_results[1]
  - AN39 is g_results[2]

Variables | Breakpoints | Expressions

Expression	Type	Value
g_results[0].B.RESULT	unsigned short	3398
g_results[1].B.RESULT	unsigned short	4095
g_results[2].B.RESULT	unsigned short	1857
Add new expression		

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 2021-06
Published by
Infineon Technologies AG
81726 Munich, Germany

Do you have a question about this document?
Email: erratum@infineon.com

Document reference
ADC_Queued_Scan_1_KIT_TC275_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.