TLE9893_2QKW62S_ADC1_POTI_SWTRG

About this document

Scope and purpose

The aim of this guide is to present the scope, the implementation, the algorithm and a demonstration of the **TLE9893_2QKW62S_ADC1_POTI_SWTRG** example code for the TLE989x Infineon Embedded Power ICs based on Arm® Cortex® M3. This example code can be found in the Keil µVision Pack Installer.

The full functionalities and characteristics of the embedded power devices are described in the datasheets and user's manual. Please refer to these documents for more detailed information. Furthermore, a low level (line-by-line) description of the code is not the aim of this document, although occasionally some codeblocks might be reported if necessary to the comprehension.

Note: *The following information is given as a hint for the implementation of the system only and shall not be regarded as a description or warranty of a certain functionality, condition or quality of the referred devices or presented software example.*

Intended audience

Design engineers, system engineers, embedded power designers

Table of contents

1 Introduction

In the TLE9893_2QKW62S_ADC1_POTI_SWTRG example, the current Poti voltage setting is read with the channel 0 of ADC1, via the input pin P2.6. The measured voltage range 0 is divided into 10 intervals: 0-500mV, 500mV-1V, ..., 4.5V-5V which are displayed on 9 LEDs. The ADC1 sequence is triggered by software.

Figure 1 shows the signal of pin 2.6 when the Poti is set to its maximum voltage (5V) whereas Figure 2 shows the signal of pin 2.6 when the Poti is switched off (0V).

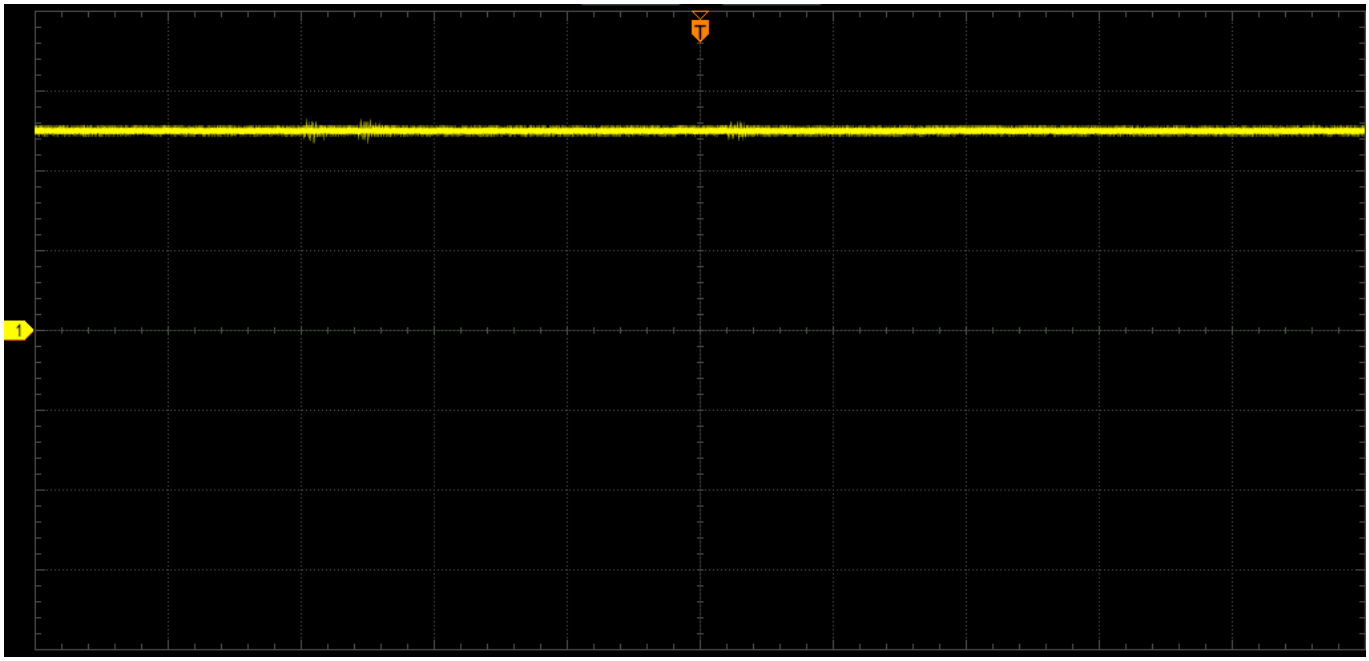

Figure 1 Capture of P2.6 with maximum Poti setting (5V)

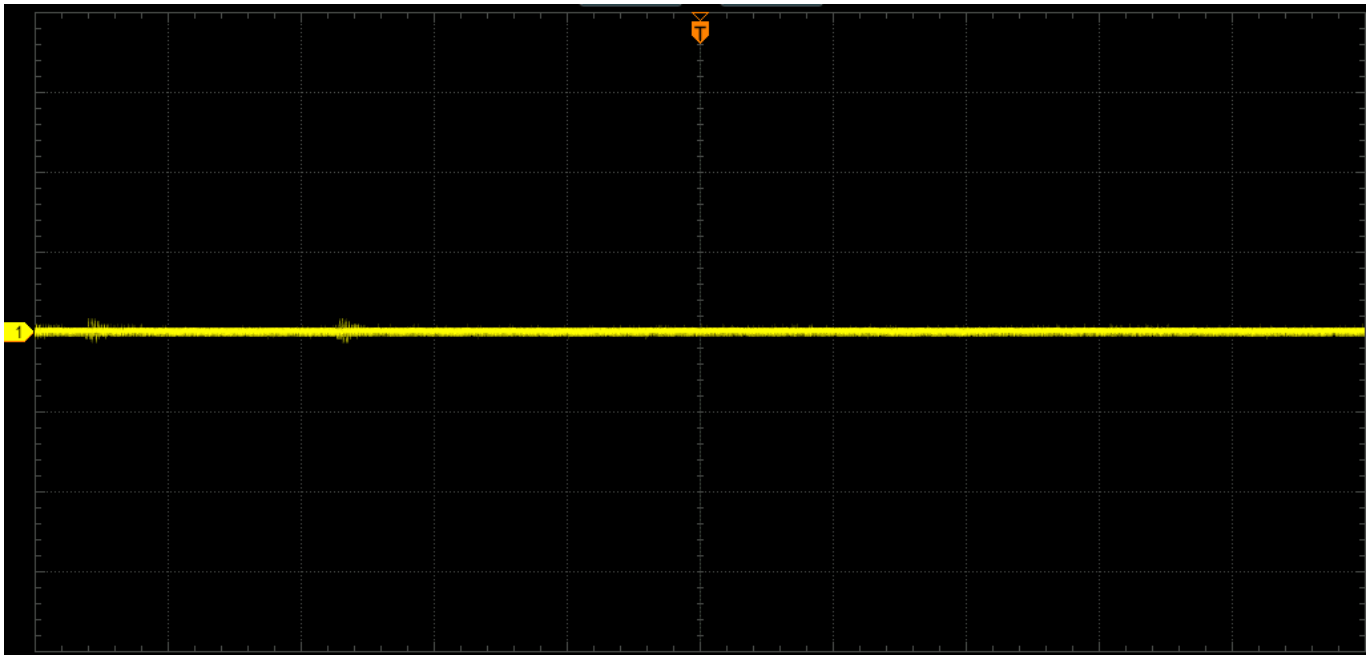

Figure 2 Capture of P2.6 with minimum Poti setting (0V)

2 Hardware

This chapter shows how to run the TLE989X_2QKW62S_ADC1_POTI_SWTRG example with the TLE988X/TLE989x evaluation board. For this the project must be opened and compiled.

Figure 3 shows the TLE988X/TLE989X evaluation board. The application code must be loaded via a debugger (e.g. ULINK or J-Link) to the board. The board must be powered with 12V (red and black connections).

First the Poti is turned to the most right stop (clockwise). In this case all LEDs are switched off.

Then the poti is slowly turned to the left (counter-clockwise). Step by step all 9 LEDs are switched on until the most left stop of the Poti is reached.

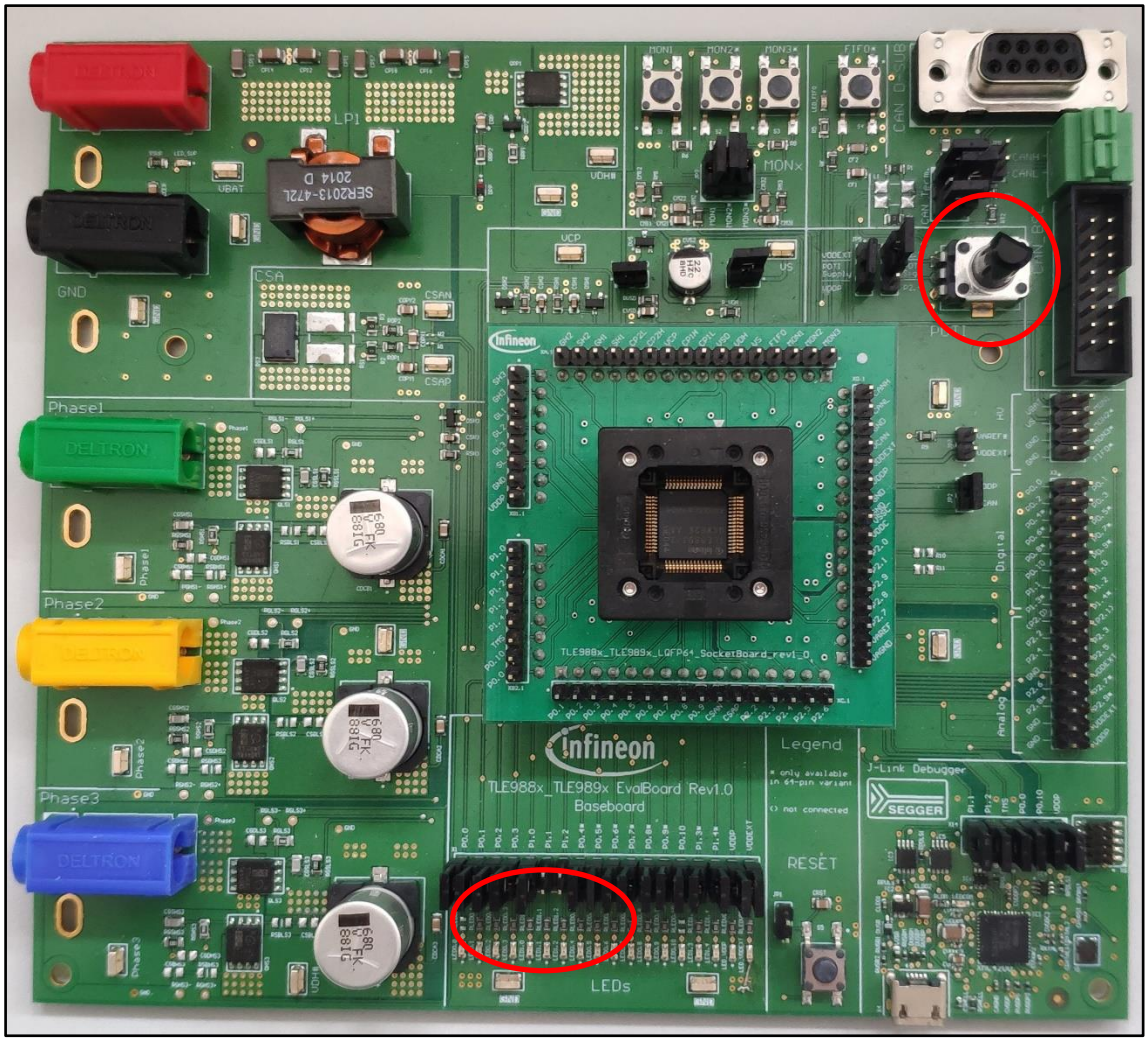

Figure 3 TLE988X/TLE989X evaluation board

3 Implementation

This chapter shows the process to follow to get a working TLE9893_2QKW62S_ADC1_POTI_SWTRG example.

3.1 Get the example via the Pack Installer for Keil

Open the Pack Installer within the Keil IDE. See Figure 4 below.

Choose the appropriate device (here TLE9893_2QKW62S) on the left-hand side. On the right-hand side, select the tab Examples, where you can access the TLE9893_2QKW62S_ADC1_POTI_SWTRG example.

Clicking on “Copy” will copy the example on your computer and open it.

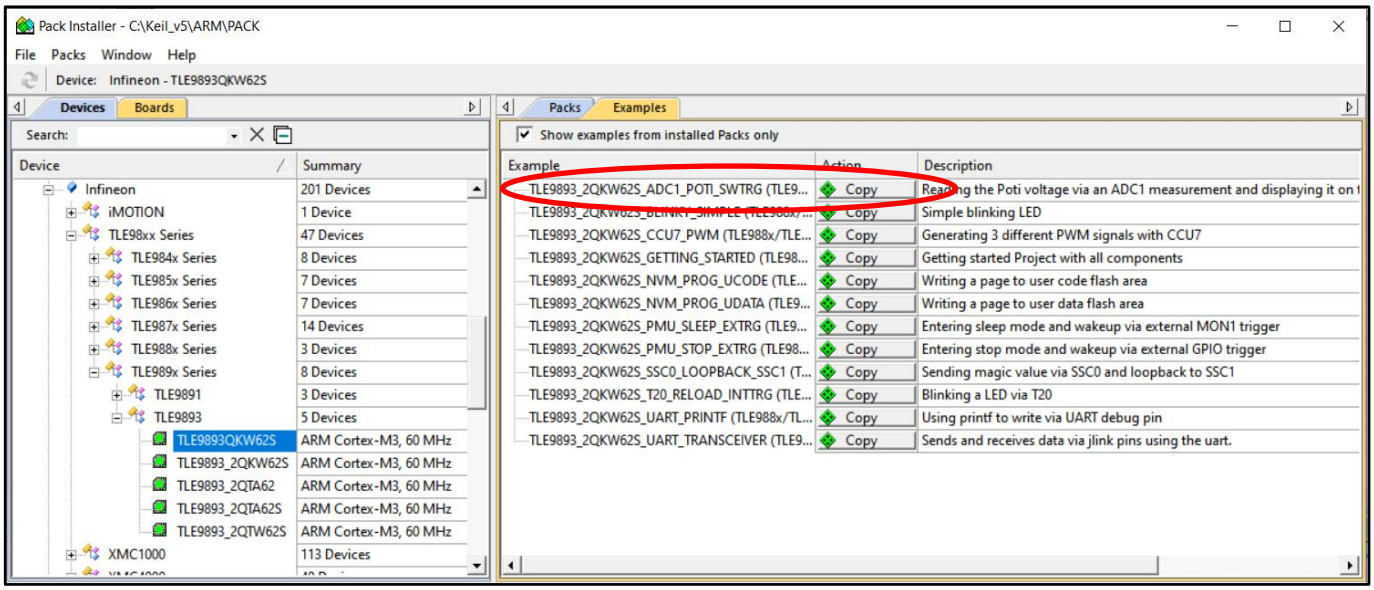

Figure 4 Keil Pack Installer

3.2 Configuration

In order to see the configured output pin for the LED, start the tool Config Wizard. It is available within the Keil IDE through a shortcut in the Tools menu.

The Config Wizard opens and shows an overall status of the current pin configuration. In Figure 5 the pin P 2.6 is configured as an input for the channel 0 of ADC1.

Furthermore, all GPIO pins P0.1 until P0.6 and P1.0 until P1.2 are configured as output pins. The pins P0.4, P0.5 and P0.6 are used for the 64-pin devices only. For the 48-pin devices only six LEDs are used for a range from 0V to 3V.

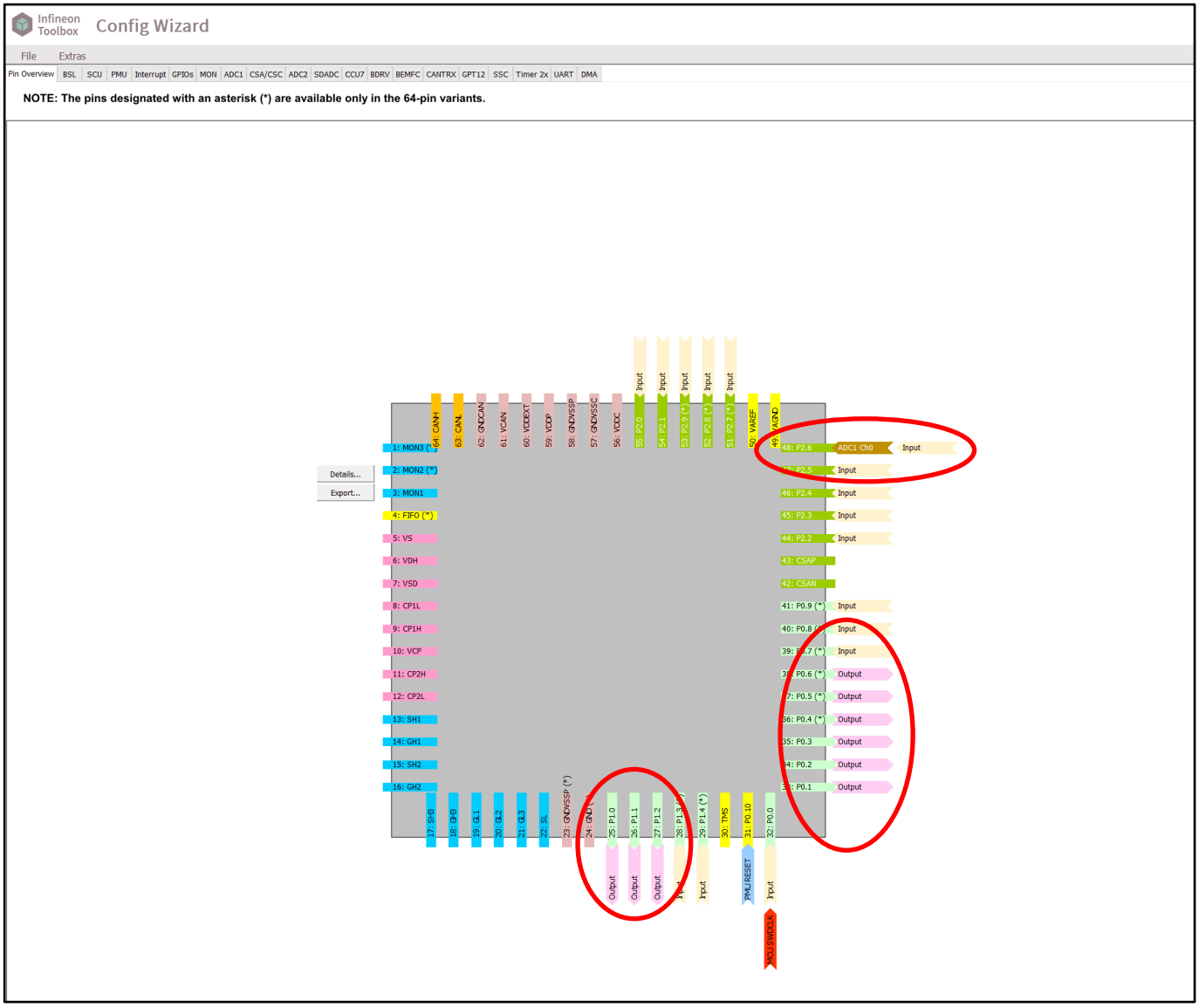

Figure 5 Config Wizard pin overview

In order to configure the ADC1 module for the TLE9893_2QKW62S_ADC1_POTI_SWTRG example, select the ADC1 tab.

The Figure 6 shows that the pin P2.6 is selected as input for the channel 0 in the box Channel Config.

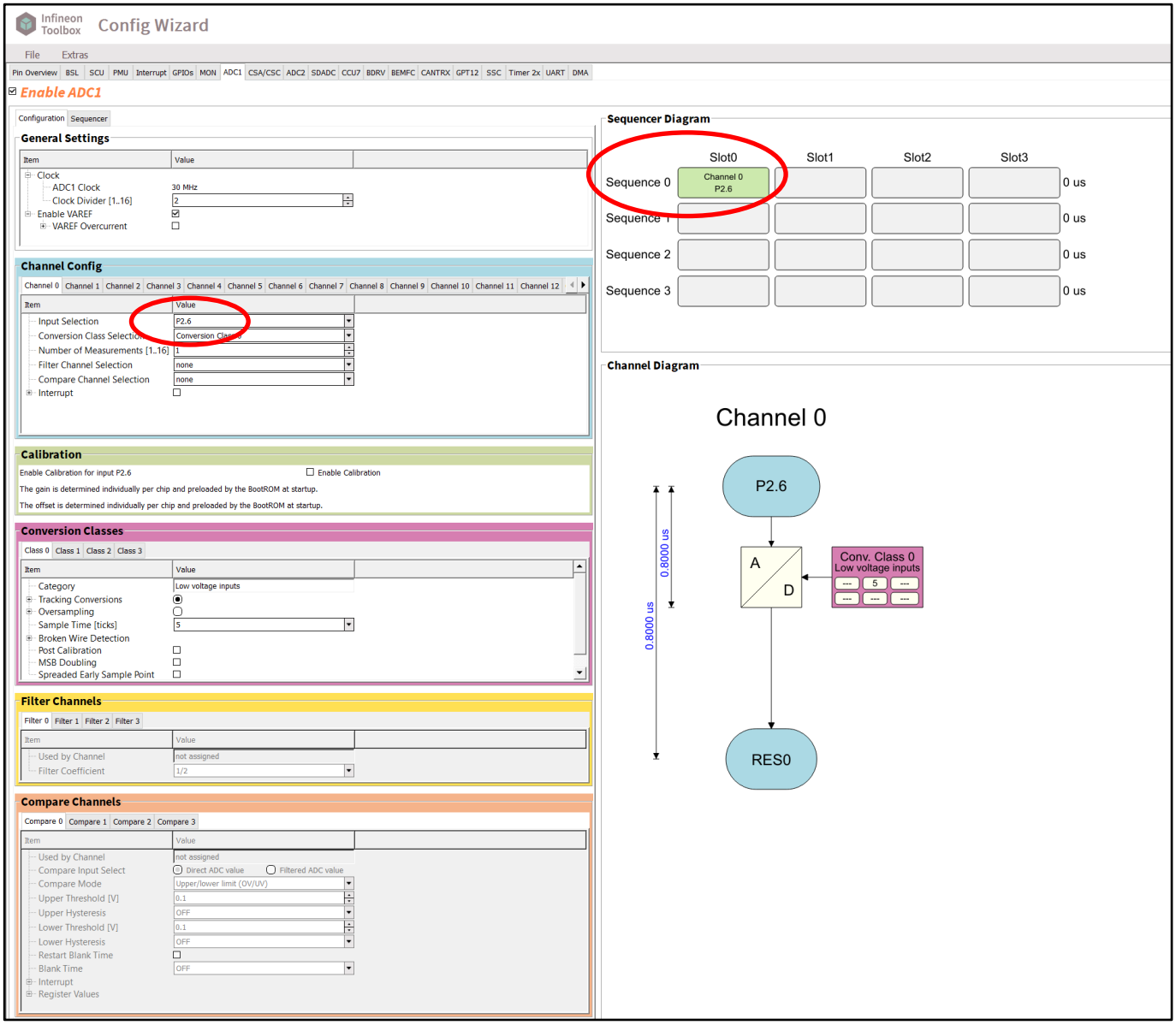

Figure 6 Config Wizard, module ADC1, tab Configuration

Next to the Configuration tab of the ADC1 module, select the Sequencer tab to configure the individual sequences. For the TLE9893_2QKW62S_ADC1_POTI_SWTRG example the software trigger is used as trigger source (see Figure 7).

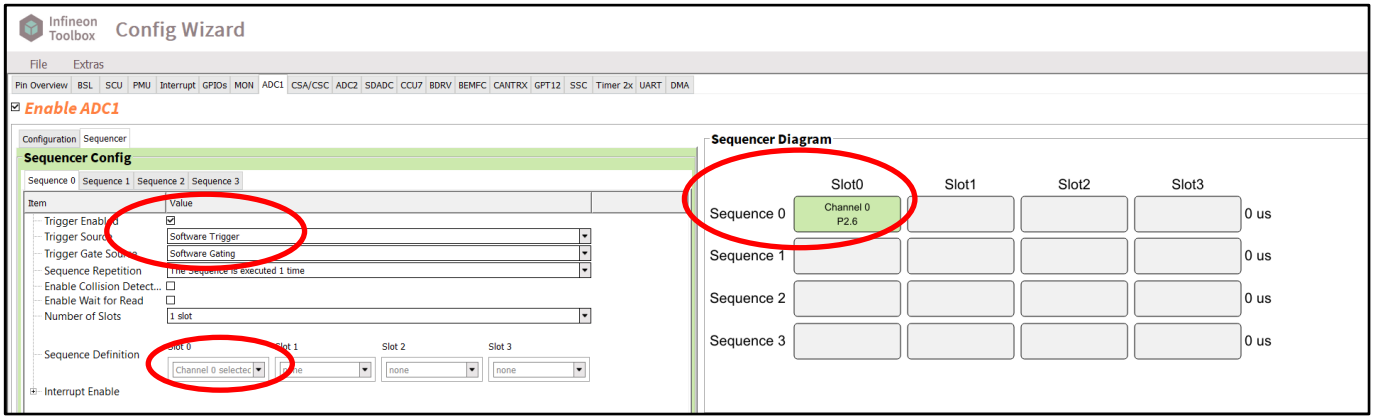

Figure 7 Config Wizard, module ADC1, tab Sequencer

Finally, save your configuration to take these changes into account (File -> Save).

3.3 Sample code

Figure 8 shows the application code of the TLE9893_2QKW62S_ADC1_POTI_SWTRG example.

Within the endless for-loop (line 109) the watchdog service is executed in order to avoid any stall during the execution.

Within this same loop, the function `switchLedAccordingPotiVoltage` is continuously called. This service function contains initially the software trigger call `ADC1_startSequence(ADC1_SEQ0)` in line 122.

Once this is successfully executed, the result of the Poti measurement in mV is read via the `ADC1_getChResult_mV(&u16_mVolt, ADC1_DCH0)` call in line 128. The result is written into the `u16_mVolt` value in case of a successful execution.

The digital result is calculated into a multiple value of 500mV. According to this value, the appropriate 9 GPIO LEDs are switched on or off.

In case of a 48-pin device, only 6 LEDs are used for a range from 0V to 3V.

```
109     for (;;)
110     {
111       /* Main watchdog service */
112       (void) PMU_serviceFailSafeWatchdog();
113
114       /* Switch LEDs according to the current Poti setting (0-5V) */
115       switchLedAccordingPotiVoltage();
116     }
117   }
118
119   void switchLedAccordingPotiVoltage()
120   {
121     /* Trigger ADC1 sequence 0 */
122     if (ERR_LOG_SUCCESS == ADC1_startSequence(ADC1_SEQ0))
123     {
124       /* Check if the data are valid */
125       if (ADC1_getCh0ResultValidSts() == 1u)
126       {
127         /* Read digital result of Poti setting in mV */
128         if (ERR_LOG_SUCCESS == ADC1_getChResult_mV(&u16_mVolt, ADC1_DCH0))
129         {
130           /* Calculate multiple of 500 mV */
131           u16_mVolt = u16_mVolt / (uint16) M_VOLT_500;
132           /* If poti voltage > 500mV then P0.1 is high, otherwise low */
133           u16_mVolt > 0 ?
134             GPIO_setP01State(GPIO_STATE_HIGH) : GPIO_setP01State(GPIO_STATE_LOW);
135           /* If poti voltage > 1000mV then P0.2 is high, otherwise low */
136           u16_mVolt > 1 ?
137             GPIO_setP02State(GPIO_STATE_HIGH) : GPIO_setP02State(GPIO_STATE_LOW);
138           /* If poti voltage > 1500mV then P0.3 is high, otherwise low */
139           u16_mVolt > 2 ?
140             GPIO_setP03State(GPIO_STATE_HIGH) : GPIO_setP03State(GPIO_STATE_LOW);
141           /* If poti voltage > 2000mV then P1.0 is high, otherwise low */
142           u16_mVolt > 3 ?
143             GPIO_setP10State(GPIO_STATE_HIGH) : GPIO_setP10State(GPIO_STATE_LOW);
144           /* If poti voltage > 2500mV then P1.1 is high, otherwise low */
145           u16_mVolt > 4 ?
146             GPIO_setP11State(GPIO_STATE_HIGH) : GPIO_setP11State(GPIO_STATE_LOW);
147           /* If poti voltage > 3000mV then P1.2 is high, otherwise low */
148           u16_mVolt > 5 ?
149             GPIO_setP12State(GPIO_STATE_HIGH) : GPIO_setP12State(GPIO_STATE_LOW);
150   #ifndef UC_FEATURE_48PIN
151           /* If poti voltage > 3500mV then P0.4 is high, otherwise low */
152           u16_mVolt > 6 ?
153             GPIO_setP04State(GPIO_STATE_HIGH) : GPIO_setP04State(GPIO_STATE_LOW);
154           /* If poti voltage > 4000mV then P0.5 is high, otherwise low */
155           u16_mVolt > 7 ?
156             GPIO_setP05State(GPIO_STATE_HIGH) : GPIO_setP05State(GPIO_STATE_LOW);
157           /* If poti voltage > 4500mV then P0.6 is high, otherwise low */
158           u16_mVolt > 8 ?
159             GPIO_setP06State(GPIO_STATE_HIGH) : GPIO_setP06State(GPIO_STATE_LOW);
160   #endif
161         }
162       }
163     }
164   }
```

Figure 8 TLE9893_2QKW62S_ADC1_POTI_SWTRG application code

References

See the code examples at **www.infineon.com**

Revision history

Document version	Date of release	Description of changes
1.0	2021-04-28	Initial version
1.1	2022-10-13	Editorial changes

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

Edition 2022-10-13

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

IMPORTANT NOTICE

The information contained in this application note is given as a hint for the implementation of the product only and shall in no event be regarded as a description or warranty of a certain functionality, condition or quality of the product. Before implementation of the product, the recipient of this application note must verify any function and other technical information given herein in the real application. 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) with respect to any and all information given in this application note.

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.