



# SPC574K72 SARADC example

## Quick Guide

# Table of Contents

- 1. Terms & Abbreviations . . . . . 1**
- 2. Introduction . . . . . 2**
  - 2.1. Prerequisites . . . . . 2
  - 2.2. HW resources . . . . . 2
- 3. Example overview . . . . . 3**
  - 3.1. System Description . . . . . 3
  - 3.2. Application Flow . . . . . 4
  - 3.3. Interrupt Handling . . . . . 4
  - 3.4. ADC module configuration . . . . . 5
- 4. Example Import & Build . . . . . 6**
- Appendix A: Document References . . . . . 7**
- Appendix B: Release Notes . . . . . 8**

# 1. Terms & Abbreviations

SARADC

Successive Approximation Register Analog-to-Digital Converter.

PIT

Periodic Interrupt Timer

IOP

I/O processor

BSP

Board Support Package

GPIO

General Purpose Input Output

EVB

Evaluation Board

NC

Not Configured

uC

Microcontroller

ISR

Interrupt Service Routine

## 2. Introduction

This example is a small functional project designed to simplify an evaluation phase of the certain peripheral on the microcontroller. It is built on the c-startup example providing necessary low-level functions like a startup code, a minimalistic hardware abstraction or predefined memory partitioning. On top of this low-level implementation, it provides the peripheral example code running on single core.

### 2.1. Prerequisites

- SPC574K72 device
- An evaluation board (SPC57xxMB + MPC5746M\_176DS)
- HighTec Development Suite, version: 4.9.3.0

### 2.2. HW resources

Hardware resources used by this example:

HW unit	channel / I/O pin	function
ADCSAR_DIG_B	PB[0]	Core [2] SARB ADC channel 0
default resources from the c-startup example		
GPIO	PA [0]	Core [0] LED control
GPIO	PA [2]	Core [2] LED control
PIT 0	Channel [0]	Core [0] periodic interrupt
PIT 0	Channel [2]	Core [2] periodic interrupt

Tab. 1. HW resources

### 3. Example overview

- SARADC B module converts AN[0] channel with 12-bit resolution.
- PIT timer interrupt routine starts the conversion.
- End of the ADC conversion event triggers an interrupt in which a measured value is stored in the global `g_saradc_result_uint` and `g_saradc_result_volts` variable.

Note: The analog input channel can be connected to the output of the potentiometer RV1 by populating the jumper J53 on the EVB.

#### 3.1. System Description

The figure below shows the system view of the example:

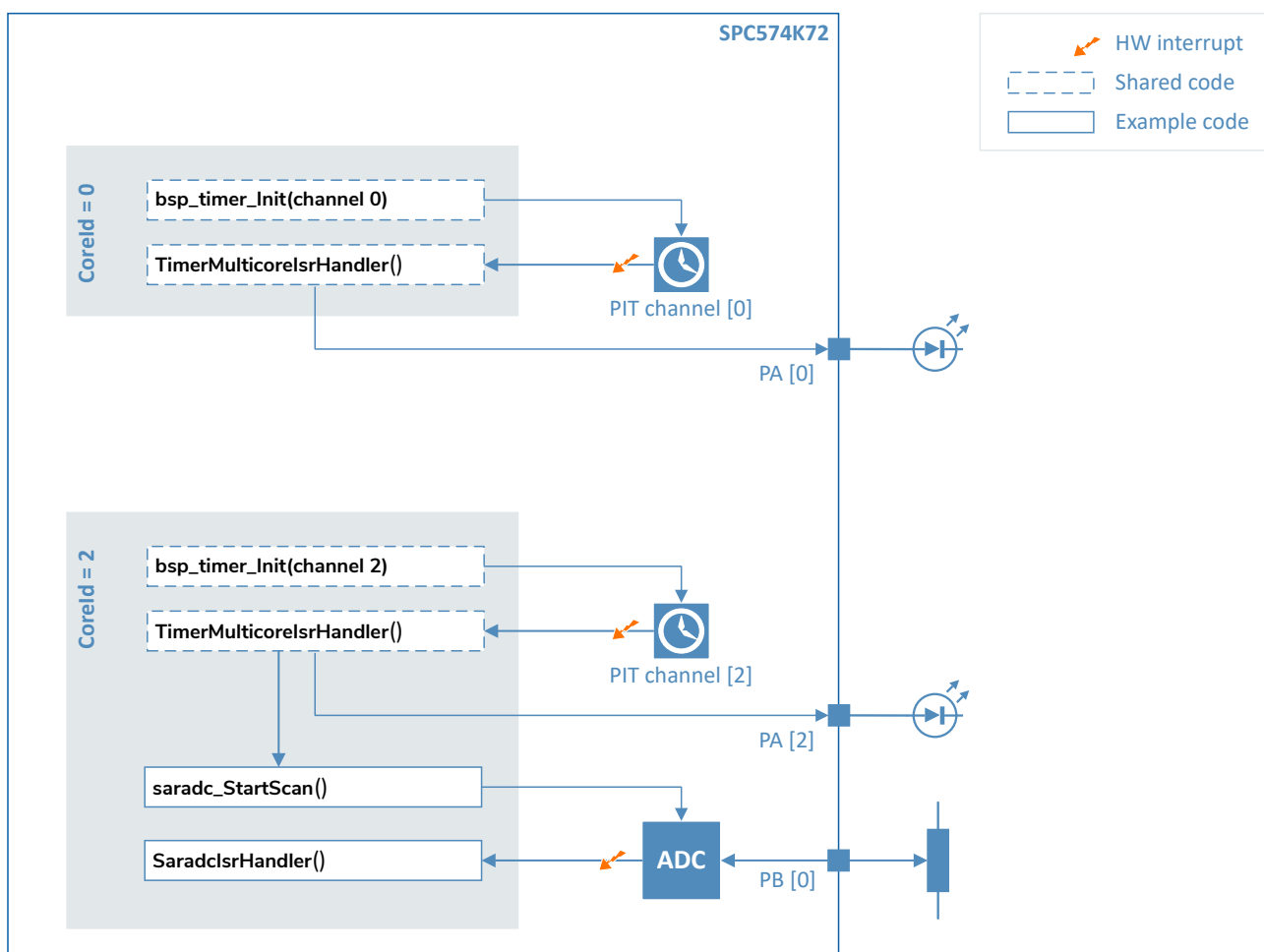


Fig. 1. System block diagram

The example code runs on a single core. Other cores run the default functionality defined by SPC5xx\_c\_startup.

The application initializes SARADC-B module to do a single conversion of analog input AN[0] triggered

by a SW request.

The PIT timer provided by BSP layer triggers an ADC conversion.

## 3.2. Application Flow

The UC\_CORE\_IOP core is responsible for the SARADC-B conversion.

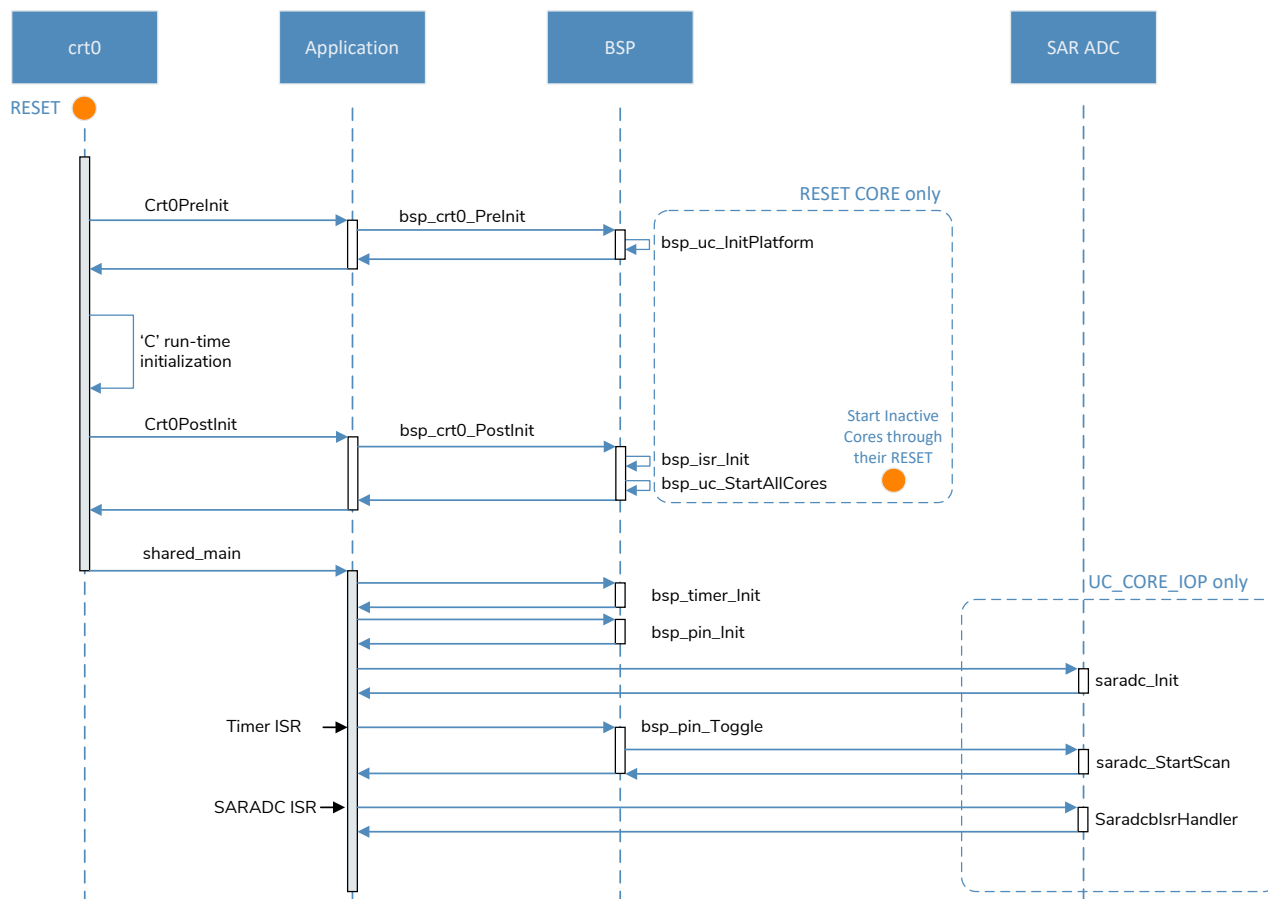


Fig. 2. Simplified individual core execution flow

## 3.3. Interrupt Handling

Implemented interrupt handling uses low-level BSP functions, for example to register each interrupt service routine (bsp\_isr\_RegisterHandler).

The SARADC initialization function assigns the "SAR ADC B combined interrupt" interrupt vector and sets the interrupt vector priority.

Code 1. /app/saradc.c

```
bsp_isr_RegisterHandler(UC_SARADCB_ISR, 10, SaradcbIsrHandler);
```

The example uses the SW interrupt mode.

## 3.4. ADC module configuration

The SARADC module is configured with these parameters:

- SARADC clock:  $f_{\text{ADC}} = 10\text{MHz}$
- Sampling phase duration  $T_{\text{SAMP}} = 1500\text{ns}$
- Single conversion triggered by SW
- End of conversion always overwrites ADC result value
- End of conversion interrupt (EoC) is triggered after each conversion.

for more details see `saradc_Init()` function below.

Code 2. `/app/saradc.c`

```

1 void saradc_Init (void)
2 {
3     SARADC_B.MCR.B.NTRGEN = 1; /* Normal trigger enabled to start a normal
4                               conversion */
5     SARADC_B.MCR.B.MODE = 0x0; /* Sets conversion mode on SARADCB, single
6                               conversion */
7     SARADC_B.MCR.B.OWREN = 1; /* SAR ADC B overwrite enable set */
8     SARADC_B.SARADC_ICIMR[0].R = 0x00000001U; /* Interrupt of ANP[0] enabled */
9     SARADC_B.IMR.B.MSKNEOC = 1U; /* NEOC interrupt is enabled */
10    SARADC_B.CTR[0].B.CRES = SARADC_CTR0_CRES;
11    SARADC_B.CTR[0].B.INPSAMP = SARADC_B_CTR0_INPSAMP;
12    SARADC_B.ICNCMR[0].R = 0x00000001U; /* Sets channel ANP[0] to be converted
13                                       in SARADCB 0 (PB[0]) */
14    SARADC_B.ICDR[0].R = 0x0U; /* Setup internal channel for SARADCB,
15                               no pre-charge */
16
17    ...
18
19    SARADC_B.MCR.B.PWDN = 0; /* Power up SAR ADC B */
20
21    ...
22
23 }
```

## 4. Example Import & Build

Follow these steps to import an example project to the HighTec IDE environment:

1. From menu **File** → **Import** → **General** choose an option Existing Projects into Workspace
2. Browse for your project location
3. Select project
4. Leave copy to the workspace option empty
5. Click Finish

Activate the project from menu **Project** → **Set Active Project**.

Build the project from the menu **Project** → **Build Project**.

The output binary file is located under the `_irom_build` folder.



## Appendix A: Document References

- [1] "SPC5x c-startup - 'C' run-time initialization", HighTec EDV Systeme GmbH, 2018
- [2] "SPC5x c-startup - Linker file template", HighTec EDV Systeme GmbH, 2018
- [3] "SPC5x c-startup - Hardware Abstraction Layer", HighTec EDV Systeme GmbH, 2018

## Appendix B: Release Notes

Version	Date	Changes to the previous version
1.0	March 2018	Initial version based on SPC574K72 c-startup example V1.1



HighTec EDV-Systeme GmbH  
Feldmannstrasse 98, D-66119 Saarbrücken  
[info@hightec-rt.com](mailto:info@hightec-rt.com)  
+49-681-92613-16  
[www.hightec-rt.com](http://www.hightec-rt.com)