



SPC570S50 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 3
 - 3.3. Interrupt Handling 4
 - 3.4. ADC module configuration 5
- 4. Example Import & Build 6**
- Appendix A: Document References 7**
- Appendix B: Document history 8**

1. Terms & Abbreviations

SARADC

Successive Approximation Register Analog-to-Digital Converter.

PIT

Periodic Interrupt Timer

IOP

I/O processor

BSP

Board Startup Package

crt0

'C' run-time environment initialization code

HAL

Hardware Abstraction Layer

GPIO

General Purpose Input Output

EVB

Evaluation Board

NC

Not Configured

uC

Microcontroller

PIT

Periodic Interrupt Timer

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

- SPC570S50 device
- An evaluation board (SPC57xxMB + SPC570S50)
- HighTec Development Suite, version: 4.9.3.0

2.2. HW resources

Hardware resources used by this example:

Tab. 1. HW resources

HW unit	channel / I/O pin	function
ADCSAR_B	PC[1]	Core [0] SARADC_B channel [0]
ADCSAR_B	Channel[0]	Core [0] SARADC_B combined interrupt
default resources from the c-startup example		
GPIO	PA [0]	Core [0] LED control
PIT 0	Channel [0]	Core [0] periodic interrupt

Note

The PC[1] is not connected directly to potentiometer populated on SPC57xxMB evaluation board. For correct behavior to connect PC[1] to J53:1 via wire.

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:

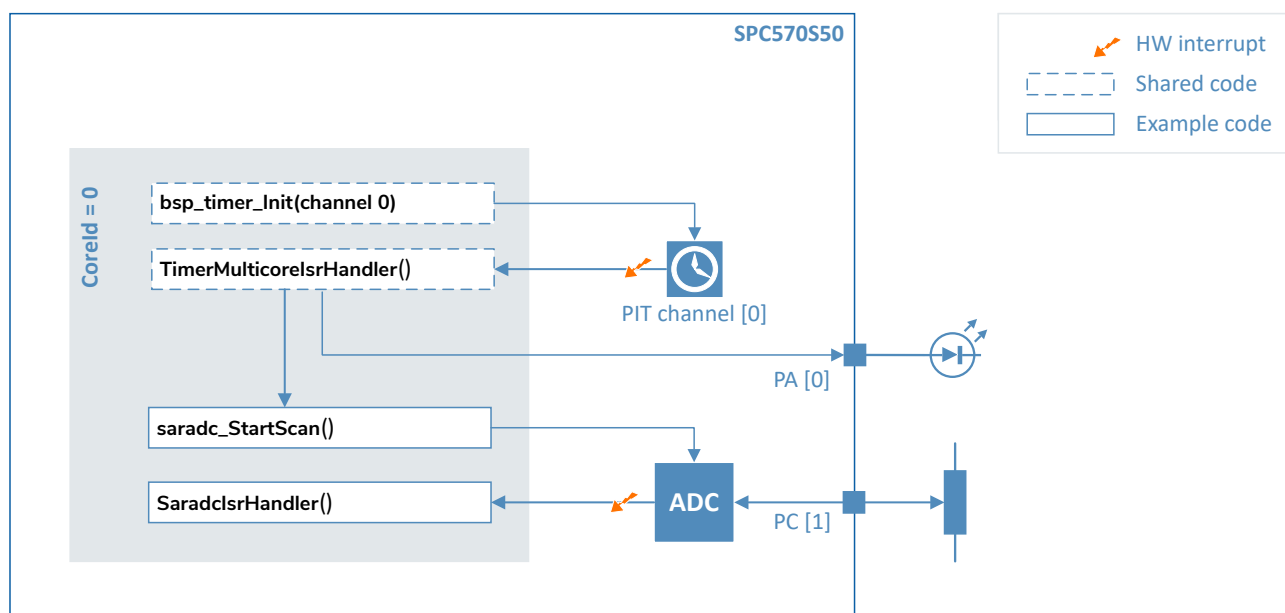


Fig. 1. System block diagram

The example code runs on a single core.

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

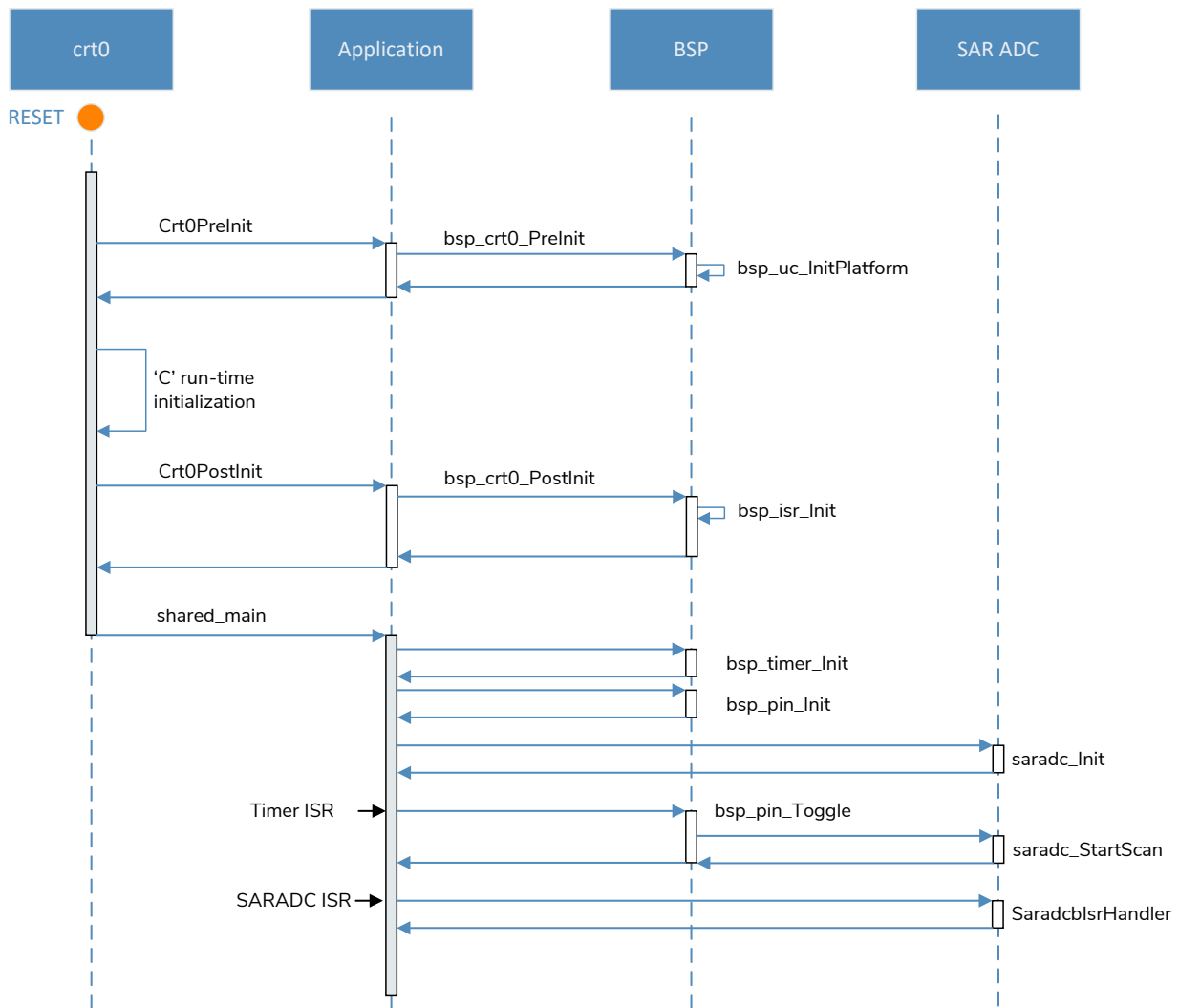


Fig. 2. Simplified individual core execution flow

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   SAR_ADC_12BIT_B0.MCR.B.NTRGEN = 1; /* Normal trigger enabled to start a
2       normal conversion */
3   SAR_ADC_12BIT_B0.MCR.B.MODE = 0x0; /* Sets conversion mode on SARADCB,
4       single conversion */
5   SAR_ADC_12BIT_B0.MCR.B.OWREN = 1; /* SAR ADC B overwrite enable set */
6   SAR_ADC_12BIT_B0.ICIMR0.R = 0x00000001U; /* Interrupt of ANP[0] enabled */
7   SAR_ADC_12BIT_B0.IMR.B.MSKNEOC = 1U; /* NEOC interrupt is enabled */
8   SAR_ADC_12BIT_B0.CTR[0].B.CRES = SARADC_CTR0_CRES;
9   SAR_ADC_12BIT_B0.CTR[0].B.INPSAMP = SARADC_B_CTR0_INPSAMP;
10  SAR_ADC_12BIT_B0.ICNMR0.R = 0x00000001U; /* Sets channel ANP[0] to be
11       converted in SARADCB 0 (PB[0])*/
12  SAR_ADC_12BIT_B0.ICDR[0].R = 0x0U; /* Setup internal channel for SARADCB,
13       no pre-charge */
14
15  ...
16
17  SAR_ADC_12BIT_B0.MCR.B.PWDN = 0; /* Power up SAR ADC B */
18
19  ...
20
21 }
```

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 the copy to the workspace option check box 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: Document history

Version	Date	Changes to the previous version
1.0	February 2018	Initial version