How to get the best ADC accuracy in PSOC™ Control C3 MCU

About this document

Scope and purpose

This application note explains the analog-to-digital converter (ADC) principle and how to improve ADC accuracy methods in the PSOC™ Control C3 MCU.

Intended audience

This document is intended for anyone who uses the ADC of the PSOC™ Control C3 MCUs.

This application note assumes that you are familiar with PSOC™ Control C3 and the ModusToolbox™. If you are new to PSOC™ Control C3, see AN238329 - Getting started with PSOC™ Control C3 MCU on ModusToolbox™. If you are new to the ModusToolbox™, see the ModusToolbox™ home page.

Associated part family

All PSOC™ Control C3 devices.

Please read the sections "Important notice" and "Warnings" at the end of this document

Table of contents

1 Introduction

The PSOC™ Control C3 MCU family is designed for real-time control, enhanced sensing, secure, and low-power operations. Some target applications for these MCUs are:

- Motor control in power tools, home appliances, industrial drives, light electric vehicles, robotics, and drones
- Digital power control in switched mode power supply (SMPS) and PFC applications for LED lighting, EV chargers, solar inverters, servers, and PC power supplies
- Wide bandgap technologies (for example, SiC and GaN) based motor control and power conversion applications

The PSOC™ Control C3 MCU contains a high-performance programmable analog subsystem (HPPASS). It has one 12-bit ADC with up to 16 parallel sampling channels. The ADC supports multiple S/H, which enables synchronous sampling on several channels.

The ADC senses the voltage and converts it to a digital code, post process the ADC and therefore, making first-level decisions as close to the sensor as possible. The accuracy of the ADC does not only depend on its performance and features but also on the overall application design around the ADC. If the user wants to achieve nominal accuracy in a practical application, sufficient attention should be paid to the design of the software configuration and peripheral circuit.

This application note explains the ADC basic working principle and related ADC parameters, the ADC features of the PSOC™ Control C3 MCU, and recommendations on how to improve ADC accuracy.

2 ADC principle and error definition

This section introduces the ADC principle to help understand ADC operation, related ADC parameters, and explanations of the different types and sources of ADC errors.

2.1 ADC principle

The ADC of the PSOC™ Control C3 MCU uses the successive approximation register (SAR) method; it is also known as the successive approximation register analog-to-digital converter (SAR ADC). It uses a capacitor network to compare the analog input voltage with a reference voltage, generated from VREF and VSSA. This reference voltage is adapted step by step through a successive approximation.

The ADC conversion includes sampling, quantization, and post-processing steps. In the sampling phase, the voltage of the external signal must be sampled to the sampling capacitor of ADC within the specified sampling time; that is, during the closing of the sampling switch SW, the external input signal must charge the sampling capacitor C_{VIN} through the external input resistance R_{EXT} and the ADC sampling resistance R_{VIN}, as shown in Figure 1.

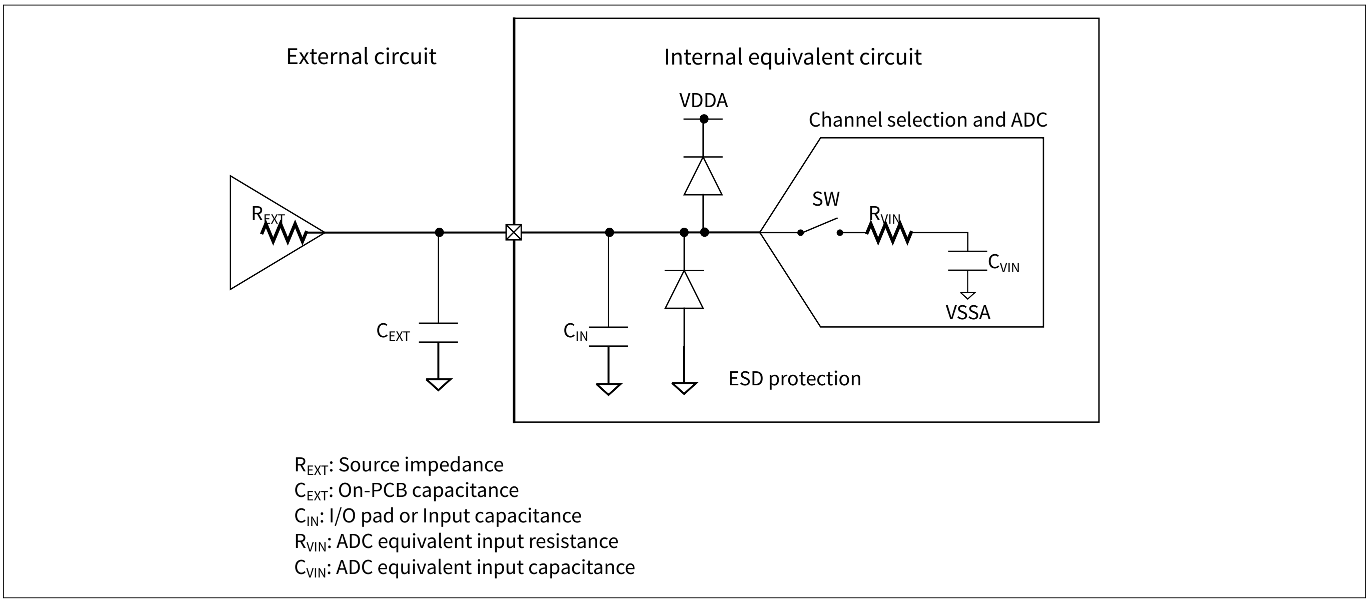

External circuit
Internal equivalent circuit
VDDA
Channel selection and ADC
SW
R_{EXT}
R_{VIN}
C_{VIN}
VSSA
C_{EXT}
C_{IN}
ESD protection
R_{EXT}: Source impedance
C_{EXT}: On-PCB capacitance
C_{IN}: I/O pad or Input capacitance
R_{VIN}: ADC equivalent input resistance
C_{VIN}: ADC equivalent input capacitance

Figure 1 ADC equivalent circuit for analog input

Each sampling process can be simplified as an external signal charging the sampling capacitor through input impedance and sampling resistance. When the sampling time is over, the sampling error is expressed as the difference between the voltage on the sampling capacitor and the voltage on the signal source. In an ideal sampling process, the voltage difference should be kept within 0.5 LSB (LSB is the minimum voltage resolution of SAR ADC, and 0.5 LSB is the quantization error of SAR ADC).

In the quantization phase, the sampling switch is turned on first and then driven by the ADC clock. Based on the switching capacitor technology, the voltage on the ADC sampling capacitor is compared with the reference voltage with different weights step by step, and the value on each bit of the n-bit data is determined bit by bit (n is the resolution of the ADC), and then the digital code value is coded and output. In the quantization process, the reference voltage VREF needs to charge the switched capacitor network. VREF benchmarks need to be stable during quantification.

2.2 Transfer characteristic and error definition

This section lists the main errors that have an effect on ADC conversion accuracy. These error values are specified in the ADC specifications section of the PSOC™ Control C3 MCU datasheet.

The following diagrams show the ideal transfer characteristic of an ADC and the definitions for the different kinds of errors:

- Offset error (OFFSETERR)
- Gain error (GAINERR)
- Differential non-linearity error (DNLE)
- Integral non-linearity error (INLE)
- Total unadjusted error (TUE)

2.2.1 Ideal transfer characteristic

Figure 2 defines the ideal transfer characteristic for an ADC.

The Ideal Transfer Curve (1) transfers each input to an output.

The Ideal ADC Transfer Curve (2) includes a quantization error. Since all analog input values are presumed to exist, they must be quantized by partitioning the continuum into discrete digital values. All analog values within a given range (quantization step) are represented by the same digital value, which corresponds to the nominal mid-range value. That is the reason for the quantization uncertainty of +/- 0.5 LSB, which is a natural error and inherent to each ADC.

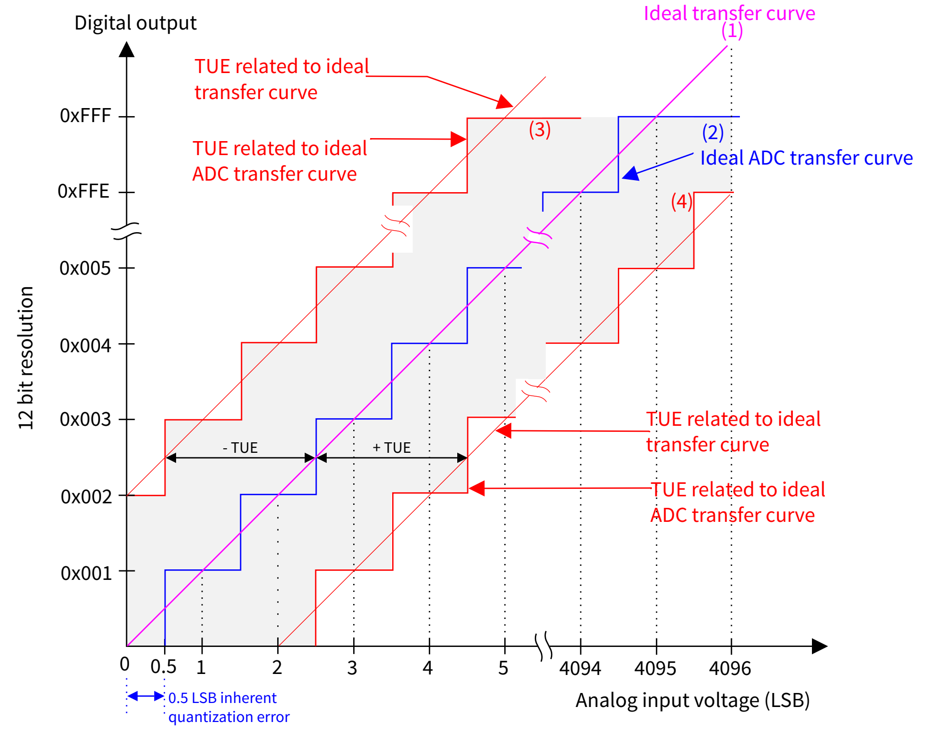

Digital output
Ideal transfer curve
(1)
TUE related to ideal transfer curve
TUE related to ideal ADC transfer curve
0xFFF
0xFFE
0x005
0x004
0x003
0x002
0x001
12 bit resolution
(3)
(2)
Ideal ADC transfer curve
(4)
- TUE
+ TUE
TUE related to ideal transfer curve
TUE related to ideal ADC transfer curve
0
0.5
1
2
3
4
5
4094
4095
4096
Analog input voltage (LSB)
0.5 LSB inherent quantization error

Figure 2 **Ideal transfer characteristic**

The analog input voltage range must be within VSS, up to VREF. The quantization step size is 1 LSB = VREF/2^{12} in the PSOC™ Control C3 MCU.

According to the Ideal Transfer Curve (1), the first digital transition, from 0 to 1, is shifted to the analog value of 0.5 LSB to get a minimum quantization uncertainty. That is why the first step width of the Ideal ADC Transfer Curve (2) is 0.5 LSB, and the last step width is 1.5 LSB with a digital output range from 0 to (2^{12} - 1).

The compensated inherent quantization error in relation to the analog input voltage is shown in Figure 3.

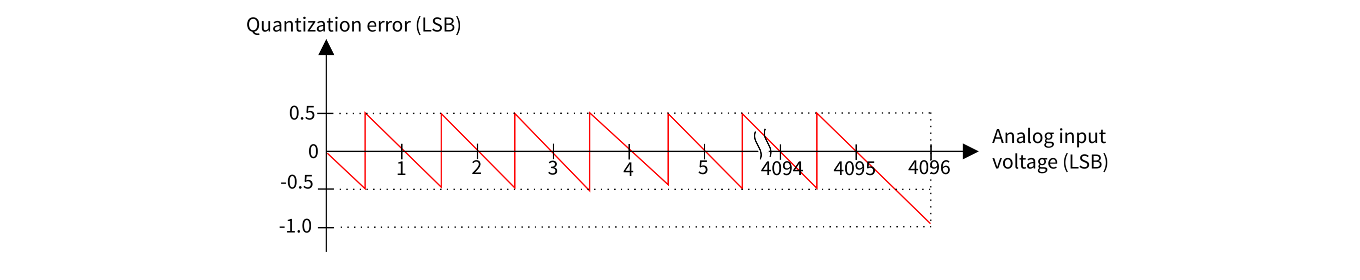


Figure 3 **Quantization error**

The total unadjusted error (TUE) includes all ADC-related inaccuracies, such as production process deviations and internal noise. The TUE consists of offset error, gain error, differential non-linearity error (DNLE), and integral non-linearity error (INLE), but it is not the sum of individually measured errors. Because some ADC errors, such as offset and gain, can compensate for each other, the TUE can be far less than the absolute sum of all individual errors. Figure 2 shows the definition of the TUE in relation to the Ideal ADC Transfer Curve (1).

The real result of the ADC is in the range of the ideal ADC transfer curve (2) +/- TUE. This area is grayed out in Figure 2 and is between both TUE related to Ideal ADC Transfer Curves (3) and (4).

***Note**: ADC reference voltages are considered ideal. An incorrect reference voltage generates an additional error.*

2.2.2 Offset error

The offset error is the deviation from the Ideal ADC Transfer Curve at the lowest transition level on the Real ADC Transfer Curve. It is the input voltage required to bring the digital output to zero and can be measured by determining the first digital transition, from 0 to 1, of the ADC. The offset error affects all codes by the same amount.

In the following figure, all other kinds of errors (gain, DNLE, and INLE) are excluded.

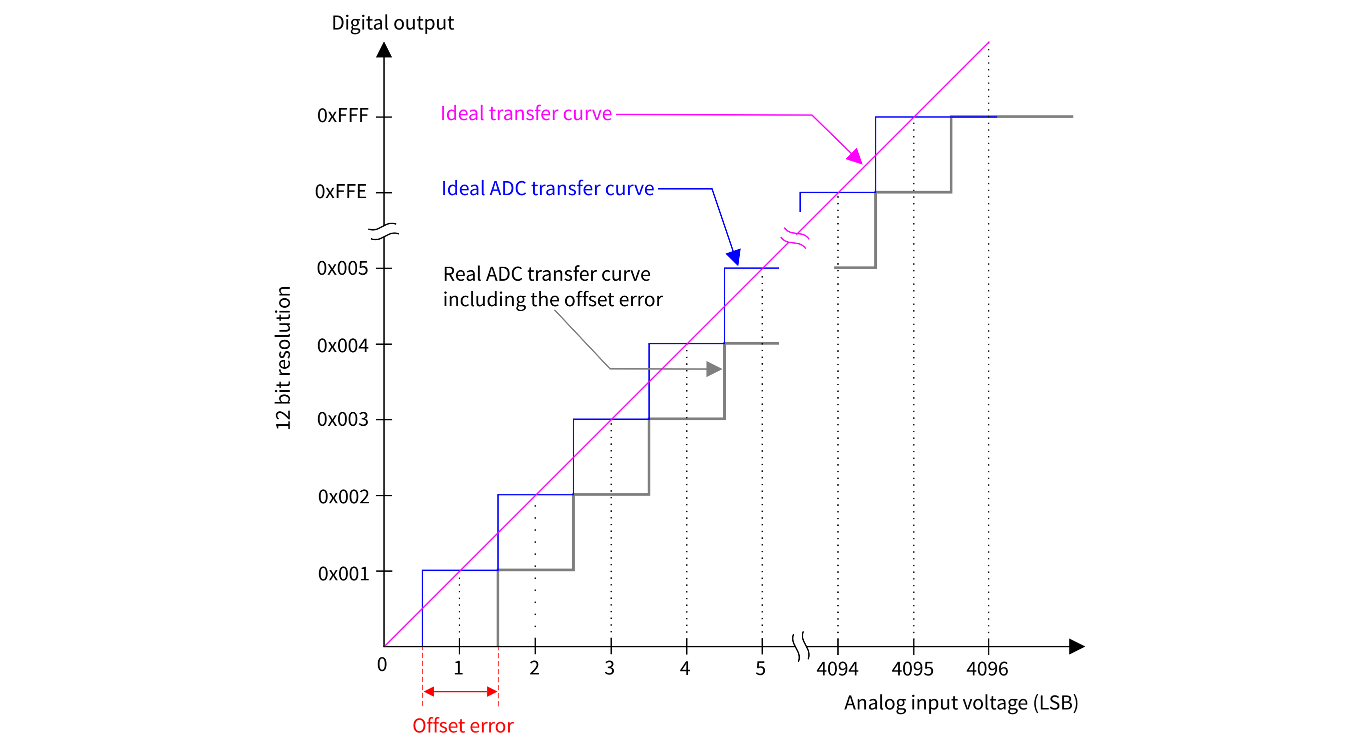


Figure 4 **Offset error**

2.2.3 Gain error

The gain error is the difference between the slopes of the Real ADC Transfer Curve and the Ideal ADC Transfer Curve at the maximum digital out value.

In the following figure, all other kinds of errors (offset, DNLE, and INLE) are excluded.

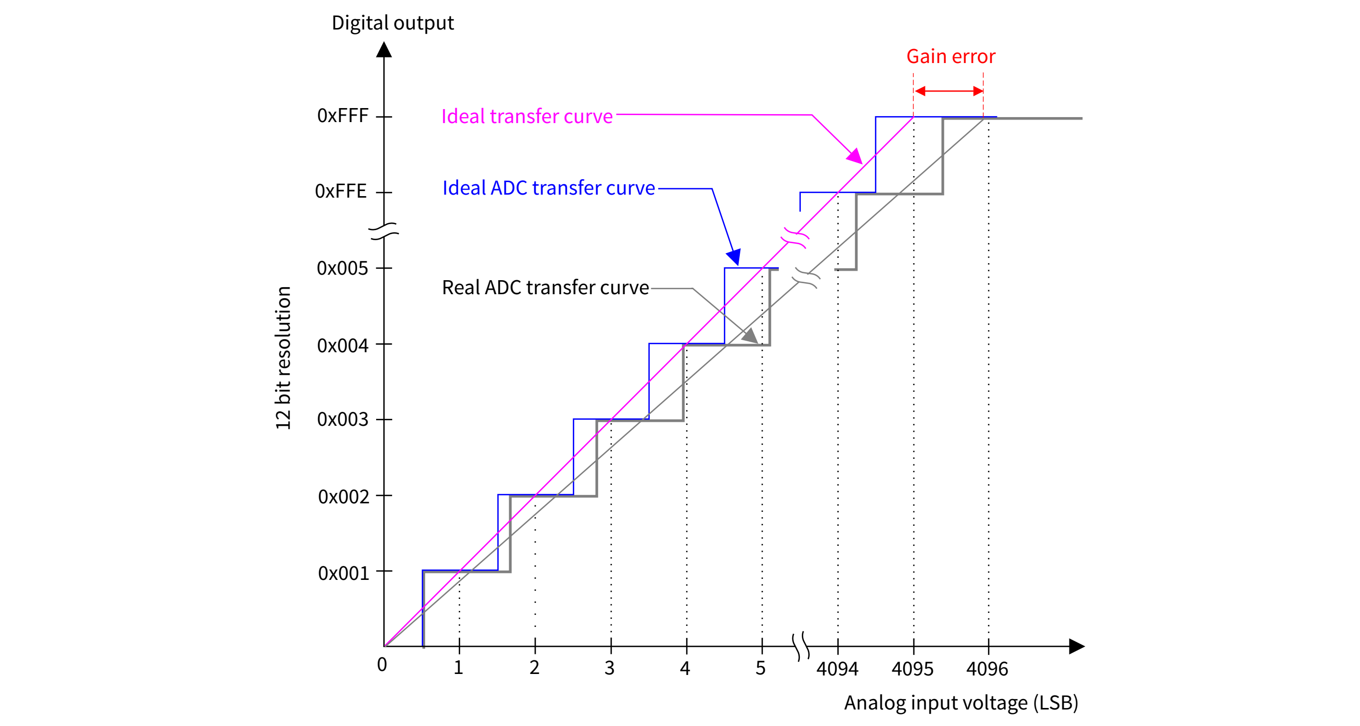


Figure 5 **Gain error**

2.2.4 Differential non-linearity error

The differential non-linearity error (DNLE) describes variations in the analog value between adjacent pairs of digital numbers over the full range of the digital output.

If each transition step width is exactly 1 LSB, the DNLE is zero.

If the transitions are 1 LSB +/- 1 LSB, then there is a possibility of missing codes. If a missing code occurs, then one value of the digital output is missing; for example, the digital output may jump from 0011 to 0101 and miss out on 0100 (see Figure 6).

If the DNLE is less than 1 LSB, then a missing code is automatically excluded. In the following figure, all other kinds of errors (offset, gain, and INLE) are excluded.

If the output code always increases with an increase in the analog input and always decreases with a decrease in the analog input, then the ADC is monotonic. The ADC is called monotonic when the DNLE is in the range of -1 LSB ≤ DNL ≤ 1LSB.

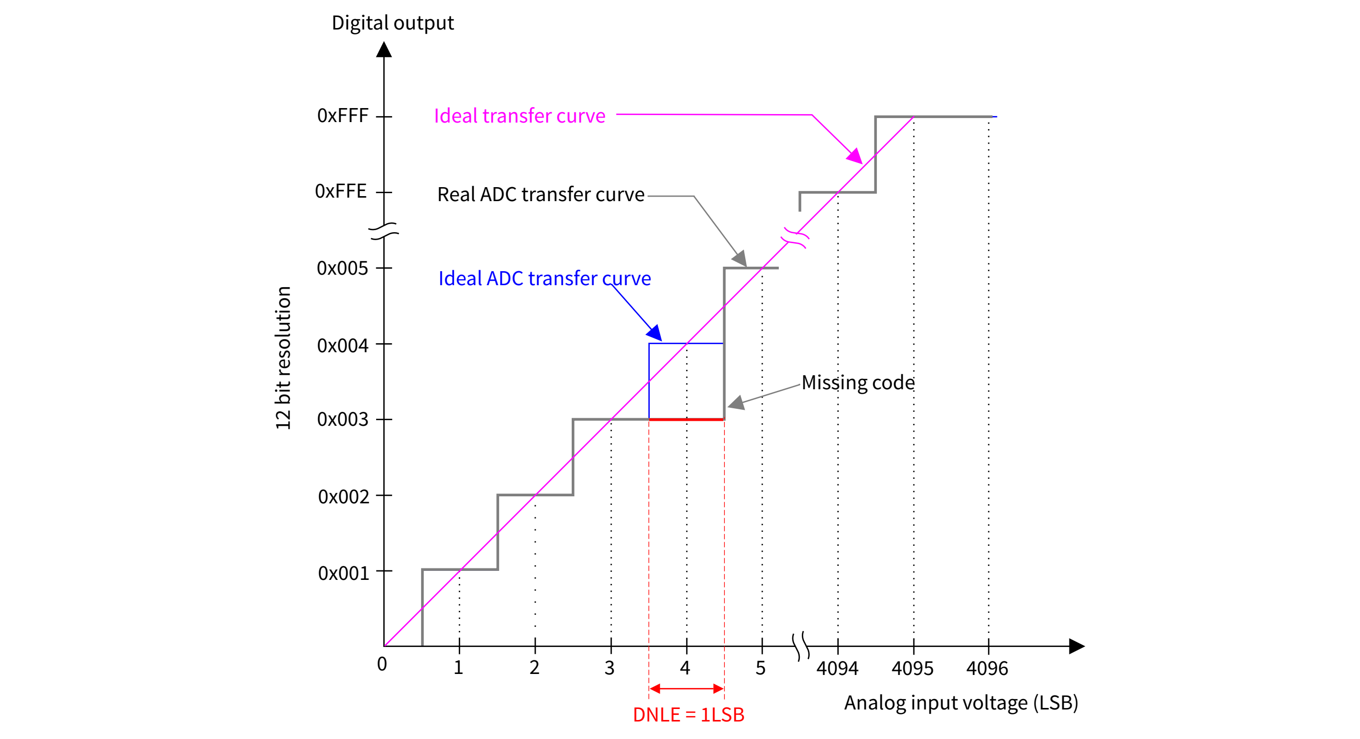


Figure 6 **Differential non-linearity error**

2.2.5 Integral non-linearity error

The integral non-linearity error (INLE) is the maximum difference between the Ideal ADC Transfer Curve and the adjusted Real ADC Transfer Curve (without offset and gain error). In the following figure, DNLE is excluded.

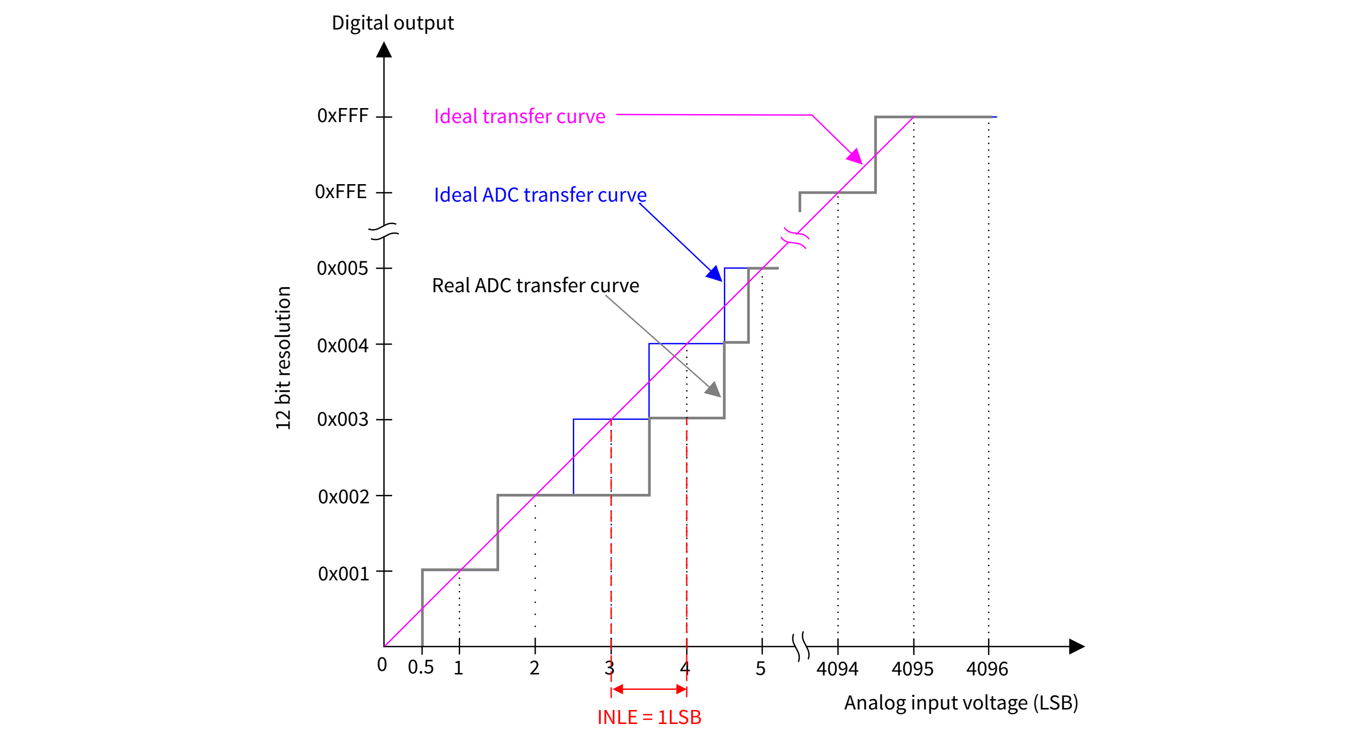


Figure 7 **Integral non-linearity error**

3 PSOC™ Control C3 ADC features

The high-performance programmable analog subsystem (HPPASS) of the PSOC™ Control C3 MCU has one 12-bit SAR ADC. The SAR ADC has the following features:

- When operating in a single-channel, the maximum speed of the ADC is 12 Msps
- 16 samplers, including 12x direct I/O connections and 4x muxed connections, which can produce simultaneous sampling of external analog signals
- All samplers support gains of 1, 3, 6, and 12
- Up to 16 dedicated analog pads, connected to up to 16 parallel sample stages
- Two additional GPIOs can be used as analog inputs by AMUX
- Two internal voltage signals (temperature sensor and DAC output voltage) can be connected to a muxed sampler by AMUX
- AMUX can be controlled by a SAR sequencer, firmware, or autonomous controller (AC)
- 21 channels: 16 channels for dedicated analog inputs, two channels for additional GPIOs, and three channels for internal voltage signals
- Post processing
  - Pseudo differential conversion
  - Result signed/unsigned format
  - Two 16-tap finite impulse response (FIR) filters
  - Hardware averaging (result accumulation, no shifting function) of a configurable number of samples (2, 4, 8, 16)
  - A 32-entry FIFO that can be subdivided into 1, 2, or 4 FIFOs
  - Eight limit detection resources
- SAR sequencer
  - Up to eight scan groups
  - Up to 16 samplers can be enabled for simultaneous sampling in a given group
  - Every group can select input for four muxed samplers
  - Every group can be triggered by firmware, triggers from other peripherals (TCPWM and GPIO), or an autonomous controller (AC)
  - Every group can select one of three available programmable sample times
  - Two priorities, programmable per group
  - Every group supports a continuous mode
- Self-calibration support for offset, linearity, and gain
- Interrupt generation
- 4 FIFO level triggers, which can trigger other on-chip peripherals (such as DMA, GPIO, and TCPWM)
- On-chip temperature sensor

3.1 ADC block diagram

Figure 8 shows the SAR ADC block diagram. The SAR ADC mainly consists of a SAR sequencer, 16 samplers, the SAR ADC core, and post-processing blocks.

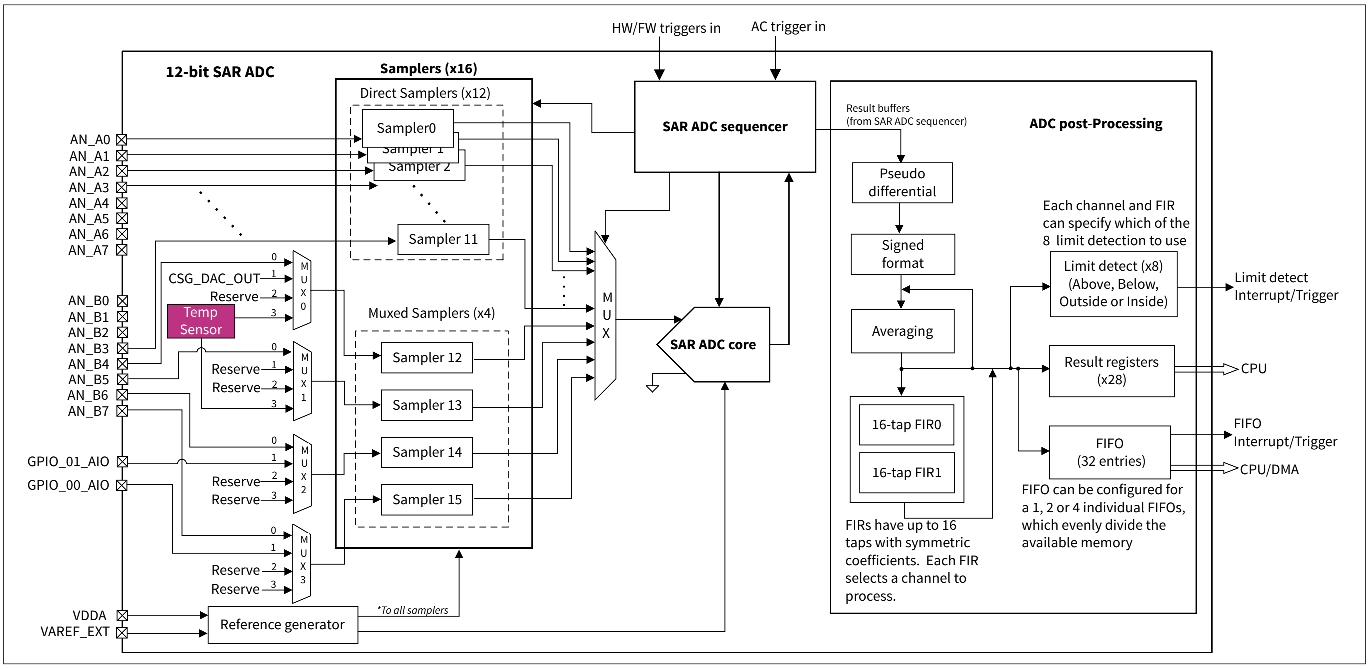


Figure 8 **ADC block diagram**

Note: *The user can select VDDA or external reference from the VAREF_EXT pin for SAR ADC reference voltage by setting the VREF_SEL fields of the HPPASS_SAR_CFG_CTRL register. To guarantee SAR ADC performance, select the external reference from the VAREF_EXT pin.*

For more details on SAR ADC, see the PSOC™ Control C3 MCU architecture reference manual.

3.2 ADC channel mapping

The ADC contains 12 direct samplers and four muxed samplers, which correspond to direct input channels and muxed input channels. It has 28 channel inputs. However, some muxed inputs are reserved, and therefore, the channels are discontinuous in the PSOC™ Control C3 MCU. See Table 1 for actual port-to-channel mapping.

Table 1 **PSOC™ Control C3 MCU SAR ADC channel mapping**

Channel number	Sampler	Channel type	Input
0	Sampler 0	Direct	AN_A0
1	Sampler 1	Direct	AN_A1
2	Sampler 2	Direct	AN_A2
3	Sampler 3	Direct	AN_A3
4	Sampler 4	Direct	AN_A4
5	Sampler 5	Direct	AN_A5
6	Sampler 6	Direct	AN_A6
7	Sampler 7	Direct	AN_A7
8	Sampler 8	Direct	AN_B0
9	Sampler 9	Direct	AN_B1

(table continues...)

Table 1 (continued) PSOC™ Control C3 MCU SAR ADC channel mapping

Channel number	Sampler	Channel type	Input
10	Sampler 10	Direct	AN_B2
11	Sampler 11	Direct	AN_B3
12	Sampler 12 for MUX 0	MUX0 select 0	AN_B4
13		MUX0 select 1	CSG DAC OUT
14		MUX0 select 2	Reserve
15		MUX0 select 3	Temperature sensor
16	Sampler 13 for MUX 1	MUX1 select 0	AN_B5
17		MUX1 select 1	Reserve
18		MUX1 select 2	Reserve
19		MUX1 select 3	Temperature sensor
20	Sampler 14 for MUX 2	MUX2 select 0	AN_B6
21		MUX2 select 1	GPIO_01_AIO
22		MUX2 select 2	Reserve
23		MUX2 select 3	Reserve
24	Sampler 15 for MUX 3	MUX3 select 0	AN_B7
25		MUX3 select 1	GPIO_00_AIO
26		MUX3 select 2	Reserve
27		MUX3 select 3	Reserve

The SAR ADC channel supports the various result post processing features. See the ADC post processing section for details on the features.

3.3 ADC post processing

When the results are generated from SAR ADC, any selected post processing for that channel is executed. The ADC post processing has the following features:

- Pseudo differential hardware subtraction
- Signed format
- Averaging (result accumulation, no shifting function, user should right shift by software)
- Finite impulse response filter (FIR) processing
- Limit detection

See the ADC post processing section of the PSOC™ Control C3 MCU architecture reference manual for more details.

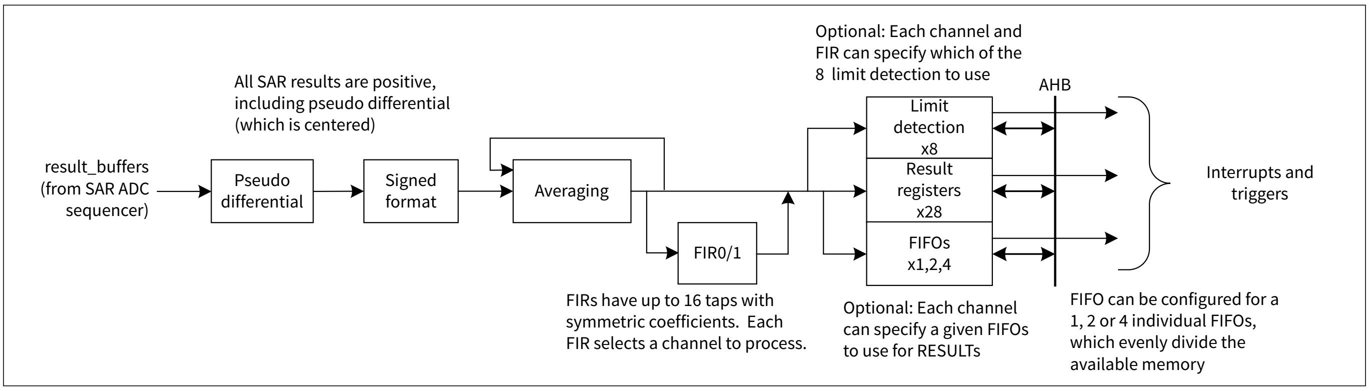

Optional: Each channel and FIR can specify which of the 8 limit detection to use
All SAR results are positive, including pseudo differential (which is centered)
AHB
result_buffers (from SAR ADC sequencer)
Pseudo differential
Signed format
Averaging
FIR0/1
Limit detection x8
Result registers x28
FIFOs x1,2,4
Interrupts and triggers
FIRs have up to 16 taps with symmetric coefficients. Each FIR selects a channel to process.
Optional: Each channel can specify a given FIFOs to use for RESULTs
FIFO can be configured for a 1, 2 or 4 individual FIFOs, which evenly divide the available memory

Figure 9 **SAR ADC post processing data flow**

Depending on the application, enable the ADC post-processing pseudo differential, averaging feature, and finite impulse response filter (FIR) to improve ADC accuracy.

4 How to improve ADC accuracy

The following sections discuss the methods to improve ADC accuracy by hardware and software.

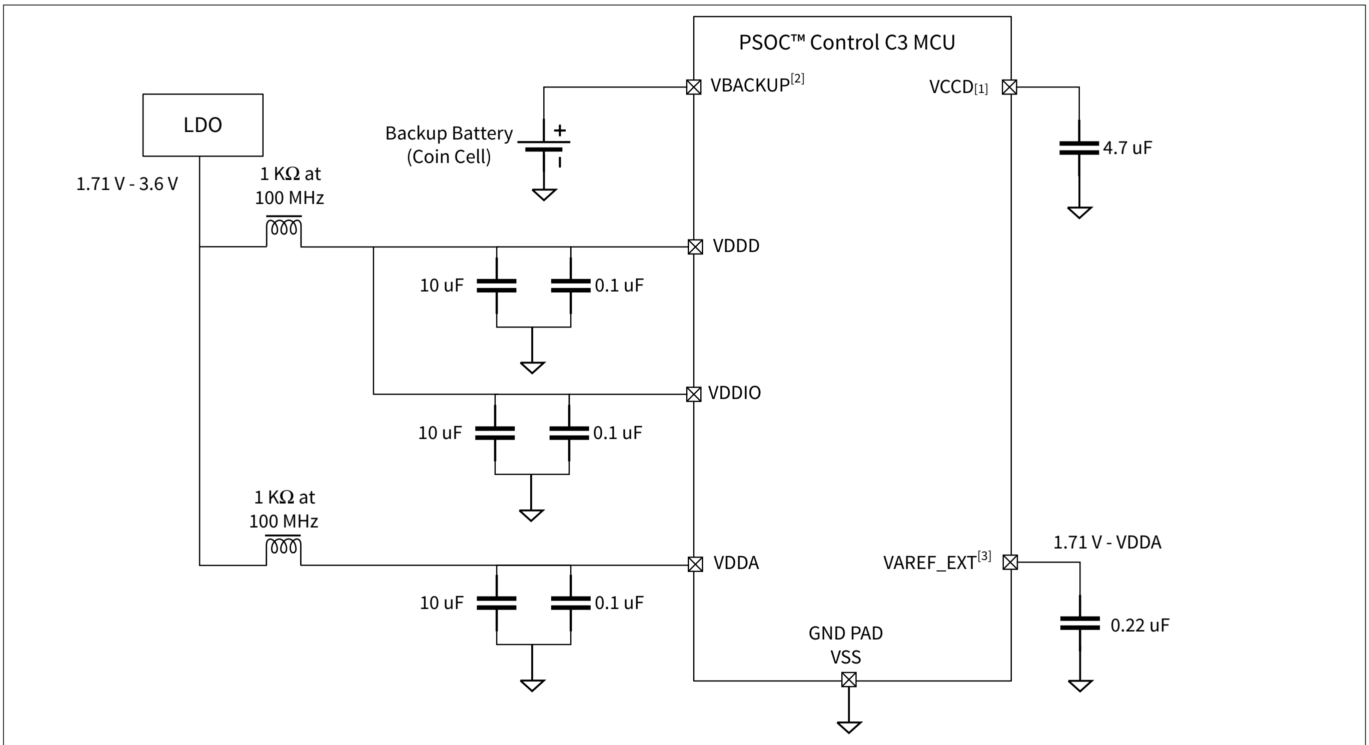
4.1 Improvement of ADC accuracy by external environment

4.1.1 MCU power supply

Linear regulators have better output in terms of noise. The mains must be stepped down, rectified, and filtered, then fed to linear regulators. It is highly recommended to connect the filter capacitors to the rectifier output. See the datasheet of the used linear regulator.

If you are using a switching power supply, it is recommended to have a linear regulator to supply the analog stage.

Figure 10 shows the typical powering configuration for the PSOC™ Control C3 MCU with a single regulated supply used for all supply pins and VBACKUP connected to a coin cell. The isolated sources can be used; for example, to isolate the analog circuits, VDDA can be connected to an independent regulated supply between 1.71 V to 3.6 V.



1. VCCD pin is only used to connect an external bypass capacitor for the integrated core logic supply LDO. Do not use to power external circuits.
2. VBACKUP can be connected to a coin cell (1.4 V to 3.6 V) or a supercapacitor depending on the application. If not used, it is connected to VDDD.
3. VAREF_EXT can either be connected to a dedicated reference voltage generator (1.71 V to VDDA) or to VDDA externally.

Figure 10 **Power supply block diagram**

For details of the power supply, see the datasheet and AN239527 - PSOC™ Control C3 MCU hardware design guide.

The VDDA pin must be connected to two external decoupling capacitors (0.1 uF Ceramic + 10 uF Tantalum or Ceramic).

The VAREF pin must be connected to a decoupling capacitor of at least 200 nF. Keep the distance between this capacitor and the package pins minimal. Use ceramic capacitors only because they have lower ESR and parasitic inductance than electrolytic capacitors.

To improve the ADC accuracy by connecting a separate external ADC reference voltage input on VAREF, the voltage on VAREF may range from 1.71 V to VDDA.

4.1.2 PCB layout recommendations

It is recommended to separate the analog and digital circuitry on the PCB. This avoids tracks crossing each other. The tracks carrying digital signals may introduce high-frequency noise in analog signals because of coupling.

It is recommended to use different planes for analog and digital grounds. If there are a lot of analog circuitry, then an analog ground plane is recommended.

It is desirable to have separate analog and digital power supplies in cases where there are a lot of analog and digital circuits external to the microcontroller. The VDDD/VDDIO, VDDA, and VAREF_EXT pins can be powered from separate power supplies.

It is also recommended to connect the analog and digital grounds in a star network. This means that you must connect the analog and digital grounds at only one point. This prevents the introduction of noise in the analog power supply circuit due to digital signal switching. This also prevents current surges from affecting the analog circuit.

For more PCB layout recommendations, check the AN239527 - PSOC™ Control C3 MCU hardware design guide.

4.1.3 Reduce crosstalk between I/Os

Due to the capacitive coupling between pins and even bonding wires inside the chip, the crosstalk between I/O will have a significant impact on the sampling accuracy of ADC, especially when the analog sampling channel of ADC is adjacent to the digital I/O.

The noise produced by crosstalk can be reduced by shielding the analog signal by placing ground tracks across it, as shown in Figure 11.

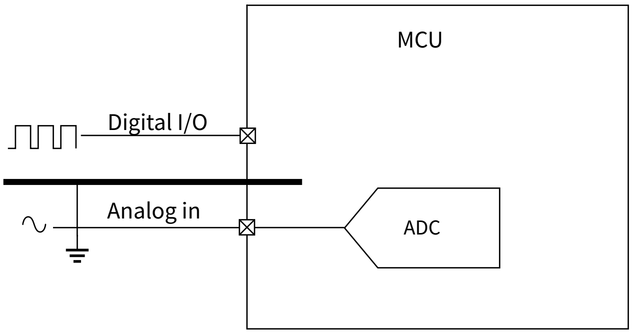


Figure 11 Crosstalk between I/Os

4.2 Software method of improving the accuracy of ADC

4.2.1 ADC calibration

The SAR ADC supports the offset, linearity, and gain calibrations, calibrating offset and linearity for each input sampler and gain for each gain setting. Offset is due to the comparator, sampler, or common mode voltage mismatches, and linearity mainly originates from process mismatches. Gain errors are due to reference buffer offsets and reference divider mismatches.

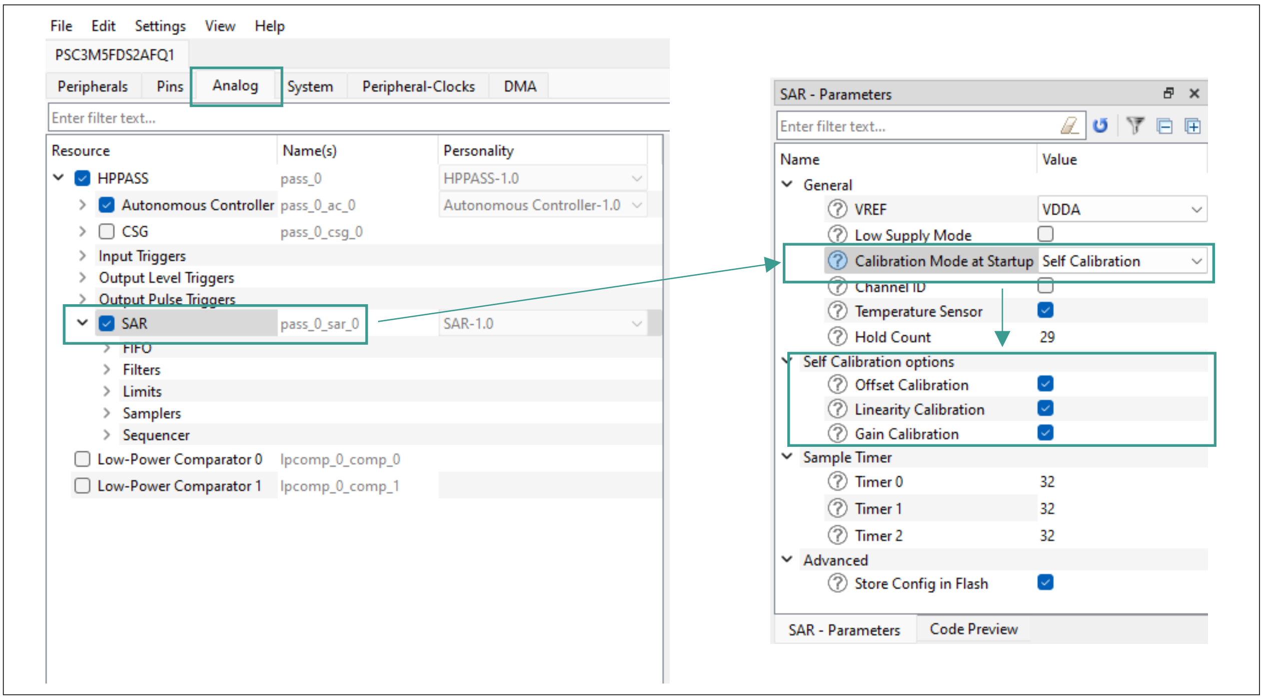
When certain events, such as changing reference voltages or gain settings, need to run at least an offset calibration because the offset does not scale with the reference voltage. Disabling and re-enabling the SAR ADC with the same configuration settings does not require another calibration run; however, it is recommended to run offset calibration if the temperature changes drastically. Reference and supply voltages must be within the operating ranges and be sufficiently stable during calibration; otherwise, calibration must be repeated.

The SAR ADC of the PSOC™ Control C3 MCU supports self-calibration and factory calibration to reduce the effects of ADC-related ADC error.

- **Self-calibration**

  Sets the STARTUP_CAL_OFFSET_EN, STARTUP_CAL_LINEARITY_EN, and STARTUP_CAL_GAIN_EN bits of HPPASS_SAR_CFG_CTRL register to respectively enable offset, linearity, and gain calibration during SAR ADC start-up. All selected calibration routines in this register are run when AC enables the SAR. The user must wait for BLOCK_READY, which indicates that SAR ADC calibration and initialization are complete, before starting the application.

  ModusToolbox™ is used to enable SAR ADC calibration. The following figure shows the self-calibration configuration in the device configurator of ModusToolbox™.



Resource	Name(s)	Personality
☑ HPPASS	pass_0	HPPASS-1.0
☑ Autonomous Controller	pass_0_ac_0	Autonomous Controller-1.0
□ CSG	pass_0_csg_0	
Input Triggers		
Output Level Triggers		
Output Pulse Triggers		
☑ SAR	pass_0_sar_0	SAR-1.0
FIFO		
Filters		
Limits		
Samplers		
Sequencer		
□ Low-Power Comparator 0	lpcomp_0_comp_0	
□ Low-Power Comparator 1	lpcomp_0_comp_1	

Name	Value
General	
VREF	VDDA
Low Supply Mode	□
Calibration Mode at Startup	Self Calibration
Channel ID	□
Temperature Sensor	☑
Hold Count	29
Self Calibration options	
Offset Calibration	☑
Linearity Calibration	☑
Gain Calibration	☑
Sample Timer	
Timer 0	32
Timer 1	32
Timer 2	32
Advanced	
Store Config in Flash	☑

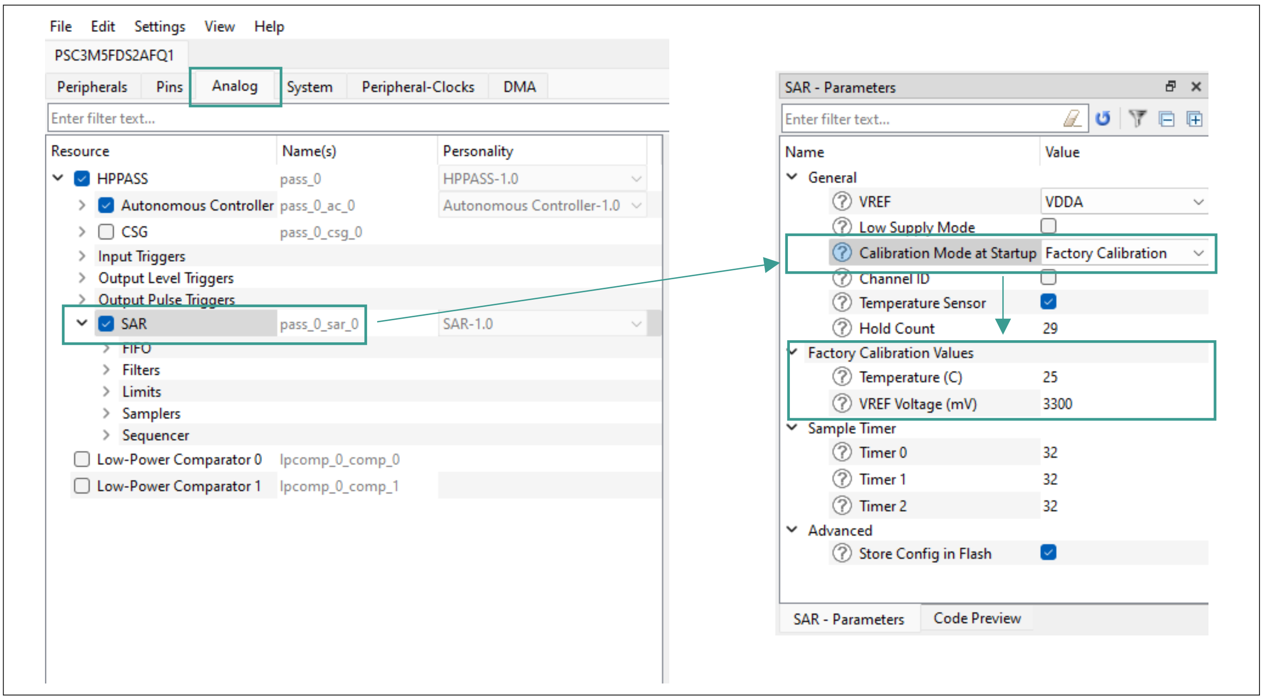
Figure 12 Enable SAR ADC self-calibration in ModusToolbox™

- **Factory-calibration**

  When the MCU is produced, the SAR ADC is calibrated at the specified reference voltage in the factory, and then store these calibration values to SFLASH. The user can copy these factory calibration values to the SAR ADC calibration registers in HPPASS initialization without enabling self-calibration to speed up HPPASS startup.

  The Infineon Peripheral Driver Library (PDL) implements the copying of factory calibration values to the SAR ADC calibration registers in the HPPASS initialization API. For more information, refer to the Peripheral Driver Library API Reference Manual.

  The factory calibration can be enabled in the device configurator of ModusToolbox™, the following figure shows the factory calibration in the device configurator.


File Edit Settings View Help
PSC3M5FDS2AFQ1
Peripherals Pins Analog System Peripheral-Clocks DMA
Enter filter text...
SAR - Parameters
Enter filter text...
SAR - Parameters Code Preview

Resource	Name(s)	Personality
☑ HPPASS	pass_0	HPPASS-1.0
☑ Autonomous Controller	pass_0_ac_0	Autonomous Controller-1.0
□ CSG	pass_0_csg_0	
Input Triggers		
Output Level Triggers		
Output Pulse Triggers		
☑ SAR	pass_0_sar_0	SAR-1.0
FIFO		
Filters		
Limits		
Samplers		
Sequencer		
□ Low-Power Comparator 0	lpcomp_0_comp_0	
□ Low-Power Comparator 1	lpcomp_0_comp_1	

Name	Value
General	
VREF	VDDA
Low Supply Mode	□
Calibration Mode at Startup	Factory Calibration
Channel ID	□
Temperature Sensor	☑
Hold Count	29
Factory Calibration Values	
Temperature (C)	25
VREF Voltage (mV)	3300
Sample Timer	
Timer 0	32
Timer 1	32
Timer 2	32
Advanced	
Store Config in Flash	☑

Figure 13 **Enable SAR ADC factory calibration in ModusToolbox™**

***Note**:* *Infineon recommends that users use factory calibration in application. Enable factory calibration in the device configurator of ModusToolbox™, and the HPPASS PDL initialization API `Cy_HPPASS_Init()` will automatically handle it.*

***Note**:* *When the device enters DeepSleep mode, the value of the SAR ADC calibration registers will be lost. To reduce another SAR ADC calibration wait time when wake up from DeepSleep mode, it should read back and store the SAR ADC calibration register values to DeepSleep capable RAM. On wake up, write these values back into the SAR ADC calibration registers. The user can register the PDL API `Cy_HPPASS_DeepSleepCallback()` to system power management (SysPm). About the startup flow, see the HPPASS startup chapter of the PSOC™ Control C3 MCU architecture reference manual for more details.*

Because the offset calibration is temperature-dependent, when temperature changes drastically, it is recommended to update the offset calibration registers for every sampler. There are two methods to implement this:

- The first method uses the factory calibration values to adjust the offset calibration values by using an internal temperature sensor. SFLASH stores -40, 25, and 125 offset calibration values (SFLASH_SAR_CALOFFST_0/1/2/3_N40C/125C/25C) for every sampler at the factory.

  The user can call the PDL API `Cy_HPPASS_SAR_Adjust()` to adjust offset calibration values based on the current temperature and reference voltage. For more information, see the Peripheral Driver Library API Reference Manual.

  Infineon recommends that users use this method to reduce the calibration time
- The second method is to re-initialize HPPASS and enable offset self-calibration during start-up, but this method involves additional time

4.2.2 Pseudo differential measurement

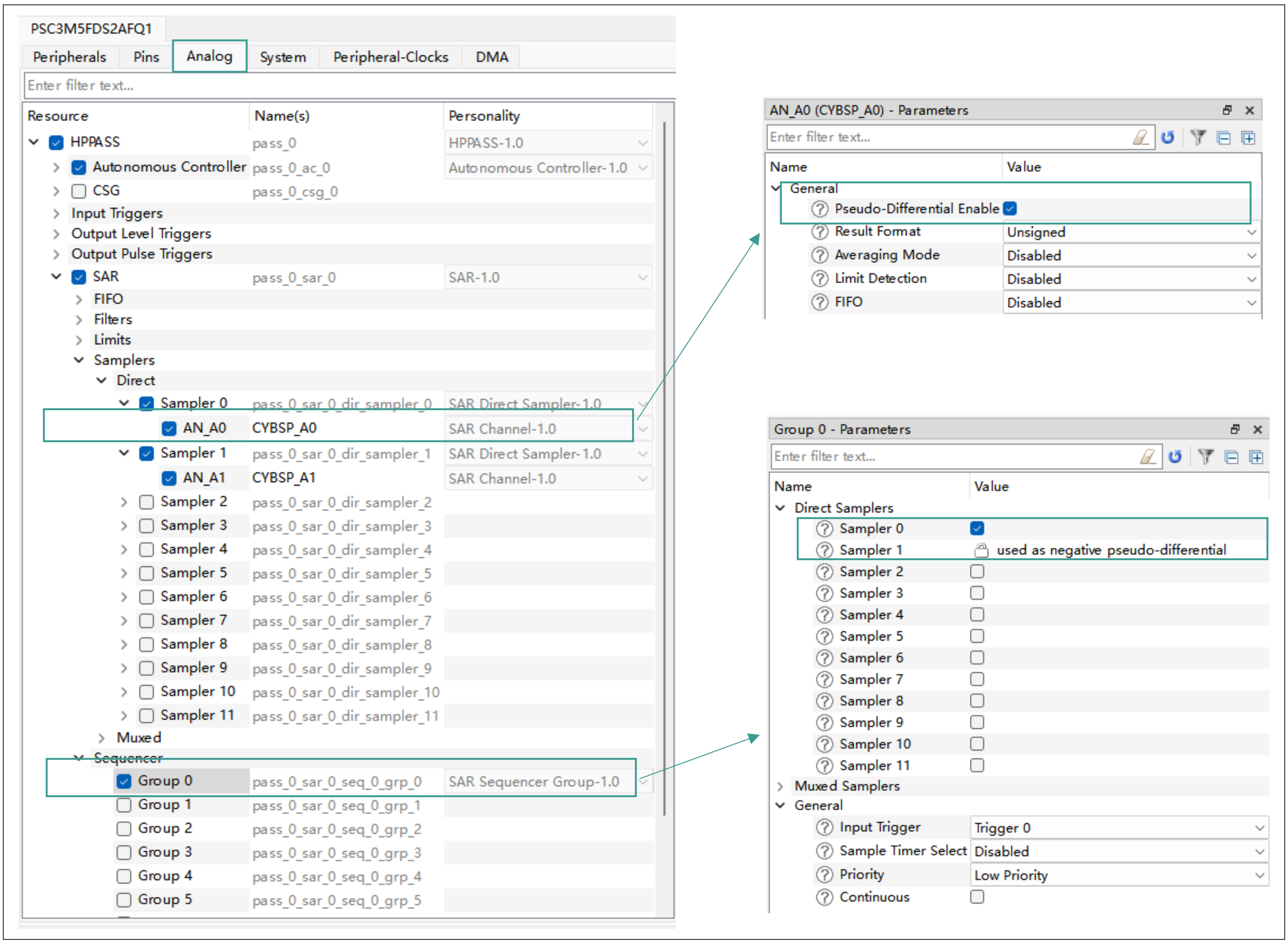
The SAR ADC differential input in motor control applications is used to enhance the noise immunity and improve signal integrity by measuring the difference between two voltages, thereby canceling out common-mode noise and disturbances that can affect the accuracy of motor control.

Although all the inputs of the PSOC™ Control C3 MCU ADC are single-ended inputs, but the SAR ADC post processing block supports pseudo differential inputs. Based on a given channel configuration, two channels can be specified as differential inputs, which are sampled simultaneously, converted and subtracted in hardware, and presented to the user as one channel result. See the PSOC™ Control C3 MCU architecture reference manual for more details of pseudo differential.

The SAR ADC has direct channels and muxed channels, the following describes the channels definition for pseudo differential operation. For more details about the ADC channels, see the ADC channel mapping section.

- **Direct channels:** Only even/odd sampler pairs are supported (where the even sampler is considered the positive input and the odd sampler is the negative input). Therefore, a pseudo differential configuration can only be specified and is controlled by evenly numbered direct channels. Pseudo differential configuration in the odd direct channels will be ignored
- **Muxed samplers:** Only even/odd MUX sampler pairs are supported. Therefore, a pseudo differential configuration can only be specified in the channels associated with the even MUX0 and MUX2. Pseudo differential configuration in channels associated with MUX1 and MUX3 will be ignored

The pseudo differential can be easily configured in the device configurator of the ModusToolbox™. Figure 14 shows the configuration in the device configurator where AN_A0 is the differential positive input and AN_A1 is the differential negative input, and enable the samplers in ADC group 0.



PSC3M5FDS2AFQ1

Resource	Name(s)	Personality
☑ HPPASS	pass_0	HPPASS-1.0
☑ Autonomous Controller	pass_0_ac_0	Autonomous Controller-1.0
□ CSG	pass_0_csg_0	
Input Triggers		
Output Level Triggers		
Output Pulse Triggers		
☑ SAR	pass_0_sar_0	SAR-1.0
FIFO		
Filters		
Limits		
Samplers		
Direct		
☑ Sampler 0	pass_0_sar_0_dir_sampler_0	SAR Direct Sampler-1.0
☑ AN_A0	CYBSP_A0	SAR Channel-1.0
☑ Sampler 1	pass_0_sar_0_dir_sampler_1	SAR Direct Sampler-1.0
☑ AN_A1	CYBSP_A1	SAR Channel-1.0
□ Sampler 2	pass_0_sar_0_dir_sampler_2	
□ Sampler 3	pass_0_sar_0_dir_sampler_3	
□ Sampler 4	pass_0_sar_0_dir_sampler_4	
□ Sampler 5	pass_0_sar_0_dir_sampler_5	
□ Sampler 6	pass_0_sar_0_dir_sampler_6	
□ Sampler 7	pass_0_sar_0_dir_sampler_7	
□ Sampler 8	pass_0_sar_0_dir_sampler_8	
□ Sampler 9	pass_0_sar_0_dir_sampler_9	
□ Sampler 10	pass_0_sar_0_dir_sampler_10	
□ Sampler 11	pass_0_sar_0_dir_sampler_11	
Muxed		
Sequencer		
☑ Group 0	pass_0_sar_0_seq_0_grp_0	SAR Sequencer Group-1.0
□ Group 1	pass_0_sar_0_seq_0_grp_1	
□ Group 2	pass_0_sar_0_seq_0_grp_2	
□ Group 3	pass_0_sar_0_seq_0_grp_3	
□ Group 4	pass_0_sar_0_seq_0_grp_4	
□ Group 5	pass_0_sar_0_seq_0_grp_5	

AN_A0 (CYBSP_A0) - Parameters

Name	Value
General	
Pseudo-Differential Enable	☑
Result Format	Unsigned
Averaging Mode	Disabled
Limit Detection	Disabled
FIFO	Disabled

Group 0 - Parameters

Name	Value
Direct Samplers	
Sampler 0	☑
Sampler 1	used as negative pseudo-differential
Sampler 2	□
Sampler 3	□
Sampler 4	□
Sampler 5	□
Sampler 6	□
Sampler 7	□
Sampler 8	□
Sampler 9	□
Sampler 10	□
Sampler 11	□
Muxed Samplers	
General	
Input Trigger	Trigger 0
Sample Timer Select	Disabled
Priority	Low Priority
Continuous	□

Figure 14 **ADC pseudo differential configuration in ModusToolbox™**

4.2.3 Averaging samples

The principle of this method is to increase ADC precision but decrease ADC conversion speed. If the measured analog signal produces unstable ADC values, then the mean value of the given input signal can be obtained by averaging a set of values. Variation can be caused by signal noise or noise generated by the microcontroller itself (high-speed digital signals coupled to the analog input signal).

The advantage of averaging is to improve ADC precision without any hardware changes. The disadvantage is that the conversion speed is lower as well as the frequency response.

The ADC channel post-processing of the PSOC™ Control C3 MCU supports hardware averaging operation, note that it only has the result accumulation function, no shift function, user should right shift by software. The number of samples can be configured to 2, 4, 8, or 16. When averaging is selected for a given channel, a result is only generated and stored for every N triggers. See the SAR ADC Post processing chapter of PSOC™ Control C3 MCU architecture reference manual for the details.

The user has a choice of triggering a given group, which contains averaged channels N times (or periodically), or continuous triggering may be set in the group, or the autonomous controller (AC) may be used to periodically re-trigger the given group. See the PSOC™ Control C3 MCU architecture reference manual for the details of HPPASS AC.

4.2.3.1 Averaging example using AC

This example uses AC to periodically re-trigger the ADC group. The benefit of the AC option is for the case when an averaged conversion is triggered by one external event, or when the measurement is at a random rate, for example, triggered by firmware.

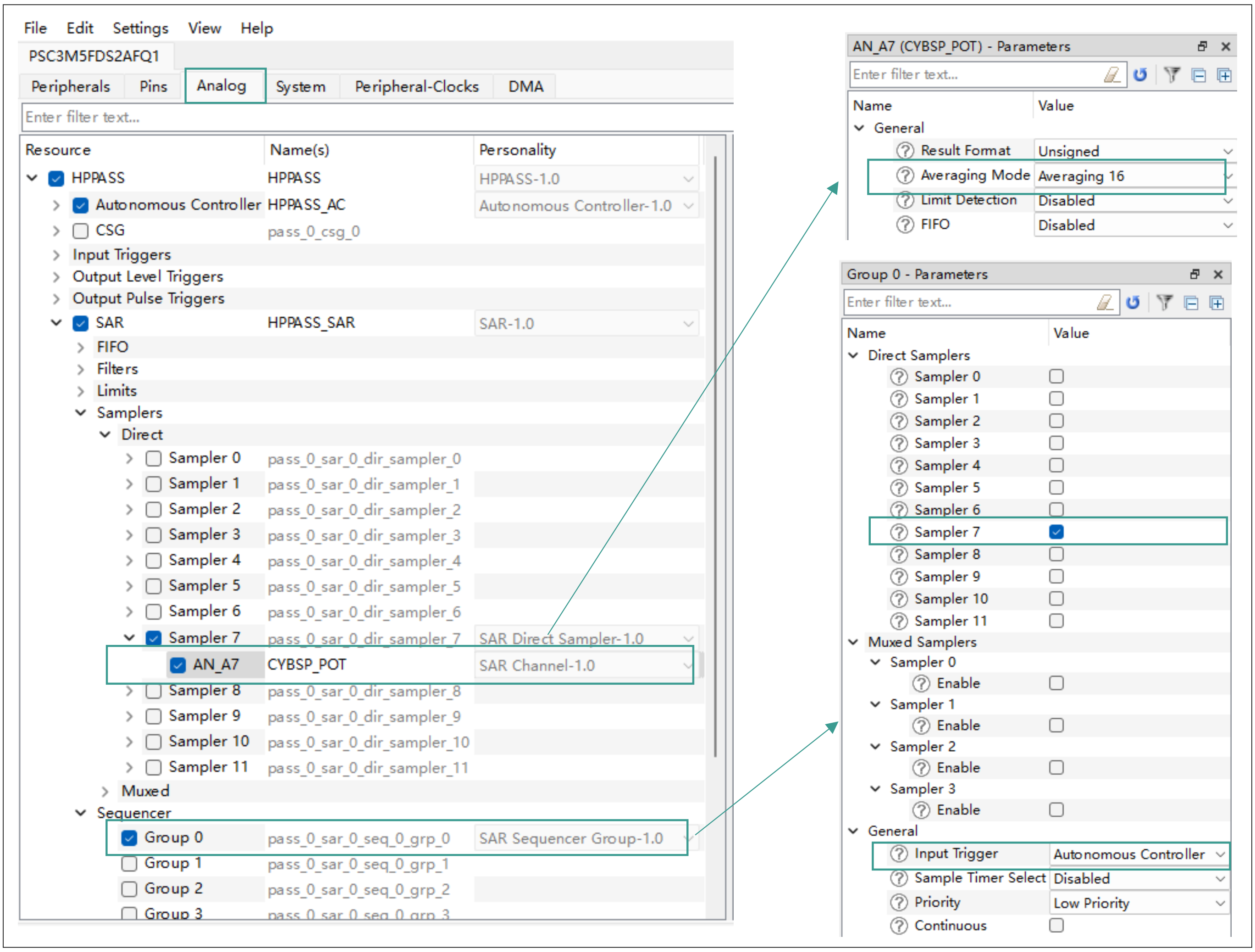
The following routine is AC state code fragments, it waits for the external hardware trigger (use GPIO edge to trigger in this example), then triggers the SAR ADC group 0 N times (16 in this example) each time waiting for the SAR ADC group 0 conversion done (SAR_ENTRY_DONE) condition, and then jumps back to wait for the next external hardware trigger.

```
STATE:TOP
    ACTION=WAIT_FOR   COND=HW_TR0_IN

STATE:TR_SAR
    SAR_TR=0x01   ACTION=WAIT_FOR   COND=SAR_ENTRY0_DONE

STATE:
    ACTION=BRANCH_IF_FALSE   BR_ADDR=TR_SAR   COND=CNT_DONE   CNT=15

STATE:
    ACTION=BRANCH_IF_TRUE   BR_ADDR=TOP   COND=TRUE
```

***Note**: The above code fragments are written in an easy-to-read format (versus register writes), in which a state is delimited by the STATE: with an optional label and the code as <FIELD_NAME> = <FIELD_VALUE> (from HPPASS_ACTRLR_TTCFG registers). See the PSOC™ Control C3 MCU architecture reference manual for more details.*

Configure HPPASS AC states, hardware trigger input 0, SAR ADC channel 7, and group 0 in the device configurator of ModusToolbox™. Figure 15 shows the SAR ADC configuration. It enables sampler 7 and configures the averaging mode of the AN_AN7 channel to 16. Sampler 7 has been enabled in group 0, and select an autonomous controller (AC) to trigger group 0.

The following example uses SAR ADC channel 7 to demonstrate the averaging function; the ADC channel 7 is connected to the potentiometer in the PSOC™ Control C3 Evaluation Kit. See the KIT_PSC3M5_EVK PSOC™ Control C3 Evaluation Kit guide for details.



File Edit Settings View Help
PSC3M5FDS2AFQ1
Peripherals Pins Analog System Peripheral-Clocks DMA
Enter filter text...

Resource	Name(s)	Personality
☑ HPPASS	HPPASS	HPPASS-1.0
☑ Autonomous Controller	HPPASS_AC	Autonomous Controller-1.0
☐ CSG	pass_0_csg_0	
Input Triggers		
Output Level Triggers		
Output Pulse Triggers		
☑ SAR	HPPASS_SAR	SAR-1.0
FIFO		
Filters		
Limits		
Samplers		
Direct		
☐ Sampler 0	pass_0_sar_0_dir_sampler_0	
☐ Sampler 1	pass_0_sar_0_dir_sampler_1	
☐ Sampler 2	pass_0_sar_0_dir_sampler_2	
☐ Sampler 3	pass_0_sar_0_dir_sampler_3	
☐ Sampler 4	pass_0_sar_0_dir_sampler_4	
☐ Sampler 5	pass_0_sar_0_dir_sampler_5	
☐ Sampler 6	pass_0_sar_0_dir_sampler_6	
☑ Sampler 7	pass_0_sar_0_dir_sampler_7	SAR Direct Sampler-1.0
☑ AN_A7	CYBSP_POT	SAR Channel-1.0
☐ Sampler 8	pass_0_sar_0_dir_sampler_8	
☐ Sampler 9	pass_0_sar_0_dir_sampler_9	
☐ Sampler 10	pass_0_sar_0_dir_sampler_10	
☐ Sampler 11	pass_0_sar_0_dir_sampler_11	
Muxed		
Sequencer		
☑ Group 0	pass_0_sar_0_seq_0_grp_0	SAR Sequencer Group-1.0
☐ Group 1	pass_0_sar_0_seq_0_grp_1	
☐ Group 2	pass_0_sar_0_seq_0_grp_2	
☐ Group 3	pass_0_sar_0_seq_0_grp_3	

AN_A7 (CYBSP_POT) - Parameters

Name	Value
General	
Result Format	Unsigned
Averaging Mode	Averaging 16
Limit Detection	Disabled
FIFO	Disabled

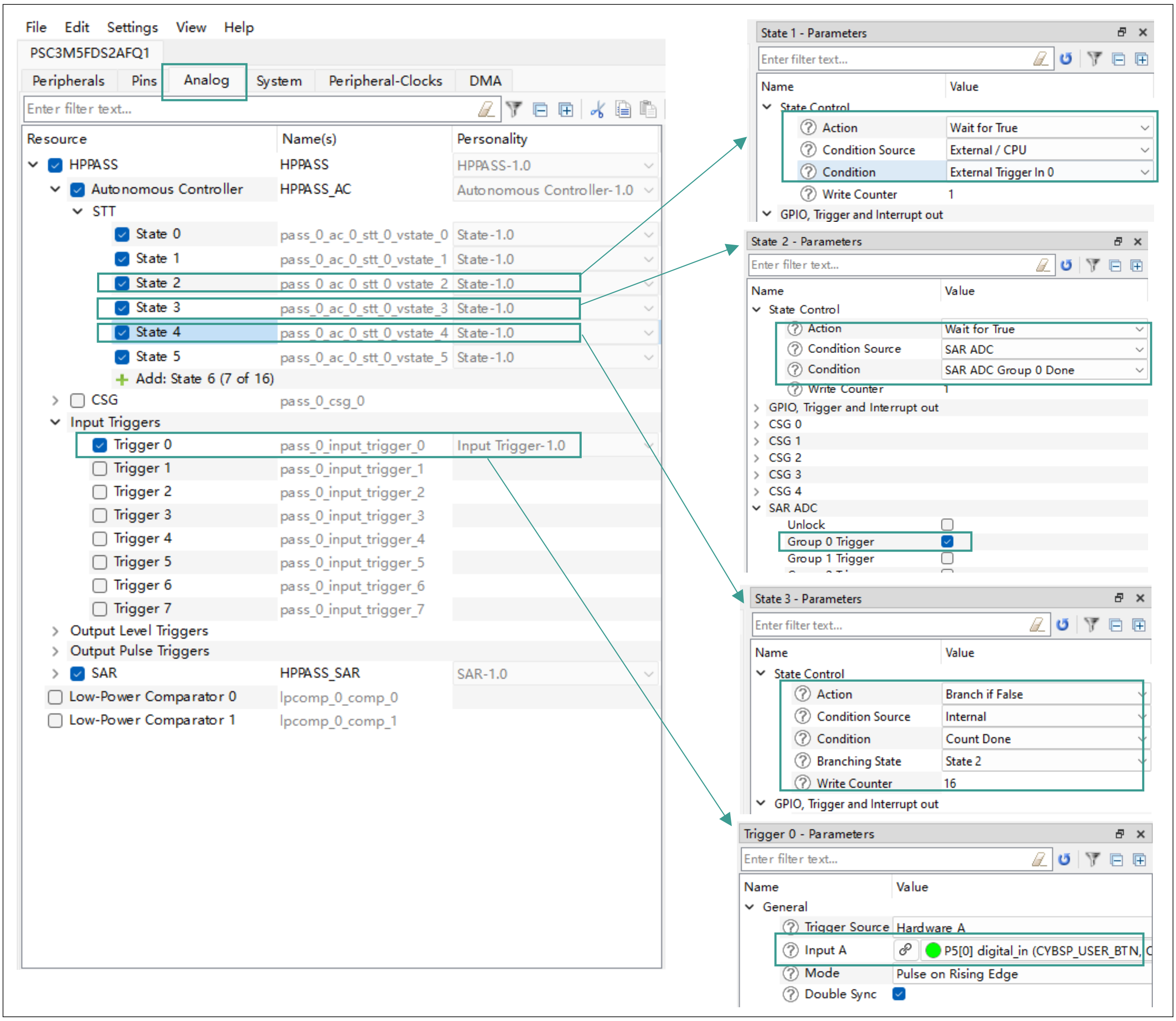
Group 0 - Parameters

Name	Value
Direct Samplers	
Sampler 0	☐
Sampler 1	☐
Sampler 2	☐
Sampler 3	☐
Sampler 4	☐
Sampler 5	☐
Sampler 6	☐
Sampler 7	☑
Sampler 8	☐
Sampler 9	☐
Sampler 10	☐
Sampler 11	☐
Muxed Samplers	
Sampler 0	
Enable	☐
Sampler 1	
Enable	☐
Sampler 2	
Enable	☐
Sampler 3	
Enable	☐
General	
Input Trigger	Autonomous Controller
Sample Timer Select	Disabled
Priority	Low Priority
Continuous	☐

Figure 15 SAR ADC channel averaging and group configuration in ModusToolbox™

The AC states and external input trigger configuration are shown in Figure 16. The example configures external input trigger 0 to GPIO P5.0 and P5.0 connects to the user button in the PSOC™ Control C3 Evaluation Kit. Follow the above code fragments to configure AC states:

- State 0: Enable SAR ADC and wait for the block ready
- State 1: Wait for external input trigger 0 (wait for the user button to be pressed)
- State 2: Trigger the SAR ADC group 0 conversion, and then wait for the SAR ADC group 0 conversion to be done
- State 3: The initial value of the counter is 16. Check whether the counter is 0. If it is not 0, jump to state 2; if it is equal to 0, jump to the next state
- State 4: Jump to state 1 to wait for the next external hardware trigger
- State 5: Stop AC



Resource	Name(s)	Personality
☑ HPPASS	HPPASS	HPPASS-1.0
☑ Autonomous Controller	HPPASS_AC	Autonomous Controller-1.0
STT		
☑ State 0	pass_0_ac_0_stt_0_vstate_0	State-1.0
☑ State 1	pass_0_ac_0_stt_0_vstate_1	State-1.0
☑ State 2	pass_0_ac_0_stt_0_vstate_2	State-1.0
☑ State 3	pass_0_ac_0_stt_0_vstate_3	State-1.0
☑ State 4	pass_0_ac_0_stt_0_vstate_4	State-1.0
☑ State 5	pass_0_ac_0_stt_0_vstate_5	State-1.0
Add: State 6 (7 of 16)		
□ CSG	pass_0_csg_0	
Input Triggers		
☑ Trigger 0	pass_0_input_trigger_0	Input Trigger-1.0
□ Trigger 1	pass_0_input_trigger_1	
□ Trigger 2	pass_0_input_trigger_2	
□ Trigger 3	pass_0_input_trigger_3	
□ Trigger 4	pass_0_input_trigger_4	
□ Trigger 5	pass_0_input_trigger_5	
□ Trigger 6	pass_0_input_trigger_6	
□ Trigger 7	pass_0_input_trigger_7	
Output Level Triggers		
Output Pulse Triggers		
☑ SAR	HPPASS_SAR	SAR-1.0
□ Low-Power Comparator 0	lpcomp_0_comp_0	
□ Low-Power Comparator 1	lpcomp_0_comp_1	

State 1 - Parameters

Name	Value
State Control	
Action	Wait for True
Condition Source	External / CPU
Condition	External Trigger In 0
Write Counter	1
GPIO, Trigger and Interrupt out	

State 2 - Parameters

Name	Value
State Control	
Action	Wait for True
Condition Source	SAR ADC
Condition	SAR ADC Group 0 Done
Write Counter	1
GPIO, Trigger and Interrupt out	
CSG 0	
CSG 1	
CSG 2	
CSG 3	
CSG 4	
SAR ADC	
Unlock	□
Group 0 Trigger	☑
Group 1 Trigger	□

State 3 - Parameters

Name	Value
State Control	
Action	Branch if False
Condition Source	Internal
Condition	Count Done
Branching State	State 2
Write Counter	16
GPIO, Trigger and Interrupt out	

Trigger 0 - Parameters

Name	Value
General	
Trigger Source	Hardware A
Input A	P5[0] digital_in (CYBSP_USER_BTN, C
Mode	Pulse on Rising Edge
Double Sync	☑

Figure 16 AC and hardware input trigger configuration in ModusToolbox™

The following code snippet demonstrates the example program to read the ADC channel averaging result. When the user button is pressed, AC automatically triggers SAR ADC group 0 16 times, and then SAR ADC generates group 0 done interrupt and read the channel result from the result register.

Code listing 1: `main.c` file

```
#include "cyhal.h"
#include "cybsp.h"
#include "cy_pdl.h"
#include "cy_retarget_io.h"

/*******************************************************************************
* Global Variables
*******************************************************************************/
/* The interrupt configuration structure of ADC group 0 done */
cy_stc_sysint_t adc_group0_done_intr_config =
{
    .intrSrc = pass_interrupt_sar_entry_done_0_IRQn,
    .intrPriority = 0U,
};

/* ADC channel result buffer */
uint16_t adc_result_buf = 0;

/* ADC group 0 interrupt flag */
volatile bool adc_group0_int_flag = false;

/*******************************************************************************
* Function Prototypes
*******************************************************************************/
/* ADC group 0 done interrupt handler */
void adc_group0_done_intr_handler(void);

/*******************************************************************************
* Function Name: main
********************************************************************************
* Summary:
* This is the main function.
*
* Parameters:
*  void
*
* Return:
*  int
*
*******************************************************************************/
int main(void)
{
    cy_rslt_t result;

    /* Initialize the device and board peripherals */
    result = cybsp_init();
    /* Board init failed. Stop program execution */
    if (result != CY_RSLT_SUCCESS)
    {
        CY_ASSERT(0);
    }
    /* Initialize retarget-io to use the debug UART port */
    result = cy_retarget_io_init_fc(CYBSP_DEBUG_UART_TX, CYBSP_DEBUG_UART_RX,
            CYBSP_DEBUG_UART_CTS,CYBSP_DEBUG_UART_RTS,CY_RETARGET_IO_BAUDRATE);
    /* retarget-io init failed. Stop program execution */
    if (result != CY_RSLT_SUCCESS)
    {
        CY_ASSERT(0);
    }

    /* Configure ADC result interrupt */
    Cy_HPPASS_SAR_Result_SetInterruptMask(CY_HPPASS_INTR_SAR_RESULT_GROUP_0);
    Cy_SysInt_Init(&adc_group0_done_intr_config, adc_group0_done_intr_handler);
    NVIC_EnableIRQ(adc_group0_done_intr_config.intrSrc);

    /* Start the HPPASS autonomous controller (AC) from state 0 */
    if(CY_HPPASS_SUCCESS != Cy_HPPASS_AC_Start(0U, 1000U))
    {
        CY_ASSERT(0);
    }

    /* Clear global variables */
    adc_group0_int_flag = false;

    /* \x1b[2J\x1b[;H - ANSI ESC sequence for clear screen */
    printf("\x1b[2J\x1b[;H");

printf("********************************************************************************\r\n");
    printf("PSoC Contorl C3 MCU: HPPASS SAR ADC averaging example\r\n");
    printf("CPU frequency: %ld Hz\r\n", (unsigned long)cy_delayFreqHz);

printf("********************************************************************************\r\n");
    printf("Press SW2 key to trigger SAR ADC conversion\r\n");

    /* Enable global interrupts */
    __enable_irq();

    for (;;)
    {
        /* Check whether the ADC conversion is complete */
        if(adc_group0_int_flag)
        {
            adc_group0_int_flag = false;
            /* Convert the ADC result to voltage */
            float32_t volts = Cy_HPPASS_SAR_CountsTo_Volts(CY_HPPASS_SAR_CHAN_7_IDX, 3300,
adc_result_buf);
            printf("ADC AN_A7 channel result = 0x%x, voltage = %.4fV\r\n", adc_result_buf,
volts);
        }
    }
}

/*******************************************************************************
```

```
* Function Name: adc_group0_done_intr_handler
********************************************************************************
* Summary:
* This function is the ADC group 0 done interrupt handler
*
* Parameters:
*  void
*
* Return:
*  void
*
*******************************************************************************/
void adc_group0_done_intr_handler(void)
{
    /* Clear SAR result interrupt */
    Cy_HPPASS_SAR_Result_ClearInterrupt(CY_HPPASS_INTR_SAR_RESULT_GROUP_0);
    /* Read channel result from register */
    adc_result_buf = Cy_HPPASS_SAR_Result_ChannelRead(CY_HPPASS_SAR_CHAN_7_IDX);
    adc_group0_int_flag = true;
}
```

4.2.4 Digital signal filtering

This method uses digital signal-processing techniques. In principle, averaging is also a simple digital filter with a specific frequency response. However, if the noise frequency spectrum is known, a digital filter can be designed that minimizes noise influence and maximizes ADC frequency response. For example, if the noise in the measured signal is coming from the 50 Hz power lines, then an appropriate digital filter suppresses only the 50 Hz frequency and delivers the data signal without this noise.

The ADC post-processing block of PSOC™ Control C3 MCU has two finite impulse response (FIR) resources; each of the FIR resources can select one of the available channel results to process. User can use this FIR to filter out noise at a specified frequency. The FIR supported 16-tap with symmetric coefficients. The following figure shows the FIR filter construction.

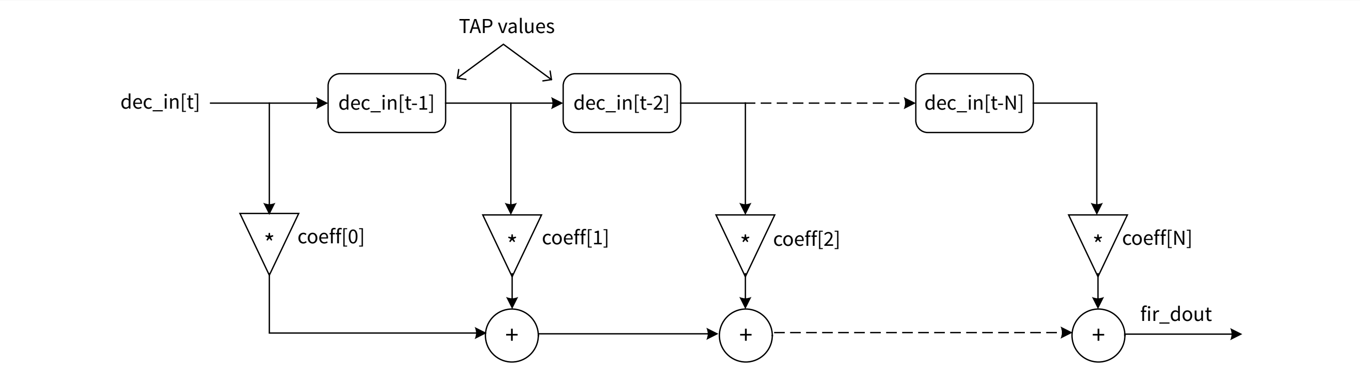
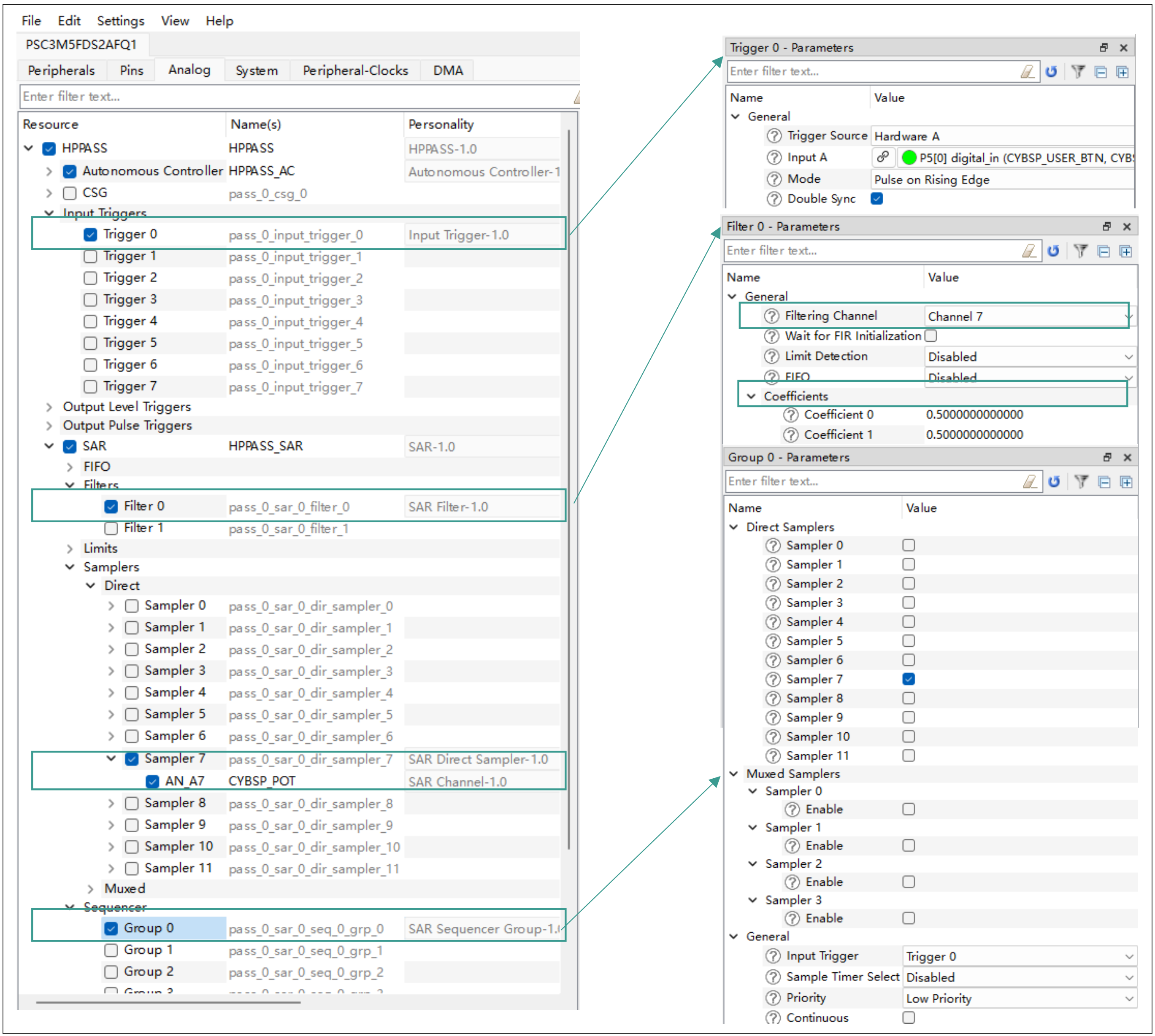


Figure 17 **FIR filter construction**

Dec_in signifies the input data to the FIR from the decimator, while the [t] signifies the instance in time. As data enters the FIR, it is first multiplied by the initial coefficient value (coeff[0]), then it is added with the product (*) of the previous decimator data [t-1] and the next stored coefficient value (coeff[1]), and so on. This continues until it reaches the number of TAPs enabled (N), which is a programmable value up to 16. For more details on FIR, see the SAR ADC Post processing chapter of PSOC™ Control C3 MCU architecture reference manual.

4.2.4.1 FIR example

The following example configures SAR ADC channel 7, enables SAR ADC sampler 7 in group 0, and configures GPIO P5.0 to trigger SAR ADC group 0. Then enable filter FIR 0 and set the filtering channel of FIR 0 to SAR ADC channel 7. Figure 18 shows the SAR ADC and filter configuration.



Resource	Name(s)	Personality
☑ HPPASS	HPPASS	HPPASS-1.0
☑ Autonomous Controller	HPPASS_AC	Autonomous Controller-1
□ CSG	pass_0_csg_0	
Input Triggers		
☑ Trigger 0	pass_0_input_trigger_0	Input Trigger-1.0
□ Trigger 1	pass_0_input_trigger_1	
□ Trigger 2	pass_0_input_trigger_2	
□ Trigger 3	pass_0_input_trigger_3	
□ Trigger 4	pass_0_input_trigger_4	
□ Trigger 5	pass_0_input_trigger_5	
□ Trigger 6	pass_0_input_trigger_6	
□ Trigger 7	pass_0_input_trigger_7	
Output Level Triggers		
Output Pulse Triggers		
☑ SAR	HPPASS_SAR	SAR-1.0
FIFO		
Filters		
☑ Filter 0	pass_0_sar_0_filter_0	SAR Filter-1.0
□ Filter 1	pass_0_sar_0_filter_1	
Limits		
Samplers		
Direct		
□ Sampler 0	pass_0_sar_0_dir_sampler_0	
□ Sampler 1	pass_0_sar_0_dir_sampler_1	
□ Sampler 2	pass_0_sar_0_dir_sampler_2	
□ Sampler 3	pass_0_sar_0_dir_sampler_3	
□ Sampler 4	pass_0_sar_0_dir_sampler_4	
□ Sampler 5	pass_0_sar_0_dir_sampler_5	
□ Sampler 6	pass_0_sar_0_dir_sampler_6	
☑ Sampler 7	pass_0_sar_0_dir_sampler_7	SAR Direct Sampler-1.0
☑ AN_A7	CYBSP_POT	SAR Channel-1.0
□ Sampler 8	pass_0_sar_0_dir_sampler_8	
□ Sampler 9	pass_0_sar_0_dir_sampler_9	
□ Sampler 10	pass_0_sar_0_dir_sampler_10	
□ Sampler 11	pass_0_sar_0_dir_sampler_11	
Muxed		
Sequencer		
☑ Group 0	pass_0_sar_0_seq_grp_0	SAR Sequencer Group-1.
□ Group 1	pass_0_sar_0_seq_grp_1	
□ Group 2	pass_0_sar_0_seq_grp_2	

Trigger 0 - Parameters

Name	Value
General	
Trigger Source	Hardware A
Input A	P5[0] digital_in (CYBSP_USER_BTN, CYB
Mode	Pulse on Rising Edge
Double Sync	☑

Filter 0 - Parameters

Name	Value
General	
Filtering Channel	Channel 7
Wait for FIR Initialization	□
Limit Detection	Disabled
FIFO	Disabled
Coefficients	
Coefficient 0	0.5000000000000
Coefficient 1	0.5000000000000

Group 0 - Parameters

Name	Value
Direct Samplers	
Sampler 0	□
Sampler 1	□
Sampler 2	□
Sampler 3	□
Sampler 4	□
Sampler 5	□
Sampler 6	□
Sampler 7	☑
Sampler 8	□
Sampler 9	□
Sampler 10	□
Sampler 11	□
Muxed Samplers	
Sampler 0	
Enable	□
Sampler 1	
Enable	□
Sampler 2	
Enable	□
Sampler 3	
Enable	□
General	
Input Trigger	Trigger 0
Sample Timer Select	Disabled
Priority	Low Priority
Continuous	□

Figure 18 SAR ADC and filter configuration in ModusToolbox™

The user has the choice to get a result on every conversion or set "wait for FIR initialization" in the FIR configuration to wait for 16 samples before the first valid result is available by the `Cy_HPPASS_SAR_Result_FirRead()` function or FIFO.

The FIR supported 16-tap with symmetric coefficients, use third-party tools (for example, the filter designer tool of MATLAB) to generate coefficients, and set these coefficients in the FIR configuration of the device configurator.

References

The following are the PSOC™ Control C3 family series datasheets and reference manuals. Contact Technical support to obtain more documents.

- Datasheets
  - PSOC™ Control C3 - PSC3P5xD, PSC3M5xD datasheet
  - PSOC™ Control C3 - PSC3P2xD, PSC3M3xD datasheet
- Reference manuals
  - PSOC™ Control C3 MCU architecture reference manual
  - PSOC™ Control C3 MCU registers reference manual
- Application notes
  - AN238329 - Getting started with PSOC™ Control C3 MCU on ModusToolbox™ software
  - AN239527 - PSOC™ Control C3 MCU hardware design guide
  - AN239385 - KIT_PSC3M5_EVK PSOC™ Control C3 Evaluation Kit guide

Revision history

Document revision	Date	Description of changes
**	2024-05-14	Initial release.
*A	2024-09-24	Added explanation for ADC averaging function in PSOC™ Control C3 ADC features, ADC post processing, and Averaging samples sections. Added Pseudo differential measurement section.
*B	2024-12-03	Changed distribution from restricted to public.

Trademarks

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

Edition 2024-12-03

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about any aspect of this document?

Email: erratum@infineon.com

Document reference
IFX-epo1708593150771

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.

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.