AN13523

LPC553x/LPC55S3x ADC with Hardware Trigger and ADC Calculator Tool

Rev. 4 — 20 November 2023 Application note

Document information

Information	Content
Keywords	AN13523, LPC553x, ADC, CTimer, INPUTMUX
Abstract	This application note introduces the ADC features available with the LPC553x/LPC55S3x device and the attached tool to calculate sampling time or source impedance.

1 Introduction

The LPC553x/LPC55S3x MCU family is part of the EdgeVerse Edge computing platform and is built on the general-purpose Cortex-M33-based microcontroller introduced with the LPC5500 series. LPC553x/LPC55S3x uses a 16-bit analog-to-digital converter (ADC) which is a dual successive approximation ADC. The ADC allows for differential 16/13-bit resolution and single-ended 16/12-bit resolution operations.

This application note introduces the ADC features available with the LPC553x/LPC55S3x device and the attached tool to calculate sampling time or source impedance. An example is provided to demonstrate the hardware triggering capability implemented with the Input Multiplexing (INPUTMUX) module using the CTimer to trigger ADC conversions.

2 ADC features

The ADC module has two instances, namely ADC0 and ADC1, with the following features:

- Linear successive approximation algorithm
- Differential operation with 16-bit or 13-bit resolution
- Single-ended operation with 16-bit or 12-bit resolution
- Support for two simultaneous single-ended conversions

For conversion of external pins, a channel support is provided for analog input channels (up to 20 channels) from internal sources. It has a configurable analog input sample time and speed options to accommodate low-power modes. Also, it supports up to four trigger sources with different priority.

The ADC module supports three different modes of operation, as shown in Table 1.

Table 1. Different modes of operation

Modes	Description
Run	Normal operation
Deep-sleep or Sleep	Operation continues if the Doze Enable (CTRL[DOZEN]) bit is clear and the module is using an external or internal clock source, which operates during Deep-sleep/Sleep modes.
Deep power down	The Doze Enable (CTRL[DOZEN]) bit is ignored and the module waits for the current transfer to complete any pending operation before acknowledging Deep Power-down mode entry.

3 ADC trigger interconnections

ADC command execution is initiated from up to four trigger sources. Each trigger can be software generated by writing 0b1 to the corresponding SWTRIG[SWTn] bit field. Alternatively, hardware triggers can be generated from asynchronous input sources at the periphery of the module. For example, to trigger a conversion periodically, use a PWM signal. When a hardware trigger input is enabled, hardware trigger events are detected on the rising-edge of the associated hardware trigger source. Each trigger source is assigned a priority via the associated priority control field (TCTRLa[TPRI]). Each of the trigger sources is associated with a command buffer via the associated command select field (TCTRLa[TCMD]).

INPUTMUX provides signal routing options for internal peripherals. Some peripheral inputs are multiplexed to multiple input sources. The sources can be external pins, interrupts, output signals of other peripherals, or other internal signals.

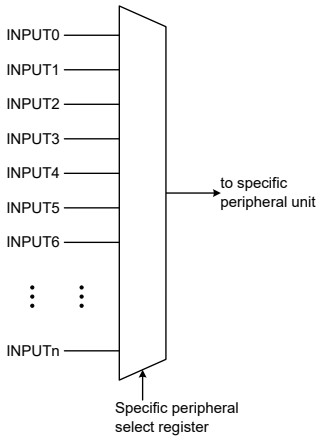

Figure 1. Generic input multiplexing block

The ADC has multiple options for both instances, ADC0 and ADC1, to trigger the analog conversions. The four ADC0 trigger input connections can be chosen from the list available in the “ADCn Trigger Input Connections ADCn_TRIG0-ADCn_TRIG3” section from the *LPC553x/LPC55S3x Reference Manual* (document LPC553xRM). This document focuses on using the CTimer as a trigger.

CTimer trigger options for ADC0 are as follows:

- 000101 - T0_MAT3
- 000110 - T1_MAT3
- 000111 - T2_MAT3
- 001000 - T3_MAT3
- 001001 - T4_MAT3

4 ADC CTimer example

For this example, the following three modules are set up:

- ADC
- CTimer
- INPUTMUX

This example is created with the SDK version 2.14.0 using MCUXpresso 11.8.0.

Although it is not necessary to assign an external pin for the CTimer, in this example, the PIO1_22 is used to verify the behavior of the CTimer. In addition, the ADC0 uses an external pin to measure an external analog source. A simple block diagram of the example is shown in Figure 2.

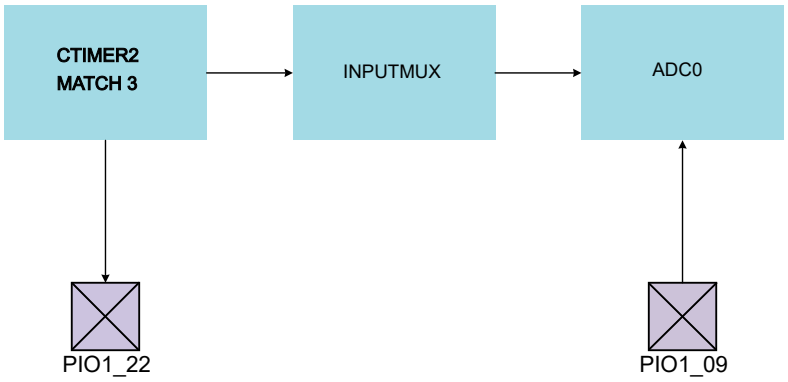

Figure 2. ADC CTimer example interconnections

The ADC is configured in the following manner:

- The ADC input frequency is 5.7 MHz connected from the PLL0 clock configured at 45.8 MHz.
  Note: *Use low frequencies to execute the calibration flow.*
- Select 128 ADC conversions, which are averaged to calculate each calibration value.
  Selecting a higher number of averages lead to more accurate conversions after completing calibration. ADC analog circuits are pre-enabled and ready to execute conversions without startup delays (at the cost of higher DC current consumption). The reference voltage is the voltage on the VDDA pin.

```
/*Attaching low frequency clock from PLL for calibration purposes: ADC0 Frequency at 5.7MHz*/
CLOCK_SetClkDiv(kCLOCK_DivAdc0Clk, 0U, true);            /*!< Reset ADC0CLKDIV divider counter and halt it */
CLOCK_SetClkDiv(kCLOCK_DivAdc0Clk, 8U, false);           /*!< Set ADC0CLKDIV divider to value 8 */
CLOCK_AttachClk(kPLL0_to_ADC0);                 /*!< Switch ADC0 to PLL0 */

/* Disable VREF power down */
POWER_DisablePD(kPDRUNCFG_PD_VREF);

LPADC_GetDefaultConfig(&mLpadcConfigStruct);
mLpadcConfigStruct.enableAnalogPreliminary = true;
mLpadcConfigStruct.referenceVoltageSource = kLPADC_ReferenceVoltageAlt3; /* VDDA */
mLpadcConfigStruct.conversionAverageMode = kLPADC_ConversionAverage128;
LPADC_Init(DEMO_LPADC_BASE, &mLpadcConfigStruct);
```

- Once the auto-calibration is finished, choose a higher frequency clock for the ADC. In this example, 48 MHz is used.
- The command configuration is set for channel 0 associated with the A-side using a high-resolution conversion. An analog signal ADC0IN0A is selected for conversion available on pin PIO1_9.
- Each ADC command independently makes a channel and conversion type selection. In this example, a single-ended operation is chosen. It is possible to do the conversion in differential mode but only limited pairs are available as differential channels. For the available pin pairings, refer to the *LPC553x/LPC55S3x Reference Manual* (document LPC553xRM).
- Also, to trigger the ADC conversions, the signal of another module is used. As a result, the trigger configuration is set to hardware trigger.

```
/* Set conversion CMD configuration. */
LPADC_GetDefaultConvCommandConfig(&mLpadcCommandConfigStruct);
mLpadcCommandConfigStruct.channelNumber = 0U;
mLpadcCommandConfigStruct.sampleChannelMode = kLPADC_SampleChannelSingleEndSideA;
mLpadcCommandConfigStruct.conversionResolutionMode = kLPADC_ConversionResolutionHigh;

LPADC_SetConvCommandConfig(DEMO_LPADC_BASE, DEMO_LPADC_USER_CMDID, &mLpadcCommandConfigStruct);

/* Set trigger configuration. */
LPADC_GetDefaultConvTriggerConfig(&mLpadcTriggerConfigStruct);
mLpadcTriggerConfigStruct.targetCommandId       = 1U;
mLpadcTriggerConfigStruct.enableHardwareTrigger = true;
LPADC_SetConvTriggerConfig(DEMO_LPADC_BASE, 0U, &mLpadcTriggerConfigStruct); /* Configure the trigger0. */
```

The CTimer is configured in the following manner:

- The CTimer input frequency is 96 MHz. Any CTimer0 to CTimer4 Match 3 signal can be configured from the available trigger options for the ADC0 instance. In this case CTimer 2 Match 3 is used and configured as a timer mode, which increments on every APB bus clock.
- The counter is reset after every match and toggles the output at a 1 kHz frequency.
- As the signal is attached to the ADC trigger, it is not necessary to enable the interrupt for the CTimer. However, if it is necessary to change the match value or other setting of the CTimer, enable an interrupt for the other required actions.

```
/* Use 96 MHz clock for Ctimer2 */
CLOCK_SetClkDiv(kCLOCK_DivCtimer2Clk, 0u, false);
CLOCK_SetClkDiv(kCLOCK_DivCtimer2Clk, 1u, true);
CLOCK_AttachClk(kFRO_HF_to_CTIMER2);

CTIMER_GetDefaultConfig(&config);

CTIMER_Init(CTIMER, &config);

/* Configuration 0 */
matchConfig0.enableCounterReset = true;
matchConfig0.enableCounterStop  = false;
matchConfig0.matchValue         = CTIMER_CLK_FREQ / 2000;
matchConfig0.outControl         = kCTIMER_Output_Toggle;
matchConfig0.outPinInitState    = false;
matchConfig0.enableInterrupt    = false;

CTIMER_SetupMatch(CTIMER, CTIMER_MAT3_OUT, &matchConfig0);
CTIMER_StartTimer(CTIMER);
```

The INPUTMUX is configured in the following manner:

- The INPUTMUX module is initialized prior to attaching the appropriate signals.
- In this case, route the ADC0 instance trigger input signal from the CTimer2 Match 3 signal using INPUTMUX.

```
INPUTMUX_Init(INPUTMUX);
INPUTMUX_AttachSignal(INPUTMUX, 0U, kINPUTMUX_Ctimer2M3ToAdc0Trigger);
```

It is possible to verify that the trigger signal is effectively working as expected, by checking the CTimer external pin PIO1_22 at header J10 pin 9 on the LPC55S36-EVK board. It is toggling at a rate of 1 kHz. As discussed previously, each ADC conversion is triggered on the rising edge of the CTimer waveform, as shown in Figure 3. The three blue arrows in this figure represent rising edges where the ADC0 is triggered.

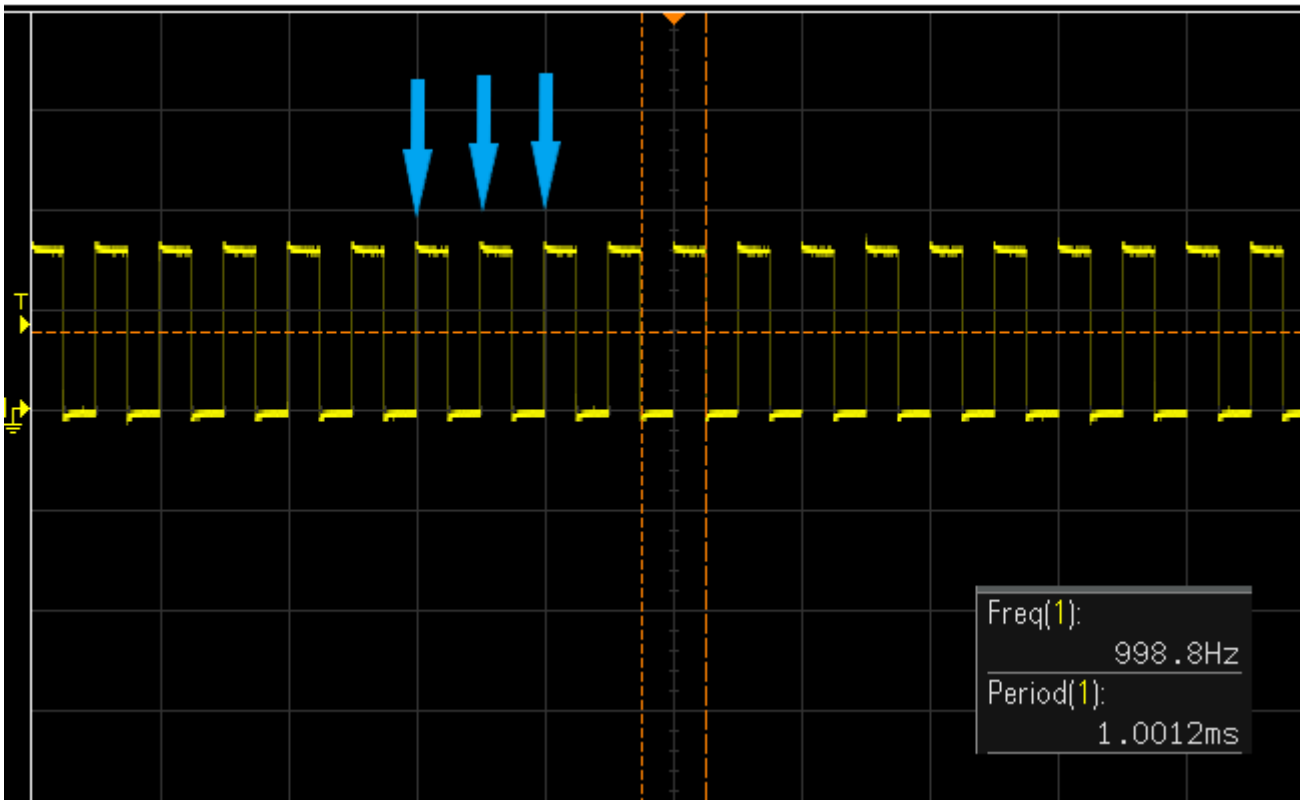

Figure 3. CTimer match rate shown through external pin

Figure 4 shows an example of the ADC result when measuring 3.3 V. We must expect the maximum value printed in the terminal window for the 16-bit resolution. In this instance, this value is "65535". To change the converted value, change the voltage on pin PIO1_9 at header J7 pin 1 on the LPC55S36-EVK board.

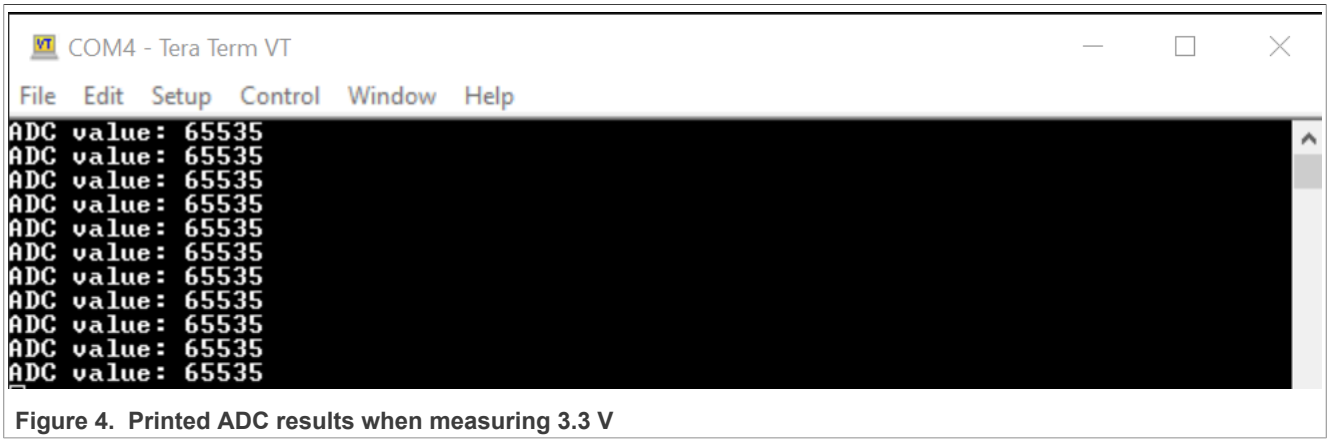

Figure 4. Printed ADC results when measuring 3.3 V

5 ADC calculation tool

The objective of the ADC calculation tool is to define maximum sampling rates achieved depending on the input signal impedance characteristics. To sample the input voltage accurately, the source resistance and ADC sample time must be chosen appropriately.

Equation (1) provides the required sample time for a fixed source resistance (R_{AS}):

$$min\ t_{SMP} = B \times \left[R_{AS} \times \left(C_{AS} + C_P + C_{ia}\right) + C_{ia} \times \left(R_{AS} + R_I\right)\right] \quad (1)$$

$$B = -ln\left(\frac{LSB_{ERR}}{2^N}\right) \quad (2)$$

Where:

- B implies the adjusted resolution that is based on the chosen sampling error.
- N implies ADC resolution, that is, 12 for 13-bit and 12-bit mode, 16 for 16-bit mode.
- LSB_{ERR} implies the value of acceptable sampling error in LSBs, that is, sampling within ¼ LSB accuracy.

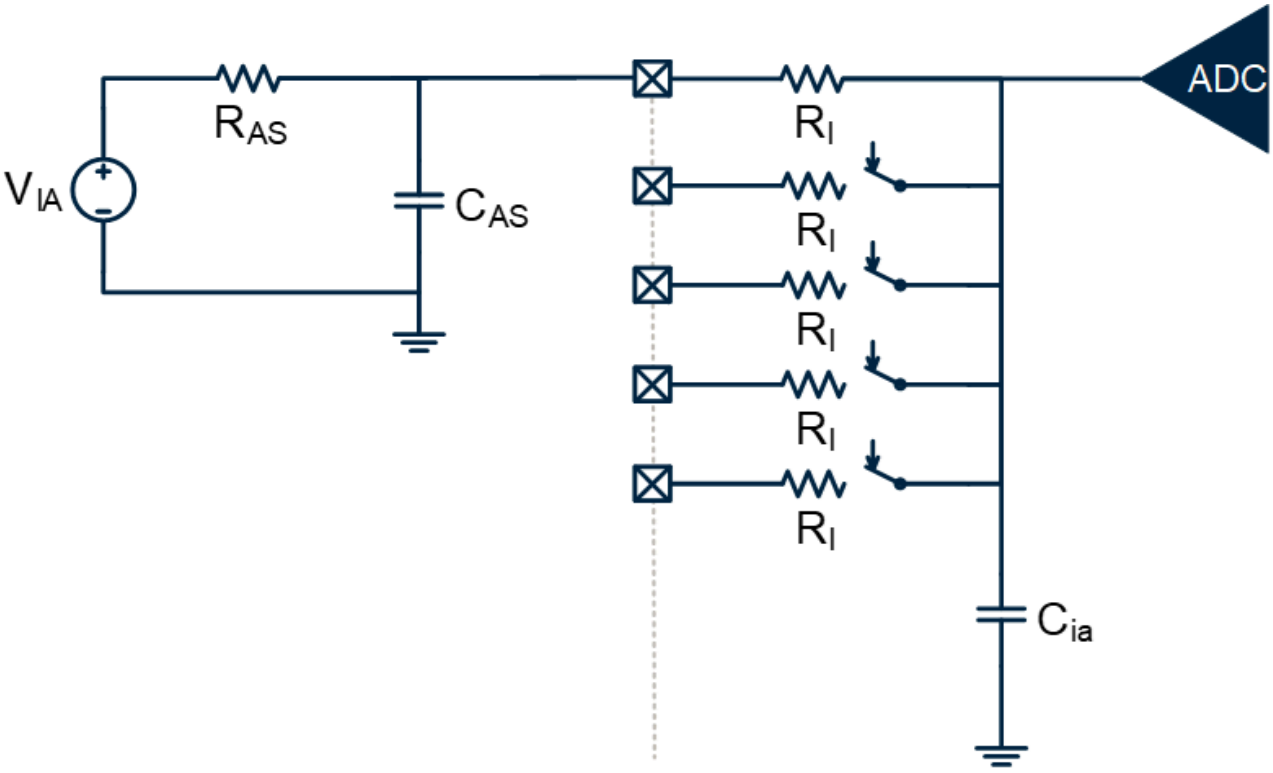

Figure 5. ADC simple circuitry diagram

The user-configured sample time is determined by the ADC input clock frequency (f_{ADCK}) and the sample time select (STS) bits in the ADC command register, which chooses the number of the sample cycles. When STS is programmed to a non-zero value, the sample time is ($3 + 2^{STS}$) ADCK cycles. The shortest sample time maximizes the conversion speed for lower impedance inputs. Extending sample time allows higher impedance inputs to be accurately sampled, see equation (3):

$$user\ t_{SMP} = \frac{user_{STS}}{f_{ADCK}} \quad (3)$$

Where:

- user $_{STS}$ implies the number of ADC clock cycles during the sample time, and is programmable to 3, 5, 7, 11, 19, 35, 67 or 131 ADCK cycles, ensuring that user $_{STS}$ ≥ min $_{STS}$. It depends on the value chosen at the register CMDHn[STS].
- user t_{SMP} must be configured to be greater than or equal to min t_{SMP}.

To find the maximum source resistance that allows sampling at the desired accuracy, set user t_{SMP} > min t_{SMP} and solve for R_{AS} in equation (4):

$$R_{AS} < \frac{\frac{user_{STS}}{f_{ADCK}{}^{xB}} - \left(R_1 x C_{ia}\right)}{C_{AS} + C_P + 2C_{ia}} \quad (4)$$

The first section of this tool is shown in Figure 6. The table in this figure specifies the required sample time for a fixed R_{AS}. The user can input the source resistance, source capacitance (both external components), the resolution, and the value of the acceptable sampling error in LSB (LSB_{ERR}) of the ADC.

The values provided in the LPC553x/LPC55S3x data sheet are as follows:

- C_P: Parasitic cap of pad/package
- C_{ia}: Input capacitance
- R_I: Input resistance

ADC min Sample Time t_{SMP}

min sample time based on a fixed R_{AS}

Input... enter yellow highlighted cells		
Resolution	16	bits
C_{AS}	100E-12	F
R_{AS}	5.0E+3	Ω
LSB_{ERR}	1/8	LSB
C_P	3E-12	F
Cia	4E-12	F
B	13.17	

	$R_I(\Omega)$	min t_{SMP}
High Speed		
V_{REFP} = 1.8V	1.6E+3	7.39E-6
V_{REFP} = 3.0V	1.1E+3	7.37E-6
Standard Muxed		
V_{REFP} = 1.8V	3.2E+3	7.48E-6
V_{REFP} = 3.0V	1.8E+3	7.40E-6
Standard Dedicated		
V_{REFP} = 1.8V	300	7.33E-6
V_{REFP} = 3.0V	300	7.33E-6
Internal		
V_{REFP} = 1.8V	1.9E+3	7.41E-6
V_{REFP} = 3.0V	1.1E+3	7.37E-6

Figure 6. ADC sample time calculator

The second section of the tool can provide the maximum source resistance based on the sample time and ADC frequency used. In the first section, the user can change the yellow highlighted cells according to the parameters that they are working with.

The additional parameters in the calculator tool are as follows:

- f_{ADCK}: Input clock frequency
- CYC_{SMP_MIN}: Minimum sample cycles required for $T_{SMP} > T_{SMP\_REQ}$
- CYC_{SMP_USER}: Sample cycles set by the user using CMDHn[STS]
- T_{SMP}: Sample time set by the user

ADC Conversion Time CT and R_{AS} Calculator

To calculate maximum R_{AS} with given sample time and ADC input clock frequency

Input... enter yellow highlighted cells

Resolution	12	bits	Sample Time Select	131	ADCK
C_{AS}	1E-9	F	Hardware Average	1	
ADC Source CLK	100E+6	Hz	Power Select	high power	
ADCnCLKDIV[DIV]	3				
LSB_{ERR}	1/2	LSB			
C_P	3E-12	F			
C_{ia}	4E-12	F			
B	9.01				
f_{ADCK}	33E+6	MHz	max f_{ADCK}	60.0E+6	MHz

	R_I (Ω)	min STS	user STS	user t_{SMP}	max R_{AS}	CT	Msps
High Speed							
V_{REFP} = 1.8V	1.6E+3	5	131	3.9E-6	426.7E+0	4.4E-6	0.23
V_{REFP} = 3.0V	1.1E+3	5	131	3.9E-6	428.7E+0	4.4E-6	0.23
Standard Muxed							
V_{REFP} = 1.8V	3.2E+3	5	131	3.9E-6	420.4E+0	4.4E-6	0.23
V_{REFP} = 3.0V	1.8E+3	5	131	3.9E-6	425.9E+0	4.4E-6	0.23
Standard							
V_{REFP} = 1.8V	300	5	131	3.9E-6	431.9E+0	4.4E-6	0.23
V_{REFP} = 3.0V	300	5	131	3.9E-6	431.9E+0	4.4E-6	0.23
Internal							
V_{REFP} = 1.8V	1.9E+3	5	131	3.9E-6	425.6E+0	4.4E-6	0.23
V_{REFP} = 3.0V	1.1E+3	5	131	3.9E-6	428.7E+0	4.4E-6	0.23

Figure 7. ADC sampling frequency and R_{AS} calculator

6 ADC basic concepts

This section lists the basic concepts of ADC as follows:

- **Resolution**: The number of bits in the ADC digital output representing an analog input signal. For LPC553x/LPC55S3x, the resolution can be configured to 12, 13, and 16-bit resolution.
- **Reference Voltage**: The ADC requires a reference voltage to create a successive approximation comparison with the analog input to produce a digital output. The digital output is the ratio of the analog input to this reference voltage.

$$V_{REF} = V_{REFH} - V_{REFL} \tag{5}$$

  Where:
  - V_{REFH} implies a high reference voltage.
  - V_{REFL} implies a low reference voltage.
- **ADC output formula**: The conversion equation of the ADC is used to calculate the digital output corresponding to a particular analog input voltage. This equation assumes an ideal analog-to-digital conversion with no errors as follows:

$$ADC\ \ Digital\ \ Output = \frac{2^N \times Analog\ \ Input\ \ Voltage}{Reference\ \ Voltage} \tag{6}$$

  Where:
  - N implies the ADC resolution. For the LPC553x, this value is 12/13/16.
- **Least significant bits (LSB):** A unit of voltage equal to the smallest resolution of the ADC, that is, the smallest incremental voltage causing a change in the digital output. The LSB is equal to the reference voltage divided by the maximum count of the ADC shown in equation (7):

$$LSB = \frac{V_{REF}}{2^N} \tag{7}$$

Where:

– N implies the ADC resolution.

– V_{REF} implies the analog reference voltage.

- **ADC actual transfer function**: The ADC converts an input voltage to a corresponding digital code. The curve describing this behavior is the actual transfer function and includes all the errors inherent to the ADC module itself.
- **ADC ideal transfer function**: The ideal transfer function represents the behavior of the ADC. The assumption is that the ADC is perfectly linear or a given change in input voltage changes the conversion code irrespective of the initial level of the input.

7 Sources of error in ADC measurements

This section presents some typical factors that prevent the ADC from performing the accurate analog-to-digital measurements.

Reference voltage noise: The ADC output is directly proportional to the analog input voltage and the reference voltage. An unstable reference voltage (for example, caused by noise in the supply rail) causes the changes in the converted digital outputs. For example, a reference voltage of 5 V and an input voltage of 1 V gives 819 for a 12-bit resolution using the ADC output formula. With a 50 mV increase in the absolute reference voltage (that is, V_{REF} = 5.05 V), the new converted value for the same 1 V input voltage is now 811. The resulting reference voltage noise error is 811-819 = -8 LSB.

Analog input signal noise: Small but high-frequency variations in the analog input signal can potentially cause big conversion errors during ADC sampling time. Electromagnetic emissions induce the noise from surrounding electrical devices (EMI noise) and therefore, the conversion accuracy is negatively impacted. The least significant bits are constantly changing due to the signal variations. Therefore, if the noise present in the input signal is higher than 1 LSB, the number of reliable bits reduces effectively in the conversion result.

Analog-signal source resistance: The impedance of the analog signal source or series resistance (RIN) between the source and the input pin causes a voltage drop across it because of the current flowing into the pin.

Temperature influence: The temperature of the system can have a major influence on ADC accuracy, mainly causing offset error drift and gain error drift. The ADC reference voltage also changes with temperature change. These errors can be compensated with adjustments to the microcontroller firmware as follows:

- Monitoring the internal band gap voltage to verify that the reference voltage has not changed.
- Characterizing the system over the temperature range of the application to account for the errors.

8 References

Table 2 lists the resources that can be referred for more information.

Table 2. References

Resource	Link/how to access
LPC553x Reference Manual	LPC553xRM
LPC553x product data sheet	LPC553x
Errata sheet LPC553x	LPC553x_ES
Hardware Design Guidelines for LPC55(S)xx Microcontrollers	AN13033

9 Note about the source code in the document

Example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2023 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials must be provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE

10 Revision history

Table 3 summarizes the revisions done to this document.

Table 3. Revision history

Revision number	Release date	Description
4	20 November 2023	Attached the .xls to the document
3	6 October 2023	• Updated images to codeblocks for Section 4 • Added Section 9 • Used conditioning for the equations
2	22 September 2023	• Updated Identifier, abstract, and keywords in metadata • Updated the AN to the latest style sheet • Updated SDK and IDE versions • Restructured the entire document • Added legal information • Updated web links for Section 8 • Updated Figure 1 and Figure 2 to SVG
1	20 April 2022	Updated Section 5
0	2 February 2022	Initial public release

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

EdgeVerse — is a trademark of NXP B.V.

Contents

1 Introduction 2
2 ADC features 2
3 ADC trigger interconnections 2
4 ADC CTimer example 3
5 ADC calculation tool 6
6 ADC basic concepts 9
7 Sources of error in ADC measurements 10
8 References 10
9 Note about the source code in the document 11
10 Revision history 11
Legal information 12

For more information, please visit: https://www.nxp.com

Date of release: 20 November 2023
Document identifier: AN13523