AN13209

RT500 ADC Temperature Sensor Usage

Rev. 0 — 05/2021 Application Note

Contents

1 Introduction

This application note provides an overview of how to use RT500 SDK software with EVK hardware to validate the RT500 built-in temperature sensor using the ADC controller. The ADC input channel 7 is mapped to an internal temperature sensor.

ADC watermark interrupt would be asserted once the number of data words stored in the ADC Result FIFO are greater than the watermark value. In ADC ISR, the watermark flag is cleared by reading the conversion result value. When the conversion done, two valid results are stored in the FIFO, and the temperature can be calculated using a specific formula within two results.

2 Temperature sensor overview

The temperature sensor transducer uses an intrinsic pn-junction diode reference and outputs a CTAT voltage (Complement To Absolute Temperature). The output voltage varies inversely with device temperature with an absolute accuracy of better than ±5 °C over the full temperature range (-40 °C to +105 °C). The temperature sensor is only approximately linear with a slight curvature. The output voltage is measured over different ranges of temperatures and fit with linear-least-square lines.

2.1 Characteristics

The following table shows the temperature sensor static and dynamic characteristics.

(VDDA_BIAS = 3.3 V, All other supplies = 1.8 V)

Symbol	Parameter	Conditions	Notes	Min	Typ	Max	Unit
DT_{sen}	sensor temperature accuracy	T_{amb} = -20 °C to 70 °C	1	-	-	2.77	°C
E_L	linearity error	Tamb = -20 °C to 70 °C		-	-	2.79	°C

1. Absolute temperature accuracy. Based on characterization. Not tested in production

2.2 Connection between ADC channel input and analog output

The following figure shows the connection between ADC channel input and analog output.

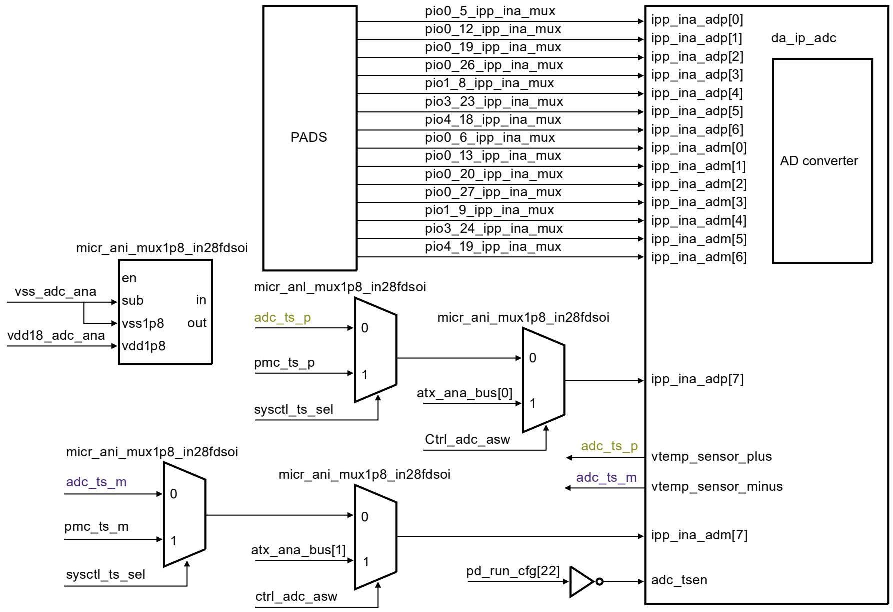


Figure 1. ADC channel input and analog output

2.3 Registers needed for ADC temperature sensor

2.3.1 Temperature sensor control (SYSCTL0_TEMPSENSORCTL)

This register enables the on-chip temperature sensor to be measured by the ADC.

Field	Description
31-1 —	Reserved
Field	**Description**
0 TSSRC	Temperature Sensor Source 0 - ADC Built-in Temperature Sensor 1 - PMC Temperature Sensor

2.3.2 Run configuration register 1 clear (SYSCTL0_PDRUNCFG1_CLR)

Writing a 1 to a bit position in this register clears the corresponding position in PDRUNCFG1. This is a write-only register.

22 ADC_LP	ADC low power mode To minimize power consumption in deep_sleep or deep_powerdown mode both ADC_LP and ADC_PD bits must be set. 0 - No effect 1 - Clears the PDRUNCFG0 Bit
21 ADC_PD	ADC analog functions To minimize power consumption in deep_sleep or deep_powerdown mode both ADC_LP and ADC_PD bits must be set. 0 - No effect 1 - Clears the PDRUNCFG0 Bit

2.4 Using typical parameter provided by the data sheet

Use the typical parameters provided by the RT500 data sheet to perform a temperature reading. An approximate transfer function describes the temperature sensor.

Temp = 25 - ((V_{temp} -V_{temp25})/m)

Where:

- V_{temp} is the voltage of the temperature sensor channel at the ambient temperature.
- V_{temp25} is the voltage of the temperature sensor channel at 25ºC.
- m is the temperature sensor slope.

In application code, the user executes a conversion on the temperature sensor channel and converts the conversion result to a voltage (VTEMP) based on the voltage reference used. The ambient temperature is then calculated using the above equation. VTEMP25 and the temperature sensor slope values are specified constant values from the ADC Electrical information in the device data sheet.

When converting on the temperature sensor channel, the following command selections must be configured:

- Differential mode (CMDLn[DIFF] = 0x1)
- Maximum averaging (CMDHn[AVGS] = 0x7)
- Maximum sample time (CMDHn[STS] = 0x7)

3 Demo application

3.1 Environment

3.1.1 Hardware environment

- Board
  - MIMXRT595EVK
- Debugger
  - Integrated CMSIS-DAP debugger on the board
- Miscellaneous
  - 1 Micro USB cable
  - PC

- Board Setup
  - No special settings are needed

3.1.2 Software environment

- Tool chain
  - IAR embedded workbench 8.50.1 or MCUXpresso IDE v11.3.0 or Keil 5.29
- Software package
  - SDK_2.9.0_EVK-MIMXRT595

3.2 Project overview

1. Follow the Getting Started with MCUXpresso SDK for MIMXRT500 (available inside SDK➜docs) to go through the steps for opening lpadc_temperature_measurement project (SDK\boards\evkmimxrt685\driver_examples\lpadc\temperature_measurement).
2. Open the file lpadc_temperature_measurement.c (lpadc_temperature_measurement\source) and find the values for the temperature slope and intercept that helps to calculate the ambient temperature based on the formula that is given in Using typical parameter provided by the data sheet. These are specified constant values from the ADC Electricals information in the device data sheet.

```
#define DEMO_LPADC_TEMPERATURE_SLOPE                                  \
    -1.5738f /* Temperature sensor slope with the unit as mV/Celsius. \
             Please refer to the Data Sheet for details. */
#define DEMO_LPADC_TEMPERATURE_INTERCEPT                                              \
809.55f /* The voltage of the temperature sensor channel at 0 Celsius with the unit as mV.
\
             Please refer to the Data Sheet for details. */
```

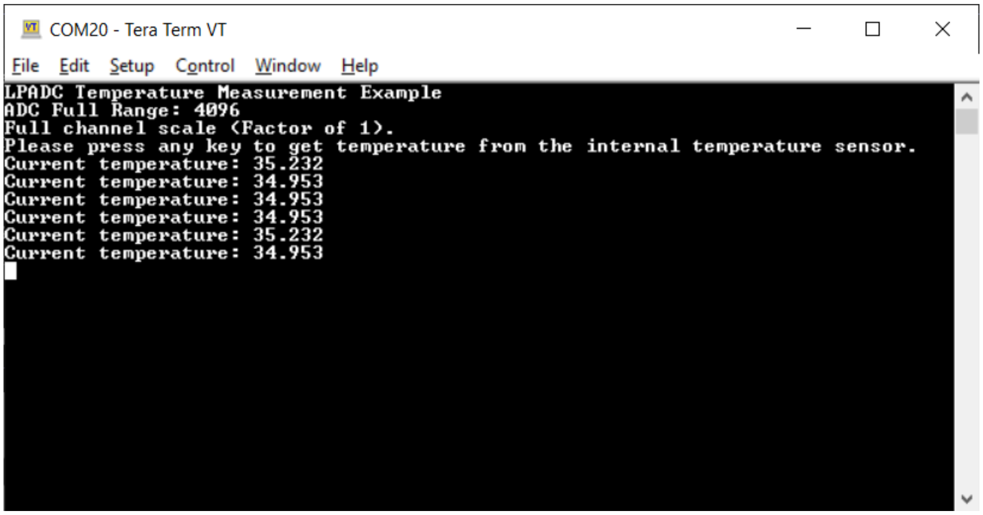
3. In SYSCTL>TEMPSENSORCTL, select between the two temperatures sensors available; ADC and PMC, the ADC is selected by default and no changes are required to use it. Note that the sensor is already enabled in the PDRUN configuration at the beginning of the main function.

```
/* Power up the ADC temperature sensor via SYSCTL. */
SYSCTL0->PDRUNCFG0_CLR = SYSCTL0_PDRUNCFG0_ADC_TEMPSNS_PD_MASK;
```

4. The STS bits of CMDH1 register are set to 7. So, the sample time is 131 (3 + 2STS) ADCK cycles since a sample time of over 35 is required. A long sample time allows higher impedance inputs to be accurately sampled. This is done by modifying the `smapleTimeMode` which is member of conversion commands configuration structure inside `LPADC_GetDefaultConvCommandConfig` function. This change is already done in the ADC configuration part of code.

```
g_LpadcCommandConfigStruct.sampleTimeMode = kLPADC_SampleTimeADCK131;
```

5. Follow the Getting Started with MCUXpresso SDK for MIMXRT500 (available inside SDK➜docs) to go through the steps for building and running lpadc_temperature_measurement demo.
6. When running the demo, type any key in the debug console which triggers the conversion.
7. Result:



4 Conclusion

The example shows how the SDK software with EVK hardware can be used to validate RT500 built-in temperature sensor using the ADC controller. It also shows, how the built-in temperature sensor can be used to indicate temperature value based on ADC output value.

5 References

1. RT500 Reference Manual
2. RT500 Data Sheet
3. MCUXpresso SDK Release Notes for EVK-MIMXRT595 (available inside SDK)
4. Getting Started with MCUXpresso SDK for EVK-MIMXRT595 (available inside SDK)

6 Revision history

Table 1. Revision history

Revision number	Date	Substantive changes
0	05/2021	Initial release

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

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

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 05/2021

Document identifier: AN13209