PSoC 4 Die Temperature (DieTemp)

1.0



Features

- Typical measured accuracy of +/- 1 ºC
- Valid operating range from –40 °C to +85 °C

General Description

The Die Temperature component provides access to the analog signal whose voltage represents the temperature of the die. It also provides the API needed to convert the digital voltage representation of that analog signal to a temperature. This component does not provide the mechanism to convert the analog voltage to a digital value. That must be done in the design by connecting the signal to the ADC in the device and sampling the voltage.

When to Use DieTemp

This component can be used to get a rough temperature measurement of the device. This value will also be correlated with the temperature within the enclosure the device is in.

Input/Output Connections

temp – Analog

This analog output is the connection of the die temperature to the SAR ADC for conversion.

Component Parameters

There are no parameters or user interface available for this component.

Placement

The DieTemp component uses the die temperature sensor, which is a part of SAR block.

Application Programming Interface

Application Programming Interface (API) routines allow you to configure the component using software. The following table lists and describes the interface to each function. The subsequent sections cover each function in more detail.

By default, PSoC Creator assigns the instance name "DieTemp_1" to the first instance of a component in a given design. You can rename it to any unique value that follows the syntactic rules for identifiers. The instance name becomes the prefix of every global function name, variable, and constant symbol. For readability, the instance name used in the following table is "DieTemp".

int32 DieTemp_CountsTo_Celsius(int32 adcCounts)

Description: Converts the ADC output to degrees Celsius.

Parameters: (int32) adcCounts

Return Value: Die Temperature in degrees Celsius.

Sample Firmware Source Code

PSoC Creator provides numerous example projects that include schematics and example code in the Find Example Project dialog. For component-specific examples, open the dialog from the Component Catalog or an instance of the component in a schematic. For general examples, open the dialog from the Start Page or **File** menu. As needed, use the **Filter Options** in the dialog to narrow the list of projects available to select.

Refer to the "Find Example Project" topic in the PSoC Creator Help for more information.

MISRA Compliance

This section describes the MISRA-C:2004 compliance and deviations for the component. There are two types of deviations defined:

- project deviations – deviations that are applicable for all PSoC Creator components
- specific deviations – deviations that are applicable only for this component

This section provides information on component-specific deviations. Project deviations are described in the MISRA Compliance section of the *System Reference Guide* along with information on the MISRA compliance verification environment.

The Die Temperature component does not have any specific deviations.

API Memory Usage

The component memory usage varies significantly, depending on the compiler, device, number of APIs used and component configuration. The following table provides the memory usage for all APIs available in the given component configuration.

The measurements have been done with the associated compiler configured in Release mode with optimization set for Size. For a specific design, the map file generated by the compiler can be analyzed to determine the memory usage.

Configuration	PSoC 4 (GCC)	
	Flash Bytes	SRAM Bytes
Default	80	0

Functional Description

For all supported PSoC4 families except PSoC Analog Coprocessor, the DieTemp component is always used in conjunction with the ADC SAR Sequencer to sample the analog voltage and produce a digital value. Since the temperature changes slowly, this signal is commonly used with the injection channel to sample the value infrequently compared to other signals being converted by the ADC.

Recommended ADC SAR Sequencer settings for a die temperature measurement:

Note This configuration assumes that channel 0 is used by the application and must be connected to a driver.

Figure 1. Component Connection

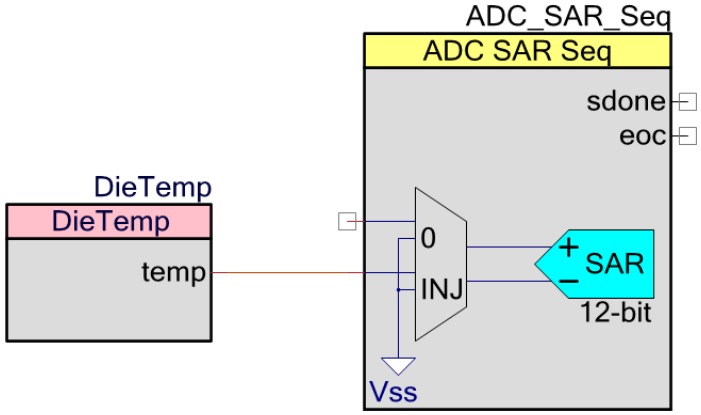


Figure 2 and Figure 3 show recommended ADC SAR Sequencer Settings.

Figure 2. ADC SAR Sequencer Settings – General Tab

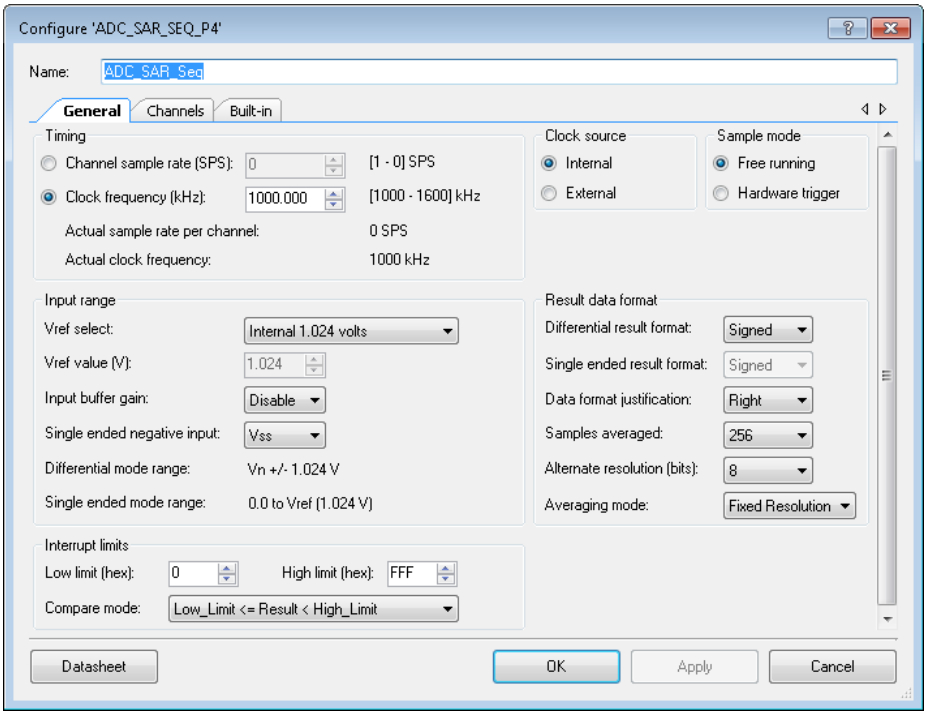
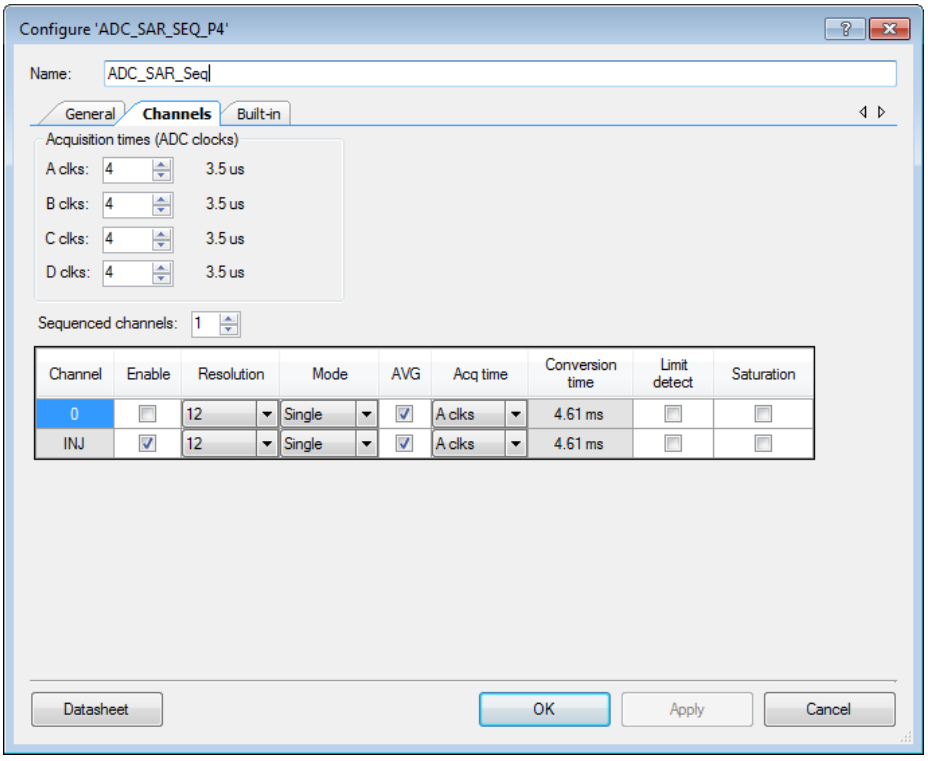


Figure 3. ADC SAR Sequencer Settings – Channels Tab



Key parameters of ADC SAR Sequencer include:

- Channel Mode [1] – Single
- Resolution [1] – 12 bit
- Data format justification – Right
- Reference – Internal, 1.024 volts
- Sample rate – <= 100 ksps
- Samples averaged – value >16 may be used to reduce noise
- Averaging mode – Fixed Resolution
- Single ended negative input – Vss.

Not following the proposed settings may lead to unexpected behavior of the component or poor accuracy of measurements. However, the value of Vref can be different from 1024 mV. If the value of Vref in the ADC SAR Sequencer settings is different from the proposed, the value of adcCounts that is passed to DieTemp_CountsTo_Celsius API should be adjusted:

$$ADCCountsCorrected = (Ref_{act}/1.024) * ADCCounts$$

Where:

- *ADCCountsCorrected* is the corrected number of ADC counts, which should be used as a parameter for the DieTemp_CountsTo_Celsius API.
- *ADCCounts* is the actual measurement result from ADC SAR.
- Ref_{act} is the actual voltage reference in volts.

Using the Injection channel is recommended because conversion can only be triggered with a firmware trigger and it is not a part of a regular scan. If the component is used with other channels, it is recommended to configure acquisition time to be at least 5 µs, otherwise temperature values will be unreliable due to temperature sensor settling time.

Triggering the Injection channel is recommended no sooner than one second after the previous trigger has completed.

The component is only intended for operation with the on-chip ADC SAR and should not be connected to any other analog resource or routed off the chip.

---

1 For Injection channel (or the one to which the component is connected.)

For the PSoC Analog Coprocessor family, the DieTemp component is used in conjunction with the Scan_ADC component (named ADC) shown in Figure 4.

Figure 4. Component Connection

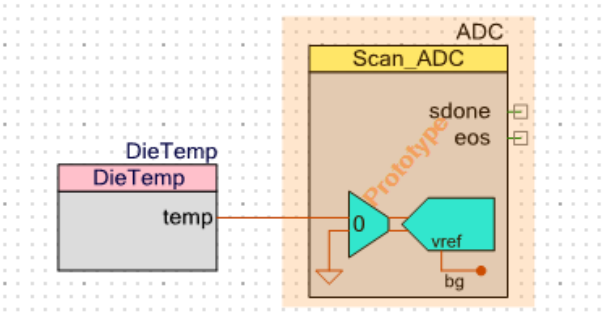
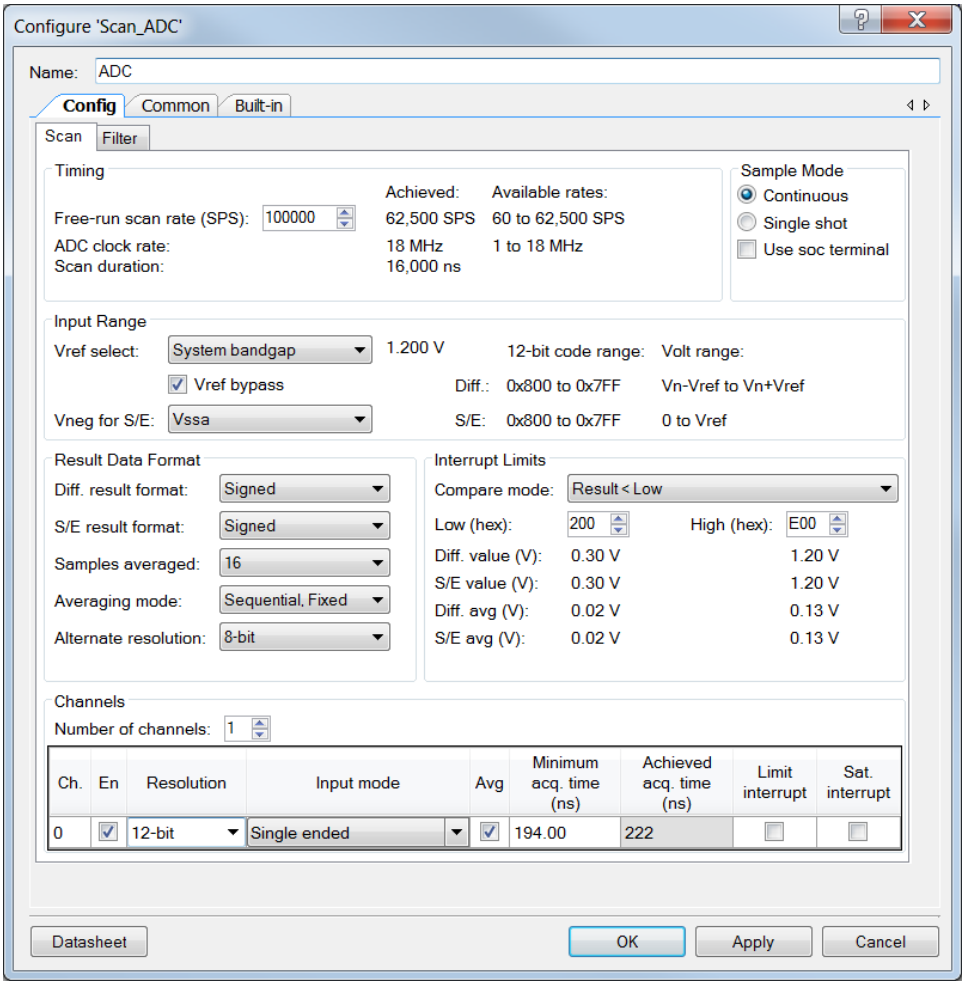


Figure 5 shows the recommended settings for a die temperature measurement.

Figure 5. ADC Settings – Config Tab



Key parameters of the ADC include:

- VPOS connected to the temp sensor diode
- VNEG connected to VSSA
- 12-bit output from the Results register
- Vref select: System bandgap
- Averaging is enabled (checkbox is checked) for appropriate channel
- Samples averaged ≥ 16
- Averaging mode: Sequential, Fixed

Not following the proposed settings may lead to unexpected behavior of the component or poor accuracy of measurements.

Resources

Configuration	Resource Type
	Die tempersture sensor of the SAR block
Default	1

DC and AC Electrical Characteristics

Parameter	Description	Min	Typ	Max	Units	Details/Conditions
$T_{SENSACC}$	Temperature sensor accuracy	–5	±1	+5	°C	–40 to +85 °C

Component Changes

This section lists the major changes in the component from the previous version.

Version	Description of Changes	Reason for Changes / Impact
1.0.g	Edited datasheet.	Added guidance for how to use the ADC with the PSoC Analog Coprocessor family. Removed characterization note because it doesn't apply to this component.
1.0.f	Edited datasheet.	Final characterization data for PSoC 4000S, PSoC 4100S and PSoC Analog Coprocessor devices is not available at this time. Once the data is available, the component datasheet will be updated on the Cypress web site.
1.0.e	Updated the Component Catalog visibility expression to support PSoC 4100S and PSoC Analog Coprocessor devices.	Added new device support.
1.0.d	Updated datasheet.	
1.0.c	Updated datasheet.	Updated the MISRA Compliance section. Clarified the Features section. Clarified the Functional Description section.
1.0.b	Updated the Functional Description section of the datasheet.	To clarify ADC SAR Sequencer required settings and timing information.
1.0.a	Updated datasheet.	Updated MISRA Compliance section.
1.0	First release	

© Cypress Semiconductor Corporation, 2013-2016. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.