AVR Internal Temperature Sensor

Some AVR devices have an internal temperature sensor. It can be used to measure the core temperature of the device (not the ambient temperature around the device). The measured voltage has a linear relationship to the temperature. The voltage sensitivity is approximately 1 mV/°C, the accuracy of the temperature measurement is ±10°C.

The temperature measurement is based on the on-chip temperature sensor that is coupled to a single ended ADC channel. Selecting ADC channel 8 by writing '1000' to ADMUX.MUX[3:0] enables the temperature sensor. The internal 1.1 V voltage reference must also be selected for the ADC voltage reference source. When the temperature sensor is enabled, the ADC converter can be used in single conversion mode to measure the voltage over the temperature sensor.

Sample Measurement Data

Temperature	-45°C	+25°C	+85°C
Voltage	242mV	314mV	380mV

(/local--files/8avr:avrtemp/temp1.png)

Calibration

The results from temperature measurements have offset and gain errors. The internal temperature reference can be corrected for these errors by making calibration measurements at one or two known temperatures and adjusting the output values. This can result in very precise temperature measurements, sometimes as accurate as ±2°C.
More detail can be found in this application note. (http://www.atmel.com/Images/Atmel-8108-Calibration-of-the-AVR's-Internal-Temperature-Reference_ApplicationNote_AVR122.pdf)

Configuring the ADC

The internal 1.1 V voltage reference must be selected for the ADC voltage reference source when using the internal temperature sensor. Writing “11” to the **REFS1** and **REFS0** bits of the **ADMUX register** selects the internal 1.1 V Voltage Reference.

The ADC has multiple input channels and modes of operation. The **Single Conversion** mode can be used to convert the temperature sensor signal connected to channel 8. To select channel 8, writing “1000” to the MUX3 thru MUX0 bits selects channel 8 or the temperature sensor.

Once the conversion is complete, the result is stored in two 8-bit ADC data registers ADCH (higher 8-bits) and ADCL (lower 8-bits). The 10-bit result can be either left justified or right justified. If ADLAR bit is set to a "1", then the result is left adjusted to the upper 10-bits of the two registers. If set to "0" then the result occupies the lower 10 bits of the two registers. By default, each bit is cleared and the word is right justified.

This code statement will set the bits as described.
ADMUX = (1«REFS1) | (1«REFS0) | (0«ADLR) | (1«MUX3) | (0«MUX2) | (0«MUX1) | (0«MUX0);

Name: ADMUX
Offset: 0x7C
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	REFS1	REFS0	ADLAR		MUX3	MUX2	MUX1	MUX0
Access	R/W	R/W	R/W		R/W	R/W	R/W	R/W
Reset	0	0	0		0	0	0	0

Bits 7:6 – REFSn: Reference Selection [n = 1:0]
These bits select the voltage reference for the ADC. If these bits are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSRA is set). The internal voltage reference options may not be used if an external reference voltage is being applied to the AREF pin.

Table 29-3 ADC Voltage Reference Selection

REFS[1:0]	Voltage Reference Selection
00	AREF, Internal V_{ref} turned off
01	AV_{CC} with external capacitor at AREF pin
10	Reserved
11	Internal 1.1V Voltage Reference with external capacitor at AREF pin

(/local--files/8avr:avrtemp/temp2.png)

Bits 3:0 – MUXn: Analog Channel Selection [n = 3:0]
The value of these bits selects which analog inputs are connected to the ADC. If these bits are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSRA on page 323 is set).

Table 29-4 Input Channel Selection

MUX[3:0]	Single Ended Input
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5

(/local--files/8avr:avrtemp/temp3.png)

MUX[3:0]	Single Ended Input
0110	ADC6
0111	ADC7
1000	Temperature sensor
1001	Reserved
1010	Reserved
1011	Reserved
1100	Reserved
1101	Reserved
1110	1.1V (V_{BG})
1111	0V (GND)

(/local--files/8avr:avrtemp/temp4.png)

Configuring the ADC Clock and Conversion Timing

The ADC can prescale the system clock to provide an ADC clock that is between 50 kHz and 200 kHz to get maximum resolution. If an ADC resolution of less than 10-bits is required, then the ADC clock frequency can be higher than 200 kHz. At 1 MHz it is possible to achieve up to eight bits of resolution.

The prescaler value is selected with **ADPS bits** in **ADCSRA Register**. For example; writing “110” to the **ADCSRA register** selects the divide by 64 pre-scaler resulting in a 125 KHz ADC clock when an 8 MHz oscillator clock is used.

ADC Control and Status Register A

Name: ADCSRA
Offset: 0x7A
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

(/local--files/8avr:avrtemp/temp5.png)

Bit 7 – ADEN: ADC Enable
Writing this bit to one enables the ADC. By writing it to zero, the ADC is turned off. Turning the ADC off while a conversion is in progress, will terminate this conversion.

Bit 6 – ADSC: ADC Start Conversion
In Single Conversion mode, write this bit to one to start each conversion. In Free Running mode, write this bit to one to start the first conversion. The first conversion after ADSC has been written after the ADC has been enabled, or if ADSC is written at the same time as the ADC is enabled, will take 25 ADC clock cycles instead of the normal 13. This first conversion performs initialization of the ADC.

ADSC will read as one as long as a conversion is in progress. When the conversion is complete, it returns to zero. Writing zero to this bit has no effect.

Bit 5 – ADATE: ADC Auto Trigger Enable
When this bit is written to one, Auto Triggering of the ADC is enabled. The ADC will start a conversion on a positive edge of the selected trigger signal. The trigger source is selected by setting the ADC Trigger Select bits, ADTS in ADCSRB.

Bit 4 – ADIF: ADC Interrupt Flag
This bit is set when an ADC conversion completes and the Data Registers are updated. The ADC Conversion Complete Interrupt is executed if the ADIE bit and the I-bit in SREG are set. ADIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ADIF is cleared by writing a logical one to the flag. Beware that if doing a Read-Modify-Write on ADCSRA, a pending interrupt can be disabled. This also applies if the SBI and CBI instructions are used.

Bit 3 – ADIE: ADC Interrupt Enable
When this bit is written to one and the I-bit in SREG is set, the ADC Conversion Complete Interrupt is activated.

Bits 2:0 – ADPSn: ADC Prescaler Select [n = 2:0]
These bits determine the division factor between the system clock frequency and the input clock to the ADC.

(/local--files/8avr:avrtemp/temp6.png)

Table 29-5 Input Channel Selection

ADPS[2:0]	Division Factor
000	2
001	2

(/local--files/8avr:avrtemp/temp7.png)

ADPS[2:0]	Division Factor
010	4
011	8
100	16
101	32
110	64
111	128

(/local--files/8avr:avrtemp/temp8.png)

Starting a Conversion

In single conversion mode the **ADSC bit** in the **ADCSRA register** must be set to a logical one state to start the ADC conversion. This bit remains at logic high while the conversion is in progress and is cleared by the hardware, once the conversion is complete.

The first conversion after the ADC is switched on takes 25 ADC clock cycles in order to initialize the analog circuitry. Then, for further conversions, it takes 13 ADC clock cycles (13.5 for Auto-triggered conversions).

Sample Project

A sample project for using the Temperature Sensor is available here. (/8avr:avradc)