
Atmel AVR291: 8MHz Internal Oscillator Calibration for USB Low Speed on Atmel ATmega32U4RC



**8-bit Atmel
Microcontrollers**

Application Note

Features

- Atmel® ATmega32U4RC MCU
- 8MHz internal RC oscillator
- Temperature range: -20°C to +85°C
- USB low speed
- Adjustable RC frequency with $\pm 1\%$ accuracy
- Tune RC oscillator at 3V and 5V
- Tune RC oscillator to any temperature
- Reduce BOM cost

1 Introduction

This application note describes a fast and accurate method for calibrating the internal RC oscillator on the Atmel ATmega32U4RC MCU. It offers simple and flexible calibration firmware source code, which allows device runtime calibration to for maintaining USB low-speed data transfers within the USB specification (1.5MHz $\pm 1.5\%$).

Please note that this application note is not applicable to all ATmega32U4 family devices. The part numbers supported by this application note are as follows:

- ATmega32U4RC-AU
- ATmega32U4RC-MU

Rev. 8384A-AVR-05/11



2 Theory of operation – the internal RC oscillator

In production, the Atmel ATmega32U4RC internal RC oscillator is calibrated at 5V (at one point, 25°C) and at 3V (at two points, 25°C and 85°C). Refer to the datasheet for further information about the factory setting of this device.

The accuracy of the factory calibration is within $\pm 1\%$. It is guaranteed for a temperature range between 0°C and 40°C. The calibration parameters are saved within the signature row. At reset, manufacturing calibration parameter 3v_25° is loaded into the OSCCAL register, meaning that the device is calibrated to operate at 3V and 25°C by default.

Since the device is only calibrated to operate between 0°C and 40°C, this application note provides a way to extended the range of operation (-20°C to 85°C). The method used is to update the OSCCAL register, depending on the voltage and the temperature, to recalibrate the internal RC while in runtime mode.

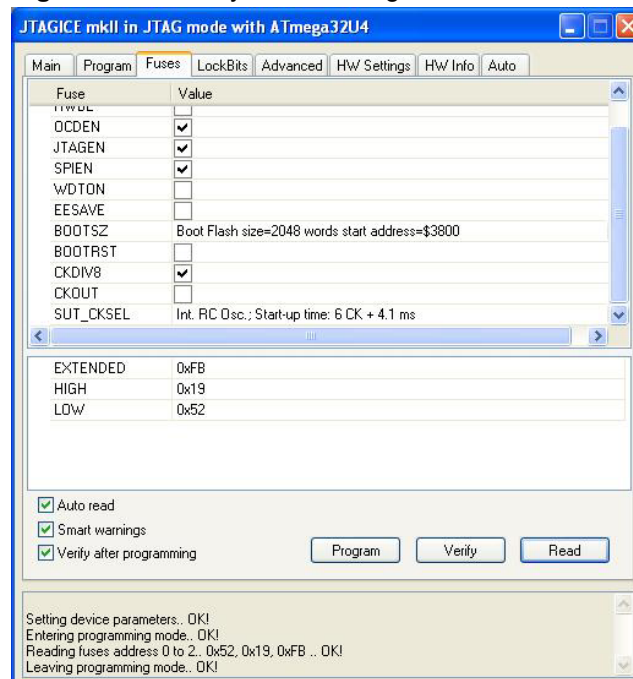
The purpose of this application note is to use the internal RC oscillator to build a USB low-speed application. Thus, due to the USB requirement, we will consider only two possible voltages: 3V and 5V.

2.1 Clock selection

The AVR® fuse settings control the system clock source being used. By default, the ATmega32U4RC is configured to use the calibrated internal RC oscillator, and so no modification needs to be done on the fuse setting. For this application note, we will use the default setting.

If the device does not have its default fuse setting, please note that the default fuse setting is as shown in [Figure 2-1](#):

Figure 2-1. Factory fuses setting.



2.2 Signature row and OSCCAL register

The calibration bytes are stored in the signature row. In addition to the calibration bytes, the signature row also contains the temperature sensor values at 25°C and 85°C. These values are used in the calibration algorithm. [Table 2-1](#) shows the address and the definition of each byte.

Table 2-1. Signature bytes.

Name	Address	Data	Definition
OSCCAL	0x00 Hi	--	Oscillator calibration byte (8MHz) at 25°C/3V
OSCCAL85	0x01 Hi	--	Oscillator calibration byte (8MHz) at 85°C/3V
OSCCAL25_5V	0x02 Hi	--	Oscillator calibration byte (8MHz) at 25°C/5V
TEMP25L	0x1C Lo	--	Temperature sensor low byte (room temp)
TEMP25H	0x1C Hi	--	Temperature sensor high byte (room temp)
TEMP85L	0x1D Lo	--	Temperature sensor low byte (hot temp)
TEMP85H	0x1D Hi	--	Temperature sensor high byte (hot temp)

The OSCCAL register is used to calibrate the internal RC oscillator and remove the frequency variation. It has the following configuration:

Figure 2-2. OSCCAL register.

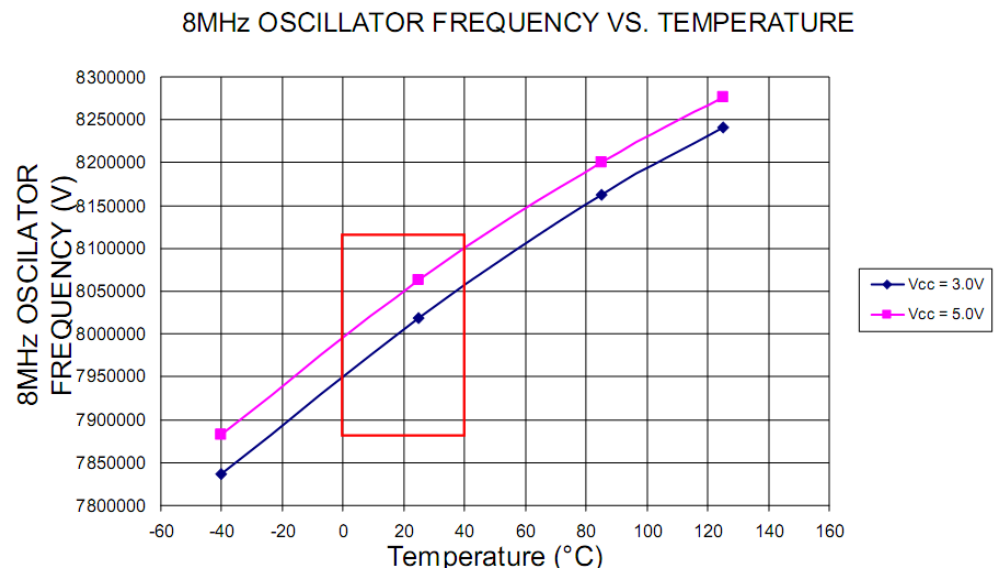
Oscillator Calibration Register – OSCCAL

Bit	7	6	5	4	3	2	1	0	
	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	OSCCAL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	Device Specific Calibration Value								

3 Oscillator characteristics

The frequency of the internal RC oscillator depends on the temperature and operating voltage. As seen from [Figure 3-1](#), frequency increases directly with increasing temperature, and increases slightly with increasing voltage. We can also see that the behavior at each voltage is similar, and an offset can be added to any 3V value to get the 5V value.

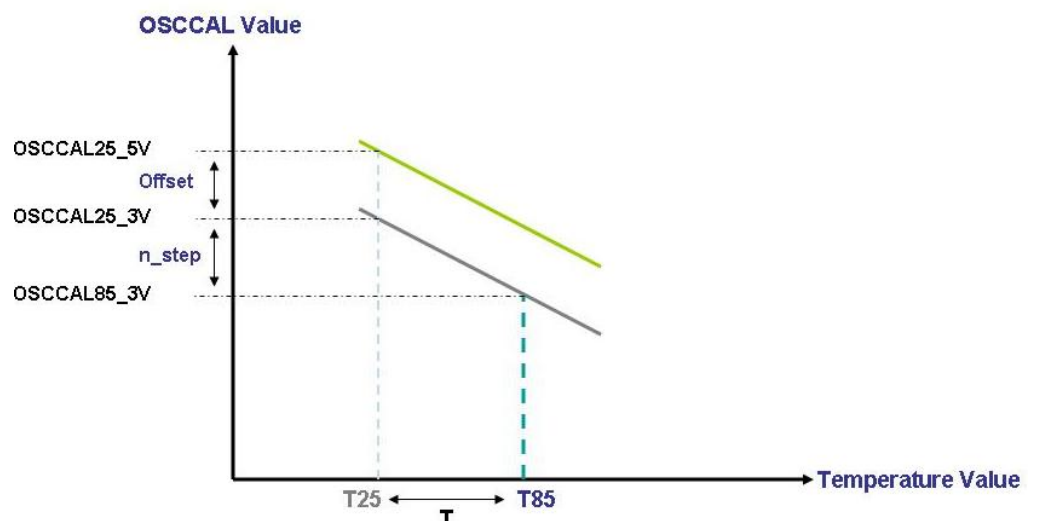
Figure 3-1. Oscillator frequency and influence of temperature and operating voltage.



3.1 OSCCAL vs. temperature

[Figure 3-2](#) shows the OSCCAL behavior according to the temperature. As explained above, the variation because of the voltage is similar in both cases (3V and 5V); an offset can be added to the 3V OSCCAL value to determine the 5V OSCCAL value.

Figure 3-2. OSCCAL vs. temperature.

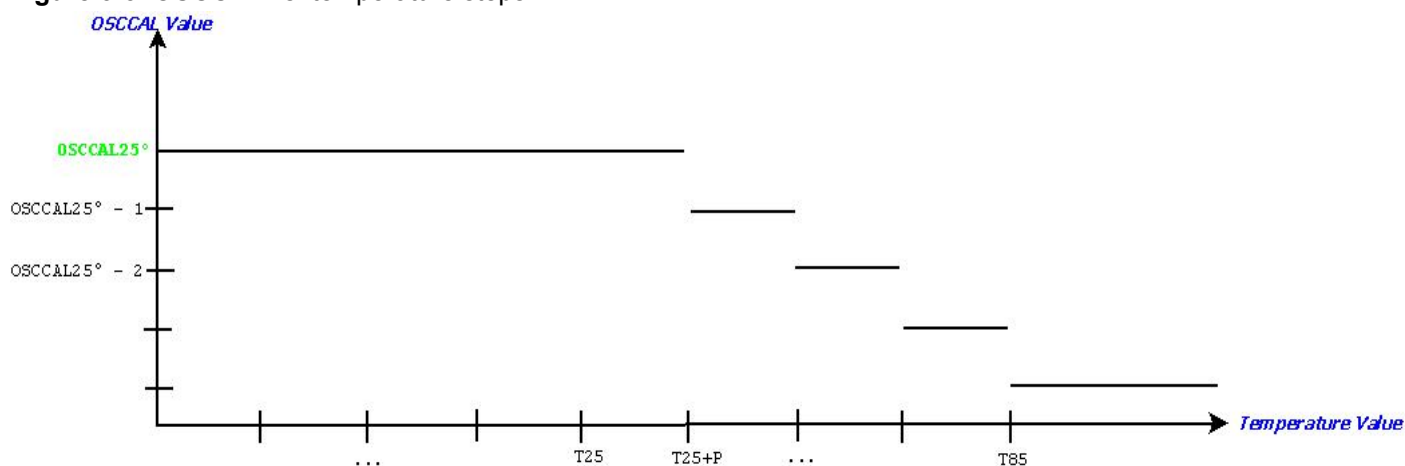


The variation in OSCCAL is not significant between 25°C and 85°C (between 1 and 6, approximately). This avoids any heavy processing needed to determine the OSCCAL value according to the temperature. Instead of calculating the value using the line equation, it can be determined by dividing the temperature into different steps, and assigning each step with an OSCCAL value. To do this, the following values are defined:

- Offset = OSCCAL25°_3V – OSCCAL85°_3V
- T = Temperature85° - Temperature25°
- n_step = OSCCAL25°_3V - OSCCAL25°_5V
- P = T / n_step: temperature step value

Figure 3-3 shows how an OSCCAL value is set according to the temperature range:

Figure 3-3. OSCCAL vs. temperature steps.



4 Calibration implementation

4.1 8MHz internal oscillator calibration

The internal RC oscillator calibration is implemented as a driver, and is included in a C file (`cal_inter_rc.c`) and the associated header file (`cal_inter_rc.h`). The functions used to perform the internal oscillator calibration are available in the C file, and are explained below.

4.1.1 `get_cal_values()`

This function allows reading of the signature bytes to get the factory calibration. This factory calibration is done at three points: 25°C at 3V, 25°C at 5V, and 85°C at 3V. Each point is represented by the temperature and OSCCAL values.

From the signature bytes, this function will read the following parameters:

- `osc_cal_25_3v`: OSCCAL value for 25°C at 3V
- `osc_cal_85_3v`: OSCCAL value for 85°C at 3V
- `osc_cal_25_5v`: OSCCAL value for 25°C at 5V
- `temp_sensor_25`: raw value of the temperature sensor at 25°C
- `temp_sensor_85`: raw value of the temperature sensor at 85°C

Using these parameters, this function sets the following additional parameters, which will be used by the `calibrate_inter_osc()` function (see Section 4.1.2):

- Offset: the offset to apply to get the OSCCAL value at 5V using the value at 3V
- Period: the temperature step value. An OSCCAL value matches each step

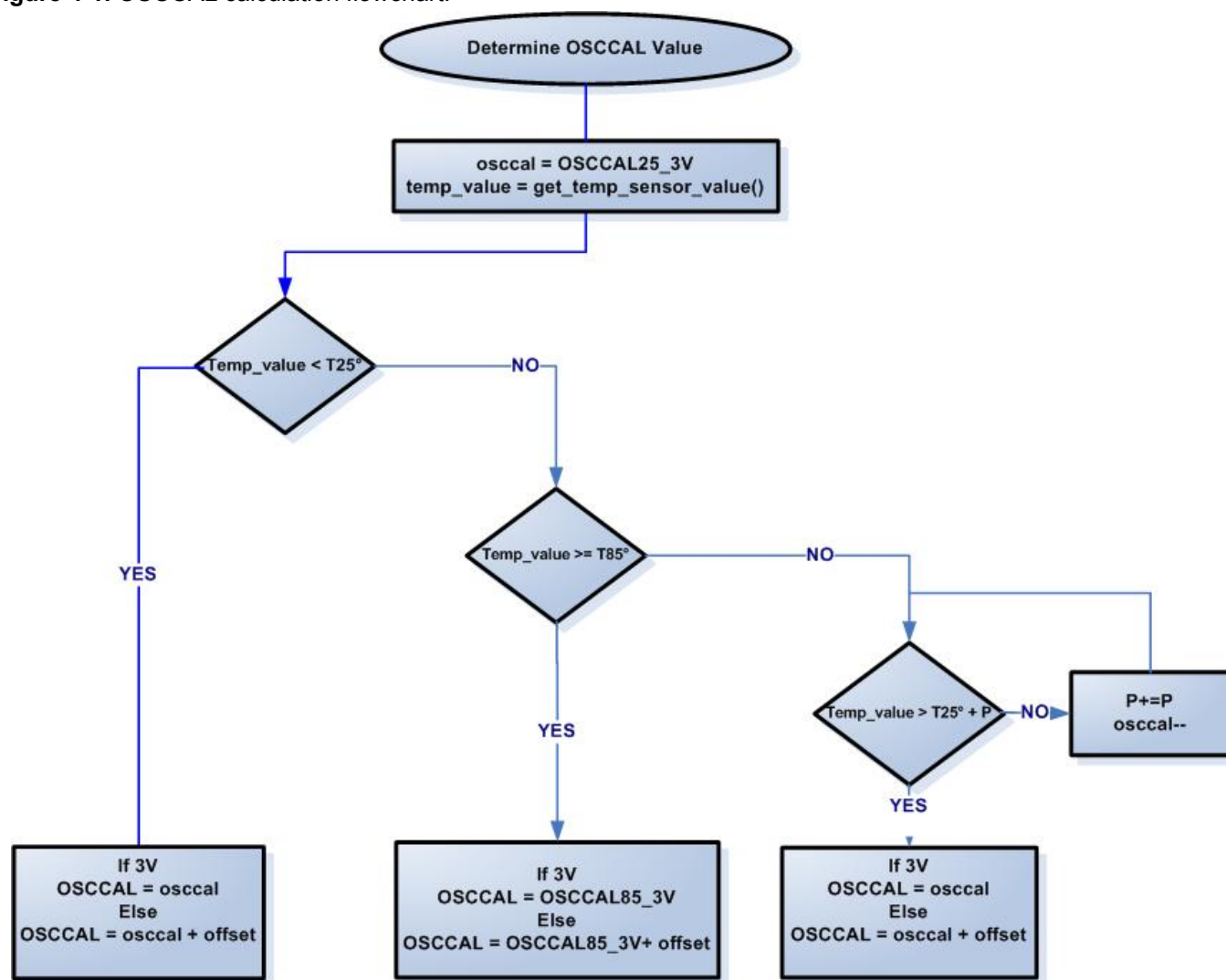
4.1.2 `calibrate_inter_osc()`

This is the main function, which is used to calibrate the internal RC oscillator according to the voltage and the temperature.

This function has, as a parameter, the voltage used by the application. Since the aim of this calibration is to use the internal oscillator for USB low speed, the only possible voltages are 3V (`vcc_3`) and 5V (`vcc_5`).

The function flowchart is shown in [Figure 4-1](#).

Figure 4-1. OSCCAL calculation flowchart.



4.2 Temperature sensor

As explained above, the OSCCAL value is calculated according to the temperature value (or temperature range). This temperature value is measured by the internal temperature sensor. For more information regarding this sensor, please refer to the [ATmega32U4RC datasheet](#).

To use this temperature sensor, a driver module is available that provides the required functions to get the temperature value. The driver is made of two files:

- temp_sensor.c: contains the functions' source code
- temp_sensor.h: contains the functions' prototypes that could be called in the application

From this driver, the function used in this example is:

- get_temp_sensor_val(): read and return the internal temperature sensor ADC channel. For this application, there is no need to get the real temperature value; the raw value provided by the ADC through the temperature sensor channel is enough

4.3 Signature byte access

As explained above, the default calibration values are stored in the signature row. A flash driver is available to read these values and provide easy access to the signature row.

The flash driver module is made of two files:

- `flash_drv.s`: contains the functions' assembly code
- `flash_drv.h`: contains the functions' prototypes that could be called in the application

From this driver, the function used to read the signature bytes is:

- `flash_read_sig()`: read and return the value of the signature byte at the address passed by the parameter of this function

5 Low speed USB HID mouse with internal RC oscillator

The USB HID mouse is an example of the use of the internal RC oscillator calibration. This example uses the internal RC oscillator to maintain the USB communication within the USB specification, 1.5MHz \pm 1%.

To calibrate the oscillator, the `calibrate_inter_osc()` function is called from the main function to set the first calibration. A scheduled task is added to manage calls to the `calibrate_inter_osc()` function from time to time to continue to calibrate the oscillator.

This example is available on the Atmel website, and can be used to build a USB low-speed mouse by using the internal RC oscillator without any crystal.



6 Conclusion

An external crystal adds significant cost to an application. This application note offers a way to build a USB application using the internal RC oscillator without a crystal, thus reducing the BOM cost and the PCB area. The calibrated internal RC oscillator can maintain the USB low-speed bandwidth, but the user must remember that the internal RC oscillator can provide a clock source only when building USB low-speed applications.

7 Table of contents

Features	1
1 Introduction	1
2 Theory of operation – the internal RC oscillator	2
2.1 Clock selection	2
2.2 Signature row and OSCCAL register	3
3 Oscillator characteristics	4
3.1 OSCCAL vs. temperature.....	4
4 Calibration implementation.....	6
4.1 8MHz internal oscillator calibration.....	6
4.1.1 get_cal_values()	6
4.1.2 calibrate_inter_osc()	6
4.2 Temperature sensor	7
4.3 Signature byte access	8
5 Low speed USB HID mouse with internal RC oscillator	9
6 Conclusion	10
7 Table of contents	11



Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: (+1)(408) 441-0311
Fax: (+1)(408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parking 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chou-ku, Tokyo 104-0033
JAPAN
Tel: (+81) 3523-3551
Fax: (+81) 3523-7581

© 2011 Atmel Corporation. All rights reserved.

Atmel®, Atmel logo and combinations thereof, AVR®, AVR® logo and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. **EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.** Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.