AN3964
Application note

STM32L1x temperature sensor example

Introduction

This application note describes a method of implementing a simple application for temperature measurement using the STM32L-DISCOVERY and 32L152CDISCOVERY boards. The solution described in this document uses the integrated temperature sensor of the STM32L1x microcontroller. The factory or user calibration method is described to improve the accuracy of the temperature sensor.

The demonstration application does not require any additional hardware. Once the STM32L-DISCOVERY and 32L152CDISCOVERY are updated with the associated firmware and is powered-up through a USB cable connected to the host PC, the application is ready to display the temperature of the STM32L1x microcontroller.

The temperature sensor example firmware is included in the STM32L1x discovery firmware package (STSW-STM32072) available from http://www.st.com.

Reference documents

- STM32L-DISCOVERY and 32L152CDISCOVERY user manual (UM1079)
- Getting started with software development toolchains for the STM32L-DISCOVERY and 32L152CDISCOVERY boards (UM1451)
- STM32L1x current consumption measurement and touch sensing demonstration (AN3413)
- Ultra-low-power STM32L15xx6/8/B datasheet
- Ultra-low-power STM32L151xC and STM32L152xC datasheet
- Ultra-low-power STM32L151xD and STM32L152xD datasheet
- Ultra-low-power STM32L162xD datasheet
- STM32L100xx, STM32L151xx, STM32L152xx and STM32L162xx advanced ARM-based 32-bit MCUs reference manual(RM0038)

The above documents are available at http://www.st.com.

Table 1. Application products and firmware

Type	Part numbers and product categories
MCU evaluation tools	STM32L-DISCOVERY, 32L152CDISCOVERY
MCU software	STSW-STM32072

Contents

1 Application overview 5

1.1 Temperature sensor 5

1.2 Temperature measurement and data processing 5

1.3 Application example description 6

2 Getting started 8

2.1 Setting up the board 8

2.2 Using the demonstration application 8

2.2.1 Temperature sensor calibration 8

2.3 Estimation of temperature sensor engineering tolerance 10

3 Software description 11

3.1 STM32L1x peripherals used by the application 11

3.2 STM32L15x standard firmware library configuration 12

4 Conclusion 12

5 Revision history 13

List of tables

Table 1. Application products and firmware 1
Table 2. Document revision history 13

List of figures

Figure 1. Example LCD display 7
Figure 2. Transfer characteristics of the temperature sensor 9

1 Application overview

This section describes how the temperature sensor works and how the temperature measurement is performed by the STM32L1x microcontroller embedded on the STM32L-DISCOVERY or 32L152CDISCOVERY.

A brief description of how the example temperature measurement application was implemented follows afterwards.

STM32L1xxDISCOVERY stands either for STM32L-DISCOVERY or 32L152CDISCOVERY evaluation kit throughout the document.

1.1 Temperature sensor

The temperature sensor integrated in the STM32L1x microcontroller provides an analog output voltage proportional to the chip junction temperature of the device.

Note: Please note that the temperature information provided by sensor is the thermal chip junction temperature (actual temperature of semiconductor surface) and may differ from the ambient temperature. Please see section “Thermal characteristics” of product datasheet for more details.

The integrated temperature sensor provides reasonably linear characteristics with a deviation typically of ± 1% from linear asymptotic functions and a temperature range equal to that of the device (–40 °C to 85 °C) with a maximum junction temperature of 150 °C.

The sensor provides good linearity but quite poor interchangeability and must be calibrated to obtain good overall accuracy. If the application is designed to only measure the relative temperature variations, the temperature sensor does not need to be calibrated.

1.2 Temperature measurement and data processing

The temperature sensor is internally connected to Channel 16 (ADC_IN16) of the ADC (analog-to-digital converter) in the STM32L1x and is used to sample and convert the temperature sensor output voltage. The raw ADC data must be further processed to display the temperature in a standardized unit of measurement (Celsius, Farenheit or Kelvin).

The ADC reference voltage ($V_{DDA} = V_{REF+}$) is connected to the 3 V V_{DD} power supply of the STM32L1xxDISCOVERY boards. If the V_{DD} value is not accurately known, as in case of battery-operated applications, it must be measured to obtain a correct overall ADC conversion range (see below section for details).

Temperature measurement on battery-operated devices

The power supply voltage applied to the microcontroller is subject to change on devices directly powered from a battery. The value converted by the ADC follows the drift of the battery voltage if the ADC reference voltage is tied to V_{DDA}, which is the case for devices in low pin-count packages. The supply voltage needs to be known to compensate for such voltage drift. The actual supply voltage (V_{DDA}) can be determined by using the embedded internal voltage reference (V_{REFINT}). The value sampled by the ADC (Val_V_{REFINT}) on ADC_IN17 internal reference input can be expressed by the following formula:

$$Val\_V_{REFINT} = V_{REFINT} \times 2^{12} / V_{REF+} = V_{REFINT} \times 4096 / V_{DDA}$$

The accurate embedded internal reference voltage (V_{REFINT}) is individually sampled by the ADC, and the converted value for each device ($Val\_V_{REFINT\_CAL}$) is stored during the manufacturing process in the protected memory area at address VREFINT_CAL specified in the product datasheet. The internal reference voltage calibration data is a 12-bit unsigned number (right-aligned bits, stored in 2 bytes) acquired by the STM32L1x ADC referenced to

$$V_{VREF\_MEAS} = V_{REF+} = 3V \pm 0.01V$$

The total accuracy of the factory measured calibration data is then provided with an accuracy of ± 5 mV (refer to the datasheet for more details).

We can determine the actual V_{DDA} voltage by using the formula above as follows:

$$V_{DDA} = 3 \times Val\_V_{REFINT\_CAL} / Val\_V_{REFINT}$$

The temperature sensor data, ValTS_bat, are sampled with the ADC scale referenced to the actual V_{DDA} value determined at the previous steps. Since the temperature sensor factory calibration data are acquired with the ADC scale set to 3 V, we need to normalize ValTS_bat to get the temperature sensor data (ValTS) as it would be acquired with ADC scale set to 3 V. ValTS_bat can be normalized by using the formula below:

$$ValTS = 3 \times ValTS\_bat / V_{DDA}$$

If the ADC is referenced to the 3 V power supply (which is the case of the STM32L1 Discovery) such a normalization is not needed and the sampled temperature data can be directly used to determine the temperature as described in *Section 2.2.1: Temperature sensor calibration*.

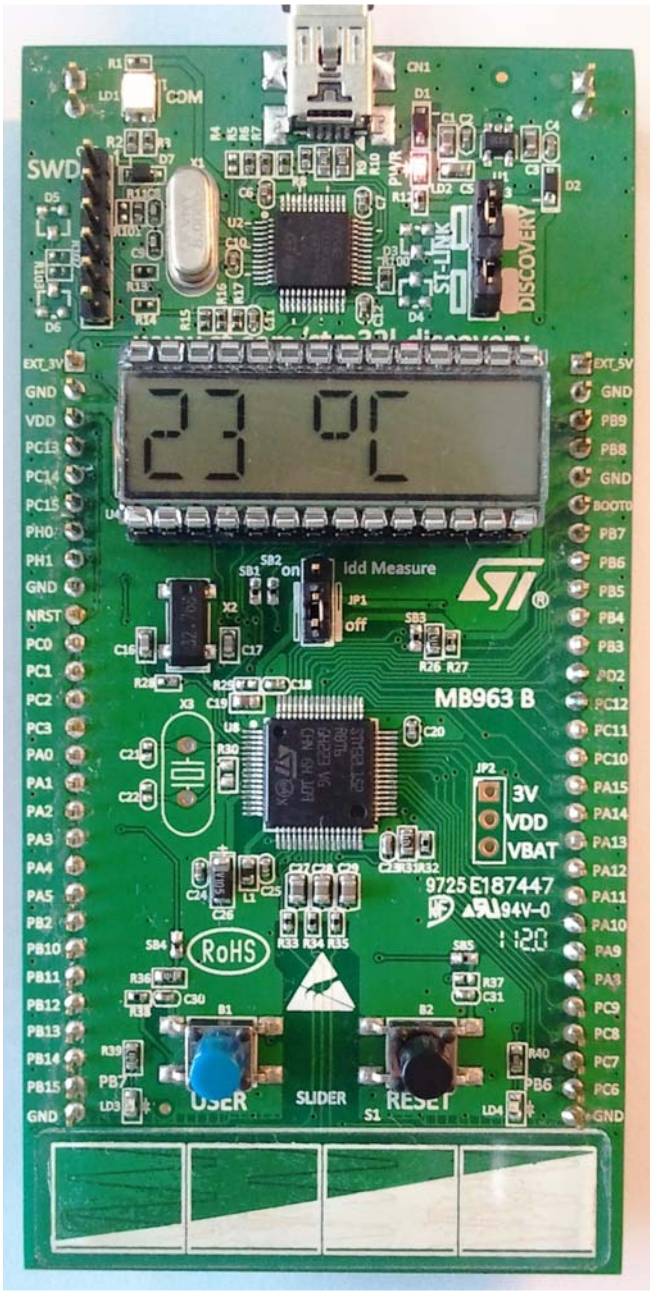
1.3 Application example description

Every 2 seconds the application acquires 16 samples from the temperature sensor voltage. The ADC raw data are filtered and averaged using an interquartile mean algorithm to reduce noise from the power supply system and the result is recalculated into standard units of temperature measurement (°C, in this example).

The LCD display is updated every 2 seconds either by ADC raw data or by the current temperature value in degrees Celsius. The user can switch between both temperature data representations by pressing the user button.

To demonstrate the low power capabilities of the STM32L1x ultra-low power microcontroller, the CPU is switched to Stop mode with the RTC (real-time clock) wake-up set to 2 seconds within the time interval between temperature sensor data measurements. The ADC data acquisition and data transfers are managed by direct memory access (DMA) while the CPU is in Low-power Sleep mode. The CPU is in Run mode at 16 MHz based on the HSI oscillator clock) only during the initialization phase and during the data processing period.

Figure 1. Example LCD display



2 Getting started

Before getting started, the firmware must be updated and hardware configured as described in the following sections.

2.1 Setting up the board

Updating the firmware

The STM32L1x program memory needs to be updated with the firmware associated with this application note. For information on how to update the firmware, please read the 'readme.txt' file in the project folder.

Used hardware components

This application example uses the hardware components available on the STM32L1xxDISCOVERY boards: the embedded peripherals of the STM32L1x microcontroller, the 6-digit LCD glass display and the user push-button. No additional components are required.

STM32L1xxDISCOVERY hardware settings

The I_{DD} jumper JP1 must be placed in the ON position.

Both jumpers on CN3 must be fitted to enable communication between the STM32L1x microcontroller and the ST-Link debugging tool through the serial wire debug (SWD) interface.

Note: All solder bridges must be in their default state as described in UM1079.

2.2 Using the demonstration application

It is very easy to start using the demonstration firmware.

When powered up, the temperature sensor application example first displays a welcome message before immediately displaying the current temperature in degrees Celsius with a 2-second refresh rate. When the User button is pressed once, the display shows the mean value of an array of 16 samples acquired by the ADC. One more press of the User button toggles between displaying the current temperature in degrees Celsius or the averaged value. The averaged value can be used later as a calibration point with a known temperature to improve overall accuracy of the temperature measurements.

2.2.1 Temperature sensor calibration

The temperature sensor calibration data are stored during the manufacturing process in the protected memory area from where the user can read it and use it to improve the accuracy of the temperature measurements. The two-point calibration data is measured during production:

- At ambient temperature (30 °C ± 5°C): TS_CAL1
- At hot temperature (110 °C ± 5°C): TS_CAL2.

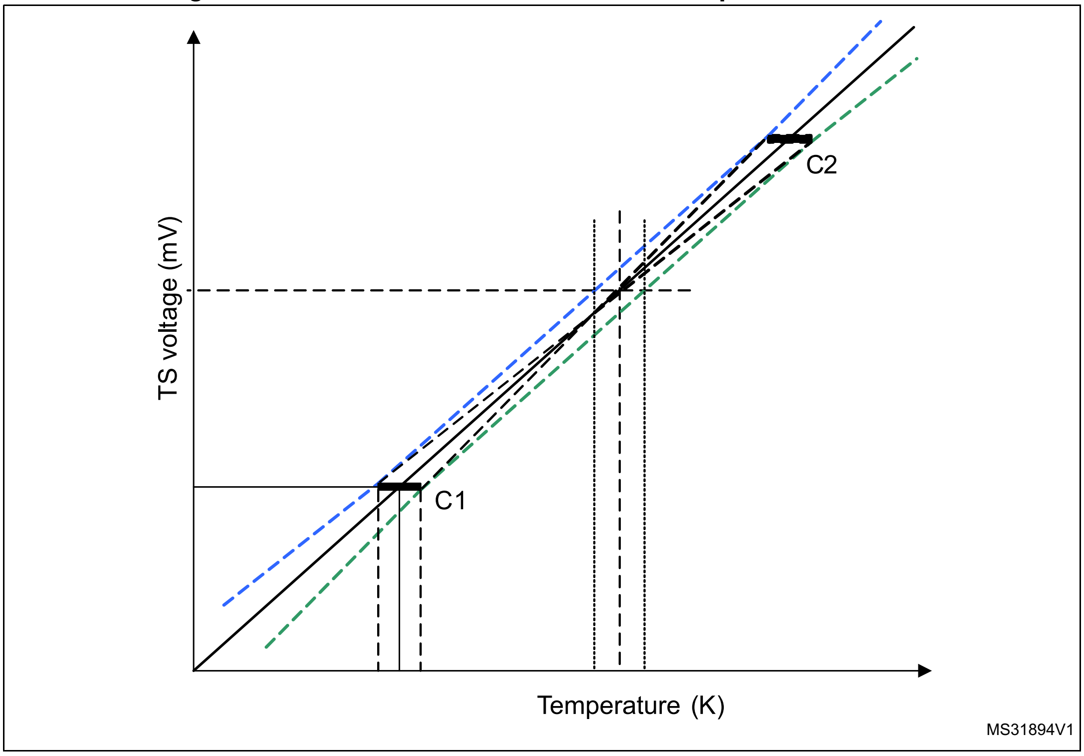
Refer to the product datasheet for the memory address where calibration data are stored.

The temperature sensor calibration data is a 12-bit unsigned number (stored in 2 bytes) acquired by the STM32L1x ADC with a 3 V (± 10 mV) reference voltage.

The factory calibration data are tested for validity when the example application is initialized. If data is present in the memory, it is used for temperature calculation. Otherwise, the user calibration data stored during user calibration in EEPROM memory area is tested and used instead. If the user calibration data is not available either, the default values are used for calculation. The factory calibration or user calibration data provides good accuracy of the temperature measurement.

The use of the default calibration data, which is statistically based on the typical characteristics of the temperature sensor, may provide less accurate temperature estimations due to significant variations of the temperature sensor characteristics during the manufacturing process. It is recommended to use either the factory calibration data or to perform the two-point calibration of the temperature sensor, which respects the individual characteristics of the temperature sensor, to obtain reasonably accurate measurements.

Figure 2. Transfer characteristics of the temperature sensor



The temperature can be evaluated from the digital value, ValTS, sampled by the ADC using linear approximation. It can be applied if the coordinates of two calibration points C1 and C2 are known as shown in *Figure 2*.

The current temperature can be evaluated as follows where the cold temperature coordinate pair is designated as (TC1, ValC1) and the hot temperature pair as (TC2, ValC2):

$$Temp = (TC2 - TC1) / (ValC2 - ValC1) \times (ValTS - ValC1) + TC1$$

Using the factory calibration data the formula can be rewritten as follows:

$$Temp = 80 / (TS\_CAL2 - TS\_CAL1) \times (ValTS - TS\_CAL1) + 30$$

2.3 Estimation of temperature sensor engineering tolerance

The two-point calibration method significantly improves the accuracy of the measurement as can be seen in *Figure 2*. The bias of the temperature measurement is mainly given by two sources; the temperature margin of the calibration points and the linearity of the sensor. Other sources of bias such as the ADC reference voltage margin can be effectively reduced. It can be neglected for factory calibrated values measured with the 3 V (± 10 mV) reference voltage.

The engineering tolerance of the temperature estimation is illustrated in *Figure 2* where it is limited by the two boundary lines of the minimum biased values (green) and the maximum biased values (blue). The area between the calibration points has a constant tolerance with a slight increase of the tolerance outside. For this reason, the recommended position of the calibration points should be as close as possible to the maximum and minimum values of the measurement range.

3 Software description

3.1 STM32L1x peripherals used by the application

This application example uses the following STM32L1x peripherals with the settings described below. For more information, please refer to the STM32L151xx datasheet.

Analog-to-digital converter (ADC)

The ADC performs analog-to-digital conversions of the internal reference voltage (4 samples) and of the temperature sensor voltage (16 samples) driven by DMA.

- ADC resolution: 12-bit
- ADC conversion mode: Scan mode driven by DMA
- ADC sampling time: 384 cycles

SysTick timer

The SysTick timer is used only to generate the delay needed for display refresh and is disabled during temperature measurements.

General-purpose inputs/outputs (GPIOs)

Ports C and E are connected to the User push-button and the LEDs.

- PB1 is set as an input floating pin with interrupt connected to User push-button
- PB7 (green LD3) and PB6 (blue LD4) are set as an output push-pull.
- During low power modes, I/Os are placed in analog input mode to reduce power consumption except for a few pins related to the hardware interface (PB7 - green LD3 and PB6 - blue LD4). It means that all Schmitt triggers on unused standard I/O pins are disabled to reduce power consumption.

LCD controller

The several functions available in the firmware library for the liquid crystal display (LCD) are used to initialize, clear, display strings and scroll messages needed in the application code.

Clocks

The high-speed internal (HSI) RC oscillator is selected as the main clock source.

The application manages the peripheral clocks depending on the selected power saving mode. When the device enters Stop mode, the HSI oscillator is switched OFF and the LSE crystal oscillator feeds the RTC until the device is woken up by an external event (RTC wakeup or USER button pushed). When exiting Stop mode, the MCU switches back the system clock from the defailt MSI oscillator to the HSI oscillator.

3.2 STM32L15x standard firmware library configuration

The *stm32l*1xx_conf.h file of the STM32L1x standard firmware library allows you to configure the library by enabling the peripheral functions used by the application.

The header files of the library modules are included in the *stm32l*1xx_conf.h file as listed below:

- #include stm32l1xx_adc.h
- #include stm32l1xx_exti.h
- #include stm32l1xx_flash.h
- #include stm32l1xx_gpio.h
- #include stm32l1xx_syscfg.h
- #include stm32l1xx_lcd.h
- #include stm32l1xx_pwr.h
- #include stm32l1xx_rcc.h
- #include stm32l1xx_rtc.h
- #include misc.h

The corresponding library modules must be included in the project for successful compilation and linking.

4 Conclusion

This application note shows how to use the internal temperature sensor embedded in your STM32L1x microcontroller.

The firmware example associated with this application note allows you to explore the temperature sensing capability of STM32L1x microcontrollers and at the same time demonstrate its ultra low-power features. It can be used as a starting point for your own development.

5 Revision history

Table 2. Document revision history

Date	Revision	Changes
27-Sep-2011	1	Initial release.
04-Jul-2013	2	Added 32L152CDISCOVERY and related information. Addeed reference to STSW-STM32072 firmware. Replaced STM32L by STM32L1x in the whole document. Updated memory address in *Section 1.2: Temperature measurement and data processing*. Updated *Section 2.2.1: Temperature sensor calibration* to add *Section : Temperature measurement on battery-operated devices*. Changed reference voltage in *Section 2.3: Estimation of temperature sensor engineering tolerance*. Updated *Section : Clocks*.
16-Jul-2013	3	Updated *Table 1: Application products and firmware*.