AN3397
Application note

Temperature sensor example using the STM8L-DISCOVERY board

Introduction

This application note proposes two methods for measuring and displaying temperature using the STM8L built-in temperature sensor connected to the STM8L-DISCOVERY board. The two methods are illustrated in the application code example provided with this application note within two configurable modes: normal and advanced.

More generally, this application note explains the care that should be taken with the temperature sensor, and describes the peripheral settings and configuration required to develop the temperature sensor application.

The temperature sensor application does not require any additional hardware. Once the STM8L-DISCOVERY is powered up through a USB cable connected to the host PC, the application immediately starts, and the chip temperature is continuously displayed on the liquid crystal display (LCD). The minimum and maximum temperatures recorded over the past runtime period are automatically stored by the application, and are available for display by pressing the 'user' push button on the STM8L-DISCOVERY. The temperature tendency information is displayed as a bar graph on the LCD.

Reference documents

- STM8L-DISCOVERY user manual (UM0970)
- Developing and debugging your STM8L-DISCOVERY application code user manual (UM0991)
- Getting started with STM8L-DISCOVERY user manual (UM1014)
- STM8L15xx and STM8L162x microcontroller family reference manual (RM0031)
- STM8L151x4, STM8L151x6, STM8L152x4, STM8L152x6 datasheet
- STM8L151x8, STM8L152x8, STM8L151R6, STM8L152R6 datasheet
- STM8L151C2/K2/G2/F2 and STM8L151C3/K3/G3/F3 datasheet
- STM8L162R8 and STM8L162M8 datasheet

The above documents are available on *www.st.com*.

Table 1. Applicable products and tools

Type	Part number or product category
Microcontrollers	STM8L Ultra Low Power
Evaluation tools	STM8L-DISCOVERY

Contents

List of tables

List of figures

1 Temperature sensor overview

This section provides an overview of the temperature sensor (TS) embedded in STM8L15xxx devices. It describes how the TS works, the data processing required, and the accuracy that can be expected.

1.1 Temperature sensor

The STM8L15xxx built-in TS is based on a semiconductor diode embedded in the MCU. The sensor provides an analog output voltage proportional to the junction temperature of the device.

Note: The temperature information provided by the sensor is the thermal junction temperature (temperature of the semiconductor), and may differ from the ambient temperature (MCU package).

The diode bias current is held as constant as possible. Consequently, the forward bias voltage across the diode is temperature dependent, with a temperature coefficient of about 1.62 mV/°C (see the device datasheet). The TS provides a reasonably linear characteristic with a typical deviation of ±1 % (±2 % max) from linear asymptotic functions and a temperature range equal to that of the device (-40 °C to 85 °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 measure only the relative temperature variations, the temperature sensor does not need to be calibrated.

1.2 Temperature measurement and data processing

The TS is internally connected to the ADC_TS channel of the analog-to-digital converter (ADC), which is used to sample and convert the sensor analog output voltage. The raw ADC data must be further processed to display the temperature in standardized units of measurement (for example, Celsius, Fahrenheit or Kelvin)

The ADC reference voltage is connected to the 3-V V_{DD} power supply of the STM8L-DISCOVERY board. If the V_{DD} value is not accurately known, as is the case with battery-operated applications, it must be measured to obtain a correct overall ADC conversion range. To easily measure the power supply voltage, we recommend using the embedded internal reference voltage (bandgap). The device is well calibrated during the manufacturing process, and its calibration data is stored in the protected memory area at address 0x4910.

Note: Check that the bandgap factory calibration data (VREFINT_Factory_CONV) is available from your STM8L15xxx MCU device.

The embedded bandgap voltage calibration data is a 12-bit unsigned integer acquired by the ADC with V_{DD} = 3 V ± 10 mV (please refer to the device datasheet for more details).

1.3 Temperature sensor accuracy

Without calibration the TS is likely to give error readings in the order of ±30 °C. If the sensor is calibrated, significantly more accurate temperature measurements can be obtained. For example, if the TS characteristic slope and offset are calibrated, a five-fold improvement in accuracy over the full temperature range can be obtained. Even greater accuracy is possible if the temperature range is limited (see *Figure 1*).

STM8L15xxx devices are provided with a calibration data point, V90, which is factory measured at hot temperatures (90 °C ± 5 °C) during the manufacturing process and stored in the protected memory area of the device at address 0x4911 (see device datasheet for details). This single calibration point may be sufficient depending on the precision required by the application, but it is highly recommended to perform a second calibration point (user calibration point) at ambient temperature.

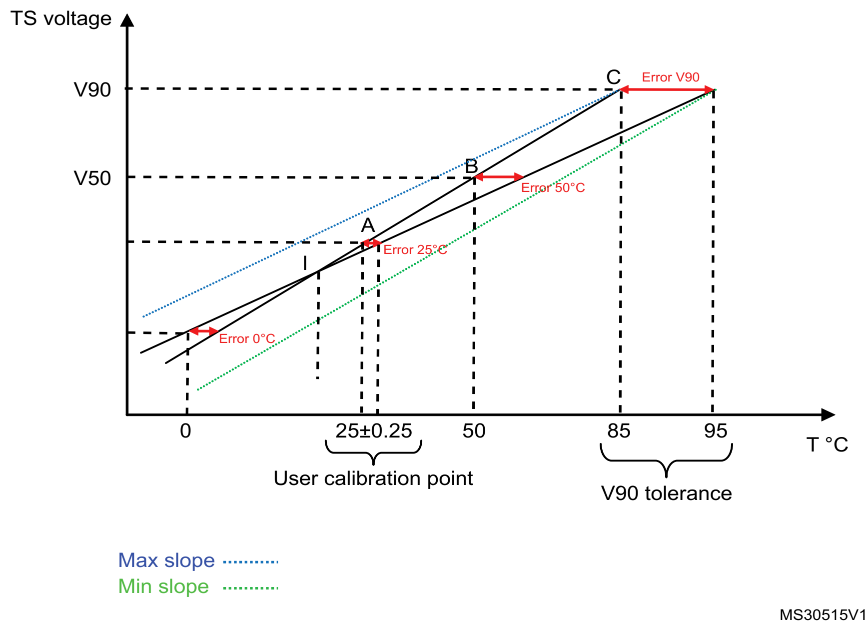
Devices with a two-point calibration are generally suitable for most embedded applications because the deviation versus the ideal linear curve is restrained and the individual characteristics of the temperature sensor are respected.

1.4 Temperature sensor engineering tolerance

The engineering tolerance of the temperature estimation is illustrated in *Figure 1*. It is limited by two boundary lines: the minimum biased values (green line) and the maximum biased values (blue line). This tolerance is directly linked to both the temperature margin of the factory calibrated data at hot temperatures (V90 = 90 °C ± 5 °C) and the Avg_slope uncertainty (1.62 mV/°C ± 0.03 mV/°C) specified in the STM8L15xxx datasheets.

Figure 1 shows that the user calibration point significantly improves the accuracy of the temperature measurement since reducing the whole temperature tolerance area to a lower range means that more reasonable results are obtained. The recommended position of the calibration points should be as close as possible to the minimum and maximum values of the measurement range to cover the full device temperature range. However, the user calibration point can be performed at ambient temperature (20 °C to 25 °C) as it is the easiest way to proceed.

Figure 1. Temperature sensor engineering tolerance



1.5 Temperature sensor error estimation

Once the temperature sensor has been correctly calibrated with both factory and user calibration points, the engineering tolerance of the temperature measurement is significantly improved (see area between the two black lines in *Figure 1*.

The temperature measurement error is caused by the following:

- Temperature sensor linearity
- ADC total unadjusted error (TUE)
- Calibration related error

Figure 1 shows that the temperature error is minimized in the neighbourhood of the user calibration point and that it slightly increases with distance from this point.

Temperature error calculation

The temperature measurement error can be calculated as explained below. Note that the user calibration point needs to be defined before the calculation.

1. The maximum linearity error is ±2 °C (see STM8L15xxx datasheets).
2. The ADC TUE is 4 LSB considering V_{REF} = 3 V (see STM8L15xxx datasheets).

To calculate the temperature measurement error linked to the ADC total unadjusted error, use *Equation 1*.

Equation 1

$$\text{ADC TUE (°C)} = \frac{\text{TUE} \times \left(\frac{V_{REF}}{12\text{bits(ADC)}}\right)}{\text{avg\_slope}}$$

Solving with values: (4 x 3/(4096 * 1.62 mV/°C) = ±1.80 °C.

3. The maximum calibration related error is considered to be ±0.25 °C at ambient temperature. In this situation, the temperature error (±0.25 °C) is linked to the accuracy of the thermometer.

Total temperature error calculation

Consequently, the total measurement error at 25 °C is the sum of the ADC TUE in °C (±1.80 °C) and the calibration related error (±0.25 °C) which is ±2.05 °C.

Note: The linearity error is cancelled at the user calibration point since it is obtained from the measurement.

By using Thales theorem, we can estimate the temperature error for any point in the temperature sensor range.

First, 'I' (see *Figure 1*) must be calculated (using Thales theorem).

$$\frac{\text{Error (25 °C)}}{\text{Error (90 °C)}} = \frac{IA}{IC}$$

Solving with values: 0. 5/10 = (24.75 - I)/(85 - I)

Therefore, I = 21.6

An example of the calculation error at 50 °C is as follows:

$$\frac{IB}{IC} = \frac{\text{Error (50 °C)}}{\text{Error (85 °C)}}$$

Solving with values: (50 - 21.6)/(85 - 21.6) = Error(50 °C)/10

Therefore, the calculation error at 50 °C = ±4.47 °C.

2 Temperature sensor application description

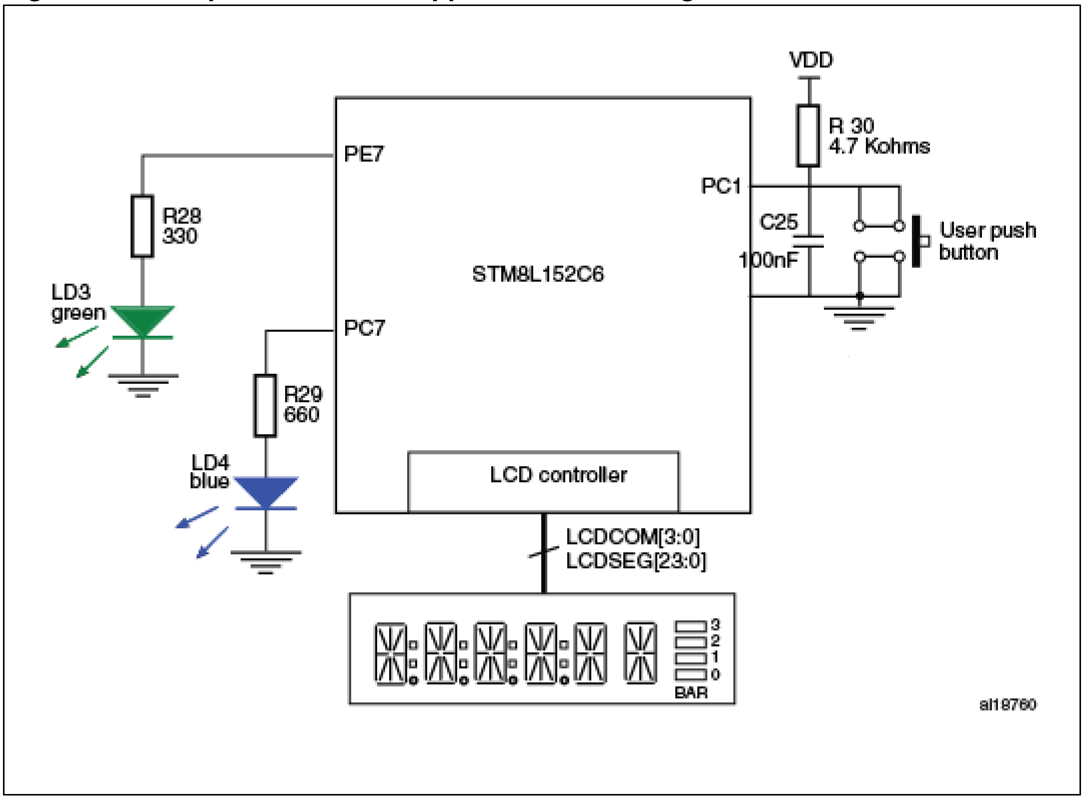
2.1 Updating the firmware

The STM8L program memory needs to be updated with the latest firmware version associated with this application note. This firmware is available on *www.st.com* (www.st.com/stm8l-discovery). For information on how to proceed, please read UM1014 (Getting started with STM8L-DISCOVERY).

2.2 Hardware required

The temperature sensor application uses the STM8L-DISCOVERY on-board 6-digit/4-bar LCD glass display and user push-button. No additional components are required.

Figure 2. Temperature sensor application block diagram



1. Legend: Green LED = LD3 green; Blue LED = LD4 blue

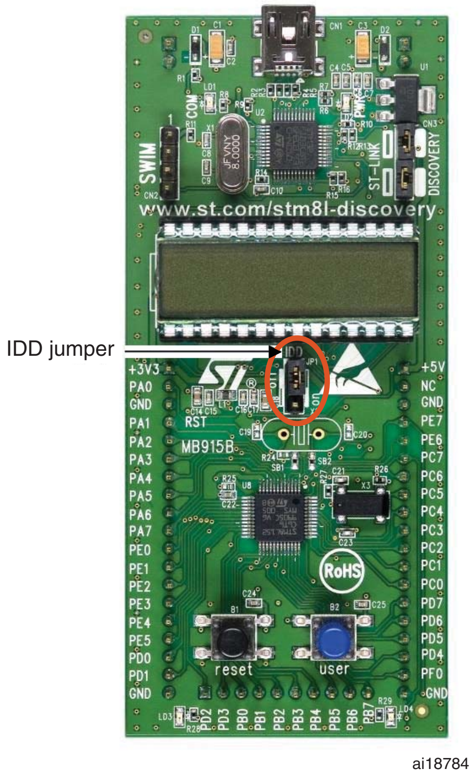
2.3 STM8L-DISCOVERY hardware settings

The IDD jumper (see *Figure 3*), JP1, should be placed in the 'off' position.

Both jumpers on CN3 must be fitted to enable communication between the STM8L microcontroller and the ST-LINK debugging tool through the serial wire interface module (SWIM).

Note: Refer to UM0970 STM8L-DISCOVERY user manual for details.

Figure 3. STM8L-DISCOVERY hardware settings



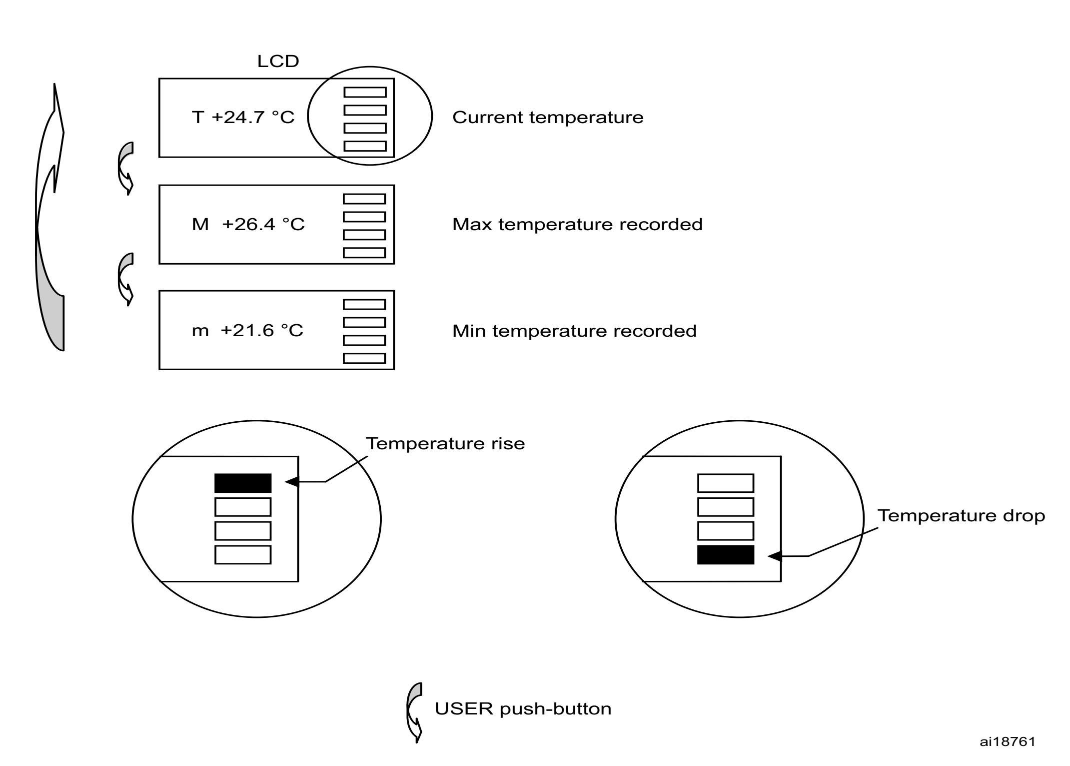
2.4 Temperature sensor application principle

The STM8L152C6 has a built-in temperature sensor (TS) which is internally connected to the analog-to-digital converter (ADC) TS input channel. The ADC converts the sensor output voltage into a digital value. The result is recalculated into standard temperature units (°C in this example) and displayed on the LCD.

The temperature value displayed on the LCD panel is refreshed after every 1 s time period. To improve measurement reliability, this data point is obtained by averaging 20 measurements. Between each temperature measurement, the STM8L152C6 enters active halt mode to minimize average power consumption, then automatically wakes up 50 ms later to get a new data point and process it.

The TS application may also display the minimum and maximum temperature values recorded since the STM8L-DISCOVERY was powered up. By default, the current temperature is displayed with the symbol 'T'. When the user push-button is pressed once, the LCD displays the maximum temperature recorded with the symbol 'M'. The second time the user push-button is pressed, the minimum temperature recorded is displayed with the symbol 'm'. Pressing the user push-button again displays the current temperature once more (see *Figure 4*).

Figure 4. TS application user interface



The TS can also be used to detect temperature variations. Using the LCD bar graph, the application displays the temperature trend over the past five-minute period. If the temperature values recorded during that time have dropped, BAR0 is turned 'on'. On the other hand, if the temperatures have increased, BAR3 is turned on.

2.4.1 Temperature calculation theory

The voltage delivered by the TS is a linear function (linearity error ±2 °C max) of the diode temperature junction.

To calculate the temperature from the voltage delivered by the TS (V_{SENSE}), *Equation 2* or *Equation 3* is used depending on the reference chosen (temperature in K or temperature in °C):

Equation 2

$$T[K] = V_{SENSE} / Avg\_slope$$

Equation 3

$$T[°C] = (V_{SENSE} / Avg\_slope) - 273.15$$

Where:

- K is the absolute temperature
- V_{SENSE} is the measured voltage corresponding to the digital value present in the ADC data register.
- Avg_slope (*Equation 4*) is the average slope of the characteristic (V_{SENSE} vs. Temperature).

Equation 4

$$Avg\_slope = (\Delta V) / (\Delta T) = (V90 - V_{SENSE}) / (T90 - T)$$

2.4.2 Operating modes

To make V90 appear in the formula, the reference has been transposed, with the origin defined by T90/V90 has been determined for the temperature characteristic. Consequently, the temperature formula in *Equation 2* and *Equation 3* becomes *Equation 5*.

Equation 5

$$T[°C] = V_{SENSE} / Avg\text{-}slope(offset = 0)$$

The TS application may be configured in two operating modes:

- Normal mode
- Advanced mode

Normal mode

Normal mode is the default mode configured in the software example. In this mode, the blue LED (see *Figure 2*) is switched off.

To activate this mode, comment the following define statement in the main.c file of the TS application code:

// #define ADVANCED_MODE

In Normal mode, the temperature is processed from default (typical) values of Avg_slope and sensor reference voltage at 90 °C (V90) which are available in the STM8L15xxx datasheets.

Avg_slope = 1.62 mV/°C

V90 = 0.597 V

This method has certain advantages. It is straightforward and the measurement is rapidly obtained as it does not require the calculation of a second calibration point. However, data which are statically based on the typical characteristics of the TS, may provide less accurate temperature estimations due to significant variations of the temperature characteristics during the manufacturing process. If an application needs very accurate measurements, it is better to use the Advanced operating mode described below.

Advanced mode

To activate advanced mode, uncomment the following define statement in the main.c file of the TS application code:

```
#define ADVANCED_MODE
```

When this mode is selected, the blue LED (see *Figure 2*) is switched on.

In advanced mode, V90 is obtained precisely by reading the TS factory calibration byte: TS_Factory_CONV_V90. This byte represents the 8 LSB of the result of the V90 12-bit ADC conversion (see STM8L15xxx datasheet). The MSB have a fixed value: 0x3.

The advantage of using this result is to get a more precise V90 value which consequently leads to greater precision when measuring the temperature. To further reduce the error, it is recommended to define another calibration point at ambient temperature to accurately define the average slope parameter and the offset. This TS application code (user calibration point) implements a technique to obtain and store this user calibration (see *Section 2.4.3: Calibration technique*) and consequently gives a more precise temperature slope. The average slope is accurately calculated from the user and factory calibration data using *Equation 4* and replacing V_{SENSE} with $V_{AMBIENT}$:

Equation 6

$$\text{Avg\_slope} = (\Delta V)/(\Delta T) = (V90 - V_{AMBIENT})/(T90 - T)$$

Before proceeding to perform the user calibration, the factory calibration data point is tested for validity when the application is initialized. If the factory calibration data point is present in the memory @0x4911, it is used for temperature calculation. Otherwise, the user calibration procedure is bypassed and the application automatically switches back to normal mode. Advanced method is more difficult to implement and has more constraints but, it gives more precise results.

Note: In any application, a user may decide to make a trade-off and use the V90 datasheet typical value and the user calibration point to obtain less accurate but nevertheless very good temperature calculation results.

2.4.3 Calibration technique

The calibration technique is a one-off procedure which must be done when the STM8L-DISCOVERY is received. It consists of calculating the average slope of the TS, and is available only when the TS application Advanced mode is selected.

The calibration technique makes use of *Equation 5* to calculate the average slope.

The factory calibration value, V90, is obtained from a fixed data address (see *Advanced mode*). $V_{AMBIENT}$ is measured during the calibration procedure. To accurately calculate the average slope, $T_{AMBIENT}$ must be measured using a temperature measuring device such as a thermometer. The `#define Tambient` must be updated with the temperature given by the thermometer.

By default, $T_{AMBIENT}$ (which can be found in the main.c file) is set to 25 °C in the TS application code.

To start the calibration procedure, press the user push-button for at least 2 s, then release the button. If calibration is successful, the message ‘CAL-OK’ is displayed on the LCD screen. If no factory calibrated value is present, or if the present value is out of range (580 mV - 614 mV), the message 'CALERR' is displayed. Press the user push-button again to continue. When calibration is successful, the value is stored in the Data EEPROM of the microcontroller. By default, the value is stored as two bytes in the Data EEPROM addresses 0x1000 (MSB) and 0x1001 (LSB).

In Advanced mode, an automatic calibration is performed by the software if no calibration has been made by the user.

Note: Automatic calibration may not be accurate because, by default, $T_{AMBIENT}$ is set to 25 °C.

3 Software description

3.1 STM8L peripherals used by the application

The application software uses the standard STM8L firmware library to control the peripherals. The peripherals and functions used by the application software are listed below:

Clock

The HSI (high speed internal) 16 MHz oscillator is selected as clock source:

- System clock divider = 0
- f_{CPU} = 16 MHz

LCD controller

The LCD controller drives the LCD glass mounted on the DISCOVERY board. It is interfaced using 24 segment lines and four common lines, and is configured as follows:

- CLK (PSC = 1 and DIV = 31)
- Duty = 1/4
- Bias = 1/3
- LCD voltage source = internal
- Contrast = 3.10 V
- No dead time
- Pulse on duration = 1

ADC

The ADC converts the TS output voltage into a digital value. The ADC peripheral is configured as follows:

- ADON = 1 and TSON = 1
- Single conversion mode
- 12-bit resolution
- ADC clock frequency = f_{CPU} (clock prescaler = 0)
- Sampling time = 192 ADC clock cycles (12 µs @ f_{CPU} = 16 MHz)
- TS channel selected

Real-time clock (RTC)

The RTC generates the time base for waking up the microcontroller from low power Active-halt mode. The RTC peripheral is configured as follows:

- Low-speed external (LSE) clock selected as RTC clock source by default. The low-speed internal (LSI) clock can also be used by removing the USE_LSE definition in the project options.
- RTC clock prescaler = RTC clock source/1
- Wakeup clock selection bits, WUCKSEL[2:0], are set to 000: RTCCLK/16 to generate an RTC counter timebase of approximately 500 µs
- Wakeup counter registers, WUTRx, are set to 100 to generate a 50 ms wakeup time

General purpose input/output (GPIO)

The GPIO ports C and E are interfaced with a user push-button and light-emitting diodes (LEDs). The GPIO peripheral is configured as follows:

- PC1 set as input floating pin with interrupt (the user push-button)
- PC7 and PE7 set as output push-pull pin (LED4 and LED3 respectively)
- When the user push-button is pressed, an interrupt with interrupt sensitivity set to falling edge and low level is generated on PC1.

3.2 Application software flowchart

3.2.1 Main TS application routine flowchart

The main TS application routine (*Figure 5*) initializes all the peripherals using the settings listed in *Section 3.1: STM8L peripherals used by the application*.

After initialization, interrupts are enabled and the application operation mode is tested.

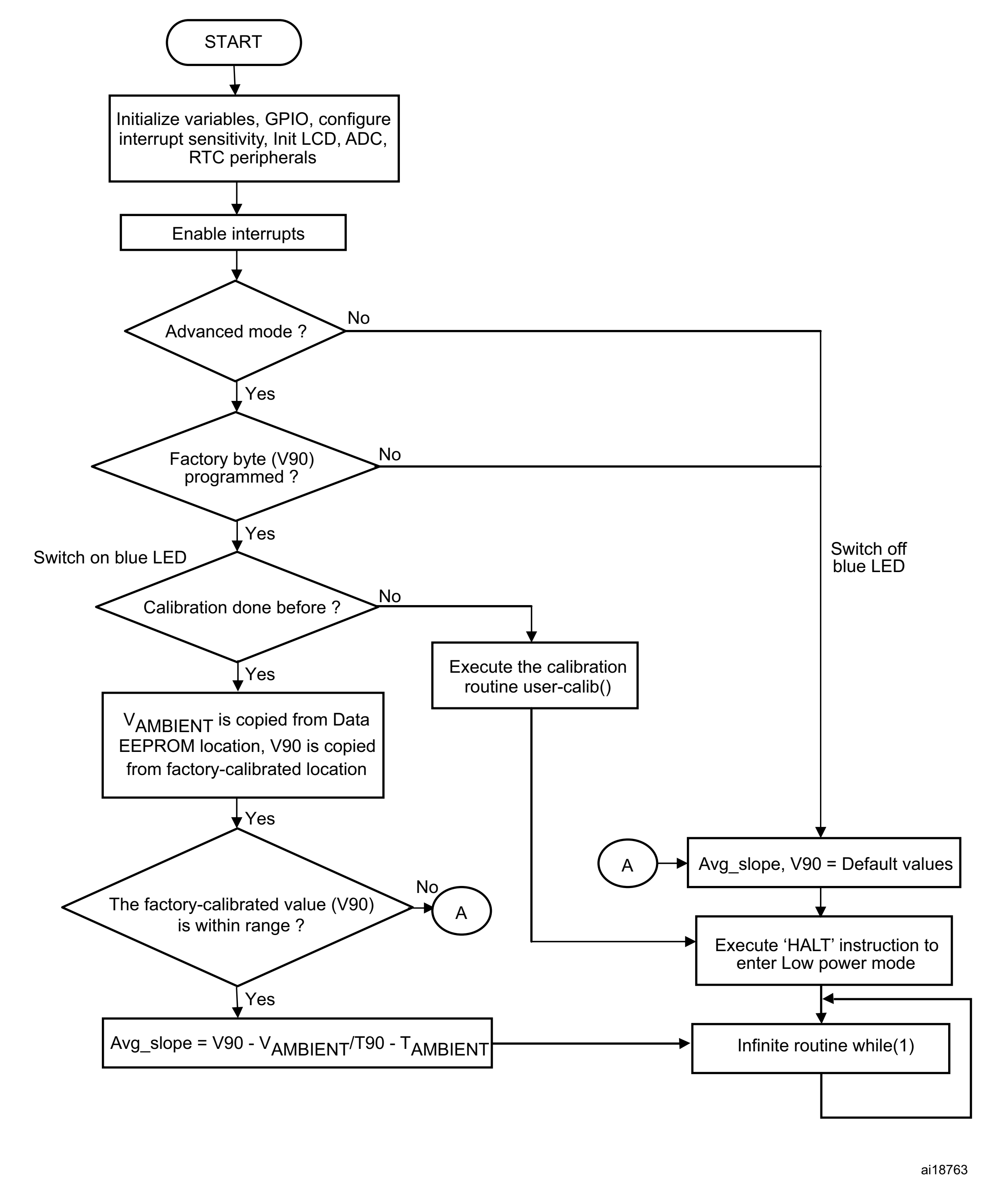
If #define ADVANCED_MODE is commented in the main.c file, the application automatically switches to Normal operating mode using the typical V90 value and the default Avg_slope (*Section 2.4.2: Operating modes*).

On the other hand, in Advanced operating mode, the factory calibration data point is tested for validity. If the factory calibration point does not exist or is not within the tolerance defined in the datasheet, the application automatically switches back to Normal mode. If the factory calibration point does exist, the application tests whether or not the user calibration point has already been processed.

If the user calibration point has been processed, the application goes directly to the infinite loop for temperature acquisition.

If the user calibration point has not been processed, the application uses this data point to accurately calculate the sensor characteristic slope before going into the main infinite loop.

Figure 5. Main TS application routine flowchart



3.2.2 User calibration routine flowchart

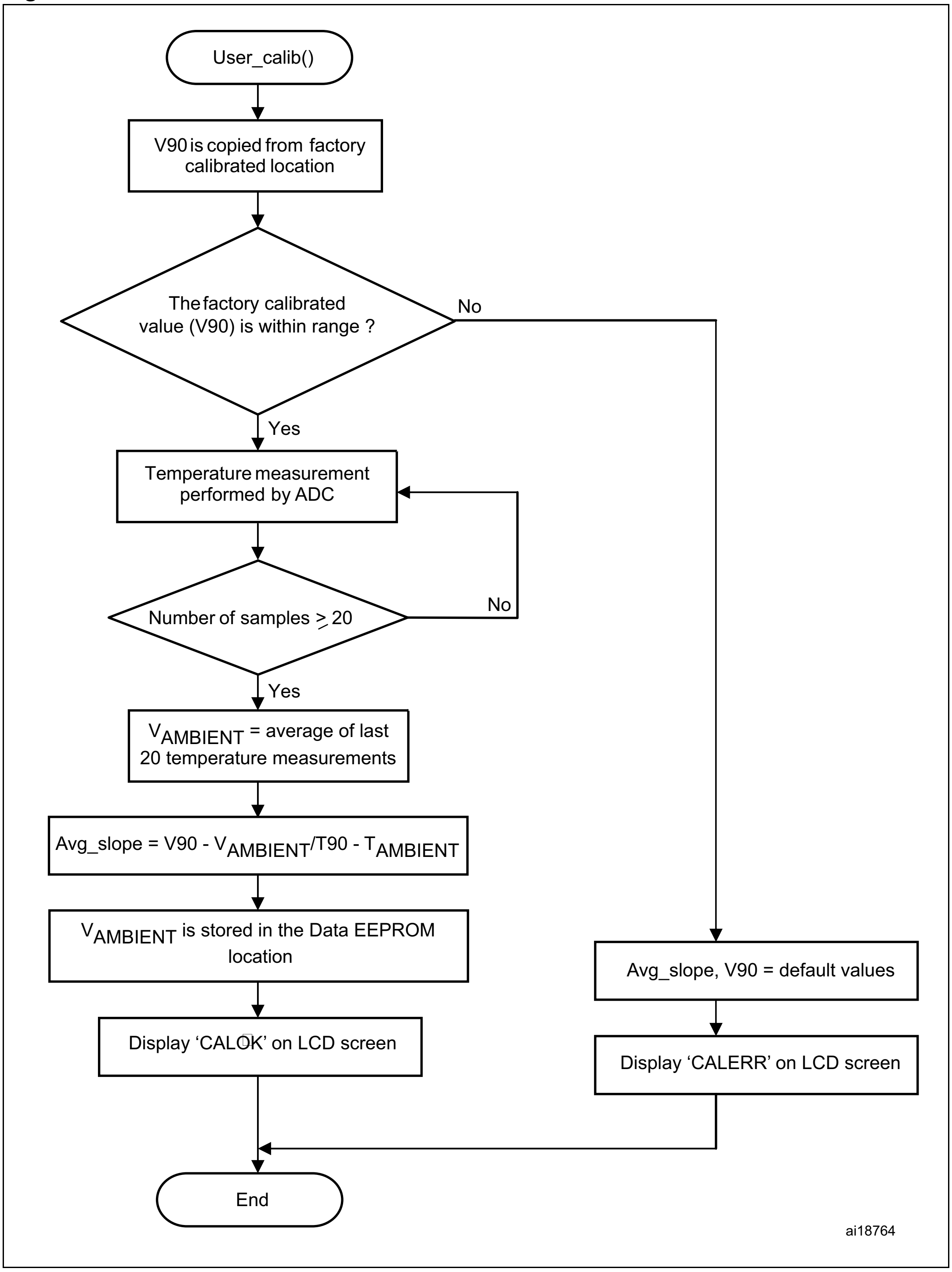
The user calibration routine performs the ADC measurement (see *Section 2.4.3: Calibration technique*) at ambient temperature ($V_{AMBIENT}$) and calculates the average slope. These values are stored in the Data EEPROM.

This routine is automatically launched in temperature sensor Advanced mode when the user calibration point has not been previously processed. It may also be forced by the user if the calibration point is to be updated.

Before proceeding to this calibration, the #define Tambient in the main.c file must be updated with the ambient temperature value of the environment. It is set by default to 25 (for 25 °C).

As shown in *Figure 6*, the temperature measurement is averaged from 20 acquisition samples, and the resulting data is stored in the EEPROM to be re-used even after STM8L-DISCOVERY is powered off.

Figure 6. User calibration routine flowchart

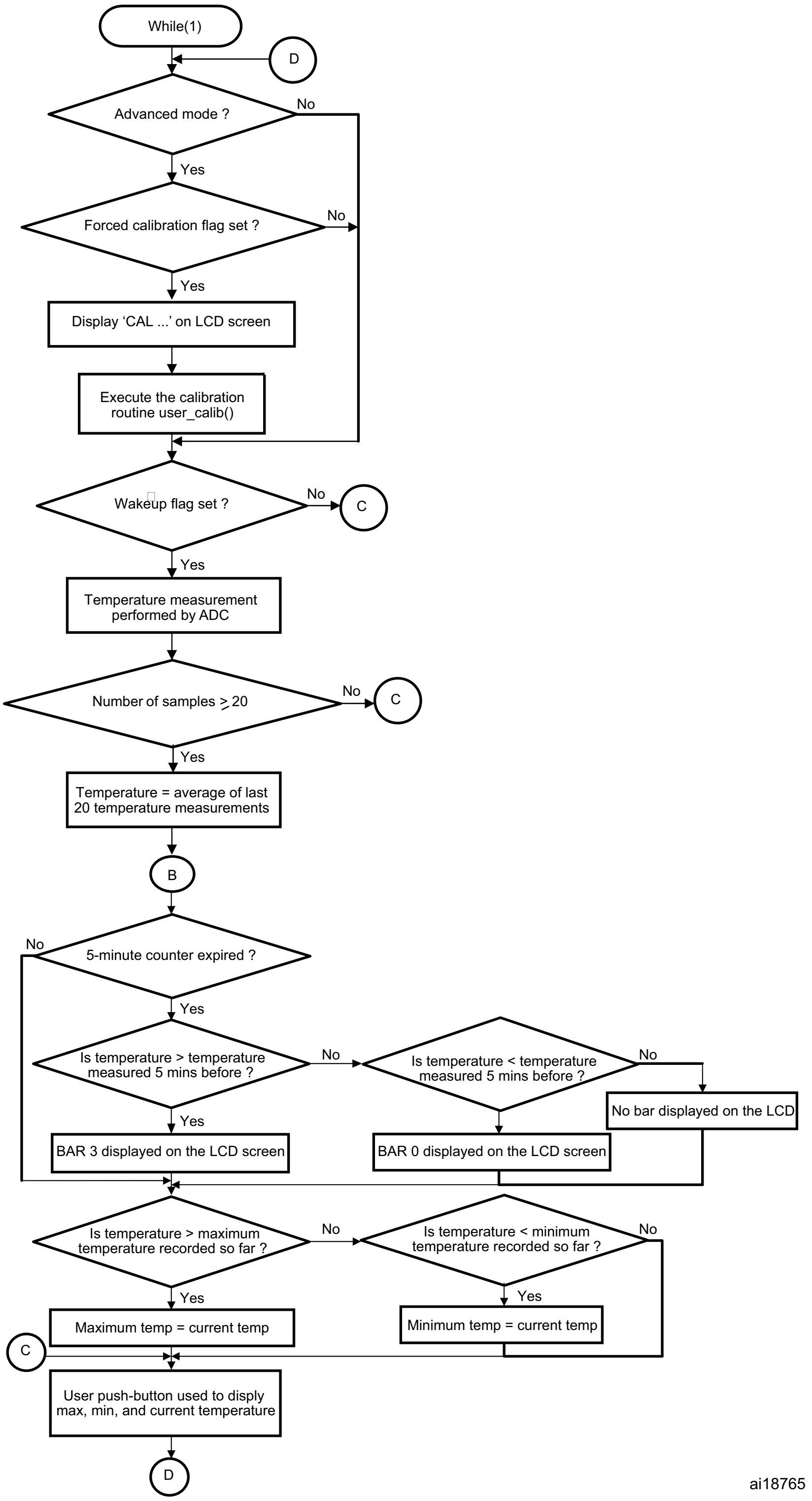


3.2.3 Infinite routine flowchart

This routine is continuously executed to manage application behavior as follows:

- Checks that ‘forced’ calibration is performed (in Advanced mode).
- Measures the temperature and displays it on the LCD screen.
- Manages the LCD bar graph to display the temperature variation over the last five minutes (see *Section 2.4: Temperature sensor application principle* for details).
- Manages the user push-button in order to alternatively display the current temperature, and the minimum and maximum temperatures recorded since the STM8L-Discovery power-up.

Figure 7. Infinite routine flowchart



4 Conclusion

The built-in TS in STM8L15xxx devices can be used for many applications requiring a measurement of temperature.

The example application provided in this application note describes how to configure the TS peripheral, and includes a description of the calibration technique. This example is a starting point for users who want to develop their own TS applications.

5 Revision history

Table 2. Document revision history

Date	Revision	Changes
12-Oct-2012	1	Initial release