Step 4: Use of sensors on B-L475E-IOT01A

Target description
The purpose of this tutorial is to explain how to measure with sensors available in the STM32L4 Discovery kit. Configuration for the temperature sensor of B-L475E-IOT01A is described step by step. After this tutorial, you will be able to collect values using the sensors available on B-L475E-IOT01A board.

The appendix of this tutorial provides guidelines on how to port an AC6 example to TrueSTUDIO.

Prerequisites
You have gone through:
Tutorial 1: Tools installation and first test
Tutorial 3: UART and new board introduction

Hardware
- Discovery kit IoT node (B-L475E-IOT01A), available on www.st.com/en/evaluation-tools/b-l475e-iot01a.html
- Standard-A-to-Micro-B USB cable

Literature
- UM2153: Discovery kit for IoT node, multi-channel communication with STM32L4
- UM1884: Description of STM32L4/L4+ HAL and low-layer drivers
- UM1859: Getting started with the X-CUBE-MEMS1 motion MEMS and environmental sensor software expansion for STM32Cube
- Getting started video: https://www.youtube.com/watch?v=6eUqxjBL_wI

Stages
- Use of sensors on B-L475E-IOT01A (page 2)
  - Hardware description (page 2)
  - Example: get temperature values using HTS221 sensor and display them on a terminal (page 3)
- Appendix: Porting an AC6 example to TrueSTUDIO (page 8)
  - Overview of the selected example and hardware (page 8)
  - Firmware execution steps: focus on porting the example to TrueSTUDIO (page 9)
HARDWARE DESCRIPTION

Main sensors available in the STM32L4 Discovery kit IoT node (B-L475E-IOT01A) are:

- Capacitive digital sensor for relative humidity and temperature (HTS221)
- 260-1260 hPa absolute digital output barometer (LPS22HB)
- 3D accelerometer and 3D gyroscope (LSM6DSL)
- High-performance 3-axis magnetometer (LIS3MDL)

- **HTS221**: capacitive digital sensor for relative humidity and temperature
- **LPS22HB**: absolute pressure sensor which functions as a 260-1260 hPa digital output barometer
- **LSM6DSL**: 3D digital accelerometer and 3D digital gyroscope
- **LIS3MDL**: ultra-low-power high performance three-axis magnetic sensor
EXAMPLE: GET TEMPERATURE VALUES USING HTS221 SENSOR AND DISPLAY THEM ON A TERMINAL (1/5)

The purpose of this section is to explain step by step how to interface with the HTS221 sensor to get temperature values and display them on a terminal.

1: Create a working project with

The starting point is the project generated with STM32CubeMX described in the previous tutorial: Follow steps described there and call the generated project L4_IOT_Sensors.

2: Copy BSP drivers to your project

The BSP (Board Support Package) drivers are available in the STM32CubeL4 package. It provides APIs corresponding to the hardware components of a board. The last version of STM32CubeL4 package is downloaded by default in STM32CubeMX repository (C:\Users\user_name\STM32Cube\Repository\STM32Cube_FW_L4_V1.11.0).

BSP location and content in the tree:

Here are the steps to follow in order to copy the BSP drivers into your project:

- Copy the STM32CubeL4/Drivers/BSP/B-L475E-IOT01 folder
- In the generated project, create a folder L4_IOT_Sensors/Drivers/BSP. Paste the copied folder there.
- Copy the STM32CubeL4/Drivers/BSP/Components folder. Paste it under L4_IOT_Sensors/Drivers/BSP.

NOTE
STM32CubeL4 used version is 1.11.0, but it can increase over time.
EXAMPLE: GET TEMPERATURE VALUES USING HTS221 SENSOR AND DISPLAY THEM ON A TERMINAL (2/5)

- Optional cleanup of working directory: as only HTS221 temperature sensor is used, some other files and folders already copied to the working directory may be removed
  - Under L4_IOT_Sensors\Drivers\BSP\B-L475E-IOT01, keep only the following files:
    - stm32l475e_iot01.c
    - stm32l475e_iot01.h
    - stm32l475e_iot01_tsensor.c
    - stm32l475e_iot01_tsensor.h
  - Under L4_IOT_Sensors\Drivers\BSP\Components, keep only the following folders:

3: Support BSP in TrueSTUDIO workspace

After being copied, the added folders appear automatically in the TrueSTUDIO workspace:
EXAMPLE: GET TEMPERATURE VALUES USING HTS221 SENSOR AND DISPLAY THEM ON A TERMINAL (3/5)

4: Update include paths

Update paths to support new header files:

- **Select the relevant project from « Project Explorer » perspective:**

![Project Explorer Screenshot](image)

- **From Project menu or File menu, go to Properties > C/C++ Build > Settings > Tool Settings > C Compiler > Directories**

- Click on ![Path Include Button](image) to include new paths

- Add “./Drivers/BSP/B-L475E-IOT01” and “./Drivers/BSP/Components/hts221” paths

The following screenshot summarizes the steps to follow:

![Configuration Settings Screenshot](image)
EXAMPLE: GET TEMPERATURE VALUES USING HTS221 SENSOR AND DISPLAY THEM ON A TERMINAL (4/5)

5: Update source files

The only file to modify is main.c, as follow:

- Include the needed header files: stm32l475e_iot01.h, stm32l475e_iot01_tsensor.h and math.h

```c
/* USER CODE BEGIN Includes */
#include "stm32l475e_iot01.h"
#include "stm32l475e_iot01_tsensor.h"
#include <math.h>
/* USER CODE END Includes */
```

- Add private values to be used to get temperature values and display messages on the terminal:

```c
/* USER CODE BEGIN PV */
/* Private variables--------------------------------------------------------*/
float temp_value = 0; // Measured temperature value
char str_tmp[100] = ""; // Formatted message to display the temperature value
uint8_t msg1[] = "****** Temperature values measurement ******\n\n";
uint8_t msg2[] = "=====> Initialize Temperature sensor HTS221 \r\n";
uint8_t msg3[] = "=====> Temperature sensor HTS221 initialized \r\n";
/* USER CODE END PV */
```

- Display messages on the terminal and initialize the temperature sensor HTS221:

```c
/* USER CODE BEGIN 2 */
HAL_UART_Transmit(&huart1,msg1,sizeof(msg1),1000);
HAL_UART_Transmit(&huart1,msg2,sizeof(msg2),1000);
BSP_TSENSOR_Init();
HAL_UART_Transmit(&huart1,msg3,sizeof(msg3),1000);
/* USER CODE END 2 */
```

- In the “while (1)” loop, read temperature value, format it and then display the message with measured value on the terminal:

```c
/* USER CODE BEGIN 3 */
temp_value = BSP_TSENSOR_ReadTemp();
tmpInt1 = temp_value;
float tmpFrac = temp_value - tmpInt1;
tmpInt2 = trunc(tmpFrac * 100);
snprintf(str_tmp,100," TEMPERATURE = %d.%02d\n\r", tmpInt1, tmpInt2);
HAL_UART_Transmit(&huart1,( uint8_t * )str_tmp,sizeof(str_tmp),1000);
```
EXAMPLE: GET TEMPERATURE VALUES USING HTS221 SENSOR AND DISPLAY THEM ON A TERMINAL (5/5)

6: Compile and run the example

- Click on Build button to compile the project.
- Click on Debug button to run the software.
- Click on Console button to open a console, select Data bits: 8 and click OK.
- TrueSTUDIO opens Debug perspective. Click on Resume button to execute the code. Terminal window displays the initialization message preceding measured temperature values:

![Terminal window showing initialization message and temperature values]

Now you are able to:

- Build your own project to measure temperature values with sensor embedded on the B-L475E-IOT01A
- Add BSP components to an STM32CubeMx generated project
- Extend the use of the board to sensors other than HTS221, to make environmental measurements
Appendix: Porting an AC6 example to TrueSTUDIO

THE EXAMPLE
The example to be used in this appendix is DataLogTerminal located under:

STM32CubeExpansion_MEMS1_V4.4.1\Projects\MultiExamples\IKS01A2\DataLogTerminal.

It is recommended to put the package under C: in order to avoid compilation errors later (because of long paths).

THE HARDWARE USED
The X-NUCLEO-IKS01A2 is a motion MEMS and environmental sensor expansion board for the STM32 64-pin Nucleo. It interfaces with NUCLEO-L476RG via I²C-bus pins.

TIPS & TRICKS

More recent versions of this package may be available over time from same link.

X-NUCLEO-IKS01A2 SENSORS

- HTS221: capacitive digital sensor for relative humidity and temperature
- LPS22HB: absolute pressure sensor which functions as a 260-1260 hPa digital output barometer
- LSM6DSL: 3D digital accelerometer and 3D digital gyroscope
- LSM303AGR: 3D digital linear acceleration sensor and 3D digital magnetic sensor
FIRMWARE EXECUTION STEPS: FOCUS ON PORTING THE EXAMPLE TO TrueSTUDIO (1/3)

1: Hardware setup
   a. Extend your Nucleo board with the X-NUCLEO-IKS01A2 shield using the Arduino connectors
   b. Connect the board with its shield to your PC

2: Example details
   A description of the DataLogTerminal example available under STM32CubeExpansion_MEMS1_V4.4.1 \Projects\Multi\Examples\IKS01A2\DataLogTerminal is provided in the readme.txt file:

   @par Example Description

   Main function is to show how to use sensor expansion board to send sensor data from a Nucleo board using UART to a connected PC or Desktop and display it on generic applications like TeraTerm.
   After connection has been established:
   - the user can view the data from various on-board environment sensors like Temperature, Humidity, and Pressure.
   - the user can also view data from various on-board MEMS sensors as well like Accelerometer, Gyroscope, and Magnetometer.

3: Porting the example to TrueSTUDIO
   To import the example based on SW4STM32 and dedicated to the NUCLEO-L476RG (SW4STM32 \STM32L476RG-Nucleo\STM32L4xx-Nucleo-DataLogTerminal) into TrueSTUDIO IDE:
   - Click on .project file to open it. The following message is displayed:
FIRMWARE EXECUTION STEPS: FOCUS ON PORTING THE EXAMPLE TO TrueSTUDIO (2/3)

- When clicking on OK, the following message pops up:

![Project converter]

The project STM32L4xx-Nucleo-DateTimeTerminal was converted without problems. Some manual configuration may still be required.

- Click OK
- Select the relevant project from «Project Explorer» perspective:

![Project Explorer]

4: Compiling and running the example

- Click on Build button to compile the project.
- Click on Configure Debug button.
- In the Debug Configuration window that pops up, make sure that the selected Debug probe is ST-LINK:

![Debug Configuration]

- From the same window, click on Debug, or click on Debug button to run the software.
Click on Console button to open a console, select Data bits: 8 and click OK.

Click on Resume button to execute the code. Terminal window displays the measured values using the sensors available in the shield X-NUCLEO-IKS01A2. It is also possible to display them using a hyperterminal application like Tera Term.

Open terminal and apply the following settings:

The values measured by the sensors of X-NUCLEO-IKS01A2 are displayed in the hyperterminal window as follows: