STM32CubeU0 STM32U083C-DK demonstration firmware

Introduction

STM32Cube is an STMicroelectronics original initiative to improve designer productivity significantly by reducing development effort, time, and cost. STM32Cube covers the whole STM32 portfolio.

STM32Cube includes:

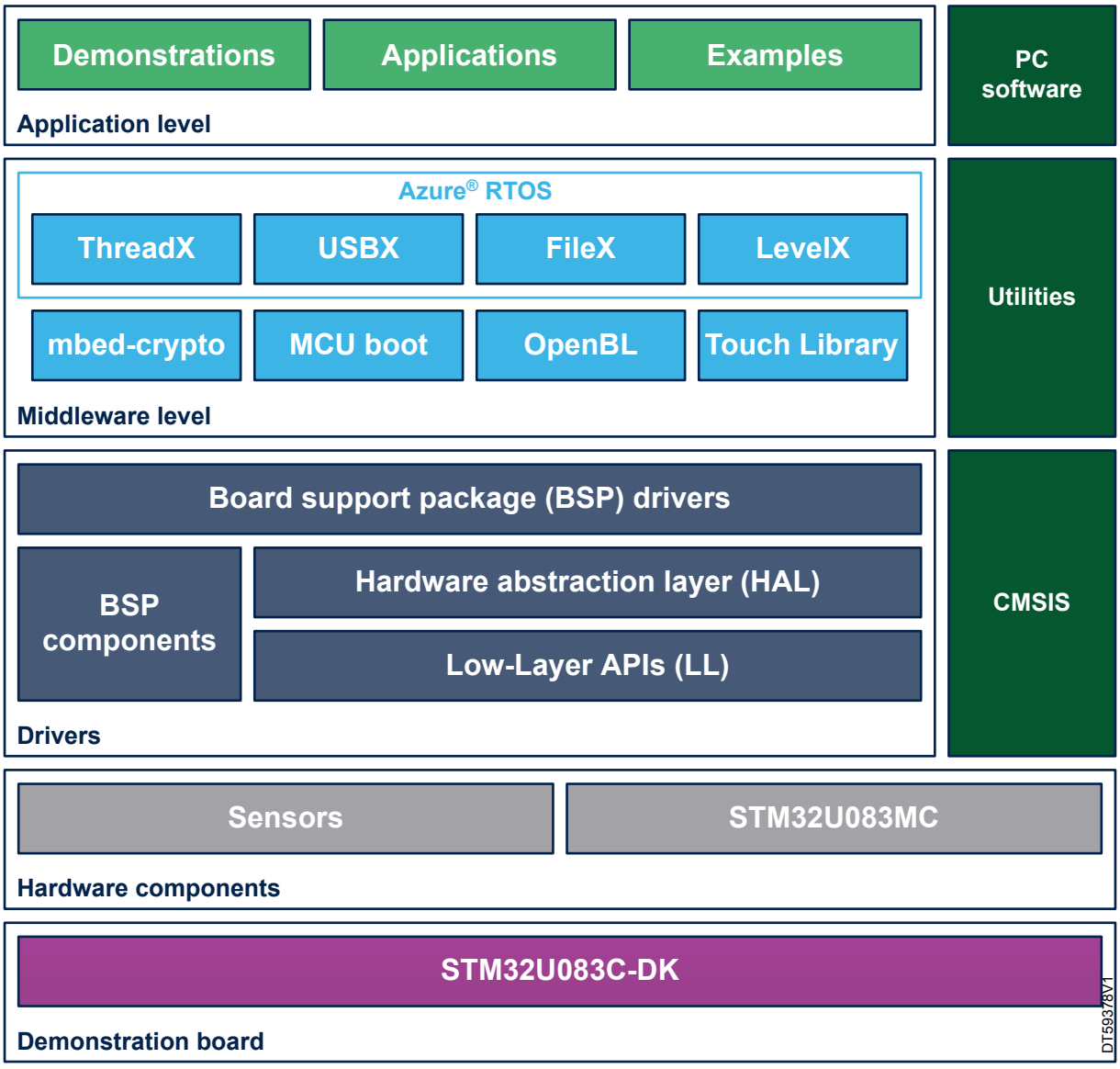
- A set of user-friendly software development tools to cover project development from conception to realization, among which are:
  - STM32CubeMX, a graphical software configuration tool that allows the automatic generation of C initialization code using graphical wizards
  - STM32CubeIDE, an all-in-one development tool with peripheral configuration, code generation, code compilation, and debug features
  - STM32CubeCLT, an all-in-one command-line development toolset with code compilation, board programming, and debug features
  - STM32CubeProgrammer (STM32CubeProg), a programming tool available in graphical and command-line versions
  - STM32CubeMonitor (STM32CubeMonitor, STM32CubeMonPwr, STM32CubeMonRF, STM32CubeMonUCPD) powerful monitoring tools to fine-tune the behavior and performance of STM32 applications in real time
- STM32Cube MCU and MPU Packages, comprehensive embedded-software platforms specific to each microcontroller and microprocessor series (such as STM32CubeU0 for the STM32U0 series), which include:
  - STM32Cube hardware abstraction layer (HAL), ensuring maximized portability across the STM32 portfolio
  - STM32Cube low-layer APIs, ensuring the best performance and footprints with a high degree of user control over hardware
  - A consistent set of middleware components such as Microsoft® Azure® RTOS, USB Device, TouchSensing, and OpenBootloader
  - All embedded software utilities with full sets of peripheral and applicative examples
- STM32Cube Expansion Packages, which contain embedded software components that complement the functionalities of the STM32Cube MCU and MPU Packages with:
  - Middleware extensions and applicative layers
  - Examples running on some specific STMicroelectronics development boards

The STM32CubeU0 Discovery board demonstration firmware is built around almost the whole STM32 capability to offer a large scope of usage based on the STM32Cube HAL BSP and utility components.

The STM32CubeU0 Discovery board demonstration firmware supports STM32U0xx devices and runs on the STM32U083C-DK Discovery board.

Within STM32CubeU0, both the HAL and LL APIs are production-ready, developed in compliance with MISRA C®:2012 guidelines and the elimination of possible runtime errors with Synopsys® Coverity® static analysis tool. Reports are available on demand.

Figure 1. STM32CubeU0 MCU Package architecture



1 General information

The STM32CubeU0 demonstration firmware runs on the STM32U083C-DK Discovery board featuring the STM32U083MC microcontroller based on the Arm® Cortex®-M0+ core.

Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

2 Getting started with the demonstration

2.1 Hardware requirements

The hardware requirements for running the demonstration application are as follows:

- The STM32U083C-DK Discovery board. Refer to Figure 2 and the user manual *Discovery kit with STM32U083MC MCU* (UM3292) for the Discovery board description.
- A USB Type-C® cable to power the STM32 Discovery board from the ST-LINK USB Type-C® connector (CN1).

The STM32U083C-DK Discovery board helps you discover the ultra-low-power functions and audio/graphics capabilities of the STM32U0 series. It offers everything beginners and experienced users need to get started quickly and develop applications easily.

Based on an STM32U083MC MCU, the STM32U083C-DK Discovery board features an embedded ST-LINK/V2 debug tool interface, an Idd current measurement panel, segmented LCD, LEDs, joystick, and two USB Type-C® connectors.

2.2 Hardware configuration to run the demonstration firmware

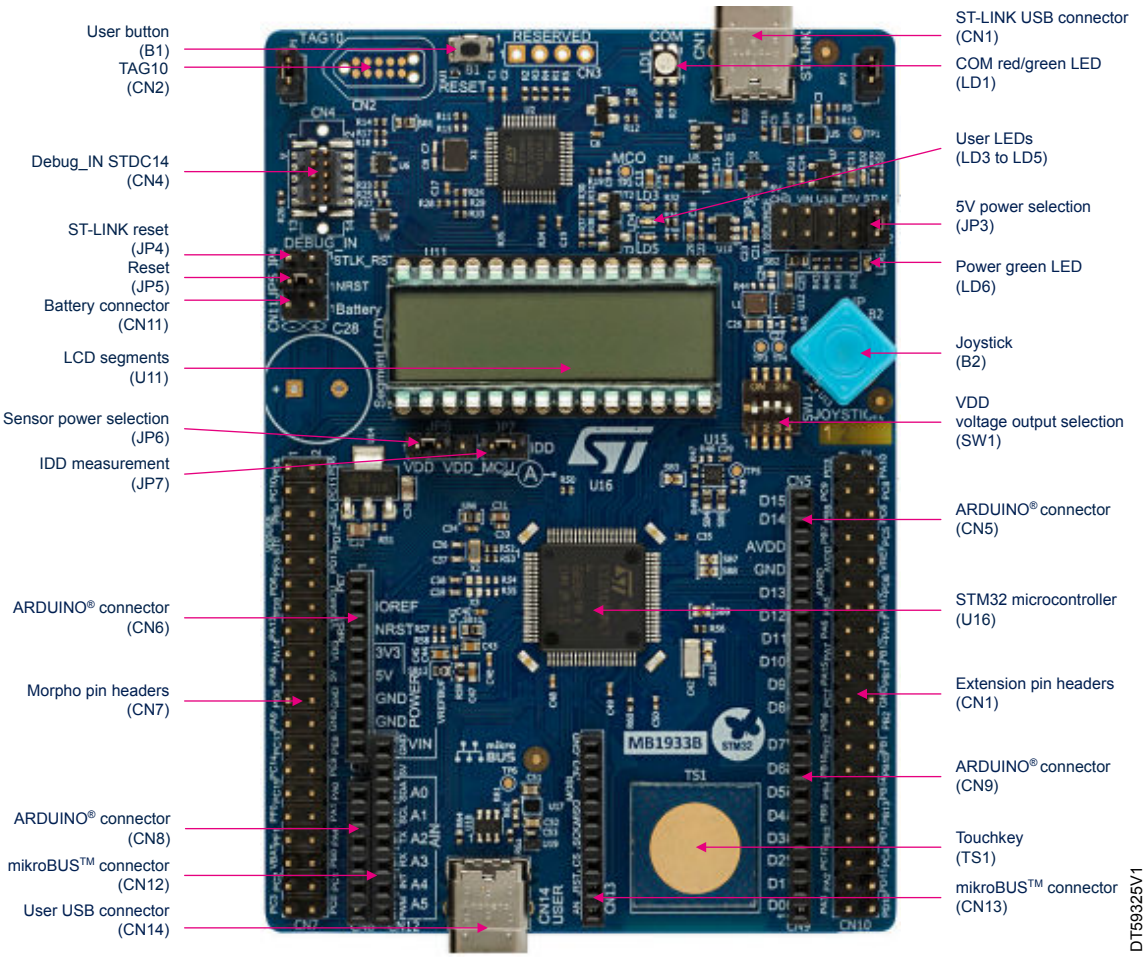
Table 1. Jumper configuration

Jumper/connector number	Position(1)
JP3	[1-2] STLK
JP4	OFF
JP5	ON
JP6	[1-2] VDD
JP7	ON

1. *Position 1 corresponds to the jumper side with a dot marking.*

Refer to the user manual *Discovery kit with STM32U083MC MCU* (UM3292) for a complete description of the jumper settings.

Figure 2. **STM32U083C-DK Discovery board**

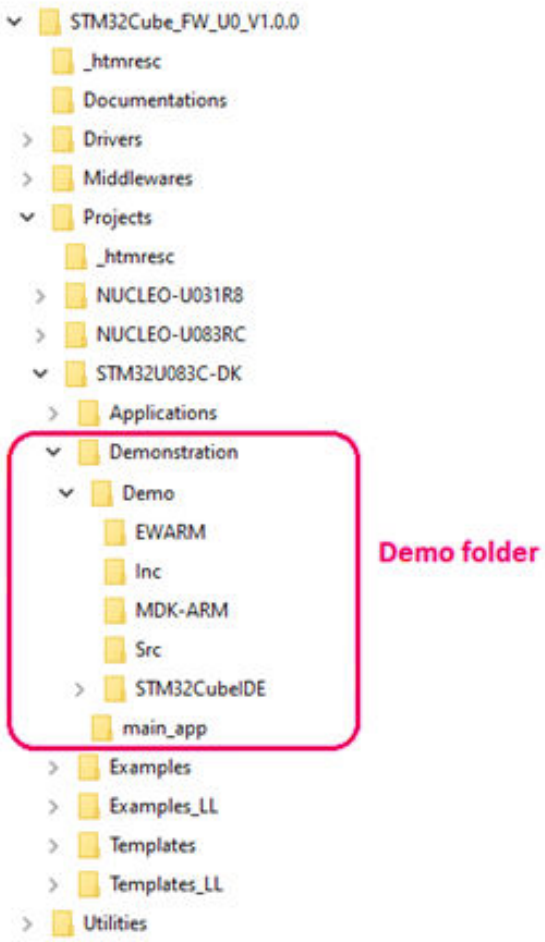


3 Demonstration firmware package

3.1 Demonstration repository

The STM32CubeU0 demonstration firmware for the STM32U083C-DK Discovery board is provided within the STM32CubeU0 firmware package as shown in Figure 3.

Figure 3. Folder structure



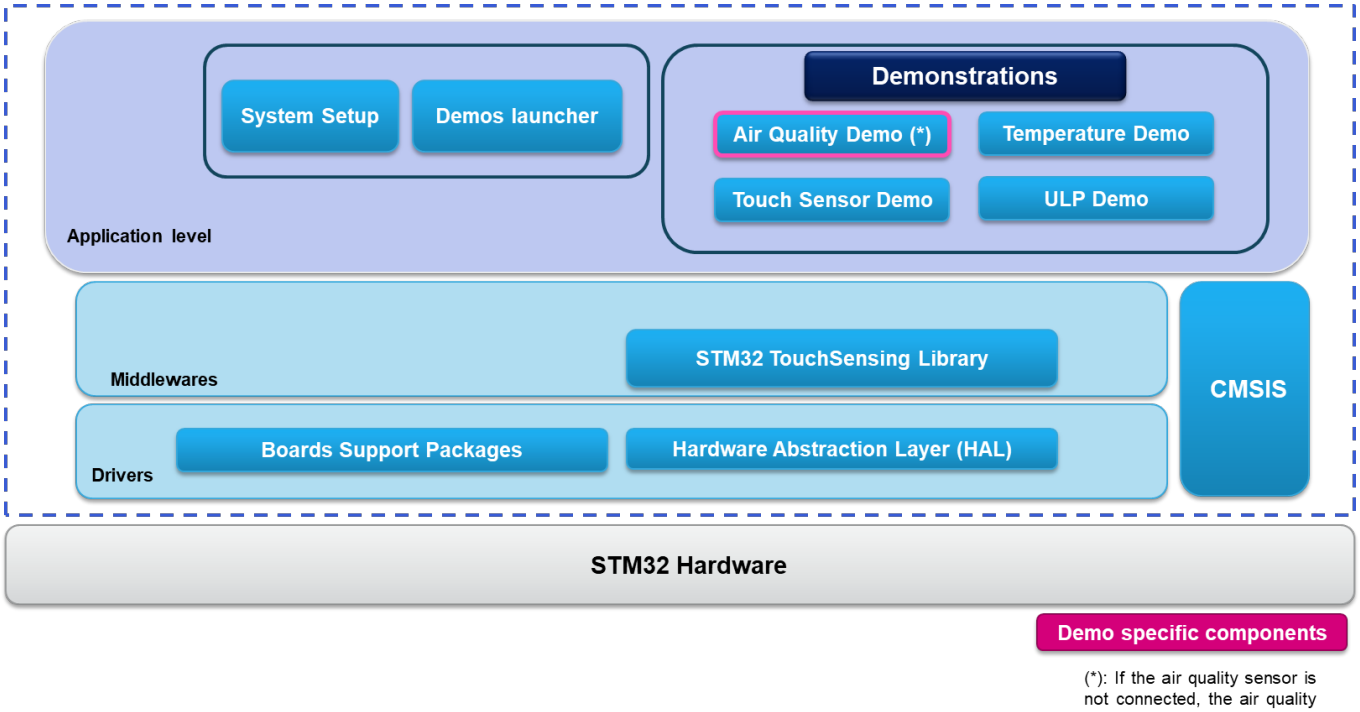
The demonstration sources are located in the projects folder of the STM32Cube package for each supported board. The sources are divided into two groups described as follows:

- **Main_App**: It contains the top-level source files for the main application and application modules. It also contains all middleware components and HAL configuration files.
- **Demo**: It contains the main files and **project settings** (a folder per toolchain containing the project settings and the linker files).

3.2 Demonstration architecture overview

The STM32CubeU0 demonstration firmware for the STM32U083C-DK Discovery board consists of a central kernel based on a set of firmware and hardware services offered by the STM32Cube middleware, the Evaluation board drivers, and a set of modules mounted on the kernel and built in a modular architecture. Each module can be reused separately in a standalone application. A specific API, which provides access to all common resources and facilitates the addition of new modules as shown in Figure 4 manages the full set of modules.

Figure 4. Demonstration architecture overview

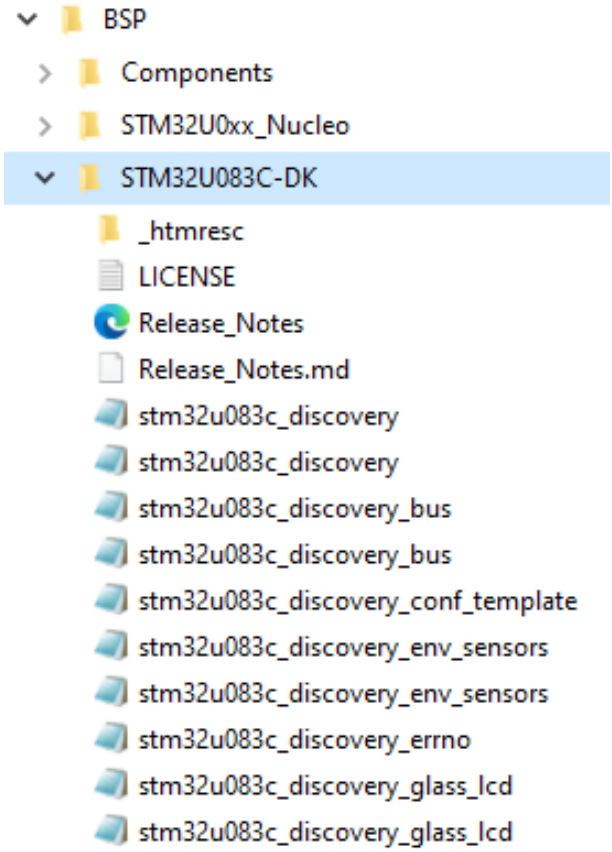


(*): If the air quality sensor is not connected, the air quality application is not displayed.

3.3 STM32U083C-DKDiscovery board BSP

Board drivers are available within the `stm32u083c_discovery_XXX.c` and `stm32u083c_discovery_XXX.h` files (refer to Figure 5), implementing the board capabilities and the bus link mechanism for the board components, such as LEDs, buttons, audio, LCD, and touch-sensing.

Figure 5. Discovery BSP structure



Dedicated BSP drivers control the components present on the STM32U083C-DK Discovery board. These are:

- The bus in `stm32u083c_discovery_bus.c` and `stm32u083c_discovery_bus.h`
- The temperature sensor environment in `stm32u083c_discovery_audio.c` and `stm32u083c_discovery_audio.c`
- The LCD glass in `stm32u083c_discovery_glass_lcd.c` and `stm32u083c_discovery_glass_lcd.h`

4 Demonstration functional description

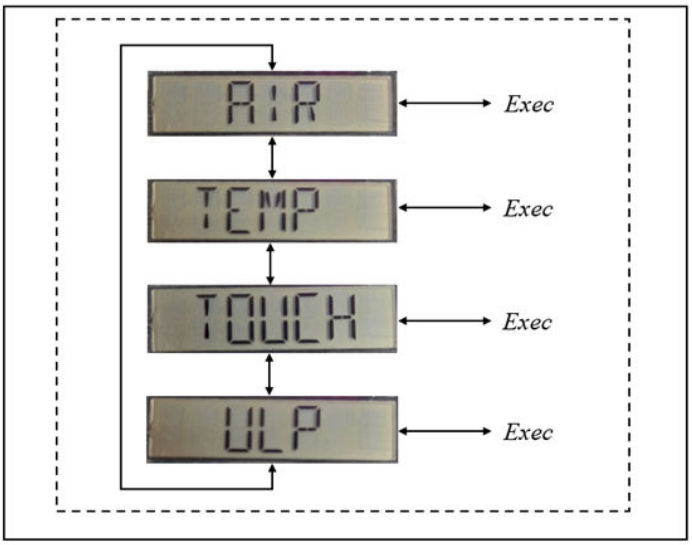
4.1 Overview

After powering up the STM32U083C-DK Discovery board, the welcome message "STM32U083C-DISCOVERY DEMO" appears on the LCD screen and the first main menu of application items is displayed.

4.2 Main menu

Figure 6 shows the main menu application tree with navigation possibilities:

Figure 6. Demonstration top menu



4.3 Navigation menu

Use the UP, DOWN, RIGHT, and LEFT joystick directions to navigate between the main menu and submenu items. To enter a submenu and launch the `Exec` function, press the SEL button. The SEL button refers to the action of vertically pressing the top of the joystick as opposed to pressing the UP, DOWN, RIGHT, and LEFT keys horizontally. The basic functions of the joystick buttons are defined as follows:

Table 2. Joystick key functions

Joystick key	Function
DOWN	Go to the next menu/submenu item
UP	Go to the previous menu/submenu item
RIGHT/SEL	Select the demonstration application/submenu item
LEFT	Stop and exit the demonstration application/submenu item

4.4 Modules and APIs

4.4.1 Air quality demonstration

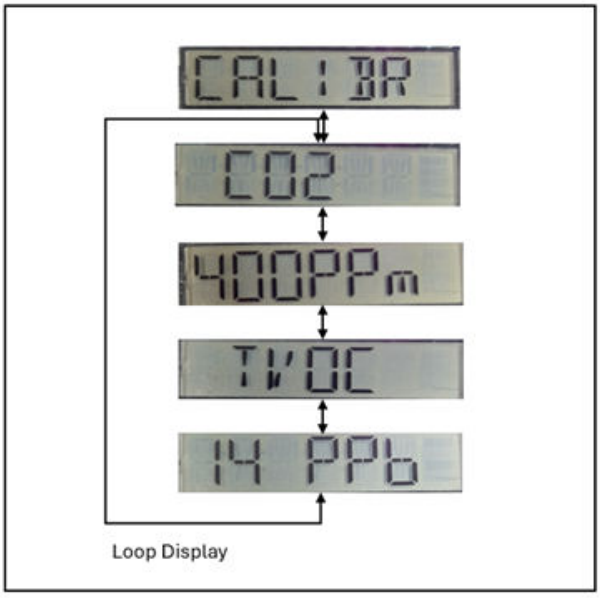
The MIKROE-2953 sensor module measures air quality. It uses an I^2C-based MICROE (CCS811) sensor, which can be easily connected to the board via CN12 and CN13.

Users can loop through the CO2 and TVOC measurements on the LCD glass screen. The application displays messages such as NORMAL/POLLUTION/HIGH POLLUTION to indicate pollution levels based on threshold values.

To switch to another demo module, press the LEFT joystick key for five seconds.

If the air quality sensor is not connected, the air quality application/demonstration is not displayed.

Figure 7. Air quality demonstration display



4.4.2 Temperature sensor demonstration

The temperature sensor module measures temperature.

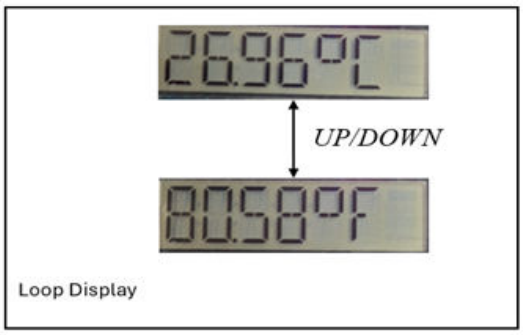
This is achieved by using an I^2C-based temperature sensor integrated into the STM32U083C-DK Discovery board.

The application continuously displays temperature measurements on the LCD glass screen.

Users can switch between celsius and fahrenheit formats using the UP/DOWN keys of the joystick

To switch to another demonstration module, press the LEFT joystick key for five seconds.

Figure 8. Temperature sensor demonstration display



4.4.3 Touch sensor demonstration

The touch-sensing module enables the detection of a contact on the touch-sensor TSC1 button after a low-power phase, using the integrated comparator device to reduce power consumption.

In this specific STM32U0xx series, some of the touch-sensing I/O pins are interconnected with the comparator module, giving the option of changing the sensing voltage level.

By changing this voltage level, physical contact can be detected earlier, depending on the value of the comparator input.

This means that the lower the level, the less time it takes to reach it, and therefore the shorter the acquisition cycle.

In other words, you detect physical contact more quickly.

The inputP of the comparator is connected to the TS1 button I/O group. The inputM is connected to the available VREF level (1/4 Vref, 1/2 Vref, 3/4 Vref, and Vref).

In this application, inputP is connected to TSC_G6_IO1 (COMP_INPUT_PLUS_IO4) and inputM to VREFINT.

With inputM at the VREF level, the threshold for touch detection is set for the Discovery board by the `tsl_user_SetThresholds()` function.

The `tsl_user_SetThresholds()` function sets the threshold according to the inputM value of the comparator.

Certain limitations might arise if the inputM level is too low. If it is too low, the touch-sensing middleware has less range, and the measurement might therefore approach the noise level.

The user needs to be cautious at this stage.

The touch-sensing module software consists of several stages:

First, the main module initializes the touch device, comparator, RTC, and touch-sensing middleware via `MX_TSC_Init()`, `MX_COMP2_Init()`, `MX_RTC_Init()`, and `MX_TOUCHSENSING_Init()` respectively.

Next, the touch-sensing/touch-wake-up module scrolls through the "RUN MODE" message twice, then initiates TSC calibration, which lasts about five seconds.

Finally, after startup, the RTC wakes up the MCU every 250 ms, in a loop while the touch-sensing/touch-wake-up module handles detection and nondetection in this way:

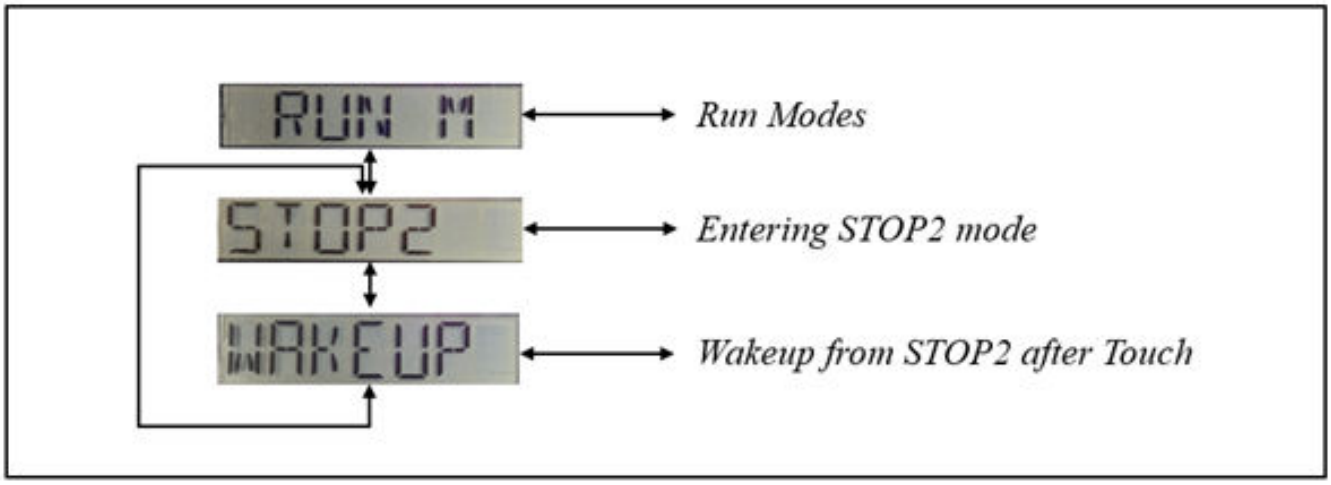
- If no contact is detected: The module displays the message "ENTER STOP2 MODE", then switches to low-power stop 2 mode. It remains in low-power mode until the RTC wakes up to determine whether or not a contact has been detected. If no contact is detected, the module returns to low-power shutdown 2 mode.
- If contact is detected: The module displays the message "WAKEUP TOUCH DETECTED" for five seconds. It returns to low-power shutdown 2 mode until RTC wakes up.

The TM32U083C-DK's LEDs can be used to monitor the status of touch detection:

- LED4 is ON when a touch is detected.
- LED4 is OFF when STM32U083C-DK enters low-power shutdown 2 mode.

To switch to another demonstration module, the user can press the left joystick key for five seconds.

Figure 9. Touch sensor demonstration display



4.4.4 ULP demonstration

Users can switch between ULP modes using the joystick UP/DOWN keys. The joystick's RIGHT or SEL button is used to select the ULP mode.

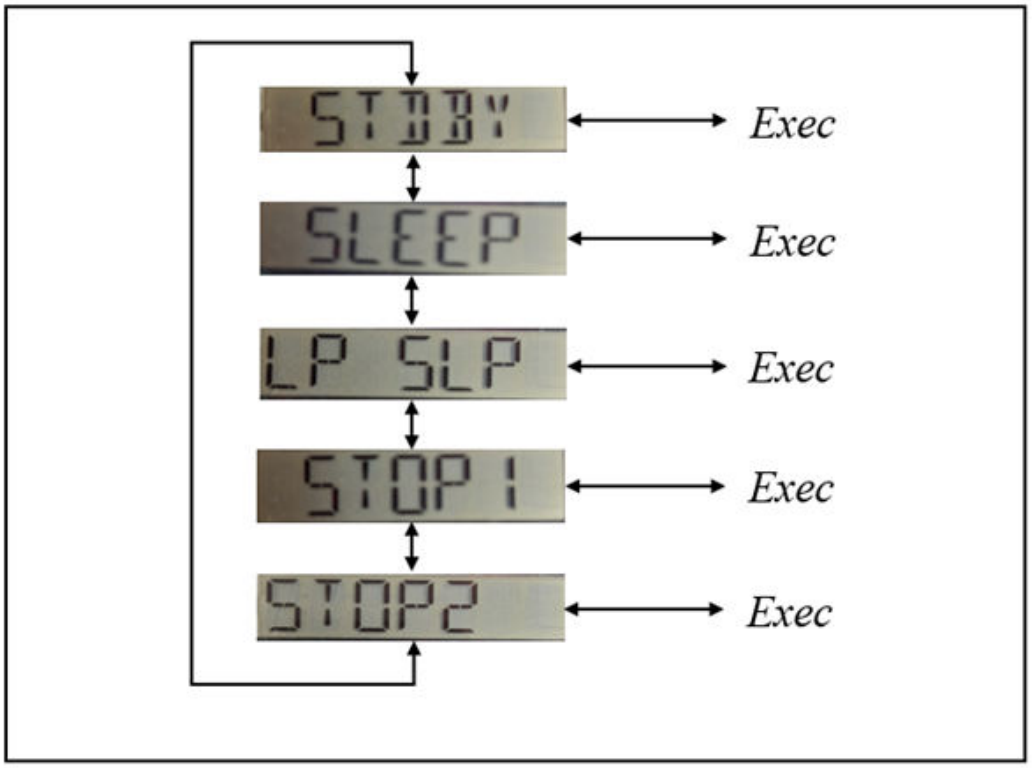
Once ULP mode has been selected, the system remains in ULP mode for around 33 seconds when exiting ULP mode.

If users wish to exit shutdown mode before around 33 seconds, they can use the joystick "SEL" button. After selecting ULP mode, the joystick "SEL" button is switched to push-button mode.

When entering ULP mode, the LCD glass shows typical power consumption (no built-in metering).

The supported ULP modes are Standby, Sleep LP sleep, Stop1, and Stop2 modes.

Figure 10. ULP demonstration display



5 Demonstration firmware settings

5.1 Clock control

The following clock configurations are used in the demonstration firmware:

- SYSCLK: 48 MHz (PLL) from MSI 4 MHz (RUN voltage range 1)

The following oscillators and PLLs are used in the demonstration firmware:

- MSI (4 MHz) as PLL source clock
- LSE (32.768 kHz) as RTC clock source

5.2 Peripherals

The peripherals used in the demonstration firmware are listed in Table 3.

Table 3. Peripherals list

Used peripherals	Application/module
ADC	Joystick BSP drivers
CORTEX	NVIC services
TSC	Touch sensor application
I^2C	On-board air and temperature sensor applications
LCD	LCD glass display
EXTI	Push-button and RTC wake-up interrupt
Flash	System settings
GPIO	All applications
PWR	System application
RCC	System application and BSP drivers
RTC	System application and backup service

5.3 Interrupts/wake-up pins

The interrupts used in the demonstration firmware are listed in Table 4.

Table 4. Interrupts list

Interrupts	Application/module	Priority, subpriority (highest=0,0)
EXTI Line 2	PB application (wake-up interrupt)	1,0
RTC_WKUP_IRQn	LCD screen dimming applications	0,0
SysTick	CortexM0+ system timer	15,0
TSC_IRQn	Touch sensor application	0,0

5.4 Programming firmware application

First of all, install the ST-LINK/V2 driver available on *www.st.com*.
There are two ways of programming the STM32U083C-DK Discovery board.

5.4.1 Using a binary file

Upload the binary `STM32CubeU0_Demo_STM32U083C-DK_VX.Y.Z.hex` using your preferred in-system programming tool.

5.4.2 Using preconfigured projects

Choose one of the supported tool chains and follow the steps below:

- Open the application folder: `Projects\STM32U083C-DK\Demonstrations`.
- Chose the desired IDE project (EWARM for IAR Systems®, MDK-ARM for Keil®, or STM32CubeIDE).
- Double-click on the project file (for example `Project.eww` for EWARM).
- Rebuild all files: Go to *Project* and select *Rebuild all*.
- Load the project image: Go to *Project* and select *Debug*.
- Run the program: Go to *Debug* and select *Go*

Revision history

Table 5. **Document revision history**

Date	Revision	Changes
22-Mar-2024	1	Initial release.

Contents

List of tables

List of figures