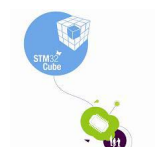## Introduction

STMCube™ initiative was originated by STMicroelectronics to ease developers' life by reducing development efforts, time and cost. STM32Cube covers STM32 portfolio.

STM32Cube Version 1.x includes:

- The STM32CubeMX, a graphical software configuration tool that allows to generate C initialization code using graphical wizards.

- A comprehensive embedded software platform, delivered per series (such as STM32CubeL0 for STM32L0 series)
    - The STM32CubeL0 HAL, an STM32 abstraction layer embedded software, ensuring maximized portability across STM32 portfolio
    - A consistent set of middleware components such as RTOS, USB, STMTouch and FatFs
    - All embedded software utilities coming with a full set of examples.

The STM32Cube L0 Discovery demonstration platform is built around the STM32Cube HAL, BSP and USB middleware components.

With dual USB connectivity, an ePaper display, a capacitive touch key, ST-LINK/V2 debugger/programmer and both STM32L0 and STM32L1 microcontrollers, this discovery kit is the best hardware to evaluate STM32 ultra-low-power solutions.

Contents

List of tables

List of figures

1 STM32CubeL0 main features

STM32CubeL0 gathers together, in a single package, all the generic embedded software components required to develop an application on STM32L0 microcontrollers. In line with the STM32Cube initiative, this set of components is highly portable, not only within the STM32L0 series but also to other STM32 series.

STM32CubeL0 is fully compatible with STM32CubeMX code generator that allows the user to generate initialization code. The package includes a low level hardware abstraction layer (HAL) that covers the microcontroller hardware, together with an extensive set of examples running on STMicroelectronics boards. The HAL is available in an open-source BSD license for user convenience.

STM32CubeL0 package features a set of middleware components with the corresponding examples. They come with very permissive license terms:

- Full USB Device stack supporting many classes (HID, MSC, CDC, Audio, DFU)
- CMSIS-RTOS implementation with FreeRTOS open source solution
- FAT File system based on open source FatFs solution
- STMTouch touch sensing solution.

A demonstration implementing all these middleware components is also provided in the STM32CubeL0 package.

The block diagram of STM32Cube is shown in *Figure 1*.

Figure 1. STM32Cube block diagram
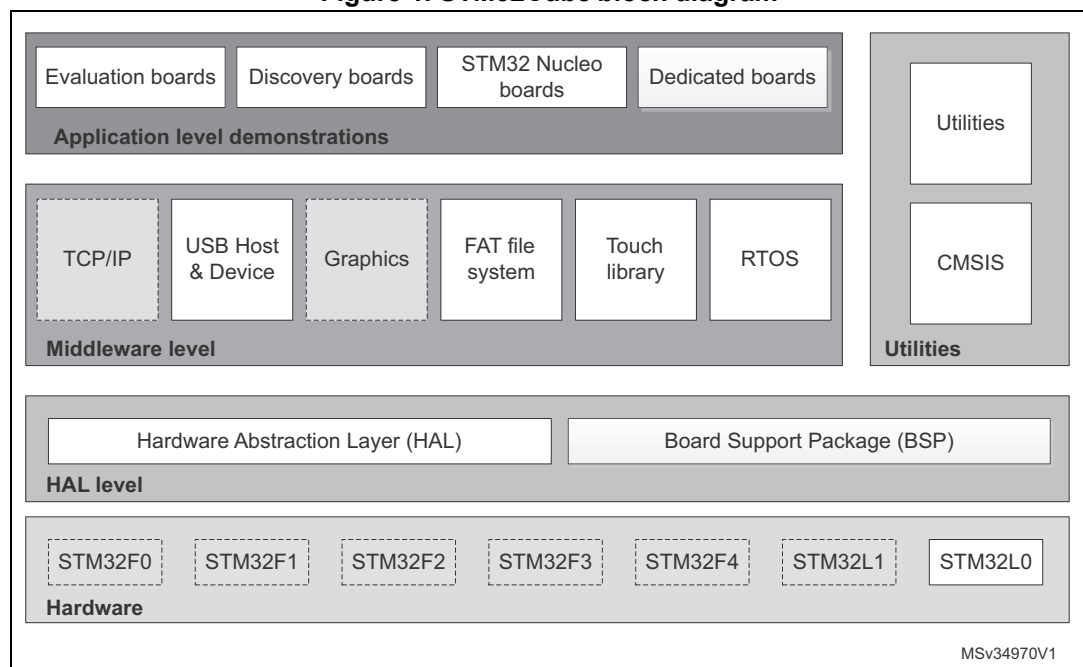
# 2 Getting started with the demonstration

2.1 Hardware requirements

The hardware requirements to start the demonstration application are as follows:

- STM32L053 Discovery board (*Figure 2*);
- One 'USB type A to Mini-B' cable to power up the STM32 Discovery board from the USB ST-LINK (USB connector CN2);
- One 'USB type A to Mini-B' cable between CN3 and the PC.

The STM32L053 Discovery kit helps you to discover the ultra-low-power features of the STM32 L0 series. It offers everything required for beginners and experienced users to get started quickly and develop applications easily.

Based on an STM32L053C8T6, it includes an ST-LINK/V2-1 embedded debug tool interface, linear touch sensor, touch keys, Idd current measurement, 2.04" E-paper display, NFC connector for PLUG-CR95HF-B board, LEDs, pushbuttons and a USB mini-B connector.

Figure 2. STM32L053 Discovery board (top and bottom views)
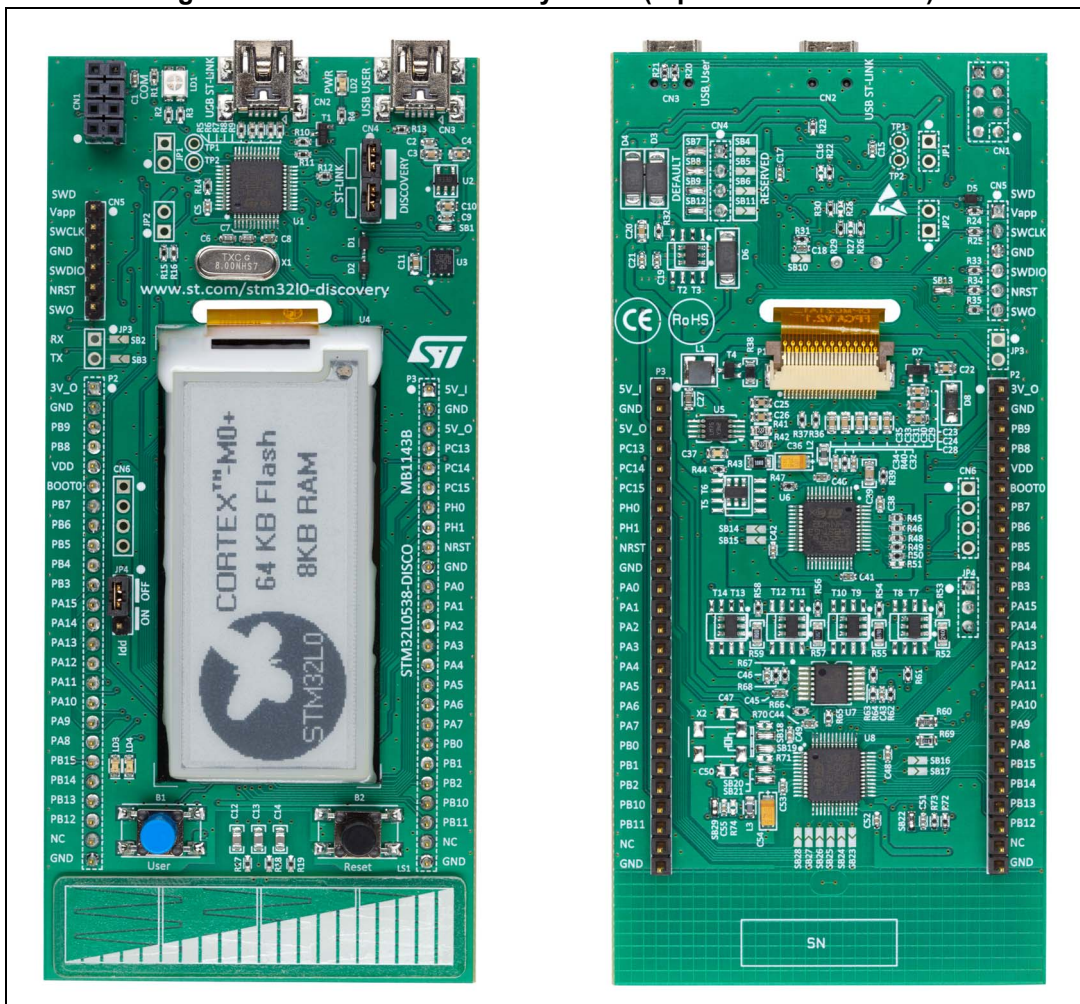
2.2 Hardware configuration

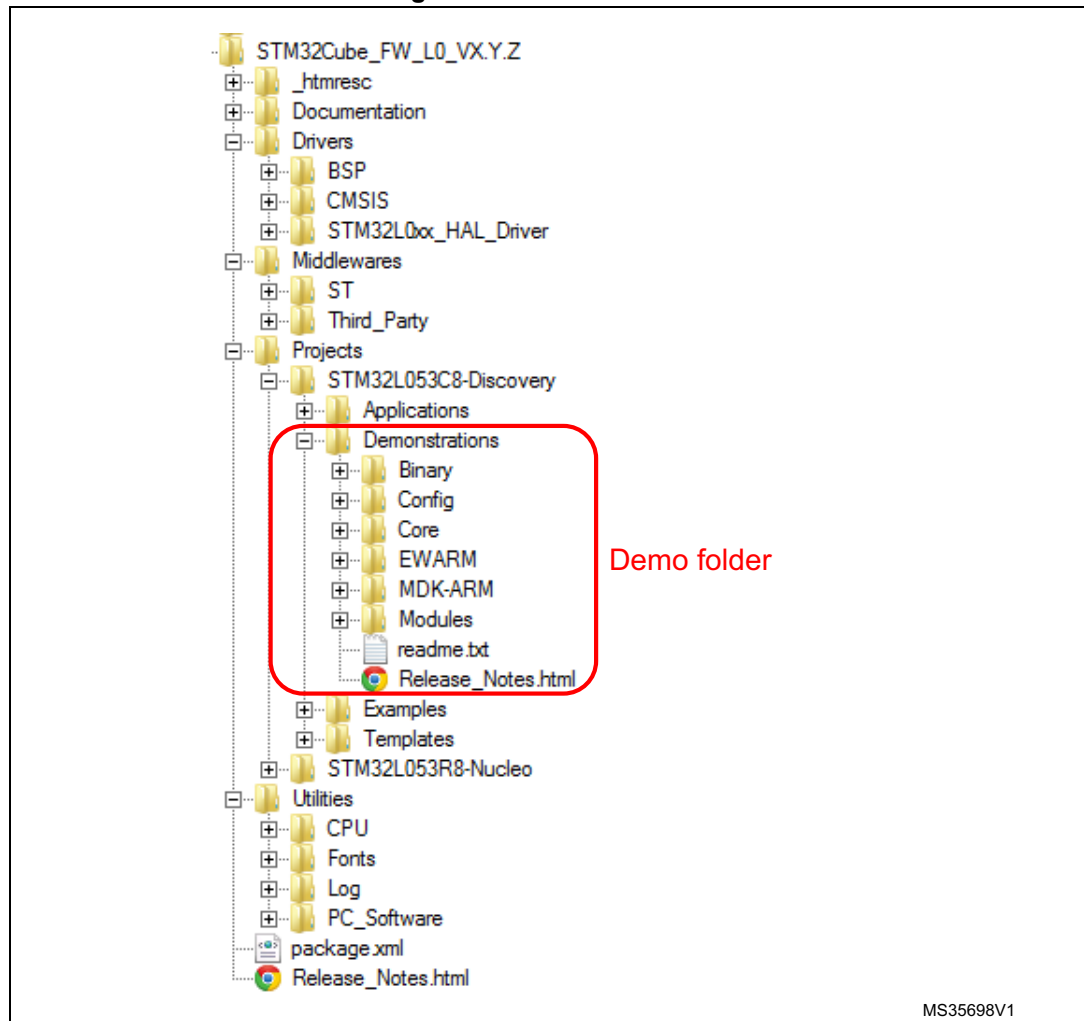Check jumpers' positions on the STM32 Discovery board as follows:

- CN4 is set on DISCOVERY mode.
- JP4 (I_{DD}) ON.

3 Demonstration firmware package

3.1 Demonstration repository

The STM32L053 Discovery demonstration is provided within the STM32CubeL0 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 the STM32L053C8 Discovery board. The sources are divided into five groups described as follows:

1. Binary: demonstration binary file in Hex format;
2. Config: contains Touch Sensor, USB and HAL configuration files;
3. Core: contains the demonstration header and source files;
4. Modules: contains source files for Idd measurement and usbHID modules;
5. Project settings: a folder per toolchain containing the project settings and the linker files.

3.2 Discovery board BSP

For each board, a set of Button and LEDs drivers is available within the *stm32l0xx_discovery.c/.h* files (see *Figure 4*), implementing the board capabilities and the bus link mechanism.

Figure 4. Discovery BSP architecture
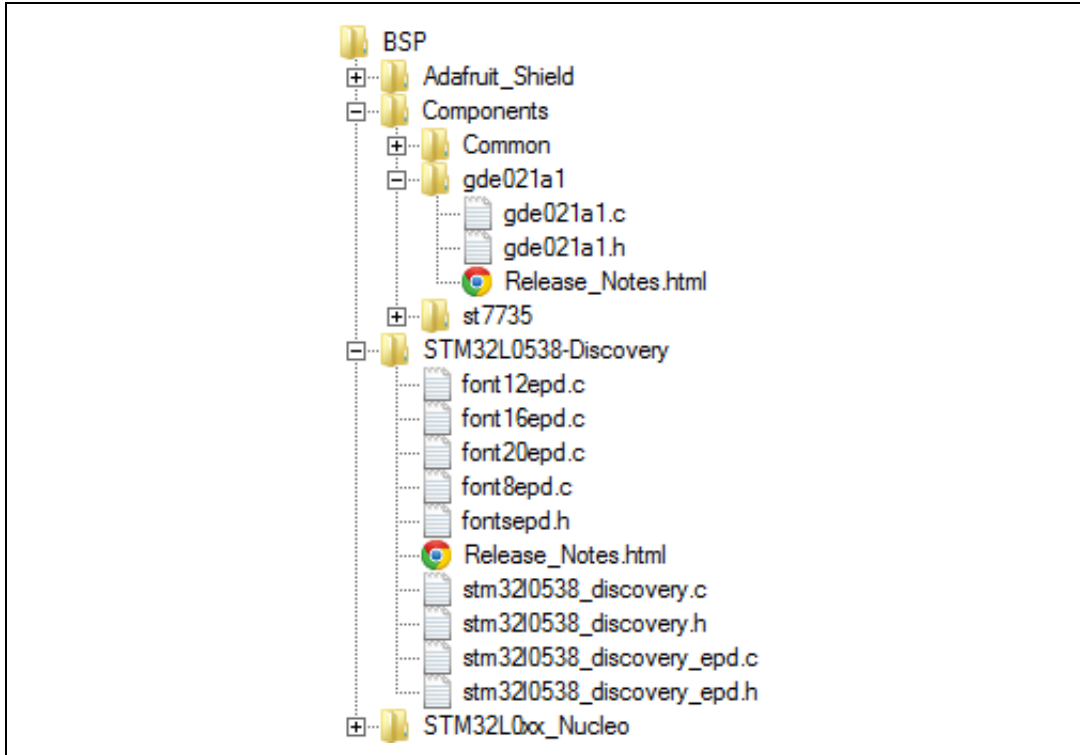
The EPD is an active Matrix Electrophoretic Display, with interface and a reference system design. The 2.04" active area contains 172x72 pixels, and has 2-bit full display capabilities. It uses 2-wire SPI to communicate with the STM32L0 chip.

The SPI peripheral is configured within the *stm32l0xx_discovery.c/.h* driver which contains also the SPI bus link mechanism and IO operations.

The EPD is controlled by a dedicated BSP EPD driver *stm32_discovery_epd.c/.h* which uses the gde021a1 component that exports in a generic way the EPD IO operations needed for its process.
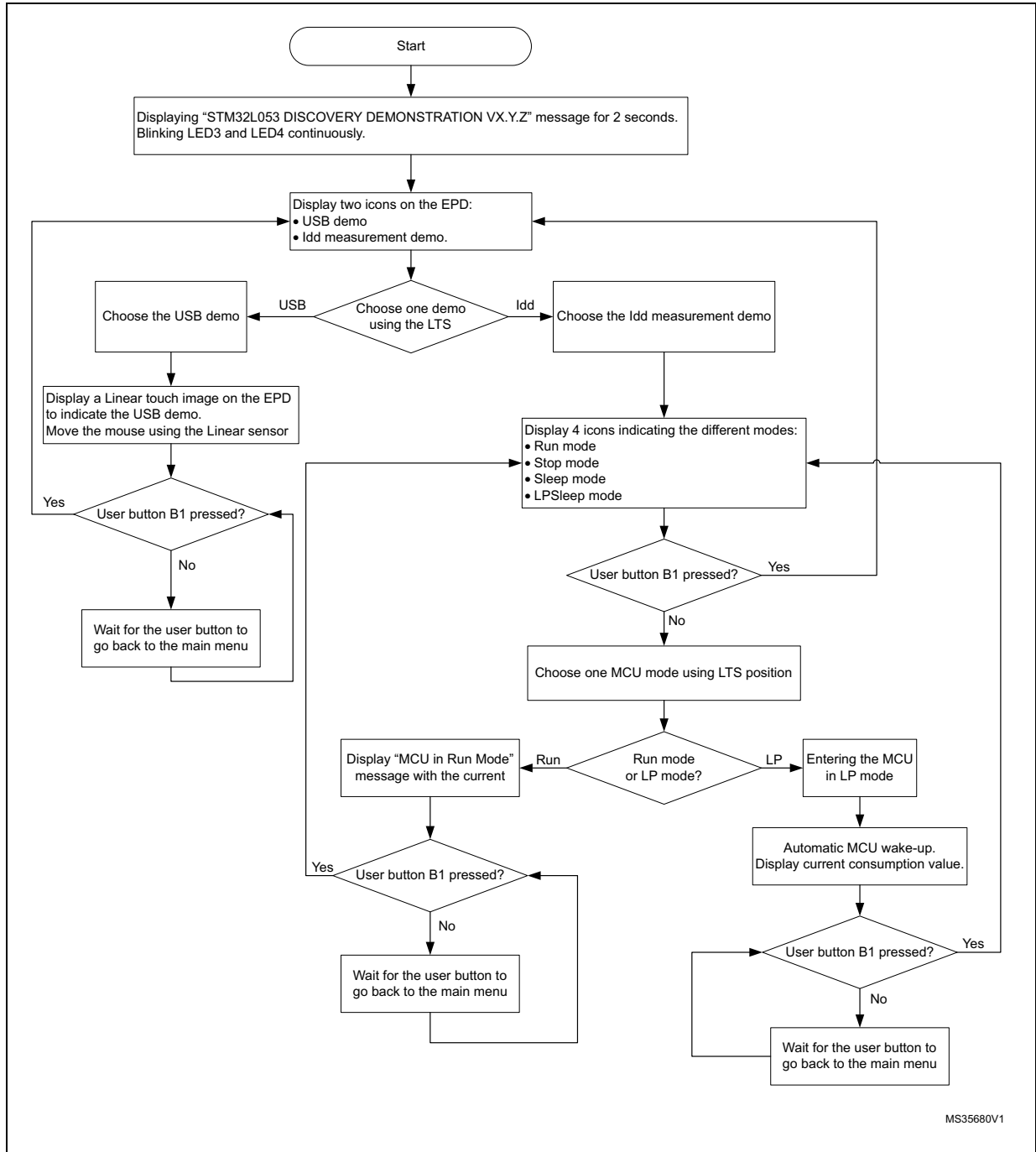
Specific fonts are used for displaying characters, the supported ones are:

1.   font8;
2.   font12;
3.   font16;
4.   font20.

4 Demonstration workflow

Figure 5 illustrates the demonstration workflow.

Figure 5. Demonstration diagram

5 Demo functional description

The STM32L053 Discovery kit demonstration, integrates two major modules: the USB Demo and the Idd measurement demo.

- USB Demo: emulates a USB mouse movement according to the Linear Sensor position.
- Idd measurement Demo: measures and displays in real time the MCU current consumption depending on the selected power mode. The current is measured and calculated thanks to a second microcontroller on the board which is STM32 L1 MCU.

5.1 Demo description

After powering the Discovery, the two leds LED3 and LED4 blink continuously. The message "STM32L053 DISCOVERY DEMONSTRATION VX.Y.Z" (see *Figure 6*) is displayed on the e-paper display (EPD) for 2 seconds.
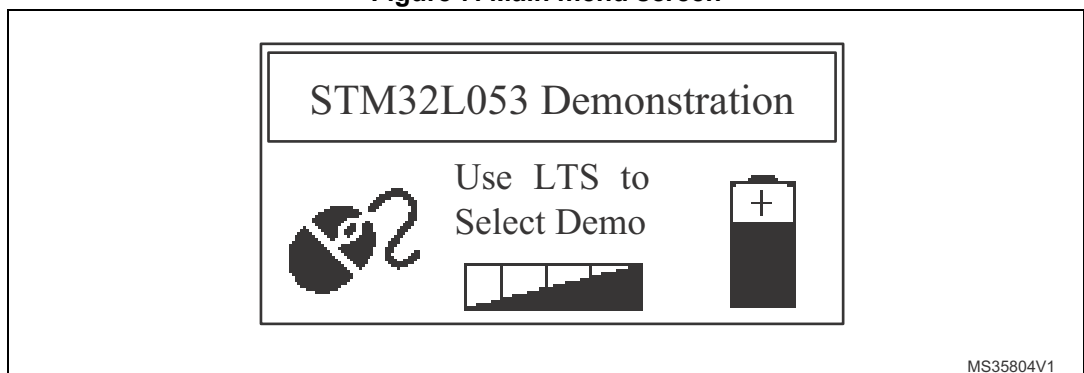
Figure 6. First display screen

Note: *VX.Y.Z refers to the demonstration version, example V1.0.1*

Then, the EPD screen displays two icons (see *Figure 7*) indicating the two possible demos.

Figure 7. Main menu screen

The linear touch sensor is used to select the desired demo to run, e.g. upper side for the USB Demo, lower side for the Idd measurement.
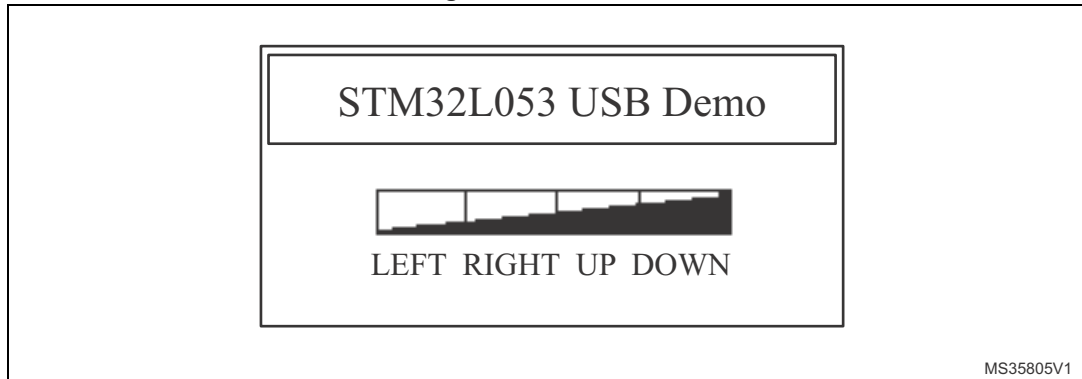
5.1.1 USB demo selected

Connect a second USB cable type A to mini-B between CN3 and the PC, then see the mouse cursor movement according to the linear touch sensor (LTS) position.

Each part of the LTS corresponds to a specific mouse movement direction (Right, Left, Up, Down), as shown in *Figure 8*.

Press User Button B1 to go back to the main menu.

Figure 8. USB demo

STM32L053 USB Demo

LEFT RIGHT UP DOWN

MS35805V1
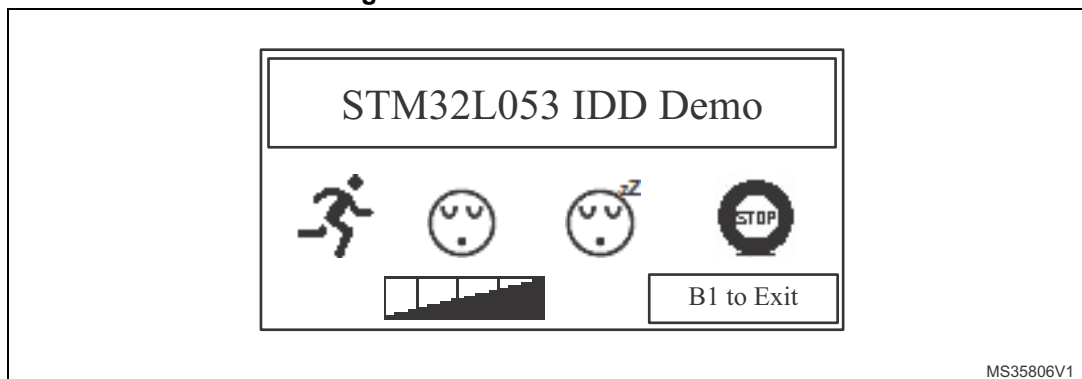
5.1.2 Idd measurement demo selected

This demo allows the measurement of the Idd values for different power modes, depending the user selection.

Figure 9 shows, from left to right, the possible power modes supported by this demo, i.e.:

1. Run mode;
2. Sleep mode;
3. Low Power Sleep mode
4. Stop mode.

Figure 9. Idd measurement demo

STM32L053 IDD Demo
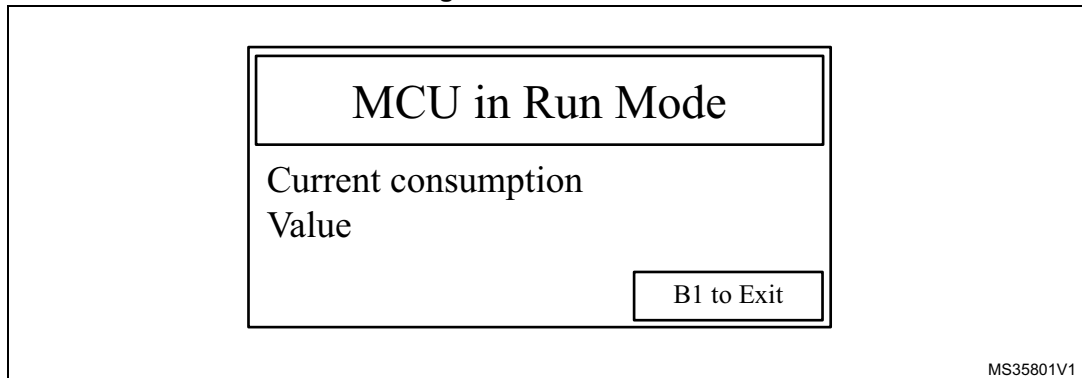
B1 to Exit

MS35806V1

To select one of the power modes, use the LTS. Each part is associated to a power mode.

Press User button B1 to go back to the main menu.

5.1.3 Run mode selected

If the Run mode is selected, the EPD displays the current consumption, as in *Figure 10*.

Figure 10. Run mode

To go back to the previous menu, press B1.

5.1.4 Stop mode selected

This section describes Idd measurement in Stop mode, the same behavior applies to the Sleep and the LPSleep modes.

Before displaying the current consumption in the LP mode, a message to indicate that the MCU is entering in Stop mode is displayed on the EPD (see *Figure 11*).

Figure 11. Stop mode
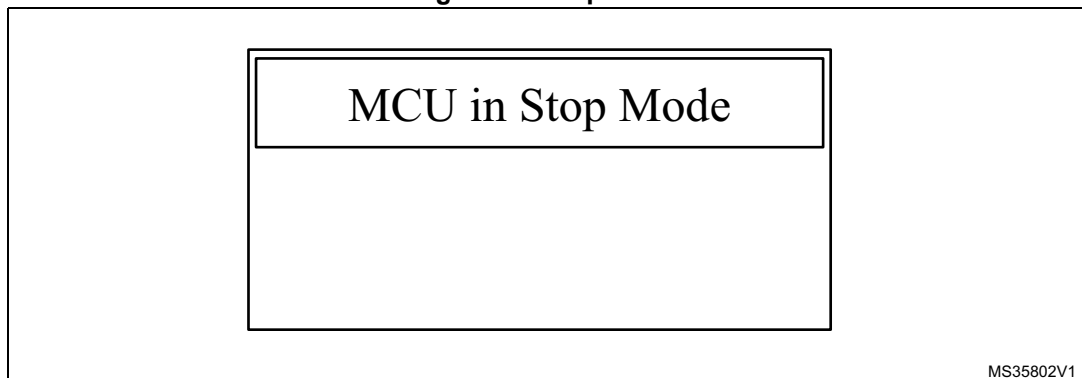
The MCU will be automatically woken up, through an external event EXTI13. This event is the end of the Idd measurement done on the STM32L1 MCU side. The EPD will display the view shown in *Figure 12*, containing the measured current value in real time.

Figure 12. MCU wake-up
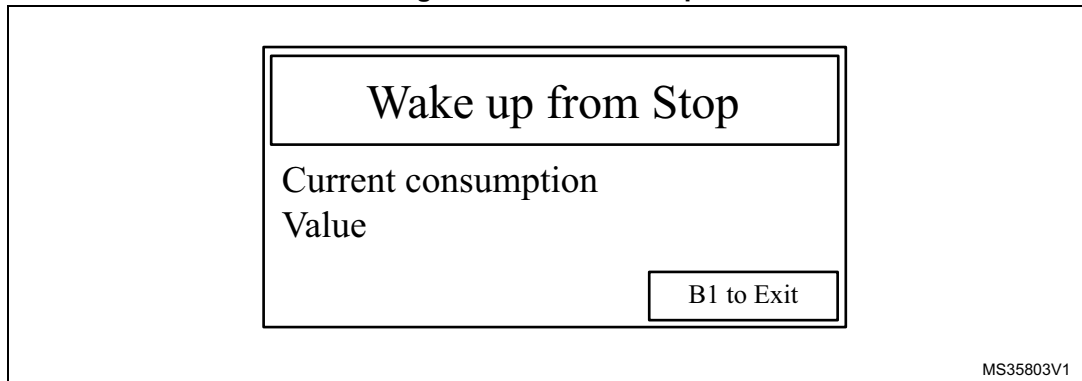
To exit this screen and go back to the menu containing the Run and the Low Power modes, the user has to press button B1.

5.2 Programming firmware application

To program the STM32L053 Discovery board with the demonstration application please proceed as follows:

1. Install the preferred Integrated Development Environment (IDE).
2. Install the ST-LINK/V2.1 driver available on ST website.

There are two ways of programming the Discovery board:

Upload the **STM32L053C8-Discovery_Demo.hex** from the firmware package available under Projects\STM32L053C8-Discovery\Demonstrations\Binary using your preferred in-system programming tool.

Choose one of two supported tool chains (IAR™ / Keil$^®$) and follow the steps below:

- Open the application folder: Projects\STM32L053C8-Discovery\Demonstrations\
- Chose the desired IDE project (EWARM for IAR, MDK-ARM for Keil)
- 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

The demonstration software as well as other software examples that allow you to discover the STM32 microcontroller features are available on ST website at www.st.com/stm32discovery.

6 Glossary

EPD: Electronic Paper Display

LTS: Linear Touch Sensor

7 Revision history

Table 1. Document revision history

Date	Revision	Changes
04-Sep-2014	1	Initial release.

IMPORTANT NOTICE – PLEASE READ CAREFULLY