UM1520
User manual

STM320518-EVAL demonstration firmware

Introduction

This user manual describes the demonstration firmware running on the STM320518-EVAL evaluation board, which can be used to evaluate the capabilities of the STM32F0518(T6) microcontroller and on-board peripherals.

This demo contains many applications that can be easily reused, such as dual interface (I2C and RF) EEprom application (ESL and DataLogger), HDMI-CEC, Infrared RC5 and SIRC Receiver and Transmitter, RTC calendar, File system FAT implementation on SD Card, Wave player using STM32 DAC peripheral, Voice recording, low power modes, Temperature sensor interfacing and TFT LCD.

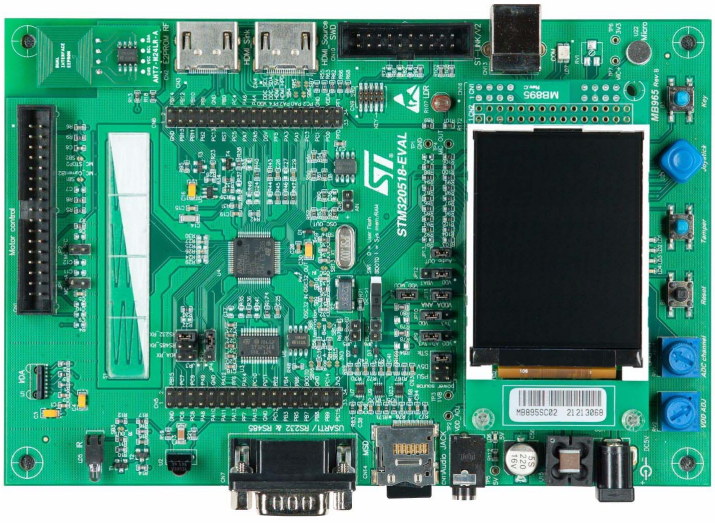
Table 1 lists the tool concerned by this user manual.

Table 1. Applicable tool

Type	Applicable products
Evaluation tool	STM320518-EVAL evaluation board

This demonstration firmware and other such firmware are available for download from the STMicroelectronics website: www.st.com.

Figure 1. STM320518-EVAL evaluation board



Contents

List of figures

1 Functional description

The STM32F051R8(T6) microcontroller evaluation board provides a development and demonstration platform for STM32F0518-based applications. It has been designed to let the user try out the major functions of the STM32F051R8(T6) microcontroller.

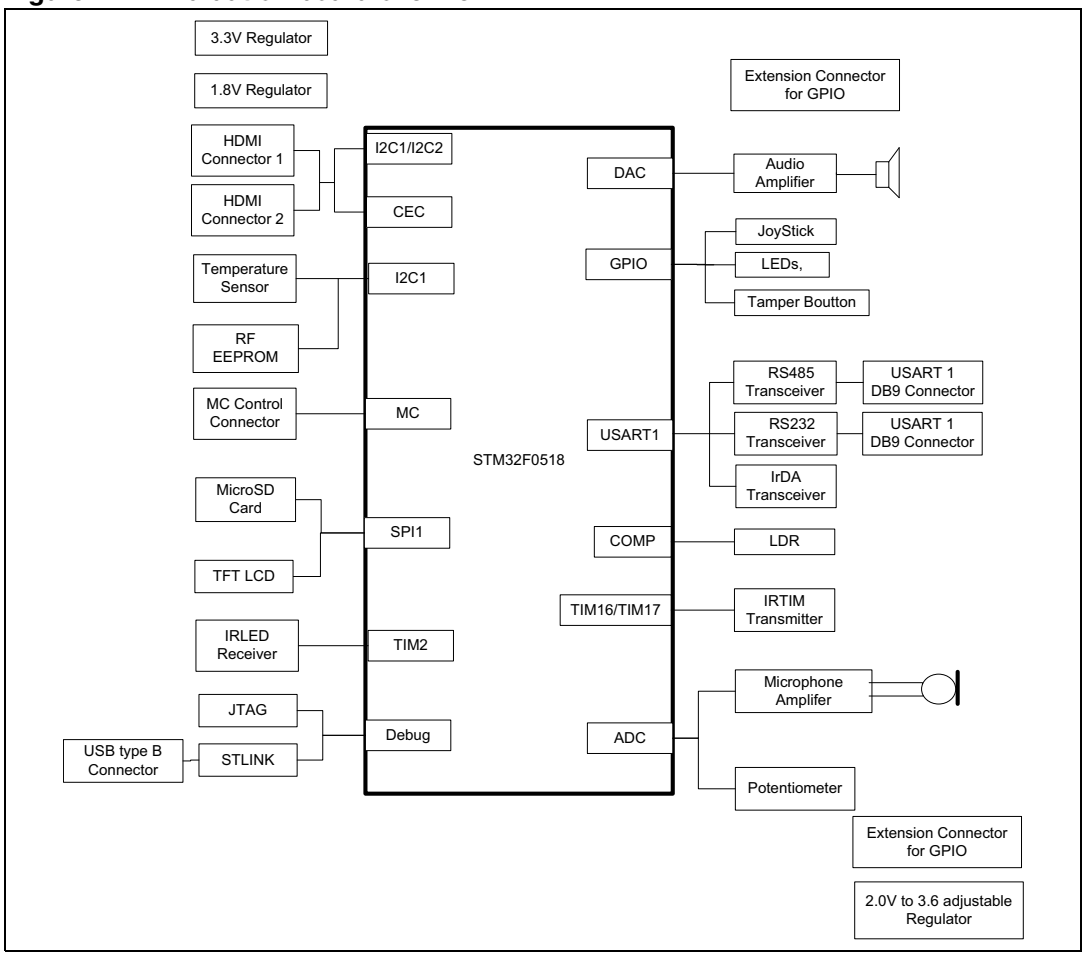
Due to the code size constraint, the STM320518-EVAL demonstration firmware is provided with 2 configurations and the STM320518-EVAL board is delivered with the demonstration config1 programmed in the internal Flash memory, and all the files needed by the demonstration are programmed in the MicroSD card. The demonstration is executed at each reset (board power-up, external reset, etc.).

In case the STM320518-EVAL board was not factory-programmed or the demonstration application was erased or need to change the STM320518-EVAL board configuration, the Bootloader, IAP or STM32 STLink Utility can be used to program this file. For more details, refer to *Section 3: STM320518-EVAL demonstration package* and *Section 4: STM320518-EVAL demonstration programming*.

Before you execute the demonstration, make sure that all EVAL board jumpers are well configured. For more details, refer to *Section 1.10.12: STM320518-EVAL board jumper configuration*.

Figure 2 summarizes the main functional blocks of the evaluation board.

Figure 2. Evaluation board overview



1.1 Power control

The evaluation board can be powered from an external 5 V supply or from the USB connector or ST-LINK/V2 connector. All other required voltages are provided by on-board voltage regulators.

1.2 Clocking

Two clock sources are available on the STM320518-EVAL evaluation board:

- 32 KHz crystal for embedded RTC
- 8 MHz crystal for the STM32F051R8 main clock system

1.3 Reset control

The reset can be generated by hardware or software:

- Reset button: activates the RESET input when pressed
- JTAG reset

1.4 Debug JTAG interface

Software debug is done via the standard ARM® JTAG connection: 20-pin IDC (insulation displacement connector) for connection to the standard ARM host interface.

1.5 Serial wire debugger interface

The Serial Wire Debug Port (SWD-DP) provides a 2-pin (clock + data) interface to the AHP-AP port.

1.6 Embedded ST-LINK/V2

An embedded ST-LINK/V2 is integrated on the board as an embedded in-circuit debugger and programmer for the STM32F051R8 MCU.

1.7 Display devices

1.7.1 LCD

A color LCD module is mounted on the STM320518-EVAL board. It is interfaced through the embedded SPI peripheral.

1.7.2 LEDs

Four general-purpose LEDs are available. They are used as a display.

1.7.3 LDR (Light Dependent Resistor)

The VDDA is divided by the resistor bridge of LDR VT9ON1 & 8.2K resistor and connected to PA1.

1.8 Interfaces

1.8.1 RS232 and RS485

The STM32F0518 evaluation board (STM320518-EVAL) provides one on-board RS-232 and RS485 serial ports. RS232 port (USART1) is accessed via CN7 connector.

1.9 IrDA

The STM320518-EVAL evaluation board supports IrDA communication. The interface is mounted on USART1 (U1 interface).

1.10 Miscellaneous peripherals

1.10.1 Joystick

Four-direction joystick with a selection key.

1.10.2 Push-buttons

The following push-buttons are available:

- Key
- Tamper
- Wakeup (Joystick Sel): used to wake up the processor from low power mode

1.10.3 12-bit analog-to-digital converter (ADC)

Varistor: ADC channel (ADC1_IN11) connected to an on-board variable resistor. The variable resistor provides a voltage in the range of 0 V to 3.3 V.

1.10.4 Audio

The STM320518-EVAL evaluation board implements a dedicated audio amplifier to be interfaced with the STM32 DAC peripheral. For the audio output, a speaker and an audio Jack are available on the board and connected to the DAC.

1.10.5 MicroSD card

The STM320518-EVAL evaluation board has a MicroSD card connector connected to the SPI1 peripheral.

1.10.6 Serial EEPROM

The STM320518-EVAL evaluation board includes a serial EEPROM connected to the SPI1 peripheral.

1.10.7 RF EEPROM

The RF EEPROM daughter board implemented on the module is the M24LR64-R. The daughter board can be connected on CN2 to STM32F051R8 via the I2C bus. The I2C address of the RF EEPROM is 0b1010E2E1E0. E0-E2 values are determined by the RF EEPROM daughter board.

1.10.8 IR LED and IR receiver

The TSOP34836 IR receiver is connected to PB3 of STM32F051R8, and a current around 100mA on the IR LED is driven by PB9 through transistors T1 and T2 on the board.

1.10.9 HDMI CEC

Two HDMI connectors, CN3 and CN4, are available on the STM320518-EVAL board.

1.10.10 Temperature sensor

The STM320518-EVAL evaluation board includes an I^2C temperature sensor connected to the I2C1 peripheral.

1.10.11 Touch slider

Touch slider is supported on the STM320518-EVAL evaluation board and connected to 4 capacitive sensing channels.

1.10.12 STM320518-EVAL board jumper configuration

To be able to run the STM320518-EVAL demo correctly, configure the following STM320518-EVAL board jumpers as follows:

- VDD Adjust: JP9 fitted pos VDD
- VDD Voltage: JP10 fitted pos 3.3V
- VDD Analog: JP11 fitted
- VDD VBat: JP12 fitted pos VDD
- VDD MCU: JP7 fitted
- Audio Output: JP13 fitted
- When running the LDR demo, jumper JP13 “Audio output” should not be fitted, to avoid noise on the speaker
- JP5, JP6 and CN8 are not fitted.

2 Running the demonstration

2.1 Menu

Due to code size constraints, the STM320518-EVAL demonstration firmware is provided with 2 configurations. *Figure 3* shows the menu system of the STM32F0518 demonstration configuration1 and *Figure 4* shows the menu system of the STM32F0518 demonstration configuration2. The main menu is shown on the left-hand side. The UP, DOWN, RIGHT and LEFT joystick directions allow the user to navigate between items in the main menu and the submenus. To enter a submenu, press the SEL push-button.

The SEL push-button designates the action of vertically pressing the top of the joystick, as opposed to moving it horizontally UP, DOWN, RIGHT or LEFT.

To exit a submenu, select the Return menu and press SEL.

Figure 3. Structure of the demonstration config1 menus

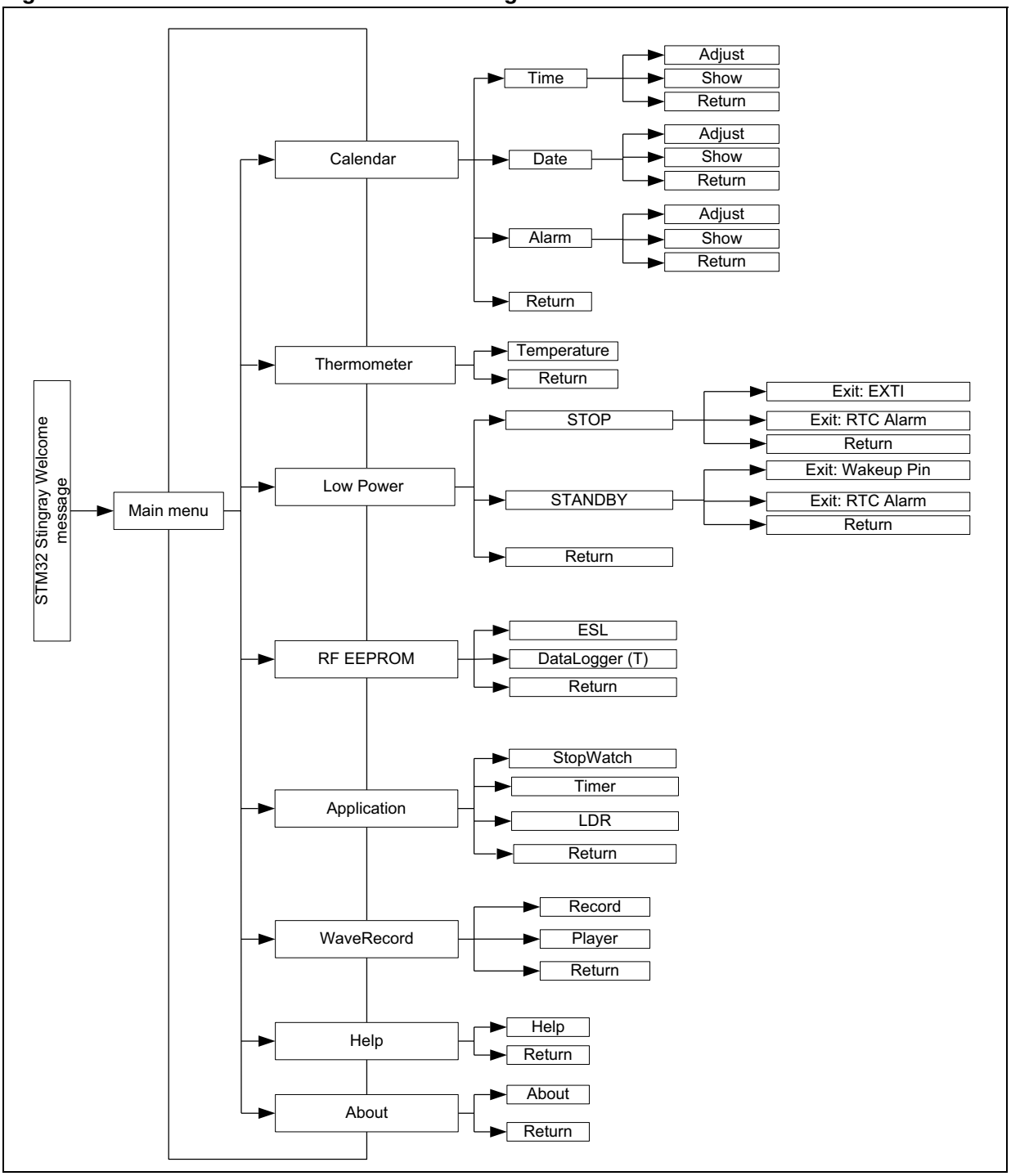
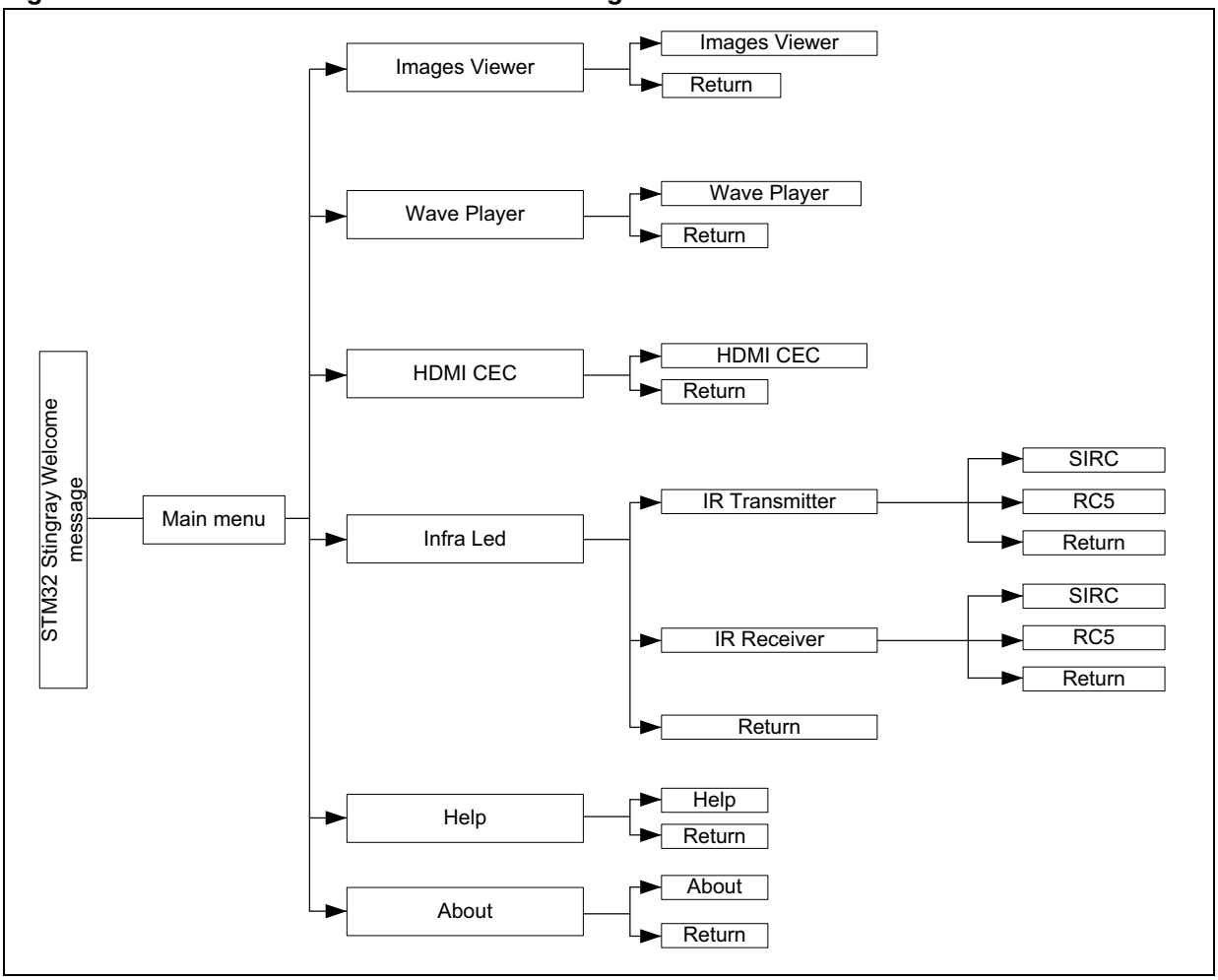


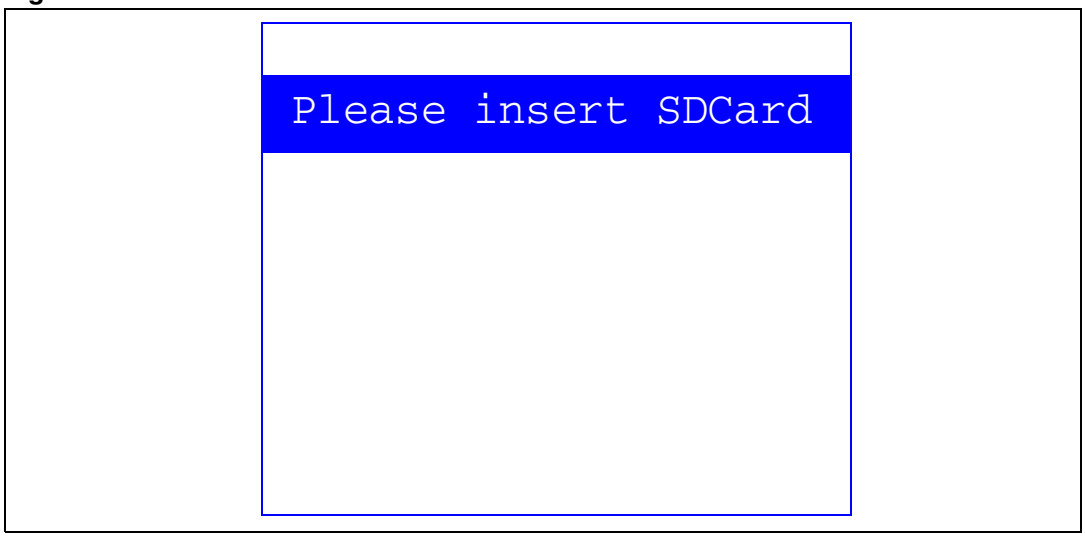
Figure 4. Structure of the demonstration config2 menus



2.1.1 Demo startup

After a board reset, at demo startup, the system checks if an SD card memory is already present in connector CN3. If no card is detected, the demo does not start and the message shown in *Figure 5* is displayed on the LCD screen.

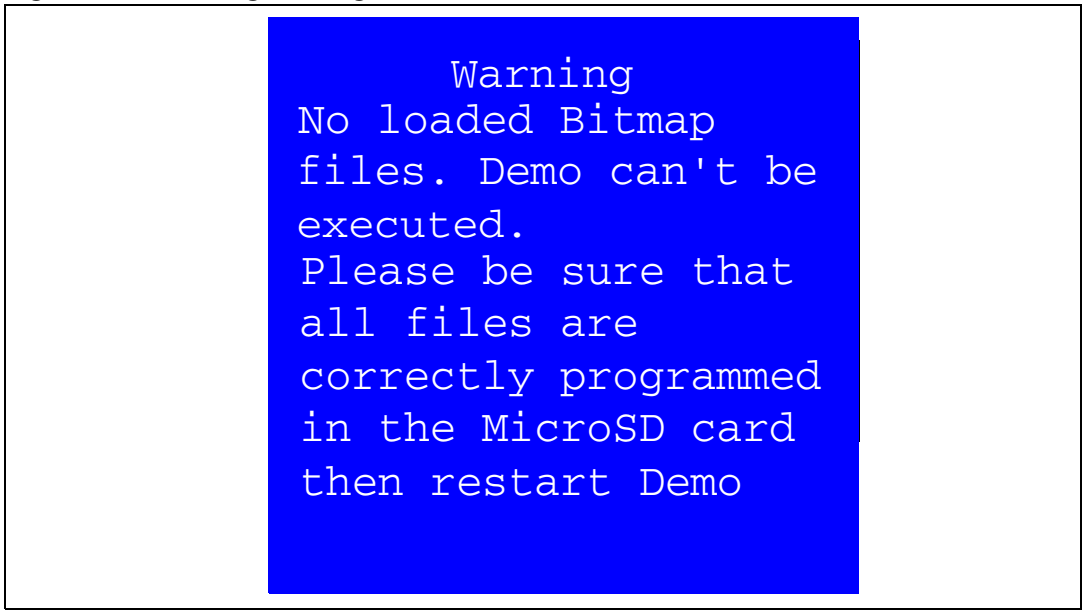
Figure 5. SD card check



The demo continues only if an SD card is inserted.

Then, the demo graphic icons and bitmap files are checked in the MicroSD card (see *Section 2.3.5: External memory organization*). All the icons have to be correctly programmed in the MicroSD card for the demo to start, so if an icon is missing, the demo does not start and the message shown in *Figure 6* is displayed on the LCD screen.

Figure 6. Warning message



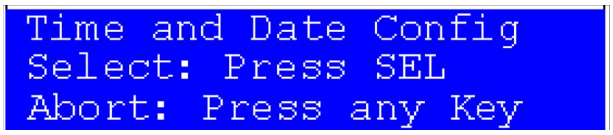
However, if the icons are correctly loaded into the SD Card memory, the welcome screen is displayed and the ST logo appears on the LCD (see *Figure 7*).

Figure 7. ST logo

When the board is powered up for the first time, the user is prompted to set the time, year, month and day. The user may choose to ignore it by pressing any key except for the SEL push-button to abort the configuration sequence. To set the time and date, the user must press SEL and follow the setting sequence.

The message shown in *Figure 8* appears on the LCD screen.

Figure 8. Time and date configuration



Note: 1 If the user chooses to configure the time and date, the Time Adjust and Date Adjust menus are displayed. Otherwise, the main menu is displayed and the user can set the time parameters in the Calendar menu. To set the time/date, use the UP/DOWN joystick and SEL push-button.

2 If the time configuration has already been done, then the number of elapsed days (higher than 1 day) from the last time the demo board was powered up appears on the LCD screen. It is soon followed by the current date.

Once the time/date have been set, the main menu appears. The main menu is displayed in the form of a set of icons. It shows all the submenus on the same screen. You can navigate using the UP, DOWN, RIGHT and LEFT joystick directions to select the required submenu. To enter a submenu, press the SEL joystick push-button, and the new submenu corresponding to the selected icon is displayed.

Figure 9. Main menu config1

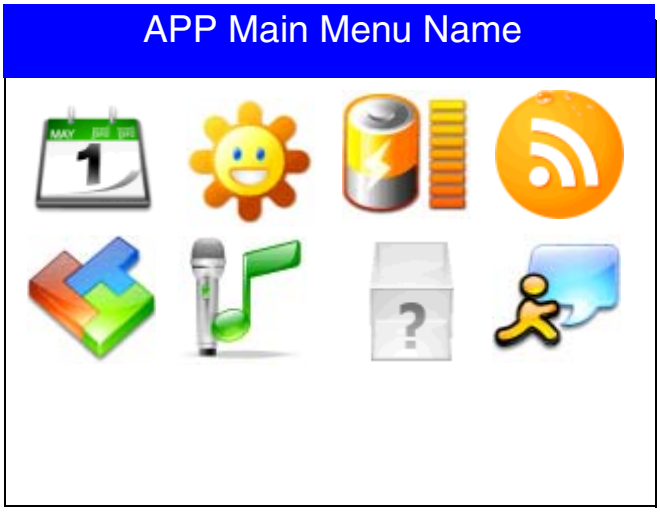
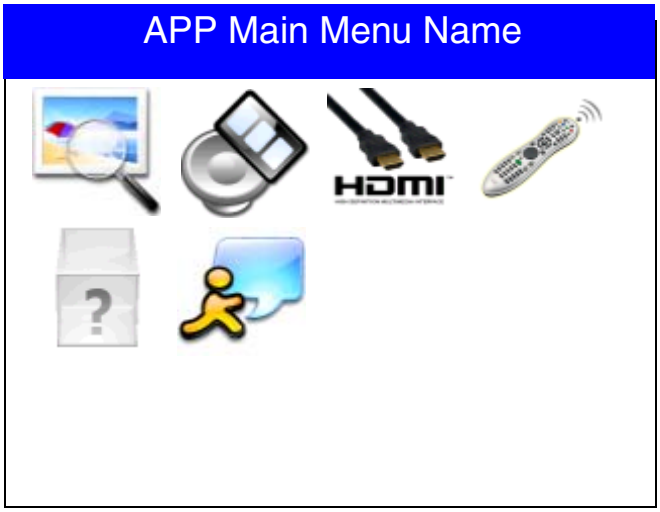


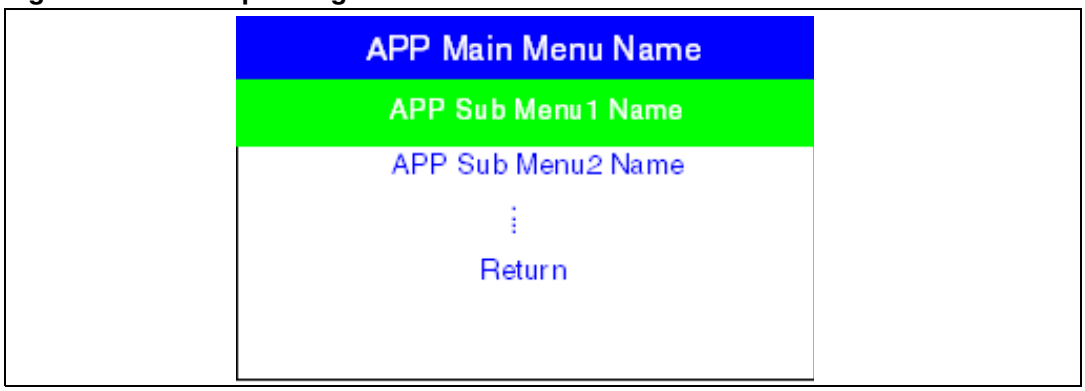
Figure 10. Main menu config2



Note: The icons shown in Figure 9 and in Figure 10 are taken from http://commons.wikimedia.org/wiki/Crystal_Clear.

Once a submenu has been selected, the name of the application is listed at the top of the display and all the corresponding submenus are listed below, as shown in *Figure 11*.

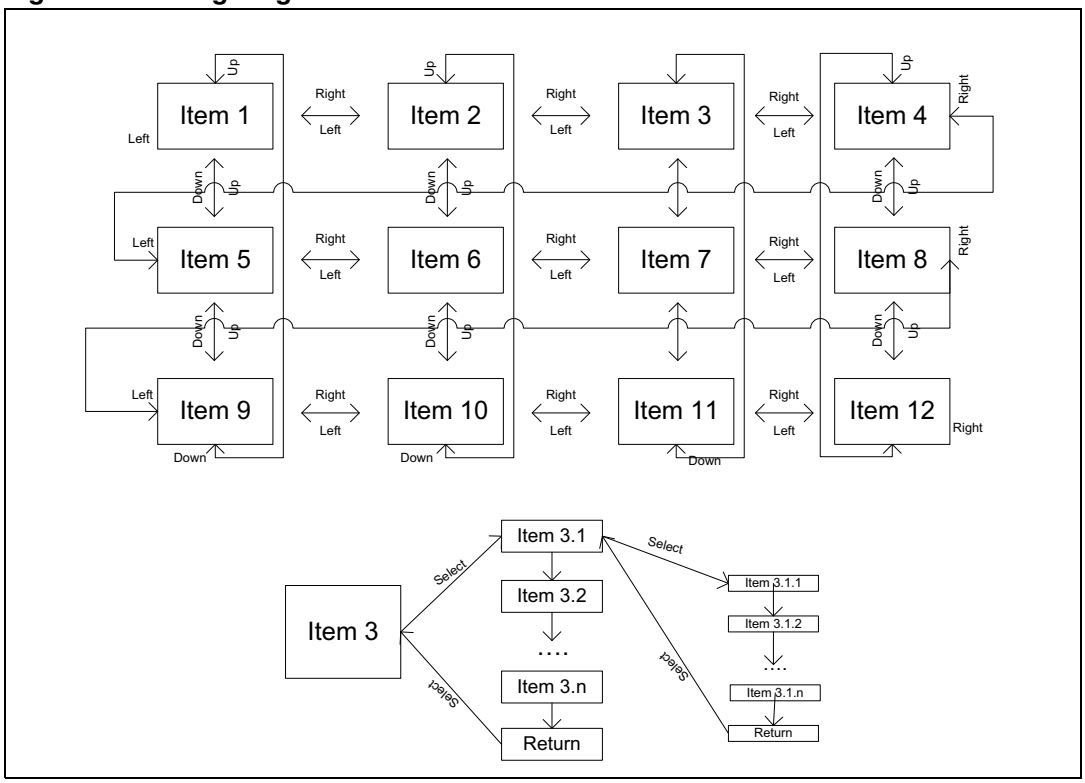
Figure 11. Corresponding submenus



2.1.2 Navigation

The demonstration menu is based on circular navigation, submenu selection, item selection and back navigation as described in *Figure 12*.

Figure 12. Navigating in the demonstration menus



The user navigates using the joystick push-buttons located on the evaluation board: RIGHT, LEFT, SEL, UP and DOWN.

- The UP, DOWN, RIGHT and LEFT push-buttons are used to perform circular navigation in the main menu and the current menu items.
- The SEL push-button selects the current item.
- The UP and DOWN push-buttons are used for vertical navigation in the submenus.
- To return to the upper menu, go to the Return line and press SEL.

2.2 Clock sources

2.2.1 Clock control

The STM32F051R8 internal clocks are derived from the HSE (clocked by the external 8 MHz crystal).

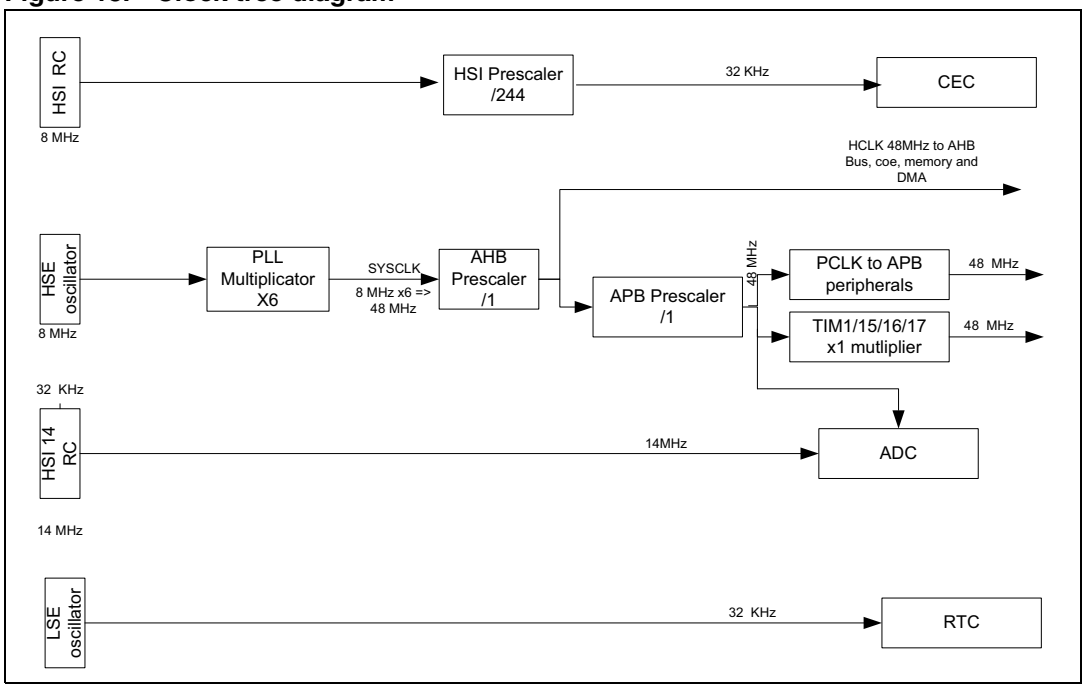
In this demo application, the various system clocks are configured as follows:

- System clock is set to 48 MHz: the PLL is used as the system clock source.
- HCLK frequency is set to 48 MHz.
- Timer clock (TIMCLK) is set to 48 MHz.
- ADC clock is set to 14 MHz.
- CEC clock is set to 32 KHz.
- PCLK is set to 48 MHz.

Only the RTC is clocked by a 32 kHz external oscillator.

Figure 13 illustrates the clock tree organization for this demo.

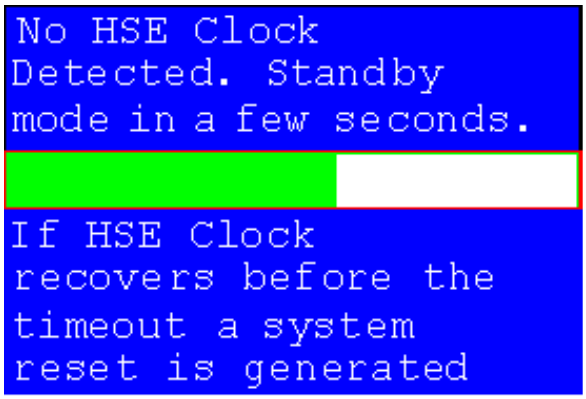
Figure 13. Clock tree diagram



2.2.2 Clock failure

At any demo level, if no clock is present on OSC_IN (broken or disconnected crystal), the message shown in *Figure 14* is displayed on the LCD screen.

Figure 14. No HSE clock detected

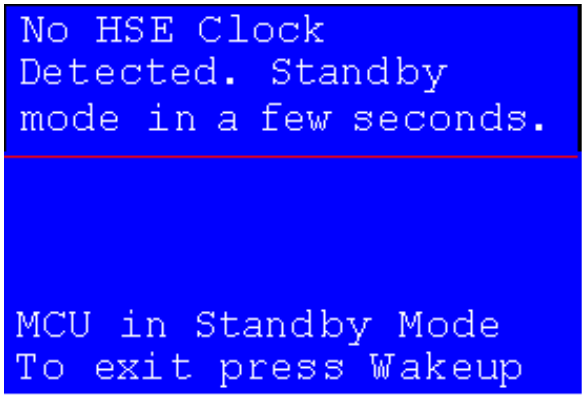


If the 8 MHz crystal is not reconnected in the next few seconds, the MCU enters the Standby mode. If the 8 MHz crystal is reconnected within a few seconds, a system reset is generated.

Note: The clock security system (CSS) feeds the MCU with the MSI OSC used as an emergency clock if no clock is detected.

When a timeout occurs, the MCU enters the Standby mode and the message shown in *Figure 15* is displayed on the LCD screen.

Figure 15. Standby mode entered



Note: The demo does not restart as long as the 8 MHz crystal is not present.

Connecting the 8 MHz crystal after reset may not restart the demo correctly. The crystal must be connected before starting the demo.

2.3 STM32F051R8(T6) resources

2.3.1 Peripherals

All used peripherals are described in *Table 1*.

Table 2. STM32F051R8(T6) demo peripherals

Used peripherals	Application
I2C1	Temperature sensor, dual interface EEPROM and CEC
EXTI	Menu navigation + joystick + push-button + low power mode + audio + Applications
GPIO	All applications + LEDs
NVIC	All applications using interrupts
PWR	Low power modes
RCC	All applications + Demo kernel
RTC	Calendar + Applications (StopWatch and Timer)
SysTick	Generate 10 ms time base
TIM15	LED toggling
TIM3 and TIM4	Voice recording + Voice player
TIM2 Tim16 and TIM17	Infra Led Transmitter and Receiver + CEC Demo
ADC1	Voice recording + IDD measure
CEC	CEC Demo
DAC	LDR (define the comparator level) + wave player
SPI1	MicroSD + Color LCD
COMP	LDR (Brightness level)

2.3.2 Interrupts

Table 2 shows all the enabled interrupts.

Table 3. STM32F051R8(T6) demo interrupts

Interrupts	Priority	Used for
SysTick	SubPriority: 0	System timing
NMI	Priority: -2	CSS interrupt
EXTI0_1	Priority: 0	Wake-Up button
EXTI4_15	Priority: 1	Menu navigation
TIM6_DAC	Priority: 0	Wave Player
I2C1 Error	Priority: 0	SMBus Alert interrupt
TIM2_UP	Priority: 0	Infra Led Receiver interrupt
TIM3_UP	Priority: 2	Wave Record interrupt
TIM14_CC	Priority: 1	Wave Record interrupt
TIM15_UP	Priority: 1	LED toggling
TIM16_UP	Priority: 0	Infra Led Transmitter interrupt
RTC	Priority: 1	Calendar, date update, Tamper and Alarm generation
CEC	Priority: 0	CEC interrupt

2.3.3 External interrupts

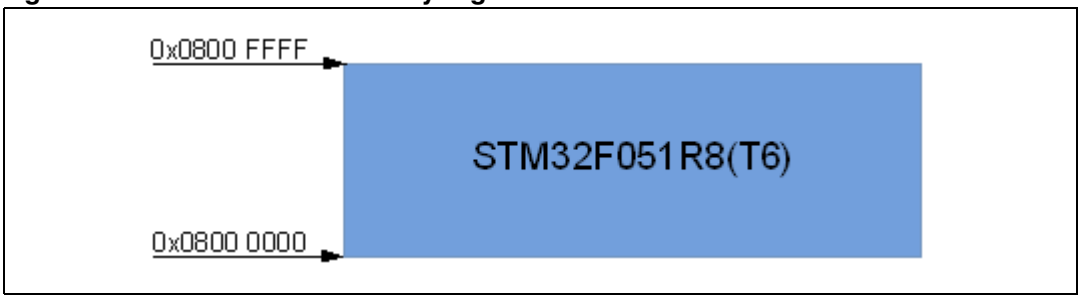
Table 3 shows all the external interrupts used by the demonstration.

Table 4. STM32F051R8(T6) demo external interrupts

External interrupts	Used for
EXTI line 0	Joystick SEL (interrupt mode, falling edge)
EXTI line 6	Joystick UP (interrupt mode, falling edge)
EXTI line 7	Joystick DOWN (interrupt mode, falling edge)
EXTI line 8	User Button (interrupt mode, falling edge)
EXTI line 15	SD Card detect (interrupt mode, rising and falling edge)
EXTI line 17	RTC Alarm (interrupt mode, rising edge)

2.3.4 Internal memory size

Figure 16. Internal Flash memory organization



2.3.5 External memory organization

The STM320518-EVAL demo is based on an embedded free FAT file system, FatFs. The file system is needed to read all media information from the on-board MicroSD card memory. The SD card memory is organized in three subdirectories:

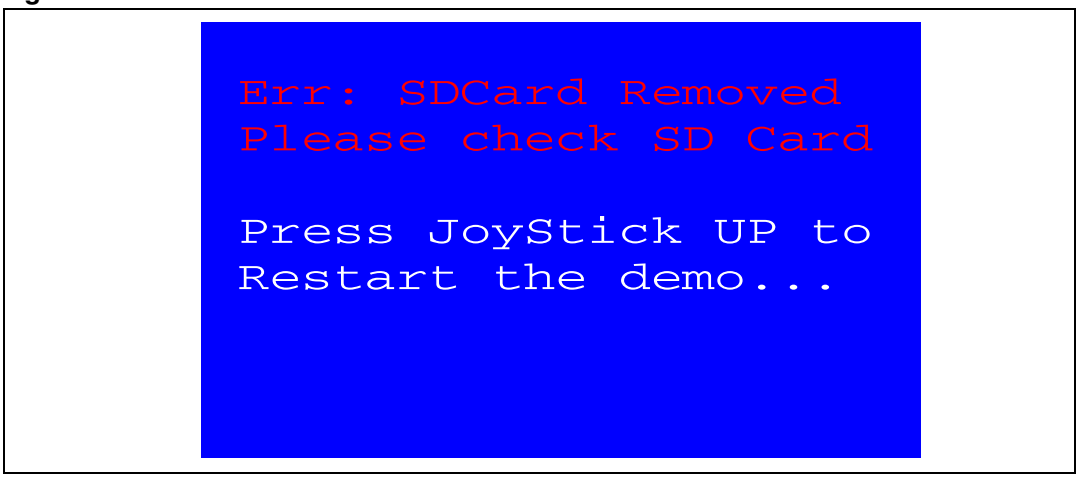
- STFILES: this directory contains all the required demo media files (icons, waves and slides). User files located in this folder cannot be handled by demo; only default files are managed.
- USER: this is a user folder. The user can add his/her own files here to be played inside the demo menus (pictures and waves). This folder is used only by the Image Viewer and Wave Player submenus. For more details on the various files properties, please refer to *Section 2.4.2: Image Viewer submenu* and *Section 2.4.3: Wave Player submenu*.
- REC: this directory contains the voice recorded wave file “Rec_wave.wav“. (This Folder will be created when the Voice Recording application is run).

The MicroSD card memory provided within STM320518-EVAL board is already programmed with the media files required to run the demonstration. These files are also available within the demonstration firmware package under Project\STM320518-EVAL\Binary\Media folder.

Note: The STFILES directory and its internal files are mandatory for demo startup. FatFs is a generic FAT file system module for small embedded systems. The FatFs is written in compliance with ANSI C and completely separated from the disk I/O layer. For more details, refer to the following link: http://elm-chan.org/fsw/ff/00index_e.htmltml.

At any demo level, if the SD card is removed, the demo stops and the message shown in Figure 17 is displayed on the LCD screen.

Figure 17. SD card removal



2.4 Demo applications

The following section provides a detailed description of each part of the demonstration.

Note: In the demonstration, the core runs at HCLK = 48 MHz.

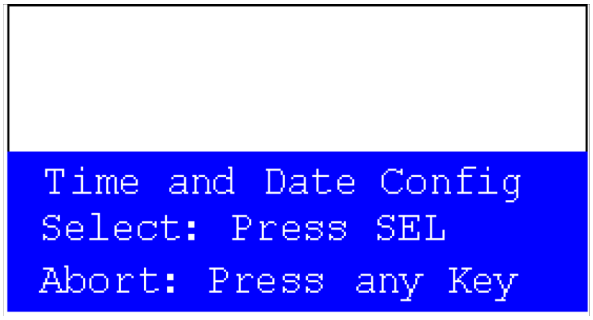
Four LEDs (LD1, LD2, LD3 and LD4) flash throughout the demonstration at a frequency depending on the core clock.

2.4.1 Calendar

The STM32F051R8 features a real-time clock (RTC) which is an independent BCD timer/counter. The RTC provides a time-of-day clock/calendar, two programmable alarm interrupts, and a periodic programmable wakeup flag with interrupt capability.

This submenu is used to configure the time, date and alarm. In any submenu, if the time and date parameters have not yet been configured, the message shown in *Figure 18* is displayed on the LCD screen.

Figure 18. Setting the time and date



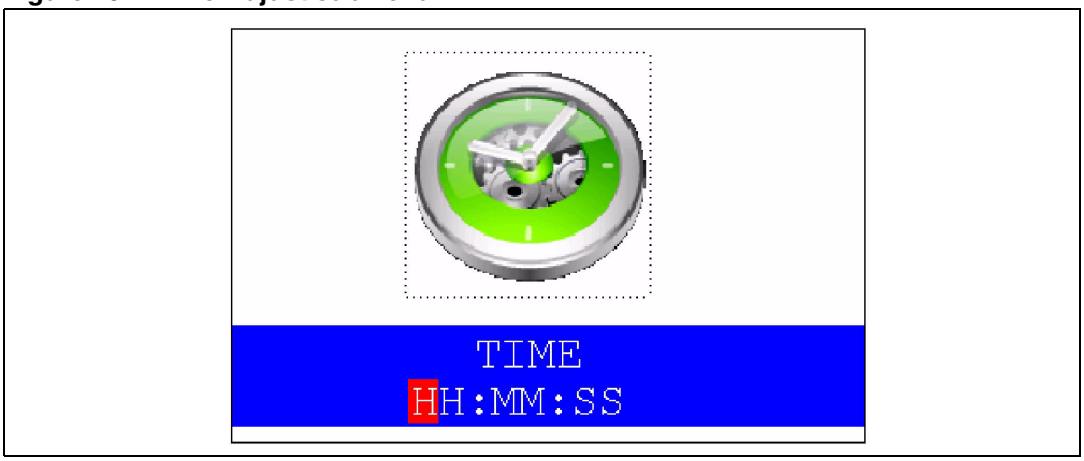
The user can choose to set or not the time, year, month and day. Press any key (except for SEL) to ignore the prompt and abort the configuration sequence. Press SEL and follow the setting sequence to set the time and date.

Time submenu

This submenu is divided into two items that allow the user to display or set the current time.

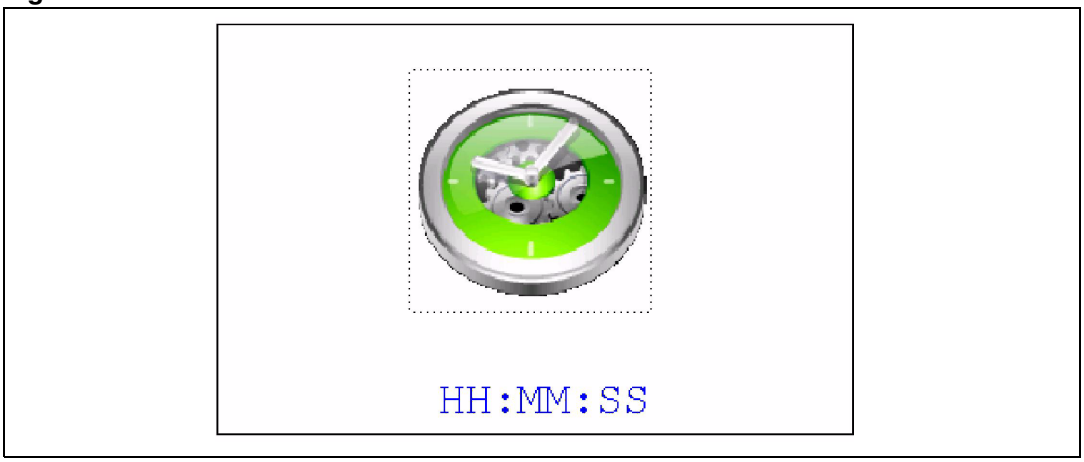
- **Time Adjust:** after the evaluation board is powered up, select this submenu to change the default time (00:00:00) to the current time. Once Time Adjust has been selected, the first digit of the hour field can be changed. Press the UP button to display the current value plus one. Press the DOWN button to display the previous digit value. After setting the digit value, press SEL, and the cursor automatically jumps to the next digit. When all the time digits have been set, the Time submenu appears. Some digit values are limited to a range of values depending on the field (hour, minutes or seconds). The following message (with the default time or the current time) is displayed on the LCD when this submenu is selected.

Figure 19. Time Adjust submenu



- **Time Show:** this item displays the current time. If time and date have not been configured before, a message is displayed, prompting the user to set the time and date or to exit to the upper submenu. When this submenu is selected, the message shown in *Figure 20* appears on the LCD. In the example, the time has not been set yet.

Figure 20. Time Show submenu



To exit the Time Show submenu, press the SEL push-button. To exit the Time submenu, select the Return line and press the SEL push-button.

Date submenu

This submenu is divided into two items that allow the user to display or set the current date.

- **Date Adjust:** select this item after each power-up in order to set the current date. If the time and date have not been configured before, a message is displayed, prompting the user to set the time and date or to exit to the upper submenu.

  The user is requested to set the current date to be stored in the application memory. The date is displayed as Year, Month, Week Nbr, Day Nbr (number of the day in the year) with the selected day shown in the month. There is no default date since the user has to set the date at least once.

  Once the submenu has been selected, the user starts by setting the Year, then the Month and the day of the selected month. The Month and the Year are selected using the UP or DOWN push-button. For the day, the UP, DOWN, RIGHT and LEFT push-buttons can be used. Press the UP push-button to display the current value plus one; press the DOWN push-button to display the previous value. To confirm the selected month, press the SEL push-button. The display then jumps to the year configuration. The same procedure is applicable for the year configuration.

  After configuring the day, press the SEL push-button to store the entered value and exit to the Date submenu. The current date value is then shown and you can change the setting if required. The messages shown in *Figure 21*, *Figure 22* and *Figure 23* are successively displayed on the LCD when this submenu is selected.

Figure 21. Setting the year

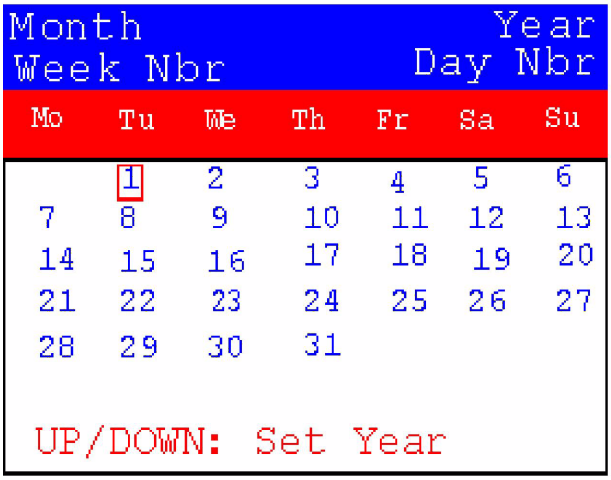


Figure 22. Setting the month

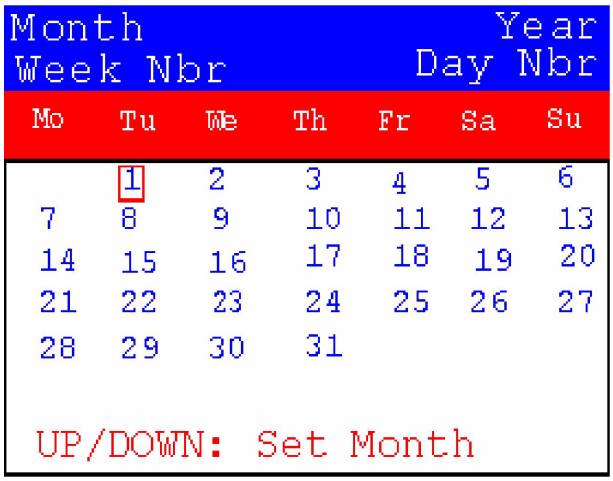
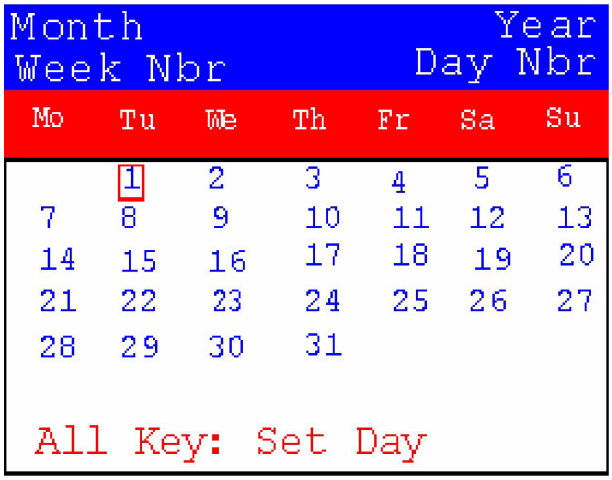


Figure 23. Setting the day of the month



- **Date Show:** this item displays the current date. If the time and date have not been configured before, a message is displayed, prompting the user to set the time and date or to exit to the upper submenu. The message shown in *Figure 24* is displayed on the LCD when the submenu is selected (with the date already configured).

Figure 24. Exiting the Date Show submenu

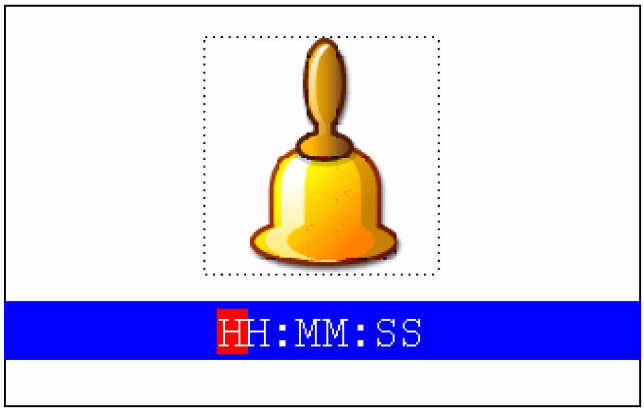
To exit this submenu, press the SEL push-button. To exit the Date submenu, select the Return line and press the SEL push-button.

Alarm submenu

Using this submenu, the user can configure the alarm activation time. When the alarm time value is reached, all the LEDs (LED1 to LED4) start flashing together, and so for 30 seconds. This submenu is divided into two items that allow the user to display or set the current alarm.

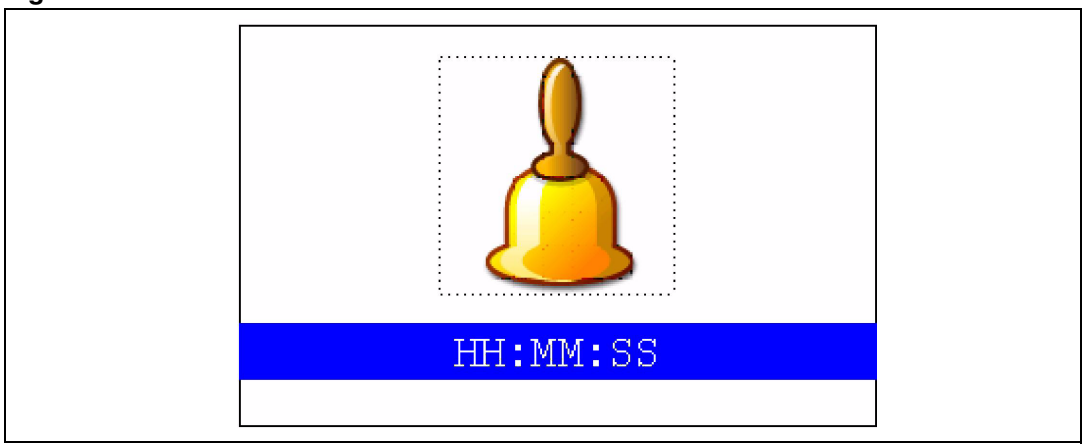
- **Alarm Adjust:** the alarm time activation is set in the same way as the time is set in the Time Adjust submenu. The following messages are successively displayed on the LCD when this submenu is selected:

Figure 25. Setting the alarm activation time



- **Alarm Show:** this item displays the current alarm time. The default Alarm activation time displayed after power-up and before setting in the Alarm Adjust submenu is 00:00:00. If the time and date have not been configured before, a message shown in *Figure 27* is displayed. Pressing SEL takes you back to the Alarm submenu. The message shown in *Figure 26* is displayed on the LCD when this submenu is selected.

Figure 26. Alarm Show submenu



To exit the Alarm Show submenu, press the SEL push-button. To exit the Alarm submenu, select the Return line and press the SEL push-button.

Note: In the Alarm Adjust and Alarm Show menus, if the time and date have not yet been configured, the message shown in Figure 27 is displayed on the LCD screen.

Figure 27. Message displayed if time and date need setting

2.4.2 Image Viewer submenu

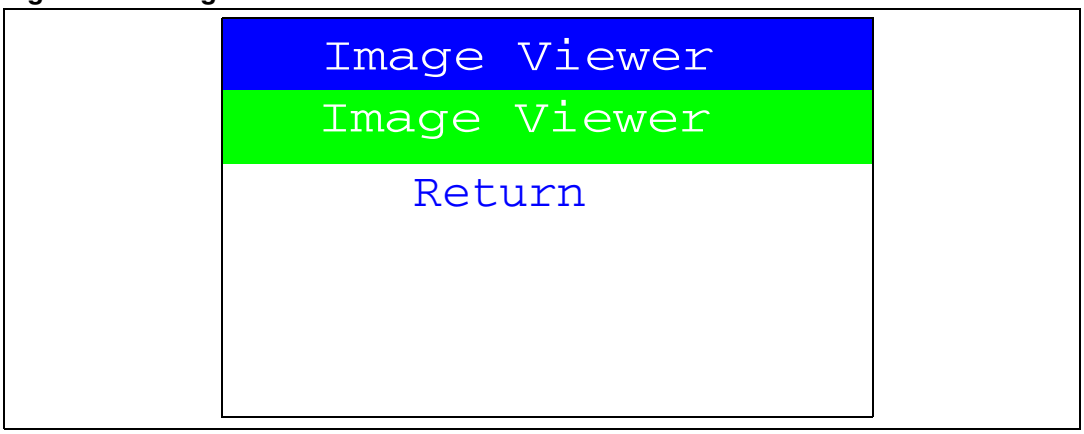
The Image Viewer submenu is used to demonstrate the LCD control performance using the embedded SPI interface. The application is a successive display of stored images.

This application reads all bitmap pictures from the USER directory (see *Section 4.1: Programming the media files*) and displays only the .BMP files having the following format:

- Bit depth: 16 bits (RGB)
- Size: 240x320

Select Image Viewer to display the submenu shown in *Figure 28*.

Figure 28. Image Viewer submenu



When Image Viewer is selected, a list box of images is displayed as shown in *Figure 29.*

Using the UP, DOWN and SEL push-buttons, the user can select and view any image from the list box.

Figure 29. STM32 Image Viewer

When Image Viewer is selected, the corresponding image is displayed and then the user can use the RIGHT and LEFT push-buttons to go to the next/previous image stored in the USER folder on the MicroSD card. If the DOWN push-button is pressed, the Image Viewer is stopped and the submenu shown in *Figure 29* is displayed.

The supported image size is 240x320. The defined maximum number of images that can be read from the MicroSD card is 4 images, selected in alphabetic order.

2.4.3 Wave Player submenu

The STM32F051R8(T6) microcontroller has an embedded DAC which can be used to generate output signals.

In this demo, any wave file stored under the USER folder in the microSD card can be opened using the file system FATFS and transferred to the internal SRAM by block (512 bytes) using the SPI interface. Timer 6 (TIM6) triggers the DAC to generate the wave signal. The voice sampling period is read from the Wave File Header. An audio amplifier is connected to the DAC interface to play the stored wave files.

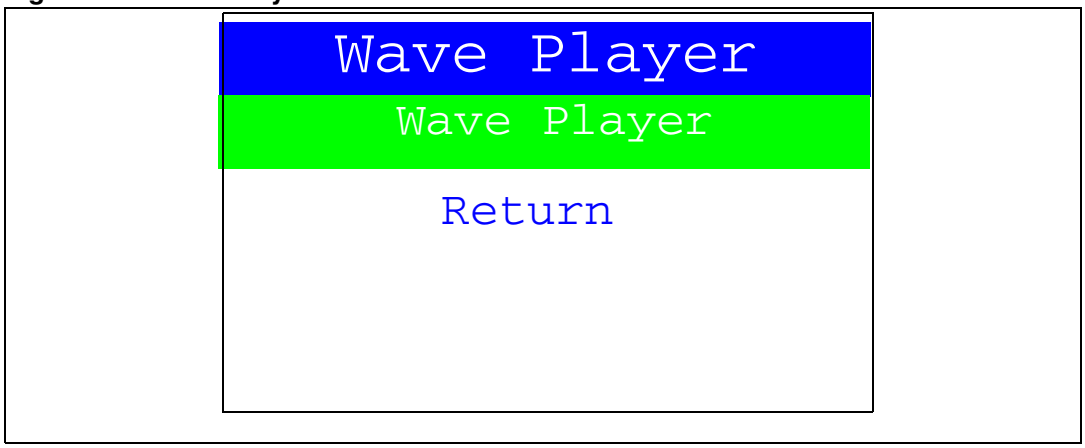
This application reads all wave files from “USER” directory (See *Section 4.1: Programming the Media files*) and displays only the WAV files having the following format:

- Audio Format: PCM (an uncompressed wave data format in which each value represents the amplitude of the signal at the time of sampling.)
- Sample rate: may be 8000, 11025, 22050 or 44100 Hz.
- Bits Per Sample: 8-bit (Audio sample data values are in the range [0-255]).
- Number of Channels: 1 (Mono)

The maximum wave files number that can be read from the microSD card is 4 files selected in alphabetic order.

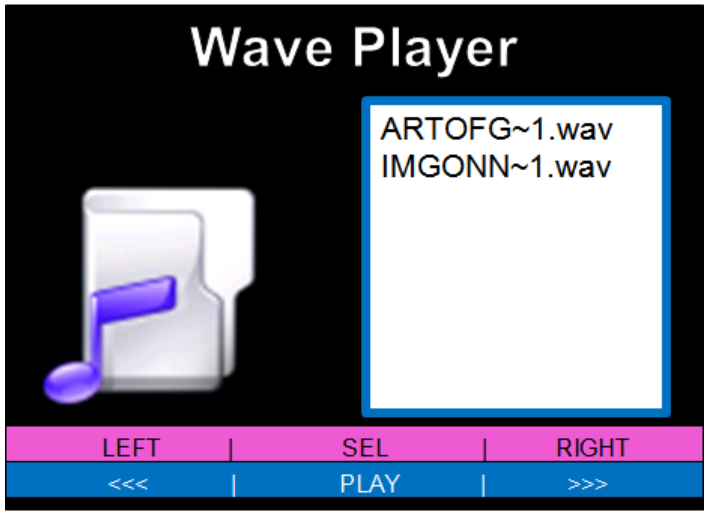
By selecting Wave Player, the submenu shown in *Figure 30* is displayed.

Figure 30. Wave Player submenu



When Wave Player is selected again, the wave player file names are displayed as a list box, as shown in *Figure 31*.

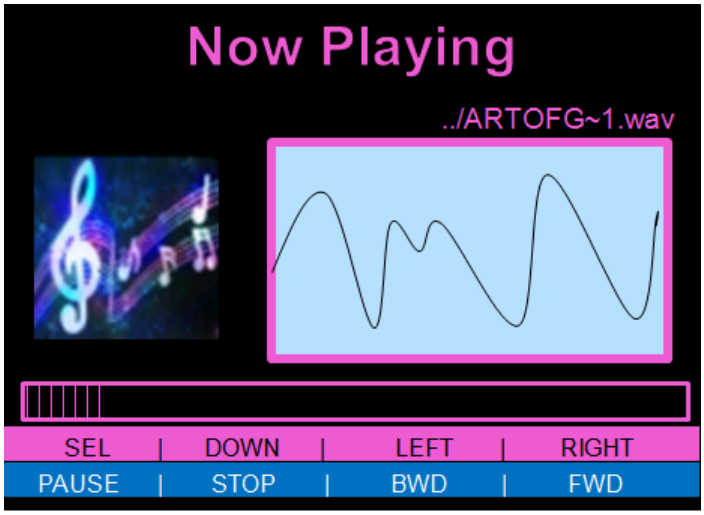
Figure 31. Wave Player interface



In *Figure 31*, the active push-buttons and their functions are displayed below. For example, at start-up, pressing the SEL JoyStick push-button causes the file to be played through the embedded speaker, whereas pressing the DOWN push-button causes to exit the Wave Player submenu.

Once the PLAY command is prompted, the submenu shown in *Figure 32* is displayed.

Figure 32. Wave Player Playing submenu



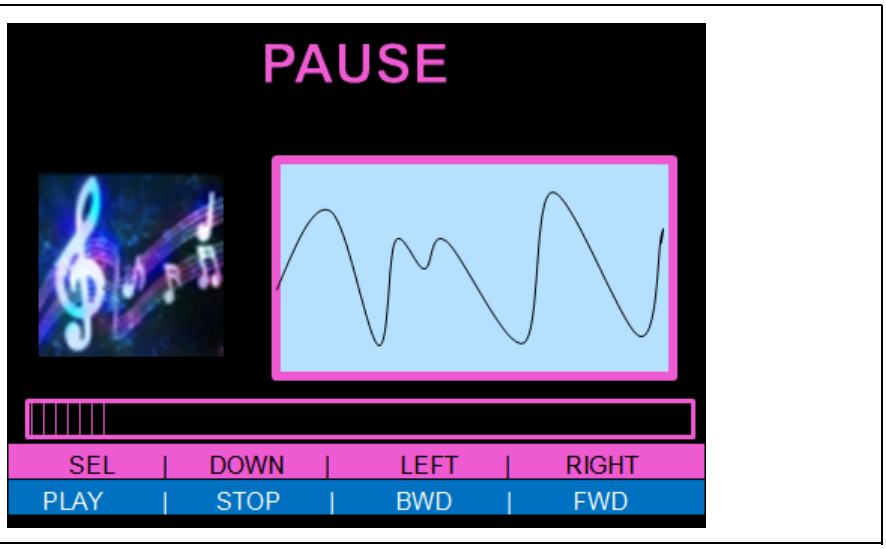
The Progress bar is displayed at the bottom of the view. The Progress bar is updated every ~1% of the audio file duration.

At this application level, pressing:

- the SEL push-button pauses the audio stream
- the LEFT push-button decrements the audio stream
- the RIGHT push-button increments the audio stream
- the DOWN push-button exits the wave player submenu

When the audio stream is paused, the menu in *Figure 33* is displayed.

Figure 33. Pause submenu



To resume playing, press the SEL push-button and the menu shown in *Figure 32* is displayed.

When the audio stream is stopped, the stream position is reset and the menu shown in *Figure 31* is displayed.

Note: The audio files provided within this package are based on a free music download from www.DanoSongs.com website.

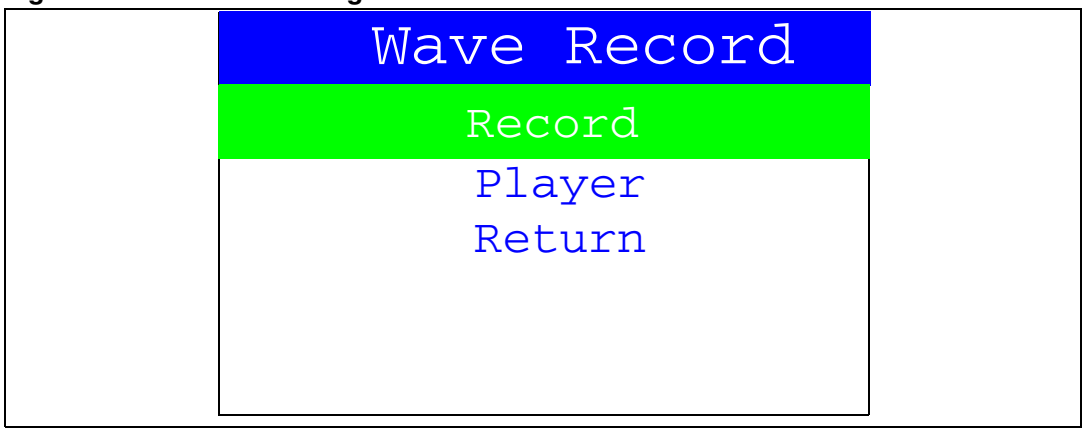
2.4.4 Wave Record

The STM32F051R8(T6) microcontroller has an embedded ADC and an embedded DAC which can be used to record a voice (signal coming from the U22 microphone).

In this demo, the recorded wave buffers can be transferred from the internal SRAM by block (512 bytes) and stored under the REC folder in the MicroSD card using the FatFs file system and based on DMA and SPI1. Timer3 (TIM3) triggers the ADC to generate the sampling frequency.

When the Wave Record submenu is selected, the message shown in *Figure 34* is displayed on the LCD.

Figure 34. Voice Recording submenu selected

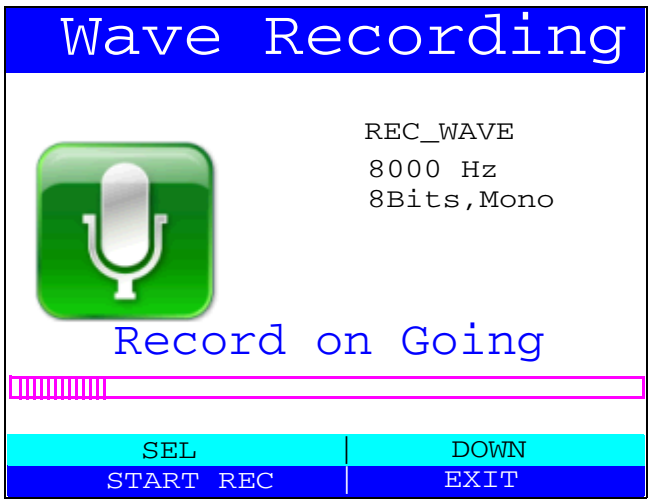


If the Record submenu has been selected by pressing the SEL push-button, the Voice Recording interface is displayed as shown in *Figure 35*.

Figure 35. Record submenu

Once the record command is activated (by pressing on SEL push button), the submenu shown in *Figure 36* is displayed.

Figure 36. Starting wave record



The recorded wave file proprieties at the right side of the Wave Record submenu.

When the DOWN push-button is pressed, the menu in *Figure 34* is displayed.

If the Wave Player submenu has been selected by pressing the SEL push-button, the wave player interface is displayed as shown in *Figure 31*.

Once the record command is activated (by pressing the SEL push-button), the submenu shown in *Figure 32* is displayed.

The Progress bar is updated every ~1% of the audio file duration.

At this application level, pressing:

- the SEL push-button pauses the audio stream
- the LEFT push-button decrements the audio stream
- the RIGHT push-button increments the audio stream
- the DOWN push-button exits the wave player submenu

When the audio stream is paused, the menu in *Figure 33* is displayed.

To resume playing, press the SEL push-button and the menu shown in *Figure 32* is displayed.

When the audio stream is stopped, the stream position is reset and the menu shown in *Figure 34* is displayed.

2.4.5 Low-power modes

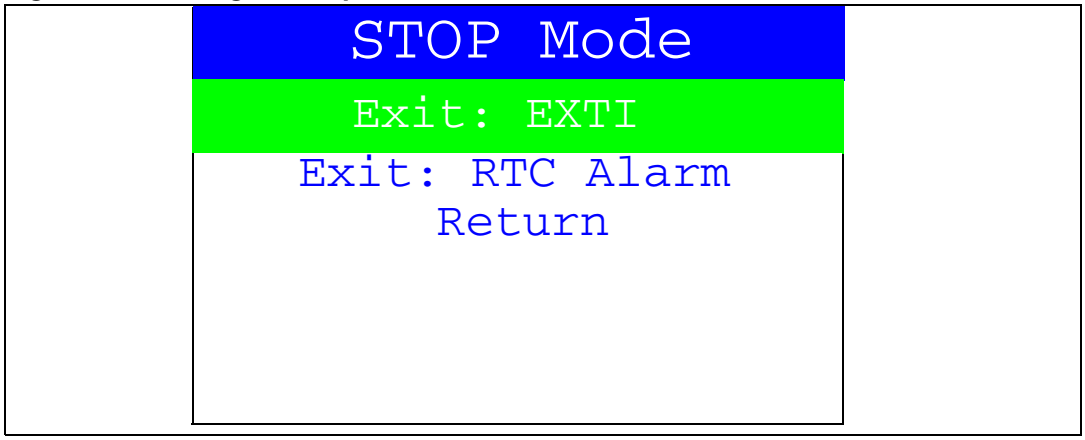
The STM32F051R8(T6) microcontroller provides different operating modes in which the power consumption is reduced. The purpose of this menu is to show the behavior of the microcontroller in different low-power modes. The Stop and Standby modes are taken as examples.

Stop mode menu

This menu allows the user to put the STM32F051R8(T6) in the Stop mode. The software performs the specific instruction sequence needed to enter the Stop mode.

In this application, the STM32F051R8(T6) can exit the Stop mode in two ways, as shown in *Figure 37*.

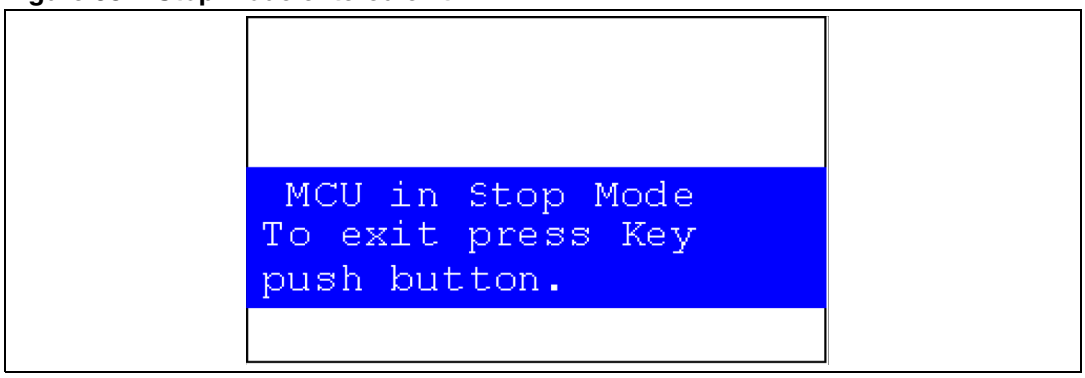
Figure 37. Exiting the Stop mode



- In the first case, The EXTI Key button is used to make the MCU exit the Stop mode.

  Once the **Stop mode** submenu has been selected, the red LEDs keep blinking until the SEL push-button is pressed, and the system enters the Stop mode. When the MCU is in the Stop mode, the message shown in *Figure 38* is displayed on the LCD.

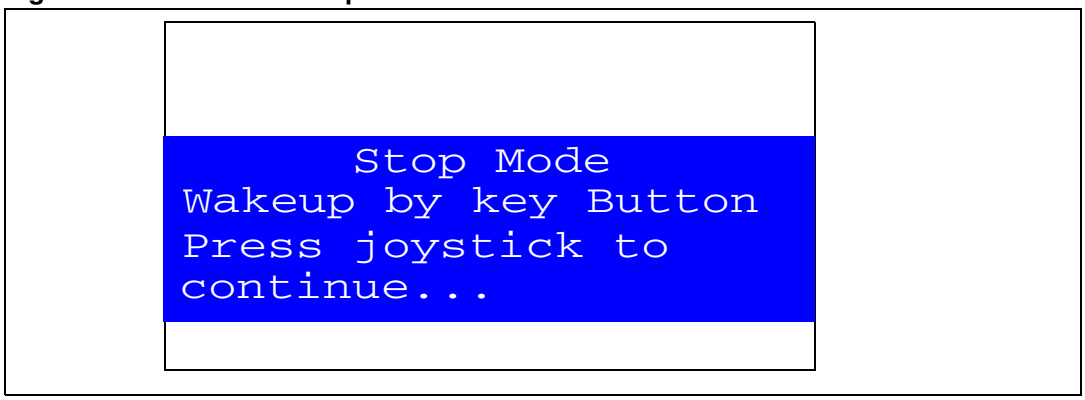
Figure 38. Stop mode entered exit EXTI



The MCU remains in the Stop mode until the Key push-button is pressed and the message shown in *Figure 39* is displayed on the LCD screen.

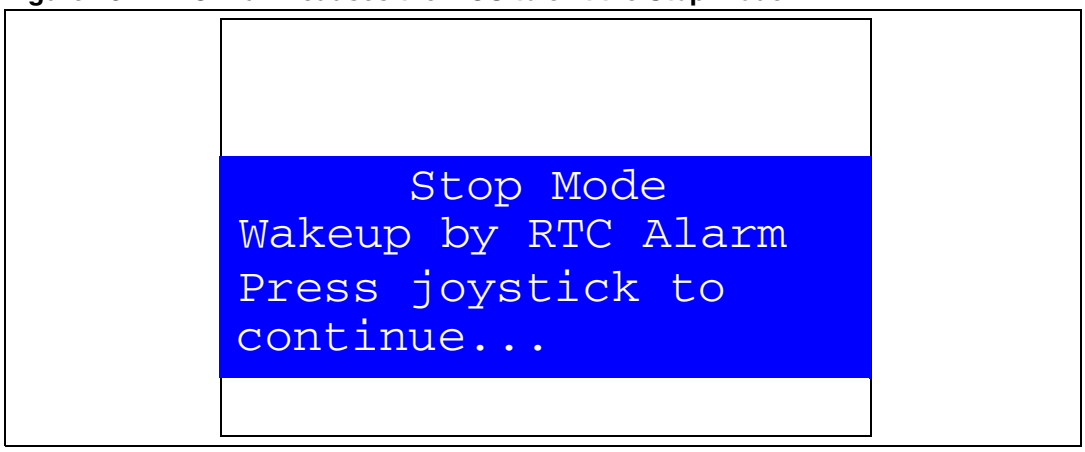
Once the Key push-button has been pressed, the MCU exits the Stop mode. The system clock is then set to 48 MHz and the application resumes execution.

Figure 39. MCU in the Stop mode Exit EXTI



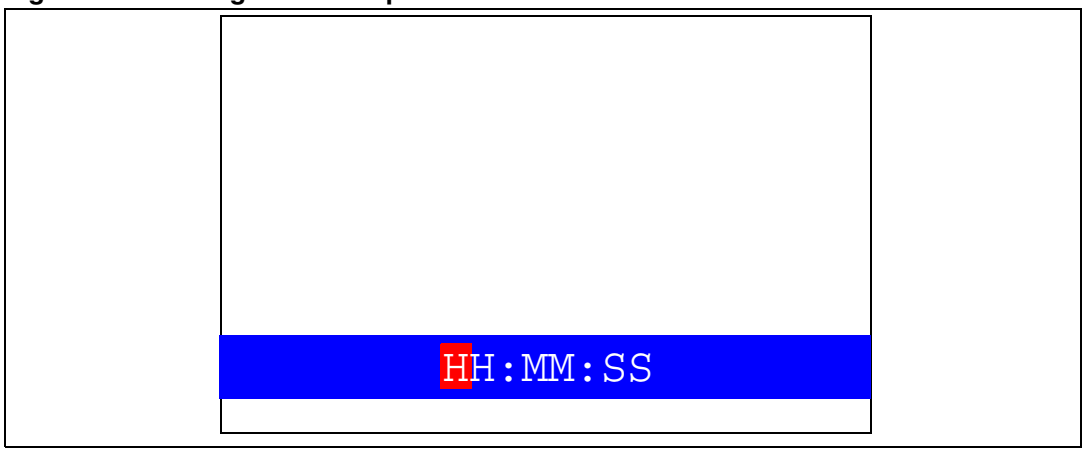
Note: if an RTC Alarm is generated while the MCU is in the Stop mode and the message shown in Figure 39 is displayed (which means that the Key push-button needs to be pressed to exit the Stop mode), the RTC Alarm causes the MCU to exit the Stop mode. The message shown in Figure 40 is then displayed.

Figure 40. RTC Alarm causes the MCU to exit the Stop mode



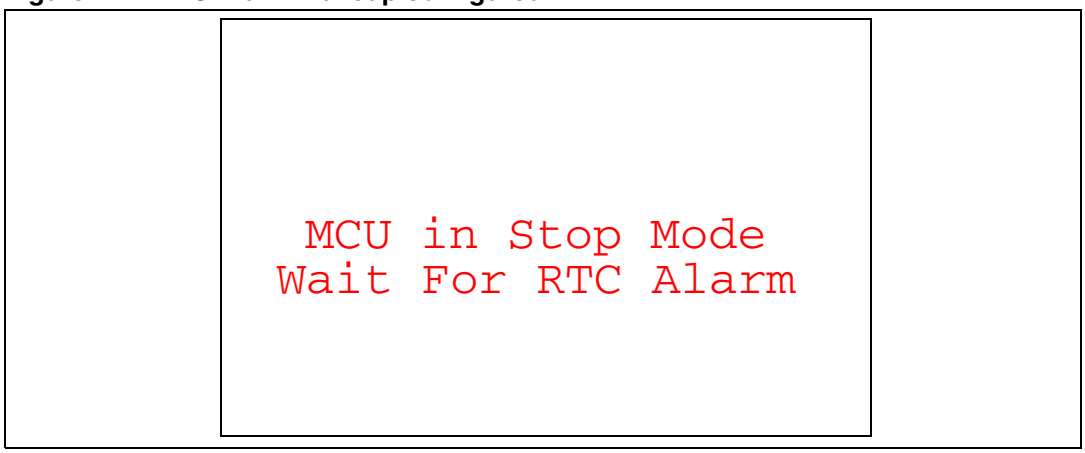
- In the second case, the RTC Alarm wakes up the MCU from the Stop mode after the programmed time has elapsed. When selecting this submenu, the user has to set the alarm to the time when the MCU is to exit the Stop mode. *Figure 41* shows how to set the wakeup time.

Figure 41. Setting the Wakeup time



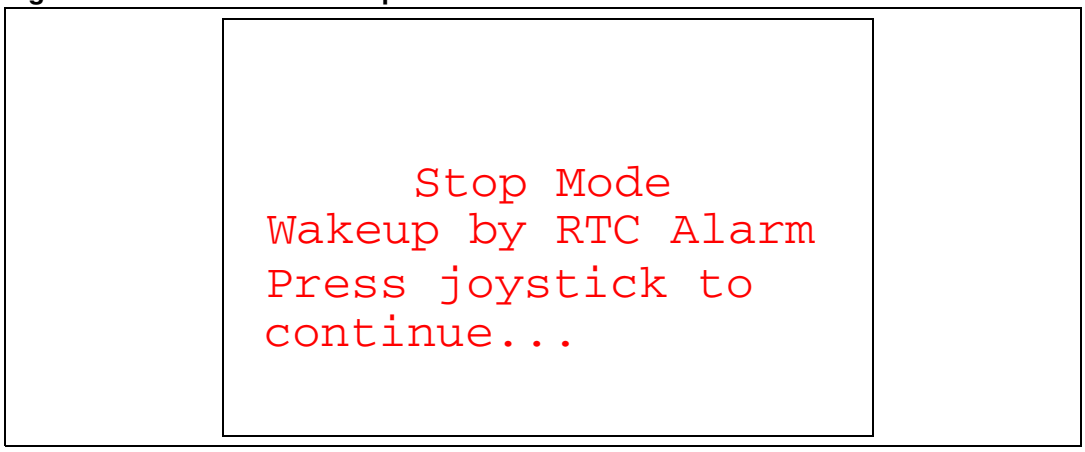
Once the alarm has been configured, the red LEDs stop blinking and the system enters the Stop mode. The message shown in *Figure 42* is displayed on the LCD.

Figure 42. RTC Alarm wakeup configured



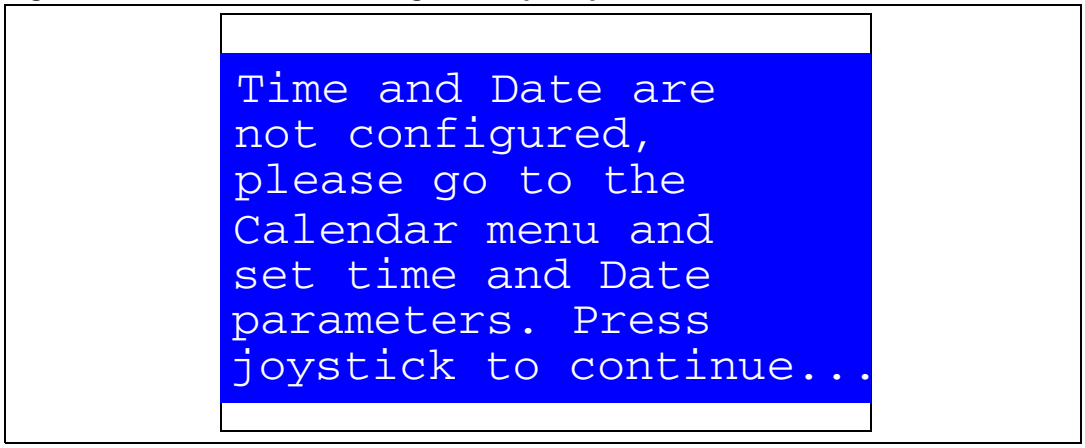
After the programmed time has elapsed, the system exits the Stop mode. The system clock is then set to 48 MHz and the application resumes execution. The message shown in *Figure 43* is displayed on the LCD screen.

Figure 43. RTC Alarm wakeup



Note: if the Time and Date have not been set, the message shown in Figure 44 is displayed on the LCD screen.

Figure 44. Time and Date configuration prompt

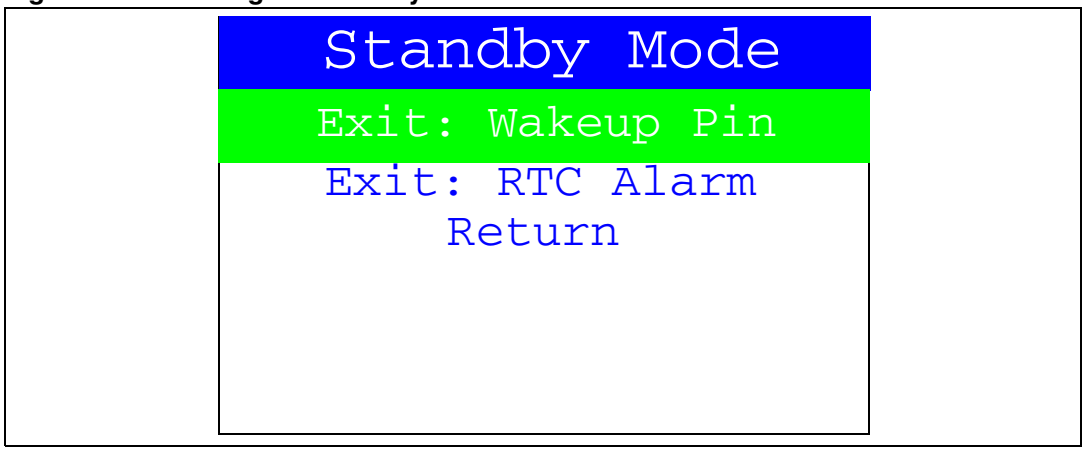


Standby mode menu

This menu allows the user to put the STM32F051R8(T6) in the Standby mode. The software runs the specific instruction sequence needed by the STM32F051R8(T6) to enter the Standby mode.

In this application, the STM32F051R8(T6) can exit the Standby mode in two ways, as shown in *Figure 45*.

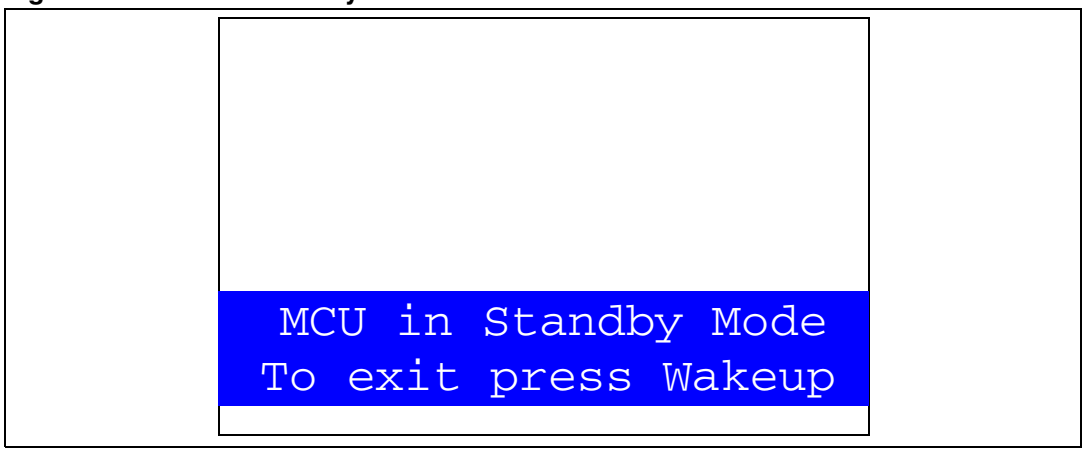
Figure 45. Entering the Standby mode



- In the first case, the Wakeup push-button is used to wake up the MCU from the Standby mode.

  Once the **Standby mode** submenu has been selected, the red LEDs keep blinking until the SEL push-button is pressed, and the system enters the Standby mode. When the MCU is in Standby mode, the message shown in *Figure 46* is displayed on the LCD.

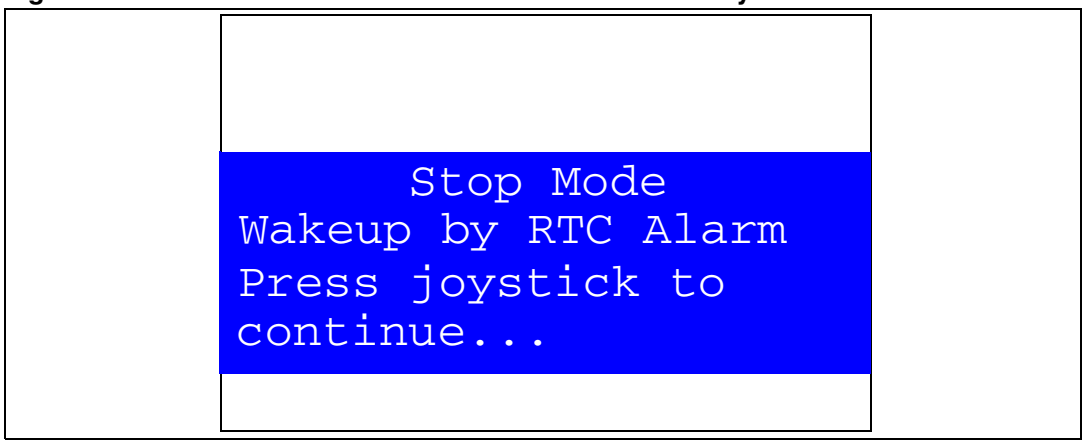
Figure 46. MCU in Standby mode



The MCU remains in Standby mode until the Wakeup push-button is pressed. Once the Wakeup push-button has been pressed, the MCU exits the Standby mode and the system reset signal is generated.

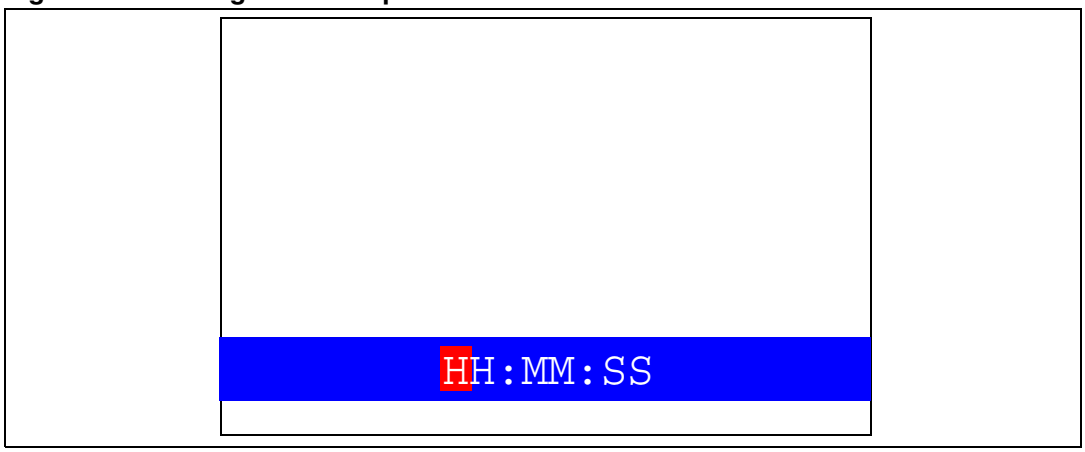
Note: if an RTC Alarm is generated while the MCU is in Standby mode and the message shown in Figure 46 is displayed (which means that the Wakeup push-button needs to be pressed to exit the Standby mode), the RTC Alarm causes the MCU to exit the Standby mode and a system reset signal is generated.

Figure 47. RTC Alarm causes the MCU to exit the Standby mode



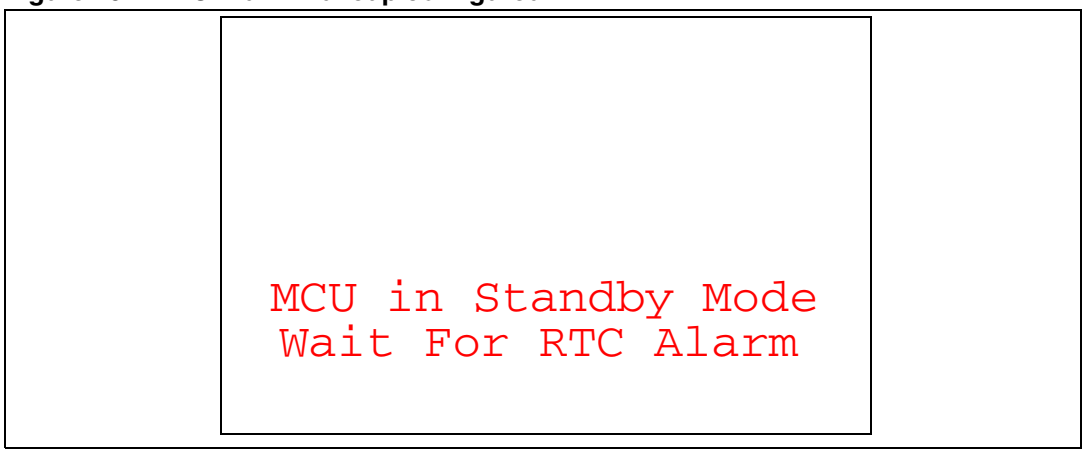
- In the second case, the RTC Alarm wakes up the MCU from the Standby mode after the programmed time has elapsed. When selecting this submenu, the user has to set the alarm to the time when the MCU is to exit the Standby mode. *Figure 48* shows how to set the wakeup time.

Figure 48. Setting the wakeup time



Once the alarm has been configured, The LEDs stop blinking and the system enters the Standby mode. The message shown in *Figure 49* is then displayed on the LCD.

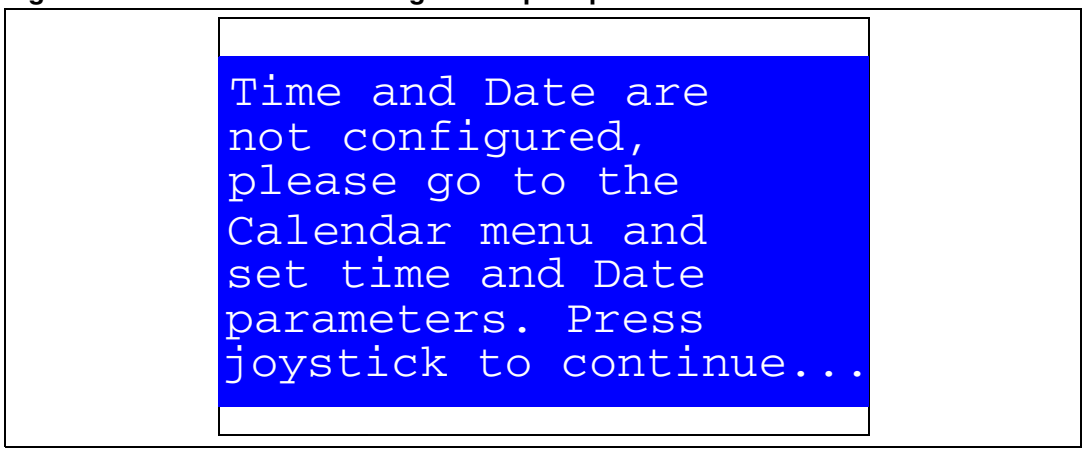
Figure 49. RTC Alarm wakeup configured



After the programmed timing has elapsed, the system exits the Standby mode and a system reset signal is generated.

Note: if the Time and Date have not been set, the message shown in Figure 50 is displayed on the LCD screen.

Figure 50. Time and Date configuration prompt

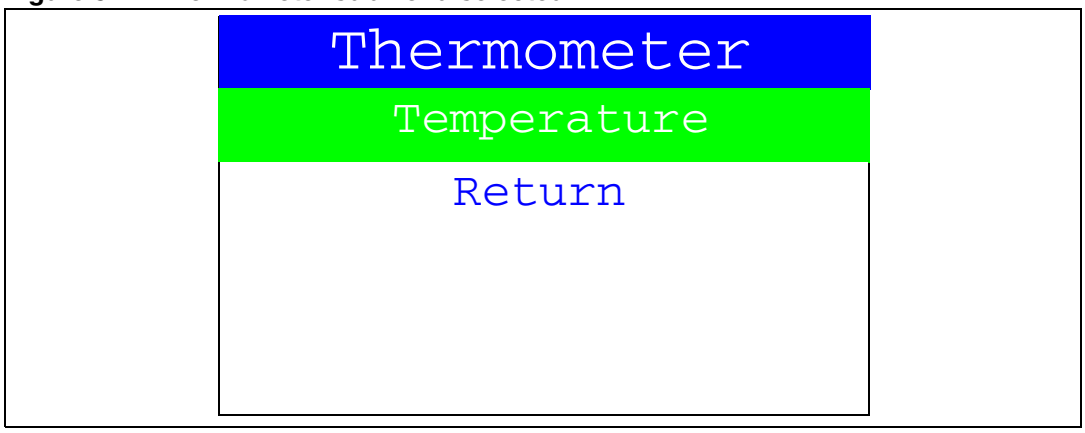


2.4.6 Thermometer

The STM32F051R8(T6) microcontroller has two embedded I^2C peripherals that can be connected to any device supporting the I^2C protocol including the system management bus (SMBus) mode. An STLM75 (or a compatible device) I^2C temperature sensor is mounted on the STM320518-EVAL board and used to capture the external temperature (-55°C to +125°C).

When the Thermometer submenu is selected, the message shown in *Figure 51* is displayed on the LCD.

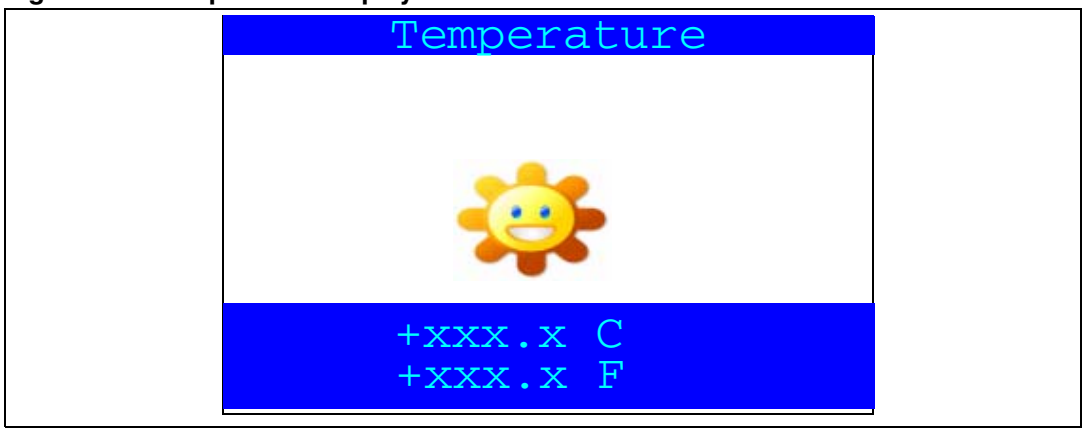
Figure 51. Thermometer submenu selected



Once the Temperature submenu has been selected by pressing the SEL push-button, the temperature value is displayed in Celsius and Fahrenheit as shown in *Figure 52*.

Press any key to return to the Thermometer submenu.

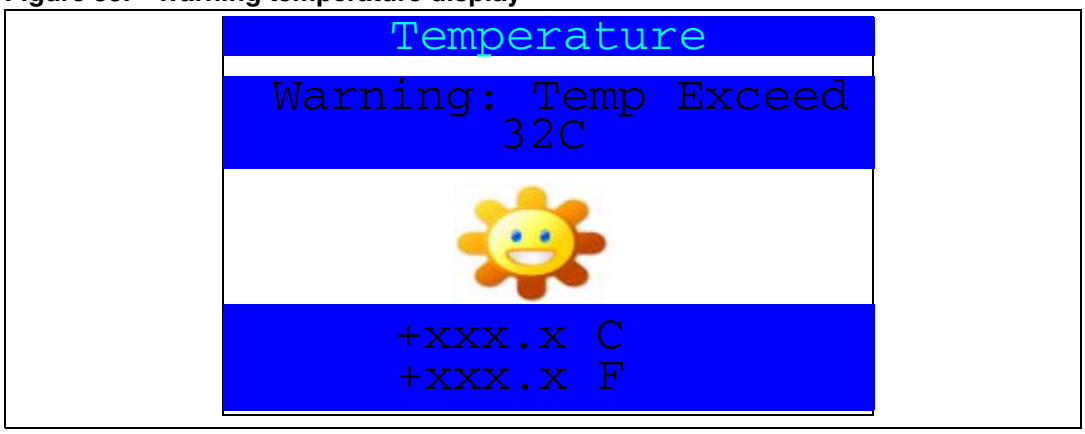
Figure 52. Temperature display



The temperature variations can be monitored easily using the STM32 I2C SMBus feature. This is managed by the SMBus Alert, which generates a dedicated interrupt informing the system that the temperature is out of the selected ranges. This can be very useful when a higher temperature needs an emergency action, as is the case in critical systems (motor control, medical...).

If the temperature exceeds the over-limit high (TEMPERATURE_TOS: Over Limit Temperature), the SMBus alert interrupt is generated and the warning message shown in *Figure 53* is displayed on the LCD screen.

Figure 53. Warning temperature display



The messages shown in *Figure 52* are displayed on the LCD screen when the temperature goes under the over limit low (TEMPERATURE_THYS: Hysteresis Temperature).

The user can configure the TOS and THYS using dedicated define values in the code. By default, the STM320518-EVAL demo sets them to (see menu.c file):

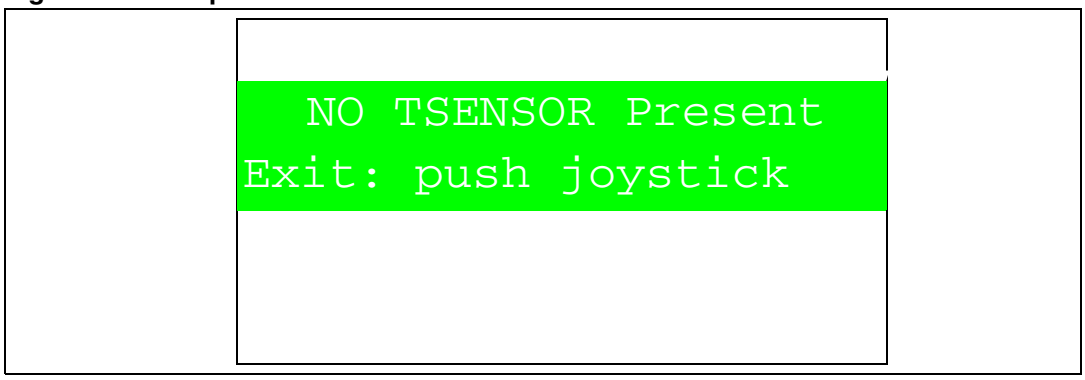
```
#define TEMPERATURE_THYS 31

#define TEMPERATURE_TOS 32
```

Press any key to return to the Thermometer submenu.

Note: Any hardware trouble with the temperature sensor is detected by a test. In such a case, the message shown in Figure 54 is displayed.

Figure 54. Temperature sensor error



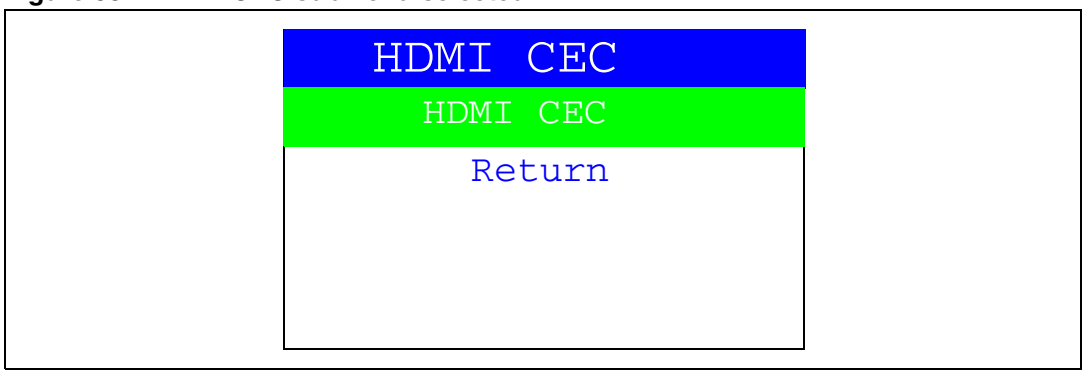
2.4.7 HDMI™ CEC submenu

The STM32F051R8(T6) microcontroller features an HDMI-CEC peripheral. The demonstration shows how to configure this peripheral and how to create a CEC network providing a high level communication between different devices using CEC protocol messages.

For more details, refer to application note AN4066: “Developing an HDMI-CEC network using an STM32F0xx microcontroller”. This application note provides a full description of the STM32F051R8(T6) embedded HDMI-CEC Controller and a step by step firmware description of the CEC peripheral configuration. An advanced demonstration firmware communicating in a real multimedia and HDMI environment is also provided to build easily the CEC applications.

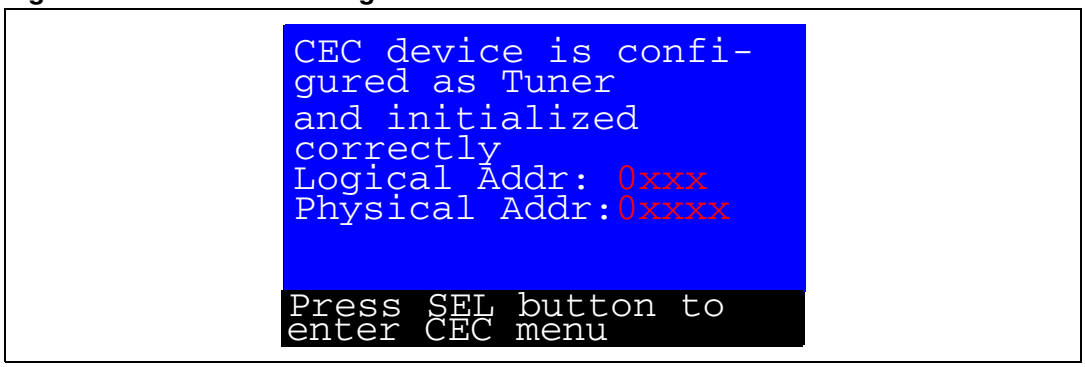
When the **HDMI CEC** submenu is selected, the message shown in *Figure 55* is displayed on the LCD.

Figure 55. HDMI CEC submenu selected



Once you select the **HDMI CEC** submenu, if no CEC error is generated, the device is configured as Tuner and the physical and logical addresses are displayed on the LCD as shown in *Figure 56*. To enter the CEC menu, press the SEL push-button.

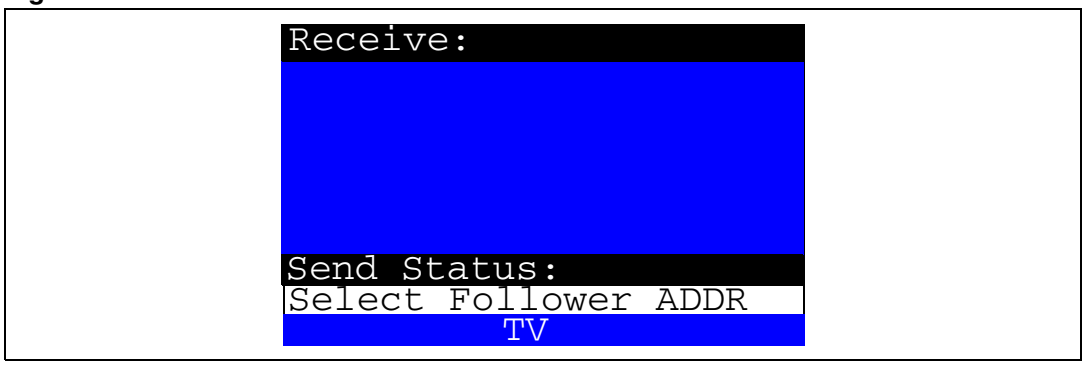
Figure 56. HDMI CEC configuration submenu



The LCD screen is divided into two parts as shown in *Figure 57*:

- a subscreen that shows the CEC receive information: receive status, sender address
- a subscreen that allows to select the follower address and the command to send

Figure 57. CEC menu



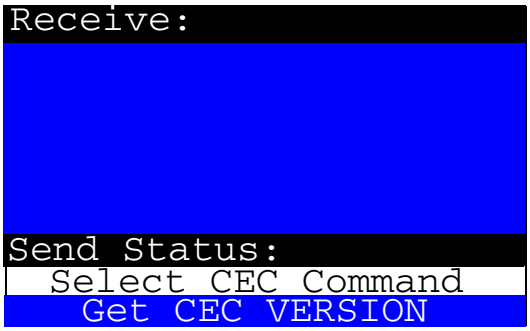
After selecting the follower address, select the command to be sent to the selected follower address using the LEFT, RIGHT and SEL buttons. After selecting the command, the CEC device sends this command to the selected follower address and displays the status of transmission as shown in *Figure 58.*

You can also send the command from the remote control by the CEC device. After selecting the address, the user presses the remote control and the message is displayed in the receiver field.

Note: 1 Phillips RC5 protocol was integrated in the HDMI-CEC application. Each RC5 command has a corresponding HDMI-CEC User Control Code.

2 For more details, refer to Section 2.4.10: InfraRed.

Figure 58. Select CEC command

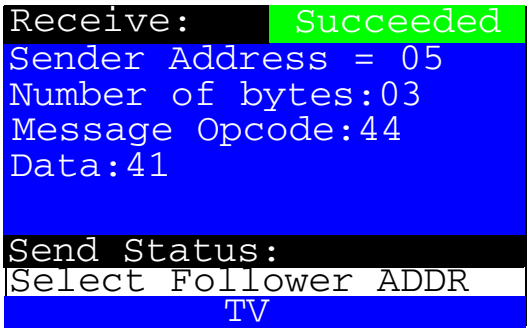


When receiving a new message, the following information can be displayed on the LCD:

- Receive status
- Sender address
- Number of bytes (including the sender address)
- Opcode message
- Data (operands)

Figure 59 shows that the device has correctly received the frame from the sender with the address: 0x5, the number of bytes received: 0x3 (header + opcode + data), the message opcode: 0x44 and data: 0x41.

Figure 59. Receive subscreen information



Normally, for the Standby command, the device is in stop mode and can wake up only when it receives a new command. If you press the Tamper button, the HDMI CEC demo stops and the menu shown in *Figure 55* is displayed.

Note: The STM32F051R8(T6) *CEC device responds only to the following commands. For other commands it sends feature abort:*

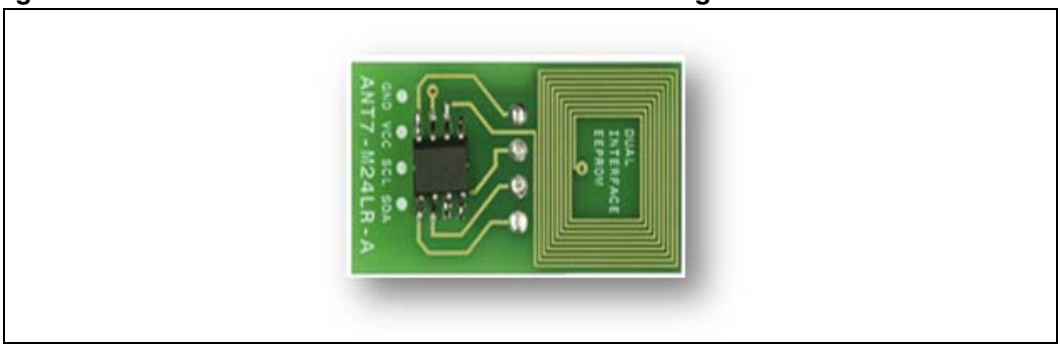
- *Standby*
- *Get CEC version*
- *Give physical address*
- *Give OSD name*

2.4.8 Dual interface EEPROM (RF/I²C)

The STM32L152ZD(T6) microcontroller has two embedded I2C peripherals that can be connected to any device supporting the I2C protocol. An ANT7-M24LR-A dual interface RF EEPROM daughter board can be connected on CN12 to STM320518-EVAL via the I2C interface.

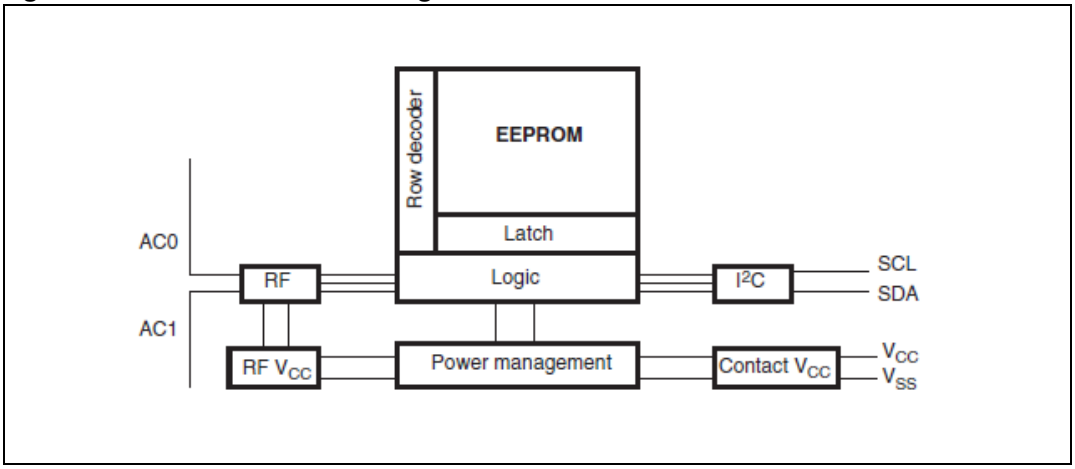
Note: For more details about the Dual Interface EEPROM daughter board, please refer to the ANT7-M24LR-A databrief.

Figure 60. ANT7-M24LR-A dual interface EEPROM daughter board



The M24LR64-R device is a dual-interface, electrically erasable programmable memory (EEPROM). It features an I2C interface and can be operated from a VCC power supply. It is also a contactless memory powered by the received carrier electromagnetic wave at 13.56 MHZ. The M24LR64-R is organized as 8192 × 8 bits in the I2C mode and as 2048 × 32 bits in the ISO 15693 and ISO 18000-3 mode 1 RF mode.

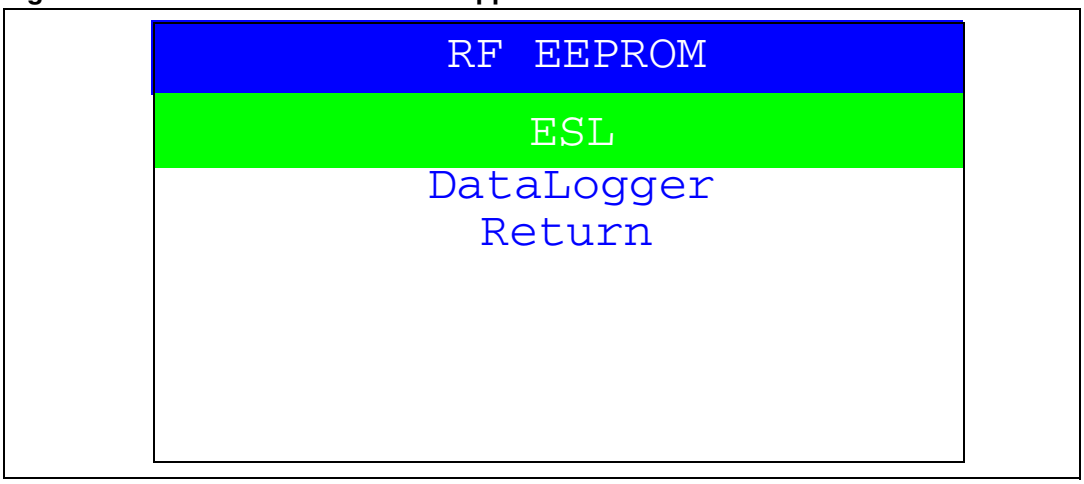
Figure 61. M24LR64-R block diagram



Note: For more details on the M24LR64 EEPROM, please refer to the datasheet.

Select the Dual Interface (RF/I²C) EEPROM menu by pressing SEL from the main menu. Two applications can be selected, as shown in *Figure 62.*

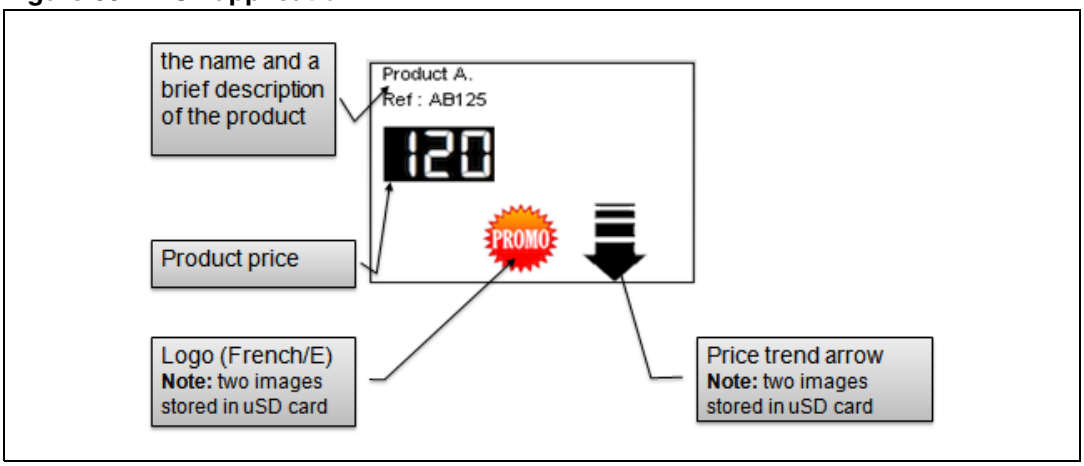
Figure 62. Dual Interface EEPROM applications menu



ESL application

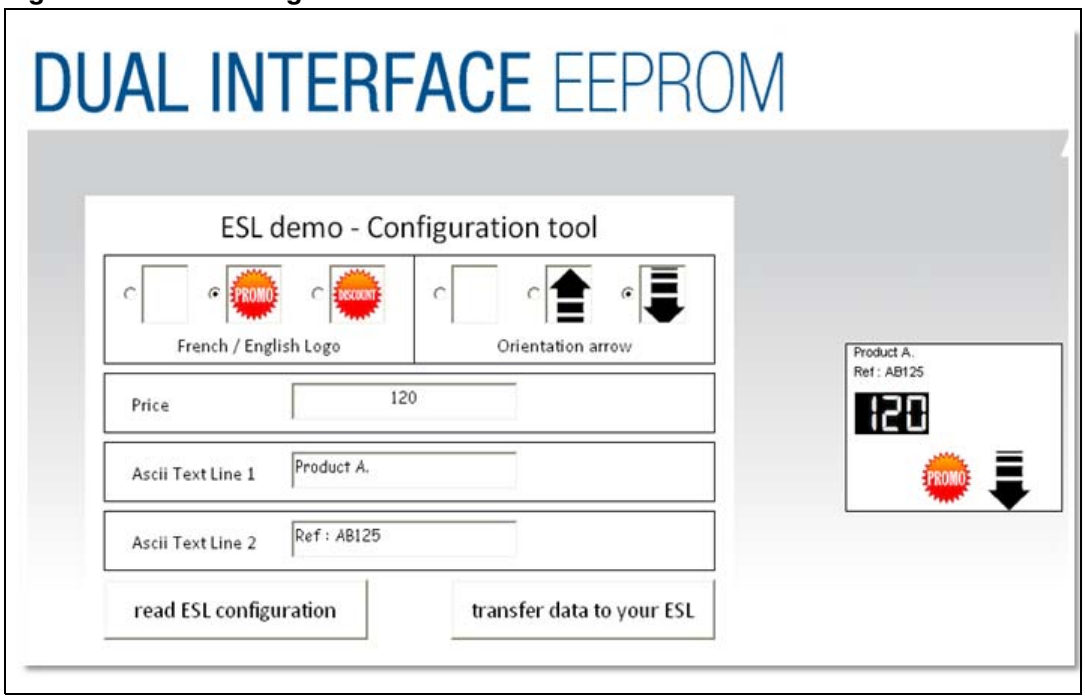
The Electronic Shelf Labeling (ESL) consists of reading (using I2C interface) the content of the dual interface EEPROM and displaying the information on the LCD screen, as shown in *Figure 63*.

Figure 63. ESL application



All this information (product name, price, logo, price trend...) is programed in the M24LR64 dual interface EEPROM in RF mode and can be updated any time using the CR95HF RFID reader/writer connected to the USB port of a PC host, with the associated M24LRxx_Application_Software PC software.

Figure 64. ESL setting menu



Using this PC software application, several parameters can be set to configure the ESL device and update the LCD display mounted on the STM320518-EVAL:

French / English Logo: Check the French (PROMO) or English (DISCOUNT) to indicate that a special price is proposed. Check the blank logo if no special price is applied.

Orientation arrow: Check the up or down arrow to indicate if the price has been increased or decreased, or the blank arrow if no indication is required.

Price: Display the product price.

Ascii Text line 1 and Line 2: Display the name and a brief description of the product. The transfer data to your ESL button of the PC ESL demo software allows to program ESL parameters in RF mode using the CR95HF RFID reader/writer demo board.

ESL data can be read from the M24LR64 dual interface EEPROM using the **read ESL configuration** button of the PC ESL demo software application. When done, ESL data written into the M24LR64 is displayed in the ESL panel of the PC ESL demo application.

*Note: For more details about this PC Software, please refer to user manual **UM0853**.*

ESL parameters mapping into the M24LR64 dual interface EEPROM are shown in *Table 5: M24LR64 memory organization (ESL application)*.

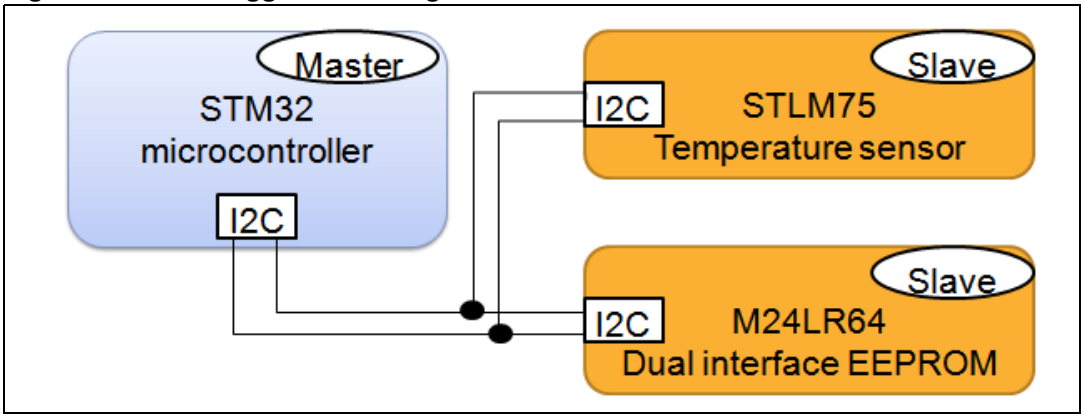
Table 5. M24LR64 memory organization (ESL application)

RF block address	bit[31:24]	bit[23:16]	bit[15:8]	bit[7:0]
0000h		Arrow	Icon	
0001h	Contains XYYYYYYYh X = number of digits (max 7) in price YYYYY = price value including decimal separator (.) coded in hexadecimal For instance, 512A99FFh means 5 digits and the price is 12.99			
0002h	Contains the ASCII codes of the string chain called ASCII TEXT LINE 1 in M24LR64 ESL demo software. The maximum size of this text line is 20 characters			
0003h				
0004h				
0005h				
0006h				
0007h	Contains the ASCII codes of the string chain called ASCII TEXT LINE 2 in M24LR64 ESL demo software. The maximum size of this text line is 20 characters			
0008h				
0009h				
000Ah				
000Bh				
000Ch	XXXX XX is the checksum of bytes from address 0000h to 000Bh YY: reserved ZZ: reserved			

DataLogger application

This dual interface EEPROM (RF/I²C) application presents a practical application for the M24LR64. It is able to record and store a 64-Kbit temperature data using the M24LR64-R dual interface EEPROM. The DataLogger microcontroller is an STM32L152ZD (T6) which communicates with the M24LR64-R using its I²C serial interface and controls an STLM75 digital temperature sensor.

Figure 65. DataLogger block diagram



Temperature acquisition parameters (temperature acquisition Start/Stop, temperature acquisition period) are stored into the M24LR64 dual interface EEPROM. Changing these values and controlling the temperature acquisition is done in RF mode using the CR95HF RFID reader/writer demo board. During the temperature acquisition, data from the STLM75 digital temperature sensor is written into the M24LR64 by the STM32L152ZD microcontroller in I²C mode and displayed on the STM32L252 eval board LCD screen.

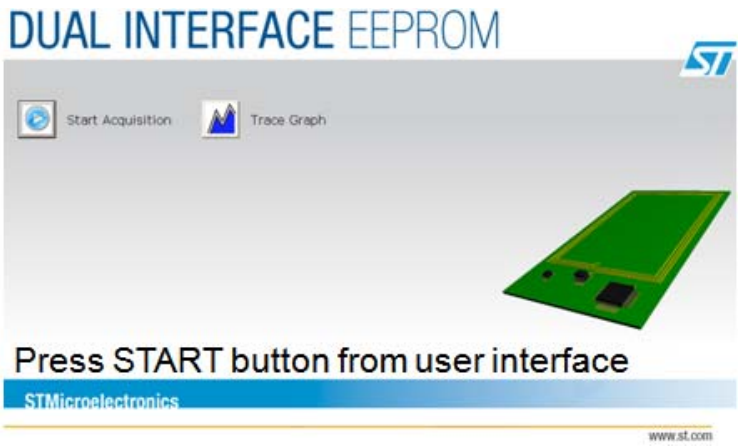
Temperature data can be read in the same time in RF mode using the CR95HF RFID reader/writer demo board and be displayed in the PC DataLogger software application.

For more details about this application, please refer to the application note AN 3109 “Developing your M24LR64-R datalogger application for temperature acquisition“.

How to run the application:

1. From the STM32L125D evaluation board, select the DataLogger submenu and then press the SEL push-button.

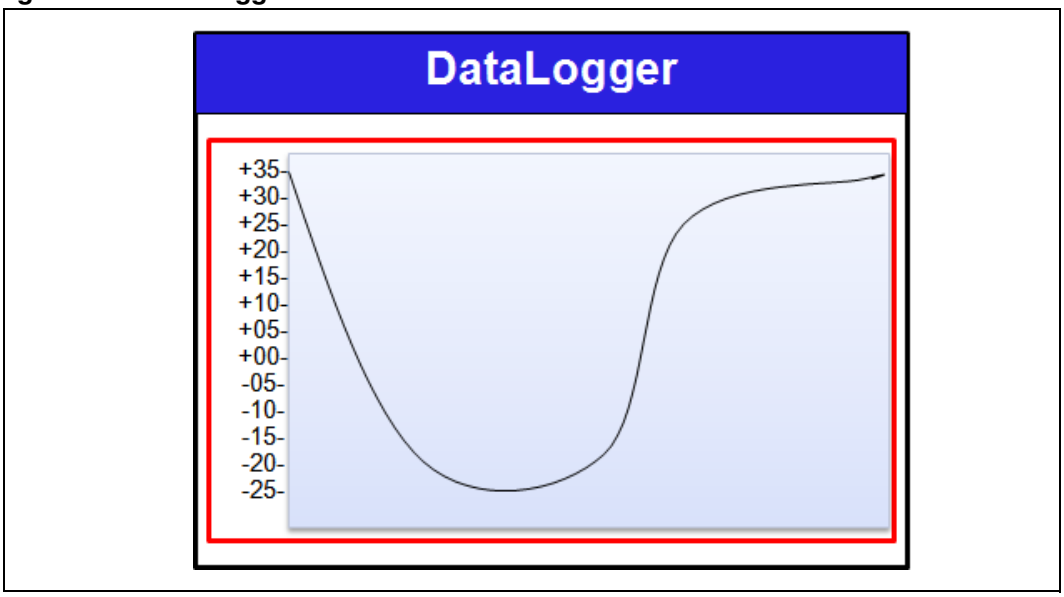
Figure 66. DataLogger dialog box



2. **From the application software side, choose “DataLogger” application and then follow the steps described on the user manual, UM0925.**

When the application starts, the data has been stored in the dual interface EEPROM and this temperature data has been displayed as a real-time curve on the TFT-LCD mounted on the evaluation board and also on the PC software interface.

Figure 67. DataLogger curve



Reference documents:

a) M24LR64-R datasheet
b) M24LR64-R tool driver install guide user manual (UM0863)
c) Using the M24LR64-R datalogger reference design user manual (UM0925)
d) How to manage M24LR64-R data transfers from the I²C bus or an RF channel
e) Application note (AN3057)
f) Application note (AN3109)
g) ANT7-M24LR-A data brief

For more information about dual interface EEPROM memories and associated demo boards and documentation, visit the dual interface EEPPROM page on www.st.com.

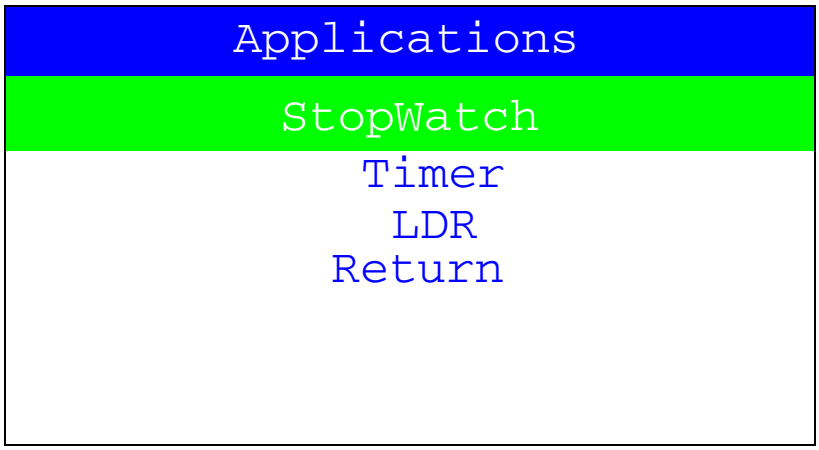
For more information about the CR95HF multi protocol RFID transceiver technical documentation and associated resources, visit the CR95HF web page on www.st.com.

2.4.9 Applications

The STM32F051R8(T6) microcontroller has a new embedded RTC Version 2 which can be used for many applications like Stopwatch and Timer.

Select Applications to display the submenu shown in *Figure 68*.

Figure 68. Applications menu



Stopwatch application

This application simulates a precise chronometer with 5 record time possibilities. For this application, an interactive human interface is developed using STM328L0518-EVAL LCD and push-buttons to allow the user to use StopWatch with real time display.

Figure 69. StopWatch submenu

After startup, a default 00:00:00:000 chronometer counter is displayed on the LCD; it correspond to [Hours]:[minutes]:[seconds]:[milliseconds].

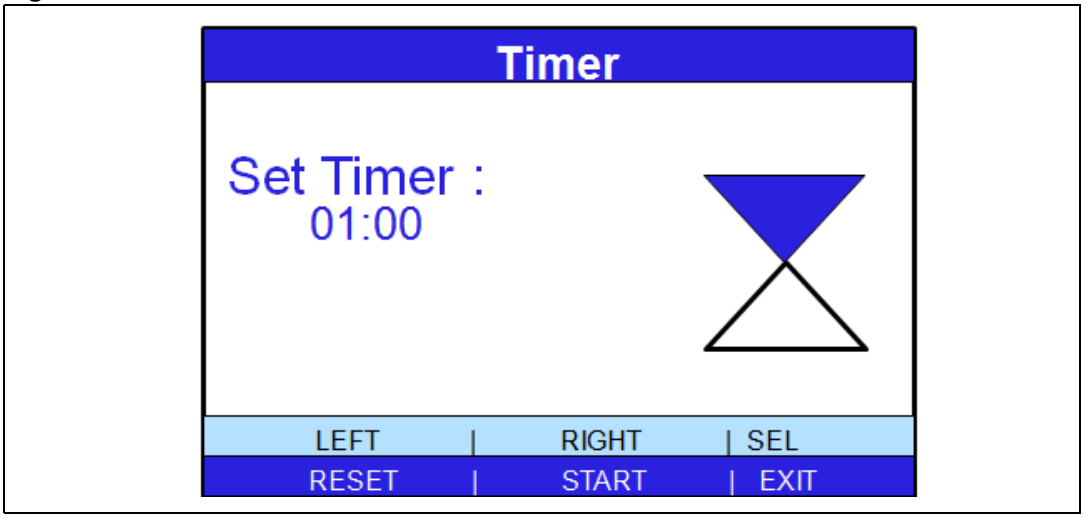
The user can manipulate the chronometer features using the joystick LEFT, RIGHT and DOWN push-buttons:

- press the joystick LEFT button to start counter.
- press the joystick RIGHT button to save trials in the backup registers (max 5 actions).
- Press the joystick DOWN button to exit.
- press and hold the Tamper button for 1 sec to reset all the backup registers.

Timer application

This application simulates a precise Hourglass. It measures the passage of a few sub seconds, seconds or minutes. For this application, an interactive human interface is developed using STM32F0518-EVAL LCD and push-buttons to allow the user to use StopWatch with real time display. After startup, the Hourglass is full and the default timer is one minute. Using the UP and DOWN push-buttons, the user can change the Timer duration (the timer duration must be higher than 13 seconds).

Figure 70. Timer submenu



The user can manipulate the Hourglass features using the joystick LEFT, RIGHT, UP and DOWN buttons:

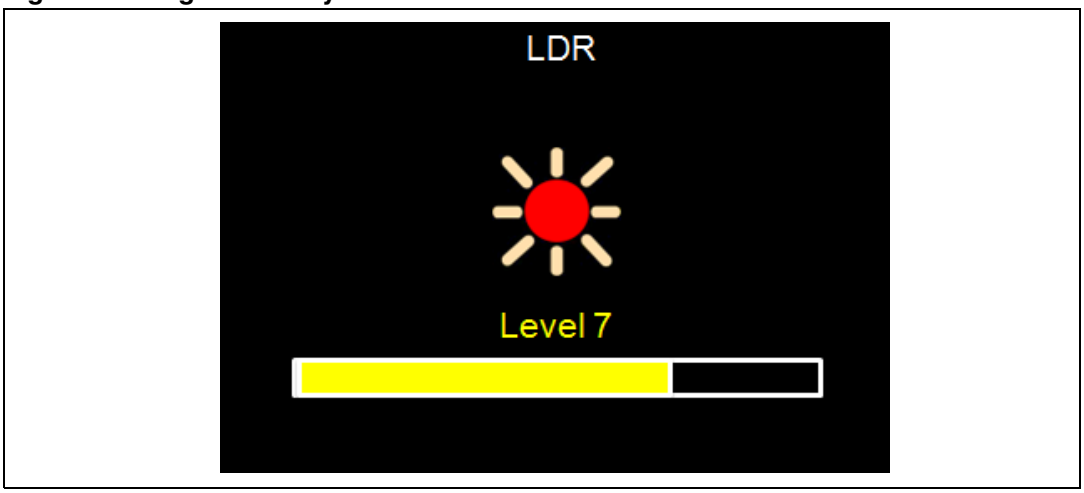
- press the joystick RIGHT button to start the timer.
- press again the joystick RIGHT button to stop the timer.
- press the joystick LEFT button to reset the timer.
- press the joystick UP/DOWN button to adjust the timer (UP and DOWN are only used when the Timer is reset).

LDR application

The STM32F051R8(T6) microcontroller has an embedded comparator which can be used in combination with the DAC peripheral for more applications, such as the measuring of light intensity (using the LDR sensor).

When the LDR submenu is selected, the message shown in *Figure 71* is displayed.

Figure 71. Light Intensity Level



According to the light intensity level, the icons are changed and also the progress bar.

2.4.10 InfraRed

The infrared technology offers important advantages as a form of wireless communication.

Nowadays, almost all audio and video equipment can be controlled using an infrared remote control.

There are many popular infrared protocols used to transmit data via infrared light, such as RC5, SIRC,...

The purpose of this demo is to provide a generic solution for implementing an IR transmitter and receiver in software using an STM32F051R8(T6) microcontroller. An example of a software implementation is provided for the free RC5 and SIRC protocol specification. Other protocols can be supported upon request (for further information, contact your local STMicroelectronics sales office).

For more details, refer to application note AN4099: “Implementing transmitters and receivers for infrared remote control protocols using STM32F0xx microcontrollers”.

Note: The infrared protocol implementation is based on free RC5 and SIRC specifications download from:

http://www.sbprojects.com/knowledge/ir/rc5.php

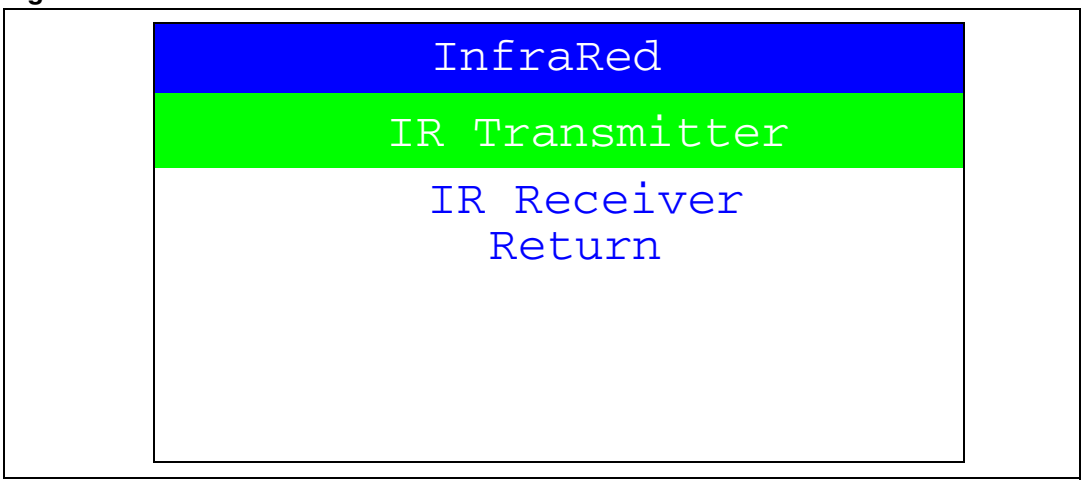
http://www.sbprojects.com/knowledge/ir/sirc.php

The STM32F051R8(T6) microcontroller has embedded Timers which can be used as follows:

- One 32-bit and one 16-bit timer, with up to 4 IC/OC, usable for IR control decoding
- two 16-bit timers, each with IC/OC and OCN, dead time generation, emergency stop and modulator gate for IR encoder.

Select InfraRed to display the submenu shown in *Figure 72*.

Figure 72. InfraRed menu

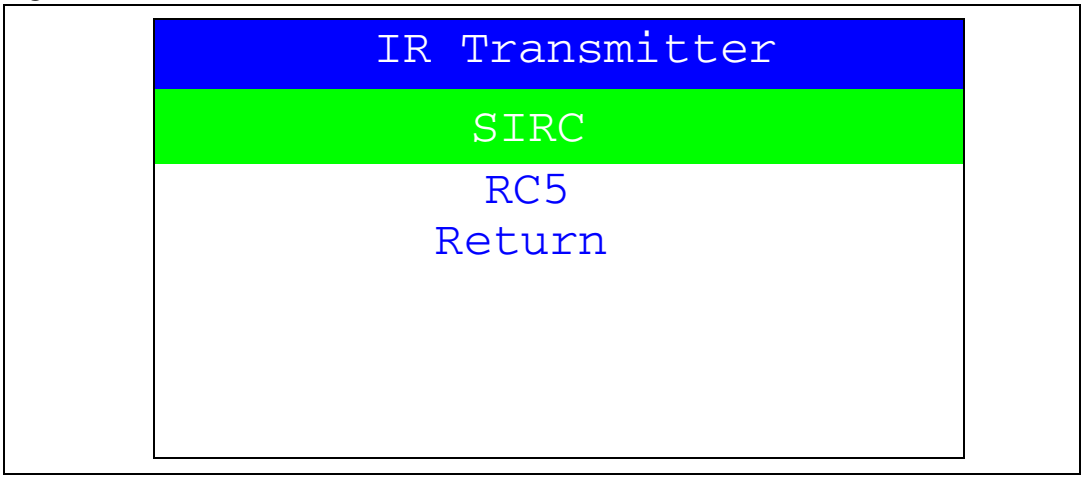


Infrared transmitter

On the STM32F0518-EVAL board, the IR LED is driven by PB9 directly with 20mA current.

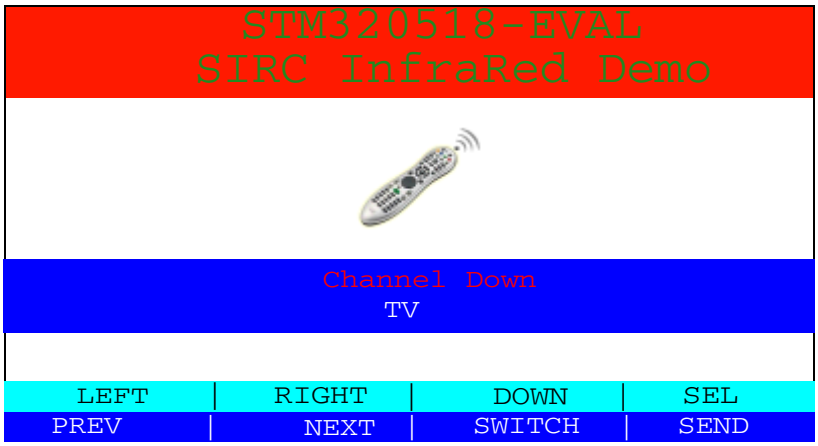
When the IR Transmitter submenu is selected, the message shown in *Figure 73* is displayed.

Figure 73. IR Transmitter menu



When a protocol has been selected from the list, the adequate submenu shown in *Figure 74* is displayed.

Figure 74. IR transmitter command menu



The user can manipulate the InfraRed transmitter features using the joystick LEFT, RIGHT, DOWN, UP and SEL buttons:

- press the joystick DOWN button to switch between device type and command.
- press the joystick LEFT button to go to the previously defined device/command.
- press the joystick RIGHT button to go to the next defined device/command.
- press the joystick SEL button to send the selected device command code.

When the joystick UP button is pressed, the infrared transmitter demo is exited and the menu shown in *Figure 73* is displayed.

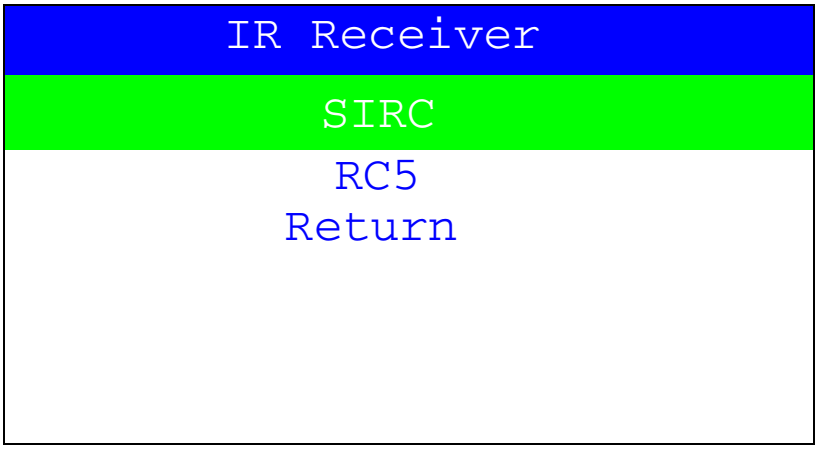
Infrared receiver

TSOP34836 IR receiver is connected to PB3 of STM32F051R8(T6) on the STM32F0518-EVAL board.

At the receiving end, a receiver detects the light pulses, which are processed to retrieve/decode the information they contain.

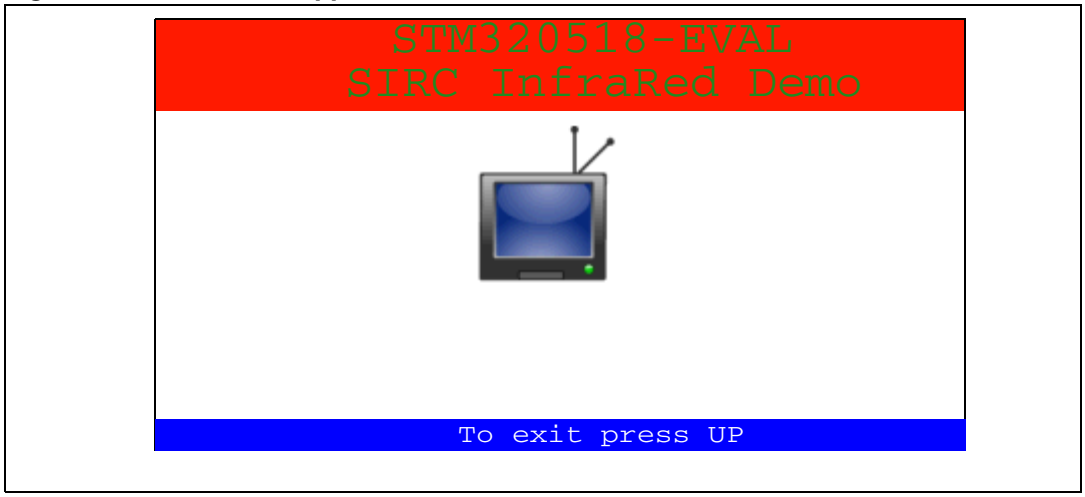
When the IR Receiver submenu is selected, the message shown in *Figure 75* is displayed.

Figure 75. IR receiver menu



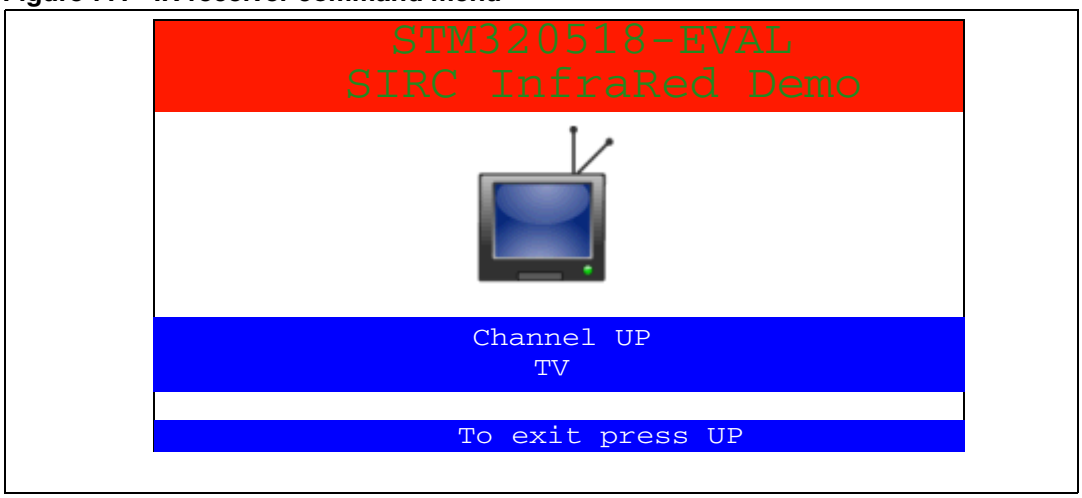
When a protocol has been selected from the list, the adequate submenu shown in *Figure 76* is displayed.

Figure 76. IR receiver application menu



When an IR frame is sent using a remote control or using the IR transmitter application running on another STM32F0518-EVAL board, this IR frame will be decoded using the IR receiver. If the protocol is compliant with the selected IR protocol, the device and the command will be displayed on the LCD screen *Figure 77.*

Figure 77. IR receiver command menu



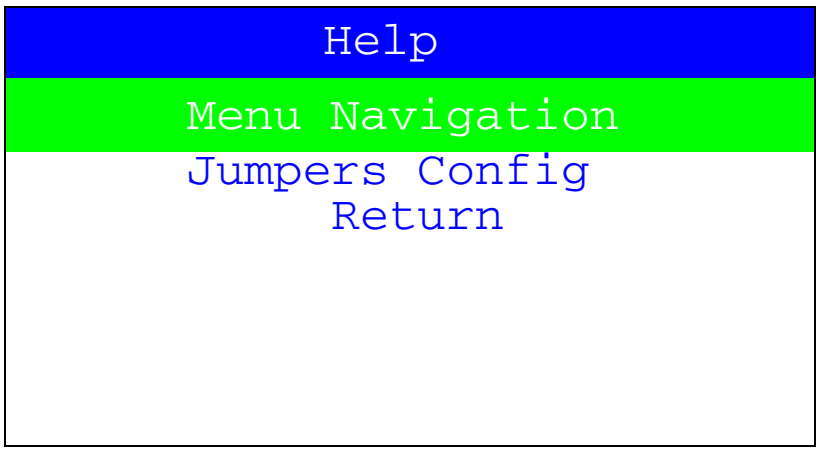
Note: For more details on the infrared receiver implementation, refer to AN3174 application note: "implementing infrared remote protocols receiver with the STM32F10xx microcontrollers". This application note provides a full description of the IR decoding.

2.4.11 Help

This submenu can help the user to configure the jumpers on the STM32F0518-EVAL evaluation board and to navigate between the menus and submenus available in the firmware demo.

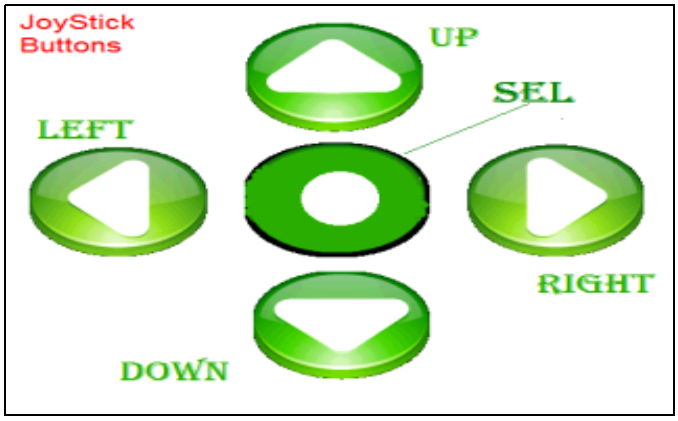
Select the Help menu by pressing SEL from the main menu. The message shown in *Figure 78* is then displayed on the LCD screen.

Figure 78. Help menu



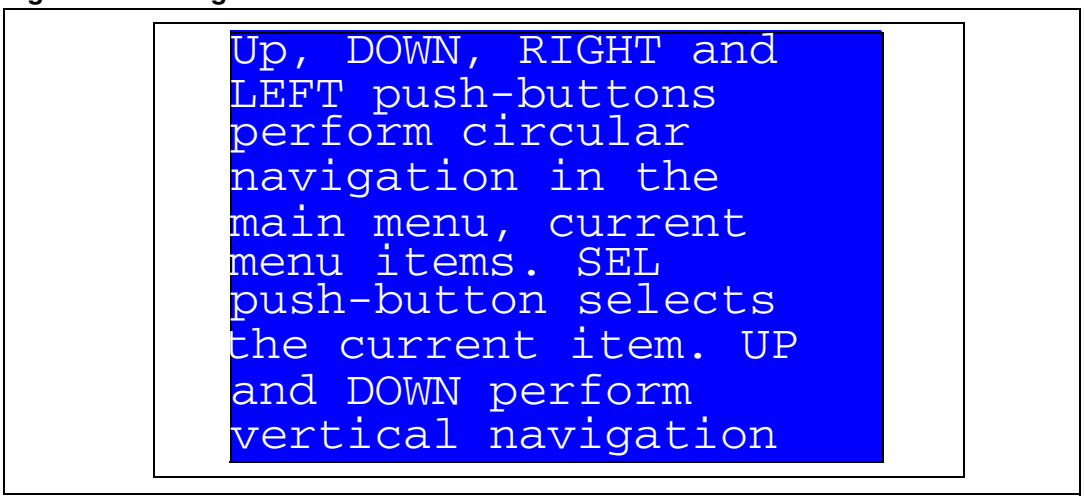
- If the Menu Navigation submenu is selected, the message shown in *Figure 79* is displayed.

Figure 79. Navigation menu-1



When any joystick button is pressed, the second navigation interface is displayed as shown in *Figure 80.*

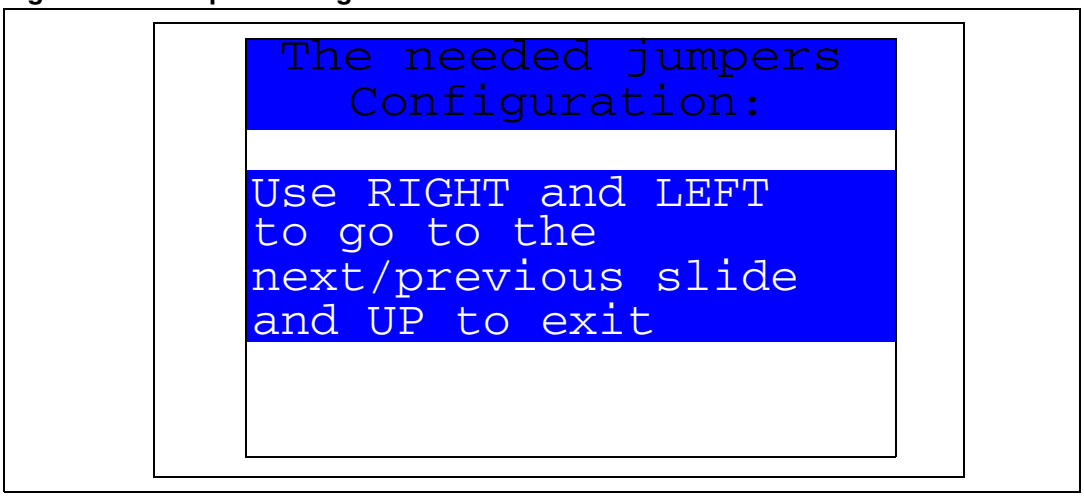
Figure 80. Navigation menu-2



Once the joystick push-button has been pressed again, the MCU exits the navigation submenu and the message shown in *Figure 78* is displayed on the LCD screen.

- If the Jumpers config submenu is selected, the message shown in *Figure 81* is displayed.

Figure 81. Jumpers config menu-1



Press RIGHT and LEFT to go to the next/previous slide and UP to exit. The following messages shown in *Figure 82*, *Figure 83 and Figure 84* are displayed sequentially on the LCD screen.

Figure 82. Jumpers config menu-2

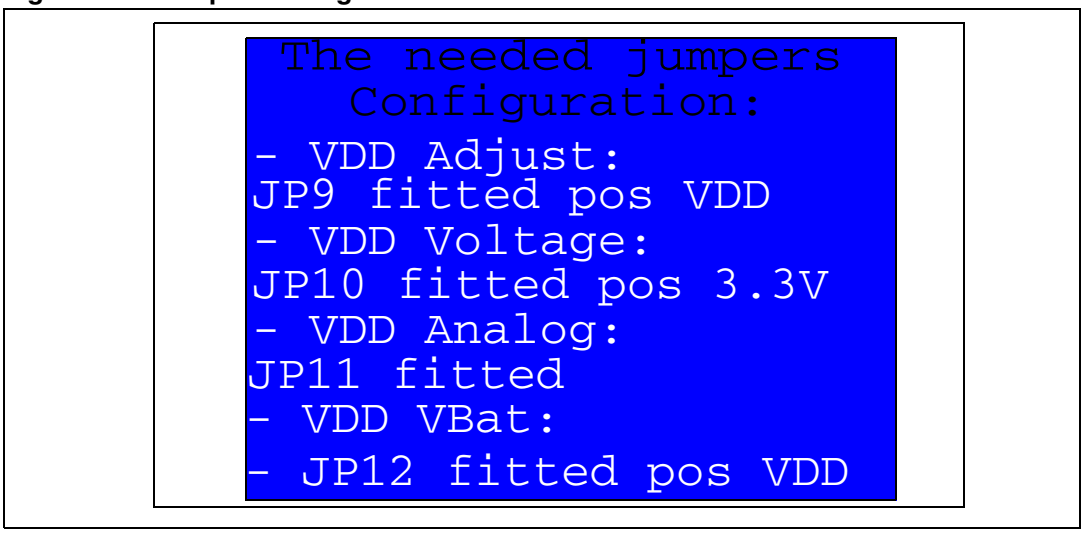


Figure 83. Jumpers config menu-3

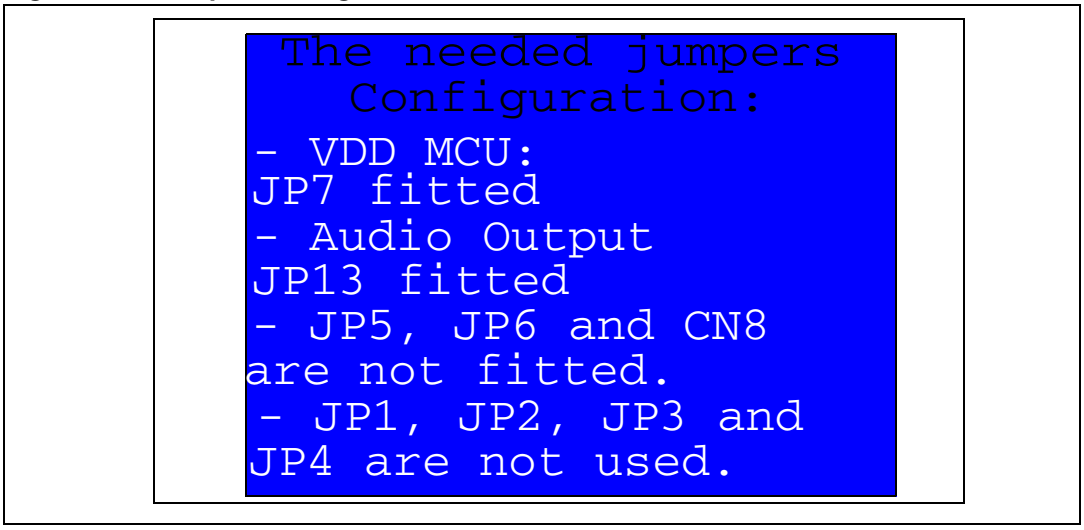
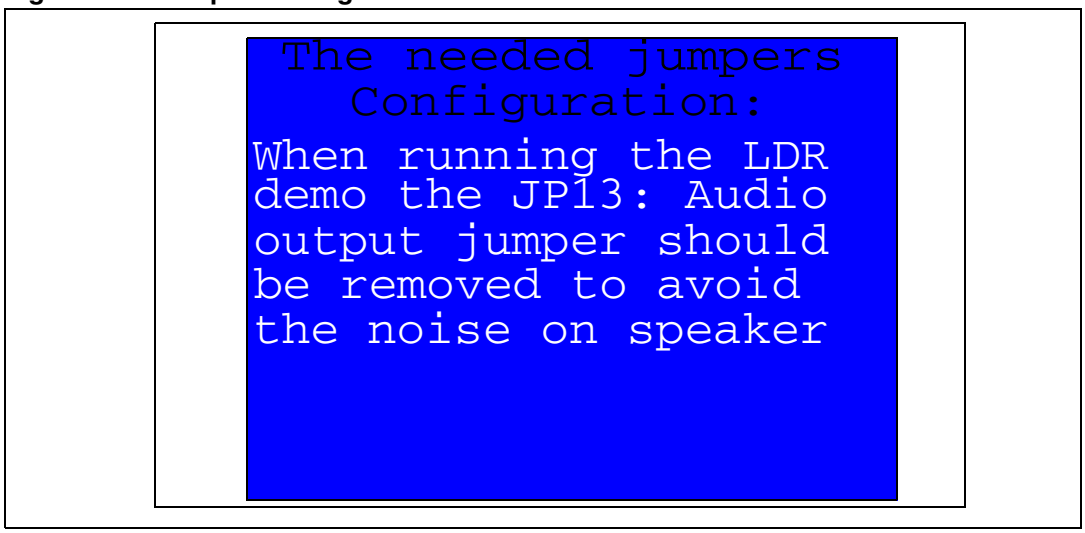


Figure 84. Jumpers config menu-4

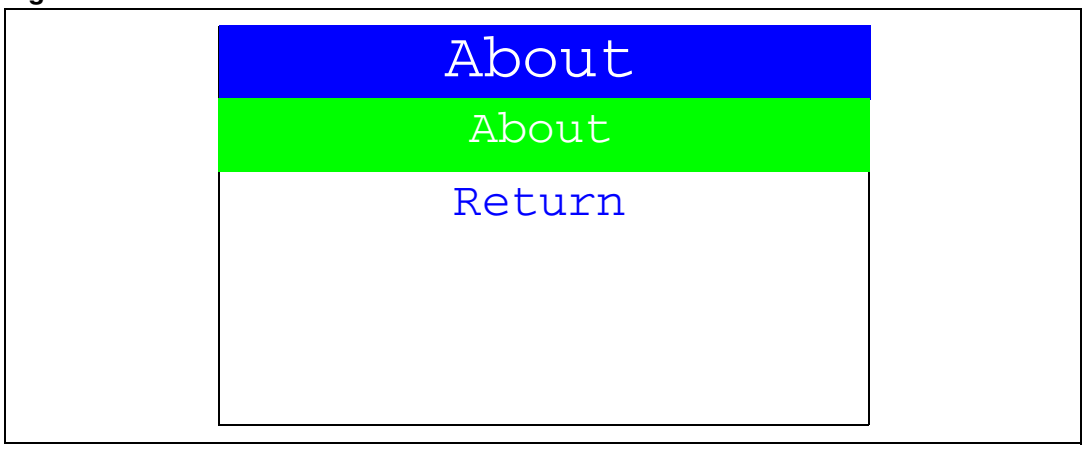


Once the SEL joystick push-button has been pressed, the MCU exits the Jumpers config submenu and the message shown in *Figure 78* is displayed on the LCD screen.

2.4.12 About submenu

This submenu shows the version of the STM32F051R8(T6) demo firmware. When the About submenu is selected, the message shown in *Figure 85* is displayed on the LCD screen.

Figure 85. About submenu

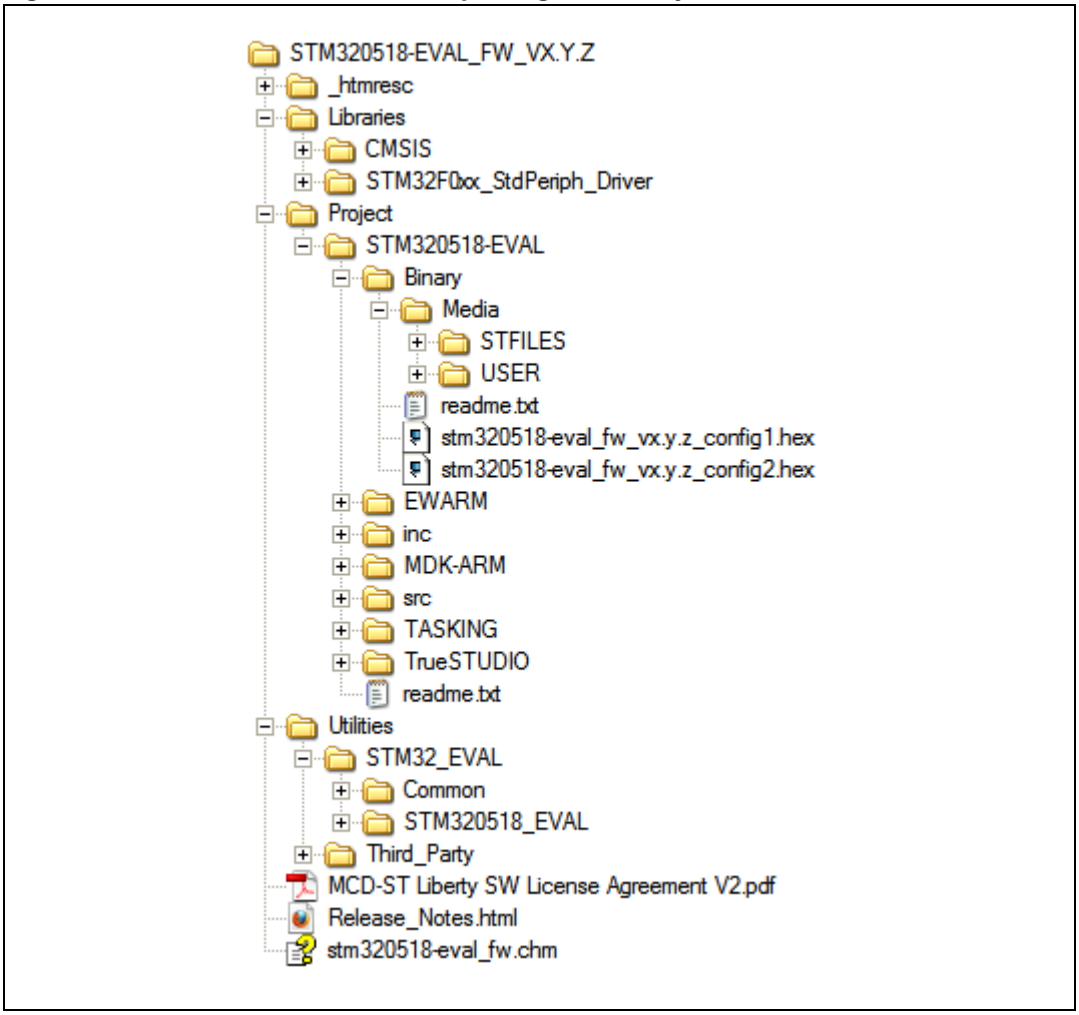


Pressing SEL then displays a message showing the STM32F0518-EVAL demo version on the LCD screen.

3 STM320518-EVAL demonstration package

The STM320518-EVAL demonstration is supplied in one single zip file. The extraction of the zip file generates one folder, STM320518-EVAL_FW_VX.Y.Z, which contains the subfolders shown in *Figure 86* and described below.

Figure 86. STM320518-EVAL demo package directory tree



1. **Libraries**: contains the Hardware Abstraction Layer (HAL) for STM32F0xx devices.
2. **Project\STM320518-EVAL**
   - Binary: contains the binary image (*.hex) of the demonstration that can be used with any in-system programming tool or the embedded Bootloader to reprogram the demonstration. It contains also the media files (*.bmp and *.wav) required to run the demonstration (under \Media subfolder).
   - EWARM: contains preconfigured projects for the EWARM toolchain
   - MDK-ARM: contains preconfigured projects for the MDK-ARM toolchain
   - TrueSTUDIO: contains preconfigured projects for the Atollic toolchain
   - TASKING: contains preconfigured projects for the TASKING toolchain

- inc: contains the demonstration header files
- src: contains the demonstration source files

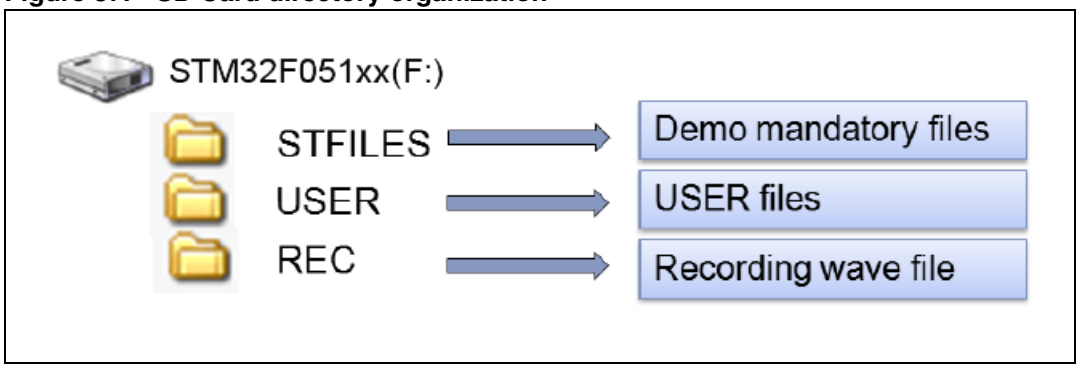
3. **Utilities**
   - STM32F0518-EVAL: contains STM320518-EVAL board drivers

4 STM320518-EVAL demonstration programming

4.1 Programming the media files

The STM320518-EVAL board comes with a MicroSD card memory preprogrammed with Audio and Image resources used by the demonstration. However, you can load your own image (*.bmp) and audio (*.wav) files in the USER directory, respectively, provided that these file formats are supported by the demonstration. For more details, please refer to *Section 2.4.2: Image Viewer submenu* and *Section 2.4.3: Wave Player submenu*.

Figure 87. SD Card directory organization



The default content of the media files (STFILES and USER directories) can be retrieved from the Binary\Media folder. If the user wants to reprogram the MicroSD card, he/she can copy the content of the Binary\Media folder to his/her own SD memory.

4.2 Programming the demo

You can program the demonstration using two methods.

4.2.1 Using Bootloader

To program the demonstration's binary image into the internal Flash memory, you have to use the stm320518-eval_fw_vx.y.z_configx.hex file located under Project\STM320518-EVAL\Binary with embedded Bootloader. For more details, please refer to Bootloader application note AN2606 *STM32™ microcontroller system memory boot mode*.

4.2.2 Using preconfigured projects

- Select the folder corresponding to your preferred toolchain (MDK-ARM, EWARM, RIDE, TASKING or TrueSTUDIO).
- Open the STM32F0518-EVAL_Demo project and rebuild all sources.
- Load the project image through your debugger.
- Restart the evaluation board (press B1: reset button).

5 Revision history

Table 6. Document revision history

Date	Revision	Changes
31-May-2012	1	Initial release.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY TWO AUTHORIZED ST REPRESENTATIVES, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER’S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2012 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com