



EvoPrimer for STM32L Ultra-low-power MCU

**Fun, easy introduction kit
for STM32L microcontrollers**

User Manual

Document version
02 September 2011

Contents

1. Introduction.....	4
1.1 Purpose of this manual.....	4
1.2 Scope of this manual.....	4
1.3 Additional help or information.....	4
2. Presentation.....	5
2.1 Base features.....	5
2.2 Target board features.....	5
2.3 Development software.....	6
3. EvoPrimer hardware.....	7
3.1 Package contents.....	7
3.2 Components overview.....	7
3.3 Target board features.....	8
3.3.1 STM32L152VBT6 microcontroller features.....	8
3.3.2 Target board power supply.....	8
3.3.3 14-segment LCD.....	9
3.3.4 Temperature sensor.....	9
3.3.5 Pressure sensor.....	9
3.3.6 Push button.....	9
3.3.7 Mini-USB connector.....	9
3.4 Base features.....	9
3.4.1 3D MEMS accelerometer.....	9
3.4.2 Power supply.....	9
3.4.3 Extension connector.....	10
3.4.4 Application-specific extension boards.....	11
4. Getting started.....	12
4.1 Switch on.....	12
4.1.1 Insert the target board.....	12
4.1.2 Connect battery, charge and power up.....	13
4.2 Using your EvoPrimer applications.....	13
4.2.1 Play.....	13
4.2.2 Standalone operation.....	14

EvoPrimer for STM32L

4.2.3 Preinstalled Weather application.....	15
4.3 Configuration menu.....	15
4.4 Compiling, programming and debugging.....	16
4.4.1 Install software.....	16
4.4.2 Explore the STM32 Toggle application.....	16
5. Managing your CircleOS applications.....	17
5.1 CircleOS architecture.....	17
5.2 The CircleOS Scheduler.....	17
5.2.1 Initialization stage.....	17
5.2.2 Periodic SysTick interrupt.....	18
5.2.3 Application scheduler.....	18
5.2.4 Periodic Timer2 interrupt.....	18
5.3 CircleOS resource usage.....	19
5.3.1 Memory usage.....	19
5.3.2 Resources used by CircleOS.....	20
5.4 Managing applications on your EvoPrimer.....	21
5.5 Selecting the current application	22
5.6 Downloading new applications.....	22
5.7 Restoring the factory configuration.....	22
5.8 Resetting your EvoPrimer.....	22
6. Developing CircleOS applications.....	23
6.1 Developing your first CircleOS application.....	23
6.2 Libraries.....	24
6.3 Debugging your application.....	24
6.4 Sharing your application with the Circle community.....	24
7. Conformity and recycling.....	25
8. Glossary.....	26
9. Index.....	27
10. History.....	28

1. Introduction

The **EvoPrimer** for STM32L Ultra-low-power line is a fun, cost-effective evaluation and development package based on Raisonance's versatile, innovative **Open4** platform and STM32L152 target board.

Note: The ST product **EvoPrimer** for STM32L Ultra-low-power line (ST order code: STM32L1526PRIMER) is a derivative product of the **Raisonance Open4**. The same Primer target boards, extension boards, software tools and sample applications are used with both the EvoPrimer and Open4.

1.1 Purpose of this manual

This manual provides an overview and installation procedure for your EvoPrimer. Armed with the knowledge in this manual users can quickly understand the target microcontroller's features and create their own applications.

1.2 Scope of this manual

This manual is applicable to all EvoPrimer versions and describes the EvoPrimer's basic use and its hardware and firmware features.

Some technical information about your EvoPrimer is available from the Ride7 interface:

- The 3D MEMS inertial sensor data sheet
- The STM32L 32-bit ultra-low-power MCU documentation
- STM32 MCU reference manual

Visit the Circle community web site at www.STM32circle.com for resources and software tool downloads.

1.3 Additional help or information

Please visit the Raisonance website: <http://www.raisonance.com/> and its support forum or contact Raisonance.

Address: Raisonance S.A.S.
17, Avenue Jean Kuntzmann,
38330 Montbonnot Saint Martin
France

Telephone: +33 4 76 61 02 30

Fax: +33 4 76 41 81 68

Email: support@raisonance.com

If you find any errors or omissions, or if you have suggestions for improving this manual, please let us know by email.

2. Presentation

Your EvoPrimer is composed of two major hardware elements:

- a base platform,
- a target board featuring an STM32L152.

The target board can be used both when connected to the base and in Standalone with the demonstration or a user application.

When connected to the base, the STM32L152 drives the hardware features on the base. This target microcontroller can be reprogrammed and application software can be created and debugged using the Ride7 Raisonance software tools when the base is connected to a PC.

The STM32L152 target board's rechargeable battery and solar cells allow this board to be used in a standalone mode with demonstration and users applications such as the provided temperature sensor application.

The following sections describe the features of the base and target board, software tools and fundamental procedures for their use.



EvoPrimer

2.1 Base features

The base provides the following capabilities:

- USB debug/programming connection (SWD for STM32)
- QVGA (320x240) backlit color LCD, with touch screen capability
- Joystick/push button (mechanical)
- 4 push buttons (based on touchscreen capability)
- Micro SD card connector
- Audio circuit with microphone/loudspeaker/jack
- 3D MEMS accelerometer
- Li-Ion battery with charge management circuitry
- Add-on connector (USART, SPI, I2C, ADC,...)

2.2 Target board features

The STM32L152 target board integrates the following capabilities:

- STM32L152 microcontroller (128KB Flash, 16KB RAM, 4KB EEPROM)
- LCD (14-segment)
- Temperature sensor
- Pressure sensor
- USB device port
- Li-Ion battery
- Solar cells
- Supports standalone operation and can take advantage of its solar cells to perform long-haul tasks in low-power mode.
- When connected to its base, it can take full advantage of the available features.

2.3 Development software

The development software offered for the EvoPrimer for STM32L provides everything you need to program the STM32L and debug applications, including:

- USB host connection for in-circuit programming and debugging.
- Ride7 integrated development environment for code editing, device programming and application debugging (debug up to 64K of code, with included version. For information about upgrade to an unlimited version of Ride, visit <http://www.stm32circle.com/resources>).
- GNU C /C++ compiler (unlimited compiling)
- CircleOS Operating System and base services. An online community available at www.STM32circle.com makes it possible to share your EvoPrimer experience with others on the forums, retrieve useful resources, application notes, firmware, demos (C sources and projects) and applications, all available for free download.

3. EvoPrimer hardware

3.1 Package contents

Your EvoPrimer unit has been carefully packed. Examine the equipment for damage that may have occurred during shipment. If you find any damage, or if any of the items are not included, please contact Raisonance.

Your EvoPrimer is delivered in two packages:

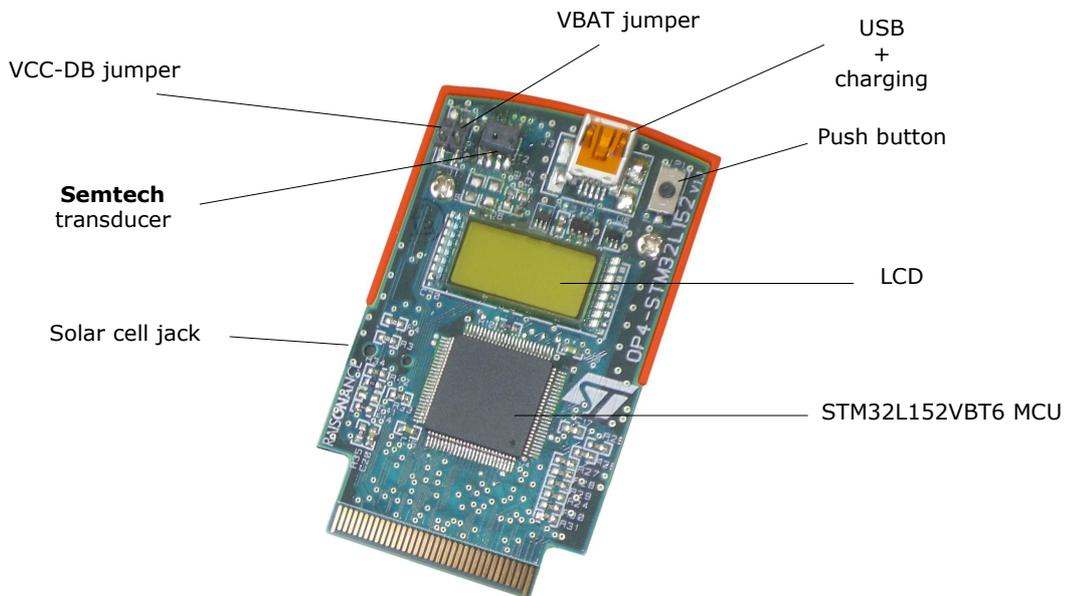
1. The EvoPrimer base with USB cable.
2. The STM32L152 target board with solar cells on a support pedestal.



Note: Download software tools after free registration at www.STM32circle.com

3.2 Components overview

Here is a brief overview of the main components of the STM32L152 target board:



3.3 Target board features

3.3.1 STM32L152VBT6 microcontroller features

The target board is equipped with an STM32L152VBT6, from STMicroelectronics' STM32L ultra-low-power line of 32-bit microcontrollers. Its main characteristics are:

- Advanced STM32 core, Harvard architecture with 3-stage pipeline, 16 CISC MIPS peak
- 128KB of Flash program memory, 16KB RAM, 4KB EEPROM with ECC and RWW
- SWD debug interface
- Low-power RTC
- Integrated LCD controller (8x40 segments)
- 4 DMA channels
- 12-bit DAC with output filter
- 12-bit ADC up to 1Msps, 25 channels
- 10 timers, 2 watchdogs, SysTick
- Up to 83 I/Os, all mappable to interrupts
- Embedded communication peripherals: 3xUSART, 2xSPI, 2xI2C, IrDA, USB
- 96-bit unique ID
- CRC calculation
- 6 low-power modes
- voltage scaling
- 2 comparators, reset system + BOR

3.3.2 Target board power supply

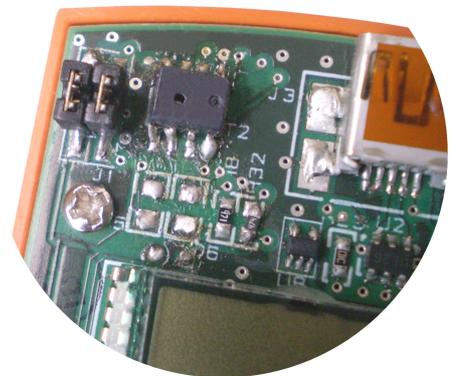
The STM32L152 target board includes its own Li-Ion battery to support use in standalone mode.

This battery can be recharged by the solar cells on the support pedestal, when this support is connected to the jack on the target board. This makes it possible to use the target board to perform long haul tasks in low-power mode.

The target board battery may also be recharged by the battery or USB connection of the base or by the USB connection of the target board.

Power supply jumpers on the target board connect/disconnect the target board's battery (VBAT jumper) and power on/off the STM32L152 (VCC_DB).

Remove these jumpers when the target board is not in use to avoid fully discharging the target board's battery.



Note: It is recommended to charge the battery for 30 hours before using it.

This operation must be performed with the target board inserted into the base and connected to the USB port to a PC (ensure the PC does not go into sleep mode during this period).



Warning: If the target battery is ever fully discharged, it is **impossible to recharge** it. To avoid this, remove VCC_DB jumper whenever the target board is not used and not connected to the base or solar cells.

3.3.3 14-segment LCD

The STM32L152 target board includes its own 14-segment LCD display which is driven by the STM32L152 and may be used in demonstration and user applications. In the temperature sensor demonstration application, it is used for temperature display when mounted on the base or when used in standalone mode.

3.3.4 Temperature sensor

The STM32L152 target board includes its own temperature sensor (STMicroelectronics' STLM20W87F) which can be used by applications when on the base or in standalone mode.

3.3.5 Pressure sensor

The STM32L152 target board includes a US9111 sensor and an Semtech SX8724 transducer which can be used by applications when on the base or in standalone mode.

3.3.6 Push button

The STM32L152 target board includes its own push button which can be used by applications when on the base or in standalone mode.

3.3.7 Mini-USB connector

The STM32L target board includes a mini-USB connector which can be used for USB 2.0 implementation in demonstration and user applications.

The 5V provided by the host computer is also used to recharge the local battery.

3.4 Base features

3.4.1 3D MEMS accelerometer

The base is equipped with a MEMS inertial sensor (LIS3LV02DL from STMicroelectronics) which can be used to select commands in coordination with a graphic pointer.

When you start the EvoPrimer for the first time, you will see a small ball moving according to the orientation of the EvoPrimer. This 3D positioning information is provided by the MEMS.

3.4.2 Power supply

The base features a 400mAh Li-Ion rechargeable battery, equipped with a voltage regulator and a battery charger.

When the battery is fully charged, the EvoPrimer can be used for about 6 hours. The duration of the batteries depends on the activity.

When the USB connector is linked to a PC, the host voltage (supplied by the PC) recharges the battery.

When no USB host is connected, the battery supplies power for the EvoPrimer.

3.4.3 Extension connector

The base has a 20-pin HE14 female right-angle extension connector that you can access when removing the orange part of the case. The extension board mechanical description is available in the www.STM32circle.com resources section. The extension connector permits usage of the following pins:

Pin	Printed name	STM32L pin	STM32L name	Description
1	V2V8	Vcc	VCC3V1	Connected to U6 regulator output (3.1V). Can provide approximately 100mA to the extension board.
2	GND	GND	GND	Ground
3	SCL	PB.6	CX_I2CSCL	I2C clock
4	SDA	PB.7	CX_I2CSDA	I2C data
5	MISO	PB.14	SDCARD_MISO	SDCARD MISO
6	SD	PB.15	SDCARD_MOSI	SDCARD MOSI
7	SCK	PB.13	SDCARD_SCK	SDCARD Clock
8	WS	PB.8	CX_TIM1	Timer
9	CANH	PD.0	CX_TIM2	Timer
10	CANL	PA.4	CX_ADC_DAC	ADC
11	ADC1	PA.2	CX_ADC1	ADC
12	ADC2	PA.3	CX_ADC2	ADC
13	A_TIM	PA.1	CX_ADC_TIM	ADC
14	CTS	PD.3	CX_USART_CTS	USART
15	RTS	PD.4	CX_USART_RTS	USART
16	TX	PD.5	CX_USART_TX	USART Tx
17	CK		P_BUTTON	Push button input. Can wakeup the target board.
18	RX	PD.6	CX_USART_RX	USART Rx
19	VEXT		VBAT	Battery voltage input for the extension board
20	GND	VSS	GND	Ground

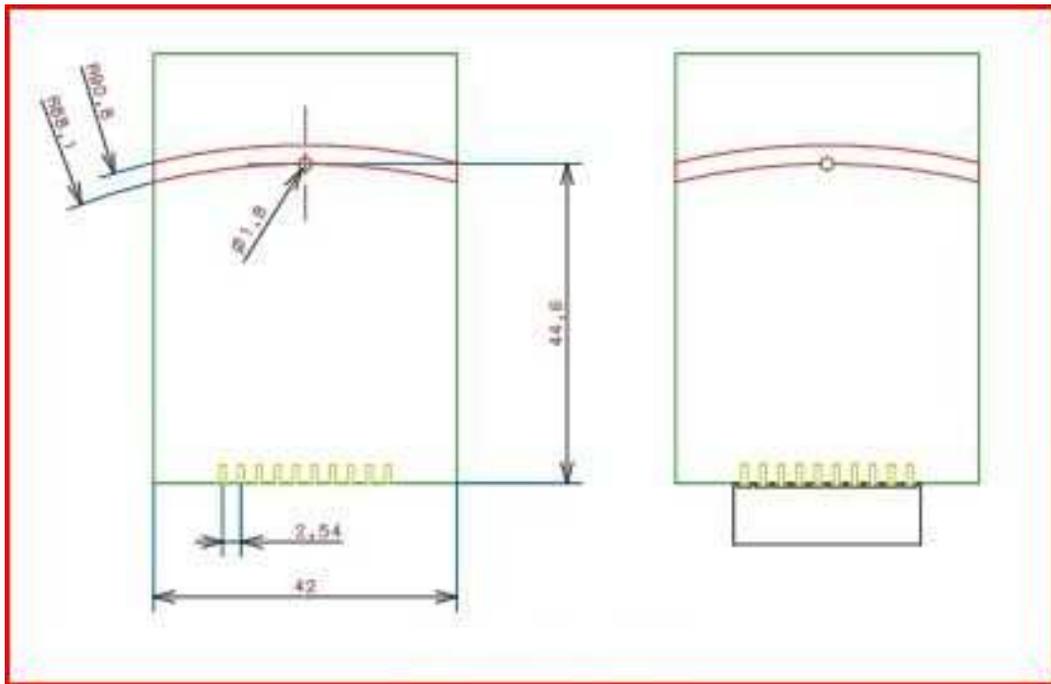
Note: Printed names on the extension boards may not accurately match the STM32L target board names as the generic base supports other processor families (namely STM8) with other features.

3.4.4 Application-specific extension boards

The base has been designed to allow extension through application-specific extension boards.

Such extension boards connect to the extension connector and are compatible (mechanically and electronically) with the previous STM32-Primer2, so you should be able to reuse your previous designs without modifying the extension board. However, some limitations may apply, for instance if you use specific processor/peripheral capabilities that are not portable between designs.

You can design your own extension board if your application requires it. The following picture shows the mechanical constraints that you should respect so that your extension board fits inside casing of the base platform.



Extension board mechanical specifications

4. Getting started

This chapter explains how to switch on your EvoPrimer and describes some of the applications that are provided with it.

4.1 Switch on

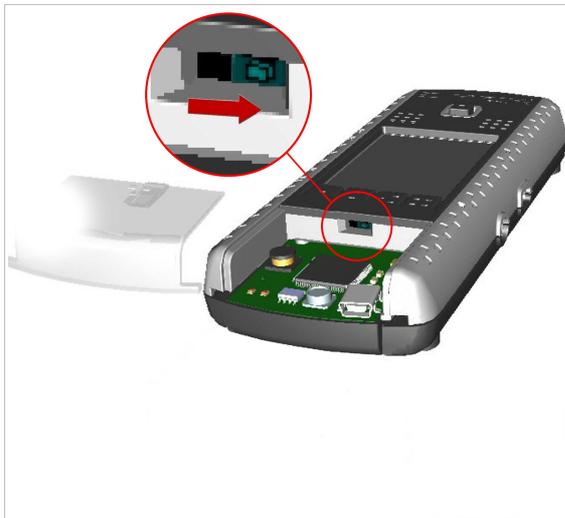
The procedure for getting started is reduced to four simple operations:

1. Insert a target board into the EvoPrimer base (see below).
2. Switch the power switch to provide current to the target board (see below).
3. Slide the transparent cover in place.
4. Press the joystick to power up the EvoPrimer.

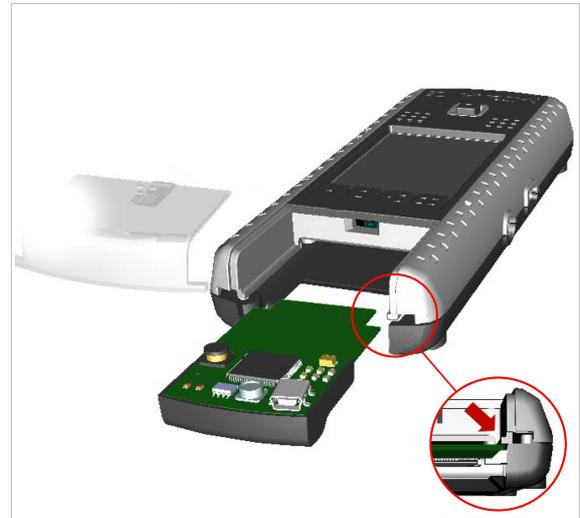
4.1.1 Insert the target board

Follow these steps to insert the target board:

1. Remove the transparent cover.
2. Switch off the battery (see image below left).
3. Insert the target board into the base, as shown in the image, ensuring that the PCB is against the guides.
4. Push until the top of the target board fits with the side of the base.



Switching OFF the Primer



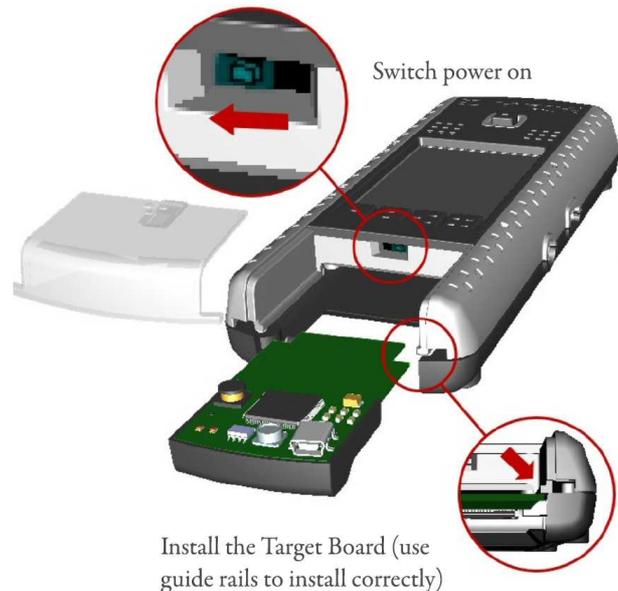
Inserting the target board

4.1.2 Connect battery, charge and power up

Your EvoPrimer is shipped with the battery disconnected, to prevent discharging and potential harm to the battery.

You must connect the battery before operating your EvoPrimer:

1. Open the EvoPrimer's case by removing the cover which is on the target board.
2. Switch the button as shown on the image to power-on your device.
3. Close the case.
4. Connect the EvoPrimer to the USB port on a PC to charge its battery. Only the red LED (L0) is lit if the battery is charging. The green LED (L1) is lit when fully charged.
5. After charging the battery, press the push button to activate the EvoPrimer.



4.2 Using your EvoPrimer applications

The base provides user interface features such as a color LCD, touchscreen, audio circuit, joystick/push button, GUI, SD card connector, MEMs accelerometer, IrDA and much more.

The target board can be:

- connected to the base to operate evaluation features (and includes additional features such as audio codec and a USB connector),
- a standalone temperature sensor application.

4.2.1 Play

After the opening screen is displayed, press the push button to call the main menu.

Tilt the EvoPrimer backward/forward or use the joystick to navigate in the menu, use the push button to select a menu command. Initial menu items include:

Config	Configure parameters for the EvoPrimer including the pointer, backlight for the LCD display and time
Breakout	This is an addictive game provided as an example. Full source is available from www.STM32circle.com .
Applic	Select an application to run. The application that you choose will then appear in the main menu.
SD card	Display information saved on the SD card, or allows to configure your EvoPrimer as a mass storage device, and transfer files from/to your PC from/to the SD card.
About	Display information about the CircleOS version and available memory.
Shutdown	Turn off the EvoPrimer. To restart the EvoPrimer, simply press the push button.
Quit	Quit the menu

Note: When you receive the EvoPrimer, the MEMS based controls are calibrated to a "zero" position that matches a 30° angle from the horizontal (corresponds to the position of a book when reading). To practice controlling the EvoPrimer, move the blue dot around the main screen. It takes a little practice.

4.2.2 Standalone operation

Some applications are designed to run on the target board in standalone mode. In this style of operation the target board is powered by its own battery with recharging provided by the solar cells.

The "Weather" application is one such example, once it has been loaded, it can be run on the target board in standalone mode.

These steps describe the procedure to use it in standalone mode:

1. Power-off the base.
2. Remove the transparent cover that is over the target board.
3. Turn the base so the display is facing downwards.
4. Hold the base with both hands and push upward on the target board with both thumbs pressing evenly on the target boards plastic drawer.
5. When the target board dislodges from the connector, remove it completely by pulling on the plastic drawer.
6. Remove then put back the VCC_DB jumper to start the application.
7. Connect the solar cells as shown here to recharge the target board's battery.
8. Make sure that a light source is available to recharge the target board battery.

Follow steps 1-5 above whenever you remove any target board from the base.



Note: The LCD screen contrast varies depending on the power supply, so in standalone mode the screen contrast may be low if the battery charge is low.

The solar cells are designed to work with natural light. It is ideal to place the target board near a window during daylight for charging.



Important: If the target board battery is ever fully discharged **it is impossible to recharge it.** To avoid this, **remove the VCC_DB jumper** whenever the target board is not connected to the base or the solar cells.

4.2.3 Preinstalled Weather application

The EvoPrimer includes a preinstalled Weather application that records temperature and pressure in standalone mode (with the target board disconnected from the base), taking advantage of the very low power mode of the STM32L152. It uses the target board's temperature sensor to display the current temperature, and to record the temperature on a regular basis (selectable from 1 second up to 1 hour). The measurements are logged into EEPROM.

Configure the application as follows:

1. Insert the target board into the base. Ensure the PCB is against the guides.
2. Set VCC_DB jumper.
3. Switch on the power switch and slide the transparent cover into place.
4. Press the joystick to power up the base. If the EvoPrimer does not start, remove and then put back the VCC_DB jumper to restart the STM32L.
5. Press the joystick to access the main menu.
6. Select **Weather** from the **Applic** menu to launch the application. The mini LCD displays alternatively the current temperature and atmospheric pressure.
7. To configure measurement interval, press the joystick and click on **Select Period**. The STM32L monitors the temperature at the selected interval and refreshes the mini LCD. The default value is 10s.
8. Two modes are available :
 - Real time : the Primer plots periodically the temperature and pressure values,
 - Log : the Primer displays previously saved values (in realtime mode or standalone mode).

Use it as follows to run the application in standalone mode:

1. Power off the EvoPrimer base, remove the VCC_DB jumper.
2. Remove the Target Board (push upward with both thumbs on the plastic drawer).
3. Set VCC_DB and VBAT jumpers.
4. Switch on the power switch.
5. Connect eventually the solar cells, or the USB cable to recharge the Target Board's battery.

4.3 Configuration menu

Configuration and test applications are preinstalled which check your EvoPrimer's capabilities.

From the main menu of the EvoPrimer, run the Config command to access the following parameters:

1. **Power:**
 - CPU Freq:** Chooses the speed of your EvoPrimer.
 - Backlight:** Tunes backlight intensity. Note that the backlight is the main source of power consumption (with the LCD monitor itself). Reducing backlight intensity extends the duration when the EvoPrimer is powered by the battery.
2. **Interface:**
 - User input:** Selects the user input for commands menu (accelerometer or joystick or accelerometer + joystick or touchscreen)
 - Menu:** Selects whether or not to display the menu with a large font (*Not available*).
 - Beep:** Selects whether or not to send a "beep" when the EvoPrimer is tapped twice.
 - TS Calibrate:** Launches a calibration procedure of the touchscreen.
3. **Autorun:** Lets the current application run whenever your EvoPrimer is started. In this mode, there is no need to select the application in order to start it.
4. **Time:** Sets your EvoPrimer time. Note: RTC clock remains valid even if EvoPrimer is Off.
5. **Test:** Performs a quick factory test of the EvoPrimer.

You can add more applications to your EvoPrimer by downloading them on the web site www.STM32circle.com/projects. Refer to Chapter 5 Managing your CircleOS applications.

4.4 Compiling, programming and debugging

4.4.1 Install software

To explore, modify and create new applications on your EvoPrimer you need to install some software. Complete tool documentation is provided with the software installation.

1. Register and download the CD-ROM-Image file on the Resource page of www.stm32circle.com.
2. Extract the files and install:
 - a) **Ride7**, the Raisonance IDE to write and debug new applications. This includes the RLink driver which is necessary to pilot the programming and debugging of your EvoPrimer from your PC.
 - b) **RKit-ARM**, the Ride7 add-on that contains:
 - the complete GNU software toolchain based on the GCC compiler
 - the Raisonance optimizing C compiler toolchain
 - some utilities to manage your Circle applications (Circle is the OS embedded on your EvoPrimer; refer to Chapter5.1 "CircleOS architecture" for details).
3. Launch Ride7.
4. Connect your PC to the Debug USB port on the base.

Note 1: Ride7 and RKit-ARM should be installed **before** connecting to this USB port.

Note 2 : 64 KB output. Licence available for free at "www.mcu-raisonance.com/STM32_registration.html"

RKit-ARM includes many datasheets and manuals, accessible from Ride7 "Documentation" window.

4.4.2 Explore the STM32 Toggle application

A Toggle example is installed with the RKit-ARM for Ride7 in the following directory:

`"[RIDE7_INSTALL_DIR]\Examples\ARM\Primer\EvoPrimer\toggle_STM32L\toggle.rprj"`

Follow these steps to execute the example on your EvoPrimer:

1. Open Ride7.
2. Select **Open | Project** and navigate to the example.
3. Click on **Project | Make Project**.
4. Once the project is built, connect your EvoPrimer to your PC through USB.
5. Click **Debug | Start**.
6. Once the debugger is ready, you can run the example through **Debug | Run**.

You can then play with the sample application on your EvoPrimer.

5. Managing your CircleOS applications

5.1 CircleOS architecture

The STM32L152 target board is equipped with an STM32L152VBT6 that contains 128KB of Flash ROM and 16KB of RAM. The EvoPrimer embeds the CircleOS operating system (source files are available from <http://www.stm32circle.com/> after free registration). It provides services that help you develop your STM32L EvoPrimer applications, including:

1. Application management,
2. LCD graphic functions,
3. MEMS functions,
4. LED, Buzzer and Push Button functions,
5. Menu functions,
6. Scheduler task,
7. Audio functions,
8. File system functions,
9. ...

CircleOS can load several independent applications.

Each application is run by CircleOS when selected, has the full availability of the CPU and can use all the RAM that is not being used by CircleOS (i.e. 16KB in the memory address range from 2000000h to 20003FFFh). It is scheduled by the CircleOS with full privileges on the device, until it explicitly quits.

5.2 The CircleOS Scheduler

CircleOS acts in several stages: An initialization stage which occurs upon device reset, a periodic SysTick interrupt, and the scheduling of applications.

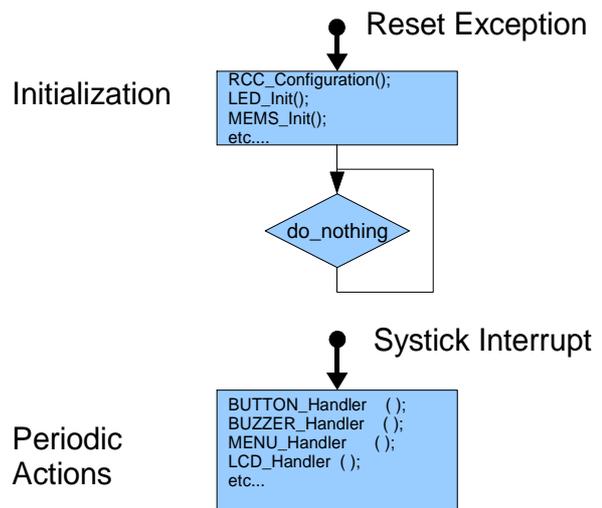
5.2.1 Initialization stage

During the initialization stage, the hardware configuration is performed, and the periodic system timer (SysTick) is installed.

The SysTick period depends on the RCC settings: it can be modified through the menu **Config | CPU Speed** to the following predefined values:

Level	CPU freq (MHz)	Systick freq (kHz)
1	9	0.75
2	12	1
3	18	1.5
4	24	2
5	32	3

The ratio $\text{CPU_freq} / \text{SysTick} = 24000$ applies for all these values.



5.2.2 Periodic SysTick interrupt

The periodic SysTick makes a call to the CircleOS SysTick interrupt handler which performs a short process on each of the STM32-EvoPrimer components: LEDs, button, buzzer, LCD and so on.

Note that the MEMS handler is called from the Timer2 interrupt, which has a higher priority than the SysTick interrupt. This ensures a fixed and precise measurement rate.

5.2.3 Application scheduler

CircleOS is the base application of the EvoPrimer. It handles the menu selections and reacts to user actions. When an application is run (usually through a menu selection), CircleOS calls an initialization routine for the application, CircleOS then repeatedly calls the application handler at the SysTick frequency until it returns a `MENU_LEAVE` value.

You can find more details about programming CircleOS applications in Chapter 6 “Developing CircleOS applications”.

5.2.4 Periodic Timer2 interrupt

The SPIs used to control the MEMS are both driven from the Timer2 interrupt handler which is triggered at a fixed rate:

Level	CPU freq (MHz)	Systick freq (kHz)	Timer2 freq (Hz)
1	9	0.75	136.5
2	12	1	182
3	18	1.5	273
4	24	2	364
5	32	3	546

Timer2 priority is **higher** than the SysTick priority. It is mandatory to keep this IRQ at the highest level in order to guarantee proper MEMS behavior.

Note: The Timer2 overflow frequency has been set to read a maximum number of measurements from the MEMS. However, this high rate is often useless, and you could easily change this frequency to save CPU time, or to get some more stable data from the MEMS.

5.3 CircleOS resource usage

CircleOS offers lots of services and functions. When a CircleOS application is built, the linker removes unnecessary parts from the produced executable.

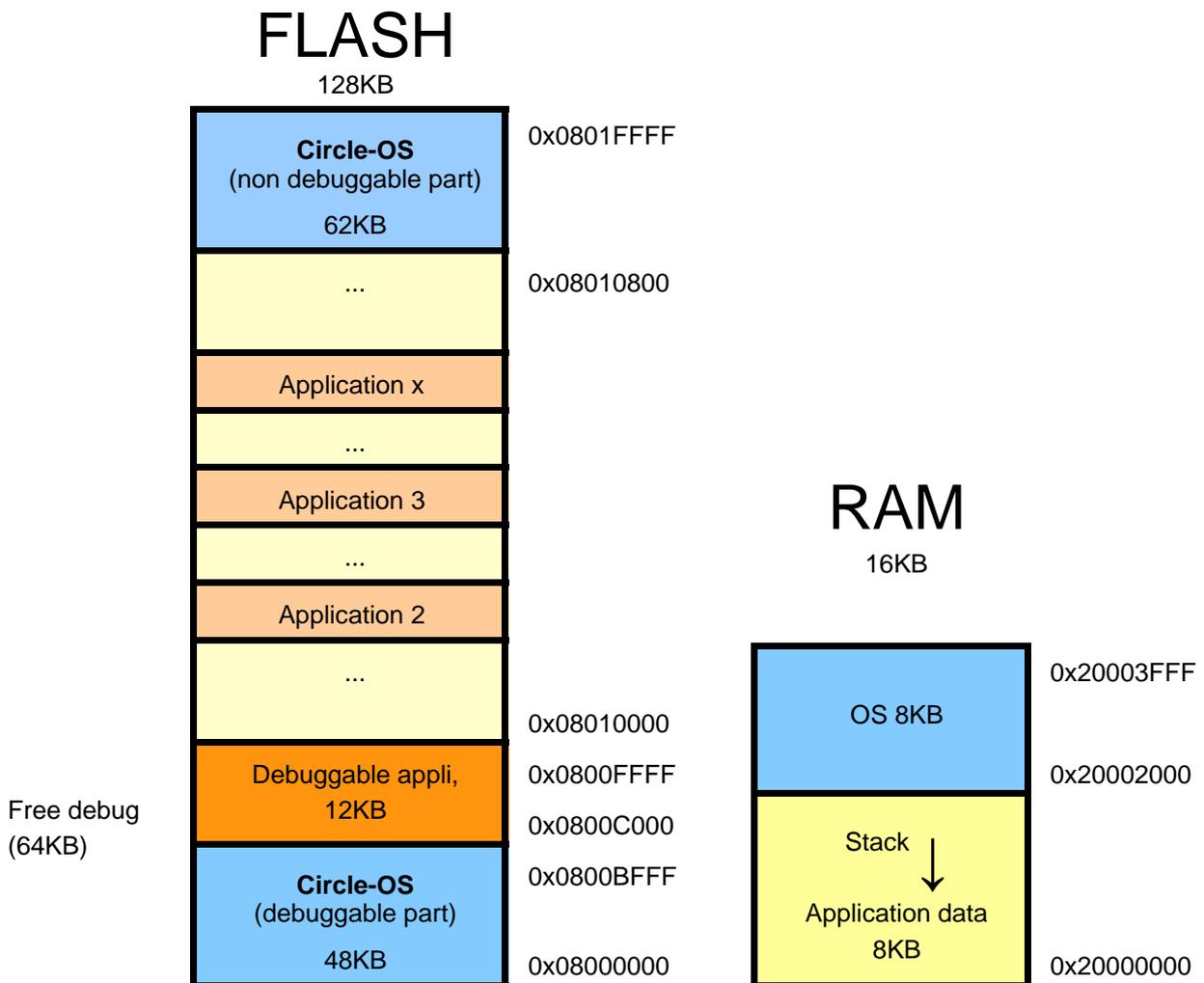
5.3.1 Memory usage

The CircleOS firmware requires 48KB of debug code in Flash, 62KB of constants and non-debuggable code in Flash and 8KB of RAM (including the stack usage for the applications).

The remaining 18KB of Flash (128 – 110) are available for applications, which can be added or removed at will using a programming tool (see below).

Flash memory can be programmed in 2KB blocks only.

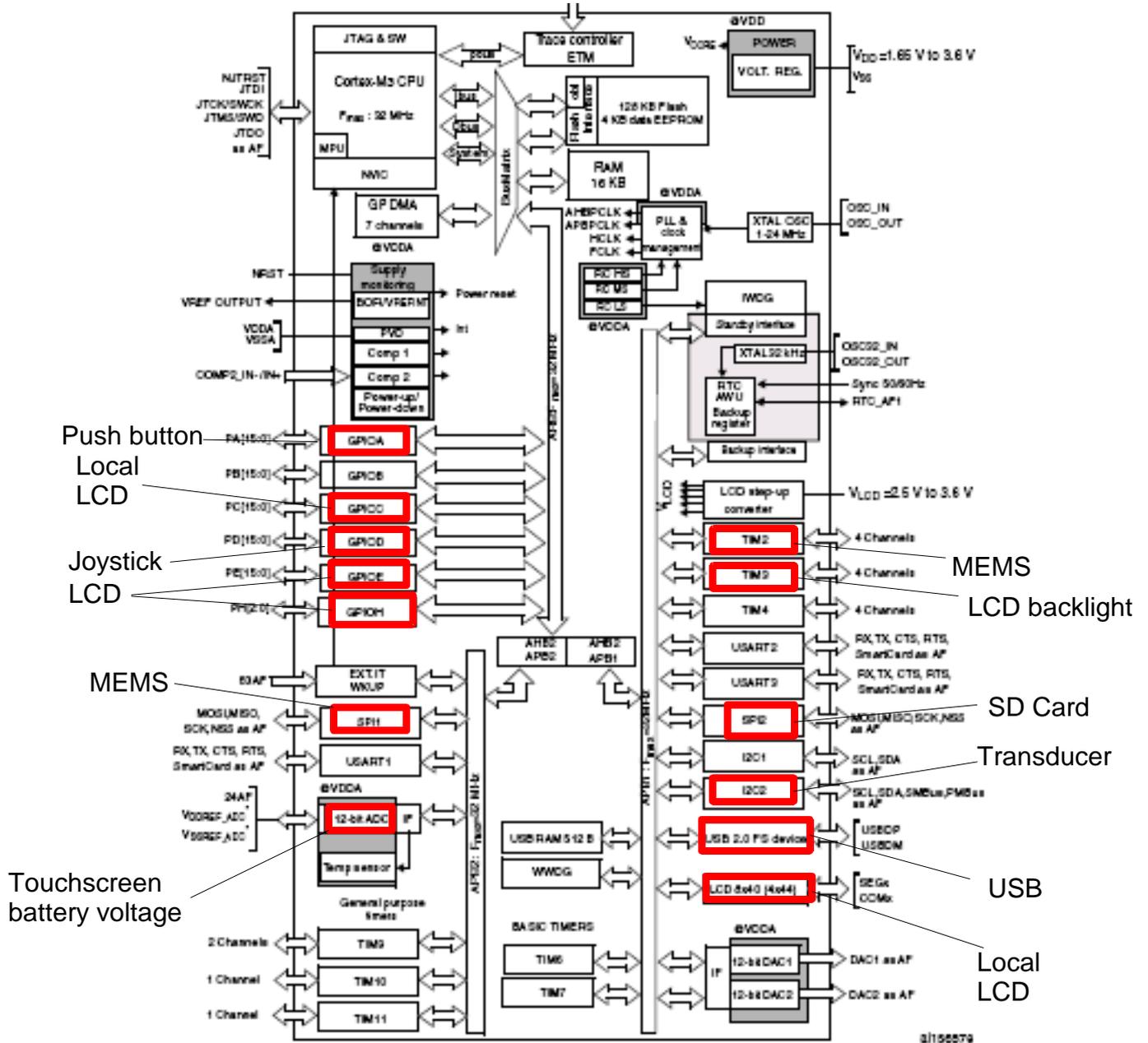
The following shows an example of memory mapping:



5.3.2 Resources used by CircleOS

The following figure (from the ST datasheet) shows the peripherals used by CircleOS:

Illustration 5.1: STM32L peripherals



5.4 Managing applications on your EvoPrimer

The applications can be managed by the **circle_mgr.exe** utility, which is in the directory [RIDE7_INSTALL_DIR]\Bin. This utility can:

- List the currently loaded CircleOS applications.
- Add new CircleOS applications.
- Remove CircleOS applications.
- Check how much FLASH memory is available.

Refer to the <http://www.stm32circle.com/> web site for more information about this utility.

The following commands are available with the **circle_mgr.exe** utility:

Command	Syntax	Description
List	L	List the loaded applications. The following information will be output: circle_mgr.exe L Reading FAT table... App0: Name=Maze, Addr=0x08006000, Size=8KB App1: Name=Breakout, Addr=0x08008000, Size=4KB Largest free block= 92KB
Add	Afilename	Add a new application (object file). circle_mgr.exe Ac:\tmp\level.o Linking file C:\tmp\level.o... Link of C:\tmp\level.o succeeded... Hex file generated... Blank-checking the FLASH area...OK Programming file _tmp_.ld.hex to flash...OK Registering application in FAT... OK
Erase	E* Eappname	circle_mgr.exe EMaze /*remove only 'Maze'*/ circle_mgr.exe E* /*remove ALL apps */
Wait	W	When a command list is launched through a batch file, the W command allows you to pause the execution and to check the intermediate results.
Start	S	Start CPU execution



Caution: Any hex file can be programmed to the Flash memory of your EvoPrimer using the **cortex_pgm.exe** utility. However, doing this will destroy your CircleOS firmware, and you will have to reinstall it if you wish to use it later (refer to chapter 5.7 for details).

5.5 Selecting the current application

One application is considered to be the “current application”. The ID of the current application is saved in the backup memory. From the main menu, you can launch it directly.

To change the current application, select the “Application” command from the main menu. Then select the application you wish to specify it as the “current application” and push the button. The new “current application” name will now appear in the main menu.

5.6 Downloading new applications

On the <http://www.stm32circle.com/projects> Circle web site you will find a database where the members can share their applications with the stm32circle community.

An application can include both the source files and the object files, or just the object files.

An application is generally made of one object file, but may occasionally have several of them. The linking of the application in such a case can be done either using the **circle_mgr.exe** software (available in the “[RIDE7_INSTALL_DIR]\bin” directory) or within the Ride7 environment. When an application is split into several object files, these object files must be placed in a library in order to pass a unique filename as an argument to **circle_mgr.exe**.

5.7 Restoring the factory configuration

If you have been experimenting with CircleOS applications and have modified your EvoPrimer configuration, you may want to restore the initial (factory) configuration.

In order to do this, follow these steps:

1. Connect your EvoPrimer’s debug USB port to your PC.
2. Power-up your EvoPrimer by pressing its button.
3. Open a command prompt from Windows (Navigate to **Start | Programs | Accessories | Command prompt**)
4. Change the current directory to the Ride7 installation directory. This can be done with the following command (adapt it to your actual configuration):

```
cd "C:\Program Files\Raisonance\Ride"
```

5. Now change the current directory to the “Bin” directory. This can be done with the following command:

```
cd Bin
```

6. Erase your EvoPrimer, reprogram it with its factory ROM image and restart the device. These operations can be done with the following single command, which takes about 30 seconds:

```
Restore_EvoPrimer_STM32L_Circle_Factory.bat
```

5.8 Resetting your EvoPrimer

A hardware reset will restart your EvoPrimer (restoring it to a clean state), but will not remove its ROM contents.

- Remove any USB cable.
- Remove the plastic case protecting the STM32L152 target board.
- Remove the VCC-DB jumper, then replace it.
- Put the plastic case back in place.
- Restart your EvoPrimer by pressing its button.

6. Developing CircleOS applications

The full source files of CircleOS are available on the <http://www.STM32circle.com/> web site. Once registered, you can download them, along with many resources for developing your application.

6.1 Developing your first CircleOS application

Creation of a CircleOS application is done automatically in Ride7:

1. Navigate to **Project | New project**.
2. Set the **Type** selection list to **New application**.
3. In the **Processor** selection list, select the **STM32L_Evo_CircleOS** device from the ARM-STM32 family, sub-family **Primer**.
4. Select an application name such as **My CircleOS application**.
5. Define the location where your new project will be created.
6. Click the **Finish** button.
Your new project will be created, with an application containing a CircleOS application skeleton as well as the **Evo_Circle_STM32L.elf** and **FAT_OP4.elf** files necessary to connect your application to CircleOS.
7. Open the *application.c* file.
8. Search for the `Application_Name` variable in the file.
9. Change the `Application_Name` value from **My App** to **HELLO**.
10. In the `Application_Handler` function, create a new string as follows:

```
const char msg[] = "Hello, World!";
```
11. Use the `DRAW_DisplayString` CircleOS service to display the `msg` string variable you just created:

```
DRAW_DisplayString(5,20,msg,sizeof(msg)); // X, Y, string, length
```
12. Build your project using the **Project | Make Project** command.
13. Connect your EvoPrimer to your PC using the USB cable (be sure to use the “debug” USB port of your EvoPrimer).
14. From Ride7, go to **Debug | Start**, which will program your application to your EvoPrimer. This may take about 10 seconds.
15. Go to **Debug | Run**.
16. On your EvoPrimer, select your application name on the main menu.

Your application is now on your EvoPrimer.

For further information about CircleOS application programming and available OS services, please visit <http://www.STM32circle.com/>.

6.2 Libraries

Some common services are offered to ease your development of CircleOS applications.

1. The STM32L firmware libraries, written by ST, that provide access to the embedded peripherals (such as timers, ADC, communication interfaces, GPIO, etc...) of the STM32L microcontroller.
2. The low-level CircleOS functions that provide an easy access to the EvoPrimer's on-board peripherals: 3D accelerometer, LCD monitor, button, buzzer, battery, LEDs.
3. The graphical functions that provide powerful high-level functionality: Menu management, pointers (linked to the 3D accelerometer), character maps, sound.

The source files of these libraries can be found:

- ST web site for the STM32L libraries (includes other documentation about the STM32 library)
- On <http://www.STM32circle.com/> web site for the CircleOS libraries (registration required).

Documentation about these libraries is accessible from Ride7.

6.3 Debugging your application

In order to debug your application:

1. Go to **Project | Properties** in Ride7.
2. In the **Configuration** selection box, select the **Circle_Debug** configuration (which is the default).

Ride7 takes care of all the settings required to switch between debug and release mode through the use of these configurations.

Note: The standard STM32L EvoPrimer is limited to debugging in the first 64KB only. A software key can be purchased on <http://www.stm32circle.com/> to allow debugging in the whole 128KB of memory.

6.4 Sharing your application with the Circle community

Once your application works properly, you can share it with the other members through the <http://www.STM32circle.com/> community.

7. Conformity and recycling

For more information on conformity and recycling, please visit the Raisonance website www.raisonance.com



ROHS Compliance (Restriction of Hazardous Substances)

Raisonance products are certified to comply with the European Union RoHS Directive (2002/95/EC) which restricts the use of six hazardous chemicals in its products for the protection of human health and the environment.

The restricted substances are as follows: lead, mercury, cadmium, hexavalent chromium, polybrominated biphenyls (PBB), and polybrominated diphenyl ethers (PBDE).



CE Compliance (Conformité Européenne)

Raisonance products are certified to comply with the European Union CE Directive.

In a domestic environment, the user is responsible for taking protective measures from possible radio interference the products may cause.



FCC Compliance (Federal Communications Commission)

Raisonance products are certified as Class A products in compliance with the American FCC requirements. In a domestic environment, the user is responsible for taking protective measures from possible radio interference the products may cause.



WEEE Compliance (The Waste Electrical & Electronic Equipment Directive)

As part of our continuing efforts to provide the best service to our users and communities, Raisonance is actively defining take-back and recycling programs for EU customers as part of the implementation of the European WEEE directive. The WEEE symbol is a reminder not to dispose of your electronic equipment in standard trash receptacles.

For more information about disposal and recycling of electronic goods, please refer to <http://www.raisonance.com/support/weee.php>.

Raisonance disposes of its electrical equipment according to the WEEE Directive (2002/96/EC).

Upon request, Raisonance can recycle customer's redundant products.

8. Glossary

Term	Description
STM32L152 Primer	EvoPrimer for STM32L152 MCU, a microcontroller evaluation & development platform
EvoPrimer	Commercial products which are derived from the Raisonance Open4 and distributed by STMicroelectronics for exploring, evaluating, developing and fine tuning applications for a variety of ST 8- and 32-bit microcontrollers. Certain use limitations may apply to these commercial products
Open4	Raisonance versatile hardware platform, which is used in a variety of commercial products that allow users to explore, evaluate, develop and fine tune applications for a range of microcontrollers that are mounted on Open4 compatible target boards. Also referred to as the "Base platform." Also distributed under the name EvoPrimer base
Target board	Boards featuring target microcontrollers which, when installed on an Open4-derived base allow execution of applications, programming of the target microcontroller and debugging of the applications running on the target microcontroller. Hardware features of target boards will vary depending on the target MCU and its features. Some features of the base platform may not be available depending on the features and capabilities of the target microcontroller
Extension board	Application-specific extension board, connected to the extension connector.
CircleOS	Embedded OS running on the Primer development kits.

9. Index

Alphabetical Index

ADC.....	8	LED (L1).....	13
Add.....	21	Li-Ion.....	8
application-specific extension boards.....	11	LIS3LV02DL.....	9
Autorun.....	15	List.....	21
Beep.....	15	Menu.....	15
CE.....	25	Open4.....	4
CircleOS architecture.....	17	Power supply.....	9
CircleOS peripherals.....	20	programming.....	16
CircleOS SysTick.....	18	push button.....	9
CISC MIPS.....	8	Raisonance	25
communication peripherals.....	8	RAM.....	8
Compiling.....	16	reset.....	22
Compliance.....	25	ROHS.....	25
Conformity.....	25	RTC.....	8
cortex_pgm.exe.....	21	Standalone operation.....	14
CPU Speed.....	17	Start.....	21
current application.....	22	STLM20W87F.....	9
DAC.....	8	STM32L152VBT6.....	8
debugging.....	16	SWD.....	8
Directive.....	25	SysTick.....	17
DMA.....	8	temperature sensor.....	9
Erase.....	21	Test.....	15
EvoPrimer.....	4	Time.....	15
Extension connector.....	10	Timer2.....	18
factory configuration.....	22	timers.....	8
FCC.....	25	TS Calibrate.....	15
Flash.....	8	User input.....	15
hardware reset.....	22	Using your EvoPrimer applications.....	13
impossible to recharge.....	8	VCC_DB.....	8
Insert a target board.....	12	Wait.....	21
LCD.....	8	watchdogs.....	8
LCD display.....	9	Web site.....	4
Lead.....	25	WEEE.....	25
LED (L0).....	13		

10. History

Date	Modifications
Sep 2011	Initial version.

Disclaimer

Information in this document is subject to change without notice and does not represent a commitment on the part of the manufacturer. The software described in this document is provided under license and may only be used or copied in accordance with the terms of the agreement. It is illegal to copy the software onto any medium, except as specifically allowed in the license or non disclosure agreement.

No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the purchaser's personal use, without prior written permission.

Every effort has been made to ensure the accuracy of this manual and to give appropriate credit to persons, companies and trademarks referenced herein.

This manual exists both in paper and electronic form (pdf).

Please check the printed version against the .pdf installed on the computer in the installation directory, for the most up-to-date version.

The examples of code used in this document are for illustration purposes only and accuracy is not guaranteed. Please check the code before use.

Copyright © Raisonance 1987-2011 All rights reserved