



# **STM32-Primer2**

**User manual**

**Fun, easy introduction kit  
for STM32 microcontrollers**

**Document version**  
2009-01-05

## Contents

1 PRESENTATION.....	4
2 THE STM32-PRIMER2 HARDWARE.....	5
2.1 List of Contents.....	5
2.2 Description.....	5
2.3 Opening the STM32-Primer2 case.....	6
2.4 Inserting and removing a MicroSD card.....	6
2.5 STM32 microcontroller features.....	7
2.6 3D MEMS accelerometer.....	7
2.7 Power supply.....	7
2.8 For further information.....	8
3 GETTING STARTED.....	9
3.1 Connect battery, charge and power up.....	9
3.2 Play.....	9
3.3 Install the Ride7 software toolset for ARM.....	10
3.4 Play with an STM32 application example.....	10
4 MANAGING YOUR CIRCLEOS APPLICATIONS.....	11
4.1 CircleOS architecture.....	11
4.2 The CircleOS Scheduler.....	11
4.2.1 Initialization stage.....	12
4.2.2 Periodic systick interrupt.....	12
4.2.3 Application scheduler.....	12
4.3 CircleOS Memory map.....	13
4.3.1 Resources used by CircleOS.....	14
4.4 Managing applications on your STM32-Primer2.....	15
4.5 Selecting the current application .....	16
4.6 Downloading new applications.....	16
4.7 Restoring the factory configuration.....	16
4.8 Resetting your STM32-Primer2.....	17
4.8.1 Hardware reset.....	17
4.8.2 Software reset.....	17

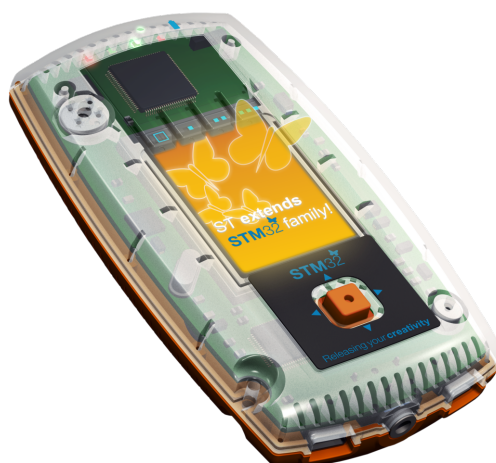
5 DEVELOPING CIRCLEOS APPLICATIONS.....	18
5.1 Developing your first CircleOS application.....	18
5.2 Libraries.....	19
5.3 Debugging your application.....	19
5.4 Sharing your application with the Circle community.....	19
6 RECYCLING.....	20

## 1 Presentation

The STM32-Primer2 is an innovative, low-cost evaluation and development package that is designed to provide a fun and easy introduction to the features of the STM32 with ARM Cortex™-M3 core.

Based on the previous STM32-Primer (released in 2007), Raisonance developed the STM32-Primer2 thanks to an innovative design and a very active user community. With more user interface options, longer lasting Li-Ion power supply, more hardware peripherals and an STM32F103V with more memory (512 KB of Flash ROM, 64 KB of RAM) than before, the Primer2 is the complete, low-risk environment for starting your application.

The STM32-Primer2's ergonomic design with MEMS-based controls (navigate by tilting the tool left, right, backward or forward), touch-screen backlit LCD display, a joystick, 4 menu push buttons, CAN and IrDA a MicroSD memory card slot and an hardware audio codec with microphone/speaker/headphones interface provide fun and easy control of the included demonstration firmware that includes graphical user interface and games based on the resources of the STM32 microcontroller.



The included firmware (CircleOS task scheduler, system services and demonstration applications) implements low level functions driving the various STM32 peripherals. In addition, it includes features for dynamic loading and management of new applications. All firmware, demos (C sources and projects) and more future applications are available for free download at the STM32-Primer2 dedicated site, <http://www.stm32circle.com/>.

The STM32-Primer2 and Ride7 software toolset provide everything needed for programming the STM32 and debugging applications, including:

1. USB host connection for in-circuit programming and debugging
2. Ride7 integrated development environment for code editing, device programming and application debugging (debug up to 32K of code, with included version. For information about upgrade to an unlimited version of Ride, visit <http://www.stm32circle.com/resources>).
3. GNU C /C++ compiler (unlimited compiling)

## 2 The STM32-Primer2 hardware

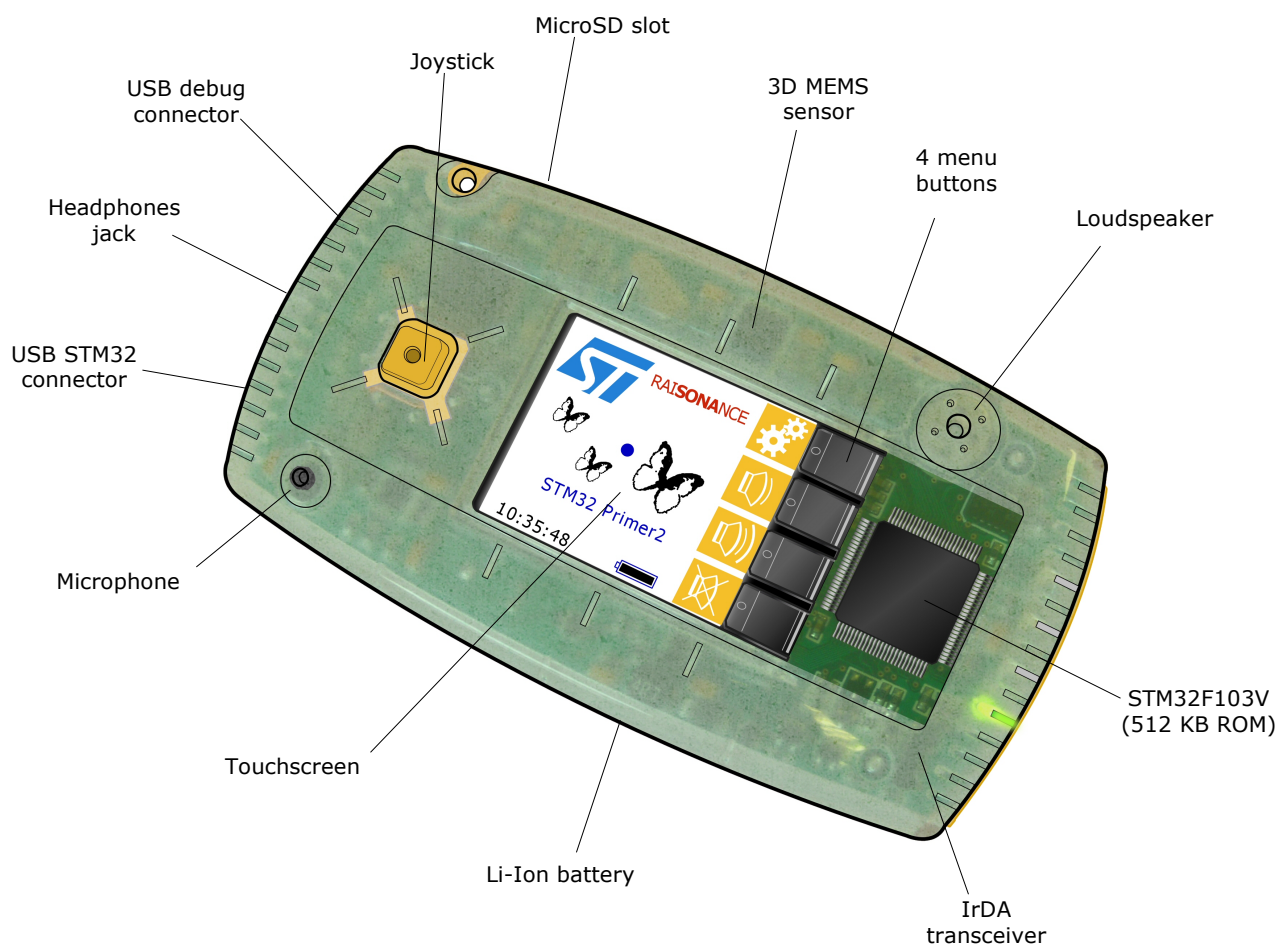
### 2.1 List of Contents

You will find when opening the box:

1. The STM32-Primer2 in its plastic case,
2. A USB cable for host PC connection that can be used to program and debug the STM32 microcontroller. Alternatively this cable can be used to communicate with the STM32-Primer2 if you program a USB client application.
3. A CD-ROM containing Ride7 along with its RKit-ARM and all the STM32-Primer2 documentation.

### 2.2 Description

Here is a brief overview of the main components of the STM32-Primer2:



The STM32-Primer2 provides the following devices:

- A touch-screen LCD color display (24-bits color, 128x160 pixels),
- 2 USB connectors:
  1. One marked "Debug" to connect to a PC running Ride for application development,
  2. One marked "STM32" that allows the embedded application to communicate with an external USB host.
- One 4-directions joystick with push button,
- 4 menu buttons,
- An on-board microphone and a loudspeaker providing sound recording and playback capabilities,
- an headphones connector,
- One accelerometer (MEMS) that captures the 3D-position information related to the STM32-Primer2, and which is used to navigate through the menus, and to move the pointer,
- One IrDA transmitter that allows communication between two STM32-Primer2,
- One extra connector is linked to some unused I/O pins of the STM32 in order to add extra peripherals.

### **2.3 Opening the STM32-Primer2 case**

Opening the plastic case of the STM32-Primer2 is easy (no screw, no clip). Just separate the two halves of the case and separate the orange part from the rest.

Note that the board is screwed onto the transparent plastic half of the case. This should not be removed as the display connector (flex) is very sensitive and may suffer from manipulation when detached from the case.

Warning: In order to avoid harming your STM32-Primer2, the board should not be unscrewed from the transparent plastic case.

### **2.4 Inserting and removing a MicroSD card**

The STM32-Primer2 is equipped with a MicroSD (Micro Secure Digital) card slot, which can be used for storing data.

In order to insert a MicroSD card (not provided) into the slot, you just have to open your STM32-Primer2 case. After the 2 plastic parts are separated away, the MicroSD card slot is available on the board, close to the battery.

Insert your MicroSD card into the slot (card contacts on the bottom), then gently slide it. Be careful not to harm your MicroSD card nor your Primer2!

Note: To remove the MicroSD card, do not pull it away from the slot, but push it deep into the slot. A spring mechanism will automatically extract it.

## 2.5 STM32 microcontroller features

The STM32-Primer2 is equipped with an STM32F103VET6, one of the new ST, Cortex-based, 32-bit microcontrollers. The main characteristics of this device are:

- ARM 32-bit Cortex™-M3 CPU, 72 MHz, 90 DMips with 1.25 DMips/MHz,
- 512KB of Flash program memory, 64KB SRAM,
- Embedded oscillators (for high-speed crystal + RTC),
- SWD debug interface,
- Fast input/output: up to 80 I/Os, ADC, DAC,
- Embedded communication peripherals: USB 2.0, CAN, USART, SPI, I2C, LIN, IrDA
- Multiple timers; watchdog, PWM, SysTick timer, ...

## 2.6 3D MEMS accelerometer

The STM32-Primer2 is equipped with a MEMS inertial sensor (LIS3LV02DL from STMicroelectronics). This device is used by the STM32-Primer2 as a human interface device to select commands, in coordination with a graphic pointer. When you start the STM32-Primer2 for the first time, you will see a small ball moving according to the orientation of the STM32-Primer2 circuit. The information about the 3D position is provided by the MEMS.

You will find complete documentation of the 3D accelerometer on the companion CD-ROM. Examples are provided that allow evaluation of various functions of the MEMS usage.

## 2.7 Power supply

The STM32-Primer2 features a 400mAh Li-Ion rechargeable battery, equipped with a voltage regulator along with a battery charger.

When one of the USB connectors is linked to a PC host voltage supplied by the PC is used to recharge the battery.

When no USB host is connected, the battery is used to supply the power for the STM32-Primer2. When the battery is fully charged, the STM32-Primer2 can be used for about 6 hours. The duration of the batteries depends on the Primer2 activity. For instance, it can be extended -or reduced- by changing the setting of the backlight intensity (see the menu "Settings") and the CPU frequency.

## **2.8 For further information**

This document describes the STM32-Primer2, basic use and its hardware and firmware features. Further information about tools intended for use with the Primer2 may be found in:

- "LIS3LV02DL MEMS Inertial Sensor Data Sheet", available from <http://www.st.com/>.
- "STM32F103 Performance AC Line", available from <http://www.st.com/>.
- "STM32F10x Flash programming manual", available from <http://www.st.com/>.
- "STM32F10x advanced ARM-based 32-bit MCU reference manual", available from <http://www.st.com/>.
- "STM32F10x Datasheet", available from <http://www.st.com/>.
- The "Cortex-M3 Technical Reference Manual" document describes the Cortex-M3 Core, and is available directly from <http://www.arm.com/>.
- "ST7732 Datasheet". This document describes the 262K colors single-chip TFT controller/driver, and is available from <http://www.sitronix.com.tw/>.
- The "LCM-TGG000177YP02-01\_SPEC.pdf" document describes the characteristics for the 128x160 display of the STM32-Primer2.
- The full GNU Compiler Collection (GCC) manuals are available from the Ride7 environment.

Please visit the Circle community web site at <http://www.stm32circle.com/> for more resources.

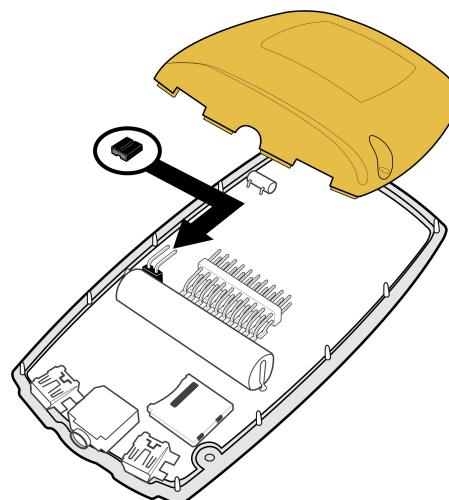


## 3 Getting started

### 3.1 Connect battery, charge and power up

Your STM32-Primer2 is initially shipped with the battery disconnected, in order to prevent discharging and potentially harming the battery. You need to connect the battery before operating your Primer2:

1. Open the STM32-Primer2's case by pulling the top and bottom halves of the case apart. There are **no** catches or releases that hold the case together.
2. Fit the jumper so that it straddles the two pins to connect the power supply.
3. Close the case.
4. Connect the Primer2 to the USB port on a PC to charge its battery. Only the Red LED (L4) will be 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 Primer2.



### 3.2 Play

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

Tilt the Primer2 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 Primer2 including the pointer, backlight for the LCD display and time
Maze	This is an addictive game provided as an example. Full source is available from <a href="http://www.stm32circle.com">www.stm32circle.com</a> .
Applic	Select an application to run. The application that you choose will then appear in the main menu.
About	Display information about the CircleOS version and available memory.
Shutdown	Turn off the Primer2. To restart the Primer2, simply press the push button.

**Note:** When you receive the Primer2, 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 Primer2, move the blue dot around the main screen. It takes a little practice.

#### Preinstalled applications

The Primer2 includes pre-installed applications:

- **Maze** – A game where the player navigates inside a labyrinth, eating dots to win points while avoiding ghosts.

- **Breakout** – A game where the player uses a paddle to bounce a ball against a wall of blocks, destroying the block to win points. Win the game by destroying all the blocks.
- **COMMTTEST** – Tests the MicroSD card connector.
- **EchoIrDA** – This is an IrDA echo application, which makes it easy to connect your Primer2 to another IrDA appliance such as a cell phone or another Primer2.

### Configuration menu

Configuration and test applications have been pre-installed in order to check your STM32-Primer2's capabilities. From the main menu of the STM32-Primer2, launch the "Config" command. The following parameters can be set:

1. **CPU Freq.** This lets you choose the speed of your STM32-Primer2.
2. **Backlight.** This lets you tune the backlight intensity. Note that the backlight is the main source of power consumption (with the LCD monitor itself). Reducing the backlight intensity allow to extend the duration when Primer2 is powered by the battery.
3. **Time.** This sets your STM32-Primer2 time. Note that the RTC clock will remain valid even if your STM32-Primer2 is shut off.
4. **Test.** This performs a quick factory test of the STM32-Primer2.

You can add more applications to your STM32-Primer2 by downloading them on the web site [www.stm32circle.com/projects](http://www.stm32circle.com/projects). Refer to the chapter 4 "Managing your CircleOS applications".

## 3.3 Install the Ride7 software toolset for ARM

The CD-ROM contains many resources to help you evaluate the STM32 family:

- 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 STM32-Primer2 from your PC.
- The Rkit-ARM, which is the Ride7 add-on that contains the complete GNU software toolchain based on the GCC compiler.
- Some utilities to manage your Circle applications (Circle is the OS embedded on your STM32-Primer2; refer to Chapter4.1 "CircleOS architecture" for details).
- Documentation: Datasheets, user manuals for the different components of the STM32-Primer2.

To install, just insert the Raisonance mini CD-ROM on your PC, a menu will appear. Just follow the online instructions for installing the software.

Note: Ride7 should be installed *before* connecting to this USB port.

## 3.4 Play with an STM32 application example

Some example are installed with the Rkit-ARM for Ride7.

For instance, the following example will help you build and program a very simple STM32-Primer2 application:

`"[RIDE7_INSTALL_DIR]\Examples\ARM\Primer\STM32Primer2\toggle\toggle.rprj"`

## 4 Managing your CircleOS applications

### 4.1 CircleOS architecture

The STM32-Primer2 is equipped with an STM32F103 that contains 512KB of FLASH ROM and 64KB of RAM.

The STM32-Primer2 embeds the CircleOS operating system (source files are available from <http://www.stm32circle.com/> after free registration). It provides services that will help you develop your STM32-Primer2 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. ...

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 will be scheduled by the CircleOS with full privileges on the device, until it explicitly quits.

### 4.2 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.

### 4.2.1 Initialization stage

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

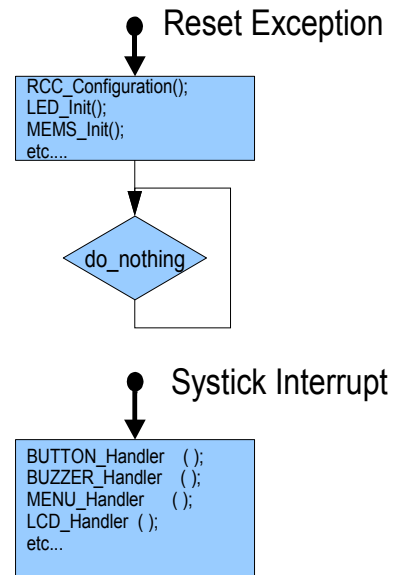
The systick period depends on the RCC settings: it can be modified through the menu "Config | CPU Speed" to the following predefined values:

Level	CPU freq	Systick Freq
1	18MHz	0.75 KHz
2	24MHz	1 KHz
3	36MHz	1,5KHz
4	48MHz	2KHz
5	72MHz	3KHz

The (CPU\_freq / SysTick = 24000) ratio is applied for all these values.

Initialization

Periodic Actions



### 4.2.2 Periodic systick interrupt

The periodic systick makes a call to the CircleOS systick interrupt handler. This interrupt handler will perform a short processing on each of the STM32-Primer2 components: LEDs, button, buzzer, LCD.

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

I2S transfers towards the audio Codec chip are handled through the I2S interrupt.

### 4.2.3 Application scheduler

CircleOS is the base application of your STM32-Primer2. It will handle the menu selections and react to user actions.

Once an application is run (usually through a menu selection), CircleOS will first call an initialization routine for the application, then will repeatedly call the application handler at the Systick frequency until it returns a MENU\_LEAVE value.

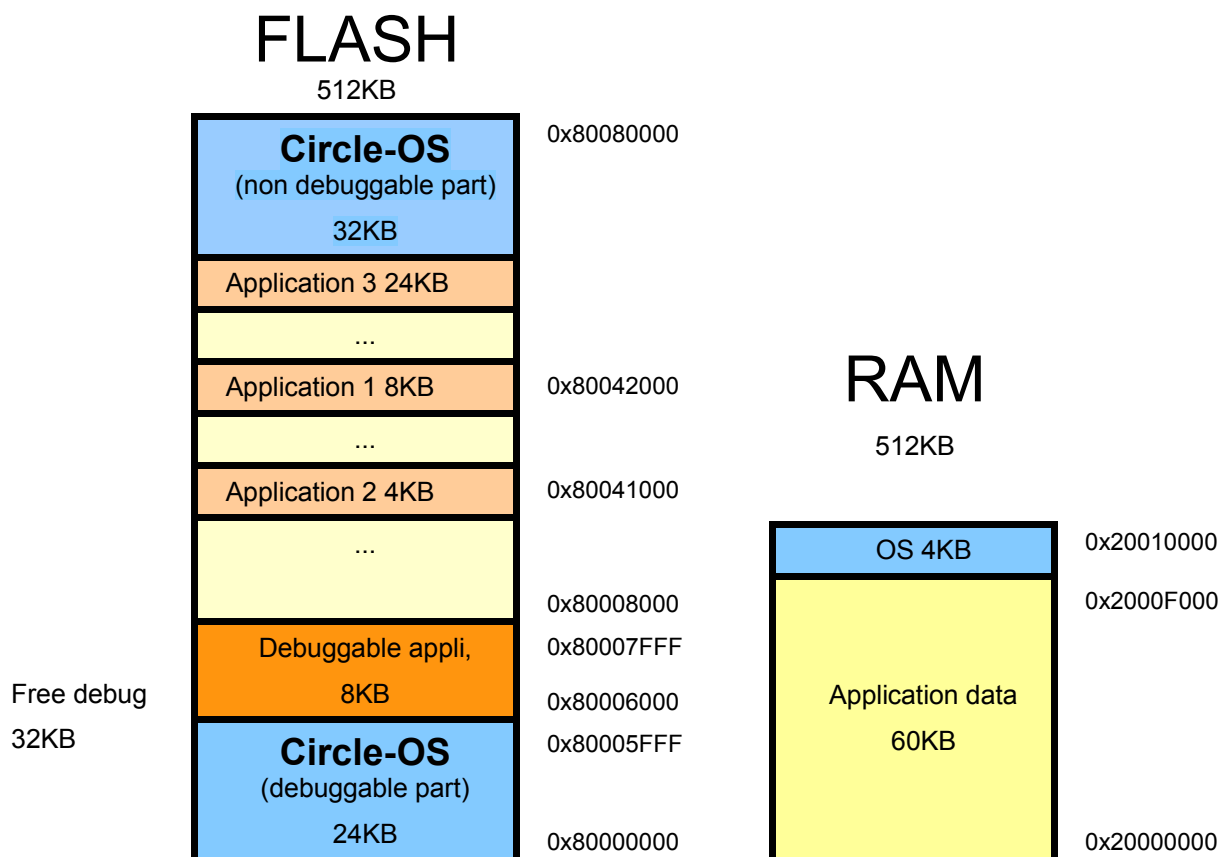
You will find more details about programming CircleOS applications in Chapter 5 "Developing CircleOS applications".

### 4.3 CircleOS Memory map

The CircleOS firmware requires 64 KB of FLASH and 4 KB of RAM (including the stack usage for the applications). The remaining 448 KB (512 – 64) are available for applications, which can be added or removed at will using a programming tool (see below).

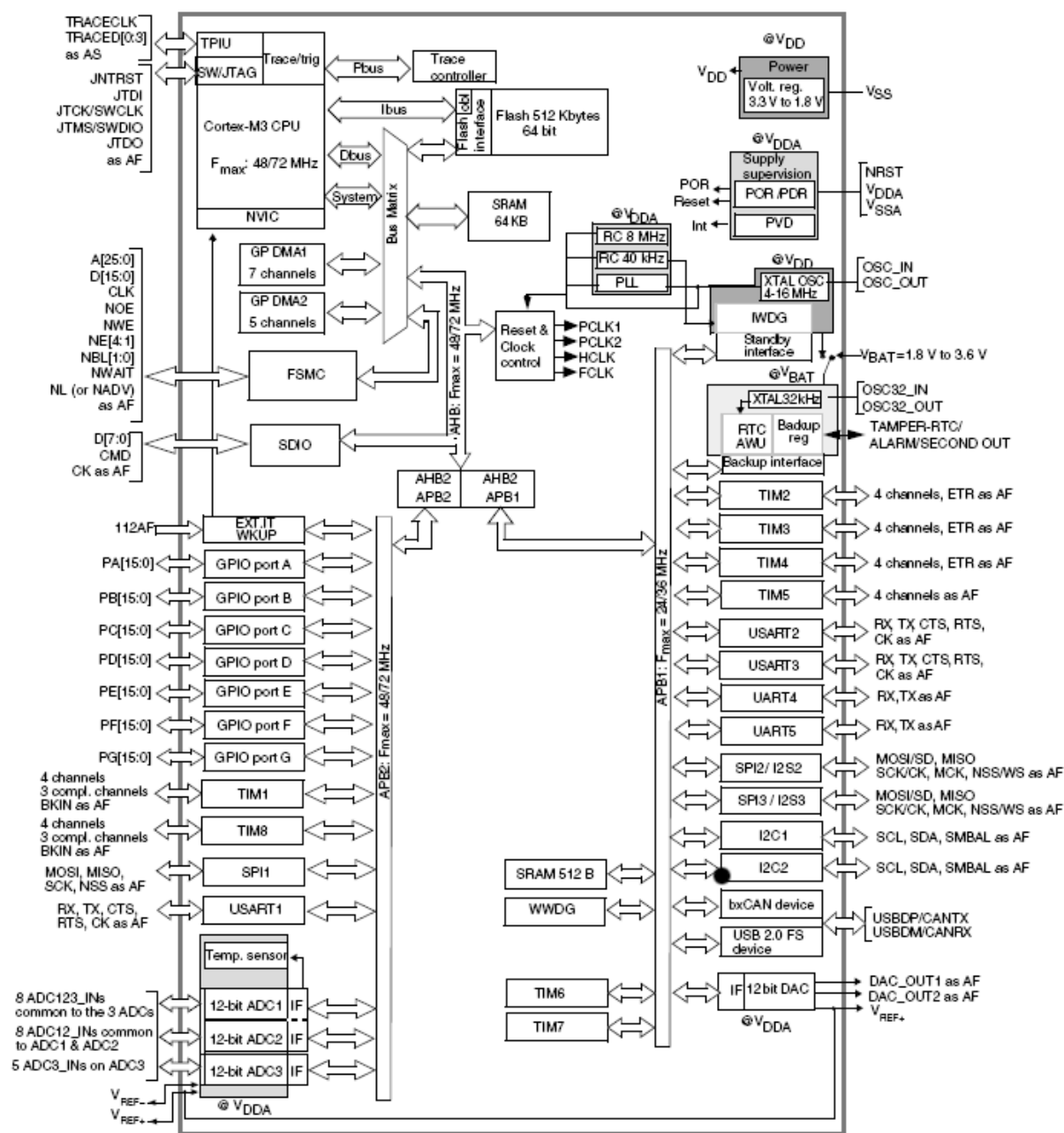
Flash memory can be programmed in 1KB blocks only.

The following figure shows an example of memory mapping:



## 4.3.1 Resources used by CircleOS

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



#### 4.4 Managing applications on your STM32-Primer2

The applications can be managed by the “**circle\_mgr.exe**” utility. You will find it in the “[RIDE7\_INSTALL\_DIR]\Bin” directory.

This utility dedicated to the STM32-Primer2 allows:

- 4 Listing of the currently loaded CircleOS applications.
- 5 Adding of new CircleOS applications.
  - Removing of CircleOS applications.
  - Checking 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	<p>List the loaded applications.</p> <p>The following information will be output:</p> <pre>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</pre>
Add	Afilename	<p>Add a new application (object file).</p> <pre>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</pre>
Erase	E* Eappname	<pre>circle_mgr.exe EMaze /*remove only 'Maze'*/ circle_mgr.exe E* /*remove ALL apps */</pre>
Wait	W	<p>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.</p>
Start	S	<p>Start CPU execution</p>

**Caution:** Any hex file can be programmed to the FLASH memory of your STM32-Primer2 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 4.7 “Restoring the factory configuration” for details).

## 4.5 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.

## 4.6 Downloading new applications

You will find on the <http://www.stm32circle.com/projects> Circle web site 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 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**.

## 4.7 Restoring the factory configuration

If you have been experimenting with CircleOS applications and have modified your STM32-Primer2 configuration, you may want to restore the initial (factory) configuration.

In order to do this, please follow these steps:

1. Connect your STM32-Primer2's debug USB port to your PC.
2. Power-up your STM32-Primer2 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 if you did not install Ride7 in its default location):

```
cd "C:\program files\Raisonance\Ride"
```

5. Now change the current directory to the STM32-specific library directory. This can be done with the following command:

```
cd lib\ARM\CircleOS
```

6. Erase your STM32-Primer2, reprogram it with its factory ROM image, which is in the “circle.hex” file. Then restart the device. These operations can be done with the following single command:

```
cortex_pgm TSTM32F103VET6 E PPrimer2_Circle_Factory.hex S
```

- The TSTM32F103VET6 option selects the appropriate device.
- The E command erases the ROM.
- The PPrimer2\_Circle\_Factory.hex command programs the device with the proper hex file.
- The S command restarts the device.

These operations take approximately 30 seconds.



## 4.8 Resetting your STM32-Primer2

In case you need to reset your STM32-Primer2, several mechanisms can be used:

### 4.8.1 Hardware reset:

A hardware reset will restart your Primer2 (restoring it to a clean state), but will not remove its ROM contents.

- Remove any USB cable.
- Open your STM32-Primer2's case to temporarily remove the battery jumper.
- Replace your battery jumper, close the case of your STM32-Primer2 and restart it by pressing its button.

### 4.8.2 Software reset

A software reset will erase all the flash memory in your STM32-Primer2 and all applications and the CircleOS will be suppressed.

- Connect your STM32-Primer2's debug USB port to your PC.
- Power-up your STM32-Primer2 by pressing its button.
- Open a command prompt from Windows (Navigate to "Start | Programs | Accessories | Command prompt")
- Reset your STM32-Primer2 using the **cortex\_pgm.exe** utility installed in "[RIDE7\_INSTALL\_DIR]\bin" directory. This must be done with the following command:  
cortex\_pgm S

To reload the initial program, use the following command:

```
cortex_pgm TSTM32F103VET6 PPrimer2_circle.hex S
```

where "Primer2\_circle.hex" is the full application, which is available on the web site.

## 5 Developing CircleOS applications

The full source files of CircleOS are available on the <http://www.stm32circle.com/> web site.

Once registered, you will be able to download them, along with many resources for developing your application.

The games originally delivered with the STM32-Primer2 show working application examples.

### 5.1 Developing your first CircleOS application

Creation of a CircleOS application is done automatically in Ride7:

1. Navigate to "Project | New project".
2. Leave the "Type" selection list to "New application".
3. In the "Processor" selection list, select the "STM32\_Primer2\_CircleOS" device from the ARM family.
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
7. Your new project will be created, with an application containing a CircleOS application skeleton as well as the Circle.elf and FAT.elf files necessary to connect your application to CircleOS.
8. Open the "Application.c" file.
9. Search for the Application\_Name variable in the file.
10. Change the Application\_Name value from "My App" to "HELLO".
11. In the Application\_Handler function, create a new string as follows:

```
const char msg[] = "Hello, World!";
```

12. Use the DRAW\_DisplayString CircleOS service to display the "msg" string variable you just created on the STM32-Primer2 display:

```
DRAW_DisplayString( 5, 20, msg, sizeof(msg)); // X, Y, string, length
```

13. Connect your STM32-Primer2 to your PC using the USB cable (be sure to use the "debug" USB port of your STM32-Primer2).
14. From Ride, go to "Debug | Start", which will program your application to your STM32-Primer2. This may take about 15 seconds.
15. Go to "Debug | Run".
16. On your STM32-Primer2, select your application name on the main menu.

Your application is now on your STM32-Primer2.

For further information about CircleOS application programming and available OS services, please visit <http://www.stm32circle.com/>.

## 5.2 Libraries

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

1. The STM32 libraries, written by ST, provide access to the embedded peripherals (such as timers, ADC, communication interfaces, thermometer, etc...) of the STM32 microcontroller.
2. The low-level CircleOS functions that provide an easy access to the STM32-Primer2'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:

- On the CD or on the ST web site for the STM32 libraries. Specific documentation about the STM32 library is also available.
- On the <http://www.stm32circle.com/> web site for the CircleOS libraries (registration required).

The libraries documentation is accessible from Ride7.

## 5.3 Debugging your application

In order to debug your application, you must go to "Project | Properties" in Ride7. In the "Configuration selection box, Circle\_Debug and Circle\_Release are available. Make sure you select the Circle\_Debug configuration (which is the default). Ride7 will take care of all the settings required for switching between debug and release mode through the use of these configurations.

Note: The standard STM32-Primer2 is limited to debug in the first 32KB only. A software key can be purchased on <http://www.stm32circle.com/> to allow debugging in the whole 512KB of memory.

## 5.4 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.

## 6 Recycling

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.



This 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>

## History

Date	Description
14/11/08	Initial version.
17/11/08	Corrected the memory mapping with the inclusion of the non-debuggable CircleOS part.
05/01/09	Corrected the images and the document title.

## Disclaimer

### 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 nondisclosure 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 Ride 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 1987-2008 All rights reserved.**