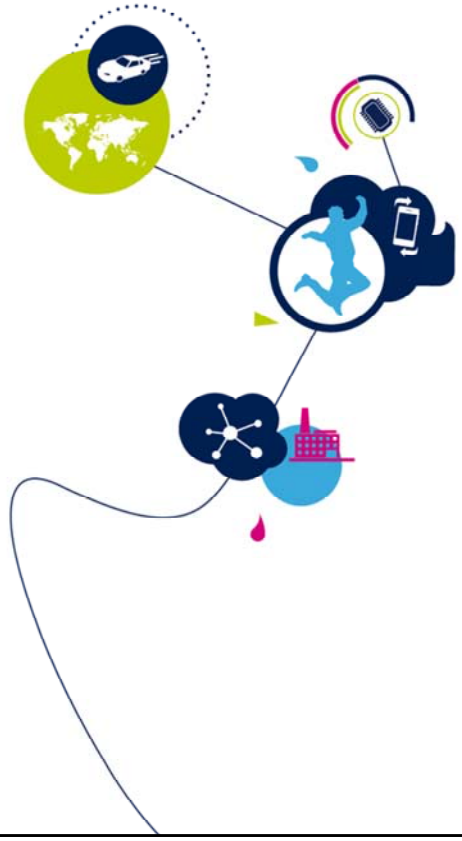
STM32WB – SYSCFG

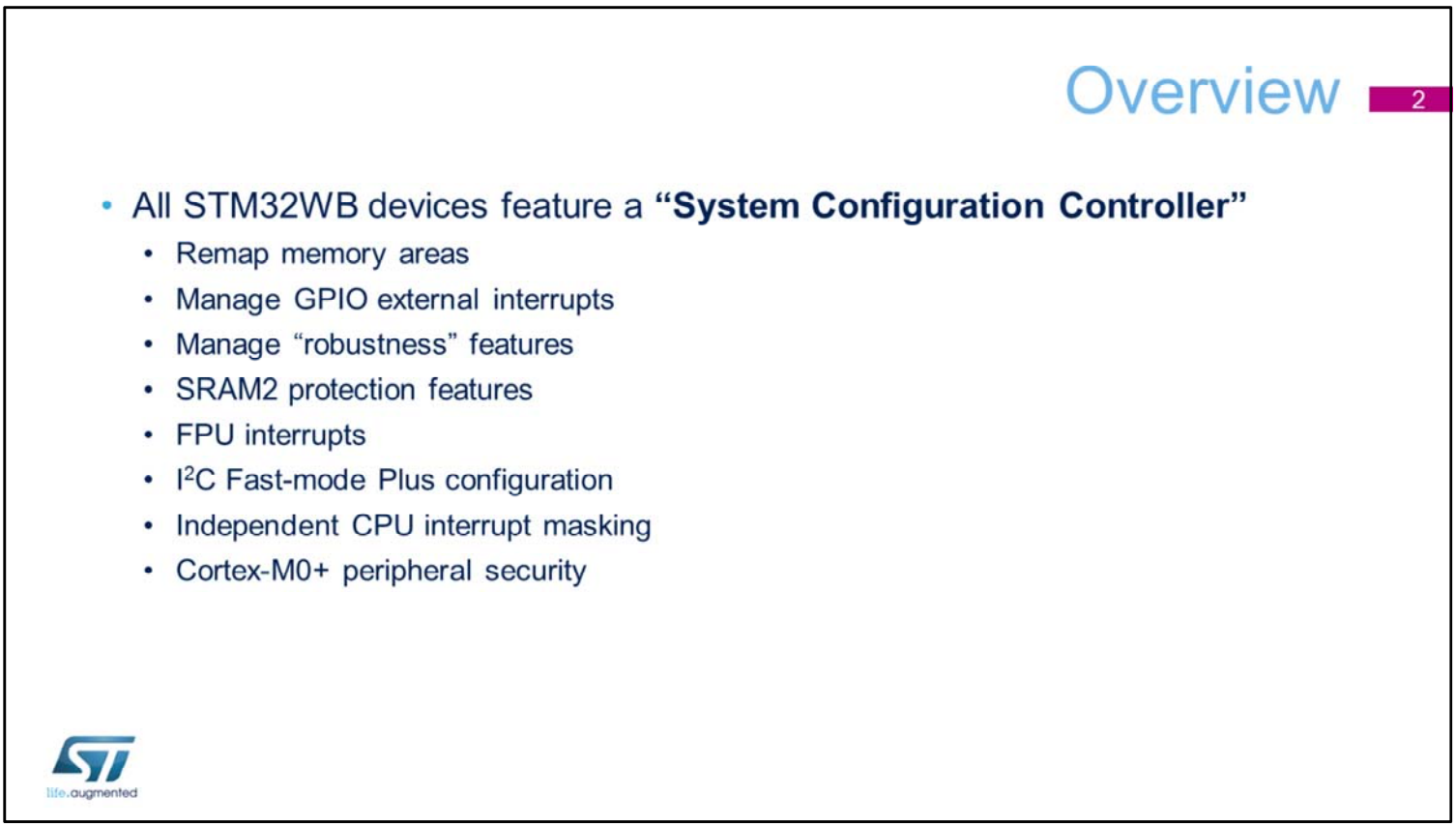
System Configuration Controller

Revision 1.0



life.augmented

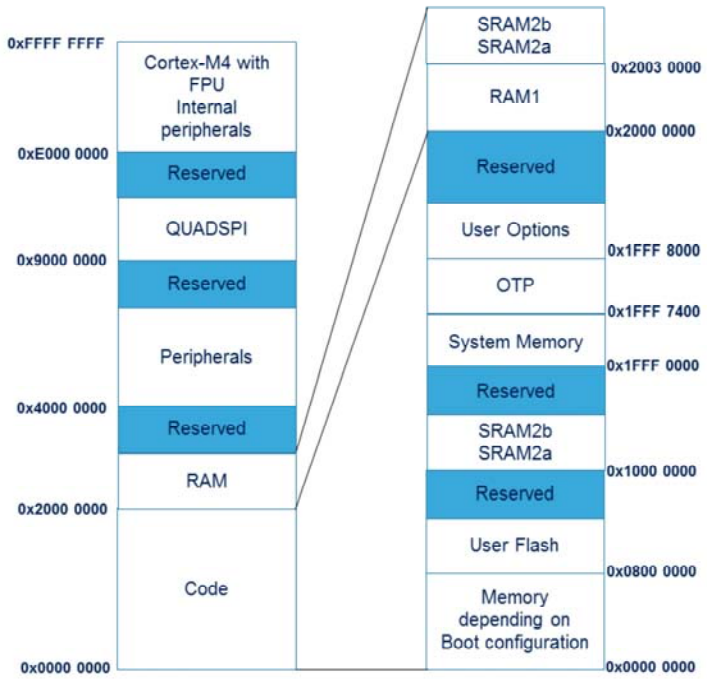
Hello, and welcome to this presentation of the STM32WB System Configuration Controller.



STM32WB devices feature a set of configuration registers. The System Configuration Controller gives access to the following features: Remapping memory areas to Cortex-M4 address 0, managing the external interrupt line connection to the GPIOs, certain robustness features, SRAM2 write-protection and erase, floating point unit interrupts, the configuration of the 20 mA high-drive I/Os used for I²C Fast-mode Plus, peripheral interrupt masking per CPU, and finally the Cortex-M0+ peripheral security.

Cortex-M4 Memory Mapping

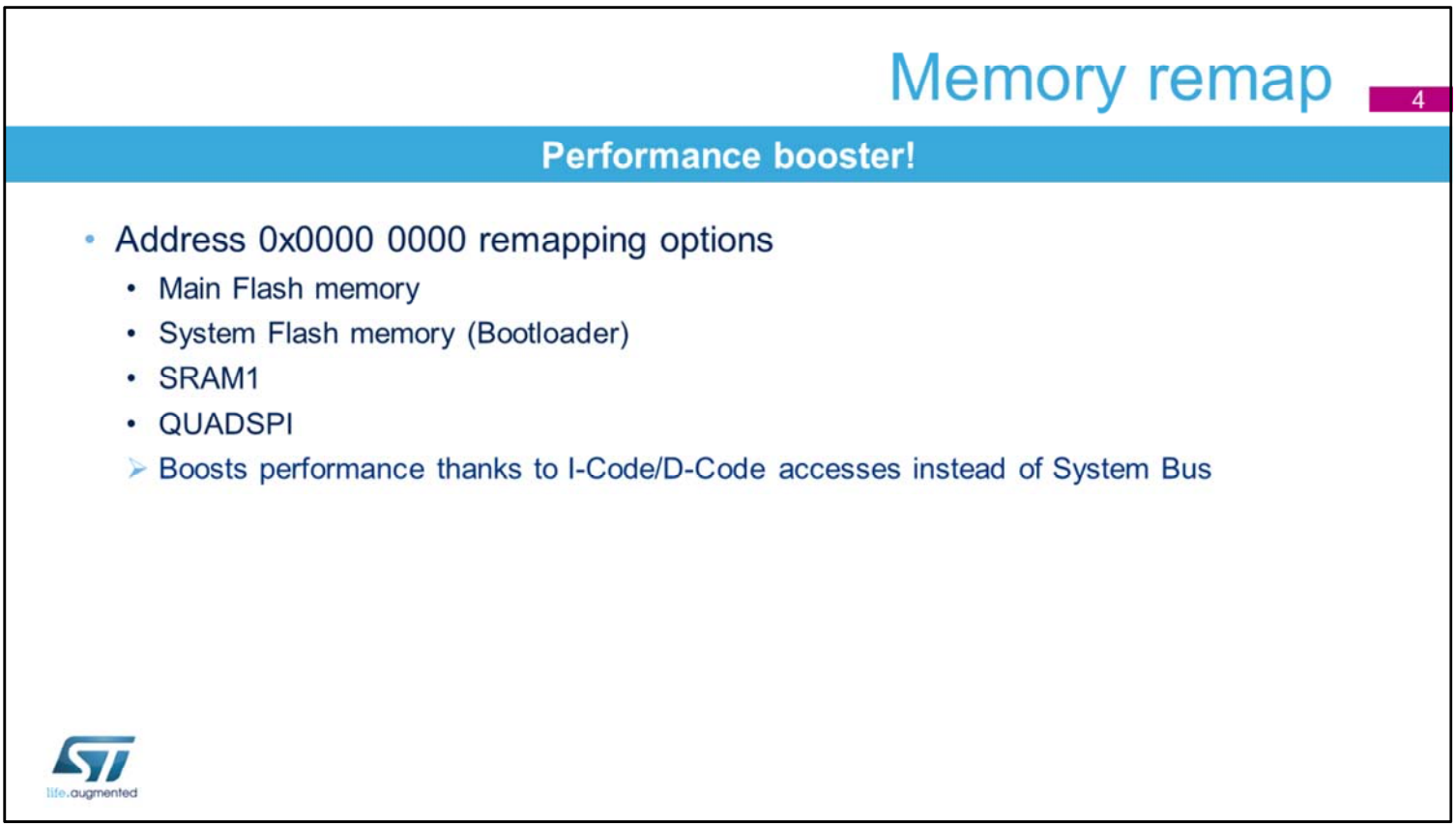
- **Flash memory: up to 1 Mbyte, single bank**
  - @0x0800 0000 (D-code and I-code)
- **SRAM: 256 Kbytes split in 3 parts:**
  - SRAM1:
    - 192 Kbyte @ 0x2000 0000 (S-bus)
  - SRAM2a (backup):
    - 32 Kbytes @ 0x1000 0000 (D-code and I-code)
    - 32 Kbytes @ 0x2003 0000 (allias on S-bus)
  - SRAM2b (non backup):
    - 32 Kbytes @ 1000 8000 (D-code and I-code)
    - 32 Kbytes @ 0x2003 8000 (allias on S-bus)
- QUADSPI
  - external memory interface



Pictured here is the 4 gigabyte linear address mapping of the STM32WB microcontroller.

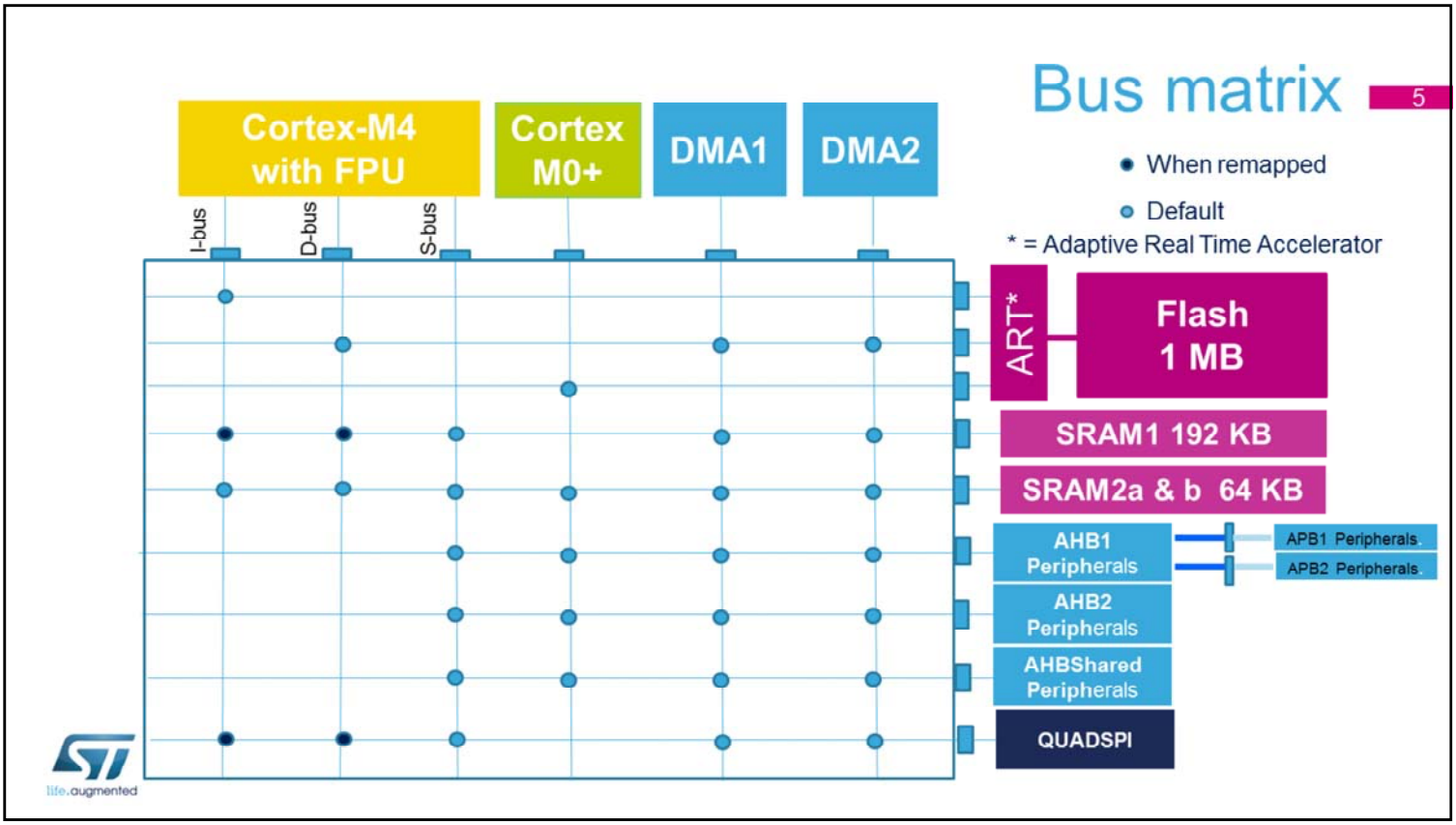
The Flash memory is up to 1 Mbytes, in a single-bank configuration.

The SRAM total size is 256 Kbytes. It is split into 3 parts: SRAM1 is 192 Kbytes starting from address 0xx20000000 and SRAM2a backup RAM is 32 Kbytes starting from address 0x10000000 and also aliased at address 0x20030000 followed by SRAM2b non-backup is also 32 Kbytes starting from address 0x10008000 and aliased at address0x20038000. SRAM1 is located in the usual ARM memory space for RAM on the S-bus, while SRAM2a and SRAM2b can also be directly accessed through Data code and Instruction code buses allowing zero wait states, used for code execution.



The memory remap at Cortex-M4 address 0 allows the boost of the performance thanks to Instruction and Data bus access instead of using the System bus.

The memory remap at address 0 is selected using the MEM_MODE bits in the System Configuration Remap register. They allow the selection of either the main Flash memory, or the system Flash memory, the SRAM1, or the QUADSPI.



Here we have the STM32WBs bus matrix. The bus masters are shown on top, and the Cortex-M4 core, the Cortex-M0+ core and the two DMA controllers communicate with the bus slaves, shown on the right via the circled intersections.

The Flash memory is read through the accelerator. Cortex-M4 instructions are fetched through Instruction bus and Literal Pools are read through the Data bus. The SRAM1 is accessed by default by the System bus, and can be accessed though I-bus and D-bus when it is remapped at address 0, shown by the dark blue circles in order to increase performance. SRAM2 is accessible through the I-bus and D-bus allowing zero-wait-state code execution and through the S-bus. The Quad-SPI can be read and executed through the System bus by default, and can be remapped at 0 to increase

performance.

The Cortex-M0+ also reads the Flash memory through the Adaptive Real Time accelerator (ART), and has access to the SRAM2a and SRAM2b memories and the AHB1, AHB2 and AHB Shared peripherals.

The two DMAs can access all memories and peripherals.

Different bus masters are able to access different memories and peripherals simultaneously via the bus matrix, enabling high performance compute operations. Simultaneous master accesses to the same bus is handled via round-robin arbitration.

Cortex-M4 Boot modes

Boot mode selection		Boot mode
nBOOT1 (option bit)	BOOT0 (pin) (or nBOOT0 option bit)	
X	0	User Flash memory
1	1	System memory (bootloader)
0	1	SRAM1

There are 3 boot modes which are selected by the nBOOT0 option bit or by the BOOT0 pin and an option bit named nBOOT1. When the BOOT0 pin or option bit is at a low level, the STM32WB boots from the User Flash memory, which is aliased at address 0. This is the standard method of booting the STM32WB.

When the BOOT0 pin or option bit is at a high level, the nBOOT1 option bit determines the boot mode.

Cortex-M4 Boot modes

- The pin BOOT0 is shared with PH3 GPIO. When remapped
  - Two additional option bits (nBOOT0 and nSWBOOT0) are used to select the boot mode
  - Also, a Flash Empty check mechanism is implemented (*).

Boot mode selection					Boot mode
nBOOT1 (option bit)	nBOOT0 (option bit)	BOOT0/PH3 (pin)	nSWBOOT0 (option bit)	Main Flash empty	
x	x	0	1	0	User Flash memory
x	x	0	1	1	System memory
x	1	x	0	x	User Flash memory
0	x	1	1	x	SRAM1
0	0	x	0	x	SRAM1
1	x	1	1	x	System memory
1	0	x	0	x	System memory

life.augmented

(*) to force the boot from system Flash instead of the main Flash if the first Flash memory location is not programmed.

In addition to the nBOOT1 option bit, Boot mode is selected either by the BOOT0 pin or the nBOOT0 option bit depending on the value of the nSWBOOT0 option bit in the FLASH_OPTR register as shown in this table.
A Flash Empty Check mechanism is implemented to force the boot from the system Flash memory instead of the main Flash memory if the first Flash memory location is not programmed.

The default level for the option bits is high, enabling the boot from the system memory portion of the Flash memory. The other option is booting from the SRAM1 memory region, which may be used for debugging purposes.

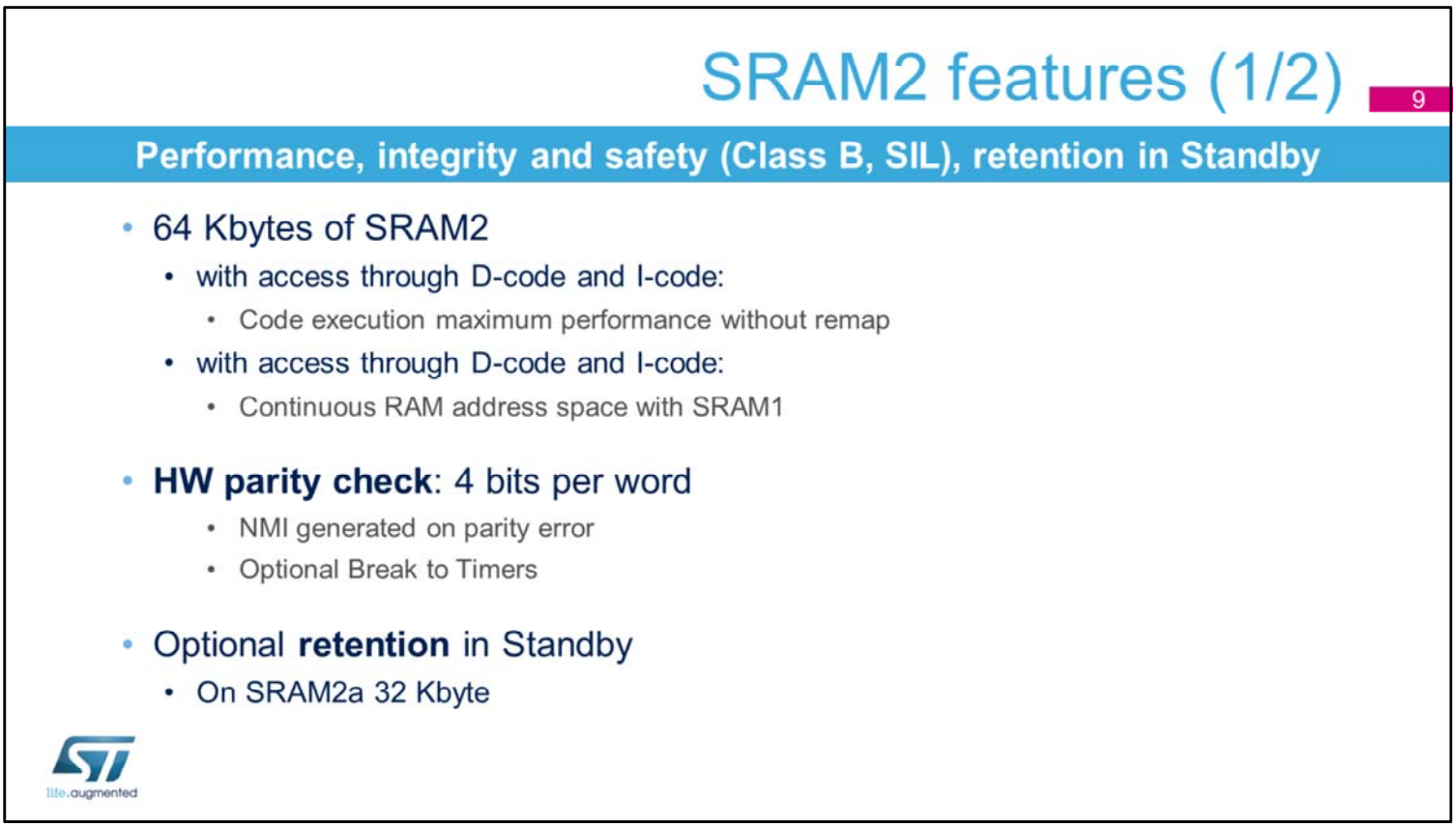
Bootloader

8

Protocol	I/Os and Comments	Comments
USART	USART1 on pins PA9/PA10	
USB	USB DFU interface on pins PA11/PA12	Bootloader checks if HSE present : USB clock is HSE If no Bootloader checks if LSE present : USB clock is MSI auto-trimmed with LSE
SPI	SPI1 on pins PA4/PA5/PA6/PA7 SPI2 on pins PB12/PB13/PB14/PB15	
I2C	I2C1 on pins PB6/PB7 I2C3 on pins PC0/PC1	I²C slave address is 0x86

life.augmented

The on-chip bootloader allows the user to program the Flash memory through a serial communications peripheral. The supported protocols are USART, USB, CAN, SPI and I²C.

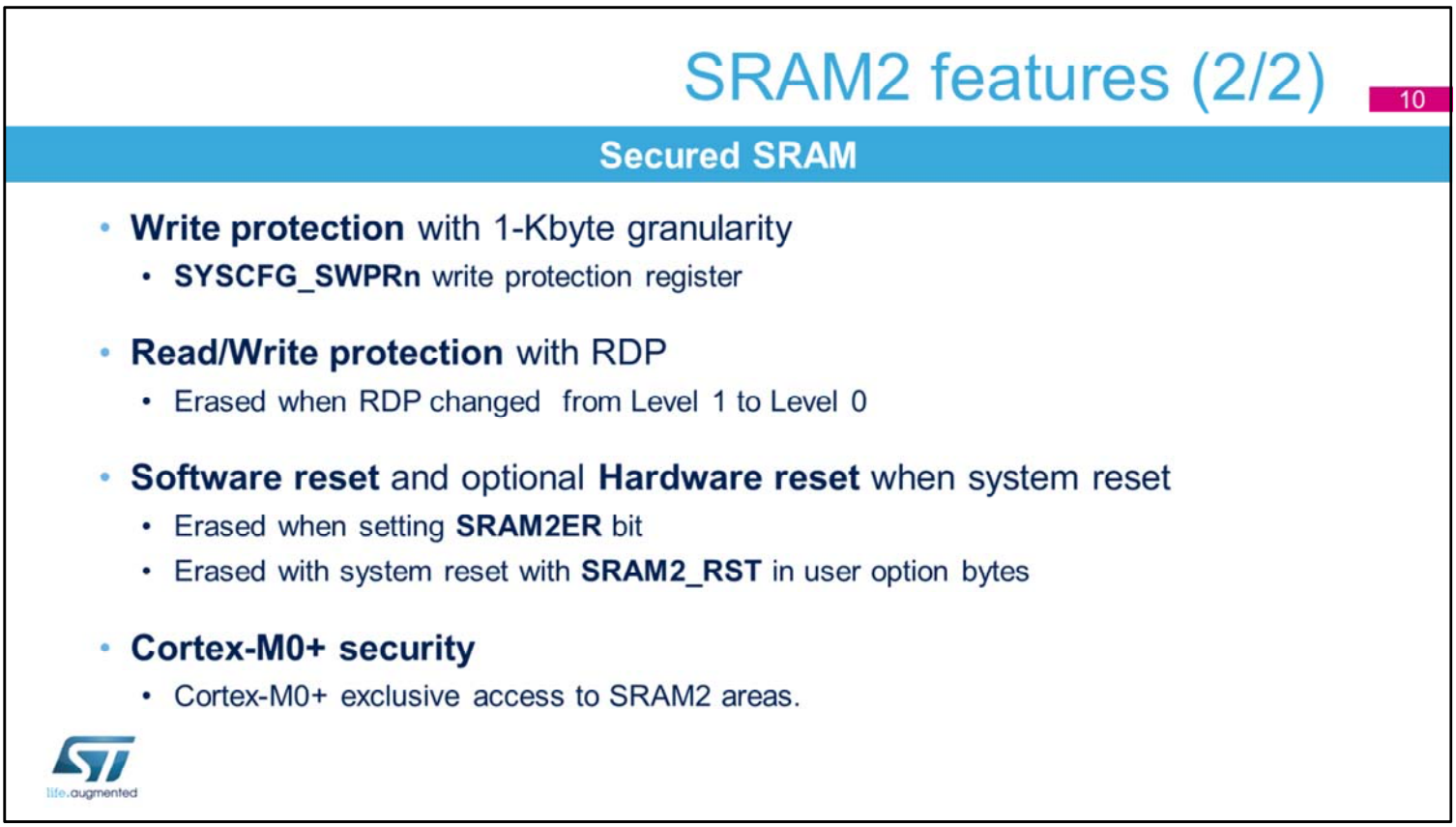


The 64 Kbytes of SRAM2 is particularly suitable for performance, integrity and safety, and low power.

The SRAM2 is accessed through the Data and Instruction busses without any remapping, which enables code execution at zero-wait-states, and also through the S-bus allowing RAM address continuity between SRAM1 and SRAM2 memories.

The SRAM2 supports parity check. The Data bus width is 36 bits because 4 bits are available for parity check (1 bit per byte) in order to increase memory robustness, as required, for instance, by Class B or SIL standards. Class B and SIL are safety standards: Class B is for Home Appliances and SIL for the Safety Integrity Level. The parity bits are computed and stored when writing into the SRAM. Then, they are automatically checked when reading. If one bit fails, a Non-Maskable Interrupt (NMI) is generated. The same error can also be linked to

the Break input of the timers.
The 32 Kbyte SRAM2a content can optionally be retained in Standby mode.



The SRAM2 is also suitable for secure applications. The SRAM2 can be write-protected with a 1-Kbyte granularity.

The SRAM2 can also be readout-protected via the RDP option byte. When protected, the SRAM2 cannot be read or written by the JTAG or serial wire debug port, and when the boot in System flash or boot in SRAM is selected. The SRAM2 is erased when the readout protection is changed from Level 1 to Level 0. Please refer to the System Memory Protections training for further details.

The SRAM2 can be erased by software by setting the SRAM2ER bit in the SRAM2 System Configuration Control and Status register. The SRAM2 can also be erased with the system reset depending on the option bit

SRAM2_RST in the user option bytes.
Two SRAM2 areas, one in SRAM2a and one in SRAM2b can be made secure via user option bytes, only giving exclusive access to the Cortex-M0+ core in these areas.

SYSCFG other features

Safety and robustness

- Safety & Robustness features in Configuration register 2
  - **SRAM2 Parity error** flag
  - **ECC lock** to connect Flash ECC error connection to TIM1/8/15/16/17 Break input
  - **PVD lock** to connect PVD interrupt to TIM1/8/15/16/17 Break input
  - **SPL lock** to connect SRAM2 parity error to TIM1/8/15/16/17 Break input
  - **CLL lock** to connect Cortex M4 Hard Fault interrupt to TIM1/8/15/16/17 Break input

=> **Put timers in application safe state in case of application crash**

life.augmented

The System Configuration Register 2 contains the control and status bits linked to safety and robustness such as the SRAM2 parity error flag, and the control bits to direct some error detections events to the timers' break inputs. This allows timer outputs to be placed in a known state during an application crash. Once programmed, the connection is locked until the next system reset. These internal events include a Flash error-code-correction event, a power voltage detector event, SRAM2 parity error event, and the Cortex M4 hard fault.

SYSCFG other features

- Manage external interrupt (EXTI) connection to GPIOx (x=A,…H)
  - 16 Multiplexers to select EXTIn between PA[n] PB[n] …. PH[n] (n=0,…15)
  - select a GPIOx pin to be used as
    - internal interconnect trigger signal to ADC
    - CPU interrupt and wakeup from Stop mode.
- Configuration register 1
  - FPU interrupts enable
  - I2C GPIO Fast-mode Plus 20 mA drive enable
    - PB6, PB7, PB8, PB9 high drive can be enabled even when not used for I2C
  - I/O analog switches voltage booster

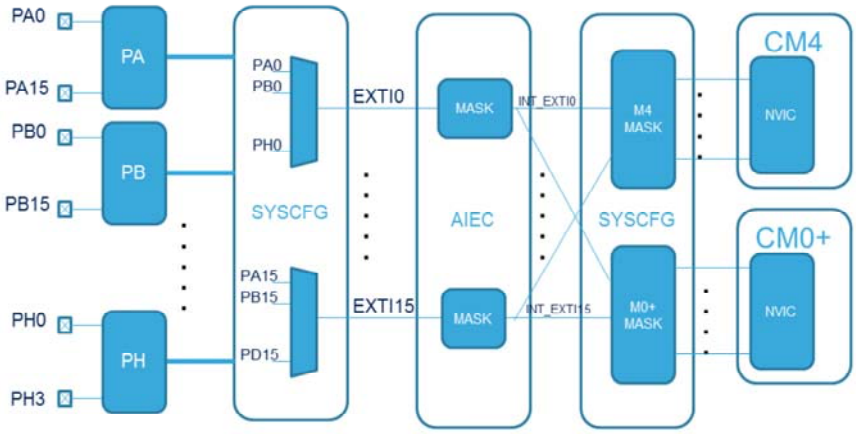
life.augmented

The System Configuration Controller manages the selection of the GPIO to the external interrupt or event signal, which is used as asynchronous external interrupt or event with wakeup from Stop capability. It also allows the selected GPIO pin to be used as an internal interconnect trigger signal to the ADC.

Configuration register 1 contains the floating point unit interrupt control bits. It contains also the I²C Fast-mode-Plus 20 mA drive enable control bits. Four I/Os can be configured with high drive mode even if they are not used as I2C alternate functions. They can be used to drive LEDs for instance.
The I/O analog switch voltage booster is also selected here.

CPU interrupt masking

- To prevent peripheral interrupt to interrupt both CPUs a masking is provided.
  - Individual masks for interrupts sharing the same CPU NVIC vector.
    - EXTIs
    - Timers
    - PVD and PVM
    - ADC
    - Comparator
    - AES
    - True RNG
    - PKA
    - Flash
    - RCC
    - RTC



Peripheral interrupts sharing the same NVIC vector have a mask to prevent them from interrupting both CPUs.

Cortex-M0+ peripheral security

- A secure Cortex-M0+ can dynamically secure peripherals:
  - AES1 (key security)
  - AES2 (full security)
  - true RNG (full security)
  - PKA (full security)
- Secure peripherals have partial or full Coretex-M0+ exclusive register access.
  - Allows key generation and storage security
  - Allow secure data encryption

life.augmented

The AES accelerator 1, the AES accelerator 2, the Public Key Accelerator and the True Random Number Generator peripherals can dynamically be made secure by the Cortex-M0+ firmware through secure register bits in the System Configuration block, enabling access to the secure part of the internal SRAM or Flash memories.

Related peripherals

- Refer to these training modules linked to this peripheral:
  - Reset and clock control (RCC)
  - Power controller (PWR)
  - Interrupts (NVIC-EXTI)
  - Flash memory (Flash)
  - System memory protections
  - Timers (TIM)
  - Inter-Integrated Circuit (I²C)
  - Advanced encryption standard hardware accelerator (AES)
  - Public Key accelerator (PKA)
  - True random number generator (RNG)

life.augmented

In addition to this training, you can refer to the Reset and Clock Control, Power Controller, Interrupts, Flash and System Memory Protections, Timers , I²C , Encryption, Public key, and true random number generator trainings.

References

- For more details, please refer to following resources:
  - AN2606: STM32 microcontroller system memory boot mode
  - AN4435: Guidelines for obtaining UL/CSA/IEC 60335 Class B certification in any STM32 application

life.augmented

For more details, please refer to application notes AN2606 STM32 microcontroller system memory boot mode and AN4435 Guidelines for obtaining UL/CSA/IEC 60335 Class B certification in any STM32 application.