Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

STM8S001J3

16 MHz STM8S 8-bit MCU, 8-Kbyte Flash memory, 128-byte data EEPROM, 10-bit ADC, 3 timers, UART, SPI, I2C

Datasheet - production data

Features

Core

- 16 MHz advanced STM8 core with Harvard architecture and 3-stage pipeline
- Extended instruction set

Memories

- Program memory: 8 Kbytes Flash memory; data retention 20 years at 55 °C after 100 cycles
- RAM: 1 Kbyte
- Data memory: 128-byte true data EEPROM; endurance up to 100 k write/erase cycles

Clock, reset and supply management

- 2.95 V to 5.5 V operating voltage
- Flexible clock control, 3 master clock sources
  - External clock input
  - Internal, user-trimmable 16 MHz RC
  - Internal low-power 128 kHz RC
- Clock security system with clock monitor
- Power management
  - Low-power modes (wait, active-halt, halt)
  - Switch-off peripheral clocks individually
  - Permanently active, low-consumption power-on and power-down reset

Interrupt management

- Nested interrupt controller with 32 interrupts
- Up to 5 external interrupts

SO8N
4.9x6 mm or 150 mils width

Timers

- Advanced control timer: 16-bit, 2 CAPCOM channels, 2 outputs, dead-time insertion and flexible synchronization
- 16-bit general purpose timer, with 3 CAPCOM channels (IC, OC or PWM)
- 8-bit basic timer with 8-bit prescaler
- Auto wakeup timer
- Window and independent watchdog timers

Communications interfaces

- UART, SmartCard, IrDA, LIN master mode
- SPI unidirectional interface up to 8 Mbit/s (master simplex mode, slave receiver only)
- I2C interface up to 400 Kbit/s

Analog to digital converter (ADC)

- 10-bit ADC, ± 1 LSB ADC with up to 3 multiplexed channels, scan mode and analog watchdog
- Internal reference voltage measurement

I/Os

- Up to 5 I/Os including 4 high-sink outputs
- Highly robust I/O design, immune against current injection

Development support

- Embedded single-wire interface module (SWIM) or fast on-chip programming and non-intrusive debugging

This is information on a product in full production.

Contents

1 Introduction 8

2 Description 9

3 Block diagram 10

4 Functional overview 11

4.1 Central processing unit STM8 11

4.2 Single wire interface module (SWIM) and debug module (DM) 12

4.3 Interrupt controller 12

4.4 Flash program memory and data EEPROM 12

4.5 Clock controller 15

4.6 Power management 16

4.7 Watchdog timers 16

4.8 Auto wakeup counter 17

4.9 TIM1 - 16-bit advanced control timer 17

4.10 TIM2 - 16-bit general purpose timer 17

4.11 TIM4 - 8-bit basic timer 17

4.12 Analog-to-digital converter (ADC1) 18

4.13 Communication interfaces 18

4.13.1 UART1 19

4.13.2 SPI 19

4.13.3 I2C 20

5 Pinouts and pin descriptions 21

5.1 STM8S001J3 SO8N pinout and pin description 21

5.2 Alternate function remapping 24

6 Memory and register map 25

6.1 Memory map 25

6.2 Register map 26

6.2.1 I/O port hardware register map 26

6.2.2 General hardware register map 27

6.2.3 CPU/SWIM/debug module/interrupt controller registers 34

7 Interrupt vector mapping 36

8 Option bytes 37

8.1 Alternate function remapping bits 39

9 Electrical characteristics 40

9.1 Parameter conditions 40

9.1.1 Minimum and maximum values 40

9.1.2 Typical values 40

9.1.3 Typical curves 40

9.1.4 Loading capacitor 40

9.1.5 Pin input voltage 40

9.2 Absolute maximum ratings 41

9.3 Operating conditions 43

9.3.1 VCAP external capacitor 45

9.3.2 Supply current characteristics 45

9.3.3 External clock sources and timing characteristics 54

9.3.4 Internal clock sources and timing characteristics 54

9.3.5 Memory characteristics 57

9.3.6 I/O port pin characteristics 58

9.3.7 SPI serial peripheral interface 66

9.3.8 I2C interface characteristics 69

9.3.9 10-bit ADC characteristics 71

9.3.10 EMC characteristics 74

10 Package information 77

10.1 SO8N package information 77

10.2 Thermal characteristics 79

10.2.1 Reference document 80

10.2.2 Selecting the product temperature range 80

11 Ordering information 81

12 STM8 development tools 82

12.1 Emulation and in-circuit debugging tools 82

12.2 Software tools 83
12.2.1 STM8 toolset 83
12.2.2 C and assembly toolchains 83
12.3 Programming tools 83
13 Revision history 84

List of tables

Table 1. STM8S001J3 features 9
Table 2. Peripheral clock gating bit assignments in CLK_PCKENR1/2 registers. 15
Table 3. TIM timer features. 18
Table 4. Legend/abbreviations for STM8S001J3 pin description tables. 21
Table 5. STM8S001J3 pin description 22
Table 6. Flash, Data EEPROM and RAM boundary addresses 26
Table 7. I/O port hardware register map. 26
Table 8. General hardware register map 28
Table 9. CPU/SWIM/debug module/interrupt controller registers. 34
Table 10. Interrupt mapping 36
Table 11. Option bytes 37
Table 12. Option byte description. 37
Table 13. STM8S001J3 alternate function remapping bits for 8-pin devices 39
Table 14. Voltage characteristics 41
Table 15. Current characteristics 42
Table 16. Thermal characteristics. 42
Table 17. General operating conditions 43
Table 18. Operating conditions at power-up/power-down 44
Table 19. Total current consumption with code execution in run mode at V_{DD} = 5 V 45
Table 20. Total current consumption with code execution in run mode at V_{DD} = 3.3 V 46
Table 21. Total current consumption in wait mode at V_{DD} = 5 V 47
Table 22. Total current consumption in wait mode at V_{DD} = 3.3 V 47
Table 23. Total current consumption in active halt mode at V_{DD} = 5 V 48
Table 24. Total current consumption in active halt mode at V_{DD} = 3.3 V 48
Table 25. Total current consumption in halt mode at V_{DD} = 5 V 49
Table 26. Total current consumption in halt mode at V_{DD} = 3.3 V 49
Table 27. Wakeup times. 49
Table 28. Total current consumption and timing in forced reset state 50
Table 29. Peripheral current consumption 50
Table 30. HSE user external clock characteristics 54
Table 31. HSI oscillator characteristics. 55
Table 32. LSI oscillator characteristics 55
Table 33. RAM and hardware registers 57
Table 34. Flash program memory and data EEPROM 57
Table 35. I/O static characteristics 58
Table 36. Output driving current (standard ports). 60
Table 37. Output driving current (true open drain ports). 60
Table 38. Output driving current (high sink ports). 61
Table 39. SPI characteristics 66
Table 40. I2C characteristics 69
Table 41. ADC characteristics 71
Table 42. ADC accuracy with R_{AIN} < 10 kΩ , V_{DD} = 5 V. 72
Table 43. ADC accuracy with R_{AIN} < 10 kΩ R_{AIN}, V_{DD} = 3.3 V 72
Table 44. EMS data 74
Table 45. EMI data 75
Table 46. ESD absolute maximum ratings 75
Table 47. Electrical sensitivities 76
Table 48. SO8N – 8-lead 4.9 x 6 mm, plastic small outline, 150 mils body width,

package mechanical data 77
Table 49. Thermal characteristics 80
Table 50. Document revision history 84

List of figures

Figure 1. STM8S001J3 block diagram 10
Figure 2. Flash memory organization 13
Figure 3. STM8S001J3 SO8N pinout 21
Figure 4. Memory map 25
Figure 5. Pin loading conditions 40
Figure 6. Pin input voltage 41
Figure 7. f_{CPUmax} versus V_{DD} 43
Figure 8. External capacitor C_{EXT} 45
Figure 9. Typ. $I_{DD(RUN)}$ vs V_{DD}, HSE user external clock, f_{CPU} = 16 MHz 51
Figure 10. Typ. $I_{DD(RUN)}$ vs f_{CPU}, HSE user external clock, V_{DD} = 5 V 51
Figure 11. Typ. $I_{DD(RUN)}$ vs V_{DD}, HSI RC osc, f_{CPU} = 16 MHz 52
Figure 12. Typ. $I_{DD(WFI)}$ vs. V_{DD} HSE user external clock, f_{CPU} = 16 MHz 52
Figure 13. Typ. $I_{DD(WFI)}$ vs. f_{CPU}, HSE user external clock, V_{DD} = 5 V 53
Figure 14. Typ. $I_{DD(WFI)}$ vs V_{DD}, HSI RC osc, f_{CPU} = 16 MHz 53
Figure 15. HSE external clock source 54
Figure 16. Typical HSI frequency variation vs V_{DD} at 4 temperatures 55
Figure 17. Typical LSI frequency variation vs V_{DD} @ 4 temperatures 56
Figure 18. Typical V_{IL} and V_{IH} vs V_{DD} @ 4 temperatures 59
Figure 19. Typical pull-up resistance vs V_{DD} @ 4 temperatures 59
Figure 20. Typical pull-up current vs V_{DD} @ 4 temperatures 60
Figure 21. Typ. V_{OL} @ V_{DD} = 5 V (standard ports) 61
Figure 22. Typ. V_{OL} @ V_{DD} = 3.3 V (standard ports) 62
Figure 23. Typ. V_{OL} @ V_{DD} = 5 V (true open drain ports) 62
Figure 24. Typ. V_{OL} @ V_{DD} = 3.3 V (true open drain ports) 63
Figure 25. Typ. V_{OL} @ V_{DD} = 5 V (high sink ports) 63
Figure 26. Typ. V_{OL} @ V_{DD} = 3.3 V (high sink ports) 64
Figure 27. Typ. V_{DD} - V_{OH} @ V_{DD} = 5 V (standard ports) 64
Figure 28. Typ. V_{DD} - V_{OH} @ V_{DD} = 3.3 V (standard ports) 65
Figure 29. Typ. V_{DD} - V_{OH} @ V_{DD} = 5 V (high sink ports) 65
Figure 30. Typ. V_{DD} - V_{OH} @ V_{DD} = 3.3 V (high sink ports) 66
Figure 31. SPI timing diagram - slave mode and CPHA = 0 67
Figure 32. SPI timing diagram - slave mode and CPHA = 1[1] 68
Figure 33. SPI timing diagram - master mode[1] 68
Figure 34. Typical application with I2C bus and timing diagram 70
Figure 35. ADC accuracy characteristics 73
Figure 36. Typical application with ADC 73
Figure 37. SO8N – 8-lead, 4.9 x 6 mm, plastic small outline, 150 mils body width, package outline . 77
Figure 38. SO8N – 8-lead 4.9 x 6 mm, plastic small outline, 150 mils body width, package recommended footprint 78
Figure 39. SO8N – 8-lead 4.9 x 6 mm, plastic small outline, 150 mils body width, marking example 79
Figure 40. STM8S001J3 ordering information scheme[1] 81

1 Introduction

This datasheet contains the description of the STM8S001J3 features, pinout, electrical characteristics, mechanical data and ordering information.

- For complete information on the STM8S microcontroller memory, registers and peripherals, please refer to the STM8S and STM8A microcontroller families reference manual (RM0016).
- For information on programming, erasing and protection of the internal Flash memory please refer to the PM0051 (How to program STM8S and STM8A Flash program memory and data EEPROM).
- For information on the debug and SWIM (single wire interface module) refer to the STM8 SWIM communication protocol and debug module user manual (UM0470).
- For information on the STM8 core, please refer to the STM8 CPU programming manual (PM0044).

2 Description

The STM8S001J3 8-bit microcontrollers offer 8 Kbytes of Flash program memory, plus integrated true data EEPROM. It is referred to as low-density device in the STM8S microcontroller family reference manual (RM0016).

The STM8S001J3 device provides the following benefits: performance, robustness and reduced system cost.

Device performance and robustness are ensured by true data EEPROM supporting up to 100000 write/erase cycles, advanced core and peripherals made in a state-of-the-art technology at 16 MHz clock frequency, robust I/Os, independent watchdogs with separate clock source, and a clock security system.

The system cost is reduced thanks to a high system integration level with internal clock oscillators, watchdog, and brown-out reset.

Full documentation is offered as well as a wide choice of development tools.

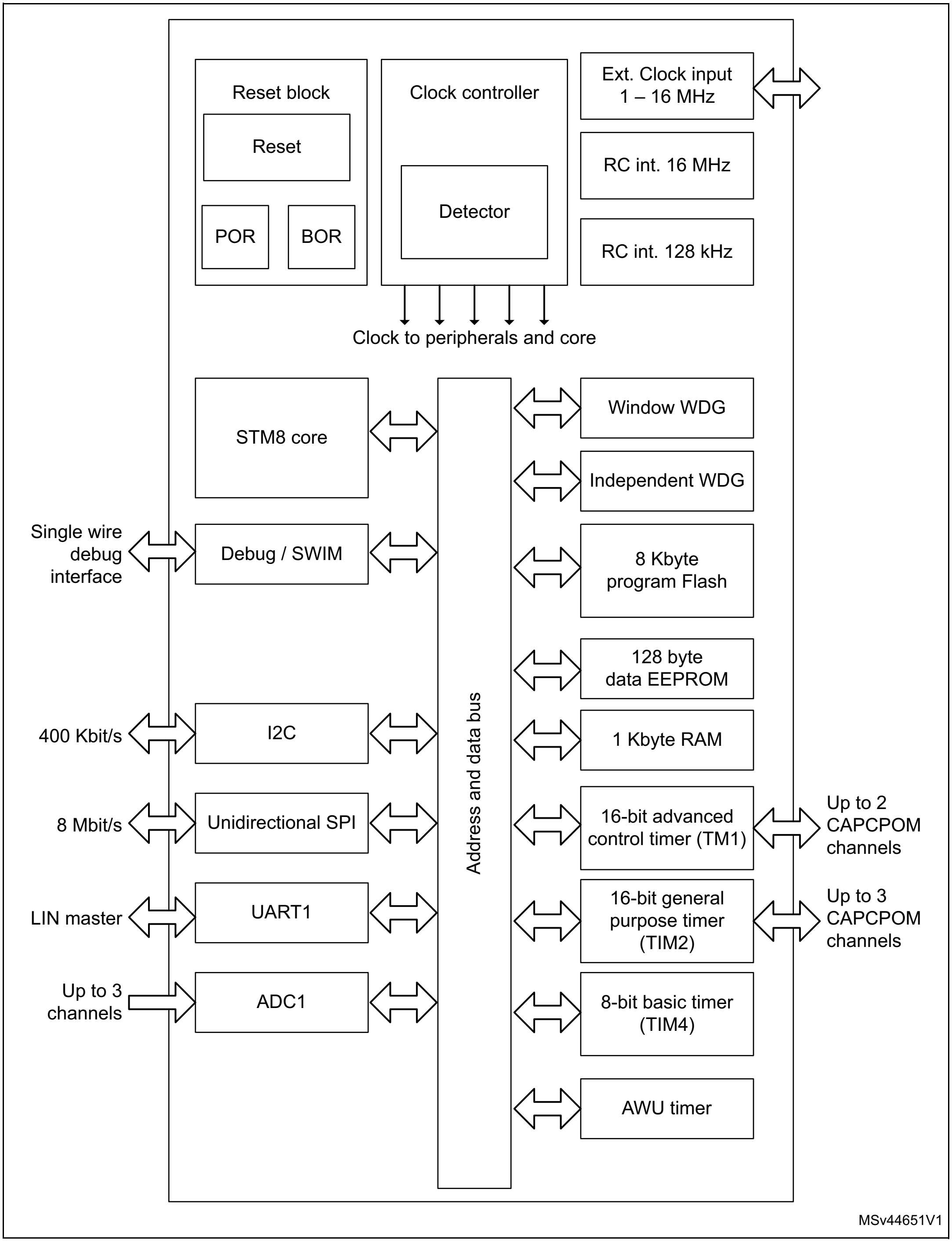
Table 1. STM8S001J3 features

Features	STM8S001J3
Pin count	8
Max. number of GPIOs (I/O)	5
External interrupt pins	5
Timer CAPCOM channels	3
Timer complementary outputs	1
A/D converter channels	3
High-sink I/Os	4
Low-density Flash program memory (byte)	8 K
RAM (byte)	1 K
True data EEPROM (byte)	128[1]
Peripheral set	Multi purpose timer (TIM1), SPI unidirectional, I2C, UART, Window WDG, independent WDG, ADC, PWM timer (TIM2), 8-bit timer (TIM4)

1. Without read-while-write capability.

3 Block diagram

Figure 1. STM8S001J3 block diagram



4 Functional overview

The following section intends to give an overview of the basic features of the STM8S001J3 functional modules and peripherals.

For more detailed information please refer to the corresponding family reference manual (RM0016).

4.1 Central processing unit STM8

The 8-bit STM8 core is designed for code efficiency and performance.

It contains six internal registers which are directly addressable in each execution context, 20 addressing modes including indexed indirect and relative addressing and 80 instructions.

Architecture and registers

- Harvard architecture
- 3-stage pipeline
- 32-bit wide program memory bus - single cycle fetching for most instructions
- X and Y 16-bit index registers - enabling indexed addressing modes with or without offset and read-modify-write type data manipulations
- 8-bit accumulator
- 24-bit program counter - 16-Mbyte linear memory space
- 16-bit stack pointer - access to a 64 K-level stack
- 8-bit condition code register - 7 condition flags for the result of the last instruction

Addressing

- 20 addressing modes
- Indexed indirect addressing mode for look-up tables located anywhere in the address space
- Stack pointer relative addressing mode for local variables and parameter passing

Instruction set

- 80 instructions with 2-byte average instruction size
- Standard data movement and logic/arithmetic functions
- 8-bit by 8-bit multiplication
- 16-bit by 8-bit and 16-bit by 16-bit division
- Bit manipulation
- Data transfer between stack and accumulator (push/pop) with direct stack access
- Data transfer using the X and Y registers or direct memory-to-memory transfers

4.2 Single wire interface module (SWIM) and debug module (DM)

The single wire interface module and debug module permits non-intrusive, real-time in-circuit debugging and fast memory programming.

SWIM

Single wire interface module for direct access to the debug module and memory programming. The interface can be activated in all device operation modes. The maximum data transmission speed is 145 byte/ms.

Debug module

The non-intrusive debugging module features a performance close to a full-featured emulator. Beside memory and peripherals, also CPU operation can be monitored in real-time by means of shadow registers.

- R/W to RAM and peripheral registers in real-time
- R/W access to all resources by stalling the CPU
- Breakpoints on all program-memory instructions (software breakpoints)
- Two advanced breakpoints, 23 predefined configurations

Recommendation for SWIM pin (pin #8) sharing

As the NRST pin is not available on this device, if the SWIM pin should be used with the I/O pin functionality, it is recommended to add a ~5 seconds delay in the firmware before changing the functionality on the pin with SWIM functions. This action allows the user to set the device into SWIM mode after the device power on and to be able to reprogram the device. If the pin with SWIM functionality is set to I/O mode immediately after the device reset, the device is unable to connect through the SWIM interface and it gets locked forever. This initial delay can be removed in the final (locked) code.

If the initial delay is not acceptable for the application there is the option that the firmware reenables the SWIM pin functionality under specific conditions such as during firmware startup or during application run. Once that this procedure is done, the SWIM interface can be used for device debug/programming.

4.3 Interrupt controller

- Nested interrupts with three software priority levels
- 32 interrupt vectors with hardware priority
- Up to 5 external interrupts including TLI
- Trap and reset interrupts

4.4 Flash program memory and data EEPROM

- 8 Kbytes of Flash program single voltage Flash memory
- 128 byte true data EEPROM
- User option byte area

Write protection (WP)

Write protection of Flash program memory and data EEPROM is provided to avoid unintentional overwriting of memory that could result from a user software malfunction.

There are two levels of write protection. The first level is known as MASS (memory access security system). MASS is always enabled and protects the main Flash program memory, data EEPROM and option bytes.

To perform in-application programming (IAP), this write protection can be removed by writing a MASS key sequence in a control register. This allows the application to modify the content of main program memory and data EEPROM, or to reprogram the device option bytes.

A second level of write protection, can be enabled to further protect a specific area of memory known as UBC (user boot code). Refer to *Figure 2*.

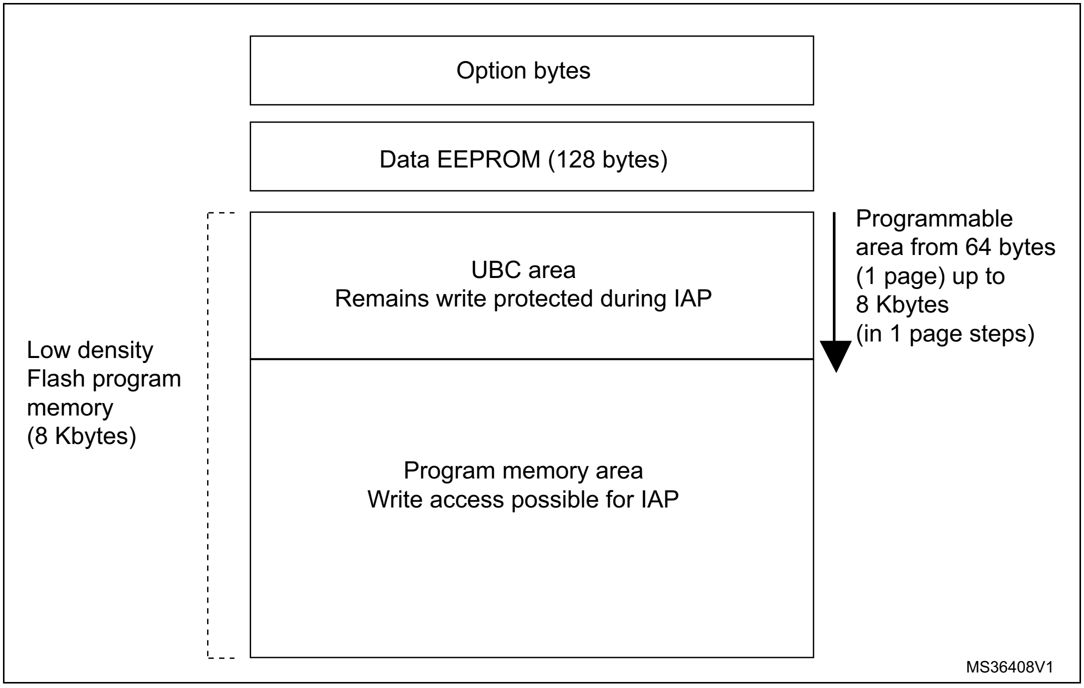
The size of the UBC is programmable through the UBC option byte (*Table 12*), in increments of 1 page (64-byte block) by programming the UBC option byte in ICP mode.

This divides the program memory into two areas:

- Main program memory: 8 Kbyte minus UBC
- User-specific boot code (UBC): Configurable up to 8 Kbyte

The UBC area remains write-protected during in-application programming. This means that the MASS keys do not unlock the UBC area. It protects the memory used to store the boot program, specific code libraries, reset and interrupt vectors, the reset routine and usually the IAP and communication routines.

Figure 2. Flash memory organization


Option bytes
Data EEPROM (128 bytes)
UBC area
Remains write protected during IAP
Programmable area from 64 bytes (1 page) up to 8 Kbytes (in 1 page steps)
Low density Flash program memory (8 Kbytes)
Program memory area
Write access possible for IAP
MS36408V1

Read-out protection (ROP)

The read-out protection blocks reading and writing the Flash program memory and data EEPROM memory in ICP mode (and debug mode). Once the read-out protection is activated, any attempt to toggle its status triggers a global erase of the program memory. Even if no protection can be considered as totally unbreakable, the feature provides a very high level of protection for a general purpose microcontroller.

Recommendation for the device's programming:

The device's 8 Kbytes program memory is not empty on virgin devices; there is code loop implemented on the reset vector. It is recommended to keep valid code loop in the device to avoid the program execution from an invalid memory address (which would be any memory address out of 8 Kbytes program memory space).

If the device's program memory is empty (0x00 content), it displays the behavior described below:

- After the power on, the “empty” code is executed (0x0000 opcodes = instructions: NEG (0x00, SP)) until the device reaches the end of the 8 Kbytes program memory (the end address = 0x9FFF).
  It takes around 4 milliseconds to reach the end of the 8 Kbytes memory space @2 MHz HSI clock.
- Once the device reaches the end of the 8 Kbytes program memory, the program continues and code from a non-existing memory is fetched and executed.
  The reading of non-existing memory is a random content which can lead to the execution of invalid instructions.
  The execution of invalid instructions generates a software reset and the program starts again. A reset can be generated every 4 milliseconds or more.

Only the “connect on-the-fly” method can be used to program the device through the SWIM interface. The “connect under-reset” method cannot be used because the NRST pin is not available on this device.

The “connect on-the-fly” mode can be used while the device is executing code, but if there is a device reset (by software reset) during the SWIM connection, this connection is aborted and it must be performed again from the debug tool. Note that the software reset occurrence can be of every 4 milliseconds, making it difficult to successfully connect to the device's debug tool (there is practically only one successful connection trial for every 10 attempts). Once that a successful connection is reached, the device can be programmed with a valid firmware without problems; therefore it is recommended that device is never erased and that is contains always a valid code loop.

4.5 Clock controller

The clock controller distributes the system clock (f_{MASTER}) coming from different oscillators to the core and the peripherals. It also manages clock gating for low power modes and ensures clock robustness.

Features

- ***Clock prescaler***: To get the best compromise between speed and current consumption the clock frequency to the CPU and peripherals can be adjusted by a programmable prescaler.
- ***Safe clock switching***: Clock sources can be changed safely on the fly in run mode through a configuration register. The clock signal is not switched until the new clock source is ready. The design guarantees glitch-free switching.
- ***Clock management***: To reduce power consumption, the clock controller can stop the clock to the core, individual peripherals or memory.
- ***Master clock sources***: Three different clock sources can be used to drive the master clock:
  - Up to 16 MHz high-speed user-external clock (HSE user-ext)
  - 16 MHz high-speed internal RC oscillator (HSI)
  - 128 kHz low-speed internal RC (LSI)
- ***Startup clock***: After reset, the microcontroller restarts by default with an internal 2 MHz clock (HSI/8). The prescaler ratio and clock source can be changed by the application program as soon as the code execution starts.
- ***Clock security system (CSS)***: This feature can be enabled by software. If an HSE clock failure occurs, the internal RC (16 MHz/8) is automatically selected by the CSS and an interrupt can optionally be generated.
- ***Configurable main clock output (CCO)***: This outputs an external clock for use by the application.

Table 2. Peripheral clock gating bit assignments in CLK_PCKENR1/2 registers

Bit	Peripheral clock	Bit	Peripheral clock	Bit	Peripheral clock	Bit	Peripheral clock
PCKEN17	TIM1	PCKEN13	UART1	PCKEN27	Reserved	PCKEN23	ADC
PCKEN16	Reserved	PCKEN12	Reserved	PCKEN26	Reserved	PCKEN22	AWU
PCKEN15	TIM2	PCKEN11	SPI	PCKEN25	Reserved	PCKEN21	Reserved
PCKEN14	TIM4	PCKEN10	I2C	PCKEN24	Reserved	PCKEN20	Reserved

4.6 Power management

For efficient power management, the application can be put in one of four different low-power modes. You can configure each mode to obtain the best compromise between the lowest power consumption, the fastest start-up time and available wakeup sources.

- ***Wait mode***: In this mode, the CPU is stopped, but peripherals are kept running. The wakeup is performed by an internal or external interrupt or reset.
- ***Active halt mode with regulator on***: In this mode, the CPU and peripheral clocks are stopped. An internal wakeup is generated at programmable intervals by the auto wake up unit (AWU). The main voltage regulator is kept powered on, so current consumption is higher than in active halt mode with regulator off, but the wakeup time is faster. Wakeup is triggered by the internal AWU interrupt, external interrupt or reset.
- ***Active halt mode with regulator off:*** This mode is the same as active halt with regulator on, except that the main voltage regulator is powered off, so the wake up time is slower.
- ***Halt mode***: In this mode the microcontroller uses the least power. The CPU and peripheral clocks are stopped, the main voltage regulator is powered off. Wakeup is triggered by external event or reset.

4.7 Watchdog timers

The watchdog system is based on two independent timers providing maximum security to the applications.

Activation of the watchdog timers is controlled by option bytes or by software. Once activated, the watchdogs cannot be disabled by the user program without performing a reset.

Window watchdog timer

The window watchdog is used to detect the occurrence of a software fault, usually generated by external interferences or by unexpected logical conditions, which cause the application program to abandon its normal sequence.

The window function can be used to trim the watchdog behavior to match the application perfectly.

The application software must refresh the counter before time-out and during a limited time window.

A reset is generated in two situations:

1. Timeout: at 16 MHz CPU clock the time-out period can be adjusted between 75 µs up to 64 ms.
2. Refresh out of window: the down-counter is refreshed before its value is lower than the one stored in the window register.

Independent watchdog timer

The independent watchdog peripheral can be used to resolve processor malfunctions due to hardware or software failures.

It is clocked by the 128 kHz LSI internal RC clock source, and thus stays active even in case of a CPU clock failure

The IWDG time base spans from 60 µs to 1 s.

4.8 Auto wakeup counter

- Used for auto wakeup from active halt mode
- Clock source: internal 128 kHz internal low frequency RC oscillator or external clock
- LSI clock can be internally connected to TIM1 input capture channel 1 for calibration

4.9 TIM1 - 16-bit advanced control timer

This is a high-end timer designed for a wide range of control applications. With its complementary outputs, dead-time control and center-aligned PWM capability, the field of applications is extended to lighting and half-bridge driver.

- 16-bit up, down and up/down autoreload counter with 16-bit prescaler
- Four independent capture/compare channels (CAPCOM) configurable as input capture, output compare, PWM generation (edge and center aligned mode) and single pulse mode output
- Synchronization module to control the timer with external signals
- Break input to force the timer outputs into a defined state
- One complementary output (CH1 with CH1N option) with adjustable dead time
- Interrupt sources: 3 x input capture/output compare, 1 x overflow/update, 1 x break

4.10 TIM2 - 16-bit general purpose timer

- 16-bit autoreload (AR) up-counter
- 15-bit prescaler adjustable to fixed power of 2 ratios 1…32768
- 3 individually configurable capture/compare channels
- PWM mode
- Interrupt sources: 3 x input capture/output compare, 1 x overflow/update

4.11 TIM4 - 8-bit basic timer

- 8-bit autoreload, adjustable prescaler ratio to any power of 2 from 1 to 128
- Clock source: CPU clock
- Interrupt source: 1 x overflow/update

Table 3. TIM timer features

Timer	Counter size (bits)	Prescaler	Counting mode	CAPCOM channels	Complem. outputs	Ext. trigger	Timer synchr-onization/ chaining
TIM1	16	Any integer from 1 to 65536	Up/down	2	1[1]	No	No
TIM2	16	Any power of 2 from 1 to 32768	Up	3	0	No	
TIM4	8	Any power of 2 from 1 to 128	Up	0	0	No	

[1] TIM1_CH2N with TIM1_CH1

4.12 Analog-to-digital converter (ADC1)

STM8S001J3 contains a 10-bit successive approximation A/D converter (ADC1) with up to three external and one internal multiplexed input channels and the following main features:

- Input voltage range: 0 to V_{DDA}
- Conversion time: 14 clock cycles
- Single and continuous, buffered continuous conversion modes
- Buffer size (10 x 10 bits)
- Scan mode for single and continuous conversion of a sequence of channels
- Analog watchdog capability with programmable upper and lower thresholds
- Analog watchdog interrupt
- Internal reference voltage on channel AIN7
- External trigger input
- Trigger from TIM1 TRGO
- End of conversion (EOC) interrupt

next paragraph :

Internal bandgap reference voltage

Channel AIN7 is internally connected to the internal bandgap reference voltage. The internal bandgap reference is constant and can be used, for example, to monitor V_{DD} or to determine the absolute voltage on external input channels. It is independent of variations in V_{DD} and ambient temperature T_A.

4.13 Communication interfaces

The following communication interfaces are implemented:

- UART1: full feature UART, synchronous mode, SmartCard mode, IrDA mode, LIN2.1 master capability
- SPI: master mode transmit/receive only, slave mode receive only, 8 Mbit/s
- I²C: up to 400 Kbit/s

4.13.1 UART1

Main features

- 1 Mbit/s full duplex SCI
- High precision baud rate generator
- Smartcard reader emulation
- IrDA SIR encoder decoder
- LIN master mode
- Single wire half duplex mode

Asynchronous communication (UART mode)

- Full duplex communication - NRZ standard format (mark/space)
- Programmable transmit and receive baud rates up to 1 Mbit/s (f_{CPU}/16) and capable of following any standard baud rate regardless of the input frequency
- Separate enable bits for transmitter and receiver
- Two receiver wakeup modes:
  - Address bit (MSB)
  - Idle line (interrupt)
- Transmission error detection with interrupt generation
- Parity control

LIN master mode

- Emission: generates 13-bit synch. break frame
- Reception: detects 11-bit break frame

4.13.2 SPI

- Maximum speed: 8 Mbit/s (f_{MASTER}/2) both for master and slave
- Unidirectional transfer: SPI master mode transmit/receive only, SPI slave mode receive only
- Simplex master synchronous transfers on two lines with a possible bidirectional data line
- Master or slave operation - selectable by software
- CRC calculation
- 1 byte Tx and Rx buffer

4.13.3 I2C

- I2C master features
  - Clock generation
  - Start and stop generation
- I2C slave features
  - Programmable I2C address detection
  - Stop bit detection
- Generation and detection of 7-bit/10-bit addressing and general call
- Supports different communication speeds
  - Standard speed (up to 100 kHz)
  - Fast speed (up to 400 kHz)

5 Pinouts and pin descriptions

This section presents the pinouts and pin descriptions for STM8S001J3. *Table 4* introduces the legends and abbreviations that are used in the upcoming subsections.

Table 4. Legend/abbreviations for STM8S001J3 pin description tables

Type	I = input, O = output, S = power supply	
Level	Input	CM = CMOS
	Output	HS = high sink

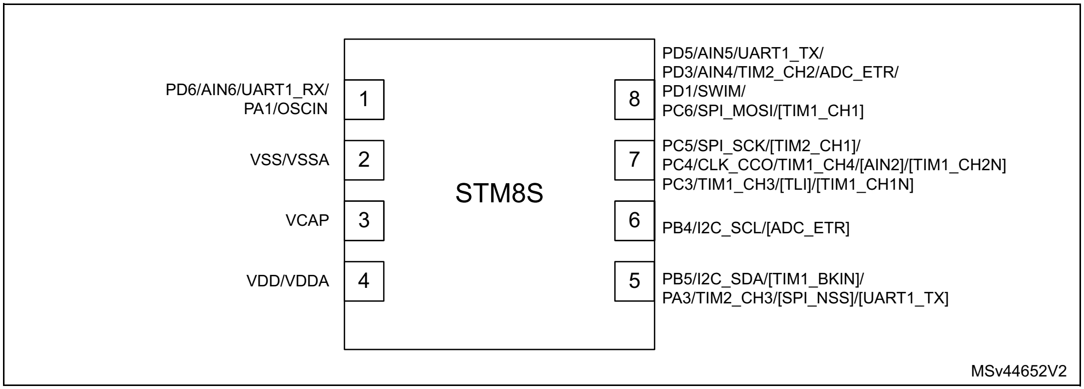
Output speed	O1 = slow (up to 2 MHz)
	O2 = fast (up to 10 MHz)
	O3 = fast/slow programmability with slow as default state after reset
	O4 = fast/slow programmability with fast as default state after reset	

Port and control configuration	Input	float = floating, wpu = weak pull-up
	Output	T = true open drain, OD = open drain, PP = push pull
Reset state	Bold **x** (pin state after internal reset release) Unless otherwise specified, the pin state is the same during the reset phase and after the internal reset release.	

5.1 STM8S001J3 SO8N pinout and pin description

Figure 3 presents the STM8S001J3 pinout image and *Table 5* below presents the device's pins description.

Figure 3. STM8S001J3 SO8N pinout



1. [ ] Alternative function option (if the same alternate function is shown twice, it indicated an exclusive choice and not a duplication of the function).

Table 5. STM8S001J3 pin description

Pin no.	Pin name	Type	Input			Output				Main function (after reset)	Default alternate function	Alternate function after remap [option bit]
SO8N			floating	wpu	Ext. interr.	High sink(1)	Speed	OD	PP			
1	PD6/ AIN6/ UART1 _RX(2)	I/O	**X**	X	X	HS	O3	X	X	Port D6	Analog input 6/ UART1 data receive	-
	PA1/ OSCIN(3)	I/O	**X**	X	X	-	O1	X	X	Port A1	External clock input (HSE clock)	-
2	VSS/VSSA	S	-	-	-	-	-	-	-	Ground		-
3	VCAP	S	-	-	-	-	-	-	-	1.8 V regulator capacitor		-
4	VDD/VDDA	S	-	-	-	-	-	-	-	Power supply		-
5	PA3/ TIM2_ CH3 [SPI_ NSS]\ [UART1_TX](2)	I/O	**X**	X	X	HS	O3	X	X	Port A3	Timer 2 channel 3	SPI master/ slave select [AFR1] UART1 data transmit [AFR1 and AFR0]
	PB5/ I2C_ SDA [TIM1_ BKIN]	I/O	**X**	-	X	-	O1	T(4)	-	Port B5	I2C data	Timer 1 - break input [AFR4]
6	PB4/ I2C_ SCL /[ADC_ETR]	I/O	**X**	-	X	-	O1	T(4)	-	Port B4	I2C clock	ADC external trigger [AFR4]
7	PC3/ TIM1_CH3 [TLI] [TIM1_ CH1N]	I/O	**X**	X	X	HS	O3	X	X	Port C3	Timer 1 - channel 3	Top level interrupt [AFR3] Timer 1 - inverted channel 1 [AFR7]
	PC4/ CLK_CCO/ TIM1_ CH4/[AIN2]/ [TIM1_ CH2N]	I/O	**X**	X	X	HS	O3	X	X	Port C4	Configurable clock output/Timer 1 - channel 4	Analog input 2 [AFR2], Timer 1 - inverted channel 2 [AFR7]
	PC5/ SPI_SCK [TIM2_ CH1]	I/O	**X**	X	X	HS	O3	X	X	Port C5	SPI clock	Timer 2 - channel 1 [AFR0]

Table 5. STM8S001J3 pin description (continued)

Pin no.	Pin name	Type	Input			Output				Main function (after reset)	Default alternate function	Alternate function after remap [option bit]
SO8N			floating	wpu	Ext. interr.	High sink[1]	Speed	OD	PP			
8	PC6/ SPI_MOSI [TIM1_ CH1]	I/O	**X**[5]	X	X	HS	O3	X	X	Port C6	SPI master out/slave in	Timer 1 - channel 1 [AFR0]
	PD1/ SWIM[5]	I/O	X	**X**[5]	X	HS	O4	X	X	Port D1	SWIM data interface	-
	PD3/ AIN4/ TIM2_ CH2/ ADC_ ETR	I/O	**X**[5]	X	X	HS	O3	X	X	Port D3	Analog input 4/ Timer 2 - channel 2/ADC external trigger	-
	PD5/ AIN5/ UART1 _TX	I/O	**X**[5]	X	X	HS	O3	X	X	Port D5	Analog input 5/ UART1 data transmit	-

1. I/O pins used simultaneously for high current source/sink must be uniformly spaced around the package. In addition, the total driven current must respect the absolute maximum ratings.
2. By remapping UART1_TX (AFR0=1 and AFR1=1) to PA3 the UART1_RX alternate function on PD6 becomes unavailable. UART1 can be then used only in Single wire half-duplex mode or in Smartcard-reader emulation mode.
3. When the MCU is in halt/active-halt mode, PA1 is automatically configured in input weak pull-up and cannot be used for waking up the device. In this mode, the output state of PA1 is not driven. It is recommended d to use PA1 only in input mode if halt/active-halt is used in the application.
4. In the open-drain output column, “T” defines a true open-drain I/O (P-buffer, weak pull-up, and protection diode to VDD are not implemented). Although PB5 itself is a true open drain GPIO with its respective internal circuitry and characteristics, V_{IN} maximum of the pin number 5 is limited by the standard GPIO PA3 which is also bonded to pin number 5.
5. The PD1 pin is in input pull-up during the reset phase and after internal reset release. This PD1 default state influences all GPIOs connected in parallel on pin# 8 (PC6, PD3, PD5).

Note: The PA2, PB0, PB1, PB2, PB3, PB6, PB7, PC1, PC2, PC7, PD0, PD2, PD4, PD7, PE5 and PF4 GPIOs should be configured after device reset in output push-pull mode with output low-state to reduce the device’s consumption and to improve its EMC immunity. The GPIOs mentioned above are not connected to pins, and they are in input-floating mode after a device reset.

Note: As several pins provide a connection to multiple GPIOs, the mode selection for any of those GPIOs impacts all the other GPIOs connected to the same pin. The user is responsible for the proper setting of the GPIO modes in order to avoid conflicts between GPIOs bonded to the same pin (including their alternate functions). For example, pull-up enabled on PD1 is also seen on PC6, PD3 and PD5. Push-pull configuration of PC3 is also seen on PC4 and PC5, etc.

5.2 Alternate function remapping

As shown in the rightmost column of the pin description table, some alternate functions can be remapped at different I/O ports by programming one of eight AFR (alternate function remap) option bits. Refer to *Section 8: Option bytes*. When the remapping option is active, the default alternate function is no longer available.

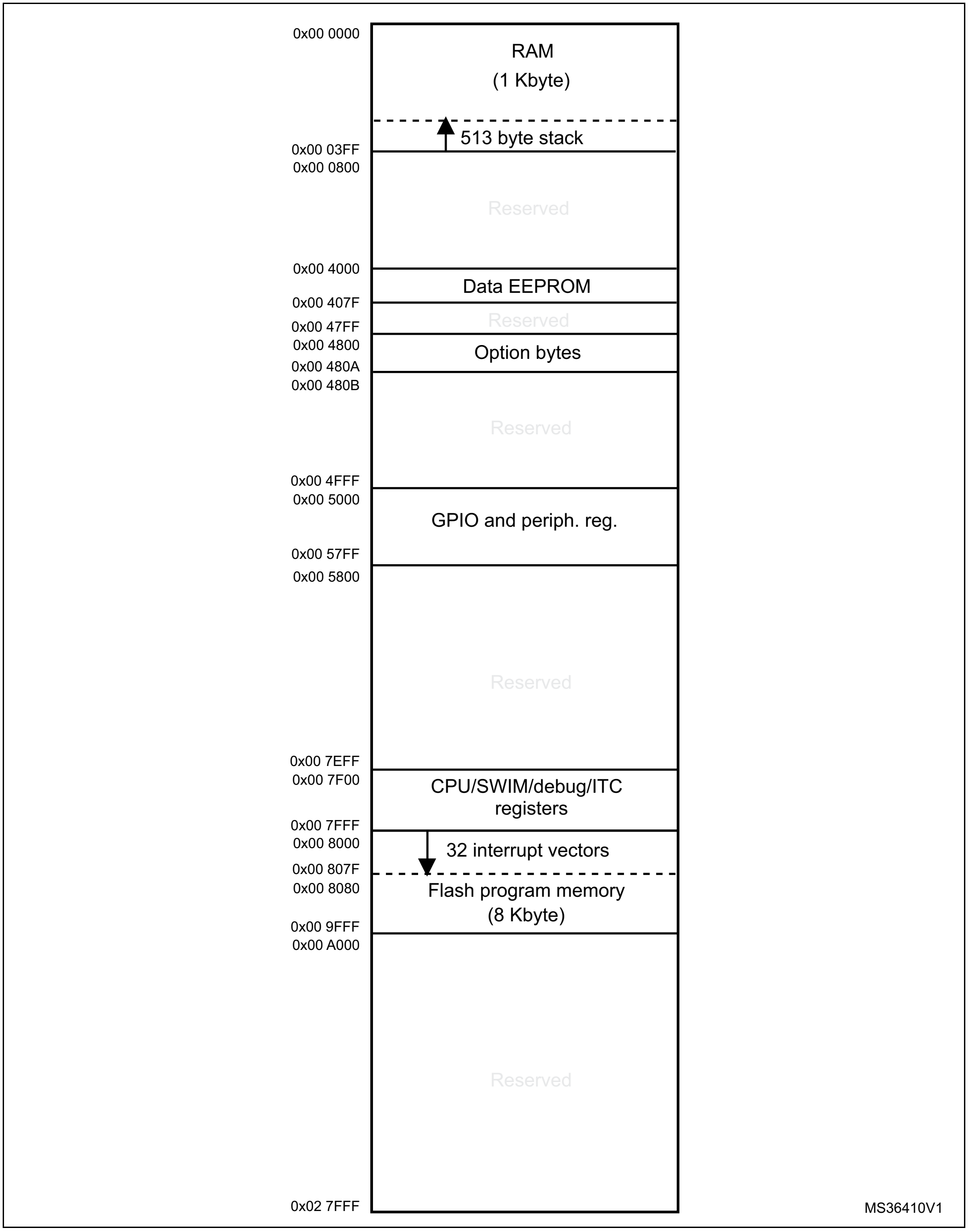
To use an alternate function, the corresponding peripheral must be enabled in the peripheral registers.

Alternate function remapping does not effect GPIO capabilities of the I/O ports (see the GPIO section of the family reference manual, RM0016).

6 Memory and register map

6.1 Memory map

Figure 4. Memory map


0x00 0000
RAM
(1 Kbyte)
513 byte stack
0x00 03FF
0x00 0800
Reserved
0x00 4000
Data EEPROM
0x00 407F
0x00 47FF
Reserved
0x00 4800
Option bytes
0x00 480A
0x00 480B
Reserved
0x00 4FFF
0x00 5000
GPIO and periph. reg.
0x00 57FF
0x00 5800
Reserved
0x00 7EFF
0x00 7F00
CPU/SWIM/debug/ITC registers
0x00 7FFF
0x00 8000
32 interrupt vectors
0x00 807F
0x00 8080
Flash program memory
(8 Kbyte)
0x00 9FFF
0x00 A000
Reserved
0x02 7FFF
MS36410V1