Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

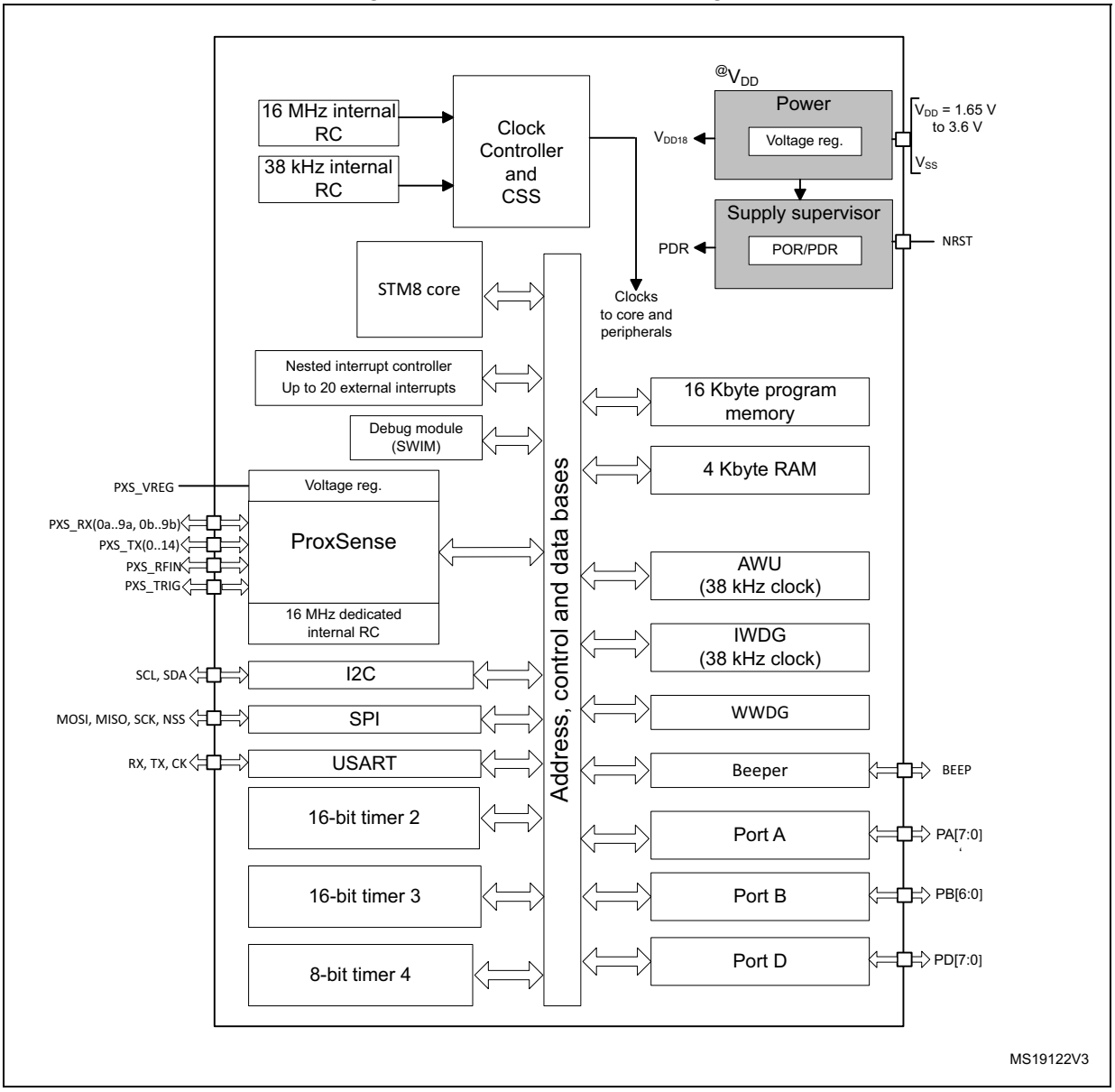
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	STM8
Core Size	8-Bit
Speed	16MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	POR, ProxSense, PWM, WDT
Number of I/O	23
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	1.65V ~ 3.6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-UFQFN Exposed Pad
Supplier Device Package	48-UFQFPN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/stm8tl53c4u6

Contents

3 Product overview

Figure 1. STM8TL5xx4 block diagram



Legend:

AWU: Auto-wakeup unit
Int. RC: internal RC oscillator
I²C: Inter-integrated circuit multimaster interface
POR/PDR: Power on reset / power down reset
SPI: Serial peripheral interface
SWIM: Single wire interface module
USART: Universal synchronous / asynchronous receiver / transmitter
IWDG: Independent watchdog
WWDG: Window watchdog
ProxSense™: capacitive sensing peripheral

3.6 Low power modes

To minimize power consumption, the product features three MCU low power modes:

- Wait mode: CPU clock stopped, selected peripherals at full clock speed.
- Active-halt mode:
  - When wakeup time is programmed in the AWU unit, the CPU and peripheral clocks are stopped. The RAM content is preserved.
  - When a ProxSense acquisition is ongoing, the wakeup is on ProxSense interrupts; the CPU and the other peripheral clocks are stopped.
- Halt mode: CPU and peripheral clocks are stopped, the device remains powered on. Wakeup is triggered by an external interrupt.

The ProxSense peripheral can return to low power mode between each conversion.
The ProxSense acquisition can be operated in Run, Wait and Active-halt modes.

3.7 Voltage regulators

The STM8TL5xx4 devices embed an internal voltage regulator for generating the 1.8 V power supply for the core and peripherals and a second internal voltage regulator providing a stable power supply (around 1.45V) for the ProxSense peripheral.

3.7.1 Dual-mode voltage regulator

This regulator has two different modes: main voltage regulator mode (MVR) and low power voltage regulator mode (LPVR). When in Active-halt mode, the regulator remains in MVR if ProxSense is active. When entering Halt or Active-halt modes, the system automatically switches from the MVR to the LPVR in order to reduce current consumption unless ProxSense is enabled.

3.7.2 ProxSense voltage regulator

This regulator provides a very stable voltage to power the ProxSense peripheral including ProxSense pins in order to be independent of any power supply variations. This regulator is switched on while the ProxSense peripheral is enabled (bit PXSEN = 1) and bit LOW_POWER is set to ‘0’ in register PXS_CR1. Otherwise, when LOW_POWER is set to ‘1’, this regulator is only enabled during conversions (while CIPF = 1 and SYNCPF = 0).

3.8 Clock control

The STM8TL5xx4 embeds a robust clock controller. It is used to distribute the system clock (SYSCLK) to the core and the peripherals and to manage clock gating for low power modes. This system clock is a 16-MHz High Speed Internal RC oscillator (HSI RC), followed by a programmable prescaler.

In addition, a 38 kHz low speed internal RC oscillator is used by the Independent watchdog (IWDG) and Auto-wakeup unit (AWU).

Table 4. STM8TL5xx4 pin description (continued)

Pin no.			Pin name	Type	Input				Output			Main function (after reset)	Alternate function	
UFQFPN48	UFQFPN28	TSSOP20			Level	floating	wpu	Ext. interrupt	High sink/source	OD	PP		Default	Remap
23	15	-	PXS_RX5a	-	-	-	-	-	-	-	-	PXS_RX5a	ProxSense receiver 5a	-
24	-	-	PXS_RX5b	-	-	-	-	-	-	-	-	PXS_RX5b	ProxSense receiver 5b	-
25	16	14	PXS_RX6a	-	-	-	-	-	-	-	-	PXS_RX6a	ProxSense receiver 6a	-
26	-	-	PXS_RX6b	-	-	-	-	-	-	-	-	PXS_RX6b	ProxSense receiver 6b	-
27	17	15	PXS_RX7a	-	-	-	-	-	-	-	-	PXS_RX7a	ProxSense receiver 7a	-
28	-	-	PXS_RX7b	-	-	-	-	-	-	-	-	PXS_RX7b	ProxSense receiver 7b	-
29	-	-	PXS_RX8a	-	-	-	-	-	-	-	-	PXS_RX8a	ProxSense receiver 8a	-
30	-	-	PXS_RX8b	-	-	-	-	-	-	-	-	PXS_RX8b	ProxSense receiver 8b	-
31	-	-	PXS_RX9a	-	-	-	-	-	-	-	-	PXS_RX9a	ProxSense receiver 9a	-
32	-	-	PXS_RX9b	-	-	-	-	-	-	-	-	PXS_RX9b	ProxSense receiver 9b	-
33	18	16	PD0/PXS_TX0	I/O	TC	X	X	X	HS	X	X	Port D0	ProxSense transmitter 0	-
34	19	17	PD1/PXS_TX1	I/O	TC	X	X	X	HS	X	X	Port D1	ProxSense transmitter 1	-
35	20	-	PD2/PXS_TX2(4)	I/O	TC	X	X	X	HS	X	X	Port D2	ProxSense transmitter 2(4)	-
36	21	-	PD3/PXS_TX3(4)	I/O	TC	X	X	X	HS	X	X	Port D3	ProxSense transmitter 3(4)	-
37	-	-	VSSIO	S	-	-	-	-	-	-	-	IOs ground	-	-
38	-	-	VDDIO	S	-	-	-	-	-	-	-	IOs power supply	-	-
39	22	18	PD4/PXS_TX4(4) / TIM2_CH1	I/O	TC	X	X	X	HS	X	X	Port D4	ProxSense transmitter 4(4)	-
													Timer 2 - channel 1	-

Table 7. General hardware register map (continued)

Address	Block	Register label	Register name	Reset status
0x00 50D3	WWDG	WWDG_CR	WWDG control register	0x7F
0x00 50D4		WWDG_WR	WWDG window register	
0x00 50D5 to 0x00 50D7	Reserved area (11 byte)			
0x00 50E0	IWDG	IWDG_KR	IWDG key register	0xXX
0x00 50E1		IWDG_PR	IWDG prescaler register	0x00
0x00 50E2		IWDG_RLR	IWDG reload register	0xFF
0x00 50E3 to 0x00 50EF	Reserved area (13 byte)			
0x00 50F0	AWU	AWU_CSR	AWU control/status register	0x00
0x00 50F1		AWU_APR	AWU asynchronous prescaler buffer register	0x3F
0x00 50F2		AWU_TBR	AWU timebase selection register	0x00
0x00 50F3	BEEP	BEEP_CSR	BEEP control/status register	0x1F
0x00 50F4 to 0x00 51FF	Reserved area (268 byte)			
0x00 5200	SPI	SPI_CR1	SPI control register 1	0x00
0x00 5201		SPI_CR2	SPI control register 2	0x00
0x00 5202		SPI_ICR	SPI interrupt control register	0x00
0x00 5203		SPI_SR	SPI status register	0x00
0x00 5204		SPI_DR	SPI data register	0x00
0x00 5205 to 0x00 520F	Reserved area (11 byte)			

Table 7. General hardware register map (continued)

Address	Block	Register label	Register name	Reset status
0x00 5250	TIM2	TIM2_CR1	TIM2 control register 1	0x00
0x00 5251		TIM2_CR2	TIM2 control register 2	0x00
0x00 5252		TIM2_SMCR	TIM2 slave mode control register	0x00
0x00 5253		TIM2_ETR	TIM2 external trigger register	0x00
0x00 5254		TIM2_IER	TIM2 interrupt enable register	0x00
0x00 5255		TIM2_SR1	TIM2 status register 1	0x00
0x00 5256		TIM2_SR2	TIM2 status register 2	0x00
0x00 5257		TIM2_EGR	TIM2 event generation register	0x00
0x00 5258		TIM2_CCMR1	TIM2 capture/compare mode register 1	0x00
0x00 5259		TIM2_CCMR2	TIM2 capture/compare mode register 2	0x00
0x00 525A		TIM2_CCER1	TIM2 capture/compare enable register 1	0x00
0x00 525B		TIM2_CNTRH	TIM2 counter register high	0x00
0x00 525C		TIM2_CNTRL	TIM2 counter register low	0x00
0x00 525D		TIM2_PSCR	TIM2 prescaler register	0x00
0x00 525E		TIM2_ARRH	TIM2 auto-reload register high	0xFF
0x00 525F		TIM2_ARRL	TIM2 auto-reload register low	0xFF
0x00 5260		TIM2_CCR1H	TIM2 capture/compare register 1 high	0x00
0x00 5261		TIM2_CCR1L	TIM2 capture/compare register 1 low	0x00
0x00 5262		TIM2_CCR2H	TIM2 capture/compare register 2 high	0x00
0x00 5263		TIM2_CCR2L	TIM2 capture/compare register 2 low	0x00
0x00 5264		TIM2_BKR	TIM2 break register	0x00
0x00 5265		TIM2_OISR	TIM2 output idle state register	0x00
0x00 5266 to 0x00 527F	Reserved area (26 byte)			

Table 7. General hardware register map (continued)

Address	Block	Register label	Register name	Reset status
0x00 5280	TIM3	TIM3_CR1	TIM3 control register 1	0x00
0x00 5281		TIM3_CR2	TIM3 control register 2	0x00
0x00 5282		TIM3_SMCR	TIM3 slave mode control register	0x00
0x00 5283		TIM3_ETR	TIM3 external trigger register	0x00
0x00 5284		TIM3_IER	TIM3 interrupt enable register	0x00
0x00 5285		TIM3_SR1	TIM3 status register 1	0x00
0x00 5286		TIM3_SR2	TIM3 status register 2	0x00
0x00 5287		TIM3_EGR	TIM3 event generation register	0x00
0x00 5288		TIM3_CCMR1	TIM3 capture/compare mode register 1	0x00
0x00 5289		TIM3_CCMR2	TIM3 capture/compare mode register 2	0x00
0x00 528A		TIM3_CCER1	TIM3 capture/compare enable register 1	0x00
0x00 528B		TIM3_CNTRH	TIM3 counter register high	0x00
0x00 528C		TIM3_CNTRL	TIM3 counter register low	0x00
0x00 528D		TIM3_PSCR	TIM3 prescaler register	0x00
0x00 528E		TIM3_ARRH	TIM3 auto-reload register high	0xFF
0x00 528F		TIM3_ARRL	TIM3 auto-reload register low	0xFF
0x00 5290		TIM3_CCR1H	TIM3 capture/compare register 1 high	0x00
0x00 5291		TIM3_CCR1L	TIM3 capture/compare register 1 low	0x00
0x00 5292		TIM3_CCR2H	TIM3 capture/compare register 2 high	0x00
0x00 5293		TIM3_CCR2L	TIM3 capture/compare register 2 low	0x00
0x00 5294		TIM3_BKR	TIM3 break register	0x00
0x00 5295		TIM3_OISR	TIM3 output idle state register	0x00
0x00 5296 to 0x00 52DF	Reserved area (74 byte)			
0x00 52E0	TIM4	TIM4_CR1	TIM4 control register 1	0x00
0x00 52E1		TIM4_CR2	TIM4 control register 2	0x00
0x00 52E2		TIM4_SMCR	TIM4 Slave mode control register	0x00
0x00 52E3		TIM4_IER	TIM4 interrupt enable register	0x00
0x00 52E4		TIM4_SR1	TIM4 Status register 1	0x00
0x00 52E5		TIM4_EGR	TIM4 event generation register	0x00
0x00 52E6		TIM4_CNTR	TIM4 counter register	0x00
0x00 52E7		TIM4_PSCR	TIM4 prescaler register	0x00
0x00 52E8		TIM4_ARR	TIM4 auto-reload register low	0xFF

Table 9. Interrupt mapping (continued)

IRQ No.	Source block	Description	Wakeup from Halt mode	Wakeup from Active-halt mode	Wakeup from Wait (WFI mode)	Wakeup from Wait (WFE mode)(1)	Vector address
22	TIM3	TIM3 capture/compare interrupt	-	-	Yes	Yes	0x00 8060
23-24	Reserved						0x00 8064-0x00 806B
25	TIM4	TIM4 update/overflow/ trigger interrupt	-	-	Yes	Yes	0x00 806C
26	SPI	SPI TX buffer empty/ RX buffer not empty/ error/wakeup interrupt	Yes	Yes	Yes	Yes	0x00 8070
27	USART	USART transmit data register empty/ transmission complete interrupt	-	-	Yes	Yes	0x00 8074
28	USART	USART received data ready/overrun error/ idle line detected/parity error/global error interrupt	-	-	Yes	Yes	0x00 8078
29	I2C	I2C interrupt(3)	Yes	Yes	Yes	Yes	0x00 807C

1. The Low power wait mode is entered when executing a WFE instruction in Low power run mode. In WFE mode, the interrupt is served if it has been previously enabled. After processing the interrupt, the processor goes back to WFE mode. When the interrupt is configured as a wakeup event, the CPU wakes up and resumes processing.
2. ProxSense activated before executing HALT instruction.
3. The device is woken up from Halt or Active-halt mode only when the address received matches the interface address.

7 Option byte

Option byte contain configurations for device hardware features as well as the memory protection of the device. They are stored in a dedicated row of the memory.

All option byte can be modified only in ICP mode (with SWIM) by accessing the EEPROM address. See *Table 10* for details on option byte addresses.

Refer to the STM8TL5xxx Flash programming manual (PM0212) and STM8 SWIM and debug manual (UM0470) for information on SWIM programming procedures.

Table 10. Option byte

Addr.	Option name	Option byte No.	Option bits								Factory default setting
			7	6	5	4	3	2	1	0	
0x4800	Read-out protection (ROP)	OPT0	ROP[7:0]								0xAA
0x4801	-	-	Must be programmed to 0x00								0x00
0x4802	User Boot code size (UBC)	OPT1	UBC[7:0]								0x00
0x4803	DATASIZE	OPT2	DATASIZE[7:0]								0x00
0x4807	PCODESIZE	OPT3	PCODESIZE[7:0]								0x00
0x4808	Window watchdog and independent window watchdog	OPT4 [3:0]	Reserved				WWDG_HALT	WWDG_HW	IWDG_HALT	IWDG_HW	0x00

Table 11. Option byte description

Option byte number	Description
OPT0	**ROP[7:0]** *Memory readout protection (ROP)* 0xAA: Readout protection disabled (write access via SWIM protocol) Refer to *Read-out protection* section in the STM8TL5xxx reference manual (RM0312) for details.
OPT1	**UBC[7:0]** *Size of the user boot code area* 0x00: no UBC 0x01-0x02: UBC contains only the interrupt vectors. 0x03: Page 0 and 1 reserved for the interrupt vectors. Page 2 is available to store user boot code. Memory is write protected ... 0xFF: Page 0 to 254 reserved for UBC, memory is write protected Refer to *User boot area (UBC)* section in the STM8TL5xxx reference manual (RM0312) for more details.

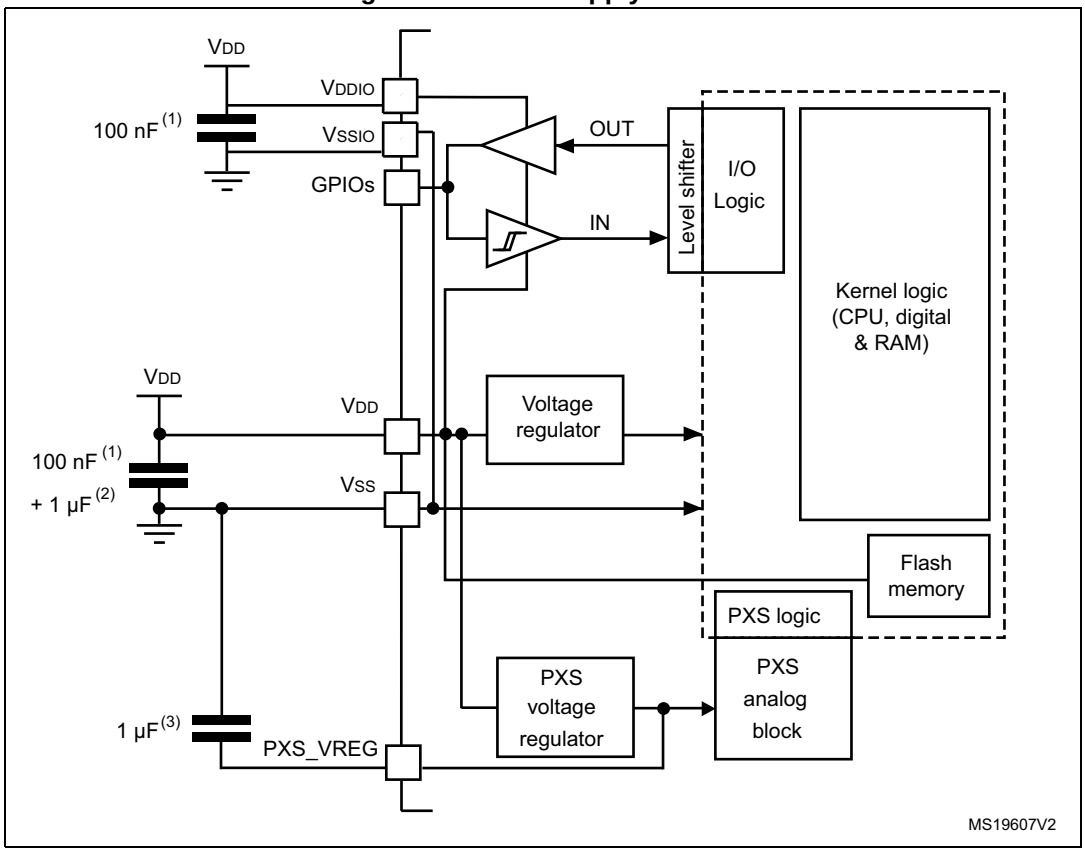
Table 11. Option byte description (continued)

Option byte number	Description
OPT2	**DATASIZE[7:0]** Size of the data EEPROM area 0x00: no data EEPROM area 0x01: 1 page reserved for data storage from 0xBFC0 to 0xBFFF 0x02: 2 pages reserved for data storage from 0xBF80 to 0xBFFF ... 0x20: 32 pages reserved for data storage from 0xB800 to 0xBFFF Refer to *Data EEPROM (DATA)* section in the STM8TL5xxx reference manual (RM0312) for more details.
OPT3	**PCODESIZE[7:0]** Size of the proprietary code area 0x00: No proprietary code area 0x03: TRAP vector and page 2 (0x8080 to 0x80BF) reserved for the proprietary code and read/write protected ... 0xFF: TRAP vector and page 2 to 254 (0x8080 to 0xBFBF) reserved for the proprietary code and read/write protected Refer to Proprietary code area (PCODE) section in the STM8TLxxxx Programming Manual(PM0212) for more details.
OPT4	**IWDG_HW:** *Independent watchdog* 0: Independent watchdog activated by software 1: Independent watchdog activated by hardware
	IWDG_HALT: *Independent window watchdog reset on Halt/Active-halt* 0: Independent watchdog continues running in Halt/Active-halt mode 1: Independent watchdog stopped in Halt/Active-halt mode
	WWDG_HW: *Window watchdog* 0: Window watchdog activated by software 1: Window watchdog activated by hardware
	WWDG_HALT: *Window watchdog reset on Halt/Active-halt* 0: Window watchdog stopped in Halt/Active-halt mode 1: Window watchdog continues running in Halt/Active-halt mode

Caution: After a device reset, read access to the program memory is not guaranteed if address 0x4807 is not programmed to 0x00.

9.3.2 Power supply

Figure 10. Power supply scheme



1. Each power supply pair must be decoupled with filtering ceramic capacitors as shown above. These capacitors must be placed as close as possible to, or below, the appropriate pins to ensure the correct functionality of the device.
2. The 1µF capacitor must be connected to the V_{DD} pin.
3. The 1µF ceramic capacitor connected to PXS_VREG must be low ESR (ESR ≤ 1Ω).

9.3.3 Power-up / power-down operating conditions

Table 17. Operating conditions at power-up / power-down

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
t_{VDD}	V_{DD} rise time rate		20	-	1300	µs/V
t_{TEMP}	Reset release delay	V_{DD} rising	-	1	-	ms
V_{POR}	Power on reset threshold		1.44(2)	-	1.65(1)	V
V_{PDR}	Power down reset threshold		1.30(2)	-	1.60(2)	V

1. Tested in production.
2. Data based on characterization results, not tested in production.

Current consumption of on-chip peripherals

Measurement made for f_{MASTER} = from 2 MHz to 16 MHz

Table 22. Peripheral current consumption

Symbol	Parameter	Typ. V_{DD} = 3.0 V	Unit
$I_{DD(TIM2)}$	TIM2 supply current(1)	9	µA/MHz
$I_{DD(TIM3)}$	TIM3 supply current(1)	9	
$I_{DD(TIM4)}$	TIM4 timer supply current(1)	4	
$I_{DD(USART)}$	USARTsupply current(2)	7	
$I_{DD(SPI)}$	SPI supply current(2)	4	
$I_{DD(I^2C)}$	I2C supply current(2)	4	

1. Data based on a differential I_{DD} measurement between all peripherals off and a timer counter running at 16 MHz. The CPU is in Wait mode in both cases. No IC/OC programmed, no I/O pin toggling. Not tested in production.
2. Data based on a differential I_{DD} measurement between the on-chip peripheral when kept under reset and not clocked and the on-chip peripheral when clocked and not kept under reset. The CPU is in Wait mode in both cases. No I/O pin toggling. Not tested in production.

Current consumption with ProxSense peripherals

Measurement made for f_{MASTER} = 16 MHz, $f_{ProxSense}$ = 16 MHz, all other peripherals off and under the following conditions:

PXS_Rx*i*CSSELR (Sampling Capacitor) = 0x10

PXS_Rx*i*EPCCSELR (Electrode Parasitic Capacitance Compensation) = 0x80

Capacitance between Tx and Rx of 10nF

Table 23. ProxSense peripheral current consumption(1)

Symbol	ProxSense transmitter Tx	ProxSense receiver Rx	Typical	Unit
$I_{DD(PXS)}$	1	1	0.6	mA
	1	4	1.1	
	1	10	2.3	

1. Data based on characterization, not tested in production

9.3.6 Clock and timing characteristics

Internal clock source

The parameters given in *Table 24* are derived from tests performed under ambient temperature and V_{DD} supply voltage. They are subject to general operating conditions for V_{DD} and T_A.

High speed internal RC oscillator

Table 24. HSI oscillator characteristics(1)

Symbol	Parameter	Conditions	Min.	Typ.	Max.	Unit
f_{HSI}	Frequency	V_{DD} = 3.0 V	-	16	-	MHz
ACC_{HSI}	Accuracy of HSI oscillator (factory calibrated)	V_{DD} = 3.0 V, T_A = 25 °C	–1	-	1	%
		1.65 V ≤ VDD ≤ 3.6 V, T_A = -40 to 85°C	–3	-	3	%
$I_{DD(HSI)}$	HSI oscillator power consumption		-	70	100	µA

1. V_{DD} = 3V, T_A = –40 to 125 °C, unless otherwise specified.

Figure 14. Typical HSI frequency vs. V_{DD}

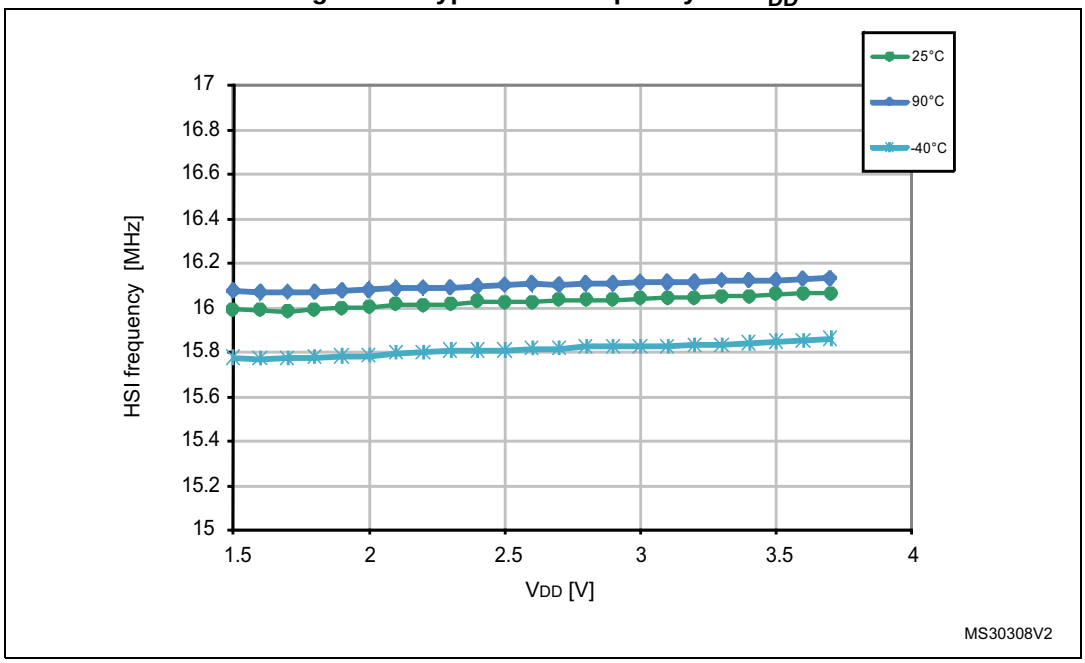
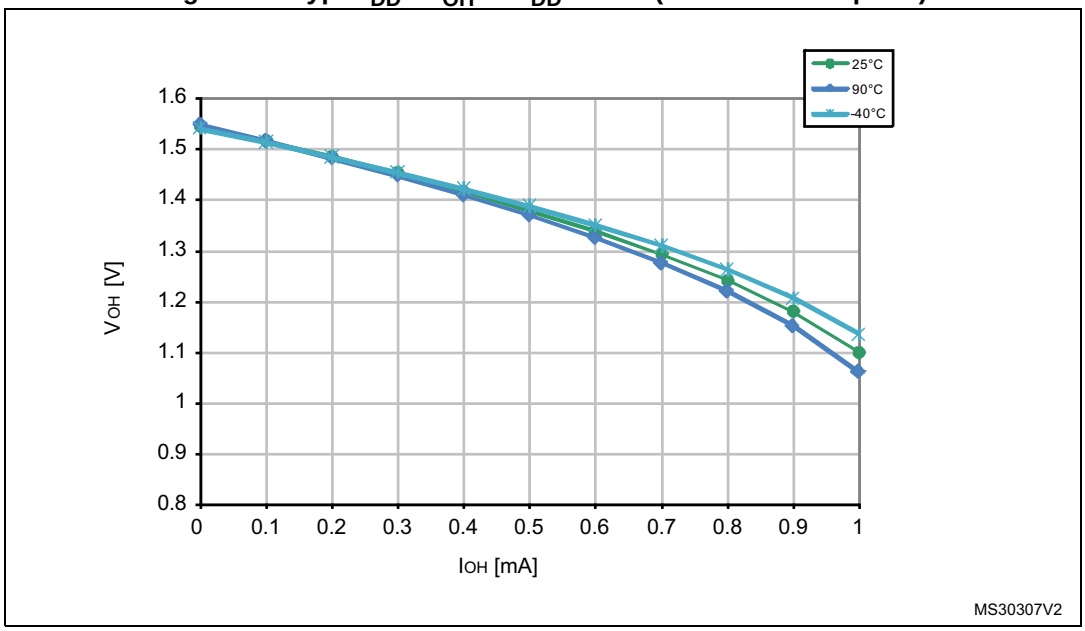


Figure 24. Typ. V_{DD} - V_{OH} at V_{DD} = 1.8V (ProxSense RX ports)



9.3.9 Communication interfaces

Serial peripheral interface (SPI)

The parameters given in *Table 34* are derived from tests performed under ambient temperature, f_{MASTER} frequency and V_{DD} supply voltage conditions summarized in *Section 9.3.1 on page 45*, unless otherwise specified. Refer to I/O port characteristics for more details on the input/output alternate function characteristics (NSS, SCK, MOSI, MISO).

Table 34. SPI characteristics

Symbol	Parameter	Conditions(1)	Min.	Max.	Unit
f_{SCK} $1/t_{c(SCK)}$	SPI clock frequency	Master mode	0	8	MHz
		Slave mode	0	8	
$t_{r(SCK)}$ $t_{f(SCK)}$	SPI clock rise and fall time	Capacitive load: C = 30 pF	-	30	ns
$t_{su(NSS)}$(2)	NSS setup time	Slave mode	4 x t_{MASTER}	-	
$t_{h(NSS)}$(2)	NSS hold time	Slave mode	80	-	
$t_{w(SCKH)}$(2) $t_{w(SCKL)}$(2)	SCK high and low time	Master mode, f_{MASTER} = 8 MHz, f_{SCK}= 4 MHz	105	145	
$t_{su(MI)}$(2) $t_{su(SI)}$(2)	Data input setup time	Master mode	30	-	
		Slave mode	3	-	
$t_{h(MI)}$(2) $t_{h(SI)}$(2)	Data input hold time	Master mode	15	-	
		Slave mode	0	-	
$t_{a(SO)}$(2)(3)	Data output access time	Slave mode	-	3x t_{MASTER}	
$t_{dis(SO)}$(2)(4)	Data output disable time	Slave mode	30	-	
$t_{v(SO)}$(2)	Data output valid time	Slave mode (after enable edge)	-	60	
$t_{v(MO)}$(2)	Data output valid time	Master mode (after enable edge)	-	20	
$t_{h(SO)}$(2)	Data output hold time	Slave mode (after enable edge)	15	-	
$t_{h(MO)}$(2)		Master mode (after enable edge)	1	-	

1. Parameters are given by selecting 10–MHz I/O output frequency.
2. Values based on design simulation and/or characterization results, and not tested in production.
3. Min time is for the minimum time to drive the output and max time is for the maximum time to validate the data.
4. Min time is for the minimum time to invalidate the output and max time is for the maximum time to put the data in Hi–Z.

11 Part numbering

Table 44. Ordering information scheme

Example: STM8 T L 5 3 G 4 U 6 TR

Device type
STM8 microcontroller family

Product type
T = Touch sensing

Device family
L = low-power

Sub-family type
5 = Projective capacitive sub-family

Peripheral set
2 = Light
3 = Basic

Pin count
C = 48 pins
G = 28 pins
F = 20 pins

Program memory size
4 = 16 Kbyte

Package type
U = UFQFPN
P = TSSOP

Temperature range
6 = -40 °C to +85 °C

Shipping
TR = Tape and Reel
Blank = Tray

1. For a list of available options (e.g. memory size, package) and orderable part numbers or for further information on any aspect of this device, please go to *www.st.com* or contact the ST Sales Office nearest to you.

12 STM8 development tools

Development tools for the STM8 microcontrollers include the very low-cost debugger and programmer tool ST-Link supported by a complete software tool package including C compiler, assembler and integrated development environment with high-level language debugger. In addition, the STM8 is to be supported by a complete range of tools including starter kits, evaluation boards and a low-cost in-circuit debugger/programmer.

12.1 Software tools

STM8 development tools are supported by a complete, free software package from STMicroelectronics that includes ST Visual Develop (STVD) IDE and the ST Visual Programmer (STVP) software interface. STVD provides seamless integration of the Cosmic and Raisonance C compilers for STM8. A free version that outputs up to 32 Kbytes of code is available.

12.1.1 STM8 toolset

STM8 toolset with STVD integrated development environment and STVP programming software is available for free download at www.st.com. This package includes:

ST Visual Develop (STVD) – Full-featured integrated development environment from ST, featuring

- Seamless integration of C and ASM toolsets
- Full-featured debugger
- Project management
- Syntax highlighting editor
- Integrated programming interface

ST Visual Programmer (STVP) – Easy-to-use, unlimited graphical interface allowing read, write and verify of your STM8 microcontroller Flash program memory, data EEPROM and option bytes. STVP also offers project mode for saving programming configurations and automating programming sequences.

12.1.2 STM-STUDIO

STM-STUDIO helps debug and diagnose STM8 and STM32 applications while they are running by reading and displaying their variables in real-time. STM-STUDIO perfectly complements traditional debugging tools to fine tune applications. It is well suited for debugging applications which cannot be stopped, such as TouchSensing applications. Its easy-to-use, graphical interface features:

- Non-intrusive read on-the-fly variables from RAM while the application is running
- Parse DWARF debugging information in the ELF application executable file
- Possibility to log data into a file, and replay later (exhaustive record display, not real-time)
- 2 types of viewers:
  - Variable viewer: Real-time waveforms, oscilloscope-like graphs
  - TouchPoint viewer: Association of 2 variables, one on the X axis, one on the Y axis

12.1.3 C and assembly toolchains

Control of C and assembly toolchains is seamlessly integrated into the STVD integrated development environment, making it possible to configure and control the building of your application directly from an easy-to-use graphical interface.

Available toolchains include:

- ***Cosmic C compiler for STM8*** – One free version that outputs up to 32 Kbytes of code is available. For more information, see www.cosmic-software.com.
- ***IAR embedded workbench*** – The C compiler for STM8 which is included in the toolset is free for up to 8Kbytes of code. For more information, see www.iar.com.
- ***Raisonance C compiler for STM8*** – One free version that outputs up to 32 Kbytes of code. For more information, see www.raisonance.com.
- ***STM8 assembler linker*** – Free assembly toolchain included in the STVD toolset, which allows you to assemble and link your application source code.

12.2 Programming tools

During the development cycle, ST-Link provides in-circuit programming of the STM8 Flash microcontroller on your application board via the SWIM protocol.

For production environments, programmers will include a complete range of gang and automated programming solutions from third-party tool developers already supplying programmers for the STM8 family.

Table 45. Document revision history (continued)

Date	Revision	Changes
28-May-2015	6	Updated *Figure 27: SPI timing diagram - slave mode and CPHA = 0*.
01-Jul-2015	7	Added the footnotes about “D” and E1” dimensions below *Table 43: TSSOP20 – 20-lead thin shrink small outline, 6.5 x 4.4 mm, 0.65 mm pitch, package mechanical data*.