Details	
Product Status	Active
Core Processor	STM8
Core Size	8-Bit
Speed	24MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	52
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	6K x 8
Voltage - Supply (Vcc/Vdd)	2.95V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/stm8s207r6t6

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 bytes/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

4.3 Interrupt controller

- Nested interrupts with three software priority levels
- 32 interrupt vectors with hardware priority
- Up to 37 external interrupts on six vectors including TLI
- Trap and reset interrupts

4.4 Flash program and data EEPROM memory

- Up to 128 Kbytes of high density Flash program single voltage Flash memory
- Up to 2K bytes true data EEPROM
- Read while write: Writing in data memory possible while executing code in program memory.
- 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 write to data EEPROM, modify the contents of main program memory or 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*.

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 lowest power consumption, 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 downcounter is refreshed before its value is lower than the one stored in the window register.

4.14.1 UART1

Main features

- One Mbit/s full duplex SCI
- SPI emulation
- High precision baud rate generator
- Smartcard 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

Synchronous communication

- Full duplex synchronous transfers
- SPI master operation
- 8-bit data communication
- Maximum speed: 1 Mbit/s at 16 MHz (f_{CPU}/16)

LIN master mode

- Emission: Generates 13-bit sync break frame
- Reception: Detects 11-bit break frame

4.14.2 UART3

Main features

- 1 Mbit/s full duplex SCI
- LIN master capable
- High precision baud rate generator

4.14.4 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)

4.14.5 beCAN

The beCAN controller (basic enhanced CAN), interfaces the CAN network and supports the CAN protocol version 2.0A and B. It has been designed to manage a high number of incoming messages efficiently with a minimum CPU load.

For safety-critical applications the beCAN controller provides all hardware functions to support the CAN time triggered communication option (TTCAN).

The maximum transmission speed is 1 Mbit.

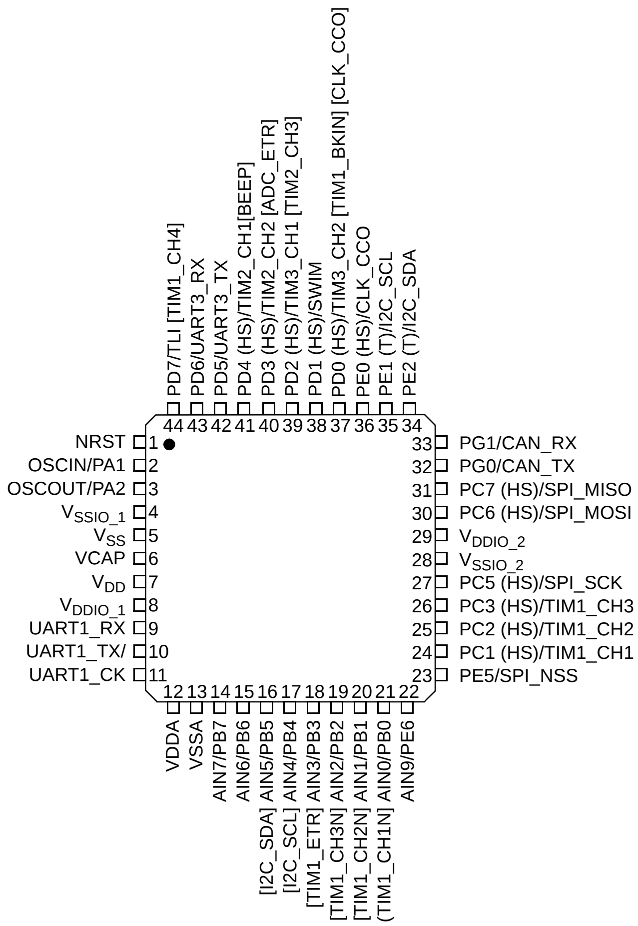
Transmission

- Three transmit mailboxes
- Configurable transmit priority by identifier or order request
- Time stamp on SOF transmission

Reception

- 8-, 11- and 29-bit ID
- One receive FIFO (3 messages deep)
- Software-efficient mailbox mapping at a unique address space
- FMI (filter match index) stored with message
- Configurable FIFO overrun
- Time stamp on SOF reception
- Six filter banks, 2 x 32 bytes (scalable to 4 x 16-bit) each, enabling various masking configurations, such as 12 filters for 29-bit ID or 48 filters for 11-bit ID
- Filtering modes:
  - Mask mode permitting ID range filtering
  - ID list mode
- Time triggered communication option
  - Disable automatic retransmission mode
  - 16-bit free running timer
  - Configurable timer resolution
  - Time stamp sent in last two data bytes

Figure 6. LQFP 44-pin pinout



1. (HS) high sink capability.
2. (T) True open drain (P-buffer and protection diode to V_{DD} not implemented).
3. [ ] alternate function remapping option (If the same alternate function is shown twice, it indicates an exclusive choice not a duplication of the function).
4. CAN_RX and CAN_TX is available on STM8S208xx devices only.

Table 6. Pin description (continued)

Pin number					Pin name	Type	Input			Output				Main function (after reset)	Default alternate function	Alternate function after remap [option bit]
LQFP80	LQFP64	LQFP48	LQFP44	LQFP32			floating	wpu	Ext. interrupt	High sink	Speed	OD	PP			
12	12	12	11	-	PA6/UART1_CK	I/O	**X**	X	X	HS	O3	X	X	**Port A6**	UART1 synchronous clock	
13	-	-	-	-	PH0	I/O	**X**	X		HS	O3	X	X	**Port H0**		
14	-	-	-	-	PH1	I/O	**X**	X		HS	O3	X	X	**Port H1**		
15	-	-	-	-	PH2	I/O	**X**	X			O1	X	X	**Port H2**		
16	-	-	-	-	PH3	I/O	**X**	X			O1	X	X	**Port H3**		
17	13	-	-	-	PF7/AIN15	I/O	**X**	X			O1	X	X	**Port F7**	Analog input 15	
18	14	-	-	-	PF6/AIN14	I/O	**X**	X			O1	X	X	**Port F6**	Analog input 14	
19	15	-	-	-	PF5/AIN13	I/O	**X**	X			O1	X	X	**Port F5**	Analog input 13	
20	16	-	-	8	PF4/AIN12	I/O	**X**	X			O1	X	X	**Port F4**	Analog input 12	
21	17	-	-	-	PF3/AIN11	I/O	**X**	X			O1	X	X	**Port F3**	Analog input 11	
22	18	-	-	-	V_{REF+}	S								ADC positive reference voltage		
23	19	13	12	9	V_{DDA}	S								Analog power supply		
24	20	14	13	10	V_{SSA}	S								Analog ground		
25	21	-	-	-	V_{REF-}	S								ADC negative reference voltage		
26	22	-	-	-	PF0/AIN10	I/O	**X**	X			O1	X	X	**Port F0**	Analog input 10	
27	23	15	14	-	PB7/AIN7	I/O	**X**	X	X		O1	X	X	**Port B7**	Analog input 7	
28	24	16	15	-	PB6/AIN6	I/O	**X**	X	X		O1	X	X	**Port B6**	Analog input 6	
29	25	17	16	11	PB5/AIN5	I/O	**X**	X	X		O1	X	X	**Port B5**	Analog input 5	I^2C_SDA [AFR6]
30	26	18	17	12	PB4/AIN4	I/O	**X**	X	X		O1	X	X	**Port B4**	Analog input 4	I^2C_SCL [AFR6]

Table 9. General hardware register map (continued)

Address	Block	Register label	Register name	Reset status
0x00 5250	TIM1	TIM1_CR1	TIM1 control register 1	0x00
0x00 5251		TIM1_CR2	TIM1 control register 2	0x00
0x00 5252		TIM1_SMCR	TIM1 slave mode control register	0x00
0x00 5253		TIM1_ETR	TIM1 external trigger register	0x00
0x00 5254		TIM1_IER	TIM1 Interrupt enable register	0x00
0x00 5255		TIM1_SR1	TIM1 status register 1	0x00
0x00 5256		TIM1_SR2	TIM1 status register 2	0x00
0x00 5257		TIM1_EGR	TIM1 event generation register	0x00
0x00 5258		TIM1_CCMR1	TIM1 capture/compare mode register 1	0x00
0x00 5259		TIM1_CCMR2	TIM1 capture/compare mode register 2	0x00
0x00 525A		TIM1_CCMR3	TIM1 capture/compare mode register 3	0x00
0x00 525B		TIM1_CCMR4	TIM1 capture/compare mode register 4	0x00
0x00 525C		TIM1_CCER1	TIM1 capture/compare enable register 1	0x00
0x00 525D		TIM1_CCER2	TIM1 capture/compare enable register 2	0x00
0x00 525E		TIM1_CNTRH	TIM1 counter high	0x00
0x00 525F		TIM1_CNTRL	TIM1 counter low	0x00
0x00 5260		TIM1_PSCRH	TIM1 prescaler register high	0x00
0x00 5261		TIM1_PSCRL	TIM1 prescaler register low	0x00
0x00 5262		TIM1_ARRH	TIM1 auto-reload register high	0xFF
0x00 5263		TIM1_ARRL	TIM1 auto-reload register low	0xFF
0x00 5264		TIM1_RCR	TIM1 repetition counter register	0x00
0x00 5265		TIM1_CCR1H	TIM1 capture/compare register 1 high	0x00
0x00 5266		TIM1_CCR1L	TIM1 capture/compare register 1 low	0x00
0x00 5267		TIM1_CCR2H	TIM1 capture/compare register 2 high	0x00
0x00 5268		TIM1_CCR2L	TIM1 capture/compare register 2 low	0x00
0x00 5269		TIM1_CCR3H	TIM1 capture/compare register 3 high	0x00
0x00 526A		TIM1_CCR3L	TIM1 capture/compare register 3 low	0x00
0x00 526B		TIM1_CCR4H	TIM1 capture/compare register 4 high	0x00
0x00 526C		TIM1_CCR4L	TIM1 capture/compare register 4 low	0x00
0x00 526D		TIM1_BKR	TIM1 break register	0x00
0x00 526E		TIM1_DTR	TIM1 dead-time register	0x00
0x00 526F		TIM1_OISR	TIM1 output idle state register	0x00
0x00 5270 to 0x00 52FF	Reserved area (147 bytes)			

Table 9. General hardware register map (continued)

Address	Block	Register label	Register name	Reset status
0x00 532B	TIM3	TIM3_ARRH	TIM3 auto-reload register high	0xFF
0x00 532C		TIM3_ARRL	TIM3 auto-reload register low	0xFF
0x00 532D		TIM3_CCR1H	TIM3 capture/compare register 1 high	0x00
0x00 532E		TIM3_CCR1L	TIM3 capture/compare register 1 low	0x00
0x00 532F		TIM3_CCR2H	TIM3 capture/compare register 2 high	0x00
0x00 5330		TIM3_CCR2L	TIM3 capture/compare register 2 low	0x00
0x00 5331 to 0x00 533F	Reserved area (15 bytes)			
0x00 5340	TIM4	TIM4_CR1	TIM4 control register 1	0x00
0x00 5341		TIM4_IER	TIM4 interrupt enable register	0x00
0x00 5342		TIM4_SR	TIM4 status register	0x00
0x00 5343		TIM4_EGR	TIM4 event generation register	0x00
0x00 5344		TIM4_CNTR	TIM4 counter	0x00
0x00 5345		TIM4_PSCR	TIM4 prescaler register	0x00
0x00 5346		TIM4_ARR	TIM4 auto-reload register	0xFF
0x00 5347 to 0x00 53FF	Reserved area (185 bytes)			
0x00 5400	ADC2	ADC _CSR	ADC control/status register	0x00
0x00 5401		ADC_CR1	ADC configuration register 1	0x00
0x00 5402		ADC_CR2	ADC configuration register 2	0x00
0x00 5403		ADC_CR3	ADC configuration register 3	0x00
0x00 5404		ADC_DRH	ADC data register high	0xXX
0x00 5405		ADC_DRL	ADC data register low	0xXX
0x00 5406		ADC_TDRH	ADC Schmitt trigger disable register high	0x00
0x00 5407		ADC_TDRL	ADC Schmitt trigger disable register low	0x00
0x00 5408 to 0x00 541F	Reserved area (24 bytes)			
0x00 5420	beCAN	CAN_MCR	CAN master control register	0x02
0x00 5421		CAN_MSR	CAN master status register	0x02
0x00 5422		CAN_TSR	CAN transmit status register	0x00
0x00 5423		CAN_TPR	CAN transmit priority register	0x0C
0x00 5424		CAN_RFR	CAN receive FIFO register	0x00
0x00 5425		CAN_IER	CAN interrupt enable register	0x00
0x00 5426		CAN_DGR	CAN diagnosis register	0x0C
0x00 5427		CAN_FPSR	CAN page selection register	0x00

Table 10. CPU/SWIM/debug module/interrupt controller registers

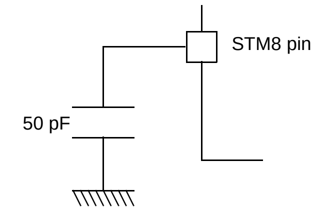
Address	Block	Register Label	Register Name	Reset Status
0x00 7F00	CPU(1)	A	Accumulator	0x00
0x00 7F01		PCE	Program counter extended	0x00
0x00 7F02		PCH	Program counter high	0x00
0x00 7F03		PCL	Program counter low	0x00
0x00 7F04		XH	X index register high	0x00
0x00 7F05		XL	X index register low	0x00
0x00 7F06		YH	Y index register high	0x00
0x00 7F07		YL	Y index register low	0x00
0x00 7F08		SPH	Stack pointer high	0x17(2)
0x00 7F09		SPL	Stack pointer low	0xFF
0x00 7F0A		CCR	Condition code register	0x28
0x00 7F0B to 0x00 7F5F	Reserved area (85 bytes)			
0x00 7F60	CPU	CFG_GCR	Global configuration register	0x00
0x00 7F70	ITC	ITC_SPR1	Interrupt software priority register 1	0xFF
0x00 7F71		ITC_SPR2	Interrupt software priority register 2	0xFF
0x00 7F72		ITC_SPR3	Interrupt software priority register 3	0xFF
0x00 7F73		ITC_SPR4	Interrupt software priority register 4	0xFF
0x00 7F74		ITC_SPR5	Interrupt software priority register 5	0xFF
0x00 7F75		ITC_SPR6	Interrupt software priority register 6	0xFF
0x00 7F76		ITC_SPR7	Interrupt software priority register 7	0xFF
0x00 7F77		ITC_SPR8	Interrupt software priority register 8	0xFF
0x00 7F78 to 0x00 7F79	Reserved area (2 bytes)			
0x00 7F80	SWIM	SWIM_CSR	SWIM control status register	0x00
0x00 7F81 to 0x00 7F8F	Reserved area (15 bytes)			
0x00 7F90	DM	DM_BK1RE	DM breakpoint 1 register extended byte	0xFF
0x00 7F91		DM_BK1RH	DM breakpoint 1 register high byte	0xFF
0x00 7F92		DM_BK1RL	DM breakpoint 1 register low byte	0xFF
0x00 7F93		DM_BK2RE	DM breakpoint 2 register extended byte	0xFF
0x00 7F94		DM_BK2RH	DM breakpoint 2 register high byte	0xFF
0x00 7F95		DM_BK2RL	DM breakpoint 2 register low byte	0xFF
0x00 7F96		DM_CR1	DM debug module control register 1	0x00
0x00 7F97		DM_CR2	DM debug module control register 2	0x00

10.1.5 Pin loading conditions

10.1.6 Loading capacitor

The loading conditions used for pin parameter measurement are shown in *Figure 10*.

Figure 10. Pin loading conditions



10.1.7 Pin input voltage

The input voltage measurement on a pin of the device is described in *Figure 11*.

Figure 11. Pin input voltage

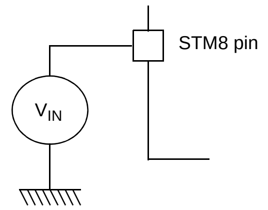


Table 21. Total current consumption with code execution in run mode at V_{DD} = 3.3 V

Symbol	Parameter	Conditions		Typ	Max(1)	Unit
$I_{DD(RUN)}$	Supply current in run mode, code executed from RAM	$f_{CPU} = f_{MASTER}$ = 24 MHz, $T_A \le$ 105 °C	HSE crystal osc. (24 MHz)	4.0		mA
			HSE user ext. clock (24 MHz)	3.7	7.3	
		$f_{CPU} = f_{MASTER}$ = 16 MHz	HSE crystal osc. (16 MHz)	2.9		
			HSE user ext. clock (16 MHz)	2.7	5.8	
			HSI RC osc. (16 MHz)	2.5	3.4	
		$f_{CPU} = f_{MASTER}$/128 = 125 kHz	HSE user ext. clock (16 MHz)	1.2	4.1	
			HSI RC osc. (16 MHz)	1.0	1.3	
		$f_{CPU} = f_{MASTER}$/128 = 15.625 kHz	HSI RC osc. (16MHz/8)	0.55		
		$f_{CPU} = f_{MASTER}$ = 128 kHz	LSI RC osc. (128 kHz)	0.45		
	Supply current in run mode, code executed from Flash	$f_{CPU} = f_{MASTER}$ = 24 MHz, $T_A \le$ 105 °C	HSE crystal osc. (24 MHz)	11.0		
			HSE user ext. clock (24 MHz)	10.8	18.0	
		$f_{CPU} = f_{MASTER}$ = 16 MHz	HSE crystal osc. (16 MHz)	8.4		
			HSE user ext. clock (16 MHz)	8.2	15.2	
			HSI RC osc. (16 MHz)	8.1	13.2	
		$f_{CPU} = f_{MASTER}$ = 2 MHz.	HSI RC osc. (16 MHz/8)(2)	1.5		
		$f_{CPU} = f_{MASTER}$/128 = 125 kHz	HSI RC osc. (16 MHz)	1.1		
		$f_{CPU} = f_{MASTER}$/128 = 15.625 kHz	HSI RC osc. (16 MHz/8)	0.6		
		$f_{CPU} = f_{MASTER}$ = 128 kHz	LSI RC osc. (128 kHz)	0.55		

1. Data based on characterization results, not tested in production.
2. Default clock configuration.

Total current consumption in halt mode

Table 26. Total current consumption in halt mode at V_{DD} = 5 V

Symbol	Parameter	Conditions	Typ	Max at 85 °C	Max at 125 °C	Unit
$I_{DD(H)}$	Supply current in halt mode	Flash in operating mode, HSI clock after wakeup	63.5			µA
		Flash in power-down mode, HSI clock after wakeup	6.5	35	100	

Table 27. Total current consumption in halt mode at V_{DD} = 3.3 V

Symbol	Parameter	Conditions	Typ	Unit
$I_{DD(H)}$	Supply current in halt mode	Flash in operating mode, HSI clock after wakeup	61.5	µA
		Flash in power-down mode, HSI clock after wakeup	4.5	

Low power mode wakeup times

Table 28. Wakeup times

Symbol	Parameter	Conditions			Typ	Max[(1)]	Unit
$t_{WU(WFI)}$	Wakeup time from wait mode to run mode[(3)]					See note[(2)]	µs
		$f_{CPU} = f_{MASTER}$ = 16 MHz.			0.56		
$t_{WU(AH)}$	Wakeup time active halt mode to run mode.[(3)]	MVR voltage regulator on[(4)]	Flash in operating mode[(5)]	HSI (after wakeup)	1[(6)]	2[(6)]	
			Flash in power-down mode[(5)]		3[(6)]		
		MVR voltage regulator off[(4)]	Flash in operating mode[(5)]		48[(6)]		
			Flash in power-down mode[(5)]		50[(6)]		
$t_{WU(H)}$	Wakeup time from halt mode to run mode[(3)]	Flash in operating mode[(5)]			52		
		Flash in power-down mode[(5)]			54		

1. Data guaranteed by design, not tested in production.
2. $t_{WU(WFI)} = 2 \times 1/f_{master} + 7 \times 1/f_{CPU}$
3. Measured from interrupt event to interrupt vector fetch.
4. Configured by the REGAH bit in the CLK_ICKR register.
5. Configured by the AHALT bit in the FLASH_CR1 register.
6. Plus 1 LSI clock depending on synchronization.

Figure 29. Typ. V_{DD} - V_{OH} @ V_{DD} = 5 V (standard ports)

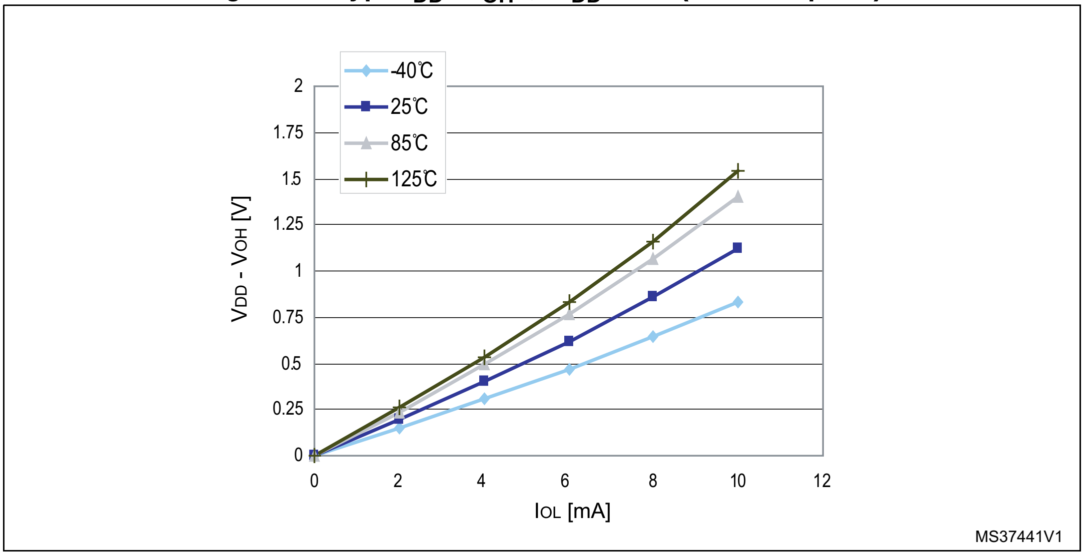


Figure 30. Typ. V_{DD} - V_{OH} @ V_{DD} = 3.3 V (standard ports)

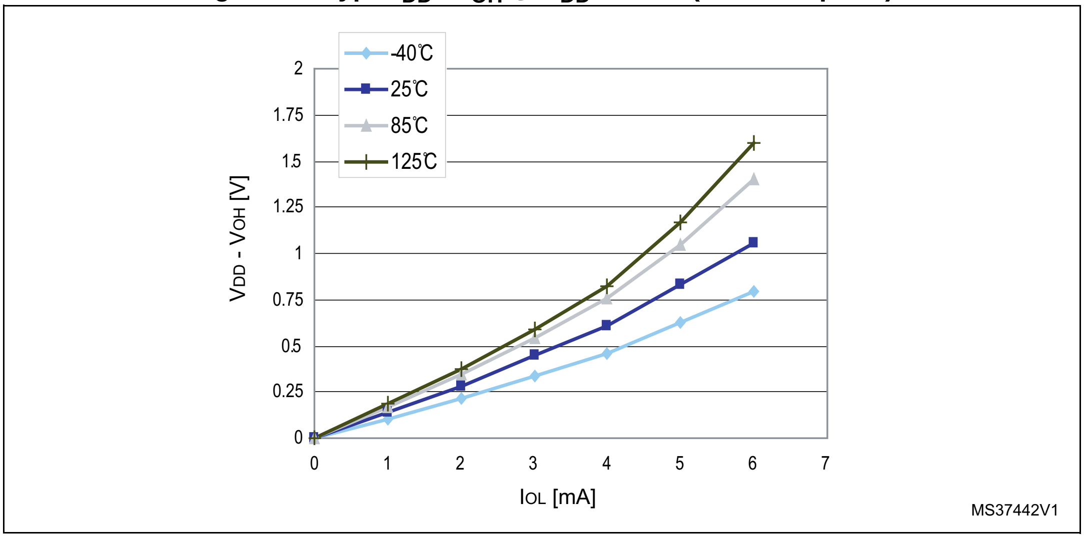
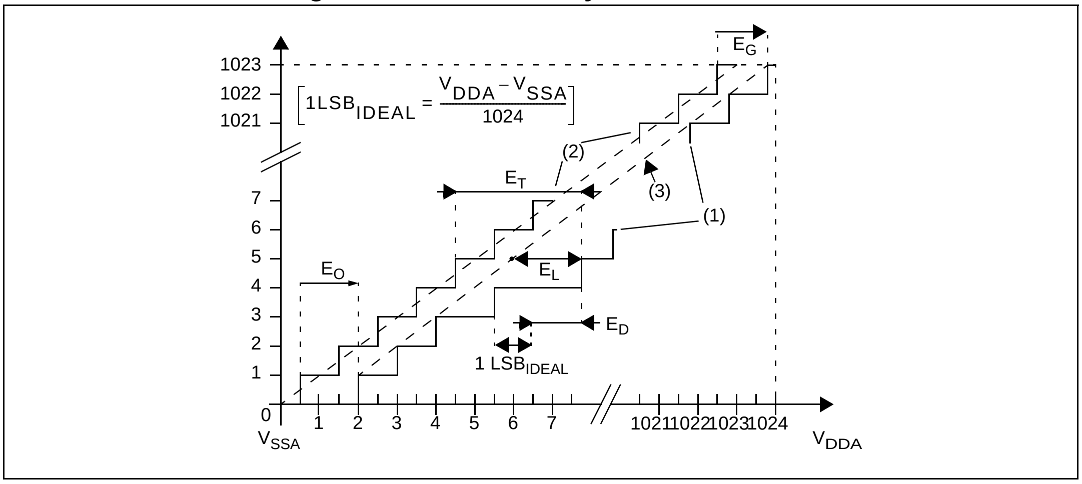
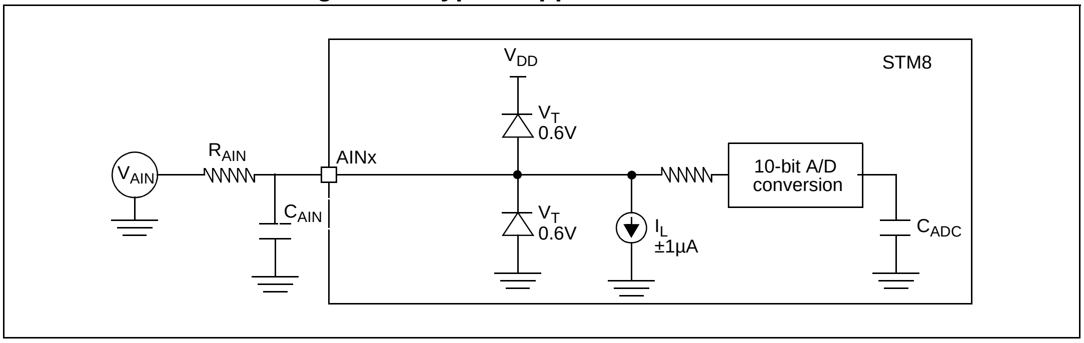


Figure 41. ADC accuracy characteristics



1. Example of an actual transfer curve.
2. The ideal transfer curve
3. End point correlation line
E_T = Total unadjusted error: maximum deviation between the actual and the ideal transfer curves.
E_O = Offset error: deviation between the first actual transition and the first ideal one.
E_G = Gain error: deviation between the last ideal transition and the last actual one.
E_D = Differential linearity error: maximum deviation between actual steps and the ideal one.
E_L = Integral linearity error: maximum deviation between any actual transition and the end point correlation line.

Figure 42. Typical application with ADC



11.1.2 LQFP64 package information

Figure 46. LQFP64 - 64-pin 14 mm x 14 mm low-profile quad flat package outline

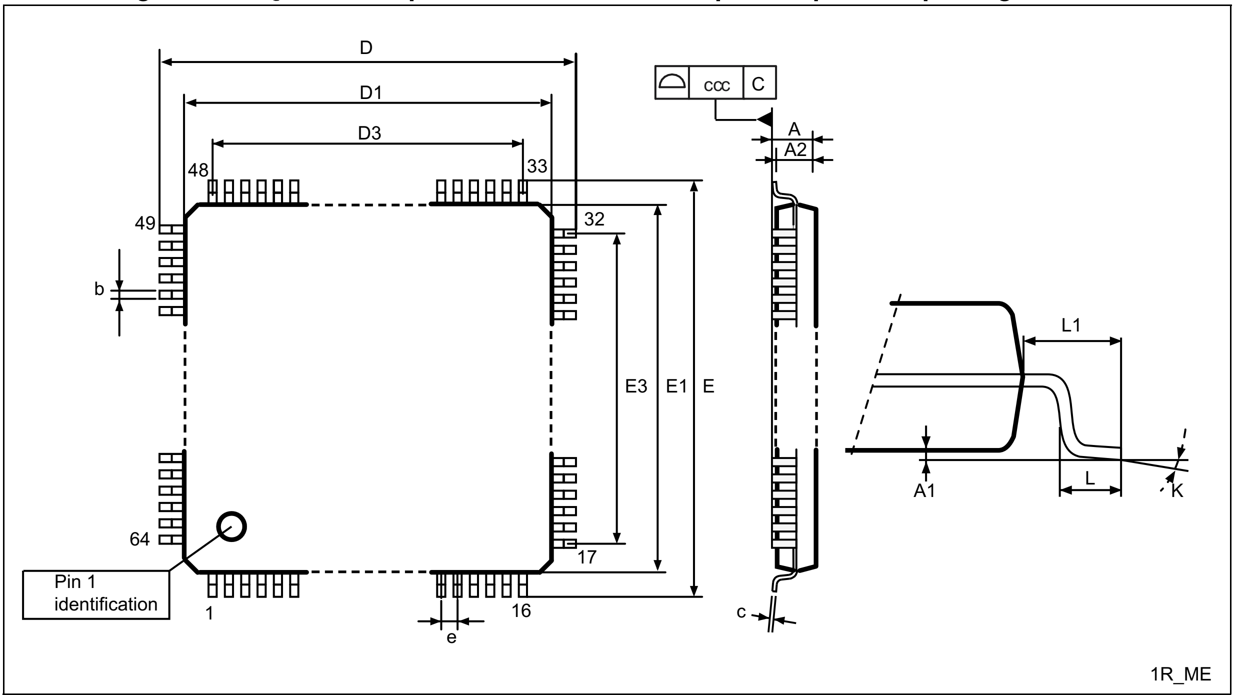


Table 52. LQFP64 - 64-pin, 14 x 14 mm low-profile quad flat package mechanical data

Symbol	mm			inches[(1)]		
	Min	Typ	Max	Min	Typ	Max
A			1.600			0.0630
A1	0.050		0.150	0.0020		0.0059
A2	1.350	1.400	1.450	0.0531	0.0551	0.0571
b	0.300	0.370	0.450	0.0118	0.0146	0.0177
C	0.090		0.200	0.0035		0.0079
D	15.800	16.000	16.200	0.6220	0.6299	0.6378
D1	13.800	14.000	14.200	0.5433	0.5512	0.5591
D3		12.000			0.4724	
E	15.800	16.000	16.200	0.6220	0.6299	0.6378
E1	13.800	14.000	14.200	0.5433	0.5512	0.5591
E3		12.000			0.4724	
e		0.800			0.0315	
L	0.450	0.600	0.750	0.0177	0.0236	0.0295
L1		1.000			0.0394	

11.1.3 LQFP48 package information

Figure 50. LQFP48 - 48-pin, 7 x 7 mm low-profile quad flat package outline

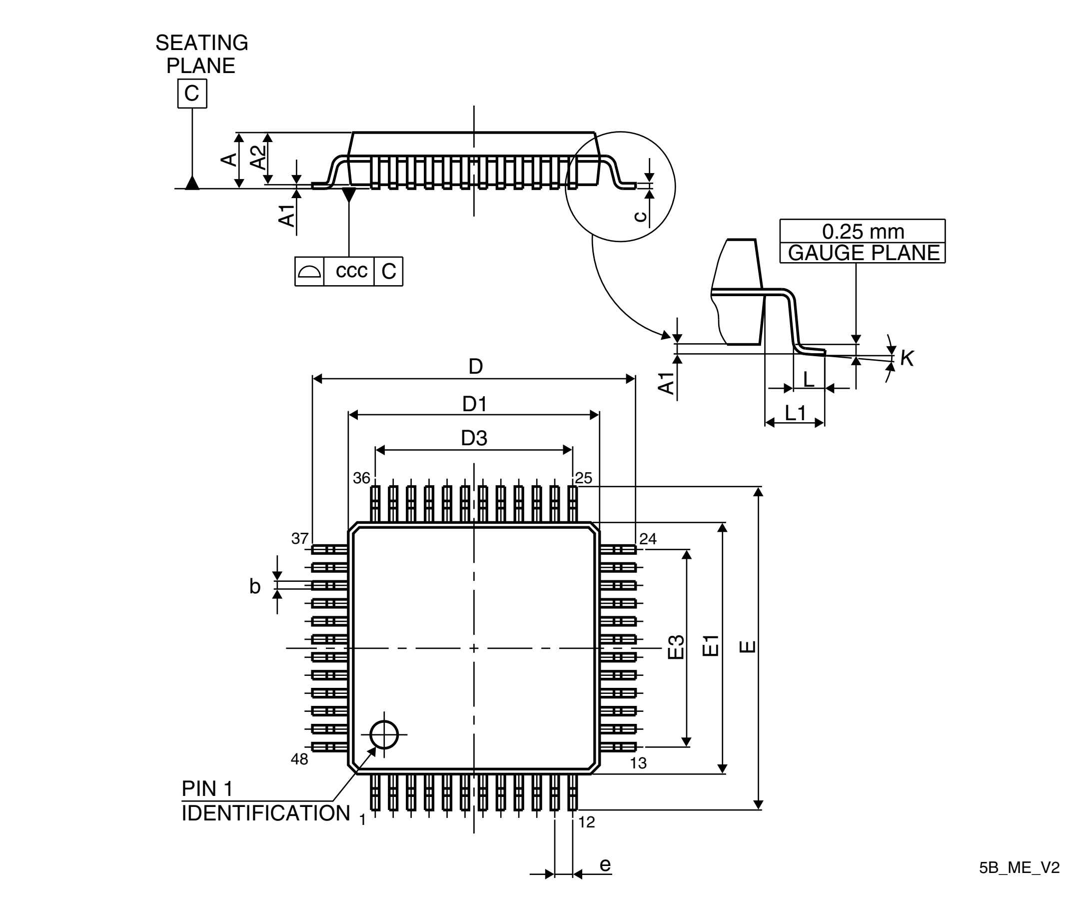
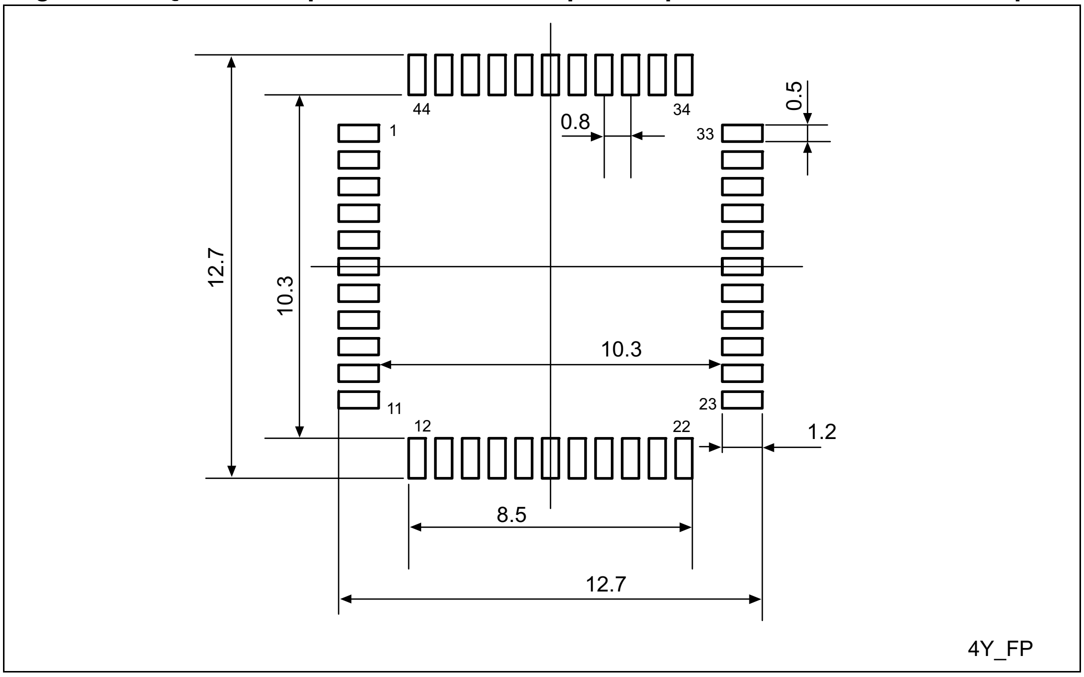


Table 54. LQFP48 - 48-pin, 7x 7 mm low-profile quad flat package mechanical

Symbol	mm			inches(1)		
	Min	Typ	Max	Min	Typ	Max
A	-	-	1.600	-	-	0.0630
A1	0.050	-	0.150	0.0020	-	0.0059
A2	1.350	1.400	1.450	0.0531	0.0551	0.0571
b	0.170	0.220	0.270	0.0067	0.0087	0.0106
c	0.090	-	0.200	0.0035	-	0.0079
D	8.800	9.000	9.200	0.3465	0.3543	0.3622
D1	6.800	7.000	7.200	0.2677	0.2756	0.2835
D3	-	5.500	-	-	0.2165	-
E	8.800	9.000	9.200	0.3465	0.3543	0.3622

Figure 54. LQFP44 - 44-pin, 10 x 10 mm low-profile quad flat recommended footprint

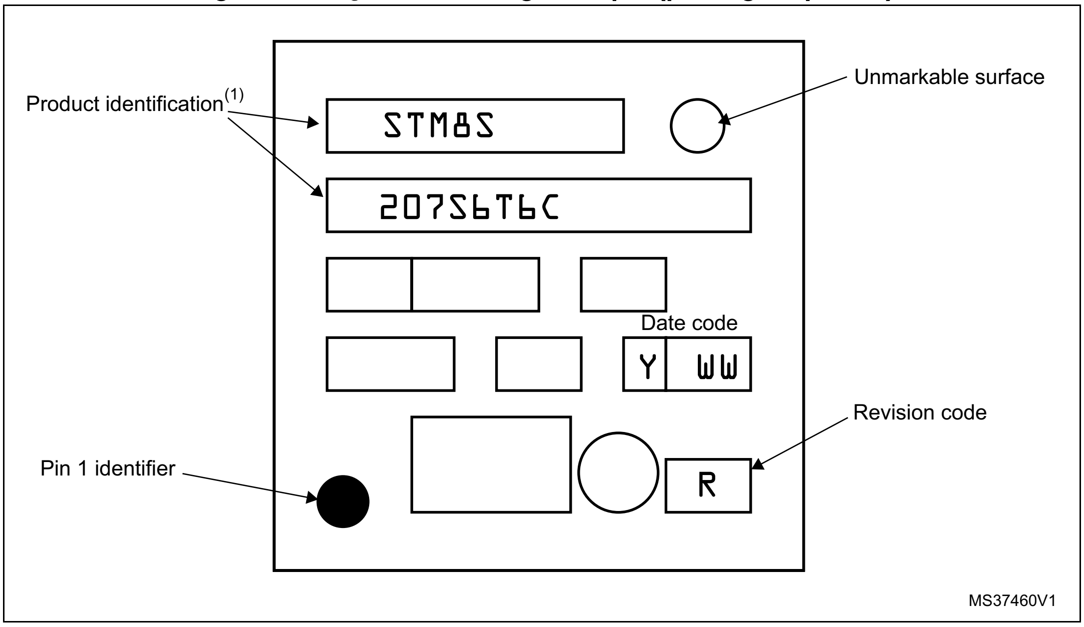


1. Dimensions are expressed in millimeters.

Device marking

The following figure shows the marking for the LQFP44 package.

Figure 55. LQFP44 marking example (package top view)



1. Parts marked as “ES”,”E” or accompanied by an Engineering Sample notification letter, are not yet qualified and therefore not yet ready to be used in production and any consequences deriving from such usage will not be at ST charge. In no event, ST will be liable for any customer usage of these engineering samples in production. ST Quality has to be contacted prior to any decision to use these Engineering samples to run qualification activity.

12.2 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.

12.2.1 STM8 toolset

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

ST Visual Develop – 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
- Support of advanced emulation features for STice such as code profiling and coverage

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

12.2.2 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 the 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.
- ***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 the application source code.

12.3 Programming tools

During the development cycle, STice provides in-circuit programming of the STM8 Flash microcontroller on the application board via the SWIM protocol. Additional tools are to include a low-cost in-circuit programmer as well as ST socket boards, which provide dedicated programming platforms with sockets for programming the STM8.

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.