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	Obsolete
Core Processor	ST7
Core Size	8-Bit
Speed	8MHz
Connectivity	I²C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	32
Program Memory Size	60KB (60K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	3.8V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f321j9tae

Figure 1. Device block diagram

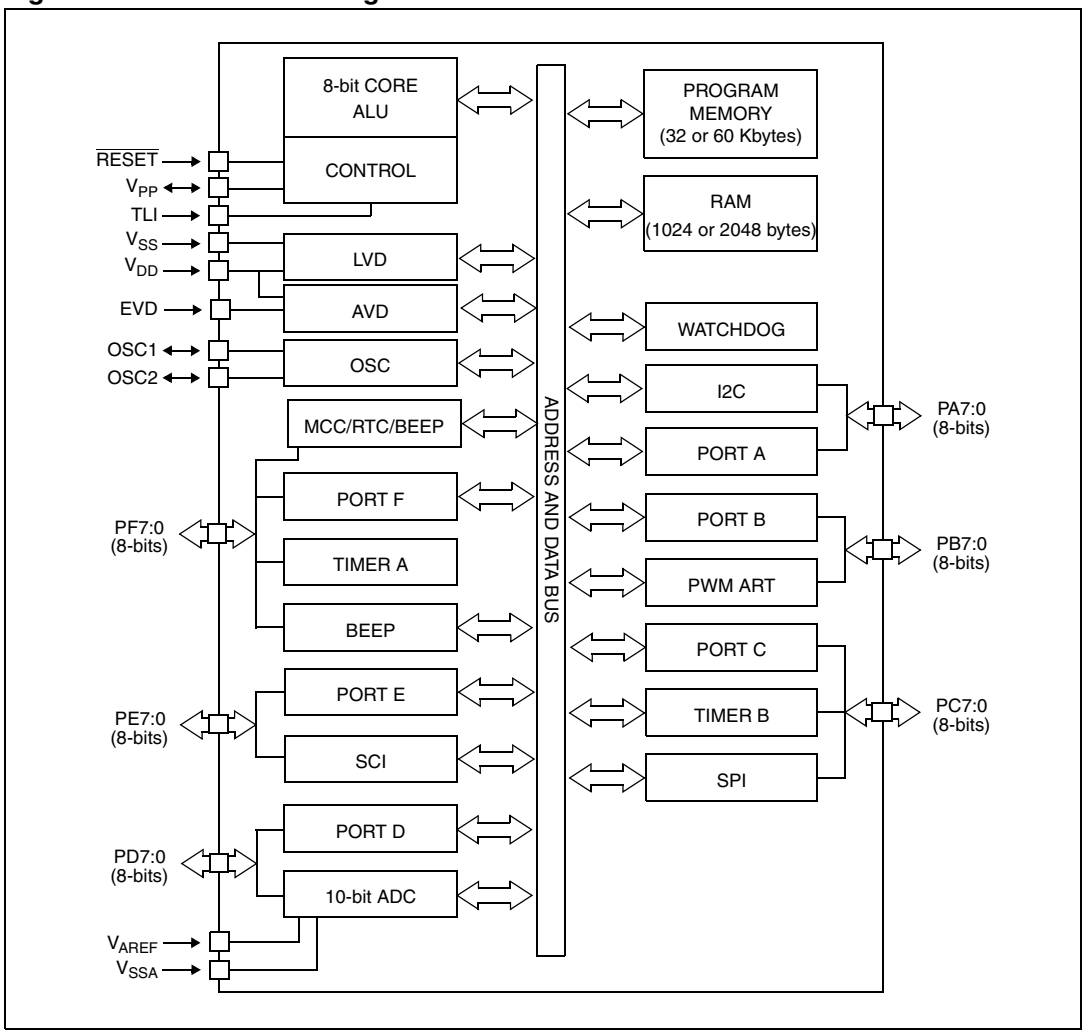
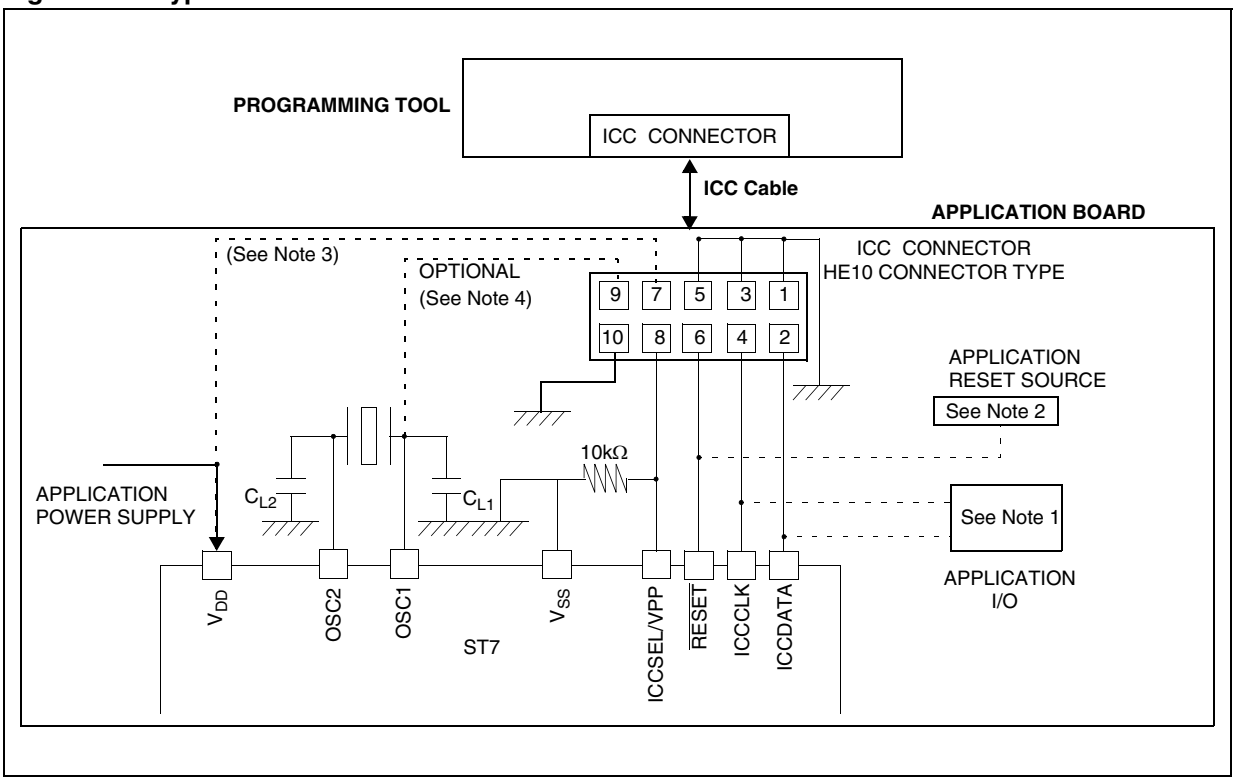


Figure 6. Typical ICC interface



1. If the ICCCLK or ICCDATA pins are only used as outputs in the application, no signal isolation is necessary. As soon as the programming tool is plugged to the board, even if an ICC session is not in progress, the ICCCLK and ICCDATA pins are not available for the application. If they are used as inputs by the application, isolation such as a serial resistor has to implemented in case another device forces the signal. Refer to the programming tool documentation for recommended resistor values.
2. During the ICC session, the programming tool must control the $\overline{\text{RESET}}$ pin. This can lead to conflicts between the programming tool and the application reset circuit if it drives more than 5mA at high level (push-pull output or pull-up resistor < 1K). A schottky diode can be used to isolate the application RESET circuit in this case. When using a classical RC network with R > 1K or a reset management IC with open-drain output and pull-up resistor > 1K, no additional components are needed. In all cases the user must ensure that no external reset is generated by the application during the ICC session.
3. The use of Pin 7 of the ICC connector depends on the programming tool architecture. This pin must be connected when using most ST programming tools (it is used to monitor the application power supply). Please refer to the programming tool manual.
4. Pin 9 has to be connected to the OSC1 or OSCIN pin of the ST7 when the clock is not available in the application or if the selected clock option is not programmed in the option byte. ST7 devices with multi-oscillator capability need to have OSC2 grounded in this case.

4.5 ICP (in-circuit programming)

To perform ICP the microcontroller must be switched to ICC (in-circuit communication) mode by an external controller or programming tool.

Depending on the ICP code downloaded in RAM, Flash memory programming can be fully customized (number of bytes to program, program locations, or selection serial communication interface for downloading).

When using an STMicroelectronics or third-party programming tool that supports ICP and the specific microcontroller device, the user needs only to implement the ICP hardware interface on the application board (see *Figure 6*). For more details on the pin locations, refer to the device pinout description.

6.6 System integrity management (SI)

The System Integrity Management block contains the Low Voltage Detector (LVD) and Auxiliary Voltage Detector (AVD) functions. It is managed by the SICSR register.

6.6.1 Low voltage detector (LVD)

The low voltage detector function (LVD) generates a static reset when the V_{DD} supply voltage is below a V_{IT-} reference value. This means that it secures the power-up as well as the power-down keeping the ST7 in reset.

The V_{IT-} reference value for a voltage drop is lower than the V_{IT+} reference value for power-on in order to avoid a parasitic reset when the MCU starts running and sinks current on the supply (hysteresis).

The LVD reset circuitry generates a reset when V_{DD} is below:

- V_{IT+} when V_{DD} is rising
- V_{IT-} when V_{DD} is falling

The LVD function is illustrated in *Figure 14*.

The voltage threshold can be configured by option byte to be low, medium or high.

Provided the minimum V_{DD} value (guaranteed for the oscillator frequency) is above V_{IT-}, the MCU can only be in two modes:

- under full software control
- in static safe reset

In these conditions, secure operation is always ensured for the application without the need for external reset hardware.

During a low voltage detector reset, the $\overline{\text{RESET}}$ pin is held low, thus permitting the MCU to reset other devices.

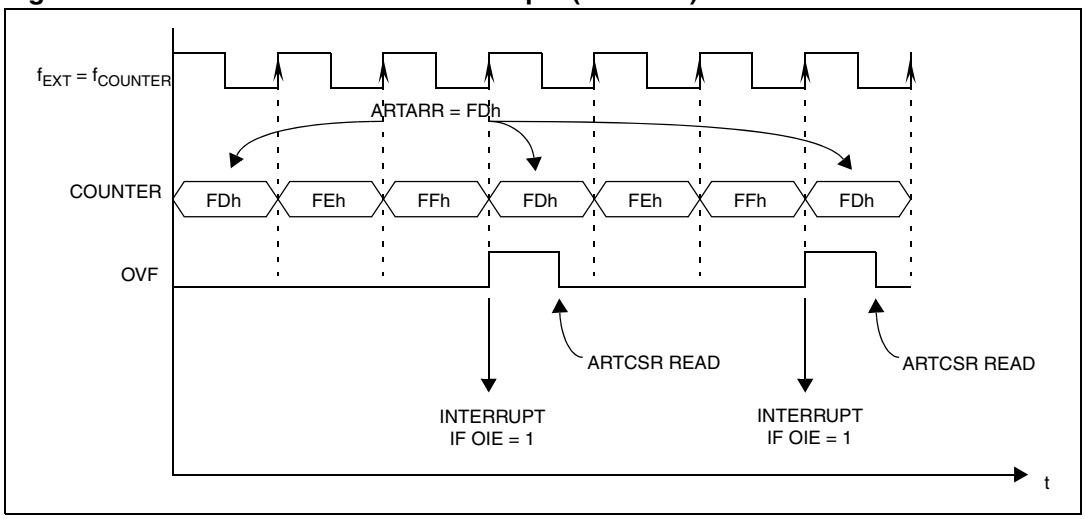
Note: The LVD allows the device to be used without any external RESET circuitry.

If the medium or low thresholds are selected, the detection may occur outside the specified operating voltage range. Below 3.8V, device operation is not guaranteed.

The LVD is an optional function which can be selected by option byte.

It is recommended to make sure that the V_{DD} supply voltage rises monotonously when the device is exiting from Reset, to ensure the application functions properly.

Figure 39. External event detector example (3 counts)



12.2.8 Input capture function

This mode allows the measurement of external signal pulse widths through ARTICRx registers.

Each input capture can generate an interrupt independently on a selected input signal transition. This event is flagged by a set of the corresponding CFx bits of the Input Capture Control/Status register (ARTICCSR).

These input capture interrupts are enabled through the CIEx bits of the ARTICCSR register.

The active transition (falling or rising edge) is software programmable through the CSx bits of the ARTICCSR register.

The read only input capture registers (ARTICRx) are used to latch the auto-reload counter value when a transition is detected on the ARTICx pin (CFx bit set in ARTICCSR register). After fetching the interrupt vector, the CFx flags can be read to identify the interrupt source.

Note: After a capture detection, data transfer in the ARTICRx register is inhibited until it is read (clearing the CFx bit).

The timer interrupt remains pending while the CFx flag is set when the interrupt is enabled (CIEx bit set). This means that the ARTICRx register has to be read at each capture event to clear the CFx flag.

The timing resolution is given by auto-reload counter cycle time ($1/f_{COUNTER}$).

Note: During Halt mode, if both the input capture and the external clock are enabled, the ARTICRx register value is not guaranteed if the input capture pin and the external clock change simultaneously.

The OC1R register value required for a specific timing application can be calculated using the following formula:

$$OC\mathit{i}R \text{ value} = \frac{t * f_{CPU} - 5}{PRESC}$$

Where:

t = Pulse period (in seconds)

f_{CPU} = CPU clock frequency (in hertz)

PRESC = Timer prescaler factor (2, 4 or 8 depending on the CC[1:0] bits; see *Table 61: Timer clock selection*)

If the timer clock is an external clock the formula is:

$$OC\mathit{i}R = t * f_{EXT} - 5$$

Where:

t = Pulse period (in seconds)

f_{EXT} = External clock frequency (in hertz)

When the value of the counter is equal to the value of the contents of the OC1R register, the OLVL1 bit is output on the OCMP1 pin (see *Figure 52*).

Note: 1 The OCF1 bit cannot be set by hardware in one pulse mode but the OCF2 bit can generate an Output Compare interrupt.

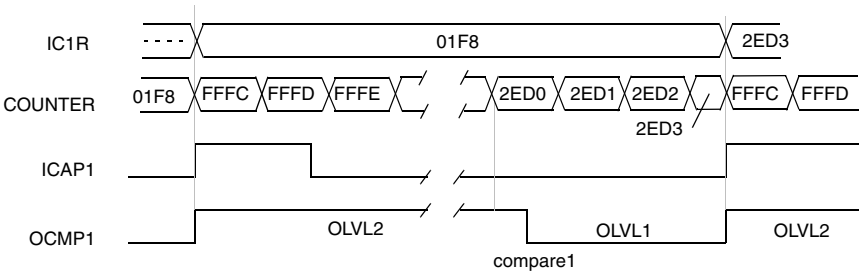
2 When the Pulse Width Modulation (PWM) and One Pulse Mode (OPM) bits are both set, the PWM mode is the only active one.

3 If OLVL1 = OLVL2 a continuous signal will be seen on the OCMP1 pin.

4 The ICAP1 pin cannot be used to perform input capture. The ICAP2 pin can be used to perform input capture (ICF2 can be set and IC2R can be loaded) but the user must take care that the counter is reset each time a valid edge occurs on the ICAP1 pin and ICF1 can also generates interrupt if ICIE is set.

5 When one pulse mode is used OC1R is dedicated to this mode. Nevertheless OC2R and OCF2 can be used to indicate a period of time has been elapsed but cannot generate an output waveform because the level OLVL2 is dedicated to the one pulse mode.

Figure 52. One pulse mode timing example



Note: IEDG1 = 1, OC1R = 2ED0h, OLVL1 = 0, OLVL2 = 1

Table 60. CR2 register description (continued)

Bit	Name	Function
5	OPM	*One Pulse Mode* 0: One Pulse Mode is not active. 1: One Pulse Mode is active, the ICAP1 pin can be used to trigger one pulse on the OCMP1 pin; the active transition is given by the IEDG1 bit. The length of the generated pulse depends on the contents of the OC1R register.
4	PWM	*Pulse Width Modulation* 0: PWM mode is not active. 1: PWM mode is active, the OCMP1 pin outputs a programmable cyclic signal; the length of the pulse depends on the value of OC1R register; the period depends on the value of OC2R register.
3:2	CC[1:0]	*Clock Control* The timer clock mode depends on these bits (see *Table 61*).
1	IEDG2	*Input Edge 2* This bit determines which type of level transition on the ICAP2 pin will trigger the capture. 0: A falling edge triggers the capture. 1: A rising edge triggers the capture.
0	EXEDG	*External Clock Edge* This bit determines which type of level transition on the external clock pin EXTCLK will trigger the counter register. 0: A falling edge triggers the counter register. 1: A rising edge triggers the counter register.

Table 61. Timer clock selection

Timer clock	CC1	CC0
f_{CPU} / 4	0	0
f_{CPU} / 2	0	1
f_{CPU} / 8	1	0
External clock (where available)(1)	1	1

1. If the external clock pin is not available, programming the external clock configuration stops the counter.

13.7.3 Control/status register (CSR)

CSR Reset value: xxxx x0xx (xxh)

7	6	5	4	3	2	1	0
ICF1	OCF1	TOF	ICF2	OCF2	TIMD	Reserved	
RO	RO	RO	RO	RO	RW	-	

13.7.13 Alternate counter low register (ACLR)

This is an 8-bit register that contains the low part of the counter value. A write to this register resets the counter. An access to this register after an access to CSR register does not clear the TOF bit in the CSR register.

ACLR — Reset value: 1111 1100 (FCh)

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

13.7.14 Input capture 2 high register (IC2HR)

This is an 8-bit read only register that contains the high part of the counter value (transferred by the Input Capture 2 event).

IC2HR — Reset value: Undefined

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

13.7.15 Input capture 2 low register (IC2LR)

This is an 8-bit read only register that contains the low part of the counter value (transferred by the Input Capture 2 event).

IC2LR — Reset value: Undefined

7	6	5	4	3	2	1	0
MSB							LSB
RO	RO	RO	RO	RO	RO	RO	RO

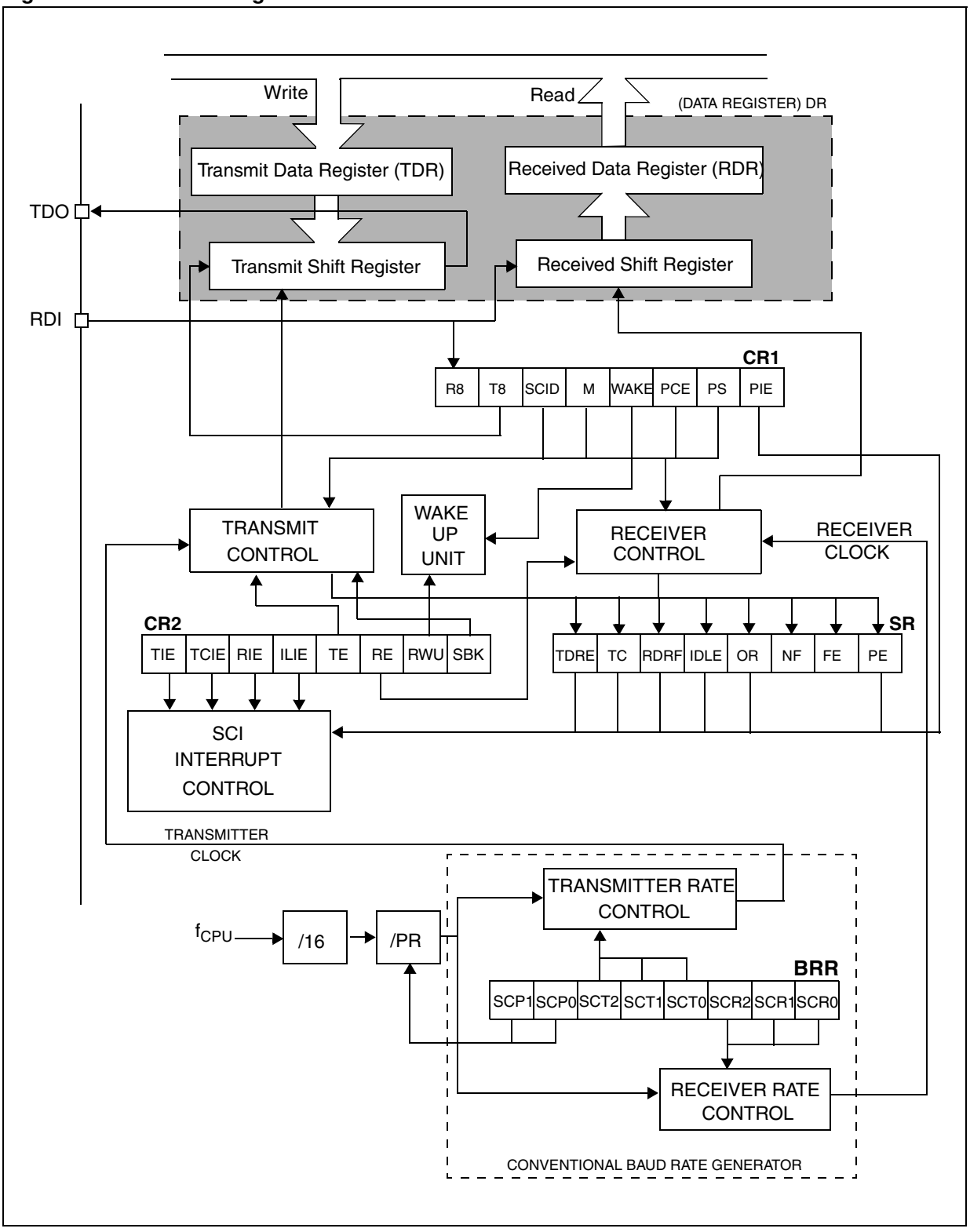
Warning: A write to the SPIDR register places data directly into the shift register for transmission.

A read to the SPIDR register returns the value located in the buffer and not the content of the shift register (see *Figure 55*).

Table 69. SPI register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0021h	SPIDR Reset value	MSB x	x	x	x	x	x	x	LSB x
0022h	SPICR Reset value	SPIE 0	SPE 0	SPR2 0	MSTR 0	CPOL x	CPHA x	SPR1 x	SPR0 x
0023h	SPICSR Reset value	SPIF 0	WCOL 0	OVR 0	MODF 0	0	SOD 0	SSM 0	SSI 0

Figure 62. SCI block diagram



Conventional baud rate generation

The baud rate for the receiver and transmitter (Rx and Tx) are set independently and calculated as follows:

$$Tx = \frac{f_{CPU}}{(16*PR)*TR} \qquad Rx = \frac{f_{CPU}}{(16*PR)*RR}$$

with:

PR = 1, 3, 4 or 13 (see SCP[1:0] bits)

TR = 1, 2, 4, 8, 16, 32, 64,128 (see SCT[2:0] bits)

RR = 1, 2, 4, 8, 16, 32, 64,128 (see SCR[2:0] bits)

All these bits are in the SCIBRR register.

Example: If f_{CPU} is 8 MHz (normal mode) and if PR = 13 and TR = RR = 1, the transmit and receive baud rates are 38400 baud.

Note: The baud rate registers MUST NOT be changed while the transmitter or the receiver is enabled.

Extended baud rate generation

The extended prescaler option provides a very fine tuning of the baud rate, using a 255 value prescaler, whereas the conventional baud rate generator retains industry standard software compatibility.

The extended baud rate generator block diagram is described in the *Figure 64*.

The output clock rate sent to the transmitter or to the receiver is the output from the 16 divider divided by a factor ranging from 1 to 255 set in the SCIERPR or the SCIETPR register.

Note: The extended prescaler is activated by setting the SCIETPR or SCIERPR register to a value other than zero. The baud rates are calculated as follows:

$$Tx = \frac{f_{CPU}}{16*ETPR*(PR*TR)} \qquad Rx = \frac{f_{CPU}}{16*ERPR*(PR*RR)}$$

with:

ETPR = 1,..,255 (see SCIETPR register)

ERPR = 1,..,255 (see SCIERPR register)

Receiver muting and wake-up feature

In multiprocessor configurations it is often desirable that only the intended message recipient should actively receive the full message contents, thus reducing redundant SCI service overhead for all non-addressed receivers.

The non-addressed devices may be placed in sleep mode by means of the muting function.

Setting the RWU bit by software puts the SCI in sleep mode:

- All the reception status bits cannot be set.
- All the receive interrupts are inhibited.

A muted receiver may be awakened by one of the following two ways:

- by Idle Line detection if the WAKE bit is reset
- by Address Mark detection if the WAKE bit is set

Consequently, the bit length must be long enough so that the 8th, 9th and 10th samples have the desired bit value. This means the clock frequency should not vary more than 6/16 (37.5%) within one bit. The sampling clock is resynchronized at each start bit, so that when receiving 10 bits (one start bit, 1 data byte, 1 stop bit), the clock deviation must not exceed 3.75%.

Note: The internal sampling clock of the microcontroller samples the pin value on every falling edge. Therefore, the internal sampling clock and the time the application expects the sampling to take place may be out of sync. For example: If the baud rate is 15.625 Kbaud (bit length is 64µs), then the 8th, 9th and 10th samples are at 28µs, 32µs and 36µs respectively (the first sample starting ideally at 0µs). But if the falling edge of the internal clock occurs just before the pin value changes, the samples would then be out of sync by ~4us. This means the entire bit length must be at least 40µs (36µs for the 10th sample + 4µs for synchronization with the internal sampling clock).

Clock deviation causes

The causes which contribute to the total deviation are:

- D_{TRA}: Deviation due to transmitter error (Local oscillator error of the transmitter or the transmitter is transmitting at a different baud rate).
- D_{QUANT}: Error due to the baud rate quantization of the receiver.
- D_{REC}: Deviation of the local oscillator of the receiver: This deviation can occur during the reception of one complete SCI message assuming that the deviation has been compensated at the beginning of the message.
- D_{TCL}: Deviation due to the transmission line (generally due to the transceivers)

All the deviations of the system should be added and compared to the SCI clock tolerance:

$$D_{TRA} + D_{QUANT} + D_{REC} + D_{TCL} < 3.75\%$$

Noise error causes

See also description of noise error in *Receiver on page 140*.

Start bit

The noise flag (NF) is set during start bit reception if one of the following conditions occurs:

1. A valid falling edge is not detected. A falling edge is considered to be valid if the 3 consecutive samples before the falling edge occurs are detected as ‘1’ and, after the falling edge occurs, during the sampling of the 16 samples, if one of the samples numbered 3, 5 or 7 is detected as a ‘1’.
2. During sampling of the 16 samples, if one of the samples numbered 8, 9 or 10 is detected as a ‘1’.

Therefore, a valid Start Bit must satisfy both the above conditions to prevent the Noise Flag getting set.

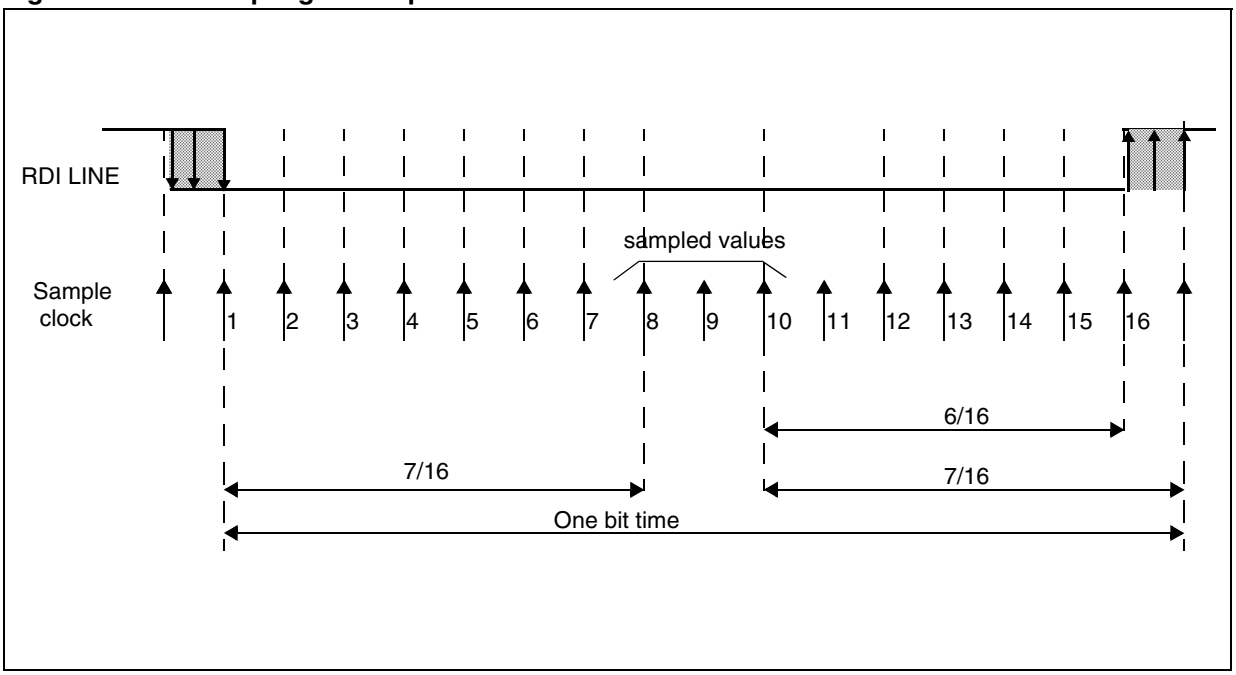
Data bits

The noise flag (NF) is set during normal data bit reception if the following condition occurs:

- During the sampling of 16 samples, if all three samples numbered 8, 9 and10 are not the same. The majority of the 8th, 9th and 10th samples is considered as the bit value.

Therefore, a valid Data Bit must have samples 8, 9 and 10 at the same value to prevent the Noise Flag from getting set.

Figure 65. Bit sampling in reception mode



15.5 Low power modes

Table 71. Effect of low power modes on SCI

Mode	Effect
Wait	No effect on SCI. SCI interrupts cause the device to exit from Wait mode.
Halt	SCI registers are frozen. In Halt mode, the SCI stops transmitting/receiving until Halt mode is exited.

15.6 Interrupts

The SCI interrupt events are connected to the same interrupt vector.

These events generate an interrupt if the corresponding Enable Control Bit is set and the interrupt mask in the CC register is reset (RIM instruction).

Table 72. SCI interrupt control/wake-up capability

Interrupt event	Event flag	Enable control bit	Exit from Wait	Exit from Halt
Transmit Data Register Empty	TDRE	TIE	Yes	No
Transmission Complete	TC	TCIE	Yes	No
Received Data Ready to be Read	RDRF	RIE	Yes	No
Overrun Error Detected	OR		Yes	No

Table 74. SCICR1 register description (continued)

Bit	Name	Function
0	PIE	*Parity interrupt enable* This bit enables the interrupt capability of the hardware parity control when a parity error is detected (PE bit set). It is set and cleared by software. 0: Parity error interrupt disabled 1: Parity error interrupt enabled

15.7.3 Control register 2 (SCICR2)

SCICR2 Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RW	RW	RW	RW	RW	RW	RW	RW

Table 75. SCICR2 register description

Bit	Name	Function
7	TIE	*Transmitter interrupt enable* This bit is set and cleared by software. 0: Interrupt is inhibited 1: An SCI interrupt is generated whenever TDRE = 1 in the SCISR register.
6	TCIE	*Transmission complete interrupt enable* This bit is set and cleared by software. 0: Interrupt is inhibited 1: An SCI interrupt is generated whenever TC = 1 in the SCISR register.
5	RIE	*Receiver interrupt enable* This bit is set and cleared by software. 0: Interrupt is inhibited 1: An SCI interrupt is generated whenever OR = 1 or RDRF = 1 in the SCISR register.
4	ILIE	*Idle line interrupt enable* This bit is set and cleared by software. 0: Interrupt is inhibited 1: An SCI interrupt is generated whenever IDLE = 1 in the SCISR register.
3	TE	*Transmitter enable* This bit enables the transmitter. It is set and cleared by software. 0: Transmitter is disabled 1: Transmitter is enabled *Notes:* *During transmission, a '0' pulse on the TE bit ('0' followed by '1') sends a preamble (idle line) after the current word.* *When TE is set there is a 1 bit-time delay before the transmission starts.* **Caution**: The TDO pin is free for general purpose I/O only when the TE and RE bits are both cleared (or if TE is never set).

Power consumption vs f_{CPU}: Flash devices

Figure 74. Typical I_{DD} in Run mode

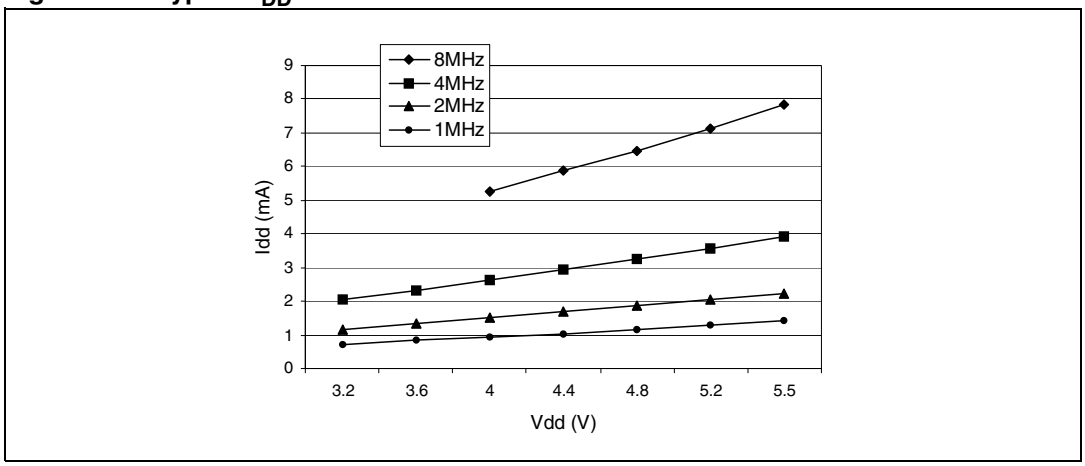


Figure 75. Typical I_{DD} in Slow mode

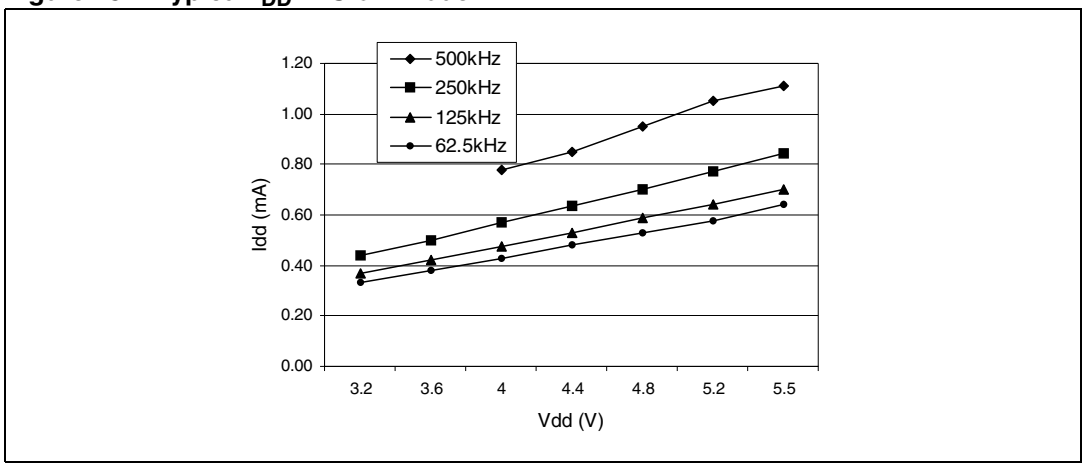


Figure 76. Typical I_{DD} in Wait mode

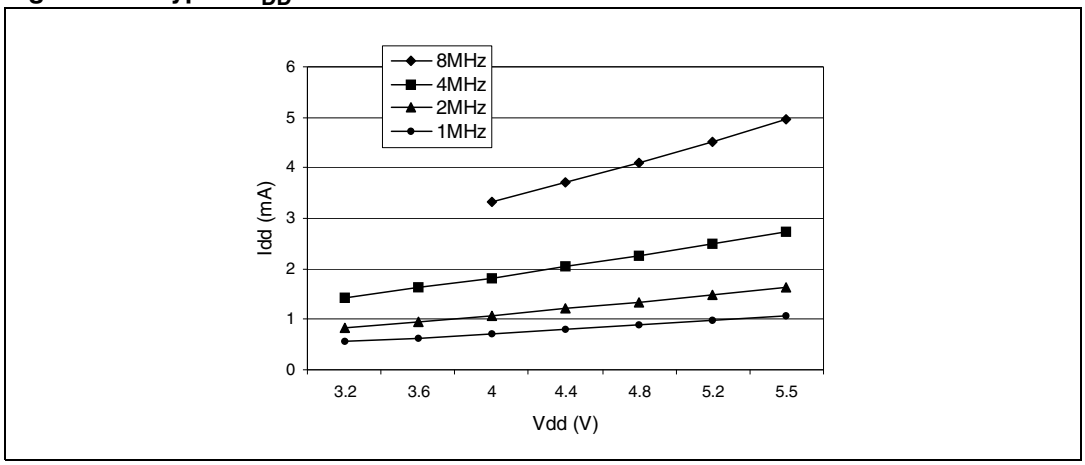


Figure 87. Typical V_{OL} versus V_{DD} (standard)

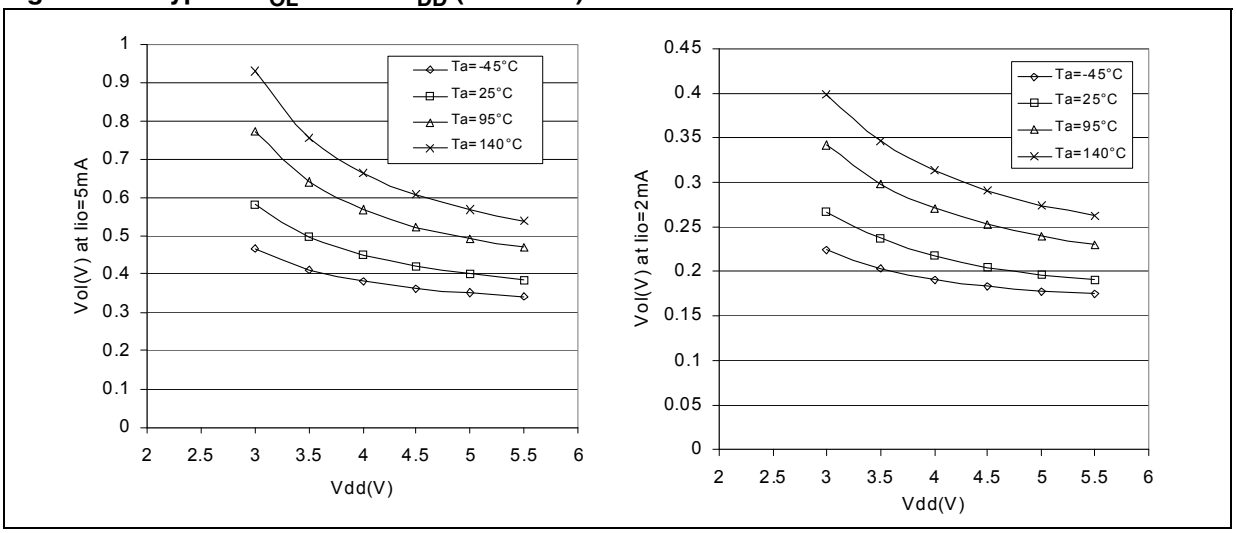


Figure 88. Typical V_{OL} versus V_{DD} (high-sink)

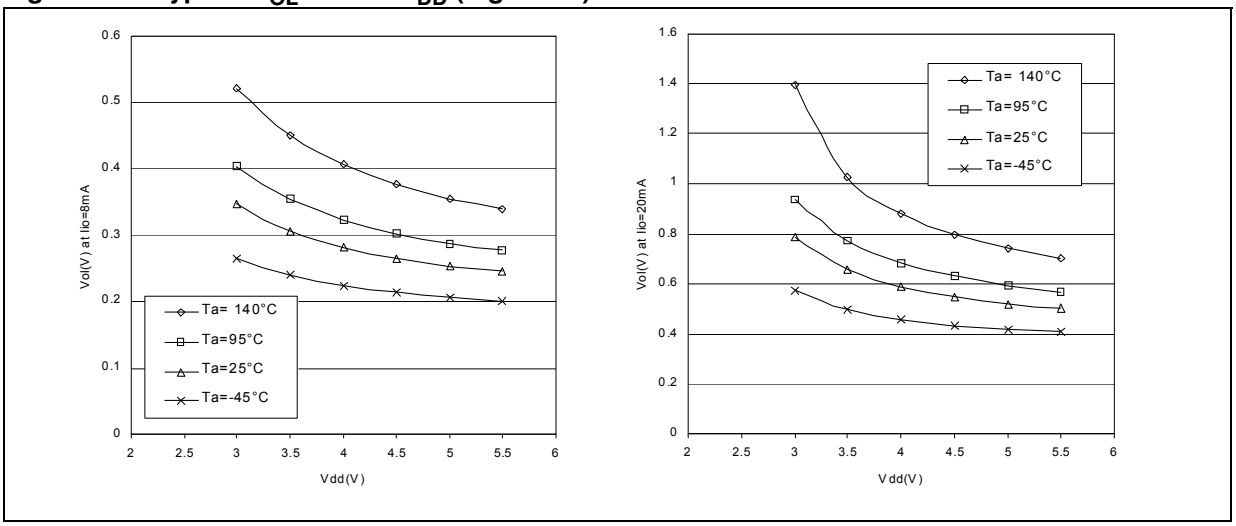
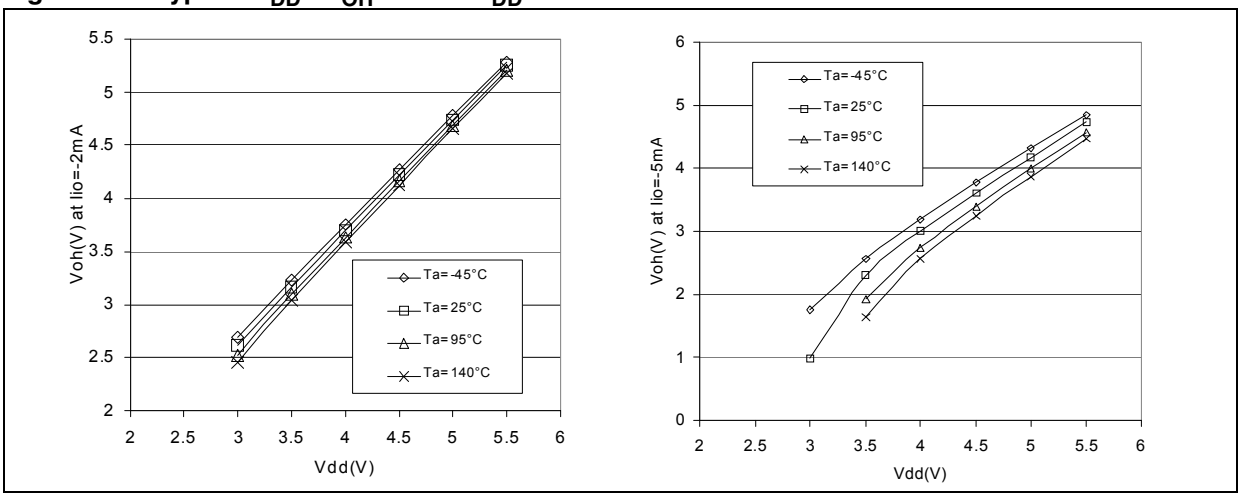


Figure 89. Typical V_{DD} - V_{OH} versus V_{DD}



19.12.3 ADC accuracy

Conditions: V_{DD} = 5V(1)

Table 136. ADC accuracy

Symbol	Parameter(1)	Conditions	Typ	Max(2)	Unit
$\|E_T\|$	Total unadjusted error	CPU in run mode @ f_{ADC} 2 MHz	3	4	LSB
$\|E_O\|$	Offset error		2	3	
$\|E_G\|$	Gain error		0.5	3	
$\|E_D\|$	Differential linearity error		1	2	
$\|E_L\|$	Integral linearity error				

1. ADC Accuracy versus Negative Injection Current: Injecting negative current may reduce the accuracy of the conversion being performed on another analog input.
Any positive injection current within the limits specified for $I_{INJ(PIN)}$ and $\Sigma I_{INJ(PIN)}$ in *Section 19.8* does not affect the ADC accuracy.
2. Data based on characterization results, monitored in production to guarantee 99.73% within ± max value from -40°C to 125°C (± 3σ distribution limits).

Figure 101. ADC error classification

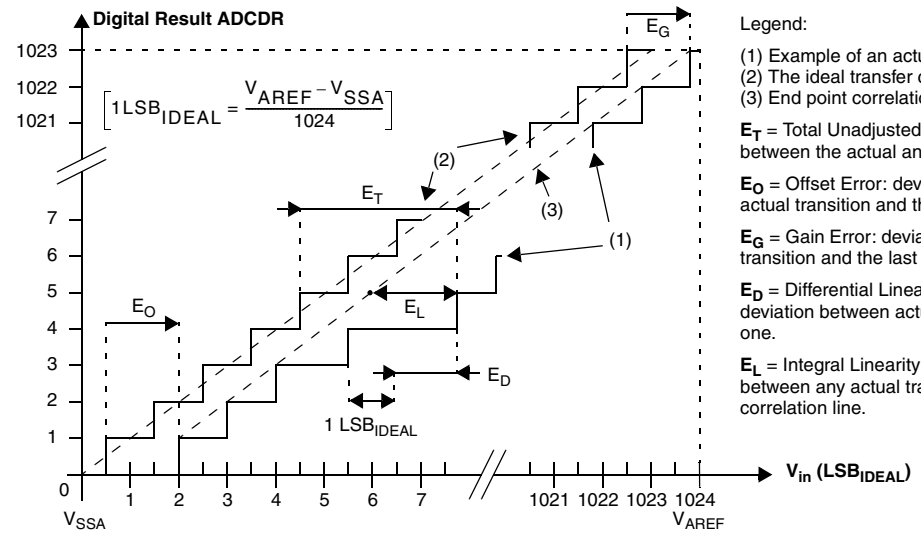


Figure 106. ST72P321xxx-Auto FastROM commercial product structure

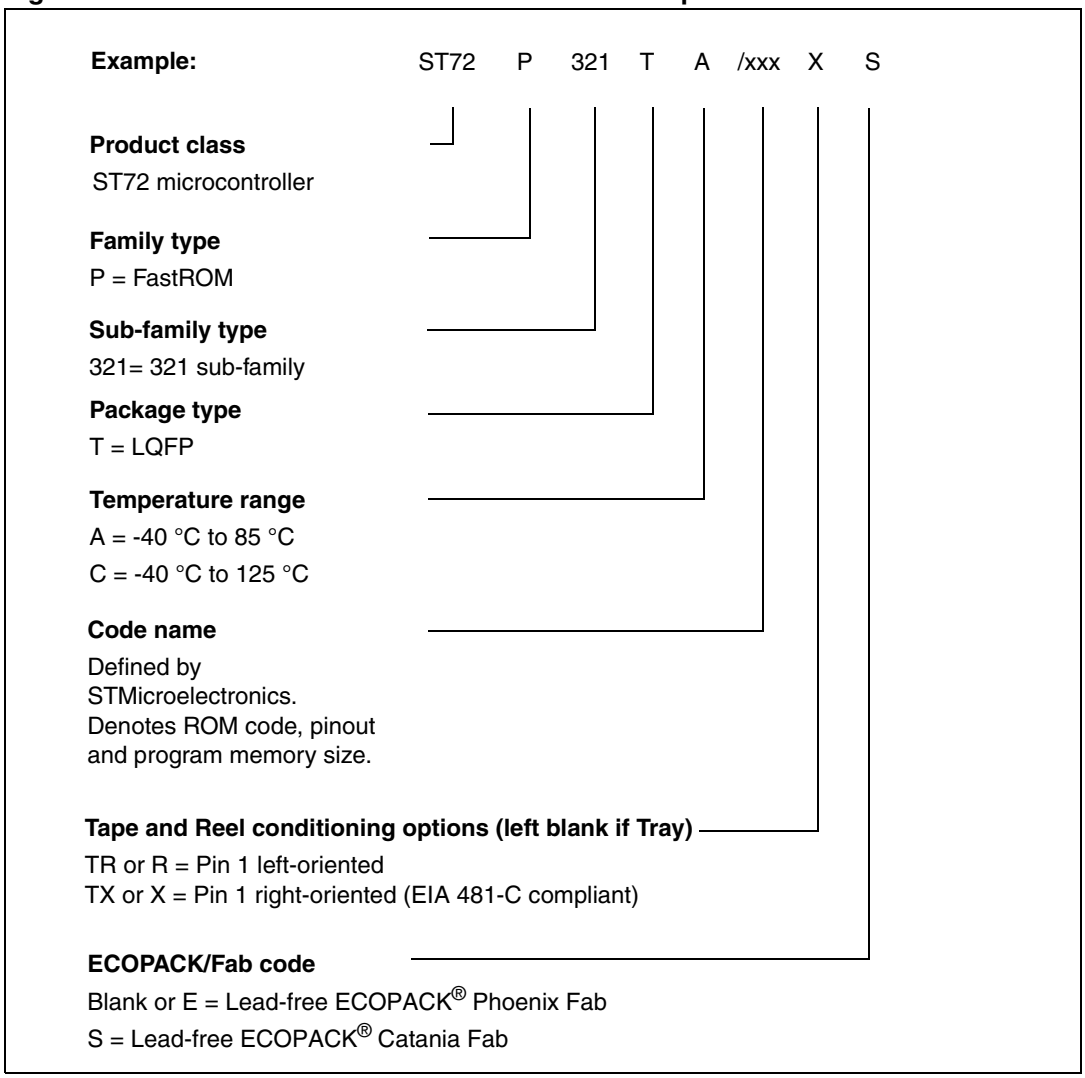


Table 144. STMicroelectronics development tools

Supported products	Emulation				Programming
	ST7 DVP3 series		ST7 EMU3 series		ICC socket board
	Emulator	Connection kit	Emulator	Active probe and T.E.B.	
ST72521M, ST72F521M	ST7MDT20-DVP3	ST7MDT20-T80/DVP	ST7MDT20M-EMU3	ST7MDT20M-TEB	ST7SB20M/xx(1)
ST72521R, ST72F521R		ST7MDT20-T64/DVP			
ST72521AR, ST72F521AR		ST7MDT20-T6A/DVP			
ST72321AR, ST72F321AR	ST7MDT20-DVP3	ST7MDT20-T6A/DVP	ST7MDT20M-EMU3	ST7MDT20M-TEB	ST7SB20M/xx(2)
ST72321R, ST72F321R		ST7MDT20-T64/DVP			
ST72321J, ST72F321J		ST7MDT20-T44/DVP	ST7MDT20J-EMU3	ST7MDT20J-TEB	ST7SB20J/xx(2)

1. Add suffix /EU, /UK, /US for the power supply of your region.
2. Add suffix /EU, /UK, /US for the power supply of your region.

Table 145. Suggested list of socket types

Device	Socket (supplied with ST7MDT20M-EMU3)	Emulator adapter (supplied with ST7MDT20M-EMU3)
LQFP80 14 X 14	YAMAICHI IC149-080-*51-*5	YAMAICHI ICP-080-7
LQFP64 14 x14	CAB 3303262	CAB 3303351
LQFP64 10 x10	YAMAICHI IC149-064-*75-*5	YAMAICHI ICP-064-6
LQFP44 10 x10	YAMAICHI IC149-044-*52-*5	YAMAICHI ICP-044-5

21.3.5 Socket and emulator adapter information

For information on the type of socket that is supplied with the emulator, refer to the suggested list of sockets in *Table 145*.

Note: Before designing the board layout, it is recommended to check the overall dimensions of the socket as they may be greater than the dimensions of the device.

For footprint and other mechanical information about these sockets and adapters, refer to the manufacturer's datasheet.

Related documentation

ST7 Visual Develop Software Key Debugging Features (AN 978)

ST7 Visual Develop for ST7 Cosmic C toolset users (AN 1938)

ST7 Visual Develop for ST7 Assembler Linker toolset users (AN 1940)

22.1.6 Clearing active interrupts outside interrupt routine

When an active interrupt request occurs at the same time as the related flag is being cleared, an unwanted reset may occur.

Note: Clearing the related interrupt mask will not generate an unwanted reset.

Concurrent interrupt context

The symptom does not occur when the interrupts are handled normally, that is, when:

- The interrupt flag is cleared within its own interrupt routine
- The interrupt flag is cleared within any interrupt routine
- The interrupt flag is cleared in any part of the code while this interrupt is disabled

If these conditions are not met, the symptom can be avoided by implementing the following sequence:

Perform SIM and RIM operation before and after resetting an active interrupt request.

Example:

```
SIM
Reset interrupt flag
RIM
```

Nested interrupt context

The symptom does not occur when the interrupts are handled normally, that is, when:

- The interrupt flag is cleared within its own interrupt routine
- The interrupt flag is cleared within any interrupt routine with higher or identical priority level
- The interrupt flag is cleared in any part of the code while this interrupt is disabled

If these conditions are not met, the symptom can be avoided by implementing the following sequence:

```
PUSH CC
SIM
Reset interrupt flag
POP CC
```