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	CANbus, I²C, SCI, SPI
Peripherals	LVD, POR, PWM, WDT
Number of I/O	64
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 ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	80-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/stmicroelectronics/st72f521m9tc

6.5.5 Internal watchdog RESET

The RESET sequence generated by an internal Watchdog counter overflow is shown in *Figure 13*.

Starting from the Watchdog counter underflow, the device $\overline{\text{RESET}}$ pin acts as an output that is pulled low during at least $t_{w(RSTL)out}$.

Figure 13. RESET sequences

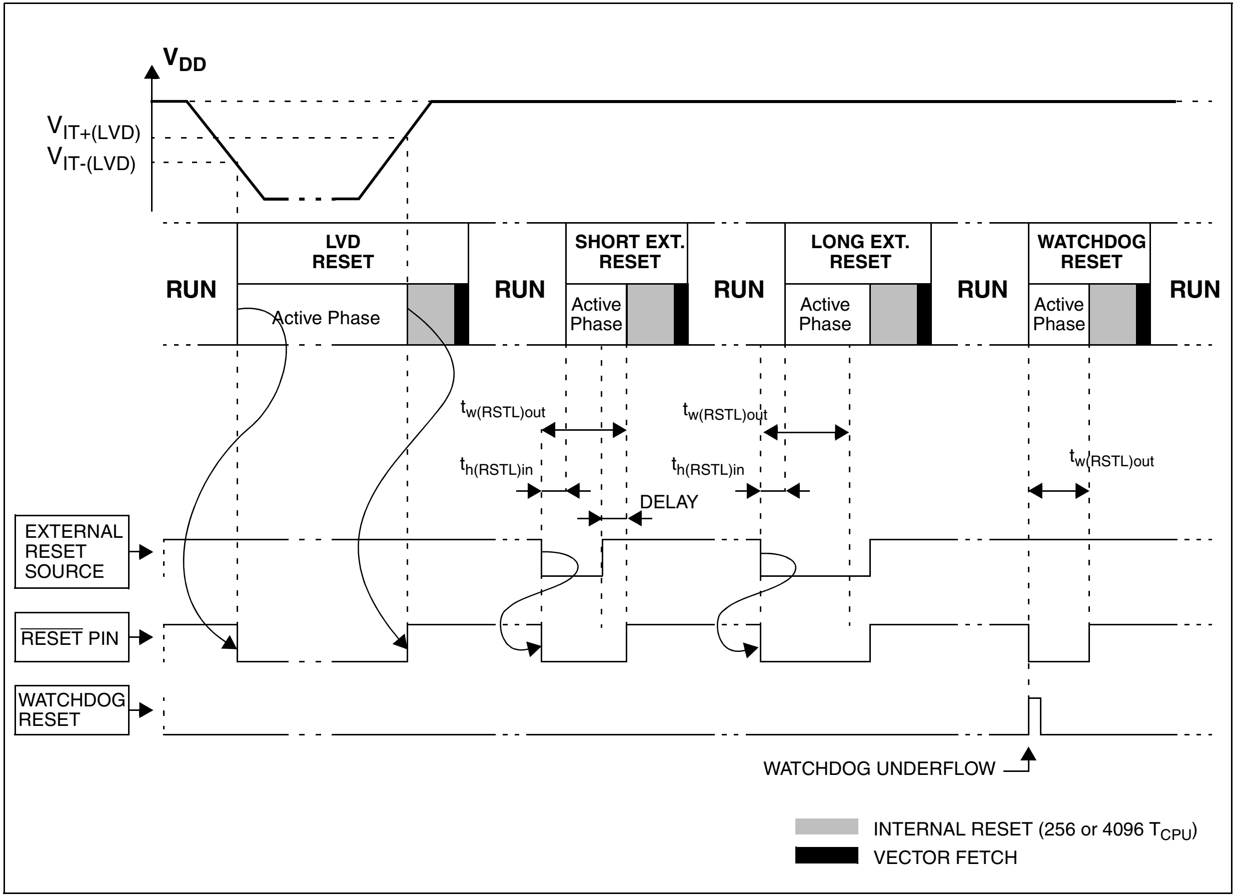
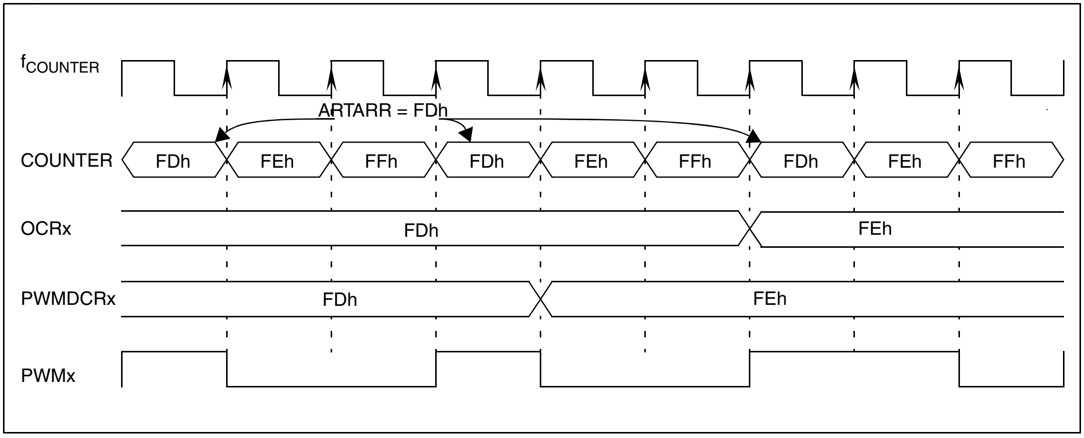


Table 26. Nested interrupts register map and reset values

Address (Hex.)	Register label	7	6	5	4	3	2	1	0
0024h	ISPR0 Reset value	ei1		ei0		MCC		TLI	
		I1_3 1	I0_3 1	I1_2 1	I0_2 1	I1_1 1	I0_1 1	1	1
0025h	ISPR1 Reset value	SPI		CAN		ei3		ei2	
		I1_7 1	I0_7 1	I1_6 1	I0_6 1	I1_5 1	I0_5 1	I1_4 1	I0_4 1
0026h	ISPR2 Reset value	AVD		SCI		TIMER B		TIMER A	
		I1_11 1	I0_11 1	I1_10 1	I0_10 1	I1_9 1	I0_9 1	I1_8 1	I0_8 1
0027h	ISPR3 Reset value	1	1	1	1	PWMART		I2C	
						I1_13 1	I0_13 1	I1_12 1	I0_12 1
0028h	EICR Reset value	IS11 0	IS10 0	IPB 0	IS21 0	IS20 0	IPA 0	TLIS 0	TLIE 0

Figure 36. Output compare control



12.2.5 Independent PWM signal generation

This mode allows up to four Pulse Width Modulated signals to be generated on the PWMx output pins with minimum core processing overhead. This function is stopped during Halt mode.

Each PWMx output signal can be selected independently using the corresponding OEx bit in the PWM Control register (PWMCR). When this bit is set, the corresponding I/O pin is configured as output push-pull alternate function.

The PWM signals all have the same frequency which is controlled by the counter period and the ARTARR register value.

$$f_{PWM} = f_{COUNTER} / (256 - ARTARR)$$

When a counter overflow occurs, the PWMx pin level is changed depending on the corresponding OPx (output polarity) bit in the PWMCR register. When the counter reaches the value contained in one of the output compare register (OCRx) the corresponding PWMx pin level is restored.

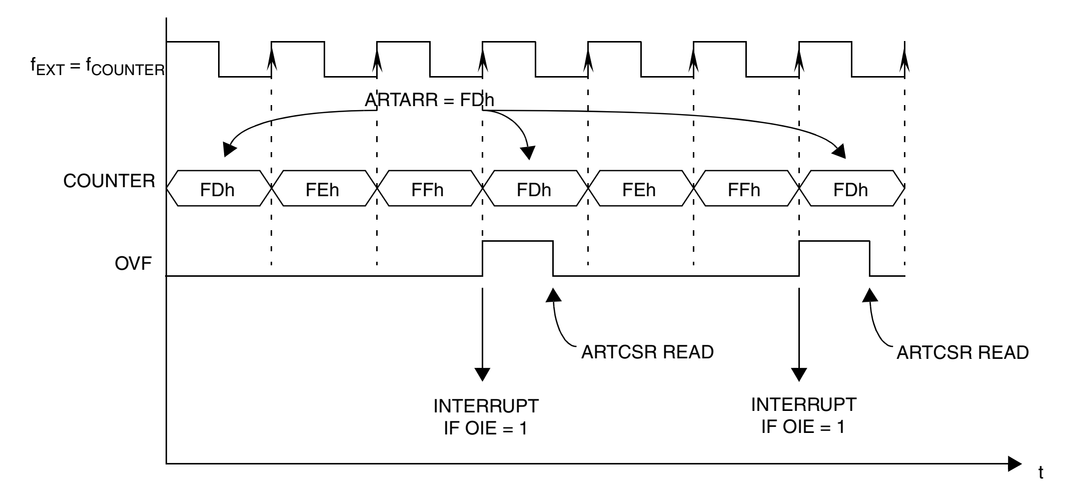
It should be noted that the reload values will also affect the value and the resolution of the duty cycle of the PWM output signal. To obtain a signal on a PWMx pin, the contents of the OCRx register must be greater than the contents of the ARTARR register.

The maximum available resolution for the PWMx duty cycle is:

$$\text{Resolution} = 1 / (256 - ARTARR)$$

Note: To get the maximum resolution (1/256), the ARTARR register must be 0. With this maximum resolution, 0% and 100% can be obtained by changing the polarity.

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.

13.3.2 External clock

The external clock (where available) is selected if CC0 = 1 and CC1 = 1 in the CR2 register.

The status of the EXEDG bit in the CR2 register determines the type of level transition on the external clock pin EXTCLK that will trigger the free running counter.

The counter is synchronized with the falling edge of the internal CPU clock.

A minimum of four falling edges of the CPU clock must occur between two consecutive active edges of the external clock; thus, the external clock frequency must be less than a quarter of the CPU clock frequency.

Figure 43. Counter timing diagram, internal clock divided by 2

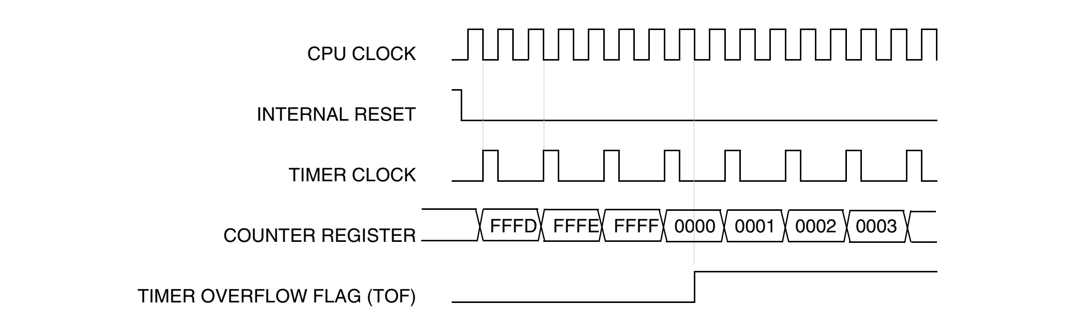


Figure 44. Counter timing diagram, internal clock divided by 4

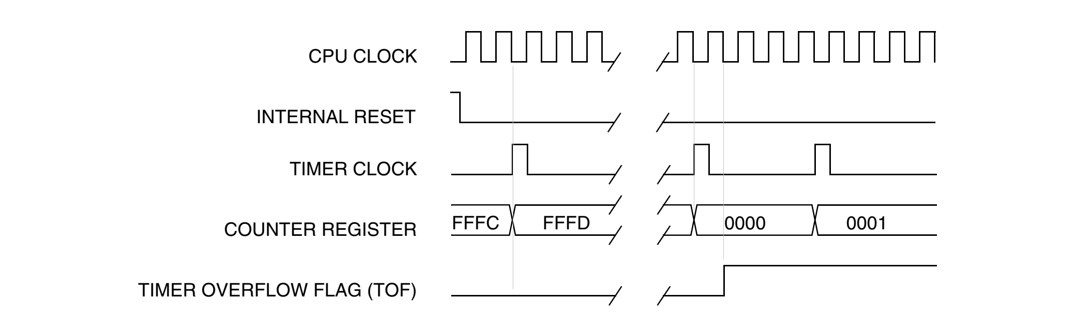
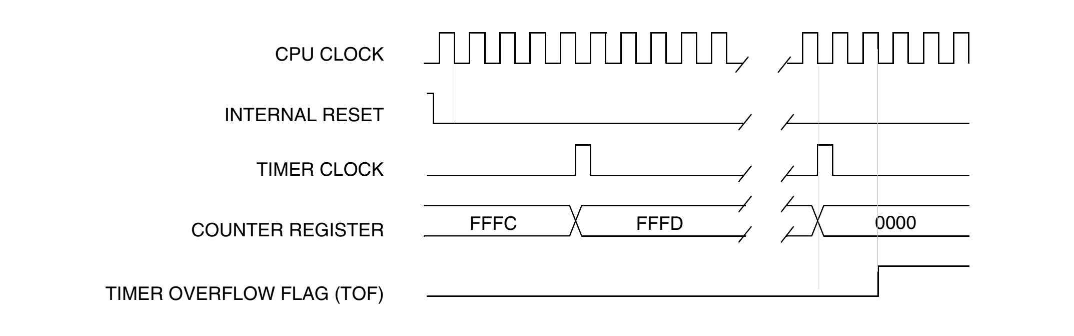


Figure 45. Counter timing diagram, internal clock divided by 8



Note: The MCU is in reset state when the internal reset signal is high; when it is low the MCU is running.

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

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

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:

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

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

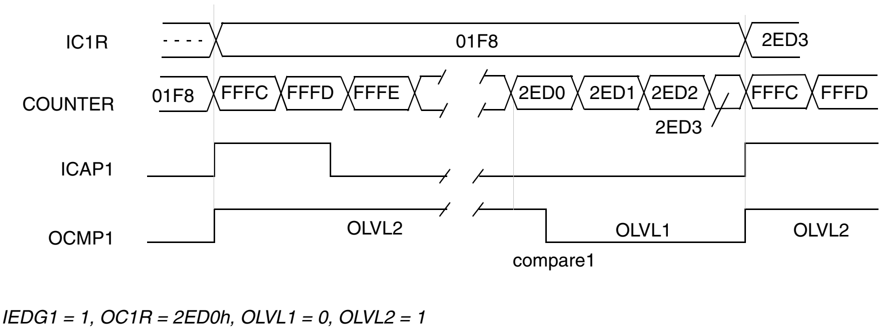


Table 58. Timer modes

Modes	Timer resources			
	Input Capture 1	Input Capture 2	Output Compare 1	Output Compare 2
One Pulse mode	No	Not recommended[1]	No	Partially[2]
PWM mode		Not recommended[3]		No

1. See *Note 4* in *Section 13.3.6 One Pulse mode*
2. See *Note 5* in *Section 13.3.6 One Pulse mode*
3. See *Note 4* in *Section 13.3.7 Pulse width modulation mode*

13.7 16-bit timer registers

Each timer is associated with 3 control and status registers, and with 6 pairs of data registers (16-bit values) relating to the 2 input captures, the 2 output compares, the counter and the alternate counter.

13.7.1 Control register 1 (CR1)

CR1 Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
ICIE	OCIE	TOIE	FOLV2	FOLV1	OLVL2	IEDG1	OLVL1
RW	RW	RW	RW	RW	RW	RW	RW

Table 59. CR1 register description

Bit	Name	Function
7	ICIE	*Input Capture Interrupt Enable* 0: Interrupt is inhibited 1: A timer interrupt is generated whenever the ICF1 or ICF2 bit of the SR register is set.
6	OCIE	*Output Compare Interrupt Enable* 0: Interrupt is inhibited 1: A timer interrupt is generated whenever the OCF1 or OCF2 bit of the SR register is set.
5	TOIE	*Timer Overflow Interrupt Enable* 0: Interrupt is inhibited 1: A timer interrupt is enabled whenever the TOF bit of the SR register is set.
4	FOLV2	*Forced Output Compare 2* This bit is set and cleared by software. 0: No effect on the OCMP2 pin 1: Forces the OLVL2 bit to be copied to the OCMP2 pin, if the OC2E bit is set and even if there is no successful comparison

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	-	

Error cases

- **BERR**: Detection of a Stop or a Start condition during a byte transfer. In this case, the EVF and BERR bits are set by hardware with an interrupt if ITE is set.
  Note that BERR will not be set if an error is detected during the first or second pulse of each 9-bit transaction:
  - ***Single Master Mode***
    If a Start or Stop is issued during the first or second pulse of a 9-bit transaction, the BERR flag will not be set and transfer will continue however the BUSY flag will be reset. To work around this, slave devices should issue a NACK when they receive a misplaced Start or Stop. The reception of a NACK or BUSY by the master in the middle of communication makes it possible to re-initiate transmission.
  - ***Multimaster Mode***
    Normally the BERR bit would be set whenever unauthorized transmission takes place while transfer is already in progress. However, an issue will arise if an external master generates an unauthorized Start or Stop while the I^{2}C master is on the first or second pulse of a 9-bit transaction. It is possible to work around this by polling the BUSY bit during I^{2}C master mode transmission. The resetting of the BUSY bit can then be handled in a similar manner as the BERR flag being set.
- **AF**: Detection of a non-acknowledge bit. In this case, the EVF and AF bits are set by hardware with an interrupt if the ITE bit is set. To resume, set the Start or Stop bit.
  The AF bit is cleared by reading the I2CSR2 register. However, if read before the completion of the transmission, the AF flag will be set again, thus possibly generating a new interrupt. Software must ensure either that the SCL line is back at 0 before reading the SR2 register, or be able to correctly handle a second interrupt during the 9th pulse of a transmitted byte.
- **ARLO:** Detection of an arbitration lost condition.
  In this case the ARLO bit is set by hardware (with an interrupt if the ITE bit is set and the interface goes automatically back to slave mode (the M/SL bit is cleared).

Note: *In all these cases, the SCL line is not held low; however, the SDA line can remain low due to possible ‘0’ bits transmitted last. It is then necessary to release both lines by software.*

Identifier low registers (IDLRx)

IDLRx Reset value: Undefined

7	6	5	4	3	2	1	0
ID[2:0]			RTR	DLC[3:0]			
R/W			R/W	R/W			

Table 102. IDLRx register description

Bit	Name	Function
7:5	ID[2:0]	*Message Identifier (LSB)* These are the least significant 3 bits of the 11-bit message identifier.
4	RTR	*Remote Transmission Request* This bit is set to indicate a remote frame and reset to indicate a data frame.
3:0	DLC[3:0]	*Data Length Code* It gives the number of bytes in the data field of the message. The valid range is 0 to 8.

Data registers (DATA0-7x)

DATA0-7x Reset value: Undefined

7	6	5	4	3	2	1	0
DATA[7:0]							
R/W							

Table 103. DATA0-7x register description

Bit	Name	Function
7:0	DATA[7:0]	*Message Data* DATA[7:0] is a message data byte. Up to eight such bytes may be part of a message. Writing to byte DATA7 initiates a transmit request and should always be done even when DATA7 is not part of the message.

17.5.3 Unexpected message transmission

Symptom

The previous message received by pCAN, even if this message did not pass the receive filter, will be retransmitted by pCAN with a correct identifier and DLC but with corrupted data. The data bytes will be a copy of the identifier bytes IDHR and IDLR in the following repetitive pattern:

DATA_0 = IDHR
DATA_1 = IDLR
DATA_2 = IDHR
DATA_3 = IDLR
etc.
DATA_7 = IDLR

If no message has been received before the problem occurs then identifier byte values are random but the data bytes are in the same repetitive pattern.

Details

The buffers of the pCAN cell are configurable as receive or transmit buffers. By default, all buffers are configured in reception. To use a buffer to transmit a CAN message the application has to reserve this buffer for transmission by setting the LOCK bit in the BCSR register. So the buffer is then locked for any further reception and reserved for transmission.

Once a transmission has been requested by a write access to data byte 7 of the buffer the application might need to abort this transmission request. To do so, the application can reset the LOCK bit in the BCSR register.

If the message is pending (RDY bit set) but not currently being transmitted, then clearing the LOCK bit will abort it immediately.

If the message is pending (RDY bit set) and currently being transmitted then the message will not be interrupted but the CAN core will wait until the end of this transmission attempt. Then software must clear the LOCK bit again to abort the transmission.

An unexpected transmission can occur:

IF the application resets the LOCK bit

WHILE the CAN core is preparing the transmission[a] **AND** there is no other transmission pending in another buffer

THEN the LOCK bit is reset but the transmission is not stopped. Instead the content of the page 0 buffer will be transmitted.

Impact on the application

pCAN will echo some messages sent by other nodes. Identifier and DLC will be correct but data are corrupted as described previously.

[a] The preparation lasts two bit times just before SOF; this is the **critical window** during which the LOCK bit must not be reset by the application.

section, the application must monitor the BUSY bit in the BCSR register and reset the LOCK bit just after the falling edge of the BUSY bit. The time between the falling edge of the BUSY bit and the SOF of the next transmission attempt is in any case long enough to guarantee that the LOCK bit is reset before the critical time window.

The “C” code sequence below shows the software workaround for both the error and arbitration lost cases.

```
_asm("SIM\n"); // Mask interrupts
CANCSR |= NRTX;// Set non automatic retransmission bit
while(!(CANBCSR & BUSY) &&// Wait till BUSY bit is set
     (CANBCSR & RDY) ); // or transmission done
while( CANBCSR & BUSY ); // Wait till BUSY bit is reset (falling
edge)
if( CANBCSR & RDY )
{ // transmission still pending -> must be aborted
  CANBCSR &= ~LOCK; //Arbitration lost => cancel transmission
safely
  while( CANBCSR & RDY );// Wait for unlock confirmed
  CANCSR &= ~NRTX;// Reset NRTX bit once abort sequence done
  _asm("RIM\n");
}
else
{ // No more abort required as RDY bit already reset
  CANCSR &= ~NRTX;// Reset NRTX bit once abort sequence done
  _asm("RIM\n"); // Enable interrupts
}
```

18.3.3 Changing the conversion channel

The application can change channels during conversion. When software modifies the CH[3:0] bits in the ADCCSR register, the current conversion is stopped, the EOC bit is cleared, and the A/D converter starts converting the newly selected channel.

18.4 Low power modes

Note: The A/D converter may be disabled by resetting the ADON bit. This feature allows reduced power consumption when no conversion is needed and between single shot conversions.

Table 111. Effect of low power modes on ADC

Mode	Effect
Wait	No effect on A/D converter
Halt	A/D converter disabled. After wake-up from Halt mode, the A/D converter requires a stabilization time t_{STAB} (see *Section 20: Electrical characteristics*) before accurate conversions can be performed.

18.5 Interrupts

None.

18.6 ADC registers

18.6.1 Control/status register (ADCCSR)

ADCCSR Reset value: 0000 0000 (00h)

7	6	5	4	3	2	1	0
EOC	SPEED	ADON	Reserved	CH[3:0]			
RO	RW	RW	-	RW			

Table 112. ADCCSR register description

Bit	Name	Function
7	EOC	*End of Conversion* This bit is set by hardware. It is cleared by hardware when software reads the ADCDRH register or writes to any bit of the ADCCSR register. 0: Conversion is not complete 1: Conversion complete
6	SPEED	*ADC clock selection* This bit is set and cleared by software. 0: $f_{ADC} = f_{CPU}/4$ 1: $f_{ADC} = f_{CPU}/2$

20.5.3 Crystal and ceramic resonator oscillators

The ST7 internal clock can be supplied with four different crystal/ceramic resonator oscillators. All the information given in this paragraph is based on characterization results with specified typical external components. In the application, the resonator and the load capacitors have to be placed as close as possible to the oscillator pins in order to minimize output distortion and start-up stabilization time. Refer to the crystal/ceramic resonator manufacturer for more details (such as frequency, package or accuracy).

Table 136. Crystal and ceramic resonator oscillators

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
f_{OSC}	Oscillator frequency[1]	LP: Low power oscillator MP: Medium power oscillator MS: Medium speed oscillator HS: High speed oscillator		1 >2 >4 >8	-	2 4 8 16	MHz
R_F	Feedback resistor[2]	-		20	-	40	kΩ
C_{L1} C_{L2}	Recommended load capacitance versus equivalent serial resistance of the crystal or ceramic resonator (R_S)[3]	$R_S = 200\Omega$ $R_S = 200\Omega$ $R_S = 200\Omega$ $R_S = 100\Omega$	LP oscillator MP oscillator MS oscillator HS oscillator	22 22 18 15	-	56 46 33 33	pF
i_2	OSC2 driving current	$V_{DD} = 5V$, $V_{IN} = V_{SS}$	LP oscillator MP oscillator MS oscillator HS oscillator	-	80 160 310 610	150 250 460 910	µA

1. The oscillator selection can be optimized in terms of supply current using a high-quality resonator with small R_S value. Refer to crystal/ceramic resonator manufacturer for more details.

2. Data based on characterization results, not tested in production. The relatively low value of the RF resistor offers a good protection against issues resulting from use in a humid environment, due to the induced leakage and the bias condition change. However, it is recommended to take this point into account if the microcontroller is used in tough humidity conditions.

3. For C_{L1} and C_{L2} it is recommended to use high-quality ceramic capacitors in the 5pF to 25pF range (typ.) designed for high-frequency applications and selected to match the requirements of the crystal or resonator. C_{L1} and C_{L2}, are usually the same size. The crystal manufacturer typically specifies a load capacitance which is the series combination of C_{L1} and C_{L2}. PCB and MCU pin capacitance must be included when sizing C_{L1} and C_{L2} (10pF can be used as a rough estimate of the combined pin and board capacitance).

Figure 94. Typical application with a crystal or ceramic resonator

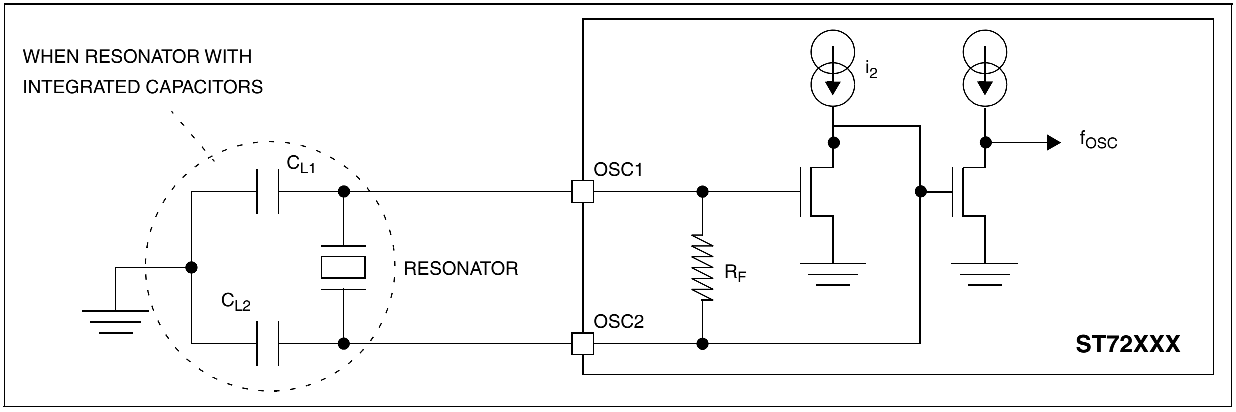


Table 137. OSCRANGE selection for typical resonators

Supplier	f_{OSC} (MHz)	Typical ceramic resonators[1]	
		Reference	Recommended OSCRANGE option bit configuration
Murata	2	CSTCC2M00G56A-R0	MP mode[2]
	4	CSTCR4M00G55B-R0	MS mode
	8	CSTCE8M00G55A-R0	HS mode
	16	CSTCE16M0G53A-R0	

1. Resonator characteristics given by the ceramic resonator manufacturer. For more information on these resonators, please consult www.murata.com.

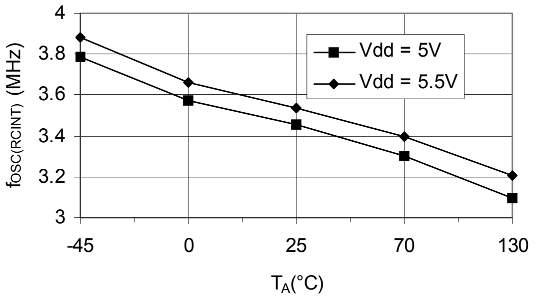
2. LP mode is not recommended for 2 MHz resonator because the peak to peak amplitude is too small (> 0.8V).

20.5.4 RC oscillators

Table 138. RC oscillator characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
$f_{OSC(RCINT)}$	Internal RC oscillator frequency (see *Figure 95*)	T_A = 25°C, V_{DD} = 5V	2	3.5	5.6	MHz

Figure 95. Typical $f_{OSC(RCINT)}$ versus T_A



Note: To reduce disturbance to the RC oscillator, it is recommended to place decoupling capacitors between V_{DD} and V_{SS} as shown in Figure 115.

20.7.3 Absolute maximum ratings (electrical sensitivity)

Based on two different tests (ESD and LU) using specific measurement methods, the product is stressed in order to determine its performance in terms of electrical sensitivity.

Electrostatic discharge (ESD)

Electrostatic discharges (a positive then a negative pulse separated by 1 second) are applied to the pins of each sample according to each pin combination. The sample size depends on the number of supply pins in the device (3 parts*(n+1) supply pin). This test conforms to the AEC-Q100-002/-003/-011 standard. For more details, refer to the application note AN1181.

Table 144. ESD absolute maximum ratings

Symbol	Ratings	Conditions	Class	Max. value[1]	Unit
$V_{ESD(HBM)}$	Electrostatic discharge voltage (Human Body Model)	T_A = +25°C conforming to AEC-Q100-002	H1C	2000	V
$V_{ESD(MM)}$	Electrostatic discharge voltage (Machine Model)	T_A = +25°C conforming to AEC-Q100-003	M2	200	

[1] Data based on characterization results, not tested in production.

Static latch-up (LU)

Two complementary static tests are required on six parts to assess the latch-up performance:

- A supply overvoltage is applied to each power supply pin.
- A current injection is applied to each input, output and configurable I/O pin.

These tests are compliant with the EIA/JESD 78 IC latch-up standard.

Table 145. Electrical sensitivities

Symbol	Parameter	Conditions	Class
LU	Static latch-up class	T_A = +25°C T_A = +85°C T_A = +125°C conforming to JESD 78	A

20.12.2 General PCB design guidelines

To obtain best results, some general design and layout rules should be followed when designing the application PCB to shield the noise-sensitive, analog physical interface from noise-generating CMOS logic signals.

- Use separate digital and analog planes. The analog ground plane should be connected to the digital ground plane via a single point on the PCB.
- Filter power to the analog power planes. It is recommended to connect capacitors, with good high frequency characteristics, between the power and ground lines, placing 0.1µF and optionally, if needed 10pF capacitors as close as possible to the ST7 power supply pins and a 1 to 10µF capacitor close to the power source (see *Figure 115*).
- The analog and digital power supplies should be connected in a star network. Do not use a resistor, as V_{AREF} is used as a reference voltage by the A/D converter and any resistance would cause a voltage drop and a loss of accuracy.
- Properly place components and route the signal traces on the PCB to shield the analog inputs. Analog signals paths should run over the analog ground plane and be as short as possible. Isolate analog signals from digital signals that may switch while the analog inputs are being sampled by the A/D converter. Do not toggle digital outputs on the same I/O port as the A/D input being converted.

Figure 115. Power supply filtering

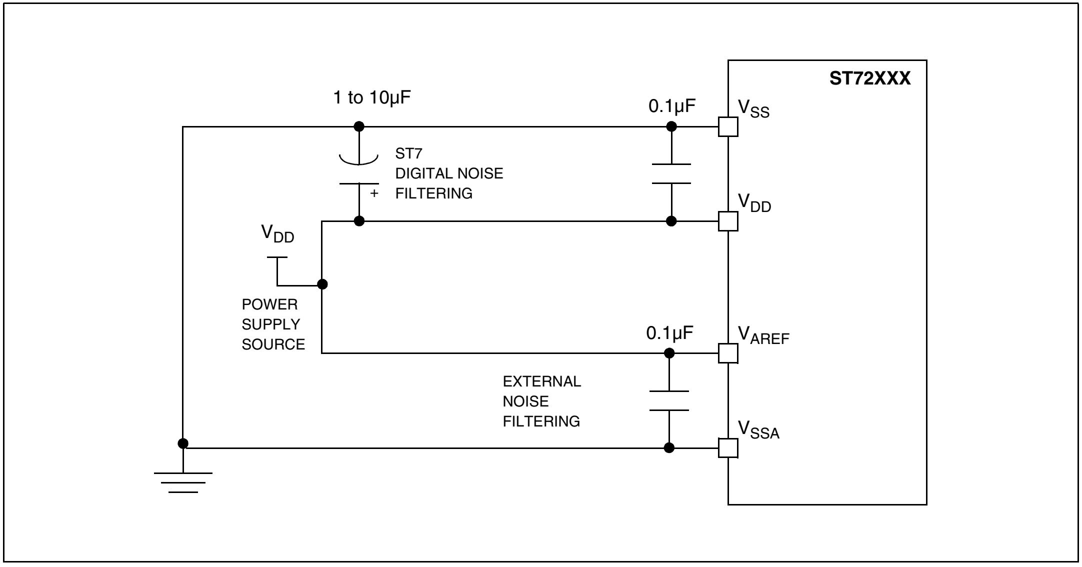


Table 164. Option byte 1 bit description (continued)

Bit	Name	Function
OPT5:4	OSCTYPE[1:0]	Oscillator type These option bits select the ST7 main clock source type. 00: Clock source = Resonator oscillator 01: Reserved 10: Clock source = Internal RC oscillator 11: Clock source = External source
OPT3:1	OSCRANGE[2:0]	Oscillator range When the resonator oscillator type is selected, these option bits select the resonator oscillator current source corresponding to the frequency range of the used resonator. Otherwise, these bits are used to select the normal operating frequency range (see *Table 166: Oscillator frequency range selection (OPT3:1)*).
OPT0	PLLOFF	PLL activation This option bit activates the PLL which allows multiplication by two of the main input clock frequency. The PLL must not be used with the internal RC oscillator or with external clock source. The PLL is guaranteed only with an input frequency between 2 and 4 MHz. 0: PLL x2 enabled 1: PLL x2 disabled **Caution**: The PLL can be enabled only if the “OSCRANGE” (OPT3:1) bits are configured to “MP - 2~4 MHz”. Otherwise, the device functionality is not guaranteed.

Table 165. Package selection (OPT7)

Version	Selected package	PKG1	PKG0
M	LQFP80	1	1
(A)R	LQFP64	1	0

Note: *On the chip, each I/O port has eight pads. Pads that are not bonded to external pins are in input pull-up configuration after reset. The configuration of these pads must be kept at reset state to avoid added current consumption.*

Table 166. Oscillator frequency range selection (OPT3:1)

Typical frequency range		OSCRANGE		
		2	1	0
LP	1~2 MHz	0	0	0
MP	2~4 MHz	0	0	1
MS	4~8 MHz	0	1	0
HS	8~16 MHz	0	1	1

22.3 Development tools

22.3.1 Introduction

Development tools for the ST7 microcontrollers include a complete range of hardware systems and software tools from STMicroelectronics and third-party tool suppliers. The range of tools includes solutions to help you evaluate microcontroller peripherals, develop and debug your application, and program your microcontrollers.

22.3.2 Evaluation tools and starter kits

ST offers complete, affordable starter kits and full-featured evaluation boards that allow you to evaluate microcontroller features and quickly start developing ST7 applications. Starter kits are complete, affordable hardware/software tool packages that include features and samples to help you quickly start developing your application. ST evaluation boards are open-design, embedded systems, which are developed and documented to serve as references for your application design. They include sample application software to help you demonstrate, learn about and implement your ST7's features.

22.3.3 Development and debugging tools

Application development for ST7 is supported by fully optimizing C Compilers and the ST7 Assembler-Linker toolchain, which are all seamlessly integrated in the ST7 integrated development environments in order to facilitate the debugging and fine-tuning of your application. The Cosmic C Compiler is available in a free version that outputs up to 16 Kbytes of code.

The range of hardware tools includes cost effective ST7-DVP3 series emulators. These tools are supported by the ST7 Toolset from STMicroelectronics, which includes the STVD7 integrated development environment (IDE) with high-level language debugger, editor, project manager and integrated programming interface.

22.3.4 Programming tools

During the development cycle, the ST7-DVP3 and ST7-EMU3 series emulators and the RLink provide in-circuit programming capability for programming the Flash microcontroller on your application board.

ST also provides a low-cost dedicated in-circuit programmer, the ST7-STICK, as well as ST7 socket boards which provide all the sockets required for programming any of the devices in a specific ST7 subfamily on a platform that can be used with any tool with in-circuit programming capability for ST7.

For production programming of ST7 devices, ST's third-party tool partners also provide a complete range of gang and automated programming solutions, which are ready to integrate into your production environment.

For additional ordering codes for spare parts, accessories and tools available for the ST7 (including from third party manufacturers), refer to the online product selector at www.st.com/mcu.

23.1.6 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
```