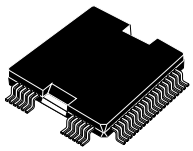
L9805E

Super smart power motor driver with 8-Bit MCU, RAM, EEPROM, ADC, WDG, Timers, PWM and H-bridge driver

Features

- 6.4-18V supply operating range
- 16 MHz maximum oscillator frequency
- 8 MHz maximum internal clock frequency
- Oscillator supervisor
- Fully static operation
- -40°C to + 150°C temperature range
- User EPROM/OTP: 16 Kbytes
- Data RAM: 256 bytes
- Data EEPROM: 128 bytes
- 64 pin HiQUAD64 package
- 10 multifunctional bidirectional I/O lines
- Two 16-bit Timers, each featuring:
  - 2 input captures
  - 2 output compares
  - External clock input (on Timer 1)
  - PWM and pulse generator modes
- Two programmable 16-bit PWM generator modules.
- K line transceiver
- CAN peripheral including bus line interface according 2A/B passive specifications



HiQUAD64

- 10-bit analog-to-digital converter
- Software watchdog for system integrity
- Master reset, power-on reset, low voltage reset
- 90mΩ DMOS H-bridge.
- 8-bit data manipulation
- 63 basic Instructions and 17 main addressing modes
- 8 x 8 unsigned multiply instruction
- True bit manipulation
- Complete development support on DOS/WINDOWS™ Real-Time Emulator
- Full software package on DOS/WINDOWS™ (C-Compiler, Cross-Assembler, Debugger).

Table 1. Device summary

Part number	Package	Packing
L9805E	HiQUAD64	Tray

Contents

List of tables

List of figures

1 General description

1.1 Introduction

The L9805E is a Super Smart Power device suited to drive resistive and inductive loads under software control. It includes a ST7 microcontroller and some peripherals. The microcontroller can execute the software contained in the program EPROM/ROM and drive, through dedicated registers, the power bridge.

The internal voltage regulators rated to the automotive environment, PWM modules, CAN transceiver and controller ISO9411 transceiver, timers, temperature sensor and the AID converter allow the device to realize by itself a complete application, in line with the most common mechatronic requirements.

1.2 OTP, ROM and EPROM devices

For development purposes the device is available in plastic HiQuad package without window rating in the OTP class.

Mass production is supported by means of ROM devices.

Engineering samples could be assembled using window packages. These are generally referenced as “EPROM devices”.

EPROM devices are erased by exposure to high intensity UV light admitted through the transparent window. This exposure discharges the floating gate to its initial state through induced photo current.

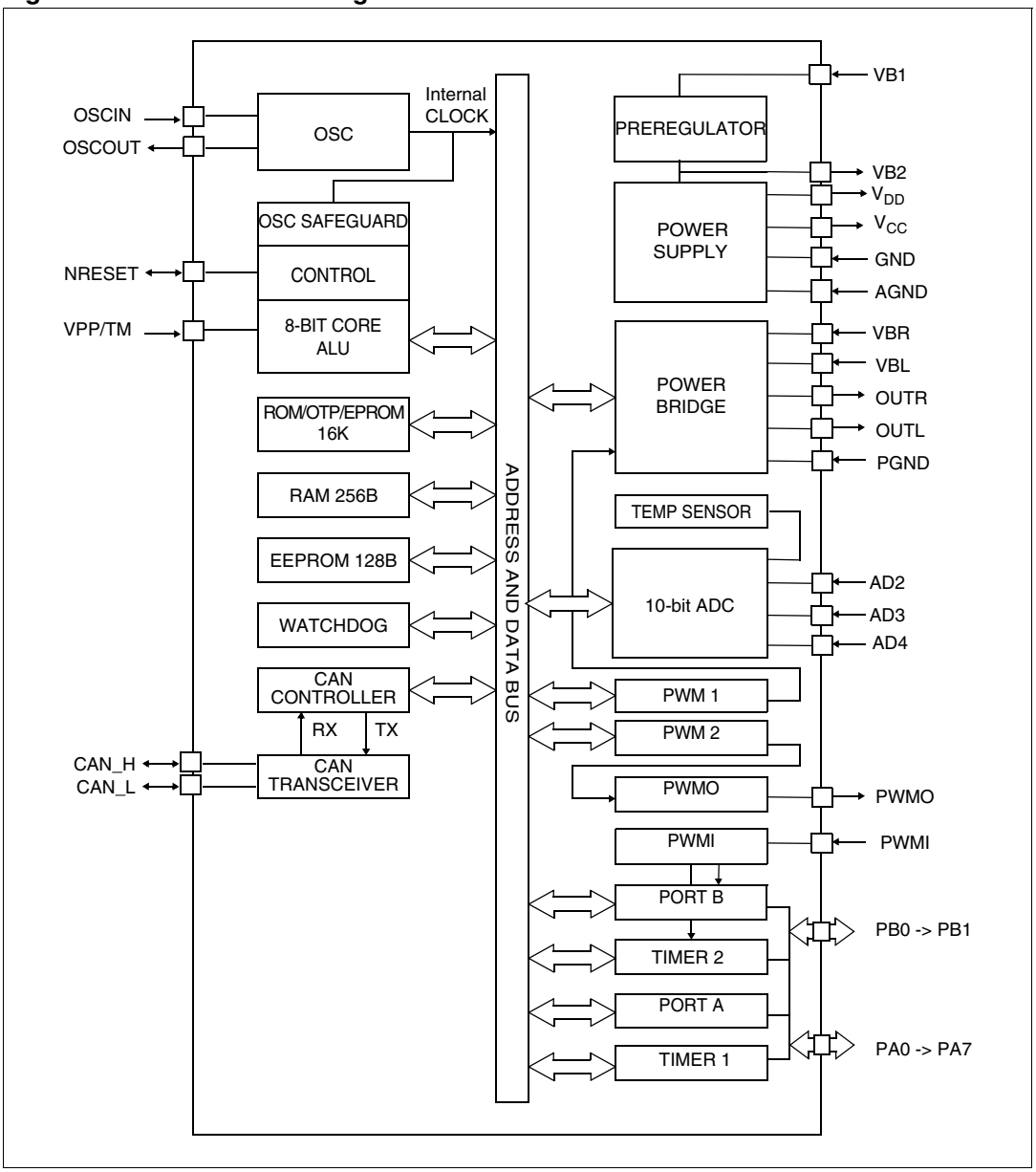
It is recommended to keep the L9805E device out of direct sunlight, since the UV content of sunlight can be sufficient to cause functional failure. Extended exposure to room level fluorescent lighting may also cause erasure.

An opaque coating (paint, tape, label, etc...) should be placed over the package window if the product is to be operated under these lighting conditions. Covering the window also reduces I_{DD} in power-saving modes due to photo-diode leakage currents.

An Ultraviolet source of wave length 2537 Å yielding a total integrated dosage of 15 Watt-sec/cm^2 is required to erase the EPROM. The device will be erased in 40 to 45minutes if such a UV lamp with a 12mW/cm^2 power rating is placed 1 inch from the device window without any interposed filters.

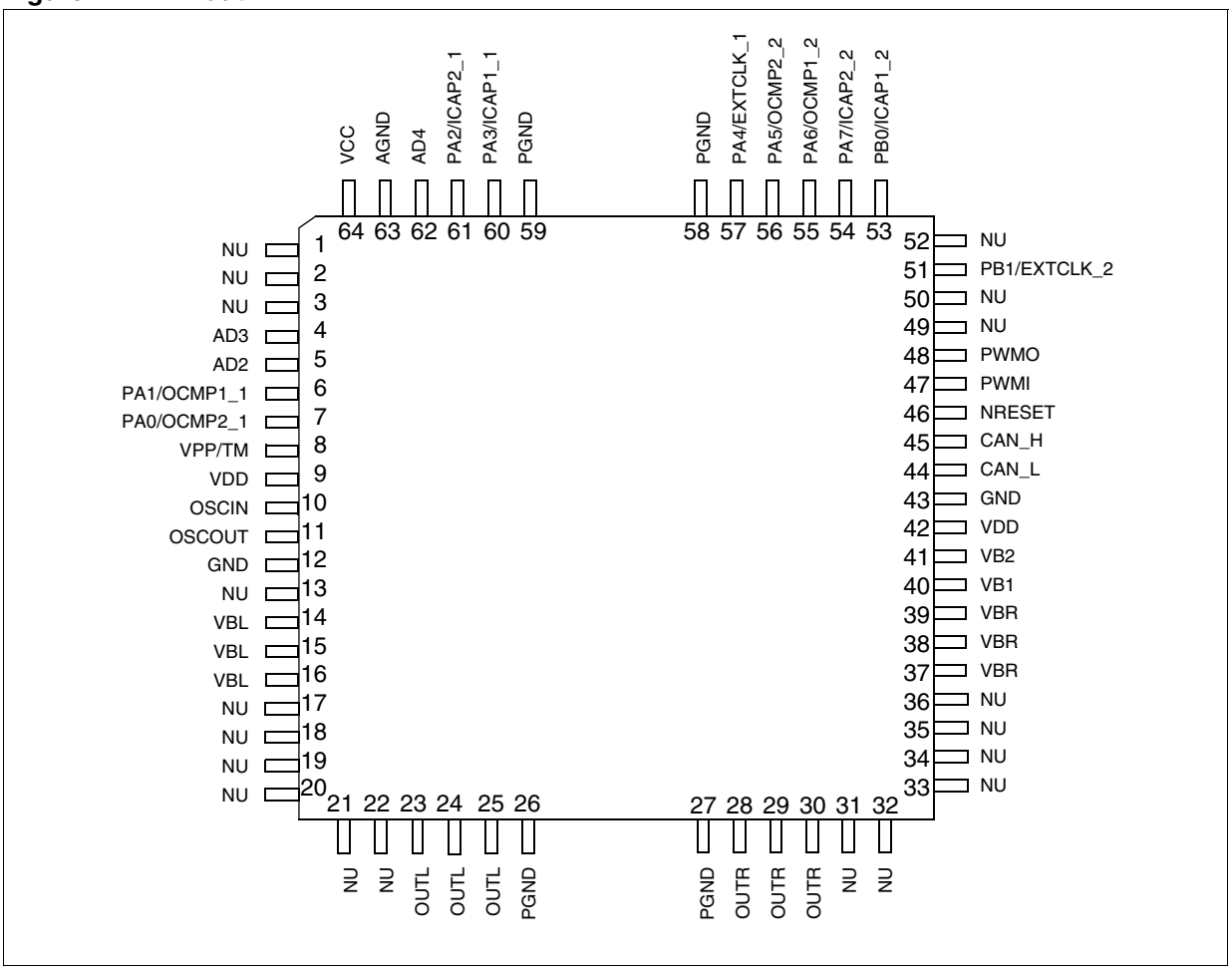
OTP and EPROM devices can be programmed by a dedicated Eprom Programming Board and software that are part of the development tool-set.

Figure 1. L9805E block diagram



1.3 Pin out

Figure 2. Pin out



1.4 Pin description

AD2-AD4: *Analog input to ADC.*

PA0/OCMP2_1-PA1/OCMP1_1: *I/Os or Output compares on Timer 1.* Alternate function software selectable (by setting OC2E or OC1E in CR2 register: bit 6 or 7 at 0031h). When used as an alternate function, this pin is a push-pull output as requested by Timer 1. Otherwise, this pin is a triggered floating input or a push-pull output.

PA2/ICAP2_1-PA3/ICAP1_1: I/*Os or Input captures on Timer 1.* Before using this I/O as alternate inputs, they must be configured by software in input mode (DDR=0). In this case, these pins are a triggered floating input. Otherwise (I/O function), these pin are triggered floating inputs or push-pull outputs.

PA4/EXTCLK_1: *PA4 I/O or External Clock on Timer 1*. Before using this I/O as alternate input, it must be configured by software in input mode (DDR=0). In this case, this pin is a triggered floating input. Otherwise (I/O function), this pin is a triggered floating input or a push-pull output.

PA5/OCMP2_2-PA6/OCMP1_2: *I/Os or Output Compares on Timer 2*. Alternate function software selectable (by setting OC2E or OC1E in CR2 register: bit 6 or 7 at 0041h). When used as alternate functions, these pins are push-pull outputs as requested by Timer 2. Otherwise, these pins are triggered floating inputs or push-pull outputs.

PA7/ICAP2_2-PB0/ICAP1_2: *I/Os or Input Captures on Timer 2*. Before using these I/Os as alternate inputs, they must be configured by software in input mode (DDR=0). In this case, these pins are triggered floating inputs. Otherwise (I/O function), these pins are triggered floating inputs or push-pull outputs.

PB1/EXTCLK_2: *PB1 I/O or External Clock on Timer 2*. Before using this I/O as alternate input, it must be configured by software in input mode (DDR=0). In this case, this pin is a triggered floating input. Otherwise (I/O function), this pin is a triggered floating input or a push-pull output.

VPP/TM: Input. This pin must be held low during normal operating modes.

VDD: Output. 5V Power supply for digital circuits, from internal voltage regulator.

OSCIN: Input Oscillator pin.

OSCOUT: Output Oscillator pin.

GND: Ground for digital circuits.

VBR: Power supply for Right half-bridge.

OUTR: Output of Right half-bridge.

PGND: Ground for power transistor.

OUTL: Output of Left half-bridge.

VBL: Power supply for Left half-bridge.

VB1: Power supply for voltage regulators.

VB2: Pre-regulated voltage for analog circuits.

CAN_L: Low side CAN bus output.

CAN_H: High side CAN bus input.

NRESET: Bidirectional. This active low signal forces the initialization of the MCU. This event is the top priority non maskable interrupt. It can be used to reset external peripherals.

PWMI: *PWM input.* Directly connected to Input Capture 2 on Timer 2.

PWMO: *PWM output.* Connected to the output of PWM2 module.

AGND: Ground for all analog circuitry (except power bridge)**.**

VCC: Output. 5V power supply for analog circuits, from internal voltage regulator.

1.5 Register & memory map

As shown in the *Table 2*, the MCU is capable of addressing 64K bytes of memories and I/O registers. In this MCU, 63742 of these bytes are user accessible.

The available memory locations consist of 128 bytes of I/O registers, 256 bytes of RAM, 128 bytes of EEPROM and 16Kbytes of user EPROM/ROM. The RAM space includes 64 bytes for the stack from 0140h to 017Fh.

The highest address bytes contain the user reset and interrupt vectors.

Table 2. Memory map

Address	Block	Register label	Register name	Reset status	Remarks
0000h 0001h 0002h 0003h	Port A	PADR .. PADDR .. PAOR ..	Data Register Data Direction Register Option Register Not Used	00h 00h 00h	R/W R/W R/W Absent
0004h 0005h 0006h 0007h	Port B	PBDR .. PBDDR .. PBOR ..	Data Register Data Direction Register Option Register Not Used	00h 00h 00h	R/W R/W R/W Absent
0008h to 000Fh	RESERVED				
0010h 0011h 0012h 0013h 0014h 0015h 0016h	PWM1	P1CYRH .. P1CYRL .. P1DRH .. P1DRL .. P1CR .. P1CTH .. P1CTL ..	PWM1 Cycle Register High PWM1 Cycle Register Low PWM1 Duty Register High PWM1 Duty Register Low PWM1 Control Register PWM1 Counter Register High PWM1 Counter Register Low	00h 00h 00h 00h 00h 00h 00h	R/W R/W R/W R/W R/W Read Only Read Only
0017h	RESERVED				
0018h 0019h 001Ah 001Bh 001Ch 001Dh 001Eh	PWM2	P2CYRH .. P2CYRL .. P2DRH .. P2DRL .. P2CR .. P2CTH .. P2CTL ..	PWM2 Cycle Register High PWM2 Cycle Register Low PWM2 Duty Register High PWM2 Duty Register Low PWM2 Control Register PWM2 Counter Register High PWM2 Counter Register Low	00h 00h 00h 00h 00h 00h 00h	R/W R/W R/W R/W R/W Read Only Read Only
001Fh	RESERVED				
0020h		MISCR ..	Miscellaneous Register	00h	see *Section 3.4*
0021h	Power Bridge	PBCSR ..	Bridge Control Status Register	00h	R/W
0022h		DCSR ..	Dedicated Control Status Register	00h	R/W
0023h to 0029h	RESERVED				
002Ah 002Bh	WDG	WDGCR .. WDGSR ..	Watchdog Control Register Watchdog Status Register	7Fh 00h	R/W R/W
002Ch	EEPROM	EECR ..	EEPROM Control register	00h	R/W
002Dh 002Eh	EPROM	ECR1 ECR2	EPROM Control register 1 EPROM Control register 2		ST INTERNAL USE ONLY
002Fh 0030h	CRC	CRCL CRCH	CRCL Test Register CRCH Test Register		ST INTERNAL USE ONLY

Table 2. Memory map (continued)

Address	Block	Register label	Register name	Reset status	Remarks
0031h 0032h 0033h 0034h-0035h 0036h-0037h 0038h-0039h 003Ah-003Bh 003Ch-003Dh 003Eh-003Fh	TIM1	T1CR2 .. T1CR1 .. T1SR .. T1IC1HR .. T1IC1LR .. T1OC1HR .. T1OC1LR .. T1CHR .. T1CLR .. T1ACHR .. T1ACLR .. T1IC2HR .. T1IC2LR .. T1OC2HR .. T1OC2LR ..	Timer 1 Control Register2 Timer 1 Control Register1 Timer 1 Status Register Timer 1 Input Capture1 High Register Timer 1 Input Capture1 Low Register Timer 1 Output Compare1 High Register Timer 1 Output Compare1 Low Register Timer 1 Counter High Register Timer 1 Counter Low Register Timer 1 Alternate Counter High Register Timer 1 Alternate Counter Low RegisteR Timer 1 Input Capture2 High Register Timer 1 Input Capture2 Low Register Timer 1 Output Compare2 High Register Timer 1 Output Compare2 Low Register	00h 00h xxh xxh xxh xxh xxh FFh FCh FFh FCh xxh xxh xxh xxh	R/W R/W Read Only Read Only Read Only R/W R/W Read Only Read Only Read Only Read Only Read Only Read Only R/W R/W
0040h	Reserved: Write Forbidden				
0041h 0042h 0043h 0044h-0045h 0046h-0047h 0048h-0049h 004Ah-004Bh 004Ch-004Dh 004Eh-004Fh	TIM2	T2CR2 .. T2CR1 .. T2SR .. T2IC1HR .. T2IC1LR .. T2OC1HR .. T2OC1LR .. T2CHR .. T2CLR .. T2ACHR .. T2ACLR .. T2IC2HR .. T2IC2LR .. T2OC2HR .. T2OC2LR ..	Timer 2 Control Register2 Timer 2 Control Register1 Timer 2 Status Register Timer 2 Input Capture1 High Register Timer 2 Input Capture1 Low Register Timer 2 Output Compare1 High Register Timer 2 Output Compare1 Low Register Timer 2 Counter High Register Timer 2 Counter Low Register Timer 2 Alternate Counter High Register Timer 2 Alternate Counter Low Register Timer 2 Input Capture2 High Register Timer 2 Input Capture2 Low Register Timer 2 Output Compare2 High Register Timer 2 Output Compare2 Low Register	00h 00h xxh xxh xxh xxh xxh FFh FCh 00h 00h xxh xxh xxh xxh	R/W R/W Read Only Read Only Read Only R/W R/W Read Only Read Only Read Only Read Only Read Only Read Only R/W R/W
0050h to 0059h	RESERVED				
005Ah 005Bh 005Ch 005Dh 005Eh 005Fh 0060h to 006Fh	CAN	CANISR .. CANICR .. CANCSR .. CANBRPR .. CANBTR .. CANPSR ..	CAN Interrupt Status Register CAN Interrupt Control Register CAN Control/Status Register CAN Baud Rate Prescaler CAN Bit Timing Register CAN Page Selection CAN First address to last address of PAGE X	00h 00h 00h 00h 23h 00h --	R/W R/W R/W R/W R/W R/W see page mapping and register description
0070h 0071h 0072h	ADC	ADCDRH .. ADCDRL .. ADCCSR ..	ADC Data Register High ADC Data Register Low ADC Control/Status Register	00h 00h 20h	Read Only Read Only R/W

Address	Block	Description
0080h to 013Fh	RAM 256 Bytes including STACK 64 bytes (0140h to 017Fh)	User variables and subroutine nesting
0140h to 017Fh		
0180h to 0BFFh	RESERVED	
0C00h to 0C7Fh	EEPROM 128 bytes	including 4 bytes reserved for temperature sensor trimming (see Section 5.5.6) 0C7CH: T0H 0C7DH: T0L 0C7EH: VT0H 0C7FH: VT0L
0C80h to BFFFh	RESERVED	
C000h to FFDFh	EPROM 16K bytes (16384 bytes)	User application code and data
FFE0h to FFFFh		Interrupt and Reset Vectors

2 Central processing unit (CPU)

2.1 Introduction

The CPU has a full 8-bit architecture. Six internal registers allow efficient 8-bit data manipulation. The CPU is capable of executing 63 basic instructions and features 17 main addressing modes.

2.2 CPU registers

The 6 CPU registers are shown in the programming model in *Figure 3*. Following an interrupt, all registers except Y are pushed onto the stack in the order shown in *Figure 4*. They are popped from stack in the reverse order.

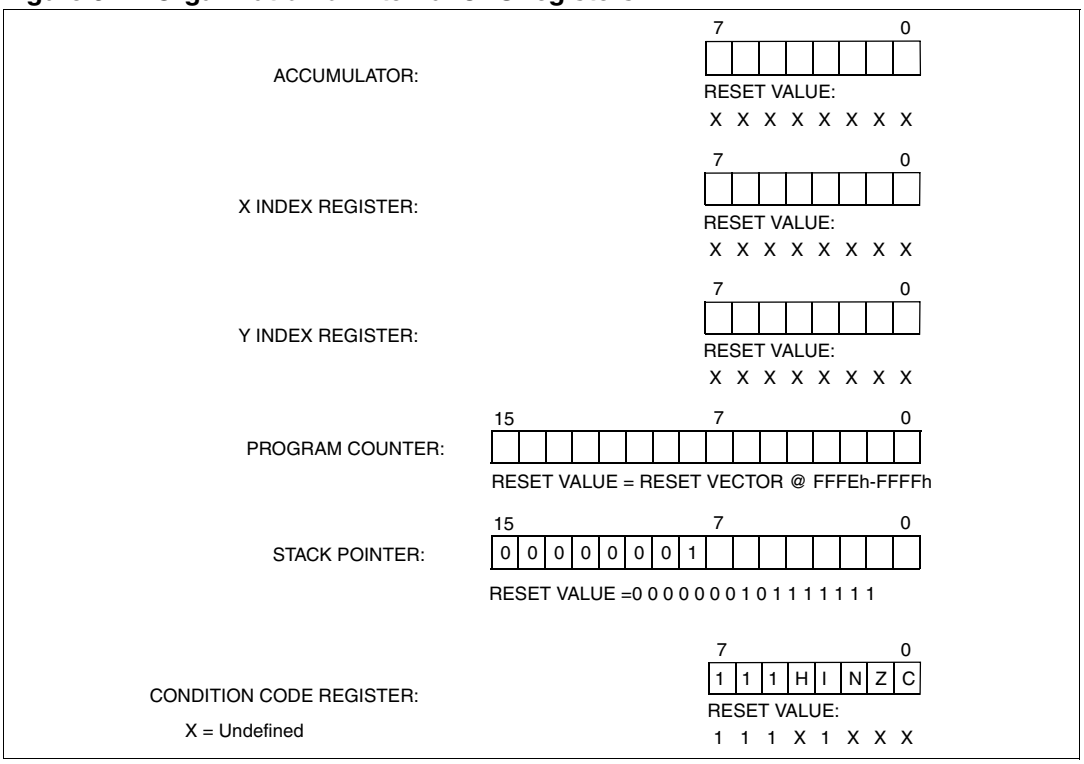
The Y register is not affected by these automatic procedures. The interrupt routine must therefore handle Y, if needed, through the PUSH and POP instructions.

Accumulator (A). The Accumulator is an 8-bit general purpose register used to hold operands and the results of the arithmetic and logic calculations as well as data manipulations.

Index Registers (X and Y). These 8-bit registers are used to create effective addresses or as temporary storage areas for data manipulation. The Cross-Assembler generates a PRECEDE instruction (PRE) to indicate that the following instruction refers to the Y register.

Program Counter (PC). The program counter is a 16-bit register containing the address of the next instruction to be executed by the CPU.

Figure 3. Organization of internal CPU registers



Stack Pointer (SP) The Stack Pointer is a 16-bit register. Since the stack is 64 bytes deep, the most significant bits are forced as indicated in *Figure 3* in order to address the stack as it is mapped in memory.

Following an MCU Reset, or after a Reset Stack Pointer instruction (RSP), the Stack Pointer is set to point to the next free location in the stack. It is then decremented after data has been pushed onto the stack and incremented before data is popped from the stack.

Note: When the lower limit is exceeded, the Stack Pointer wraps around to the stack upper limit, without indicating the stack overflow. The previously stored information is then overwritten and therefore lost.

The upper and lower limits of the stack area are shown in the Memory Map.

The stack is used to save the CPU context during subroutine calls or interrupts. The user may also directly manipulate the stack by means of the PUSH and POP instructions. In the case of an interrupt (refer to *Figure 4*), the PCL is stored at the first location pointed to by the SP. Then the other registers are stored in the next locations.

When an interrupt is received, the SP is decremented and the context is pushed on the stack.

On return from interrupt, the SP is incremented and the context is popped from the stack.

A subroutine call occupies two locations and an interrupt five locations in the stack area.

Condition Code Register (CC) The Condition Code register is a 5-bit register which indicates the result of the instruction just executed as well as the state of the processor. These bits can be individually tested by a program and specified action taken as a result of their state. The following paragraphs describe each bit of the CC register in turn.

Half carry bit (H) The H bit is set to 1 when a carry occurs between bits 3 and 4 of the ALU during an ADD or ADC instruction. The H bit is useful in BCD arithmetic subroutines.

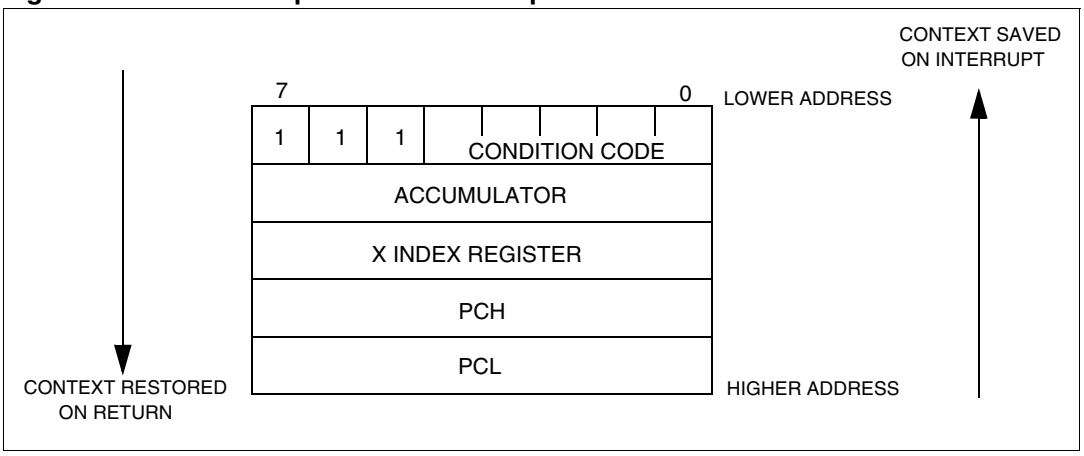
Interrupt mask (I) When the I bit is set to 1, all interrupts except the TRAP software interrupt are disabled. Clearing this bit enables interrupts to be passed to the processor core. Interrupts requested while I is set are latched and can be processed when I is cleared (only one interrupt request per interrupt enable flag can be latched).

Negative (N) When set to 1, this bit indicates that the result of the last arithmetic, logical or data manipulation is negative (i.e. the most significant bit is a logic 1).

Zero (Z) When set to 1, this bit indicates that the result of the last arithmetic, logical or data manipulation is zero.

Carry/Borrow (C) When set, C indicates that a carry or borrow out of the ALU occured during the last arithmetic operation. This bit is also affected during execution of bit test, branch, shift, rotate and store instructions.

Figure 4. Stack manipulation on interrupt



3 Clocks, reset, interrupts & power saving modes

3.1 Clock system

3.1.1 General description

The MCU accepts either a Crystal or Ceramic resonator, or an external clock signal to drive the internal oscillator. The internal clock (f_{CPU}) is derived from the external oscillator frequency (f_{OSC}). The external Oscillator clock is first divided by 2, and an additional division factor of 2, 4, 8, or 16 can be applied. In Slow Mode, to reduce the frequency of the f_{CPU}; this clock signal is also routed to the on-chip peripherals (except the CAN). The CPU clock signal consists of a square wave with a duty cycle of 50%.

The internal oscillator is designed to operate with an AT-cut parallel resonant quartz crystal resonator in the frequency range specified for f_{osc}. The circuit shown in *Figure 6* is recommended when using a crystal, and *Table 3* lists the recommended capacitance and feedback resistance values. The crystal and associated components should be mounted as close as possible to the input pins in order to minimize output distortion and start-up stabilisation time.

Use of an external CMOS oscillator is recommended when crystals outside the specified frequency ranges are to be used.

Table 3. Recommended values for 16 MHz crystal resonator

R_{SMAX}	40 Ω	60 Ω	150 Ω
C_{OSCIN}	56pF	47pF	22pF
C_{OSCOUT}	56pF	47pF	22pF
R_P	1-10 MΩ	1-10 MΩ	1-10 MΩ

Note: *R_{SMAX} is the equivalent serial resistor of the crystal (see crystal specification).*
***C_{OSCIN},C_{OSCOUT}:** Maximum total capacitances on pins OSCIN and OSCOUT (the value includes the external capacitance tied to the pin plus the parasitic capacitance of the board and of the device).*
***Rp:** External shunt resistance. Recommended value for oscillator stability is 1MΩ.*

Figure 5. External clock source connections

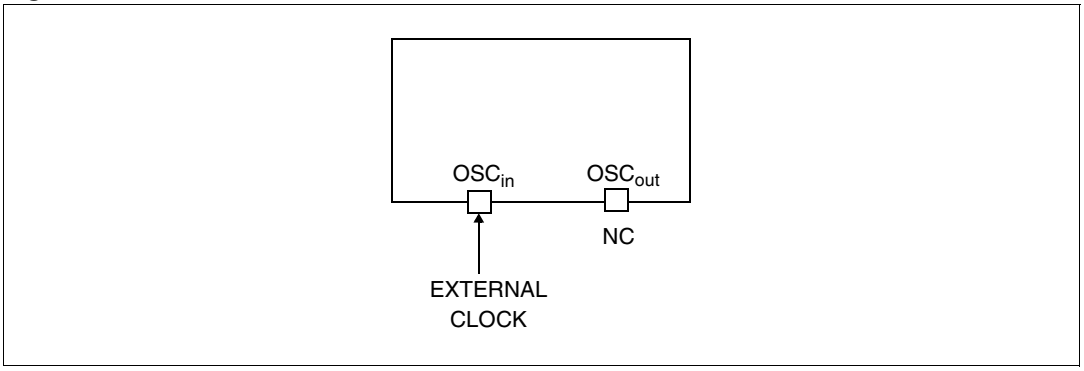


Figure 6. Crystal/Ceramic Resonator

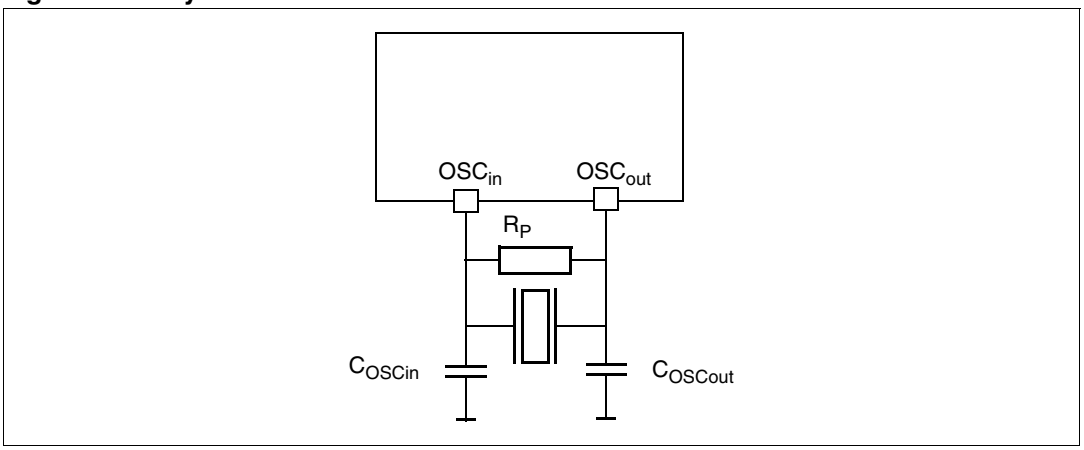
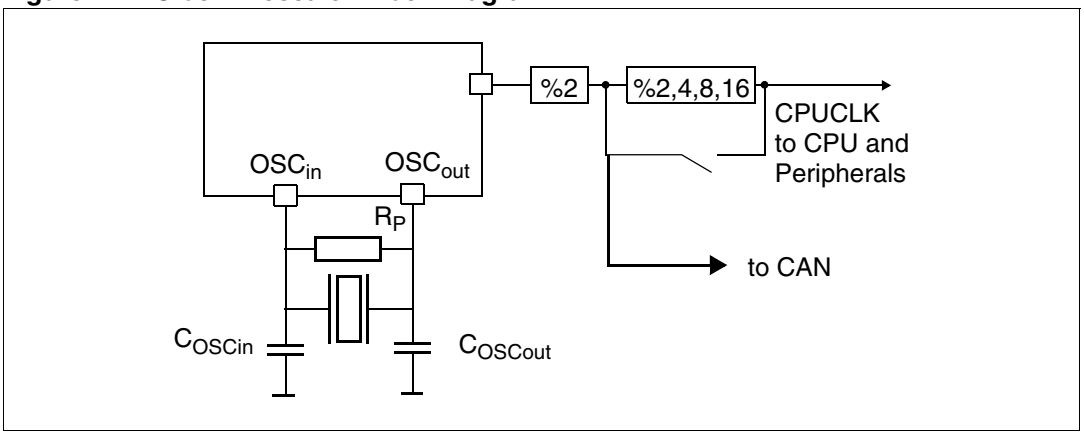


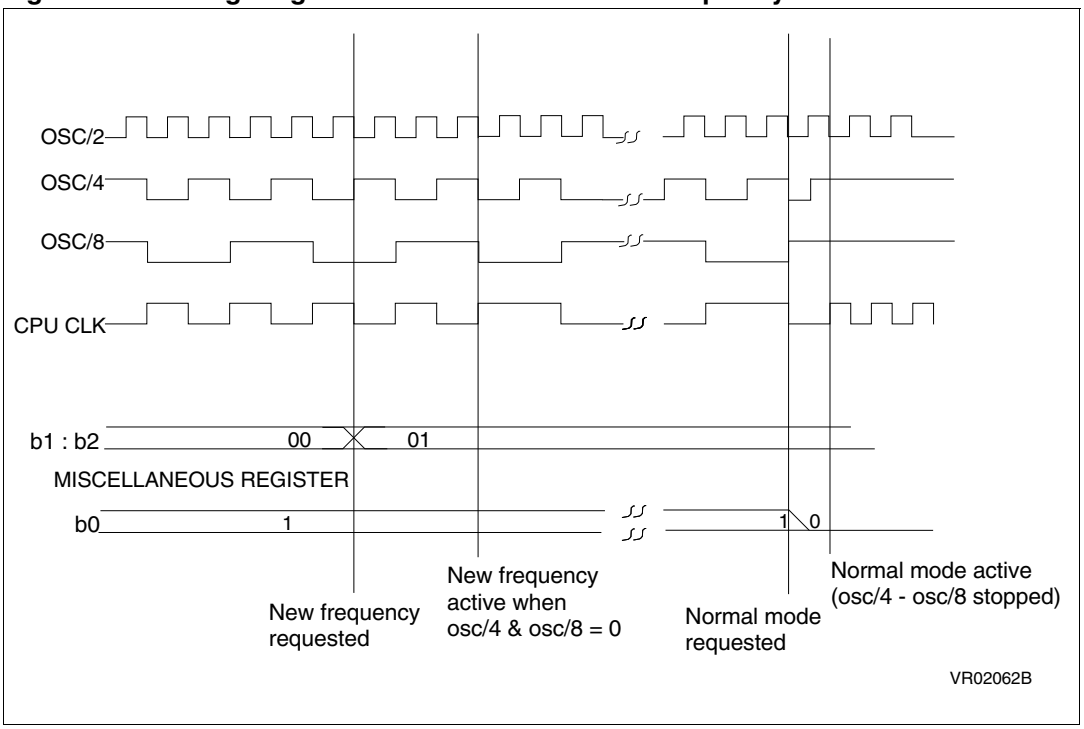
Figure 7. Clock Prescaler Block Diagram



3.1.2 External clock

An external clock may be applied to the OSCIN input with the OSCOUT pin not connected, as shown on *Figure 5*. The t_{OXOV} specifications does not apply when using an external clock input. The equivalent specification of the external clock source should be used instead of t_{OXOV} (see *Chapter 7.5*).

Figure 8. Timing Diagram for Internal CPU Clock Frequency transitions



3.2 Oscillator safeguard

The L9805E contains an oscillator safe guard function.

This function provides a real time check of the crystal oscillator generating a reset condition when the clock frequency has anomalous value.

If $f_{OSC}<f_{low}$, a reset is generated.

If $f_{OSC}>f_{high}$, a reset is generated.

A flag in the Dedicated Control Status Register indicates if the last reset was a safeguard reset.

At the output of reset state the safeguard is disabled. To activate the safeguard SFGEN bit must be set.

Note: Following a reset, the safeguard is disabled. Once activated it cannot be disabled, except by a reset.

3.2.1 Dedicated control status register

DCSR

Address 0022h - Read/Write

Reset Value:xx00 0000 (00h)

SGFL	SGFH	SFGEN	CANDS	b3	b2	b1	PIEN

b6 = **SGFH:** Safeguard high flag. Set by an Oscillator Safeguard Reset generated for frequency too high, cleared by software (writing zero) or Power On / Low Voltage Reset.

This flag is useful for distinguishing Safeguard Reset, Power On / Low Voltage Reset and Watchdog Reset.

b7 = **SGFL:** Safeguard low flag. Set by an Oscillator Safeguard Reset generated for frequency too low, cleared by software (writing zero) or Power On / Low Voltage Reset. This flag is useful for distinguishing Safeguard Reset, Power On / Low Voltage Reset and Watchdog Reset.

b5 = **SFGEN**: Safeguard enable when set. It's cleared only by hardware after a reset.

b4 = **CANDS**: CAN Transceiver disable. When this bit is set the CAN transceiver goes in Power Down Mode and does not work until this bit is reset. CANDS is 0 after reset so the standard condition is with the transceiver enabled. This bit can be used by application requiring low power consumption (see *Section 5.8* for details).

b3,b2,b1 = **not used**

b0 = **PIEN**: PWMI input enable. When set, the PWMI input line is connected to Input Capture 2 of Timer 2. Otherwise, ICAP2_2 is the alternate function of PA7. See *Figure 37* for the explanation of this function.

3.3 Watchdog system (WDG)

3.3.1 Introduction

The Watchdog is used to detect the occurrence of a software fault, usually generated by external interference or by unforeseen logical conditions, which causes the application program to give up its normal sequence. The Watchdog circuit generates an MCU reset on expiry of a programmed time period, unless the program refreshes the counter's contents before it is decremented to zero.

3.3.2 Main features

- Programmable Timer (64 increments of 12,288 CPU clock)
- Programmable Reset
- reset (if watchdog activated) after an HALT instruction or when bit timer MSB reaches zero
- Watchdog Reset indicated by status flag.

3.3.3 Functional description

The counter value stored in the CR register (bits T6:T0), is decremented every 12,288 machine cycles, and the length of the timeout period can be programmed by the user in 64 increments.

If the Watchdog is activated (the WDGA bit is set) and when the 7-bit timer (bits T6:T0) rolls over from 40h to 3Fh (T6 becomes cleared), it initiates a reset cycle pulling low the reset pin for typically 500ns.

The application program must write in the CR register at regular intervals during normal operation to prevent an MCU reset. The value to be stored in the CR register must be between FFh and C0h (see Table 1):

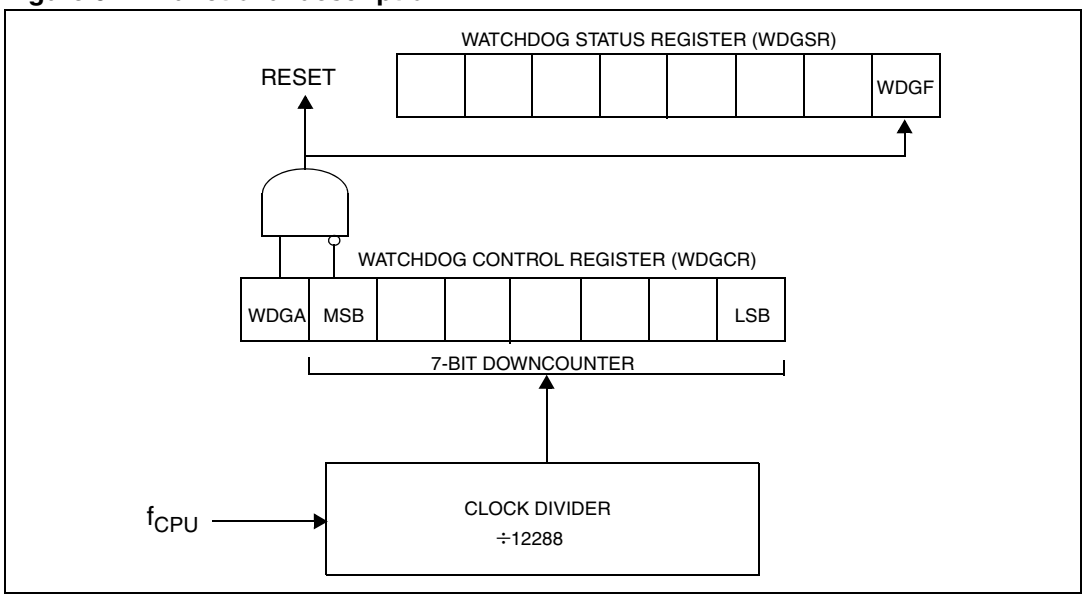
- The WDGA bit is set (watchdog enabled)
- The T6 bit is set to prevent generating an immediate reset
- The T5:T0 bit contain the number of increments which represents the time delay before the watchdog produces a reset.

Table 4. Watchdog timing (f_{OSC} = 16 MHz)

WDG Register initial value	WDG timeout period (ms)
FFh	98.3
C0h	1.54

Note: Following a reset, the watchdog is disabled. Once activated it cannot be disabled, except by a reset.
The T6 bit can be used to generate a software reset (the WDGA bit is set and the T6 bit is cleared).
If the Watchdog is activated, the HALT instruction will generate a Reset.

Figure 9. Functional description



The Watchdog delay time is defined by bits 5-0 of the Watchdog register; bit 6 must always be set in order to avoid generating an immediate reset. Conversely, this can be used to generate a software reset (bit 7 = 1, bit 6 = 0).

The Watchdog must be reloaded before bit 6 is decremented to “0” to avoid a Reset. Following a Reset, the Watchdog register will contain 7Fh (bits 0-7).

If the circuit is not used as a Watchdog (i.e. bit 7 is never set), bits 6 to 0 may be used as a simple 7-bit timer, for instance as a real time clock. Since no reset will be generated under these conditions, the Watchdog control register must be monitored by software.

A flag in the watchdog status register indicates if the last reset has a watchdog reset or not, before clearing by a write of this register.

3.3.4 Register description

Watchdog control register

(WDGCR)

Register Address: 002Ah — Read/Write

Reset Value: 0111 1111 (7Fh)

7							0
WDGA	T6	T5	T4	T3	T2	T1	T0

b7 = **WDGA:** Activation bit.

This bit is set by software and only cleared by hardware after a reset. When WDGA = 1, the watchdog can generate a reset.

0: Watchdog disabled

1: Watchdog enabled.

b6-0 =**T6-T0**: 7 bit timer (Msb to Lsb)

These bits contain the decremented value. A reset is produced when it rolls over from 40h to 3Fh (T6 become cleared).

Watchdog status register

(WDGSR)

Register Address: 002Bh — Read/Write

Reset Value(*): 0000 0000 (00h)

7							0
-	-	-	-	-	-	-	WDGF

b7-1 = **not used**

b0 = **WDGF:** Watchdog flag. Set by a Watchdog Reset, cleared by software (writing zero) or Power On / Low Voltage Reset. This flag is useful for distinguishing Power On / Low Voltage Reset and Watchdog Reset.

(*): Except in the case of Watchdog Reset.

3.4 Miscellaneous register

(MISCR)

The Miscellaneous register allows the user to select the Slow operating mode and to set the clock division prescaler factor. Bits 3, 4 determine the signal conditions which will trigger an interrupt request on I/O pins having interrupt capability.

Register Address: 0020h — Read/Write

Reset Value:0000 0000 (00h)

7							0
-	-	-	b4	b3	b2	b1	b0

b0 - *Slow mode select*

0- Normal mode - Oscillator frequency / 2 (Reset state)

1- Slow mode (Bits b1 and b2 define the prescaler factor)

b1, b2 - *CPU clock prescaler for slow mode*

b2	b1	Option
0	0	Oscillator frequency / 4
1	0	Oscillator frequency / 8
0	1	Oscillator frequency / 16
1	1	Oscillator frequency / 32

b3, b4 - *External interrupt option*

b4	b3	Option
0	0	Falling edge and low level (Reset state)
1	0	Falling edge only
0	1	Rising edge only
1	1	Rising and Falling edge

The selection issued from b3/b4 combination is applied to PA[0]..PA[7],PB0,PB1 external interrupt. The selection can be made only if I bit in CC register is reset (interrupt enabled).

b3, b4 can be written only when the Interrupt Mask (I) of the CC (Condition Code) register is set to 1.

b5,b6,b7 = **not used**

3.5 Reset

3.5.1 Introduction

There are four sources of Reset:

- NRESET pin (external source)
- Power-On Reset / Low Voltage Detection (Internal source)
- WATCHDOG (Internal Source)
- SAFEGUARD (Internal source)

The Reset Service Routine vector is located at address FFFEh-FFFFh.

3.5.2 External reset

The NRESET pin is both an input and an open-drain output with integrated pull-up resistor. When one of the internal Reset sources is active, the Reset pin is driven low to reset the whole application.

3.5.3 Reset operation

The duration of the Reset condition, which is also reflected on the output pin, is fixed at 4096 internal CPU Clock cycles. A Reset signal originating from an external source must have a duration of at least 1.5 internal CPU Clock cycles in order to be recognised. At the end of the Power-On Reset cycle, the MCU may be held in the Reset condition by an External Reset signal. The NRESET pin may thus be used to ensure V_{DD} has risen to a point where the MCU can operate correctly before the user program is run. Following a Power-On Reset event, or after exiting Halt mode, a 4096 CPU Clock cycle delay period is initiated in order to allow the oscillator to stabilise and to ensure that recovery has taken place from the Reset state.

During the Reset cycle, the device Reset pin acts as an output that is pulsed low. In its high state, an internal pull-up resistor of about 300KΩ is connected to the Reset pin. This resistor can be pulled low by external circuitry to reset the device.

3.5.4 Power-on reset - Low voltage detection

The POR/LVD function generates a static reset when the supply voltage is below a reference value. In this way, the Power-On Reset and Low Voltage Reset function are provided, in order to keep the system in safe condition when the voltage is too low.

The Power-Up and Power-Down thresholds are different, in order to avoid spurious reset when the MCU starts running and sinks current from the supply.

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

- $V_{ResetON}$ when V_{DD} is rising
- $V_{ResetOFF}$ when V_{DD} is falling

The POR/LVD function is explained in *Figure 11*.

Power-On Reset activates the reset pull up transistor performing a complete chip reset. In the same way a reset can be triggered by the watchdog, by the safeguard or by external low level at NRESET pin. An external capacitor connected between NRESET and ground can extend the power on reset period if required.

Figure 10. Power up/down behaviour

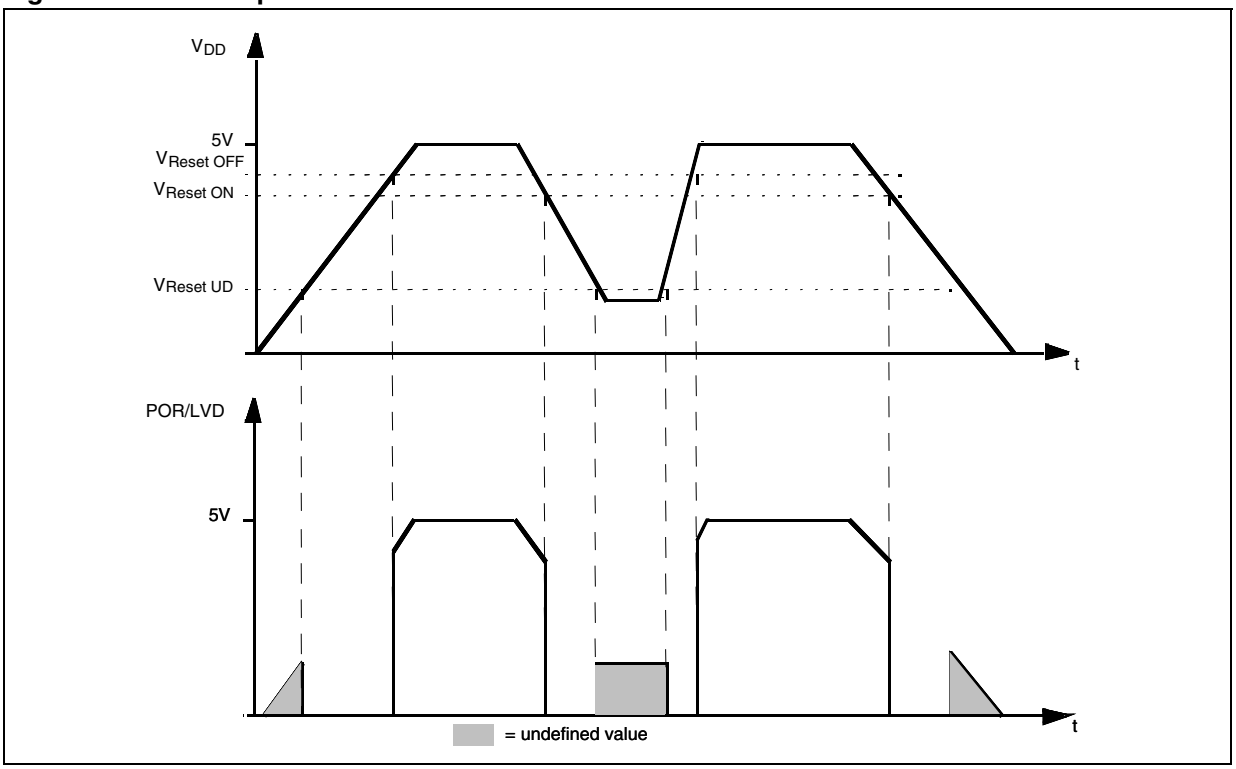
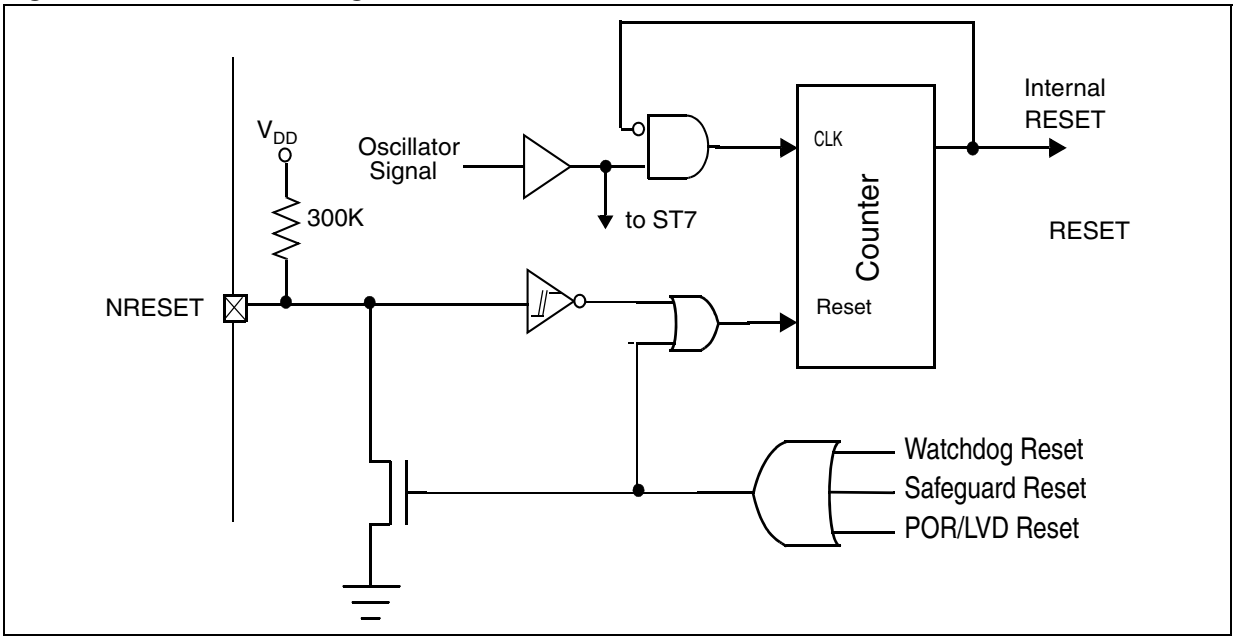


Figure 11. Reset block diagram



3.6 Interrupts

A list of interrupt sources is given in *Table 5* below, together with relevant details for each source. Interrupts are serviced according to their order of priority, starting with I0, which has the highest priority, and so to I12, which has the lowest priority.

The following list describes the origins for each interrupt level:

- I0 connected to Ports PA0-PA7, PB0-PB1
- I1 connected to CAN
- I2 connected to Power Diagnostics
- I3 connected to Output Compare of Timer 1
- I4 connected to Input Capture of TImer 1
- I5 connected to Timer 1 Overflow
- I6 connected to Output Compare of Timer 2
- I7 connected to Input Capture of TImer 2
- I8 connected to Timer 2 Overflow
- I9 connected to ADC End Of Conversion
- I10 connected to PWM 1 Overflow
- I11 connected to PWM 2 Overflow
- I12 connected to EEPROM

Exit from Halt mode may only be triggered by an External Interrupt on one of the following ports: PA0-PA7 (I0), PB0-PB1 (I0), or by an Internal Interrupt coming from CAN peripheral (I1).

If more than one input pin of a group connected to the same interrupt line are selected simultaneously, the OR of this signals generates the interrupt.

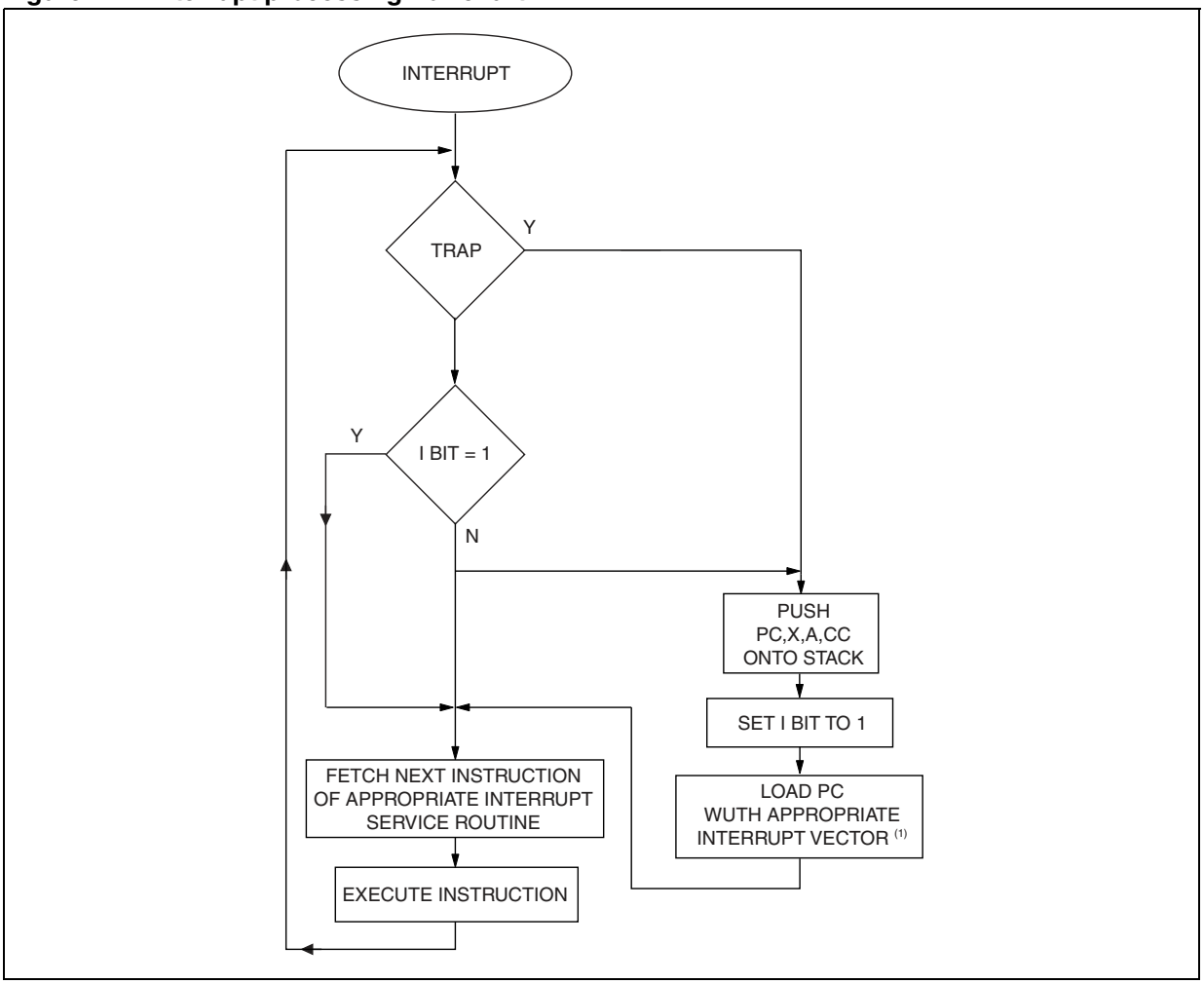
Table 5. Interrupt mapping

Interrupts	Register	Flag name	Interrupt source	Vector address
Reset	N/A	N/A	-	FFFEh-FFFFh
Software	N/A	N/A	-	FFFCh-FFFDh
Ext. interrupt (Ports PA0-PA7, PB0-PB1)	N/A	N/A	I0	FFFAh-FFFBh
Receive interrupt flag	CAN Status	RXIFi	I1	FFF8h-FFF9h
Transmit interrupt flag		TXIF		
Error interrupt pending		EPND		
Power bridge short circuit	Bridge Control Status	SC	I2	FFF6h-FFF7h
Overtemperature		OVT		
Output compare 1	Timer 1 Status	OCF1_1	I3	FFF4h-FFF5h
Output compare 2		OCF2_1		
Input capture 1	Timer 1 Status	ICF1_1	I4	FFF2h-FFF3h
Input capture 2		ICF2_1		

Table 5. Interrupt mapping (continued)

Interrupts	Register	Flag name	Interrupt source	Vector address
Timer overflow	Timer 1 Status	TOF_1	I5	FFF0h-FFF1h
Output compare 1	Timer 2 Status	OCF1_2	I6	FFEEh-FFEFh
Output compare 2		OCF2_2		
Input capture 1	Timer 2 Status	ICF1_2	I7	FFECh-FFEDh
Input capture 2		ICF2_2		
Timer overflow	Timer 2 Status	TOF_2	I8	FFEAh-FFEBh
ADC end of conversion	ADC Control	EOC	I9	FFE8h-FFE9h
PWM 1 Overflow	N/A	N/A	I10	FFE6h-FFE7h
PWM 2 Overflow	N/A	N/A	I11	FFE4h-FFE5h
EEPROM Programming	EEPROM Control	E2ITE	I12	FFE2h-FFE3h

Figure 12. Interrupt processing flowchart



Note: 1 See Table 5

3.7 Power saving modes

3.7.1 Introduction

There are three Power Saving modes. The Slow Mode may be selected by setting the relevant bits in the Miscellaneous register as detailed in *Section 3.4*. Wait and Halt modes may be entered using the WFI and HALT instructions.

3.7.2 Slow mode

In Slow mode, the oscillator frequency can be divided by 4, 8, 16 or 32 rather than by 2. The CPU and peripherals (except CAN, see Note) are clocked at this lower frequency. Slow mode is used to reduce power consumption.

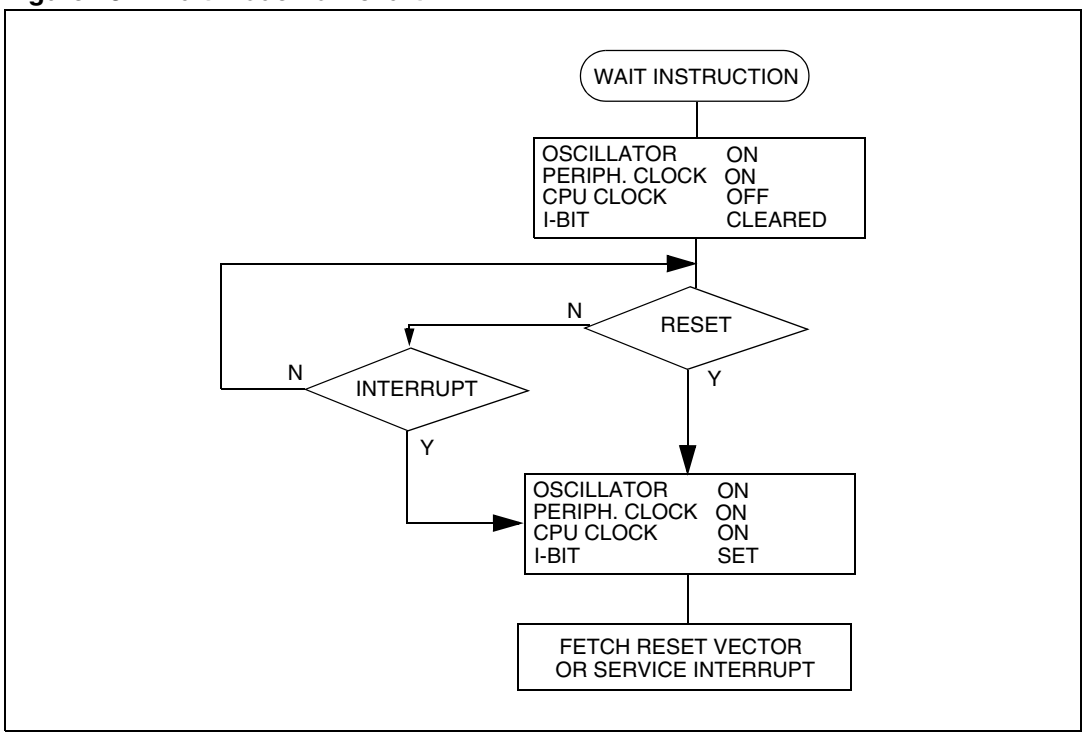
Note: Before entering Slow mode and to guarantee low power operations, the CAN Controller must be placed by software in STANDBY mode.

3.7.3 Wait mode

Wait mode places the MCU in a low power consumption mode by stopping the CPU. All peripherals remain active. During Wait mode, the I bit (CC Register) is cleared, so as to enable all interrupts. All other registers and memory remain unchanged. The MCU will remain in Wait mode until an Interrupt or Reset occurs, whereupon the Program Counter branches to the starting address of the Interrupt or Reset Service Routine.
The MCU will remain in Wait mode until a Reset or an Interrupt (coming from CAN, Timers 1 & 2, EEPROM, ADC, PWM 1 & 2, I/O ports peripherals and Power Bridge) occurs, causing its wake-up.

Refer to *Figure 12* below.

Figure 13. Wait mode flow chart



3.7.4 Halt mode

The Halt mode is the MCU lowest power consumption mode. The Halt mode is entered by executing the HALT instruction. The internal oscillator is then turned off, causing all internal processing to be stopped, including the operation of the on-chip peripherals.

When entering Halt mode, the I bit in the CC Register is cleared so as to enable External Interrupts. If an interrupt occurs, the CPU becomes active.

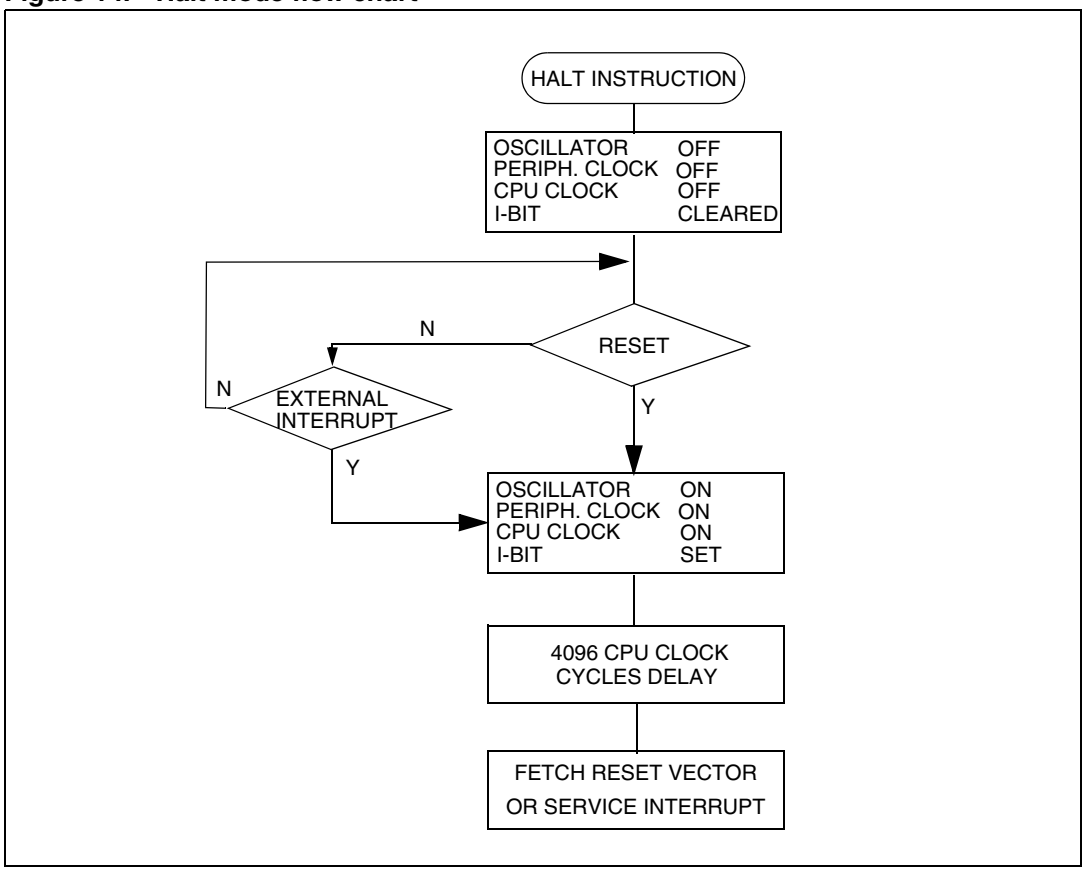
The MCU can exit the Halt mode upon reception of either an external interrupt (I0), an internal interrupt coming from the CAN peripheral (I1) or a reset. The oscillator is then turned on and a stabilisation time is provided before releasing CPU operation. The stabilisation time is 4096 CPU clock cycles.

After the start up delay, the CPU continues operation by servicing the interrupt which wakes it up or by fetching the reset vector if a reset wakes it up.

Note: The Halt mode cannot be used when the Watchdog or the Safeguard is enabled. The HALT instruction is executed while the watchdog or safeguard system is enabled, a reset is automatically generated thus resetting the entire MCU.

Halt Mode affects only the digital section of the device. All the analog circuit remain in their status, including ADC, voltage regulators, bus transceivers and power bridge.

Figure 14. Halt mode flow chart



4 Voltage regulator

4.1 Introduction

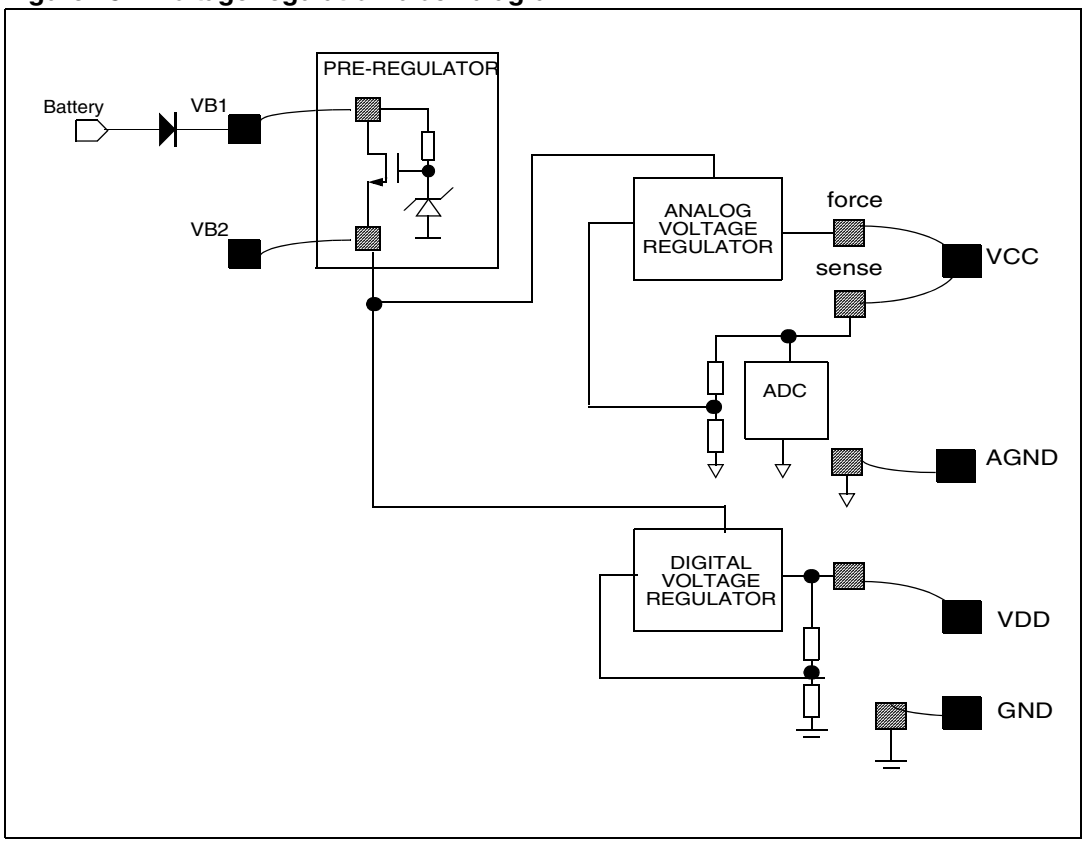
The on chip voltage regulator provides two regulated voltages, nominally 5V both. VCC supplies ADC and the analog periphery and VDD supplies the microcontroller and logic parts. These voltage are available at pins VCC and VDD to supply external components and to connect a capacitors to optimize EMI performance. A pre-regulator circuit allows to connect external tantalum capacitors to a lower (10V) voltage (VB2 pin).

4.1.1 Functional description

The main supply voltage is taken from VB1 pin. A voltage pre-regulator provides the regulated voltage on pin VB2. VB2 is the supply for the digital and analog regulators. The block diagram shows the connections between the regulators and the external pins.

In order to prevent negative spikes on the battery line to propagate on the internal supply generating spurious reset, a series diode supply VB1 pin is recommended.

Figure 15. Voltage regulation block diagram



4.2 Digital section power supply

The digital supply voltage VDD is available at pin number 42 and 9. The digital ground GND is available at pin number 43 and 12.

Pin 42 and 43 are the actual voltage regulator output and external loads must be supplied by these pins. The 100nF compensation capacitor should be connected as close as possible to pins 42 and 43.

Pin number 9 and 12 provide an external access to the internal oscillator supply. Resonator's capacitors should be grounded on pin 12.

The application board can improve noise reduction in the chip by connecting directly pin 42 to pin 9 and pin 43 to pin 12, using traces as short as possible. An additional capacitor mounted close to pin 9 and 12 can lead additional improvement.

4.2.1 VDD short circuit protection

The output current of the digital voltage regulator is controlled by a circuit that limits it to a maximum value (I_{MAXVDD}). When the output current exceeds this value the VDD voltage starts falling down. External loads must be chosen taking into account this maximum current capability of the regulator.

4.3 Analog section power supply

The analog supply voltage is available on VCC pin. The external 100nF compensation capacitor should be placed as close as possible to this pin and AGND pin.

VCC is the reference voltage for the AD conversion and must be used to supply ratiometric sensors feeding AD inputs. Any voltage drop between VCC pin and the sensor supply pin on the application board, will cause the ADC to be inaccurate when reading the sensor's output.

4.3.1 VCC Short Circuit Protection

The output current of the analog voltage regulator is controlled by a circuit that limits it to a maximum value (I_{MAXVCC}). When the output current exceeds this value the VCC voltage starts falling down. External loads must be chosen taking into account this maximum current capability of the regulator.

Warning: The pin VB2 is not short circuit protected so a short circuit on this pin will destroy the device.

5 On-chip peripherals

5.1 I/O Ports

5.1.1 Introduction

The internal I/O ports allow the transfer of data through digital inputs and outputs, the interrupt generation coming from an I/O and for specific pins, the input/output of alternate signals for the on-chip peripherals (TIMERS...).

Each pin can be programmed independently as digital input (with or without interrupt generation) or digital output.

5.1.2 Functional description

Each port has 2 main registers:

- Data Register (DR)
- Data Direction Register (DDR)

and one optional register:

- Option Register (OR)

Each I/O pin may be programmed using the corresponding register bits in the DDR and OR registers: bit X corresponding to pin X of the port. The same correspondence is used for the DR register.

The following description takes into account the OR register (for specific ports which do not provide this register refer to the I/O Port Implementation section). The generic I/O block diagram is shown in *Figure 16*.

Input modes

The input configuration is selected by clearing the corresponding DDR register bit.

In this case, reading the DR register returns the digital value applied to the external I/O pin.

Different input modes can be selected by software through the OR register.

Note: 1 1. Writing the DR register modifies the latch value but does not affect the pin status.

2 2. When switching from input to output mode, the DR register has to be written first to drive the correct level on the pin as soon as the port is configured as an output.

3 3. Do not use read/modify/write instructions (BSET or BRES) to modify the DR register

External interrupt function

When an I/O is configured as Input with Interrupt, an event on this I/O can generate an external interrupt request to the CPU.

Each pin can independently generate an interrupt request. The interrupt sensitivity is independently programmable using the sensitivity bits in the Miscellaneous register.

Each external interrupt vector is linked to a dedicated group of I/O port pins (see pinout description and interrupt section). If several input pins are selected simultaneously as interrupt source, these are logically NANDed. For this reason if one of the interrupt pins is tied low, it masks the other ones.

In case of a floating input with interrupt configuration, special care must be taken when changing the configuration (see *Figure 17*).

The external interrupts are hardware interrupts, which means that the request latch (not accessible directly by the application) is automatically cleared when the corresponding interrupt vector is fetched. To clear an unwanted pending interrupt by software, the sensitivity bits in the Miscellaneous register must be modified.

Output mode

The output configuration is selected by setting the corresponding DDR register bit. In this case, writing the DR register applies this digital value to the I/O pin through the latch. Then reading the DR register returns the previously stored value.

Two different output modes can be selected by software through the OR register: Output push-pull and open-drain.

DR register value and output pin status:

DR	Push-pull	Open-drain
0	Vss	Vss
1	VDD	Floating

Alternate function

When an on-chip peripheral is configured to use a pin, the alternate function is automatically selected.

This alternate function takes priority over the standard I/O programming.

When the signal is coming from an on-chip peripheral, the I/O pin is automatically configured in output mode (push-pull or open drain according to the peripheral).

When the signal is going to an on-chip peripheral, the I/O pin must be configured in input mode. In this case, the pin state is also digitally readable by addressing the DR register.

Note: Input pull-up configuration can cause unexpected value at the input of the alternate peripheral input. When an on-chip peripheral useS a pin as input and output, this pin has to be configured in input floating mode.

Figure 16. I/O Port general block diagram

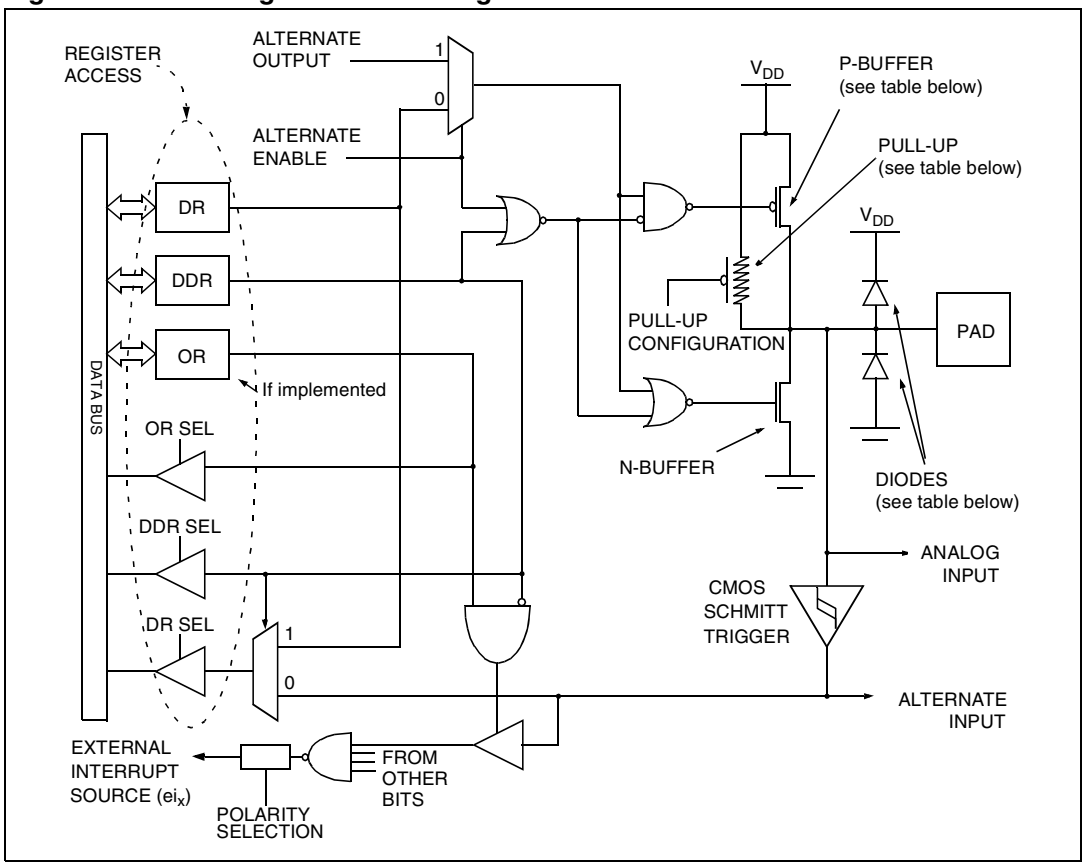


Table 6. I/O Port mode options

Configuration Mode		Pull-Up	P-Buffer	Diodes	
				to V_{DD}	to V_{SS}
Input	Floating with/without Interrupt	Off	Off	On	On
	Pull-up with/without Interrupt	On			
Output	Push-pull	Off	On		
	Open Drain (logic level)		Off		
	True Open Drain	NI	NI	NI (see note)	

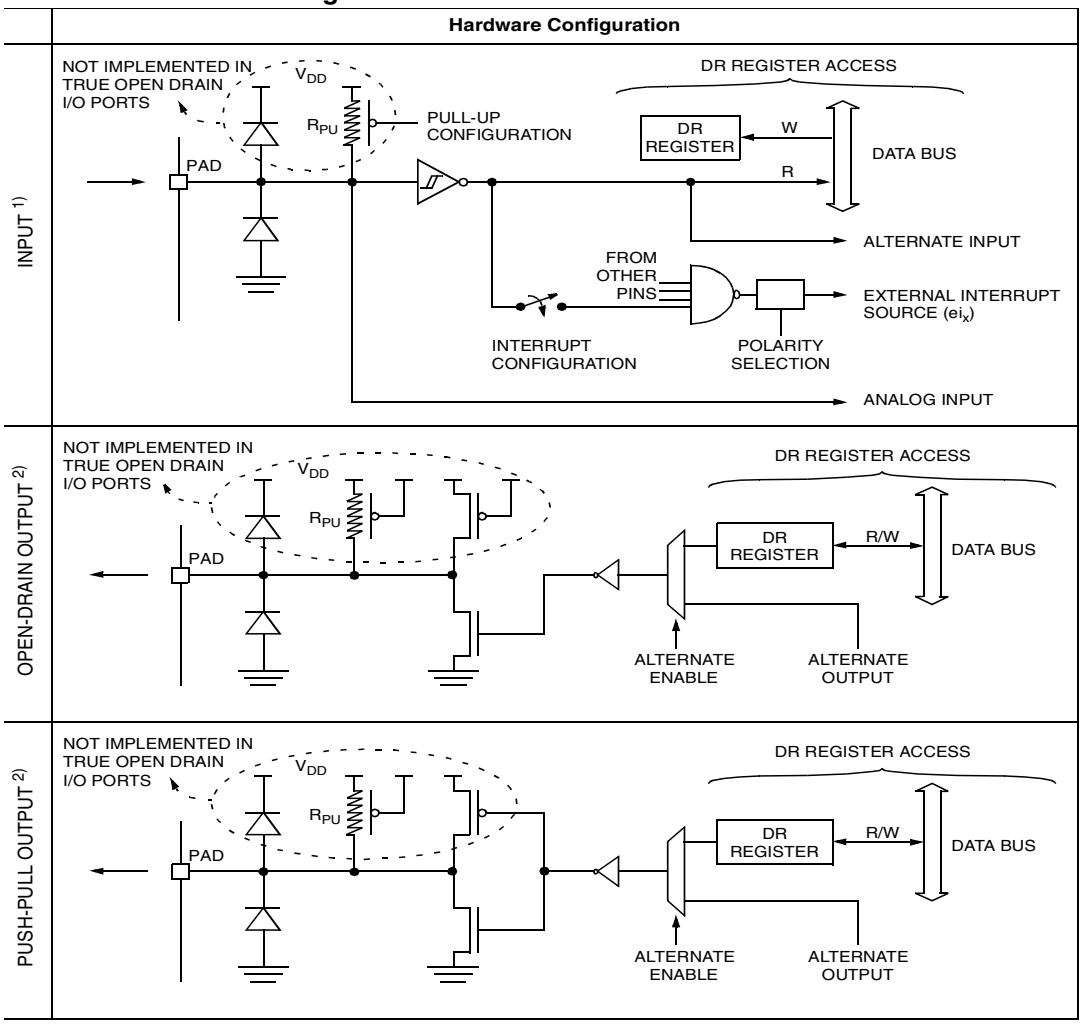
Legend: NI - not implemented

Off - implemented not activated

On - implemented and activated

Note: The diode to V_{DD} is not implemented in the true open drain pads. A local protection between the pad and V_{SS} is implemented to protect the device against positive stress.

Table 7. I/O Port configurations



Note: 1 1. When the I/O port is in input configuration and the associated alternate function is enabled as an output, reading the DR register will read the alternate function output status.

2 2. When the I/O port is in output configuration and the associated alternate function is enabled as an input, the alternate function reads the pin status given by the DR register content.

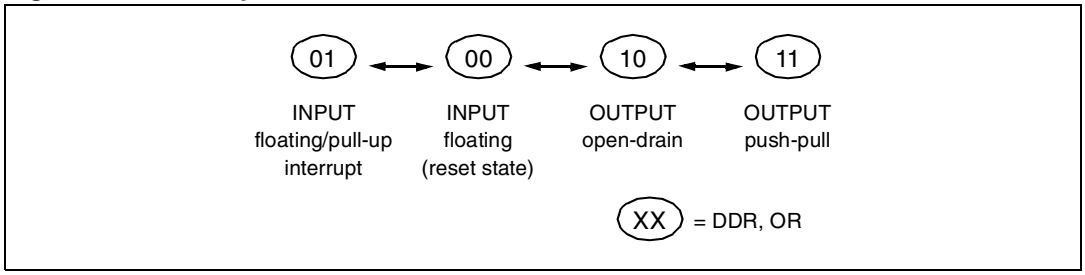
Caution: The alternate function must not be activated as long as the pin is configured as input with interrupt, in order to avoid generating spurious interrupts.

Warning: The analog input voltage level must be within the limits stated in the absolute maximum ratings.

The hardware implementation on each I/O port depends on the settings in the DDR and OR registers and specific feature of the I/O port.Switching these I/O ports from one state to another should be done in a sequence that prevents unwanted side effects. Recommended safe transitions are illustrated in *Figure 17*. Other transitions are potentially risky and should

be avoided, since they are likely to present unwanted side-effects such as spurious interrupt generation.

Figure 17. Interrupt I/O Port state transitions



I/O Port implementation

The I/O port register configurations are resumed as following.

Port PA(7:0), Port PB(2:0)

DDR	OR	MODE
0	0	input no interrupt (pull-up enabled)
0	1	input interrupt (pull-up enabled)
1	0	Open-Drain output
1	1	Push-Pull output

RESET status: DR=0, DDR=0 and OR=0 (Input mode, no interrupt).

These ports offer interrupt capabilities.

Dedicated configurations

Table 8. Port A configuration

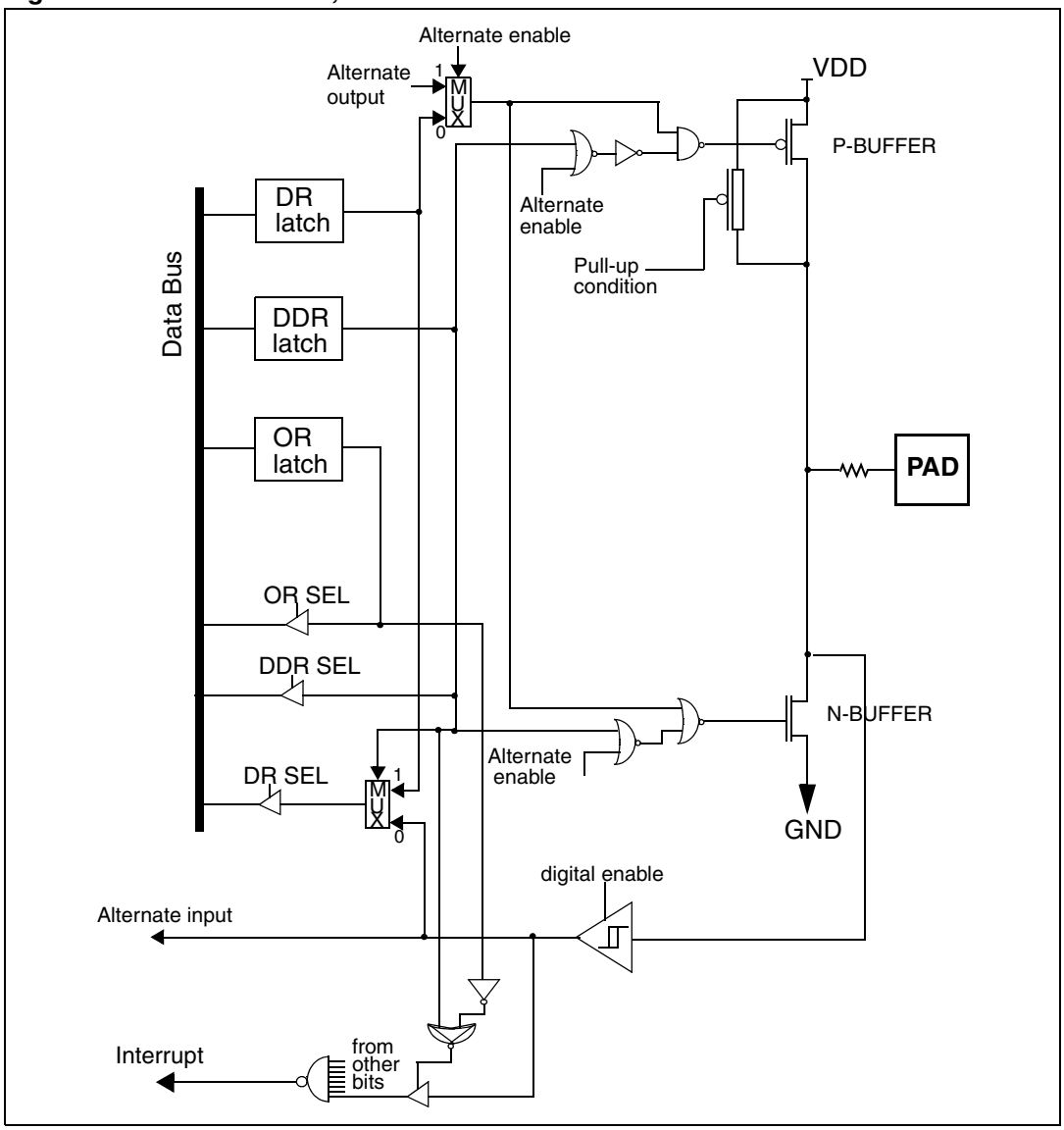
PORT A	I / O		Function	
	Input	Output	Alternate	Interrupt
PA0	triggered with pull-up	push-pull/open drain	OCMP2_1: Output Compare #2 Timer 1	wake-up interrupt (I0)
PA1	triggered with pull-up	push-pull/open drain	OCMP1_1: Output Compare #1 Timer 1	wake-up interrupt (I0)
PA2	triggered with pull-up	push-pull/open drain	ICAP2_1: Input Capture #2 Timer 1	wake-up interrupt (I0)
PA3	triggered with pull-up	push-pull/open drain	ICAP1_1: Input Capture #1 Timer 1	wake-up interrupt (I0)
PA4	triggered with pull-up	push-pull/open drain	EXTCLK_1: External Clock Timer 1	wake-up interrupt (I0)
PA5	triggered with pull-up	push-pull/open drain	OCMP2_2: Output Compare #2 Timer 2	wake-up interrupt (I0)
PA6	triggered with pull-up	push-pull/open drain	OCMP1_2: Output Compare #1 Timer 2	wake-up interrupt (I0)
PA7	triggered with pull-up	push-pull/open drain	ICAP2_2: Input Capture #2 Timer 2	wake-up interrupt (I0)

Table 9. Port B configuration

PORT B	I / O		Function	
	Input	Output	Alternate	Interrupt
PB0	triggered with pull-up	push-pull/open drain	ICAP1_2: Input Capture #1 Timer 2	wake-up interrupt (I0)
PB1	triggered with pull-up	push-pull/open drain	EXTCLK_2: External Clock Timer 2	wake-up interrupt (I0)
PB2[1]	Not connected to pad	Not connected to pad	PWMI: PWM input	

1. The PB2 bit is not connected to the external. It must be configured as an Input without interrupt, to be used only as an alternate function.

Figure 18. Ports PA0-PA7, PB0-PB1I



5.1.3 Register description

Data registers

(PADR)

Port A: 0000h

Read/Write
Reset Value: 0000 0000 (00h)

7							0
MSB							LSB

(PBDR)

Port B: 0004h

Read/Write
Reset Value: 0000 0000 (00h)

7							0
MSB	0	0	0	0			LSB

Data direction registers

(PADDR)

Port A: 0001h

Read/Write
Reset Value: 0000 0000 (00h) (input mode)

7							0
MSB							LSB

(PBDDR)

Port B: 0005h

Read/Write
Reset Value: 0000 0000 (00h) (input mode)

7							0
MSB	0	0	0	0			LSB

Option registers

(PAOR)

Port A: 0002h

Read/Write
Reset Value: 0000 0000 (00h) (no interrupt)

7							0
MSB							LSB

(PBOR)

Port B: 0006h

Read/Write
Reset Value: 0000 0000 (00h) (no interrupt)

7							0
MSB	0	0	0	0			LSB

5.2 16-Bit timer

5.2.1 Introduction

The timer consists of a 16-bit free-running counter driven by a programmable prescaler.

It may be used for a variety of purposes, including pulse length measurement of up to two input signals (*input capture*) or generation of up to two output waveforms (*output compare* and *PWM*).

Pulse lengths and waveform periods can be modulated from a few microseconds to several milliseconds using the timer prescaler and the CPU clock prescaler.

5.2.2 Main features

- Programmable prescaler: f_{cpu} divided by 2, 4 or 8.
- Overflow status flag and maskable interrupt
- External clock input (must be at least 4 times slower than the CPU clock speed) with the choice of active edge
- Output compare functions with
  - 2 dedicated 16-bit registers
  - 2 dedicated programmable signals
  - 2 dedicated status flags
  - 1 dedicated maskable interrupt
- Input capture functions with
  - 2 dedicated 16-bit registers
  - 2 dedicated active edge selection signals
  - 2 dedicated status flags
  - 1 dedicated maskable interrupt
- Pulse width modulation mode (PWM)
- One pulse mode
- 5 alternate functions on I/O ports

The Block Diagram is shown in *Figure 19 on page 43*.

Note: Some external pins are not available on all devices. Refer to the device pin out description.

5.2.3 Functional description

Counter

The main block of the Programmable Timer is a 16-bit free running upcounter and its associated 16-bit registers. The 16-bit registers are made up of two 8-bit registers called high & low.

Counter Register (CR):

- Counter High Register (CHR) is the most significant byte (MS Byte).
- Counter Low Register (CLR) is the least significant byte (LS Byte).

Alternate Counter Register (ACR)

- Alternate Counter High Register (ACHR) is the most significant byte (MS Byte).
- Alternate Counter Low Register (ACLR) is the least significant byte (LS Byte).

These two read-only 16-bit registers contain the same value but with the difference that reading the ACLR register does not clear the TOF bit (Timer overflow flag), located in the Status register (SR). (See note at the end of paragraph titled 16-bit read sequence).

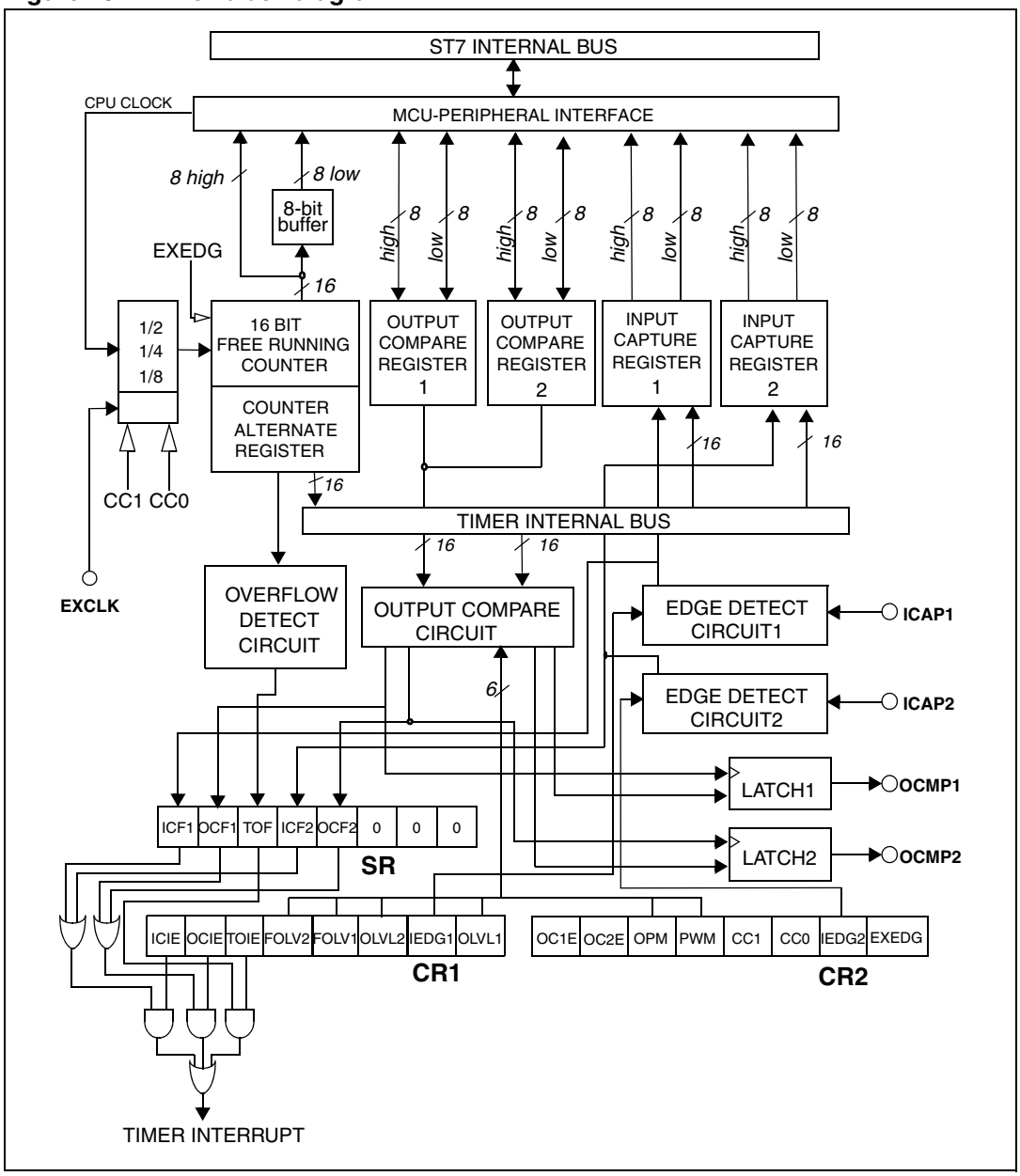
Writing in the CLR register or ACLR register resets the free running counter to the FFFCh value.

Both counters have a reset value of FFFCh (this is the only value which is reloaded in the 16-bit timer). The reset value of both counters is also FFFCh in One Pulse mode and PWM mode.

The timer clock depends on the clock control bits of the CR2 register, as illustrated in *Table 10: Clock Control Bits*. The value in the counter register repeats every 131072, 262144 or 524288 CPU clock cycles depending on the CC[1:0] bits.

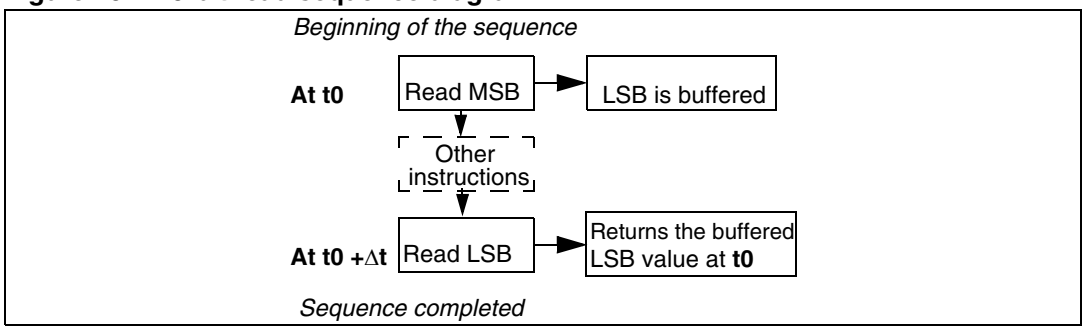
The timer frequency can be $f_{CPU}/2$, $f_{CPU}/4$, $f_{CPU}/8$ or an external frequency.

Figure 19. Timer block diagram



16-bit read sequence: (from either the Counter Register or the Alternate Counter Register)

Figure 20. 16-bit read sequence diagram.



The user must read the MSB first, then the LSB value is buffered automatically.

This buffered value remains unchanged until the 16-bit read sequence is completed, even if the user reads the MSB several times.

After a complete reading sequence, if only the CLR register or ACLR register are read, they return the LSB of the count value at the time of the read.

An overflow occurs when the counter rolls over from FFFFh to 0000h then:

- The TOF bit of the SR register is set.
- A timer interrupt is generated if:
  - TOIE bit of the CR1register is set and
  - I bit of the CCR register is cleared.

If one of these conditions is false, the interrupt remains pending to be issued as soon as they are both true.

Clearing the overflow interrupt request is done by:

1. Reading the SR register while the TOF bit is set.
2. An access (read or write) to the CLR register.

Note: The TOF bit is not cleared by accesses to ACLR register. This feature allows simultaneous use of the overflow function and reads of the free running counter at random times (for example, to measure elapsed time) without the risk of clearing the TOF bit erroneously.

The timer is not affected by WAIT mode.

In HALT mode, the counter stops counting until the mode is exited. Counting then resumes from the previous count (MCU awakened by an interrupt) or from the reset count (MCU awakened by a Reset).

External clock

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

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

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

At least 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 21. Counter timing diagram, internal clock divided by 2

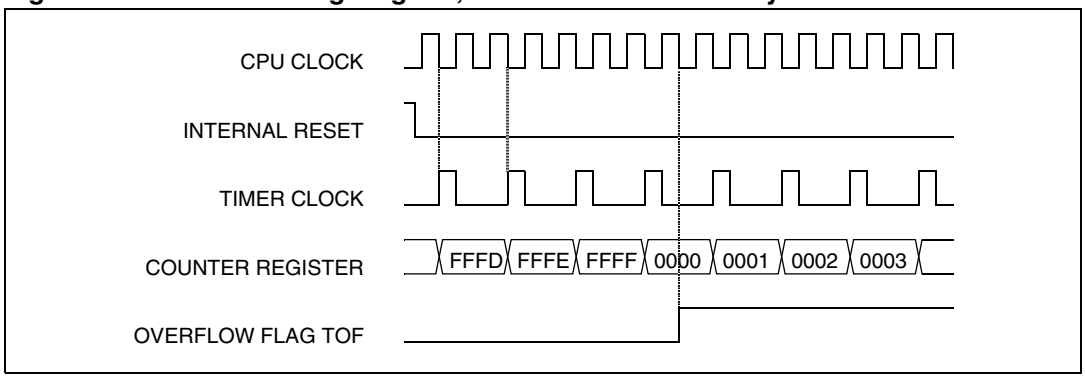


Figure 22. Counter timing diagram, internal clock divided by 4

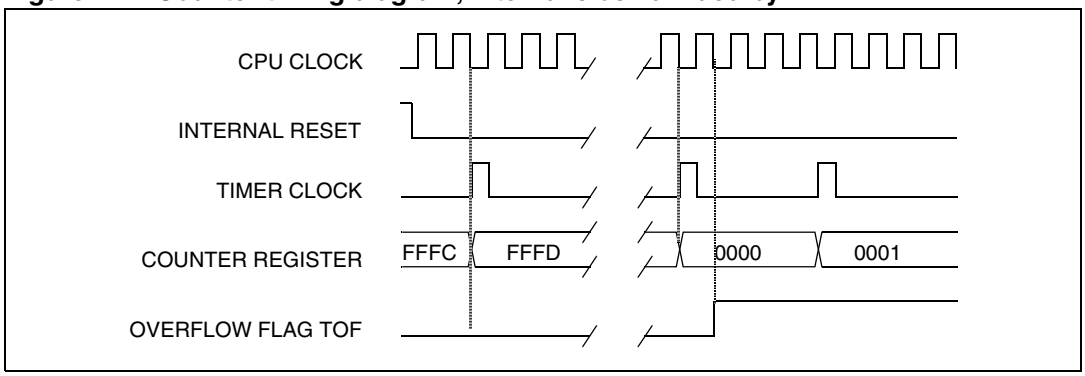
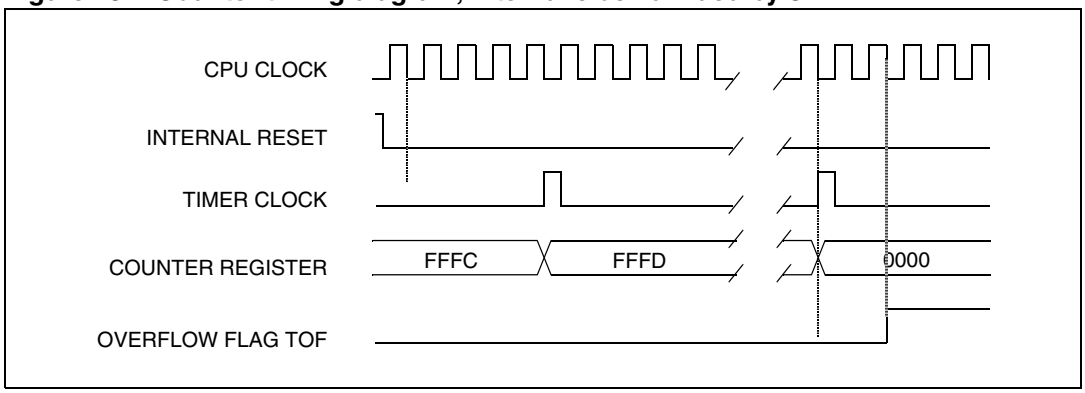


Figure 23. Counter timing diagram, internal clock divided by 8



Input capture

In this section, the index, *i*, may be 1 or 2, because there are 2 input capture functions in the 16 bit timer.

The two input capture 16-bit registers (ICR1 and ICR2) are used to latch the value of the free running counter after a transition detected by the ICAP*i* pin (see *Figure 19*)

	MS Byte	LS Byte
ICR*i*	ICHR*i*	ICLR*i*

ICR*i* register is a read-only register.

The active transition is software programmable through the IEDG*i* bit of the Control Register (CR*i*).

Timing resolution is one count of the free running counter: ($f_{CPU/(CC[1:0])}$).

Procedure

To use the input capture function select the following in the CR2 register:

- Select the timer clock (CC[1:0]) (see *Table 10: Clock Control Bits*).
- Select the edge of the active transition on the ICAP2 pin with the IEDG2 bit.

And select the following in the CR1 register:

- Set the ICIE bit to generate an interrupt after an input capture.
- Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit.

When an input capture occurs:

- ICF*i* bit is set.
- The ICR*i* register contains the value of the free running counter on the active transition on the ICAP*i* pin (see *Figure 24*).
- A timer interrupt is generated if the ICIE bit is set and the I bit is cleared in the CCR register. Otherwise, the interrupt remains pending until both conditions become true.

Clearing the Input Capture interrupt request is done by:

1. Reading the SR register while the ICF*i* bit is set.
2. An access (read or write) to the ICLR*i* register.

Note: After reading the ICHRi register, transfer of input capture data is inhibited until the ICLRi register is also read.

The ICR*i* register always contains the free running counter value which corresponds to the most recent input capture.

During HALT mode, if at least one valid input capture edge occurs on the ICAP*i* pin, the input capture detection circuitry is armed. This does not set any timer flags, and does not “wake-up” the MCU. If the MCU is awoken by an interrupt, the input capture flag will become active, and data corresponding to the first valid edge during HALT mode will be present.

Figure 24. Input capture block diagram

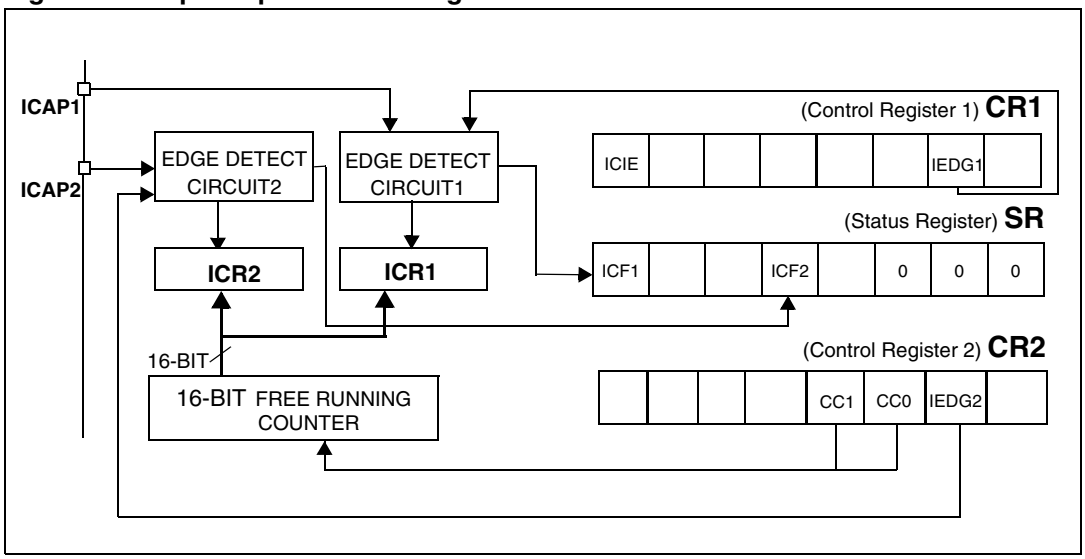
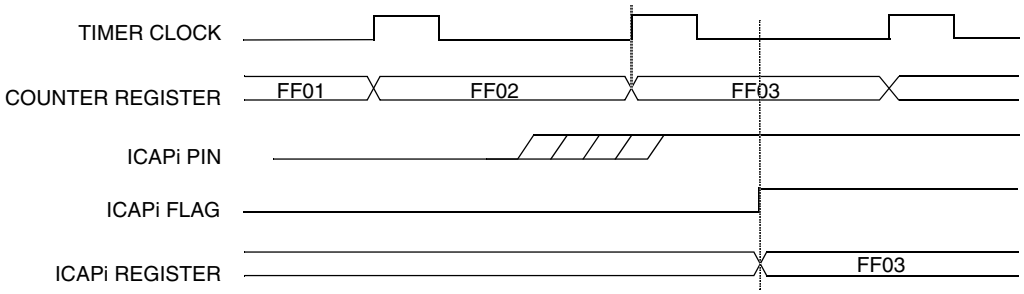


Figure 25. Input capture timing diagram



Note: Active edge is rising edge.

Output Compare

In this section, the index, *i*, may be 1 or 2 because there are 2 output compare functions in the 16-bit timer.

This function can be used to control an output waveform or indicate when a period of time has elapsed.

When a match is found between the Output Compare register and the free running counter, the output compare function:

- Assigns pins with a programmable value if the OC*i*E bit is set
- Sets a flag in the status register
- Generates an interrupt if enabled

Two 16-bit registers Output Compare Register 1 (OCR1) and Output Compare Register 2 (OCR2) contain the value to be compared to the counter register each timer clock cycle.

	MS Byte	LS Byte
OC*i*R	OC*i*HR	OC*i*LR

These registers are readable and writable and are not affected by the timer hardware. A reset event changes the OC*i*R value to 8000h.

Timing resolution is one count of the free running counter: ($f_{CPU/CC[1:0]}$).

Procedure

To use the output compare function, select the following in the CR2 register:

- Set the OC*i*E bit if an output is needed, then the OCMP*i* pin is dedicated to the output compare *i* function.
- Select the timer clock (CC[1:0]) (see *Table 10: Clock Control Bits*).

And select the following in the CR1 register:

- Select the OLVL*i* bit to be applied to the OCMP*i* pins after the match occurs.
- Set the OCIE bit to generate an interrupt if it is needed.

When a match is found between OCRi register and CR register:

- OCF*i* bit is set.
- The OCMP*i* pin takes OLVL*i* bit value (OCMP*i* pin latch is forced low during reset).
- A timer interrupt is generated if the OCIE bit is set in the CR1 register and the I bit is cleared in the Condition Code register (CC).

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

$$\Delta OCRi = \frac{\Delta t \cdot f_{CPU}}{PRESC}$$

Where:

Δt = Output compare period (in seconds)

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

PRESC = Timer prescaler factor (2, 4 or 8 depending on CC[1:0] bits, see *Table 10: Clock Control Bits*)

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

$$\Delta OCRi = \Delta t \cdot f_{EXT}$$

Where:

Δt = Output compare period (in seconds)

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

Clearing the output compare interrupt request (i.e.clearing the OCFi bit) is done by:

1. Reading the SR register while the OCFi bit is set.
2. An access (read or write) to the OCLRi register.

The following procedure is recommended to prevent the OCF*i* bit from being set between the time it is read and the write to the OCR*i* register:

– Write to the OCHR*i* register (further compares are inhibited).
– Read the SR register (first step of the clearance of the OCF*i* bit, which may be already set).
– Write to the OCLR*i* register (enables the output compare function and clears the OCF*i* bit).

Figure 26. Output compare block diagram

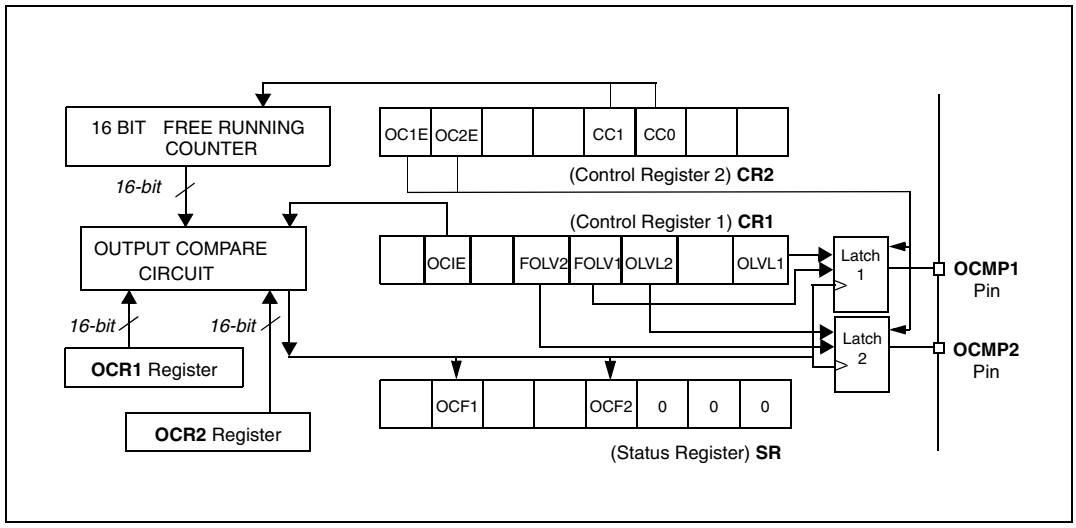
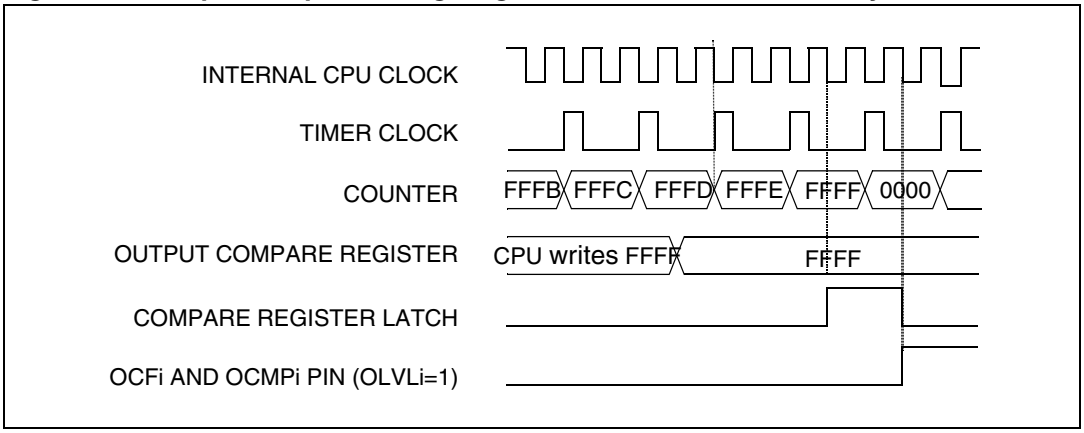


Figure 27. Output compare timing diagram, internal clock divided by 2



Forced compare output capability

The main purpose of the forced compare output capability is the easily generate a fixed frequency. The following bits of the CR1 register are used:

– When the FOLV*i* bit is set by software, the OLVL*i* bit is copied to the OCMP*i* pin.
– The OLV*i* bit has to be toggled in order to toggle the OCMP*i* pin when it is enabled (OC*i*E bit=1).
– The OCF*i* bit is then not set by hardware, and thus no interrupt request is generated.
– FOLVL*i* bits have no effect in either One-Pulse mode or PWM mode.

One pulse mode

One Pulse mode enables the generation of a pulse when an external event occurs. This mode is selected via the OPM bit in the CR2 register.

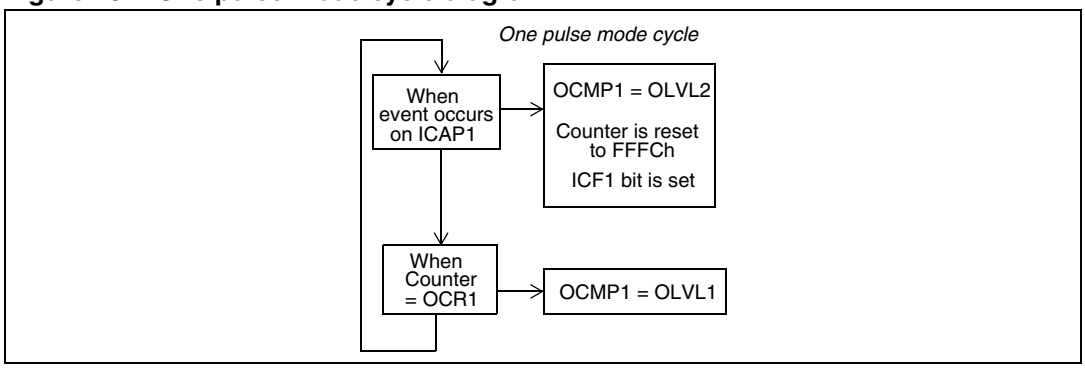
The One Pulse mode uses the Input Capture1 function and the Output Compare1 function.

Procedure

To use One Pulse mode:

1. Load the OCR1 register with the value corresponding to the length of the pulse (see the formula in the section).
2. Select the following in the CR1 register:
   - Using the OLVL1 bit, select the level to be applied to the OCMP1 pin after the pulse.
   - Using the OLVL2 bit, select the level to be applied to the OCMP1 pin during the pulse.
   - Select the edge of the active transition on the ICAP1 pin with the IEDG1 bit (the ICAP1 pin must be configured as floating input).
3. Select the following in the CR2 register:
   - Set the OC1E bit, the OCMP1 pin is then dedicated to the Output Compare 1 function.
   - Set the OPM bit.
   - Select the timer clock CC[1:0] (see *Table 10: Clock Control Bits*).

Figure 28. One pulse mode cycle diagram.



Then, on a valid event on the ICAP1 pin, the counter is initialized to FFFCh and the OLVL2 bit is loaded on the OCMP1 pin, the ICF1 bit is set and the value FFFDh is loaded in the IC1R register.

Because the ICF1 bit is set when an active edge occurs, an interrupt can be generated if the ICIE bit is set.

Clearing the Input Capture interrupt request (i.e. clearing the ICFi bit) is done in two steps:

1. Reading the SR register while the ICFi bit is set.
2. An access (read or write) to the ICLRi register.

The OCR1 register value required for a specific timing application can be calculated using the following formula:

$$OCRi\ Value = \frac{t \cdot f_{CPU}}{PRESC} - 5$$

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 10: Clock Control Bits*).

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

Where:

t = Pulse period (in seconds)

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

When the value of the counter is equal to the value of the contents of the OCR1 register, the OLVL1 bit is output on the OCMP1 pin (See <Blue HT>Figure 29).

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 can not be used to perform input capture. The ICAP2 pin can be used to perform input capture (ICF2 can be set and ICR2 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.

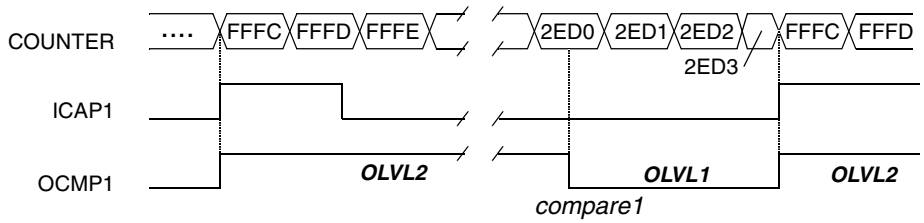
5 When One Pulse mode is used OCR1 is dedicated to this mode. Nevertheless OCR2 and OCF2 can be used to indicate that a period of time has elapsed but cannot generate an output waveform because the OLVL2 level is dedicated to One Pulse mode.

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

The ICF1 bit is set when an active edge occurs and can generate an interrupt if the ICIE bit is set.

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

Figure 29. One pulse mode timing



Note: IEDG1=1, OCR1=2ED0h, OLVL1=0, OLVL2=1

Pulse width modulation mode

Pulse Width Modulation (PWM) mode enables the generation of a signal with a frequency and pulse length determined by the value of the OCR1 and OCR2 registers.

The Pulse Width Modulation mode uses the complete Output Compare 1 function plus the OC2R register, and so these functions cannot be used when the PWM mode is activated.

Procedure

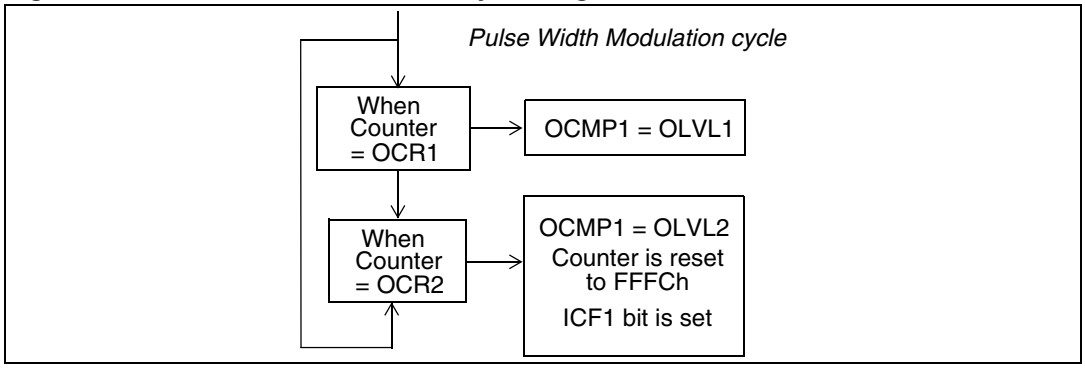
To use Pulse Width Modulation mode:

1. Load the OCR2 register with the value corresponding to the period of the signal using the formula in the section.
2. Load the OCR1 register with the value corresponding to the period of the pulse if OLVL1=0 and OLVL2=1, using the formula in the section.
3. Select the following in the CR1 register:
   - Using the OLVL1 bit, select the level to be applied to the OCMP1 pin after a successful comparison with OCR1 register.
   - Using the OLVL2 bit, select the level to be applied to the OCMP1 pin after a successful comparison with OC2R register.
4. Select the following in the CR2 register:
   - Set OC1E bit: the OCMP1 pin is then dedicated to the output compare 1 function.
   - Set the PWM bit.
   - Select the timer clock (CC[1:0]) (see *Table 10: Clock Control Bits*).

If OLVL1=1 and OLVL2=0, the length of the positive pulse is the difference between the OCR2 and OCR1 registers.

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

Figure 30. Pulse width modulation cycle diagram



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

$$\text{OCR}i\ \text{Value} = \frac{t \cdot f_{CPU}}{PRESC} - 5$$

Where:

t = Signal or pulse period (in seconds)

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

PRESC = Timer prescaler factor (2, 4 or 8 depending on CC[1:0] bits, see *Table 10: Clock Control Bits*

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

$$OCRi = t \cdot f_{EXT} - 5$$

Where:

t = Signal or pulse period (in seconds)

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

The Output Compare 2 event causes the counter to be initialized to FFFCh (See *Figure 31 on page 53*).

Note: 1 After a write instruction to the OCHRi register, the output compare function is inhibited until the OCiLR register is also written.

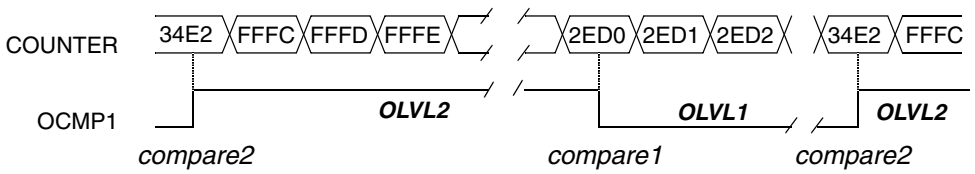
2 The OCF1 and OCF2 bits cannot be set by hardware in PWM mode, therefore the Output Compare interrupt is inhibited.

3 The ICF1 bit is set by hardware when the counter reaches the OC2R value and can produce a timer interrupt if the ICIE bit is set and the I bit is cleared.

4 In PWM mode the ICAP1 pin can not be used to perform input capture because it is disconnected from the timer. 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 after each period and ICF1 can also generate an interrupt if ICIE is set.

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

Figure 31. Pulse width modulation mode timing



Note: OCR1=2ED0h, OCR2=34E2h, OLVL1=0, OLVL2= 1

5.2.4 Register description

Each Timer is associated with three control and status registers, and with six pairs of data registers (16-bit values) relating to the two input captures, the two output compares, the counter and the alternate counter.

CONTROL REGISTER 1 (CR1)

Timer1 Register Address: 0032h

Timer2 Register Address: 0042h

Read/Write

Reset Value: 0000 0000 (00h)

7							0
ICIE	OCIE	TOIE	FOLV2	FOLV1	OLVL2	IEDG1	OLVL1

Bit 7 = **ICIE** *Input Capture Interrupt Enable.*

0: Interrupt is inhibited.

1: A timer interrupt is generated whenever the ICF1 or ICF2 bits of the SR register are set

Bit 6 = **OCIE** *Output Compare Interrupt Enable.*

0: Interrupt is inhibited.

1: A timer interrupt is generated whenever the OCF1 or OCF2 bits of the SR register are set

Bit 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.

Bit 4 = **FOLV2** *Forced Output Compare 2.*

0: No effect.

1: Forces the OLVL2 bit to be copied to the OCMP2 pin.

Bit 3 = **FOLV1** *Forced Output Compare 1.*

0: No effect.

1: Forces OLVL1 to be copied to the OCMP1 pin.

Bit 2 = **OLVL2** *Output Level 2.*
This bit is copied to the OCMP2 pin whenever a successful comparison occurs with the OCR2 register. This value is copied to the OCMP1 pin in One Pulse Mode and Pulse Width Modulation mode.

Bit 1 = **IEDG1** *Input Edge 1.*
This bit determines which type of level transition on the ICAP1 pin will trigger the capture.

0: A falling edge triggers the capture.

1: A rising edge triggers the capture.

Bit 0 = **OLVL1** *Output Level 1.*
The OLVL1 bit is copied to the OCMP1 pin whenever a successful comparison occurs with the OCR1 register.

CONTROL REGISTER 2 (CR2)

Timer1 Register Address: 0031h

Timer2 Register Address: 0041h

Read/Write

Reset Value: 0000 0000 (00h)

7							0
OC1E	OC2E	OPM	PWM	CC1	CC0	IEDG2	EXEDG

Bit 7 = **OC1E** *Output Compare 1 Enable.*

0: Output Compare 1 function is enabled, but the OCMP1 pin is a general I/O.

1: Output Compare 1 function is enabled, the OCMP1 pin is dedicated to the Output Compare 1 capability of the timer.

Bit 6 = **OC2E** *Output Compare 2 Enable.*

0: Output Compare 2 function is enabled, but the OCMP2 pin is a general I/O.

1: Output Compare 2 function is enabled, the OCMP2 pin is dedicated to the Output Compare 2 capability of the timer.

Bit 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 OCR1 register.

Bit 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 OCR1 register; the period depends on the value of OCR2 register.

Bit 3, 2 = **CC1-CC0** *Clock Control.*

The value of the timer clock depends on these bits:

Table 10. Clock Control Bits

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

Bit 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.

Bit 0 = **EXEDG** *External Clock Edge.*
This bit determines which type of level transition on the external clock pin EXCLK will trigger the free running counter.

0: A falling edge triggers the free running counter.

1: A rising edge triggers the free running counter.

STATUS REGISTER (SR)

Timer1 Register Address: 0033h

Timer2 Register Address: 0043h

Read Only

Reset Value: 0000 0000 (00h)

The three least significant bits are not used.

7							0
ICF1	OCF1	TOF	ICF2	OCF2			

Bit 7 = **ICF1** *Input Capture Flag 1.*

0: No input capture (reset value)

1: An input capture has occurred. To clear this bit, first read the SR register, then read or write the low byte of the ICR1 (ICLR1) register.

Bit 6 = **OCF1** *Output Compare Flag 1.*

0: No match (reset value)

1: The content of the free running counter has matched the content of the OCR1 register. To clear this bit, first read the SR register, then read or write the low byte of the OCR1 (OCLR1) register.

Bit 5 = **TOF** *Timer Overflow.*

0: No timer overflow (reset value)

1: The free running counter rolled over from FFFFh to 0000h. To clear this bit, first read the SR register, then read or write the low byte of the CR (CLR) register.

Note: Reading or writing the ACLR register do not clear TOF.

Bit 4 = **ICF2** *Input Capture Flag 2.*

0: No input capture (reset value)

1: An input capture has occurred.To clear this bit, first read the SR register, then read or write the low byte of the ICR2 (ICLR2) register.

Bit 3 = **OCF2** *Output Compare Flag 2.*

0: No match (reset value)

1: The content of the free running counter has matched the content of the OCR2 register. To clear this bit, first read the SR register, then read or write the low byte of the OCR2 (OCLR2) register.

Bit 2-0 = Unused.

INPUT CAPTURE 1 HIGH REGISTER (ICHR1)

Timer1 Register Address: 0034h

Timer2 Register Address: 0044h

Read Only
Reset Value: Undefined

This is an 8-bit read only register that contains the high part of the counter value (transferred by the input capture 1 event).

7							0
MSB							LSB

INPUT CAPTURE 1 LOW REGISTER (ICLR1)

Timer1 Register Address: 0035h

Timer2 Register Address: 0045h

Read Only
Reset Value: Undefined

This is an 8-bit read only register that contains the low part of the counter value (transferred by the input capture 1 event).

7							0
MSB							LSB

OUTPUT COMPARE 1 HIGH REGISTER (OCHR1)

Timer1 Register Address: 0036h

Timer2 Register Address: 0046h

Read/Write
Reset Value: 1000 0000 (80h)

This is an 8-bit register that contains the high part of the value to be compared to the CHR register.

7							0
MSB							LSB

OUTPUT COMPARE 1 LOW REGISTER (OCLR1)

Timer1 Register Address: 0037h

Timer2 Register Address: 0047h

Read/Write
Reset Value: 0000 0000 (00h)

This is an 8-bit register that contains the low part of the value to be compared to the CLR register.

7							0
MSB							LSB

OUTPUT COMPARE 2 HIGH REGISTER (OCHR2)

Timer1 Register Address: 003Eh

Timer2 Register Address: 004Eh

Read/Write
Reset Value: 1000 0000 (80h)

This is an 8-bit register that contains the high part of the value to be compared to the CHR register.

7							0
MSB							LSB

OUTPUT COMPARE 2 LOW REGISTER (OCLR2)

Timer1 Register Address: 003Fh

Timer2 Register Address: 004Fh

Read/Write
Reset Value: 0000 0000 (00h)

This is an 8-bit register that contains the low part of the value to be compared to the CLR register.

7							0
MSB							LSB

COUNTER HIGH REGISTER (CHR)

Timer1 Register Address: 0038h

Timer2 Register Address: 0048h

Read Only
Reset Value: 1111 1111 (FFh)

This is an 8-bit register that contains the high part of the counter value.

7							0
MSB							LSB

COUNTER LOW REGISTER (CLR)

Timer1 Register Address: 0039h

Timer2 Register Address: 0049h

Read/Write
Reset Value: 1111 1100 (FCh)

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 accessing the SR register clears the TOF bit.16-BIT.

7							0
MSB							LSB

ALTERNATE COUNTER HIGH REGISTER (ACHR)

Timer1 Register Address: 003Ah

Timer2 Register Address: 004Ah

Read Only
Reset Value: 1111 1111 (FFh)

This is an 8-bit register that contains the high part of the counter value.

7							0
MSB							LSB

ALTERNATE COUNTER LOW REGISTER (ACLR)

Timer1 Register Address: 003Bh

Timer2 Register Address: 004Bh

Read/Write
Reset Value: 1111 1100 (FCh)

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 SR register does not clear the TOF bit in SR register.

7							0
MSB							LSB

INPUT CAPTURE 2 HIGH REGISTER (ICHR2)

Timer1 Register Address: 003Ch

Timer2 Register Address: 004Ch

Read Only
Reset Value: Undefined

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

7							0
MSB							LSB

INPUT CAPTURE 2 LOW REGISTER (ICLR2)

Timer1 Register Address: 003Dh

Timer2 Register Address: 004Dh

Read Only
Reset Value: Undefined

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

7							0
MSB							LSB

Table 11. 16-Bit Timer Register Map and Reset Values

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
Timer1: 32 Timer2: 42	CR1 Reset Value	ICIE 0	OCIE 0	TOIE 0	FOLV2 0	FOLV1 0	OLVL2 0	IEDG1 0	OLVL1 0
Timer1: 31 Timer2: 41	CR2 Reset Value	OC1E 0	OC2E 0	OPM 0	PWM 0	CC1 0	CC0 0	IEDG2 0	EXEDG 0
Timer1: 33 Timer2: 43	SR Reset Value	ICF1 0	OCF1 0	TOF 0	ICF2 0	OCF2 0	- 0	- 0	- 0
Timer1: 34 Timer2: 44	ICHR1 Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer1: 35 Timer2: 45	ICLR1 Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer1: 36 Timer2: 46	OCHR1 Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer1: 37 Timer2: 47	OCLR1 Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer1: 3E Timer2: 4E	OCHR2 Reset Value	MSB -	-	-	-	-	-	-	LSB -

Table 11. 16-Bit Timer Register Map and Reset Values (continued)

Address (Hex.)	Register Name	7	6	5	4	3	2	1	0
Timer1: 3F Timer2: 4F	OCLR2 Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer1: 38 Timer2: 48	CHR Reset Value	MSB 1	1	1	1	1	1	1	LSB 1
Timer1: 39 Timer2: 49	CLR Reset Value	MSB 1	1	1	1	1	1	0	LSB 0
Timer1: 3A Timer2: 4A	ACHR Reset Value	MSB 1	1	1	1	1	1	1	LSB 1
Timer1: 3B Timer2: 4B	ACLR Reset Value	MSB 1	1	1	1	1	1	0	LSB 0
Timer1: 3C Timer2: 4C	ICHR2 Reset Value	MSB -	-	-	-	-	-	-	LSB -
Timer1: 3D Timer2: 4D	ICLR2 Reset Value	MSB -	-	-	-	-	-	-	LSB -

5.3 PWM generator

5.3.1 Introduction

This PWM peripheral includes a 16-bit Pulse Width Modulator (PWM) and a programmable prescaler able to generate an internal clock with period as long as $128*T_{CPU}$.

The repetition rate of the 16-Bit PWM output can be defined by a dedicated register (f_{CPU}/CYREG); its resolution is defined by the internal clock as per the prescaler programming.

Main features

- Programmable prescaler: f_{CPU} divided by 2, 4, 8, 16, 32, 64 or 128.
- 1 control register
- 2 dedicated 16-bit registers for cycle and duty control
- 1 dedicated maskable interrupt

Procedure

To use the pulse width modulation peripheral, the EN_PWM bit in CONREG register must be set.

Load PS(2:0) in CONREG register to define the programmable prescaler.

Load the CYREG register with the value defining the cycle length (in internal clock periods). The 16 bits of this register are separated in two registers: CYREGH and CYREGL.

Load the DUTYREG register with the value corresponding to the pulse length (in internal cycle periods). The 16 bits of this register are separated in two registers: DUTYREGH and DUTYREGL.

The counter is reset to zero when EN_PWM bit is reset.

Writing the DUTYREG and CYREG registers has no effect on the current PWM cycle. The cycle or duty cycle change take place only after the first overflow of the counter.

The suggested procedures to change the PWM parameters are the following:

Duty Cycle control:

– Write the low and high DUTYREG registers.

A writing only on one DUTYREG register has no effect until both registers are written.

The current PWM cycle will be completed. The new duty cycle will be effective at the following PWM cycle, with respect to the last DUTYREG writing.

Cycle control:

– Write the low and high CYREG register

A writing only on one CYREG register has no effect until both registers are written.

The current PWM cycle will be completed. The new cycle will be effective at the following PWM cycle, with respect to the last CYREG writing.

Another possible procedure is:

– Reset the EN_PWM bit.
– Write the wanted configuration in CYREG and DUTYREG.
– Set the EN_PWM bit.

If the EN_PWM bit is set after being reset, the current values of DUTYREG and CYREG are determining the output waveform, no matter if only the low or the high part, or both were written.

The first time EN_PWM is set, if CYREG and DUTYREG were not previously written, the output is permanently low, because the default value of the registers is 00h.

Changing the Prescaler ratio writing PS(2:0) in CONREG has immediate effect on the waveform frequency.

5.3.2 Functional description

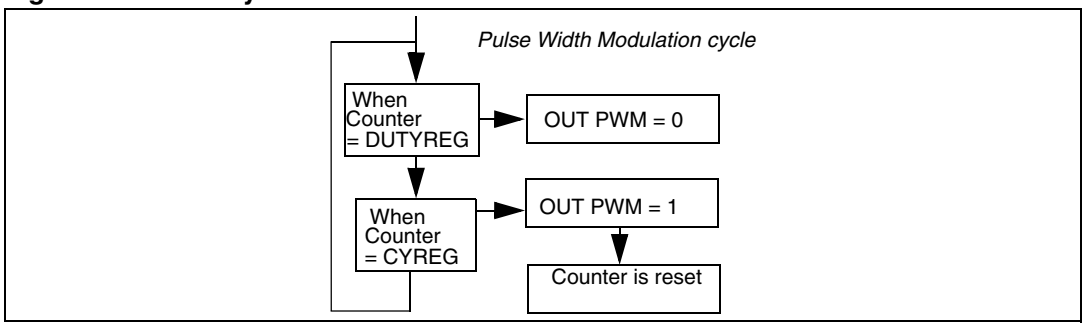
The PWM module consists of a 16-bit counter, a comparator and the cycle generation logic.

PWM generation

The counter increments continuously, clocked at internal clock generated by prescaler. Whenever the 16 bits of the counter (defined as the PWM counter) overflow, the output level is set. The overflow value is defined by CYREG register.

The state of the PWM counter is continuously compared to the PWM binary weight, as defined in DUTYREG register, and when a match occurs the output level is reset.

Figure 32. PWM cycle



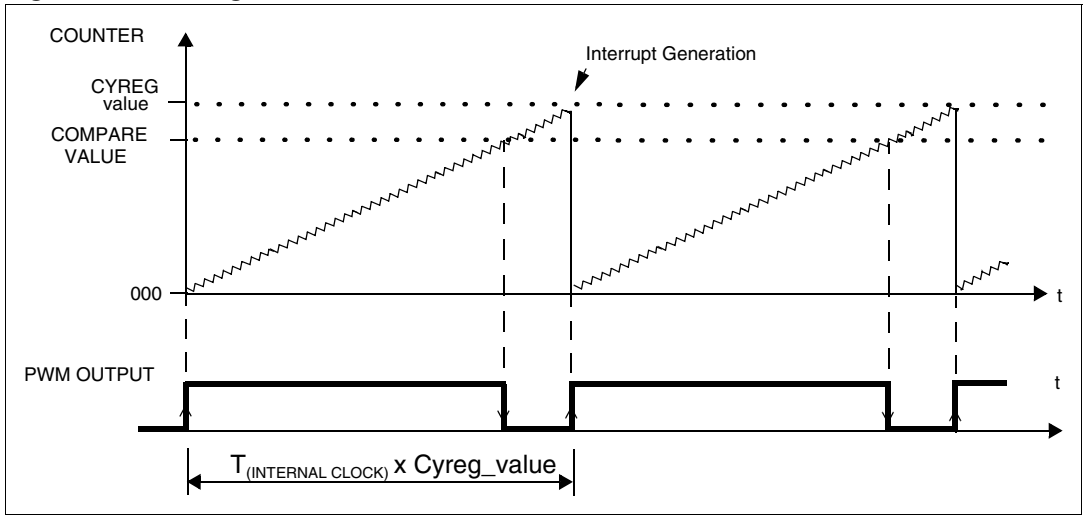
Note: If the CYREG value is minor or equal than DUTYREG value, PWM output remains set. With a DUTYREG value of 0000h, the PWM output is permanently at low level, no matter of the value of CYREG. With a DUTYREG value of FFFFh, the PWM output is permanently at high level.

Interrupt request

The EN_INT bit in CONREG register must be set to enable the interrupt generation. When the 16 bits of the counter roll-over CYCLEREG value, interrupt request is set.

The interrupt request is cleared when any of the PWM registers is written.

Figure 33. PWM generation



5.3.3 Register description

The PWM is associated with a 8-bit control registers, and with two 16-bit data registers, each split in two 8-bit registers.

PWM CYCLE REGISTER LOW (CYREGL)

PWM1 Register Address: 0011h

PWM2 Register Address: 0019h

Read/Write
Reset Value: 0000 0000 (00h)

This is an 8-bit register that contains the low part of the value to be multiplied by internal clock period.

7							0
MSB							LSB

PWM CYCLE REGISTER HIGH (CYREGH)

PWM1 Register Address: 0010h

PWM2 Register Address: 0018h

Read/Write
Reset Value: 0000 0000 (00h)

This is an 8-bit register that contains the high part of the value to be multiplied by internal clock period.

7							0
MSB							LSB

PWM DUTYCYCLE REGISTER LOW (DUTYREGL)

PWM1 Register Address: 0013h

PWM2 Register Address: 001Bh

Read/Write
Reset Value: 0000 0000 (00h)

This is an 8-bit register that contains the low part of the value corresponding to the binary weight of the PWM pulse.

7							0
MSB							LSB

PWM DUTYCYCLE REGISTER HIGH (DUTYREGH)

PWM1 Register Address: 0012h

PWM2 Register Address: 001Ah

Read/Write
Reset Value: 0000 0000 (00h)

This is an 8-bit register that contains the high part of the value corresponding to the binary weight of the PWM pulse.

7							0
MSB							LSB

PWM CONTROL REGISTER (CONREG)

PWM1 Register Address: 0014h

PWM2 Register Address: 001Ch

Read/Write
Reset Value: 0000 0000 (00h)

7			4	3	2	1	0
0	0	0	PS2	PS1	PS0	EN_ INT	EN_ PWM

Bit 0= **EN _PWM**: 1 = enables the PWM output, 0 = disables PWM output.

Bit 1= **EN _INT**: 1 = enables interrupt request, 0 disables interrupt request.

Bit 4, 3, 2= **PS2**,**PS1**,**PS0**: prescaler bits

The value of the PWM internal clock depends on these bits.

PS2	PS1	PS0	PWM internal clock
0	0	0	f_{CPU}
0	0	1	f_{CPU} / 2
0	1	0	f_{CPU} / 4
0	1	1	f_{CPU} / 8
1	0	0	f_{CPU} / 16
1	0	1	f_{CPU} / 32
1	1	0	f_{CPU} / 64
1	1	1	f_{CPU} / 128

Bit 5, 6, 7= not used.

PWM COUNTER REGISTER LOW (CTL)

PWM1 Register Address: 0016h

PWM2 Register Address: 001Eh

Read Only
Reset Value: 0000 0000 (00h)

This is an 8-bit register that contains the low part of the PWM counter value.

7							0
MSB							LSB

PWM COUNTER REGISTER HIGH (CTH)

PWM1 Register Address: 0015h

PWM2 Register Address: 001Dh

Read Only
Reset Value: 0000 0000 (00h)

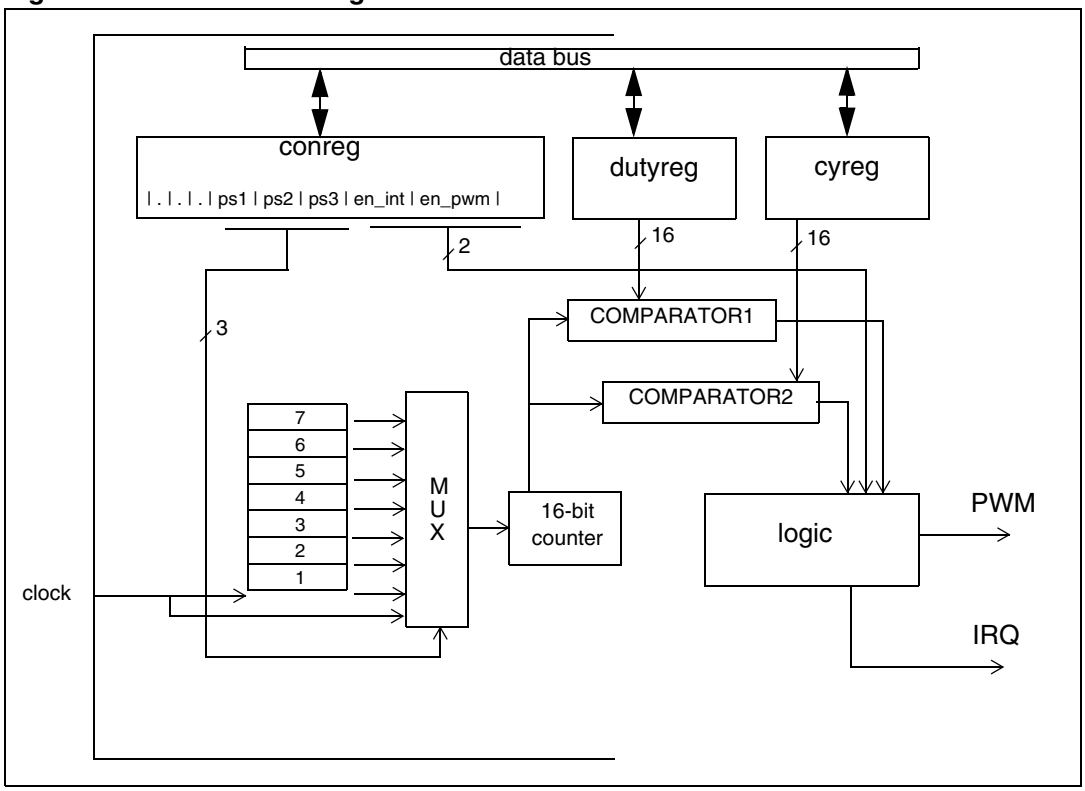
This is an 8-bit register that contains the high part of the PWM counter value.

7							0
MSB							LSB

Table 12. PWM Timing (f_{CPU} = 8MHz)

Prescaler (PS)	$T_{internal\ clock}$	CYREG @16 bit Resolution	PWM $_{cycle}$ @ f_{in}=8MHz
0	$1/f_{in} * 2^{ps}$	$1/f_{in} * 2^{ps} * 1 ..... 1/f_{in} * 2^{ps} * 65535$	0.125 µs..... ~8192 µs
1	$1/f_{in} * 2^{ps}$	$1/f_{in} * 2^{ps} * 1 ..... 1/f_{in} * 2^{ps} * 65535$	0.25 µs..... ~16384 µs
2	$1/f_{in} * 2^{ps}$	$1/f_{in} * 2^{ps} * 1 ..... 1/f_{in} * 2^{ps} * 65535$	0.5 µs..... ~32768 µs
3	$1/f_{in} * 2^{ps}$	$1/f_{in} * 2^{ps} * 1 ..... 1/f_{in} * 2^{ps} * 65535$	1 µs..... 65535 µs
4	$1/f_{in} * 2^{ps}$	$1/f_{in} * 2^{ps} * 1 ..... 1/f_{in} * 2^{ps} * 65535$	2 µs..... 131070 µs
5	$1/f_{in} * 2^{ps}$	$1/f_{in} * 2^{ps} * 1 ..... 1/f_{in} * 2^{ps} * 65535$	4 µs..... 262140 µs
6	$1/f_{in} * 2^{ps}$	$1/f_{in} * 2^{ps} * 1 ..... 1/f_{in} * 2^{ps} * 65535$	8 µs..... 524280 µs
7	$1/f_{in} * 2^{ps}$	$1/f_{in} * 2^{ps} * 1 ..... 1/f_{in} * 2^{ps} * 65535$	16 µs..... 1048560 µs

Figure 34. PWM Block Diagram



5.4 PWM I/O, K line transceiver

5.4.1 Introduction

The PWM I/O interface is a circuit able to connect internal logic circuits with external high voltage lines.

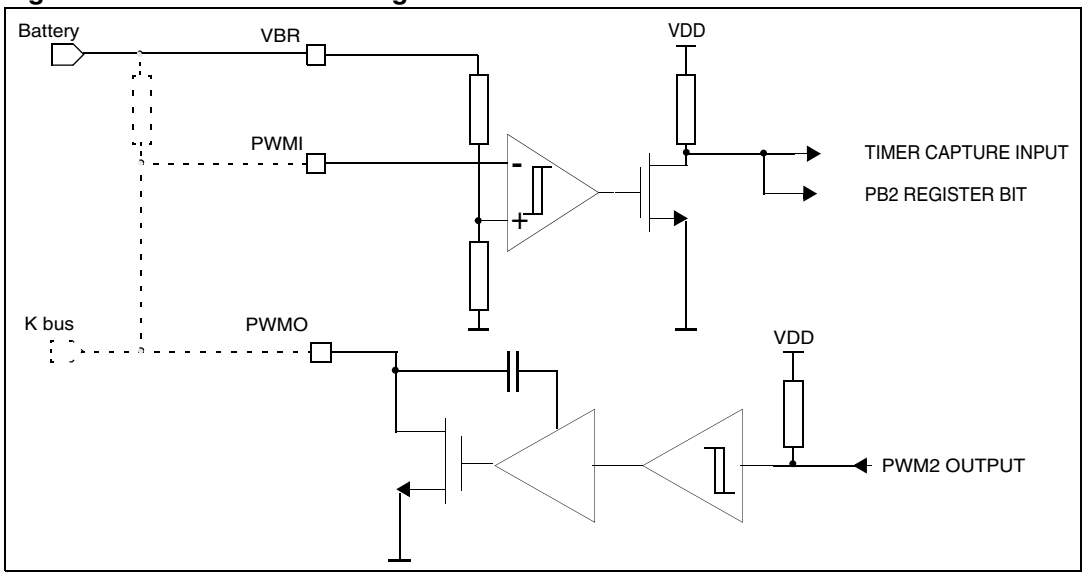
The two interfaces represent respectively the receiver and the transmitter section of a standard IS0 9141 transceiver.

Connecting PWMO and PWMI together a standard K bus (ISO 9141) can be realized.

Voltage thresholds are referred to the battery voltage connected to VBR pin. This pin must be used as reference for the K bus. Voltage drops between this pin and the battery line can cause thresholds mismatch between the L9805E ISO transceiver and the counterpart transceiver(s) connected to the same bus line.

See *Figure 35* for a block diagram description of the two interfaces.

Figure 35. PWM I/O Block Diagram



5.4.2 PWMO

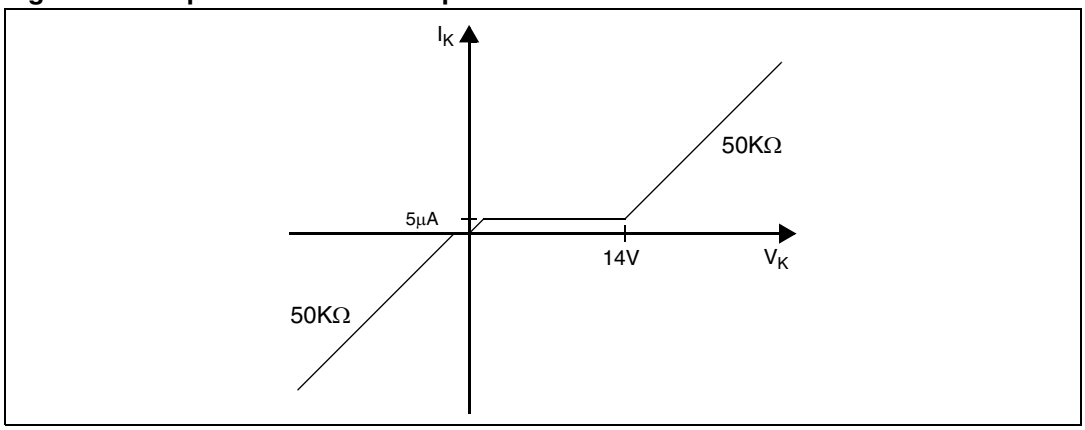
PWMO is an output line, directly driven by the PWM2 output signal. The circuit translates the logic levels of PWM2 output to voltage levels referred to the VB supply (see *Figure 35*). When PWM2=0 the open drain is switched off, in the other case the PWMO line is pulled down by the open drain driver.

PWMO is protected against short circuit to battery by a dedicated circuit that limits the current sunk by the output transistor. When the limiter is activated the voltage on PWMO pin rises up. If the limiter remains active for more than 25µs the driver is switched off.

If the battery or ground connection are lost, the PWMO line shows a controlled impedance characteristic (see *Figure 36*).

PWM0 is high at NRESET is asserted.

Figure 36. Impedance at PWMO/I pin



5.4.3 PWMI

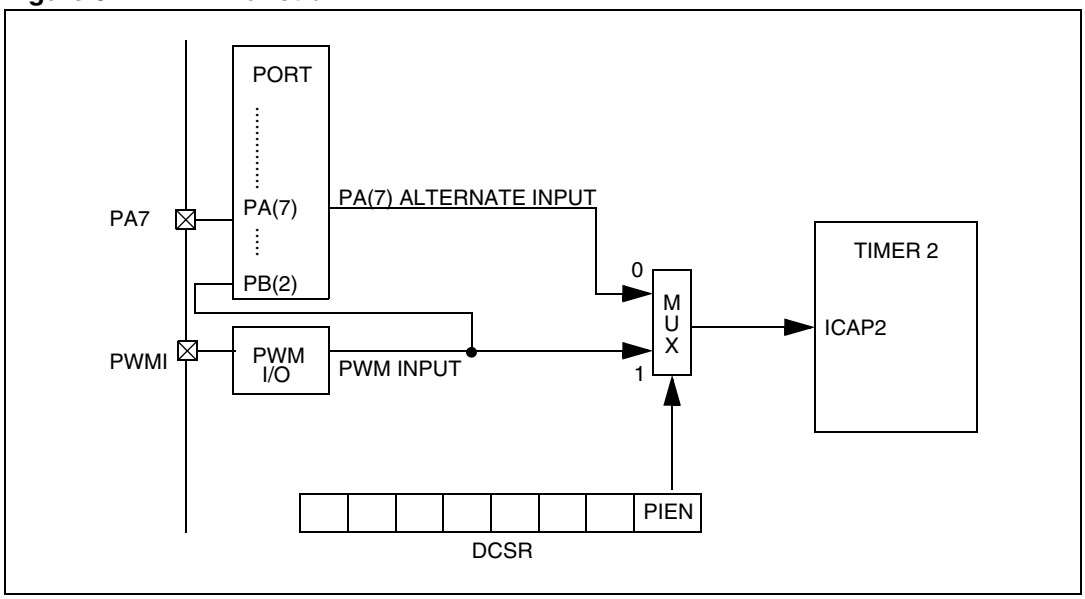
PWMI is an input line, directly connected to PB2 bit. The circuit translates the voltage levels referred to VB voltage supply to the internal logic levels (see *Figure 35*). When the voltage on PWMI pin is higher than VB/2 PB2 reads an high logic level.

If the bit PWMI in DCSR register is set (see *Section 3.2.1*), PWMI is directly connected with the Input Capture 2 on Timer 2, which is otherwise connected in alternate function to PA7 (see *Figure 37*).

An internal pull down current generator (5µA) allows to detect the Open Bus condition (external pull up missing).

If the battery or ground connection are lost, the PWMI line shows a controlled impedance characteristic (see *Figure 37*).

Figure 37. PWMI function



Describe the register DCSR (0022h) as reported in Table 1.

5.5 10-BIT A/D converter (AD10)

5.5.1 Introduction

The Analog to Digital converter is a single 10-bit successive approximation converter with 4 input channels. Analog voltage from external sources are input to the converter through AD2,AD3 and AD4 pins. Channel 1 (AD1) is connected to the internal temperature sensor (see *Section 5.5.5*).

Note: The anti aliasing filtering must be accomplished using an external RC filter. The internal AD1 channel is filtered by an RC network with approx. 1µs time constant.

5.5.2 Functional description

The result of the conversion is stored in 2 registers: the Data Register High (ADCDRH) and the Data Register Low (ADCDRL).

The A/D converter is enabled by setting the ADST bit in ADCCSR Register. Bits CH1 and CH0 of ADCCSR Register select the channel to be converted. The high and low reference voltage are connected to pins VCC and AGND.

When enabled, the A/D converter performs a complete conversion in 14µs (with system clock f_{CPU}=8Mhz). The total conversion time includes multiplex, sampling of the input voltage, 10-bit conversion and writing DRH and DRL registers.

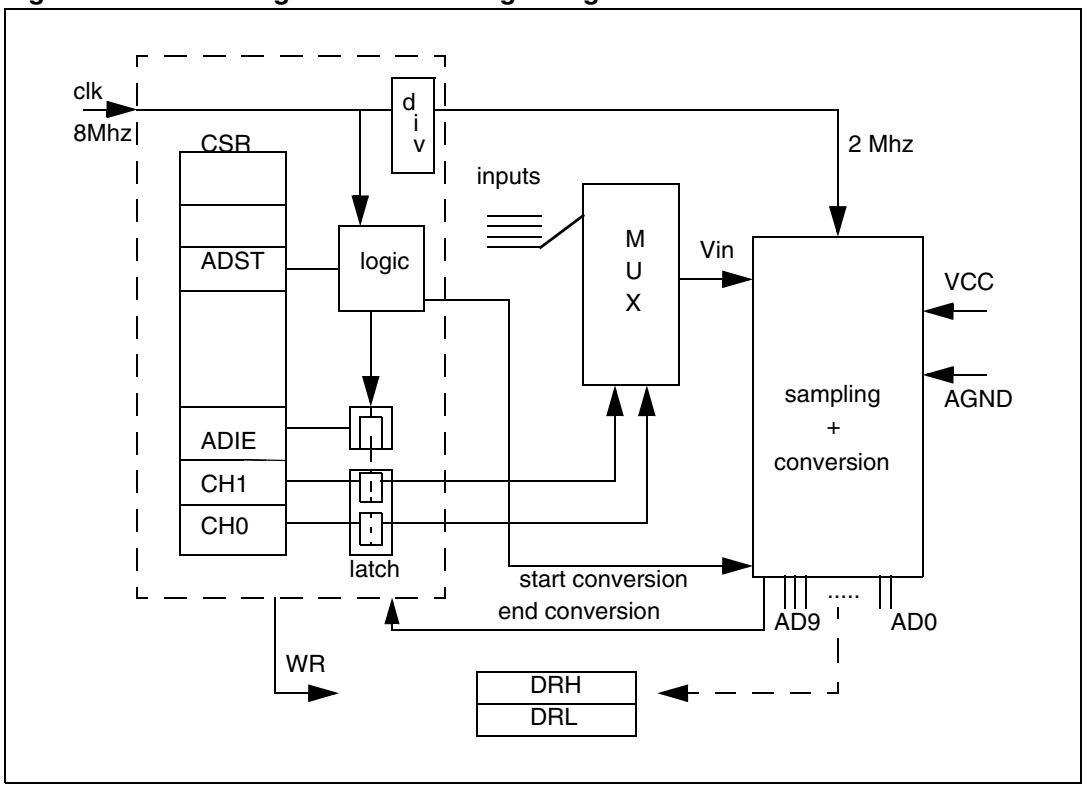
When the conversion is completed COCO bit (COnversion COmpleted) is set in ADCCSR.

A conversion starts from the moment ADST bit is set. When a conversion is running it is possible to write the ADCCSR without stopping the ADC operations, because all the data in ADCCSR are latched when ADST is set. This property allows to select a different channel to be processed during the next conversion or to manage the interrupt enable bit. The new setting will have effect on the next conversion (including interrupt generation)

At the end of the conversion ADST is reset and COCO bit is set.

Note: To start a new conversion the ADST must be set after the completion of the current one. Any writing to ADST when a conversion is running (COCO=0) has no effect since ADST bit is automatically reset by the end of conversion event.

Figure 38. Block diagram of the Analog to Digital Converter



5.5.3 Input Selections and Sampling

The input section of the ADC includes the analog multiplexer and a buffer. The input of the buffer is permanently connected to the multiplexer output. The buffer output is fed to the sample and hold circuit.

The multiplexer is driven with CH1 and CH0 bit only after ADST is set. Starting from this event, the sampler follows the selected input signal for 2.5us and then holds it for the remaining conversion time (i.e. when the conversion is actually running).

5.5.4 Interrupt Management

If ADIE bit is set in register ADCCSR, an interrupt is generated when a conversion is completed (i.e. when COCO is set).

The interrupt request is cleared when any of the ADC registers is accessd (either read or write).

Enabling/disabling the interrupt generation while the conversion is running has no effect on the current conversion. ADIE value is latched when ADST is set and this internal value holds all the conversion time long.

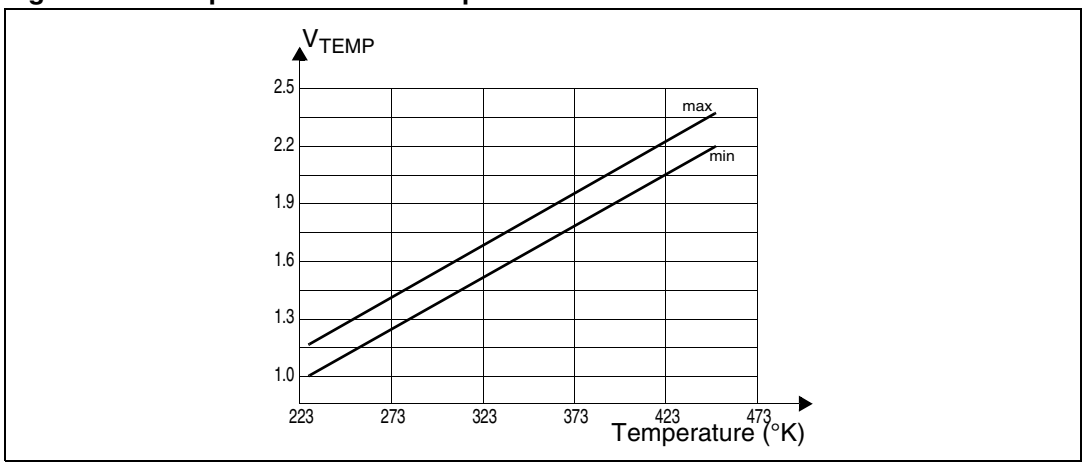
5.5.5 Temperature Sensing

The AD1 input is internally connected to the output of a temperature sensing circuit.

The sensor generates a voltage proportional to the absolute temperature of the die. It works over the whole temperature range, with a minimum resolution of 1LSB/°K (5mV/°K) (*Figure 39* shows the indicative voltage output of the sensor).

Note The voltage output of the sensor is only related to the absolute temperature of the silicon junctions. Junction temperature and ambient temperature must be related taking into account the power dissipated by the device and the thermal resistance $R_{Th\ j\text{-}a}$ between the silicon and the environment around the application board.

Figure 39. Temperature Sensor output



The output of the sensor is not ratiometric with the voltage reference for the ADC conversion (VCC). When calculating the ADC reading error of this signal the variation of VCC must be accounted. Additional errors are due to the intrinsic spread of the sensor characteristic.

5.5.6 Precise Temperature Measurement

To allow a more precise measurement of the temperature a trimming procedure can be adopted (on request).

The temperature is measured in EWS and two values are stored in four EEPROM bytes (see memory map):

T0L,T0H: temperature of the trimming measurement (in Kelvin).

VT0L,VT0H: output value of the ADC corresponding to T0 (in number of LSBs).

The corrected measurement of the temperature **in Kelvin** must be accomplished in the following way:

$$TEMP\ (in\ °K) = VTEMP * T0 / VT0$$

where VTEMP is the output code in LSB of the ADC corresponding to the measurement.

Example:

If the value stored in EEPROM are:

0C7Ch: 01h ->T0H

0C7Dh: 43h ->T0L

0C7Eh: 01h -> VT0H

0C7Fh: 5Ch -> VT0L

T0 = 0143h = 323K (50 Celsius)

VTo = 015Ch = 348 LSB (conversion of 1.7V, sensor output)

and the sensor output is 2V, converted by the ADC in code 0110011001 = 019Ah = 410LSB, the temperature of the chip is

TEMP = 019Ah * 0143h / 015Ch = 017Ch

equivalent to:

TEMP = 410 * 323 / 348 = 380 K = 107 °C

Note: The sensor circuit may have two kind of error: one translating its output characteristic up and down and the other changing its slope. The described trimming recovers only the translation errors but can not recover slope error. After trimming, being T_{TRIM} the trimming temperature, the specified precision can be achieved in the range (T_{TRIM}-80, max[T_{TRIM}+80, 150°C]). Precision is related to the read temperature in Kelvin.

5.5.7 Register description

CONTROL/STATUS REGISTER (ADCCSR)

Address: 0072h — Read/Write

Reset Value: 0010 0000 (20h)

7							0
0	0	COCO	ADIE	0	ADST	CH1	CH0

Bit 7,6 = **Reserved**

Bit 5 =**COCO** *(Read Only) Conversion Complete*

COCO is set (by the ADC) as soon as a conversion is completed (results can be read). COCO is cleared by setting ADST=1 (start of new conversion). If COCO=0 a conversion is running, if COCO=1 no conversion is running.

Bit 4 = **ADIE** *A/D Interrupt Enable*

This bit is used to enable / disable the interrupt function:
0: interrupt disabled
1: interrupt enabled

Bit 3= **Reserved**

Bit 2= **ADST** *Start Conversion*

When this bit is set a new conversion starts. ADST is automatically reset when the conversion is completed (COCO=1).

Bits 1-0 = **CH1-CH0** *Channel Selection*

These bits select the analog input to convert. See *Table 13* for reference.

Table 13. ADC channel selection table

CH1	CH0	Channel
0	0	AD1, Temperature Sensor
0	1	AD2, external input
1	0	AD3, external input
1	1	AD4, external input

DATA REGISTER HIGH (ADCDRH)

Address: 0070h — Read Only

Reset Value: 00000 0000 (00h)

7							0
0	0	0	0	0	0	AD9	AD8

Bit 1:0 = **AD9-AD8** *Analog Converted Value*

This register contains the high part of the converted analog value

DATA REGISTER LOW (ADCDRL)

Address: 0071h — Read Only

Reset Value: 00000 0000 (00h)

7							0
AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0

Bit 7:0 = **AD7-AD0** *Analog Converted Value*

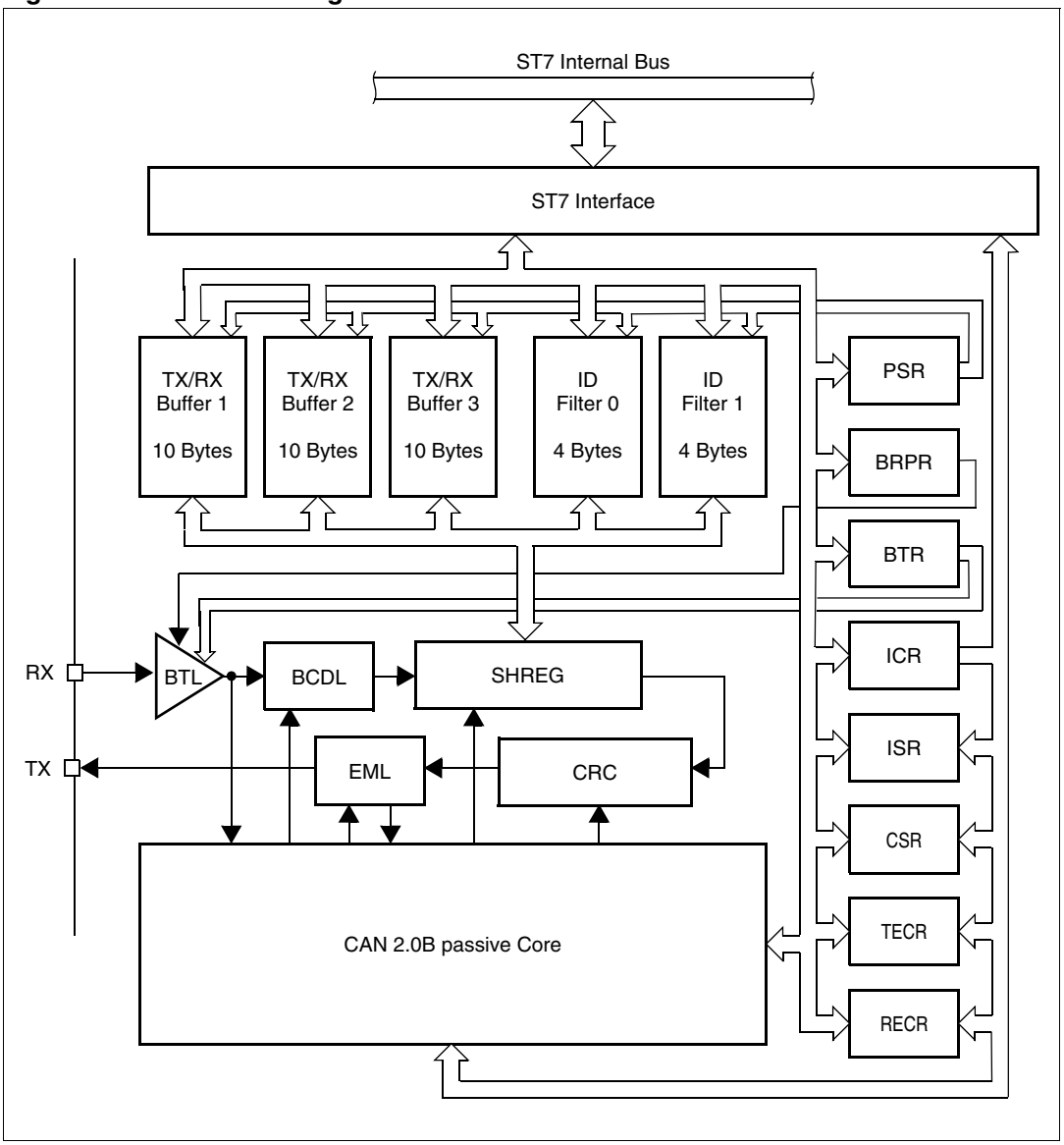
This register contains the low part of the converted analog value

5.6 Controller area network (CAN)

5.6.1 Introduction

This peripheral is designed to support serial data exchanges using a multi-master contention based priority scheme as described in CAN specification Rev. 2.0 part A. It can also be connected to a 2.0 B network without problems, since extended frames are checked for correctness and acknowledged accordingly although such frames cannot be transmitted nor received. The same applies to overload frames which are recognized but never initiated.

Figure 40. CAN block diagram



5.6.2 Main features

- Support of CAN specification 2.0A and 2.0B passive
- Three prioritized 10-byte Transmit/Receive message buffers
- Two programmable global 12-bit message acceptance filters
- Programmable baud rates up to 1 MBit/s
- Buffer flip-flopping capability in transmission
- Maskable interrupts for transmit, receive (one per buffer), error and wake-up
- Automatic low-power mode after 20 recessive bits or on demand (standby mode)
- Interrupt-driven wake-up from standby mode upon reception of dominant pulse
- Optional dominant pulse transmission on leaving standby mode
- Automatic message queuing for transmission upon writing of data byte 7
- Programmable loop-back mode for self-test operation
- Advanced error detection and diagnosis functions
- Software-efficient buffer mapping at a unique address space
- Scalable architecture.

5.6.3 Functional description

Frame formats

A summary of all the CAN frame formats is given in *Figure 41* for reference. It covers only the standard frame format since the extended one is only acknowledged.

A message begins with a start bit called Start Of Frame (SOF). This bit is followed by the arbitration field which contains the 11-bit identifier (ID) and the Remote Transmission Request bit (RTR). The RTR bit indicates whether it is a data frame or a remote request frame. A remote request frame does not have any data byte.

The control field contains the Identifier Extension bit (IDE), which indicates standard or extended format, a reserved bit (r0) and, in the last four bits, a count of the data bytes (DLC). The data field ranges from zero to eight bytes and is followed by the Cyclic Redundancy Check (CRC) used as a frame integrity check for detecting bit errors.

The acknowledgement (ACK) field comprises the ACK slot and the ACK delimiter. The bit in the ACK slot is placed on the bus by the transmitter as a recessive bit (logical 1). It is overwritten as a dominant bit (logical 0) by those receivers which have at this time received the data correctly. In this way, the transmitting node can be assured that at least one receiver has correctly received its message. Note that messages are acknowledged by the receivers regardless of the outcome of the acceptance test.

The end of the message is indicated by the End Of Frame (EOF). The intermission field defines the minimum number of bit periods separating consecutive messages. If there is no subsequent bus access by any station, the bus remains idle.

Hardware blocks

The CAN controller contains the following functional blocks (refer to *Figure 40*):

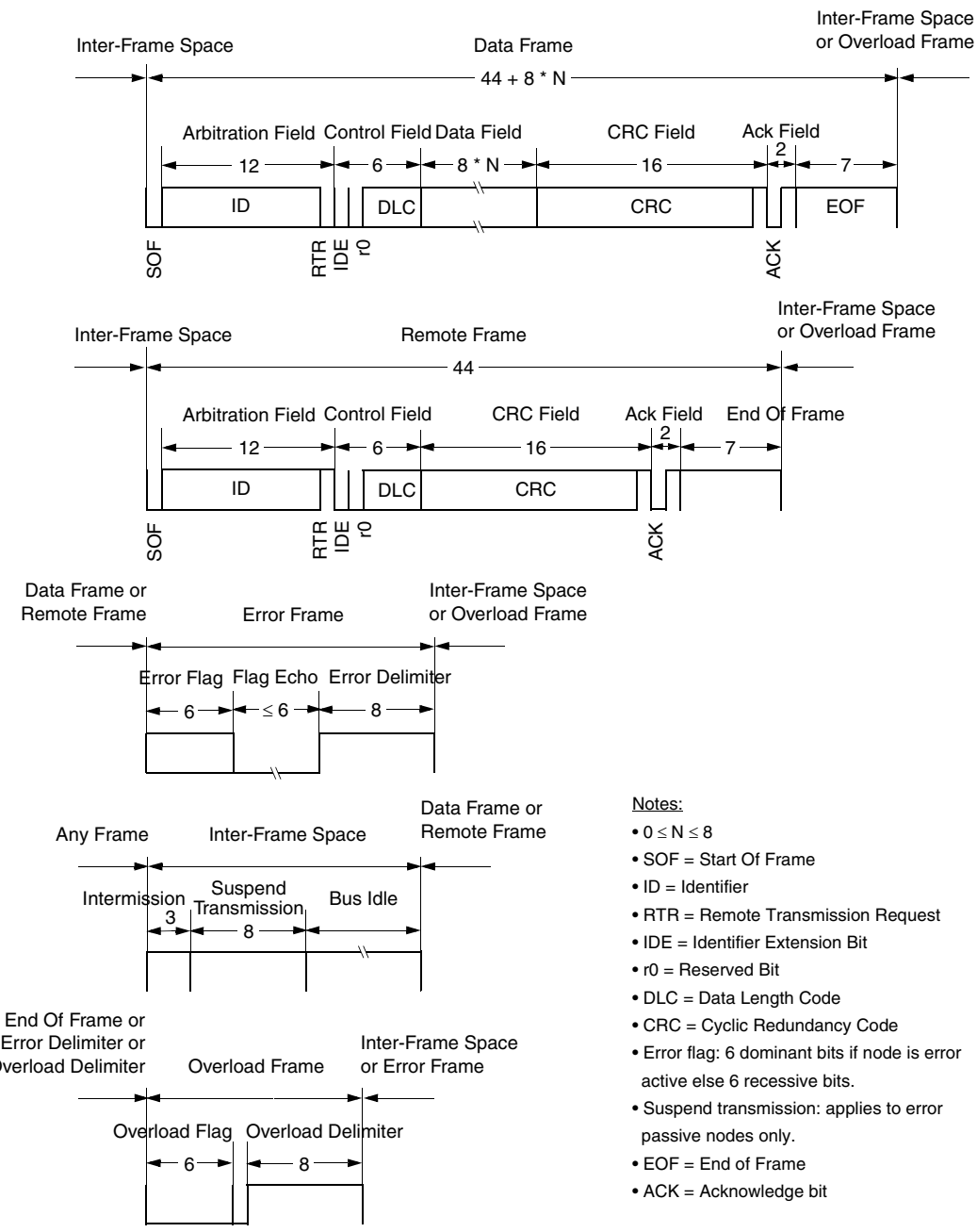
- ST7 Interface: buffering of the ST7 internal bus and address decoding of the CAN registers.
- TX/RX Buffers: three 10-byte buffers for transmission and reception of maximum length messages.
- ID Filters: two 12-bit compare and don't care masks for message acceptance filtering.
- PSR: page selection register (see memory map).
- BRPR: clock divider for different data rates.
- BTR: bit timing register.
- ICR: interrupt control register.
- ISR: interrupt status register.
- CSR: general purpose control/status register.
- TECR: transmit error counter register.
- RECR: receive error counter register.
- BTL: bit timing logic providing programmable bit sampling and bit clock generation for synchronization of the controller.
- BCDL: bit coding logic generating a NRZ-coded datastream with stuff bits.
- SHREG: 8-bit shift register for serialization of data to be transmitted and parallelisation of received data.
- CRC: 15-bit CRC calculator and checker.
- EML: error detection and management logic.
- CAN Core: CAN 2.0B passive protocol controller.

Modes of operation

The CAN Core unit assumes one of the seven states described below:

- **STANDBY**. Standby mode is entered either on a chip reset or on resetting the RUN bit in the Control/Status Register (CSR). Any on-going transmission or reception operation is not interrupted and completes normally before the Bit Time Logic and the clock prescaler are turned off for minimum power consumption. This state is signalled by the RUN bit being read-back as 0.
  Once in standby, the only event monitored is the reception of a dominant bit which causes a wake-up interrupt if the SCIE bit of the Interrupt Control Register (ICR) is set. The STANDBY mode is left by setting the RUN bit. If the WKPS bit is set in the CSR register, then the controller passes through WAKE-UP otherwise it enters RESYNC directly.
  It is important to note that the wake-up mechanism is software-driven and therefore carries a significant time overhead. All messages received after the wake-up bit and before the controller is set to run and has completed synchronization are ignored.
- **WAKE-UP**. The CAN bus line is forced to dominant for one bit time signalling the wake-up condition to all other bus members.

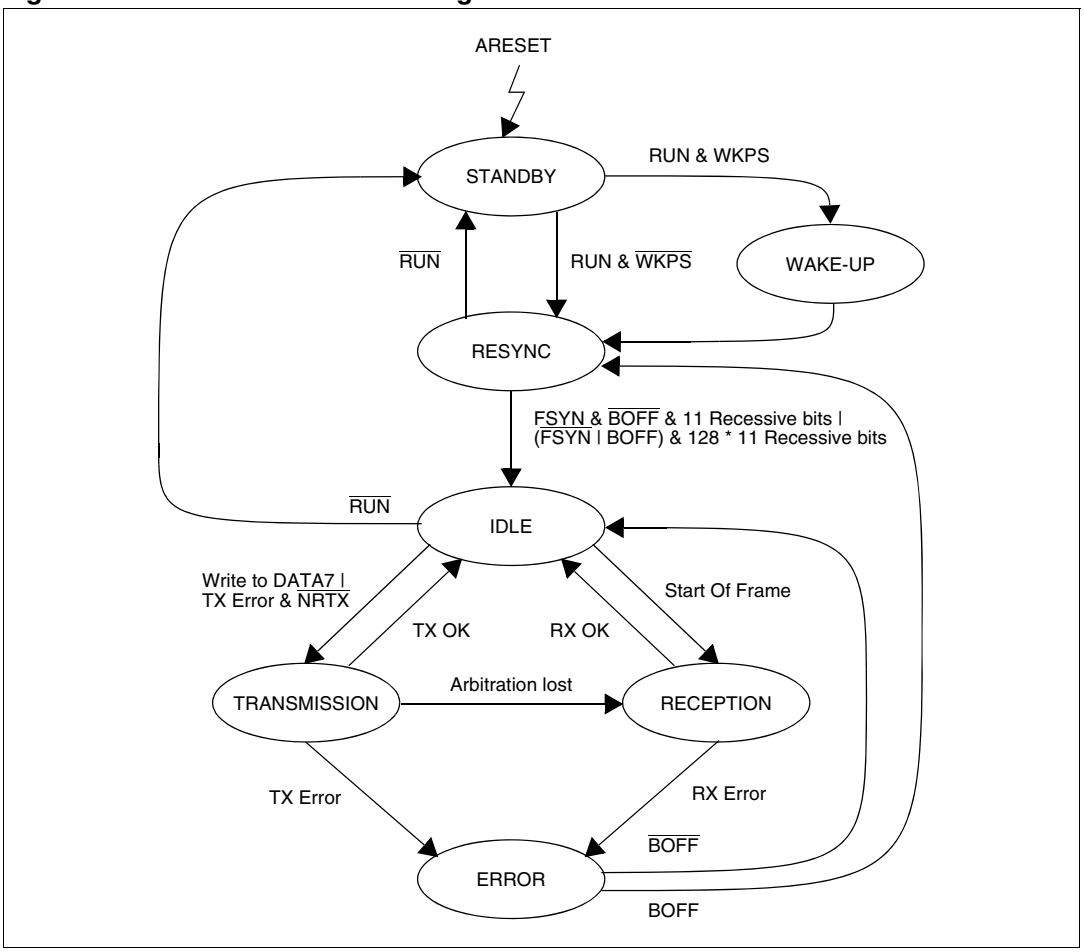
Figure 41. CAN frames



Notes:

- 0 ≤ N ≤ 8
- SOF = Start Of Frame
- ID = Identifier
- RTR = Remote Transmission Request
- IDE = Identifier Extension Bit
- r0 = Reserved Bit
- DLC = Data Length Code
- CRC = Cyclic Redundancy Code
- Error flag: 6 dominant bits if node is error active else 6 recessive bits.
- Suspend transmission: applies to error passive nodes only.
- EOF = End of Frame
- ACK = Acknowledge bit

Figure 42. CAN controller state diagram



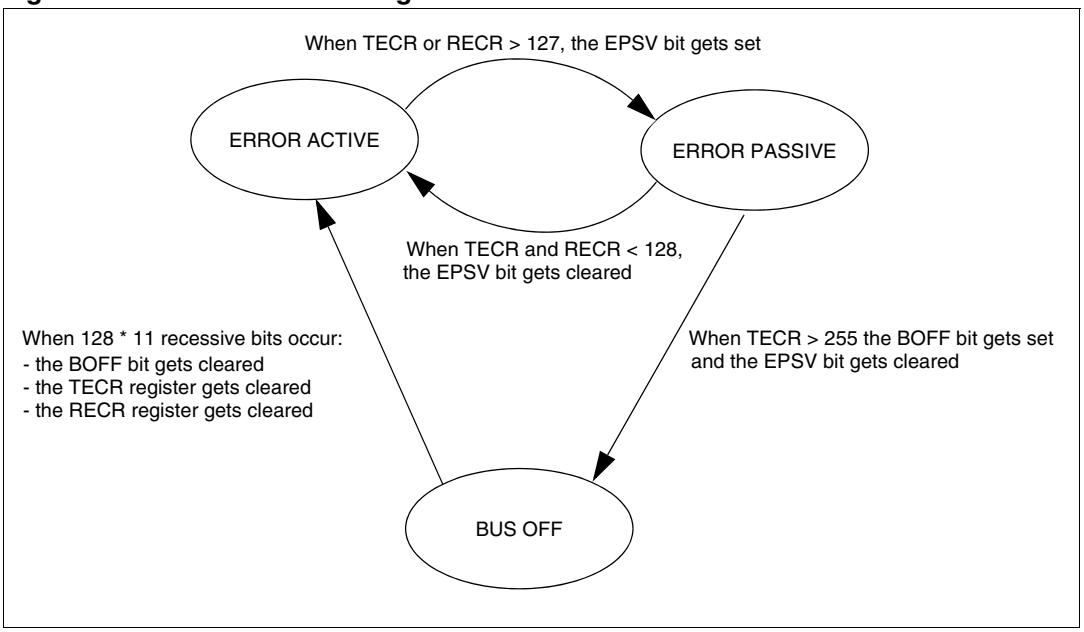
- **RESYNC**. The resynchronization mode is used to find the correct entry point for starting transmission or reception after the node has gone asynchronous either by going into the STANDBY or bus-off states.
  Resynchronization is achieved when 128 sequences of 11 recessive bits have been monitored unless the node is not bus-off and the FSYN bit in the CSR register is set in which case a single sequence of 11 recessive bits needs to be monitored.
- **IDLE**. The CAN controller looks for one of the following events: the RUN bit is reset, a Start Of Frame appears on the CAN bus or the DATA7 register of the currently active page is written to.
- **TRANSMISSION**. Once the LOCK bit of a Buffer Control/Status Register (BCSRx) has been set and read back as such, a transmit job can be submitted by writing to the DATA7 register. The message with the highest priority will be transmitted as soon as the CAN bus becomes idle. Among those messages with a pending transmission request, the highest priority is given to Buffer 3 then 2 and 1. If the transmission fails due to a lost arbitration or to an error while the NRTX bit of the CSR register is reset, then a new transmission attempt is performed . This goes on until the transmission ends successfully or until the job is cancelled by unlocking the buffer, by setting the NRTX bit or if the node ever enters bus-off or if a higher priority message becomes pending. The RDY bit in the BCSRx register, which was set since the job was submitted, gets reset. When a transmission is in progress, the BUSY bit in the BCSRx register is set. If it ends successfully then the TXIF bit in the Interrupt Status Register

(ISR) is set, else the TEIF bit is set. An interrupt is generated in either case provided the TXIE and TEIE bits of the ICR register are set. The ETX bit in the same register is used to get an early transmit interrupt and to automatically unlock the transmitting buffer upon successful completion of its job. This enables the CPU to get a new transmit job pending by the end of the current transmission while always leaving two buffers available for reception. An uninterrupted stream of messages may be transmitted in this way at no overrun risk.
Note 1: Setting the SRTE bit of the CSR register allows transmitted messages to be simultaneously received when they pass the acceptance filtering. This is particularly useful for checking the integrity of the communication path.
Note 2: When the ETX bit is reset, the buffer with the highest priority and with a pending transmission request is always transmitted. When the ETX bit is set, once a buffer participates in the arbitration phase, it is sent until it wins the arbitration even if another transmission is requested from a buffer with a higher priority.

- **RECEPTION**. Once the CAN controller has synchronized itself onto the bus activity, it is ready for reception of new messages. Every incoming message gets its identifier compared to the acceptance filters. If the bitwise comparison of the selected bits ends up with a match for at least one of the filters then that message is elected for reception and a target buffer is searched for. This buffer will be the first one - order is 1 to 3 - that has the LOCK and RDY bits of its BCSRx register reset.
  - When no such buffer exists then an overrun interrupt is generated if the ORIE bit of the ICR register has been set. In this case the identifier of the last message is made available in the Last Identifier Register (LIDHR and LIDLR) at least until it gets overwritten by a new identifier picked-up from the bus.
  - When a buffer does exist, the accepted message gets written into it, the ACC bit in the BCSRx register gets the number of the matching filter, the RDY and RXIF bits get set and an interrupt is generated if the RXIE bit in the ISR register is set.

  Up to three messages can be automatically received without intervention from the CPU because each buffer has its own set of status bits, greatly reducing the reactiveness requirements in the processing of the receive interrupts.
- **ERROR**. The error management as described in the CAN protocol is completely handled by hardware using 2 error counters which get incremented or decremented according to the error condition. Both of them may be read by the application to determine the stability of the network. Moreover, as one of the node status bits (EPSV or BOFF of the CSR register) changes, an interrupt is generated if the SCIE bit is set in the ICR Register. Refer to *Figure 43*.

Figure 43. CAN error state diagram



Bit timing logic

The bit timing logic monitors the serial bus-line and performs sampling and adjustment of the sample point by synchronizing on the start-bit edge and resynchronizing on following edges.

Its operation may be explained simply when the nominal bit time is divided into three segments as follows:

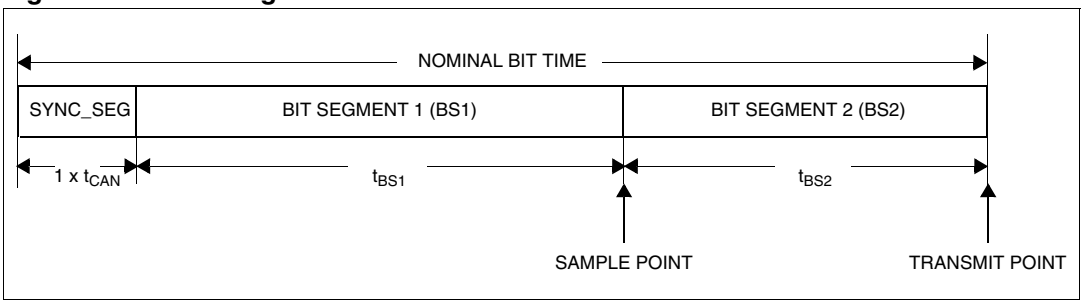
- **Synchronisation segment (SYNC_SEG)**: a bit change is expected to lie within this time segment. It has a fixed length of one time quanta (1 x t_{CAN}).
- **Bit segment 1 (BS1)**: defines the location of the sample point. It includes the PROP_SEG and PHASE_SEG1 of the CAN standard. Its duration is programmable between 1 and 16 time quanta but may be automatically lengthened to compensate for positive phase drifts due to differences in the frequency of the various nodes of the network.
- **Bit segment 2 (BS2)**: defines the location of the transmit point. It represents the PHASE_SEG2 of the CAN standard. Its duration is programmable between 1 and 8 time quanta but may also be automatically shortened to compensate for negative phase drifts.
- **Resynchronization Jump Width (RJW)**: defines an upper bound to the amount of lengthening or shortening of the bit segments. It is programmable between 1 and 4 time quanta.

To guarantee the correct behaviour of the CAN controller, SYNC_SEG + BS1 + BS2 must be greater than or equal to 5 time quanta.

For a detailed description of the CAN resynchronization mechanism and other bit timing configuration constraints, please refer to the CAN Specification - Bosch - Version 2.

As a safeguard against programming errors, the configuration of the Bit Timing Register (BTR) is only possible while the device is in STANDBY mode.

Figure 44. Bit timing



5.6.4 Register description

The CAN registers are organized as 6 general purpose registers plus 5 pages of 16 registers spanning the same address space and primarily used for message and filter storage. The page actually selected is defined by the content of the Page Selection Register. Refer to *Figure 45*.

General purpose registers

INTERRUPT STATUS REGISTER (ISR)

Adress: 005Ah - Read/Write

Reset Value: 00h

7							0
RXIF3	RXIF2	RXIF1	TXIF	SCIF	ORIF	TEIF	EPND

Bit 7 = **RXIF3** *Receive Interrupt Flag for Buffer 3*
– Read/Clear
Set by hardware to signal that a new error-free message is available in buffer 3.
Cleared by software to release buffer 3.
Also cleared by resetting bit RDY of BCSR3.

Bit 6 = **RXIF2** *Receive Interrupt Flag for Buffer 2*
– Read/Clear
Set by hardware to signal that a new error-free message is available in buffer 2.
Cleared by software to release buffer 2.
Also cleared by resetting bit RDY of BCSR2.

Bit 5 = **RXIF1** *Receive Interrupt Flag for Buffer 1*
– Read/Clear
Set by hardware to signal that a new error-free message is available in buffer 1.
Cleared by software to release buffer 1.
Also cleared by resetting bit RDY of BCSR1.

Bit 4 = **TXIF** *Transmit Interrupt Flag*
– Read/Clear
Set by hardware to signal that the highest priority message queued for transmission has been successfully transmitted (ETX = 0) or that it has passed successfully the arbitration (ETX = 1).
Cleared by software.

Bit 3 = **SCIF** *Status Change Interrupt Flag*
– Read/Clear
Set by hardware to signal the reception of a dominant bit while in standby or a change from error active to error passive and bus-off while in run. Also signals any receive error when ESCI = 1.
Cleared by software.

Bit 2 = **ORIF** *Overrun Interrupt Flag*
– Read/Clear
Set by hardware to signal that a message could not be stored because no receive buffer was available.
Cleared by software.

Bit 1 = **TEIF** *Transmit Error Interrupt Flag*
– Read/Clear
Set by hardware to signal that an error occurred during the transmission of the highest priority message queued for transmission.
Cleared by software.

Bit 0 = **EPND** *Error Interrupt Pending*
– Read Only
Set by hardware when at least one of the three error interrupt flags SCIF, ORIF or TEIF is set.
Reset by hardware when all error interrupt flags have been cleared.

Caution: Interrupt flags are reset by writing a "0" to the corresponding bit position. The appropriate way consists in writing an immediate mask or the one's complement of the register content initially read by the interrupt handler. Bit manipulation instruction BRES should never be used due to its read-modify-write nature.

INTERRUPT CONTROL REGISTER (ICR)

Adress: 005Bh - Read/Write

Reset Value: 00h

7							0
0	ESCI	RXIE	TXIE	SCIE	ORIE	TEIE	ETX

Bit 6 = **ESCI** *Extended Status Change Interrupt*
– Read/Set/Clear
Set by software to specify that SCIF is to be set on receive errors also.
Cleared by software to set SCIF only on status changes and wake-up but not on all receive errors.

Bit 5 = **RXIE** *Receive Interrupt Enable*
– Read/Set/Clear
Set by software to enable an interrupt request whenever a message has been received free of errors.
Cleared by software to disable receive interrupt requests.

Bit 4 = TXIE *Transmit Interrupt Enable*
– Read/Set/Clear
Set by software to enable an interrupt request whenever a message has been successfully transmitted.
Cleared by software to disable transmit interrupt requests.

Bit 3 = **SCIE** *Status Change Interrupt Enable*
– Read/Set/Clear
Set by software to enable an interrupt request whenever the node's status changes in run mode or whenever a dominant pulse is received in standby mode.
Cleared by software to disable status change interrupt requests.

Bit 2 = **ORIE** *Overrun Interrupt Enable*
– Read/Set/Clear
Set by software to enable an interrupt request whenever a message should be stored and no receive buffer is avalaible.
Cleared by software to disable overrun interrupt requests.

Bit 1 = **TEIE** *Transmit Error Interrupt Enable*
– Read/Set/Clear
Set by software to enable an interrupt whenever an error has been detected during transmission of a message.
Cleared by software to disable transmit error interrupts.

Bit 0 = **ETX** *Early Transmit Interrupt*
– Read/Set/Clear
Set by software to request the transmit interrupt to occur as soon as the arbitration phase has been passed successfully.
Cleared by software to request the transmit interrupt to occur at the completion of the transfer.

CONTROL/STATUS REGISTER (CSR)

Adress: 005Ch - Read/Write

Reset Value: 00h

7							0
0	BOFF	EPSV	SRTE	NRTX	FSYN	WKPS	RUN

Bit 6 = **BOFF** *Bus-Off State*
– Read Only
Set by hardware to indicate that the node is in bus-off state, i.e. the Transmit Error Counter exceeds 255.
Reset by hardware to indicate that the node is involved in bus activities.

Bit 5 = **EPSV** *Error Passive State*
– Read Only
Set by hardware to indicate that the node is error passive.
Reset by hardware to indicate that the node is either error active (BOFF = 0) or bus-off.

Bit 4 = **SRTE** *Simultaneous Receive/Transmit Enable* – Read/Set/Clear
Set by software to enable simultaneous transmission and reception of a message passing the acceptance filtering. Allows to check the integrity of the communication path.
Reset by software to discard all messages transmitted by the node. Allows remote and data frames to share the same identifier.

Bit 3 = **NRTX** *No Retransmission*
– Read/Set/Clear
Set by software to disable the retransmission of unsuccessful messages.
Cleared by software to enable retransmission of messages until success is met.

Bit 2 = **FSYN** *Fast Synchronization*
– Read/Set/Clear
Set by software to enable a fast resynchronization when leaving standby mode, i.e. wait for only 11 recessive bits in a row.
Cleared by software to enable the standard resynchronization when leaving standby mode, i.e. wait for 128 sequences of 11 recessive bits.

Bit 1 = **WKPS** *Wake-up Pulse*
– Read/Set/Clear
Set by software to generate a dominant pulse when leaving standby mode.
Cleared by software for no dominant wake-up pulse.

Bit 0 = **RUN** *CAN Enable*
– Read/Set/Clear

Set by software to leave standby mode after 128 sequences of 11 recessive bits or just 11 recessive bits if FSYN is set.
Cleared by software to request a switch to the standby or low-power mode as soon as any on-going transfer is complete. Read-back as 1 in the meantime to enable proper signalling of the standby state. The CPU clock may therefore be safely switched OFF whenever RUN is read as 0.

BAUD RATE PRESCALER REGISTER (BRPR)

Adress: 005Dh - Read/Write in Standby mode

Reset Value: 00h

7							0
RJW1	RJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0

RJW[1:0] determine the maximum number of time quanta by which a bit period may be shortened or lengthened to achieve resynchronization.
$t_{RJW} = t_{CAN} * (RJW + 1)$

BRP[5:0] determine the CAN system clock cycle time or time quanta which is used to build up the individual bit timing.
$t_{CAN} = t_{CPU} * (BRP + 1)$
Where t_{CPU} = time period of the CPU clock.
The resulting baud rate can be computed by the formula:

$$BR = \frac{1}{t_{CPU} \times (BRP + 1) \times (BS1 + BS2 + 3)}$$

Note: Writing to this register is allowed only in Standby mode to prevent any accidental CAN protocol violation through programming errors.

BIT TIMING REGISTER (BTR)

Adress: 005Eh - Read/Write in Standby mode

Reset Value: 23h

7							0
0	BS22	BS21	BS20	BS13	BS12	BS11	BS10

BS2[2:0] determine the length of Bit Segment 2.
$t_{BS2} = t_{CAN} * (BS2 + 1)$

BS1[3:0] determine the length of Bit Segment 1.
$t_{BS1} = t_{CAN} * (BS1 + 1)$

Note: Writing to this register is allowed only in Standby mode to prevent any accidental CAN protocol violation through programming errors.

PAGE SELECTION REGISTER (PSR)

Adress: 005Fh - Read/Write

Reset Value: 00h

7							0
0	0	0	0	0	PAGE2	PAGE1	PAGE0

PAGE[2:0] determine which buffer or filter page is mapped at addresses 0010h to 001Fh.

PAGE2	PAGE1	PAGE0	Page Title
0	0	0	Diagnosis
0	0	1	Buffer 1
0	1	0	Buffer 2
0	1	1	Buffer 3
1	0	0	Filters
1	0	1	Reserved
1	1	0	Reserved
1	1	1	Reserved

Page 0 Registers

LAST IDENTIFIER HIGH REGISTER (LIDHR)

Read/Write

Reset Value: Undefined

7							0
LID10	LID9	LID8	LID7	LID6	LID5	LID4	LID3

LID[10:3] are the most significant 8 bits of the last Identifier read on the CAN bus.

LAST IDENTIFIER LOW REGISTER (LIDLR)

Read/Write

Reset Value: Undefined

7							0
LID2	LID1	LID0	LRTR	LDLC3	LDLC2	LDLC1	LDLC0

LID[2:0] are the least significant 3 bits of the last Identifier read on the CAN bus.

LRTR is the last Remote Transmission Request bit read on the CAN bus.

LDLC[3:0] is the last Data Length Code read on the CAN bus.

TRANSMIT ERROR COUNTER REG. (TECR)

Read Only

Reset Value: 00h

7							0
TEC7	TEC6	TEC5	TEC4	TEC3	TEC2	TEC1	TEC0

TEC[7:0] is the least significant byte of the 9-bit Transmit Error Counter implementing part of the fault confinement mechanism of the CAN protocol. In case of an error during transmission, this counter is incremented by 8. It is decremented by 1 after every successful transmission. When the counter value exceeds 127, the CAN controller enters the error passive state. When a value of 256 is reached, the CAN controller is disconnected from the bus.

RECEIVE ERROR COUNTER REG. (RECR)

Page: 00h — Read Only

Reset Value: 00h

7							0
REC7	REC6	REC5	REC4	REC3	REC2	REC1	REC0

REC[7:0] is the Receive Error Counter implementing part of the fault confinement mechanism of the CAN protocol. In case of an error during reception, this counter is incremented by 1 or by 8 depending on the error condition as defined by the CAN standard. After every successful reception the counter is decremented by 1 or reset to 120 if its value was higher than 128. When the counter value exceeds 127, the CAN controller enters the error passive state.

Pages 1,2,3 Registers

IDENTIFIER HIGH REGISTERS (IDHRx)

Read/Write

Reset Value: Undefined

7							0
ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3

ID[10:3] are the most significant 8 bits of the 11-bit message identifier.The identifier acts as the message's name, used for bus access arbitration and acceptance filtering.

IDENTIFIER LOW REGISTERS (IDLRx)

Read/Write

Reset Value: Undefined

7							0
ID2	ID1	ID0	RTR	DLC3	DLC2	DLC1	DLC0

ID[2:0] are the least significant 3 bits of the 11-bit message identifier.

RTR is the Remote Transmission Request bit. It is set to indicate a remote frame and reset to indicate a data frame.

DLC[3:0] is the 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)

Read/Write

Reset Value: Undefined

7							0
DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

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.

BUFFER CONTROL/STATUS REGs. (BCSRx)

Read/Write

Reset Value: 00h

7							0
0	0	0	0	ACC	RDY	BUSY	LOCK

Bit 3 = **ACC** *Acceptance Code*
– Read Only
Set by hardware with the id of the highest priority filter which accepted the message stored in the buffer.
ACC = 0: Match for Filter/Mask0. Possible match for Filter/Mask1.
ACC = 1: No match for Filter/Mask0 and match for Filter/Mask1.

Reset by hardware when either RDY or RXIF gets reset.

Bit 2 = **RDY** *Message Ready*
– Read/Clear
Set by hardware to signal that a new error-free message is available (LOCK = 0) or that a transmission request is pending (LOCK = 1).
Cleared by software when LOCK = 0 to release the buffer and to clear the corresponding

RXIF bit in the Interrupt Status Register.
Cleared by hardware when LOCK = 1 to indicate that the transmission request has been serviced or cancelled.

Bit 1 = **BUSY** *Busy Buffer*
– Read Only
Set by hardware when the buffer is being filled (LOCK = 0) or emptied (LOCK = 1).
Reset by hardware when the buffer is not accessed by the CAN core for transmission nor reception purposes.

Bit 0 = **LOCK** *Lock Buffer*
– Read/Set/Clear
Set by software to lock a buffer. No more message can be received into the buffer thus preserving its content and making it available for transmission.
Cleared by software to make the buffer available for reception. Cancels any pending transmission request.
Cleared by hardware once a message has been successfully transmitted provided the early transmit interrupt mode is on. Left untouched otherwise.

Note that in order to prevent any message corruption or loss of context, LOCK cannot be set nor reset while BUSY is set. Trying to do so will result in LOCK not changing state.

Page 4 Registers

FILTER HIGH REGISTERS (FHRx)

Read/Write

Reset Value: Undefined

7							0
FIL11	FIL10	FIL9	FIL8	FIL7	FIL6	FIL5	FIL4

FIL[11:4] are the most significant 8 bits of a 12-bit message filter. The acceptance filter is compared bit by bit with the identifier and the RTR bit of the incoming message. If there is a match for the set of bits specified by the acceptance mask then the message is stored in a receive buffer.

FILTER LOW REGISTERS (FLRx)

Read/Write

Reset Value: Undefined

7							0
FIL3	FIL2	FIL1	FIL0	0	0	0	0

FIL[3:0] are the least significant 4 bits of a 12-bit message filter.

MASK HIGH REGISTERS (MHRx)

Read/Write

Reset Value: Undefined

7							0
MSK11	MSK10	MSK9	MSK8	MSK7	MSK6	MSK5	MSK4

MSK[11:4] are the most significant 8 bits of a 12-bit message mask. The acceptance mask defines which bits of the acceptance filter should match the identifier and the RTR bit of the incoming message.
MSK_i = 0: don't care.
MSK_i = 1: match required.

MASK LOW REGISTERS (MLRx)

Read/Write

Reset Value: Undefined

7							0
MSK3	MSK2	MSK1	MSK0	0	0	0	0

MSK[3:0] are the least significant 4 bits of a 12-bit message mask.

Figure 45. CAN register map

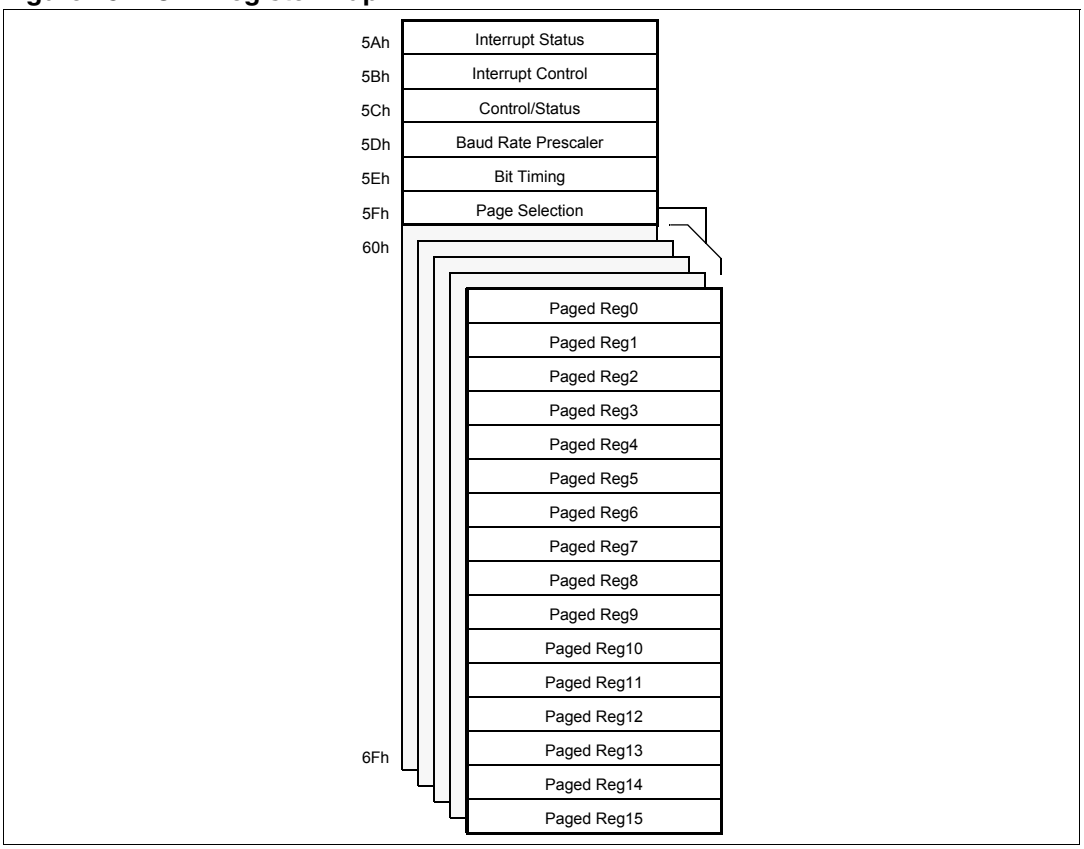


Figure 46. Page maps

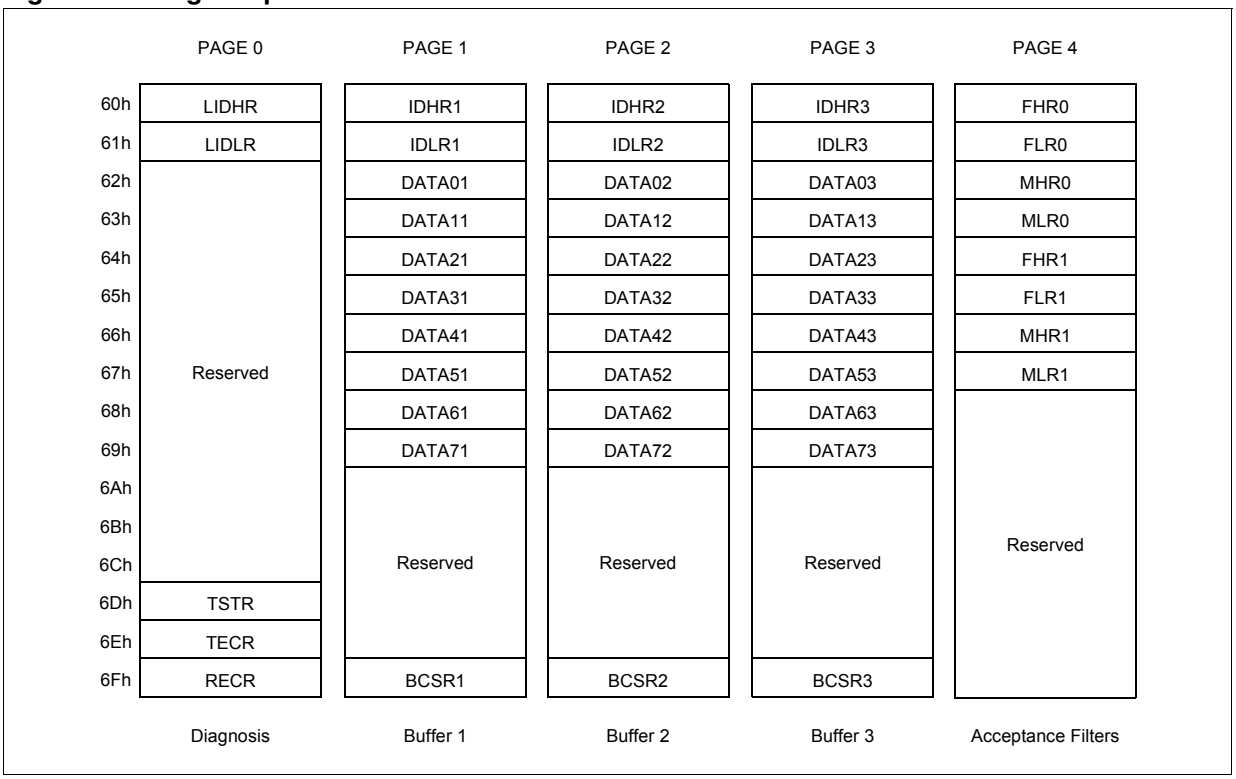


Table 14. CAN register map and reset values

Address (Hex.)	Page	Register Label	7	6	5	4	3	2	1	0
5A		**ISR** Reset Value	RXIF3 0	RXIF2 0	RXIF1 0	TXIF 0	SCIF 0	ORIF 0	TEIF 0	EPND 0
5B		**ICR** Reset Value	0	ESCI 0	RXIE 0	TXIE 0	SCIE 0	ORIE 0	TEIE 0	ETX 0
5C		**CSR** Reset Value	0	BOFF 0	EPSV 0	SRTE 0	NRTX 0	FSYN 0	WKPS 0	RUN 0
5D		**BRPR** Reset Value	RJW1 0	RJW0 0	BRP5 0	BRP4 0	BRP3 0	BRP2 0	BRP1 0	BRP0 0
5E		**BTR** Reset Value	0	BS22 0	BS21 1	BS20 0	BS13 0	BS12 0	BS11 1	BS10 1
5F		**PSR** Reset Value	0	0	0	0	0	PAGE2 0	PAGE1 0	PAGE0 0
60	0	**LIDHR** Reset Value	LID10 x	LID9 x	LID8 x	LID7 x	LID6 x	LID5 x	LID4 x	LID3 x
	1 to 3	**IDHRx** Reset Value	ID10 x	ID9 x	ID8 x	ID7 x	ID6 x	ID5 x	ID4 x	ID3 x
60, 64	4	**FHRx** Reset Value	FIL11 x	FIL10 x	FIL9 x	FIL8 x	FIL7 x	FIL6 x	FIL5 x	FIL4 x

Table 14. CAN register map and reset values (continued)

Address (Hex.)	Page	Register Label	7	6	5	4	3	2	1	0
61	0	**LIDLR** Reset Value	LID2 x	LID1 x	LID0 x	LRTR x	LDLC3 x	LDLC2 x	LDLC1 x	LDLC0 x
	1 to 3	**IDLRx** Reset Value	ID2 x	ID1 x	ID0 x	RTR x	DLC3 x	DLC2 x	DLC1 x	DLC0 x
61, 65	4	**FLRx** Reset Value	FIL3 x	FIL2 x	FIL1 x	FIL0 x	0	0	0	0
62 to 69	1 to 3	**DRx** Reset Value	MSB x	x	x	x	x	x	x	LSB x
62, 66	4	**MHRx** Reset Value	MSK11 x	MSK10 x	MSK9 x	MSK8 x	MSK7 x	MSK6 x	MSK5 x	MSK4 x
63, 67	4	**MLRx** Reset Value	MSK3 x	MSK2 x	MSK1 x	MSK0 x	0	0	0	0
6E	0	**TECR** Reset Value	MSB 0	0	0	0	0	0	0	LSB 0
6F	0	**RECR** Reset Value	MSB 0	0	0	0	0	0	0	LSB 0
	1 to 3	**BCSRx** Reset Value	0	0	0	0	ACC 0	RDY 0	BUSY 0	LOCK 0

5.7 CAN bus transceiver

5.7.1 Introduction

The CAN bus transceiver allows the connection of the microcontroller, with CAN controller unit, to a CAN bus. The transmitter section drives the CAN bus while the receiver section senses the data on the bus.

The CAN transceiver meets ISO/DIS 11898 up to 1 MBaud.

5.7.2 Main features

TRANSMITTER:

- Generation of differential Output signals
- Short Circuit protection from transients in automotive environment
- Slope control to reduce RFI and EMI
- High speed (up to 1Mbaud)
- If un-powered, L9805E CAN node does not disturb the bus lines (the transceiver is in recessive state).

RECEIVER:

- Differential input with high interference suppression
- Common mode input voltage range (V_{COM}) from -5 to 12V

5.7.3 Functional description

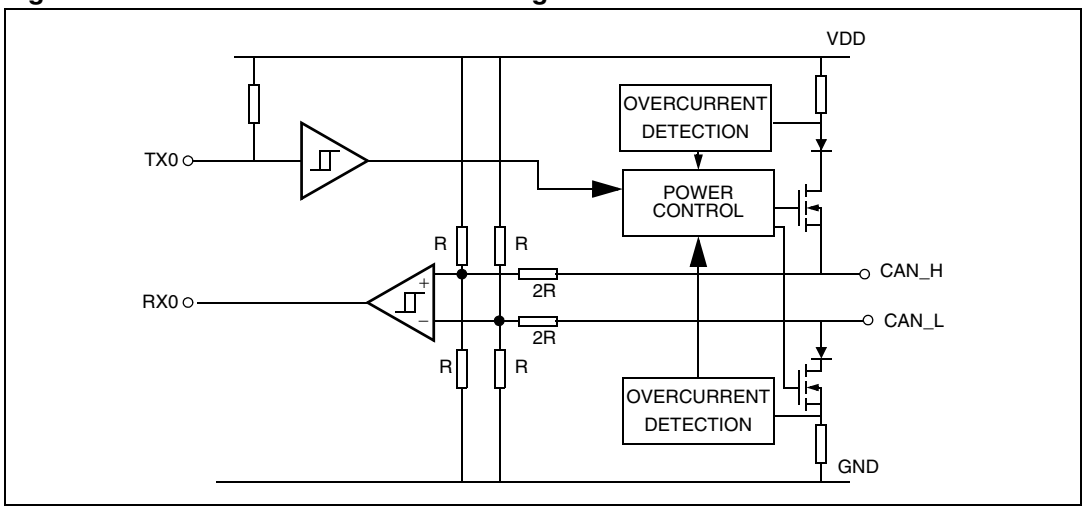
The Can Bus Transceiver is used as an interface between a CAN controller and the physical bus. The device provides transmitting capability to the CAN controller. The transceiver has one logic input pin (TX), one logic output pin (RX) and two Input/Output pins for the electrical connections to the two bus wires (CAN_L and CAN_H). The microcontroller sends data to the TX pin and it receives data from the RX pin. The transmission slew-rates of CAN_H and CAN_L voltage are controlled to reduce RFI and EMI. The transceiver is protected against short circuit or overcurrent: If I_{CANH} and/or I_{CANL} exceeds a current threshold I_{SC}, then the CAN_H and CAN_L power transistors are switched off and the transmission is disabled for T_D=25µs typical.

5.7.4 CAN transceiver disabling function

The transceiver can be disabled and forced to move in a low power consumption mode, setting CANDS bit in DCSR register. When the transceiver is in this mode it can not receive nor transmit any information to the bus. The only way to have again on board the CAN capabilities is reset CANDS bit. The CAN protocol handler can not disable nor enable the transceiver and there is no way to communicate to the controller when the transceiver is down. The disabling function has the only purpose to allow the reduction of the current consumption of the device in application not using the CAN at all or using it for particular functions (such like debugging). Current consumption reduction, when disabling the transceiver, can be as high as 15mA.

Note When the CAN capabilities of L9805E are not needed additional consumption reduction can be achieved putting the CAN controller in Standby Mode (see *Section 5.6.3*).

Figure 47. Can bus transceiver block diagram



5.8 Power bridge

5.8.1 Introduction

The power part of the device consists of two identical independent DMOS half bridges. It is suited to drive resistive and inductive loads.

5.8.2 Main features

The nominal current is 2A.

The low-side switch is a n-channel DMOS transistor while the high-side switch is a p-channel DMOS transistor. Therefore no charge pump is needed.

An anti-crossconduction circuit is included: the low side DMOS is switched on only when the high side is switched off and vice versa. This function avoids that the two DMOS are switched on together firing the high current path from battery to ground. The functions is obtained by sensing the gate voltage and therefore the delay between command and effective switch on of the DMOS doesn't have a fixed length.

The MCU controls all operations of the power stage through the BCSR dedicated register. Short circuit and overtemperature conditions are reported to the CPU using dedicated error flags.

Overtemperature and short circuit conditions switch off the bridge immediately without CPU intervention. The function of the flags is independent of the operation mode of the bridge (sink, source, Z).

In addition both the PWM modules can be directly connected to the power bridge. The power bridge offers then many driving mode alternatives:

Direct Mode: the two half bridges are directly driven by IN1 and IN2 control bit in BCSR.

PWM1 Up/Down Brake Mode: the output of PWM1 drives one side of the bridge while the other side is maintained in a fixed status.

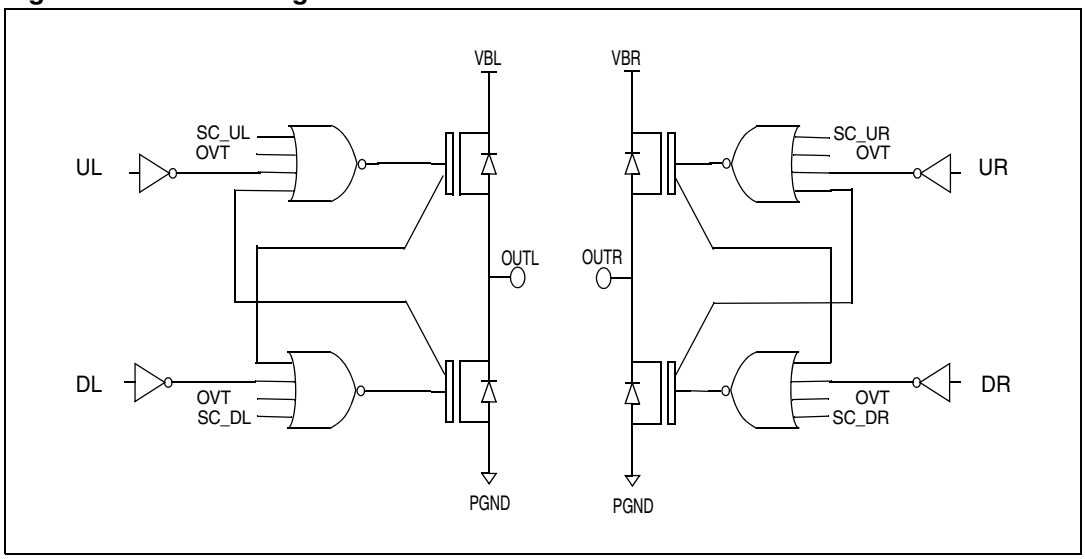
PWM1 Symmetrical Driving Mode: PWM1 line drives directly and symmetrically both side of the bridge.

PWM1/PWM2 Mode: PWM1 drives one side while PWM2 drives the other (two independent half bridges).

5.8.3 Functional description

A schematic description of the Power Bridge circuit is depicted in *Figure 48*. In this schematic the transistors must be considered in ON condition when the gate is high (set).

Figure 48. Power bridge schematic



EN bit in PBCSR is the main enable signal, active high. If EN = 0, all the bridge transistors are switched off (UL, UR, DL and DR are reset) and the outputs OUTL and OUTR are in high impedance state.

Being '0' the status of EN after reset, the bridge is in safe condition (OUTL=OUTR=Z). Therefore the safe condition is guaranteed in undervoltage condition (LVD reset) and in case of main clock (Safeguard reset) or software (Watchdog reset) failures.

Each power DMOS has its own over current detector circuit generating SC_xx signals (see *Figure 48*). SC_xx signals are ORed together to generate SC flag in PBCSR register.

SC flag is then set by hardware as soon as one of the two outputs (or both) are short to battery, ground or if the two outputs are short together (load short). This read only bit is reset only by clearing the EN bit. The rising edge of SC causes an interrupt request if the PIE bit is set in PBCSR register.

When the current monitored in any of the four DMOS of the bridge exceeds limit threshold I_{SC}, the SC bit is set and the corresponding DMOS is switched off after t_{SCPI} time. This function is dominant over any write from data bus by software (i. e. as long as SC is set, the bridge cannot be switched on).

To switch as the bridge again the EN bit must be cleared by software. This resets the SC bit. Setting again EN, the bridge is switched on. If the overcurrent condition is still present, SC is set again (and an interrupt is generated when enabled).

An internal thermal protection circuit monitors continuously the temperature of the device and drives the OVT bit in PBCSR register and, in turn, the OVT signal in *Figure 48*.

The OVT flag is set as soon as the temperature of the chip exceeds Thw and all the transistors of the bridge are switched off. This rising edge causes an interrupt request if the PIE bit is set. This read only bit is reset only by clearing the EN bit. This function is dominant over any write from data bus by software (i. e. as long as OVT is set the bridge cannot be switched on).

To switch the bridge on again the EN bit must be cleared by software. This resets the OVT bit. Setting again EN, the bridge is switched on. If the overtemperature condition is still present, OVT is set again (and an interrupt is generated when enabled).

5.8.4 Interrupt generation

Interrupt generation is controlled by PDIE bit in PBCSR register. When this bit is set Overtemperature and Short-circuit conditions generate an interrupt as described in *Section 5.8.3*.

Setting PDIE when SC and/or OVT flag are set, immediately generates an interrupt request.

The interrupt request of the power bridge is cleared when the EN bit is cleared by software.

5.8.5 Operating Modes

The status of the OUTL and OUTR power outputs is controlled by IN1, IN2, EN, PWM_EN and DIR bit in PBCSR register, plus the PWM1 and PWM2 line, according to the Functional Description Table (*Table 15*).

Note: The functional description table (Table 15) uses symbols UL,R (Up Left or Right) and DR,L (Down Left or Right) to indicate the driving signal of the four DMOS. Conventionally a transistor is in the on status when its driving signal is set ('1') while it is in off status when the driving signal is reset ('0').

Table 15. Functional description

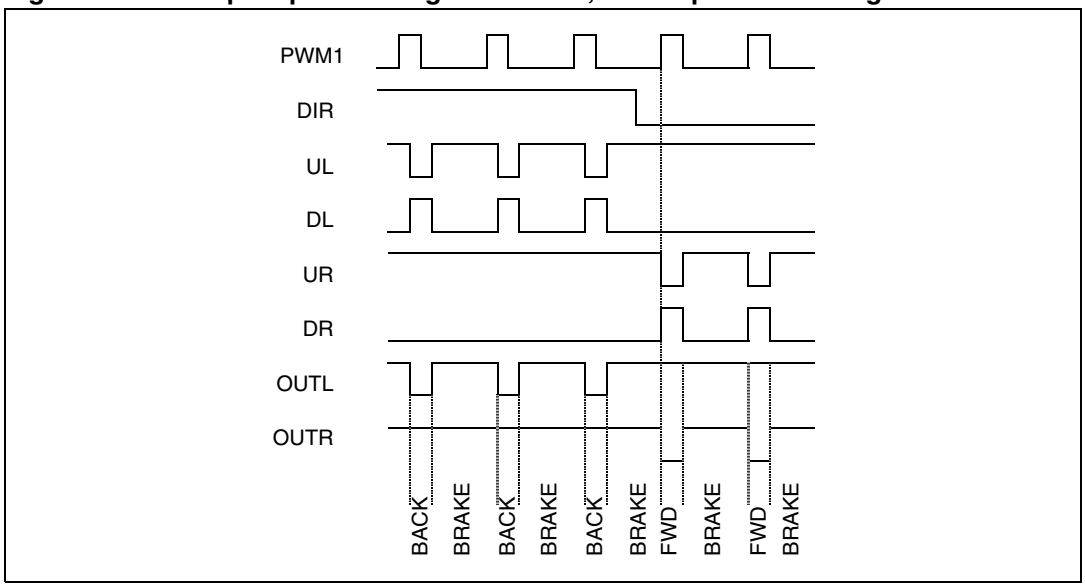
Drive	EN	PWM_EN	DIR	PWM1	IN1	IN2	UL	DL	UR	DR		Operation	Configuration
	0	X	X	X	X	X	0	0	0	0		INHIBIT	
Direct Mode	1	0	X	X	0	0	0	1	0	1		BRAKE	Full or Two Half Bridges
	1	0	X	X	0	1	0	1	1	0		BACK	Full or Two Half Bridges
	1	0	X	X	1	0	1	0	0	1		FORWARD	Full or Two Half Bridges
	1	0	X	X	1	1	1	0	1	0		BRAKE	Full or Two Half Bridges
PWM1 Up Brake Mode	1	1	0	0	0	0	1	0	1	0		BRAKE	Full Bridge
	1	1	0	1	0	0	1	0	0	1		FORWARD	Full Bridge
	1	1	1	0	0	0	1	0	1	0		BRAKE	Full Bridge
	1	1	1	1	0	0	0	1	1	0		BACK	Full Bridge

Table 15. Functional description (continued)

Drive	EN	PWM_EN	DIR	PWM1	IN1	IN2	UL	DL	UR	DR		Operation	Configuration
PWM1 Down Brake Mode	1	1	0	0	0	1	0	1	0	1		BRAKE	Full Bridge
	1	1	0	1	0	1	1	0	0	1		FORWARD	Full Bridge
	1	1	1	0	0	1	0	1	0	1		BRAKE	Full Bridge
	1	1	1	1	0	1	0	1	1	0		BACK	Full Bridge
PWM1 Symmetrical Driving Mode	1	1	0	0	1	0	0	1	1	0		BACK	Full Bridge
	1	1	0	1	1	0	1	0	0	1		FORWARD	Full Bridge
	1	1	1	0	1	0	1	0	0	1		FORWARD	Full Bridge
	1	1	1	1	1	0	0	1	1	0		BACK	Full Bridge
PWM1/PWM2 Mode	1	1	0		1	1	pwm1	$\overline{\text{pwm1}}$	pwm2	$\overline{\text{pwm2}}$		PWM1 ->left PWM2->right	Two Half Bridges
	1	1	1		1	1	$\overline{\text{pwm1}}$	pwm1	$\overline{\text{pwm2}}$	pwm2		$\overline{\text{PWM1}}$ ->left $\overline{\text{PWM2}}$->right	Two Half Bridges

Note: The DIR signal is internally synchronized with the PWM1 and PWM2 signals according to the selected Driving Mode. After writing the DIR bit in PBCSR register, the direction changes in correspondence with the first rising edge of PWM1. The same procedure is used in the case of PWM2. This allows the proper control of the direction changes. When the PWM signal is 0% or 100%, being no edges available, the DIR bit can't be latched and the direction does not change until a PWM edge occurs.

Figure 49. Example - power bridge waveform, PWM up brake driving mode



5.8.6 Register description

The power section is controlled by the microcontroller through the following register:

POWER BRIDGE CONTROL STATUS REGISTER (PBCSR)

Address: 0021h - Read/Write

Reset Value: 00000000 (00h)

7							0
PIE	OVT	SC	DIR	IN2	IN1	PWM_EN	EN

Bit 0 **= EN:** Power Bridge enable. On reset the bridge is disabled and OUTL and OUTR are in high impedance condition.

Bit 1**= PWM_EN**: PWM driving enable. When reset the bridge is driven directly by IN1 and IN2 bit (Direct Mode). When set the driving is made by PWM1 and/or PWM2 bit according to the Operation Mode selected by IN1 and IN2 bit.

Bit 2**= IN1**: Left Half Bridge control bit if PWM_EN=0, driving mode selection bit if PWM_EN=1.

Bit 3**= IN2**: Right Half Bridge control bit if PWM_EN=0, driving mode selection bit if PWM_EN=1.

The following table summarizes the driving mode selection made by PWM_EN, IN1 and IN2 bit

PWM_EN	IN1	IN2	Driving Mode
0	X	X	Direct
1	0	0	PWM1 Up Braking
1	0	1	PWM1 Down Braking
1	1	0	PWM1 Symmetrical
1	1	1	PWM1/PWM2

Bit 4**= DIR**: Direction bit. This bit is meaningless when PWM_EN=0. When PWM_EN is set the DIR bit controls the “driving direction” of the bridge. In order to implement a precise control of the direction changes, DIR value is latched by the rising edge of the pwm signal driving the bridge. When the signal does not have edges (i.e. pwm = 0% or 100%) the DIR bit can not be latched and the driving direction does not change even changing DIR bit in PBCSR.

Bit 5**= SC**: Short Circuit flag (read only)

Bit 6**= OVT**: Overtemperature flag (read only)

Bit 7**= PIE**: Power section interrupt enable.

5.9 EEPROM (EEP)

5.9.1 Introduction

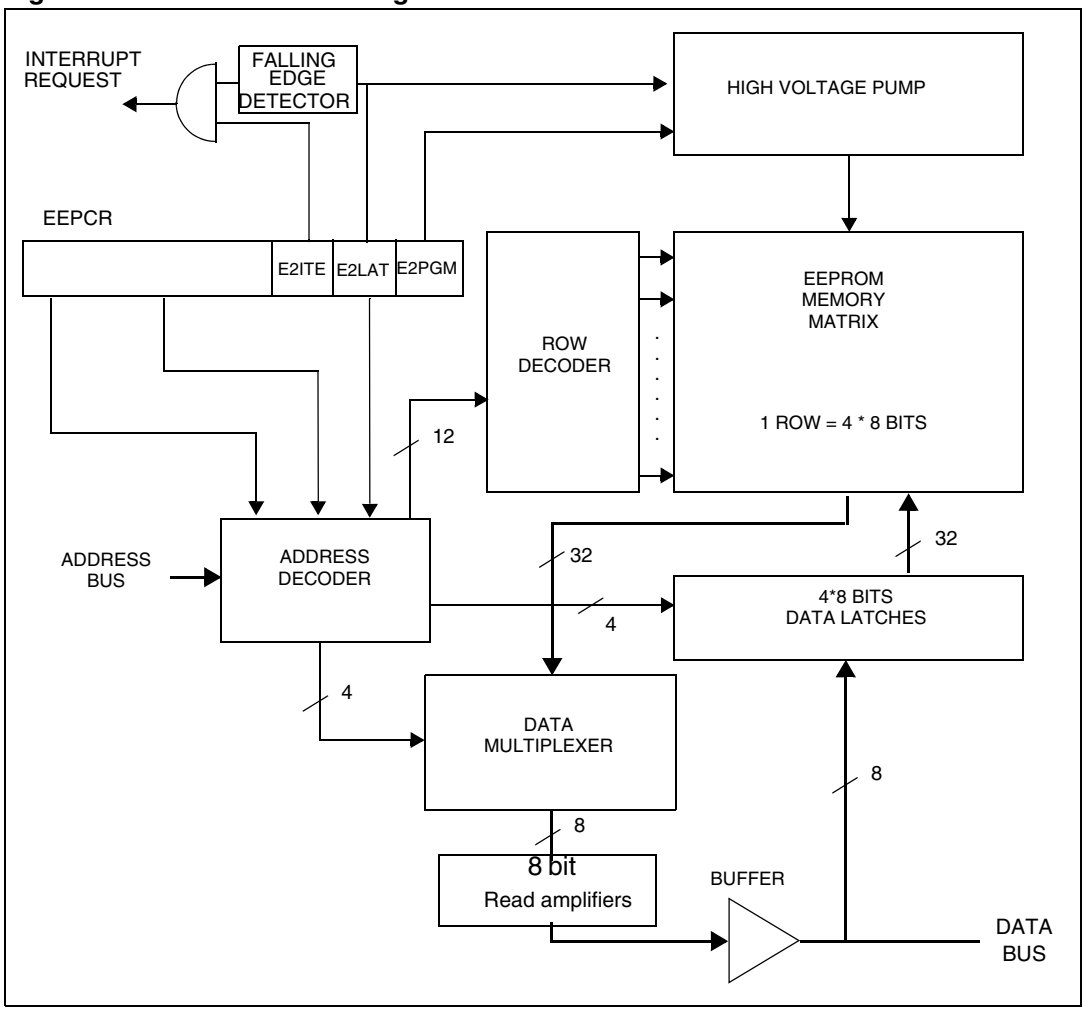
The Electrically Erasable Programmable Read Only Memory is used to store data that need a non volatile back-up. The use of the EEPROM requires a basic protocol described in this chapter.

Software or hardware reset and halt modes are managed immediately, stopping the action in progress. Wait mode does not affect the programming of the EEPROM.

The Read operation of this memory is the same of a Read-Only-Memory or RAM. The erase and programming cycles are controlled by an EEPROM control register.
The user can program 1 to 4 bytes at the same programming cycle providing that the high part of the address is the same for the bytes to be written (only address bits A1 and A0 can change).

The EEPROM is mono-voltage. A charge pump generates the high voltage internally to enable the erase and programming cycles. The erase and programming cycles are chained automatically. The global programming cycle duration is controlled by an internal circuit.

Figure 50. EEPROM block diagram



5.9.2 Functional description

Read operation (E2LAT=0)

The EEPROM can be read as a normal ROM/RAM location when the E2LAT bit of the EEPCR register is cleared. The address decoder selects the desired byte. The 8 sense amplifiers evaluate the stored byte which is put on the data bus.

Write operation (E2LAT=1)

The EEPROM programming flowchart is shown in *Figure 52*.

To access the write mode, E2LAT bit has to be set by software (the E2PGM bit remains cleared). When a write access to the EEPROM area occurs, the value is latched inside the 16 data latches according to its address.

When E2PGM bit is set by into the software, all the previous bytes written in the data latches (up to 16) are programmed in the EEPROM cells. The effective high address (row) is determined by the last EEPROM write sequence. To avoid wrong programming, the user must take care that all the bytes written between two programming sequences have the same high address: only the four Least Significant Bits of the address can change.

At the end of the programming cycle, the E2PGM and E2LAT bits are cleared simultaneously, and an interrupt is generated if the IE bit is set. The Data EEPROM interrupt request is cleared by hardware when the Data EEPROM interrupt vector is fetched.

Wait mode

The EEPROM can enter the wait mode by executing the wait instruction of the micro-controller. The EEPROM will effectively enter this mode if there is no programming in progress, in such a case the EEPROM will finish the cycle and then enter this low consumption mode.

Halt mode

The EEPROM enters the halt mode if the micro-controller did execute the HALT instruction. The EEPROM will stop the function in progress, and will enter in this low consumption mode.

Figure 51. Data EEPROM programming cycle

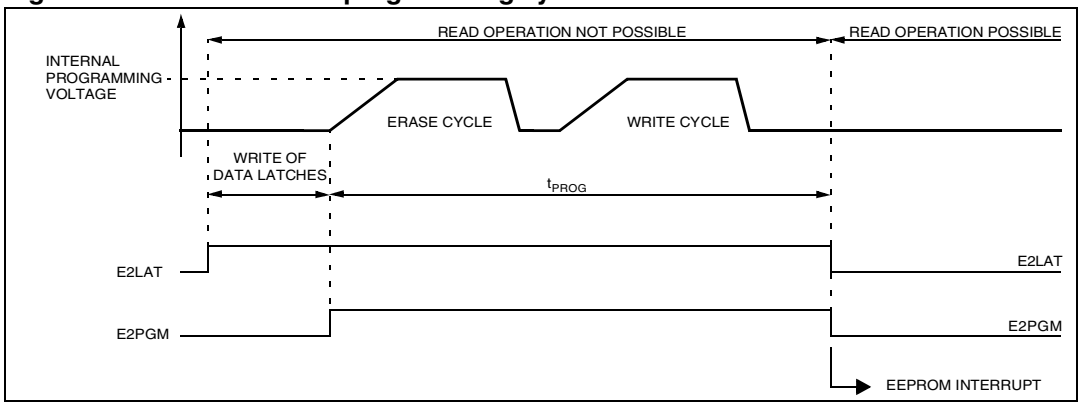
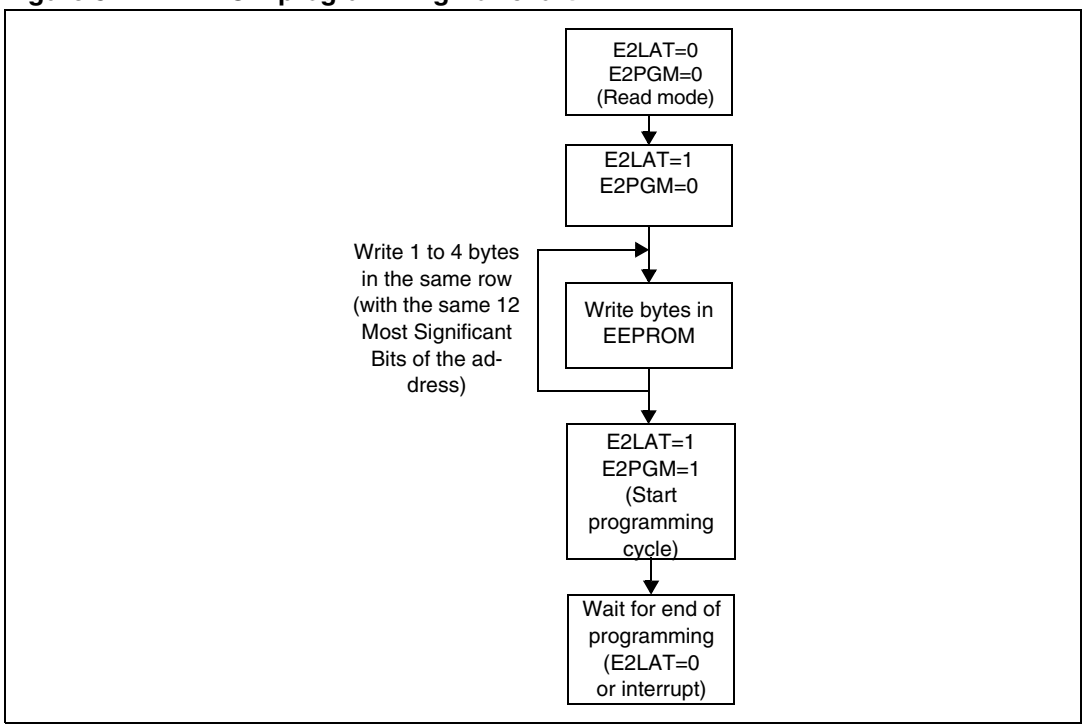


Figure 52. EEPROM programming flowchart



5.9.3 Register description

EEPROM CONTROL REGISTER (EEPCR)

Address: 002Ch - Read/Write
Reset Value: 0000 0000 (00h)

7							0
0	0	0	0	0	E2ITE	E2LAT	E2PGM

Bit 7:3 = Reserved, forced by hardware to 0.

Bit 2 = **E2ITE**: *Interrupt enable.*
This bit is set and cleared by software.

0: Interrupt disabled
1: Interrupt enabled

When the programming cycle is finished (E2PGM toggles from 1 to 0), an interrupt is generated only if E2ITE is high. The interrupt is automatically cleared when the micro-controller enters the EEPROM interrupt routine.

Bit 1 = **E2LAT**: *Read/Write mode.*
This bit is set by software. It is cleared by hardware at the end of the programming cycle. It can be cleared by software only if E2PGM=0.

0: Read mode
1: Write mode

When E2LAT=1, if the E2PGM bit is low and the micro-controller is in write mode, the 8 bit data bus is stored in one of the four groups of 8 bit data latches, selected by the address. This happens every time the device executes an EEPROM Write instruction. If E2PGM remains low, the content of the 8 bit data latches is not transferred into the matrix, because the High Voltage charge-pump does not start. The 8 data latches are selected by the lower part of the address (A<1:0> bits). If 4 consecutive write instructions are executed, by sweeping from A<1:0>=0h to A<1:0>=3h, with the same higher part of the address, all the 4 groups of data latches will be written, and they will be ready to write a whole row of the EEPROM matrix, as soon as E2PGM goes high and the charge-pump starts. If only one write instruction is executed before E2PGM goes high, only one group of data latches will be selected and only one byte of the matrix will be written. At the end of the programming cycle, E2LAT bit is automatically cleared, and the data latches are cleared.

Bit 0 = **E2PGM**: *Programming Control.*
This bit is set by software to begin the programming cycle. At the end of the programming cycle, this bit is cleared by hardware and an interrupt is generated if the E2ITE bit is set.

0: Programming finished or not started

1: Programming cycle is in progress

Note: if the E2PGM bit is cleared during the programming cycle, the memory data is not guaranteed.

Note: Care should be taken during the programming cycle. Writing to the same memory location will over-program the memory (logical AND between the two write access data result) because the data latches are only cleared at the end of the programming cycle and by the falling edge of E2LAT bit.
It is not possible to read the latched data.

Special management of wrong EEPROM access:

If a read happens while E2LAT=1, then the data bus will not be driven.

If a write access happens while E2LAT=0, then the data on the bus will not be latched.

The data latches are cleared when the user sets E2LAT bit.

Note: Care should be taken in the write routine: the software has to read back the data and rewrite in case of failure.

6 Instruction set

6.1 ST7 addressing modes

The ST7 Core features 17 different addressing modes which can be classified in 7 main groups:

Addressing Mode	Example
Inherent	nop
Immediate	ld A,#$55
Direct	ld A,$55
Indexed	ld A,($55,X)
Indirect	ld A,([$55],X)
Relative	jrne loop
Bit operation	bset byte,#5

The ST7 Instruction set is designed to minimize the number of bytes required per instruction: To do so, most of the addressing modes may be subdivided in two sub-modes called long and short:

- Long addressing mode is more powerful because it can use the full 64Kbyte address space, however it uses more bytes and more CPU cycles.
- Short addressing mode is less powerful because it can generally only access page zero (0000 - 00FFh range), but the instruction size is more compact, and faster. All memory to memory instructions use short addressing modes only (CLR, CPL, NEG, BSET, BRES, BTJT, BTJF, INC, DEC, RLC, RRC, SLL, SRL, SRA, SWAP)

The ST7 Assembler optimize the use of long and short addressing modes.

Table 16. ST7 addressing mode overview:

Mode			Syntax	Destination	Pointer Address	Pointer Size (Hex.)	Length (Bytes)
Inherent			nop				+ 0
Immediate			ld A,#$55				+ 1
Short	Direct		ld A,$10	00..FF			+ 1
Long	Direct		ld A,$1000	0000..FFFF			+ 2
No Offset	Direct	Indexed	ld A,(X)	00..FF			+ 0
Short	Direct	Indexed	ld A,($10,X)	00..1FE			+ 1
Long	Direct	Indexed	ld A,($1000,X)	0000..FFFF			+ 2
Short	Indirect		ld A,[$10]	00..FF	00..FF	byte	+ 2
Long	Indirect		ld A,[$10.w]	0000..FFFF	00..FF	word	+ 2
Short	Indirect	Indexed	ld A,([$10],X)	00..1FE	00..FF	byte	+ 2
Long	Indirect	Indexed	ld A,([$10.w],X)	0000..FFFF	00..FF	word	+ 2

Table 16. ST7 addressing mode overview: (continued)

Mode			Syntax	Destination	Pointer Address	Pointer Size (Hex.)	Length (Bytes)
Relative	Direct		jrne loop	PC+/-127			+ 1
Relative	Indirect		jrne [$10]	PC+/-127	00..FF	byte	+ 2
Bit	Direct		bset $10,#7	00..FF			+ 1
Bit	Indirect		bset [$10],#7	00..FF	00..FF	byte	+ 2
Bit	Direct	Relative	btjt $10,#7,skip	00..FF			+ 2
Bit	Indirect	Relative	btjt [$10],#7,skip	00..FF	00..FF	byte	+ 3

Inherent:

All Inherent instructions consist of a single byte. The opcode fully specifies all the required information for the CPU to process the operation.

Inherent Instruction	Function
NOP	No operation
TRAP	S/W Interrupt
WFI	Wait For Interrupt (Low Power Mode)
HALT	Halt Oscillator (Lowest Power Mode)
RET	Sub-routine Return
IRET	Interrupt Sub-routine Return
SIM	Set Interrupt Mask
RIM	Reset Interrupt Mask
SCF	Set Carry Flag
RCF	Reset Carry Flag
RSP	Reset Stack Pointer
LD	Load
CLR	Clear
PUSH/POP	Push/Pop to/from the stack
INC/DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2 Complement
MUL	Byte Multiplication
SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations
SWAP	Swap Nibbles

Immediate:

Immediate instructions have two bytes, the first byte contains the opcode, the second byte contains the operand value.

Immediate Instruction	Function
LD	Load
CP	Compare
BCP	Bit Compare
AND, OR, XOR	Logical Operations
ADC, ADD, SUB, SBC	Arithmetic Operations

Direct (short, long):

In Direct instructions, the operands are referenced by their memory address, which follows the opcode.

Available Long and Short Direct Instructions	Function
LD	Load
CP	Compare
AND, OR, XOR	Logical Operations
ADC, ADD, SUB, SBC	Arithmetic Additions/Substructions operations
BCP	Bit Compare

Short Direct Instructions Only	Function
CLR	Clear
INC, DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2 Complement
BSET, BRES	Bit Operations
BTJT, BTJF	Bit Test and Jump Operations
SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations
SWAP	Swap Nibbles
CALL, JP	Call or Jump subroutine

The direct addressing mode consists of two sub-modes:

Direct (short):

The address is a byte, thus requires only one byte after the opcode, but only allows 00 - FF addressing space.

Direct (long):

The address is a word, thus allowing 64Kb addressing space, but requires 2 bytes after the opcode.

Indexed (no offset, short, long)

In this mode, the operand is referenced by its memory address, which is defined by the unsigned addition of an index register (X or Y) with an offset which follows the opcode.

No Offset, Long and Short Indexed Instruction	Function
LD	Load
CP	Compare
AND, OR, XOR	Logical Operations
ADC, ADD, SUB, SBC	Arithmetic Additions/Substructions operations
BCP	Bit Compare

No Offset and Short Indexed Instructions Only	Function
CLR	Clear
INC, DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2's Complement
BSET, BRES	Bit Operations
BTJT, BTJF	Bit Test and Jump Operations
SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations
SWAP	Swap Nibbles
CALL, JP	Call or Jump subroutine

The indirect addressing mode consists of three sub-modes:

Indexed (no offset)

There is no offset (no extra byte after the opcode), and allows 00 - FF addressing space.

Indexed (short)

The offset is a byte, thus requires only one byte after the opcode and allows 00 - 1FE addressing space.

Indexed (long):

The offset is a word, thus allowing 64Kb addressing space and requires 2 bytes after the opcode.

Indirect (short, long):

The required data byte to do the operation is found by its memory address, located in memory (pointer).

Available Long and Short Indirect Instructions	Function
LD	Load
CP	Compare
AND, OR, XOR	Logical Operations
ADC, ADD, SUB, SBC	Arithmetic Additions/Substructions operations
BCP	Bit Compare

Short Indirect Instructions Only	Function
CLR	Clear
INC, DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2's Complement
BSET, BRES	Bit Operations
BTJT, BTJF	Bit Test and Jump Operations
SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations
SWAP	Swap Nibbles
CALL, JP	Call or Jump subroutine

The pointer address follows the opcode. The indirect addressing mode consists of two sub-modes:

Indirect (short):

The pointer address is a byte, the pointer size is a byte, thus allowing 00 - FF addressing space, and requires 1 byte after the opcode.

Indirect (long):

The pointer address is a byte, the pointer size is a word, thus allowing 64Kb addressing space, and requires 1 byte after the opcode.

Indirect indexed (short, long):

This is a combination of indirect and short indexed addressing modes. The operand is referenced by its memory address, which is defined by the unsigned addition of an index register value (X or Y) with a pointer value located in memory. The pointer address follows the opcode.

Long and Short Indirect Indexed Instructions	Function
LD	Load
CP	Compare
AND, OR, XOR	Logical Operations
ADC, ADD, SUB, SBC	Arithmetic Additions/Substructions operations
BCP	Bit Compare

Short Indirect Indexed Instructions Only	Function
CLR	Clear
INC, DEC	Increment/Decrement
TNZ	Test Negative or Zero
CPL, NEG	1 or 2 Complement
BSET, BRES	Bit Operations
BTJT, BTJF	Bit Test and Jump Operations
SLL, SRL, SRA, RLC, RRC	Shift and Rotate Operations
SWAP	Swap Nibbles
CALL, JP	Call or Jump subroutine

The indirect indexed addressing mode consists of two sub-modes:

Indirect indexed (short):

The pointer address is a byte, the pointer size is a byte, thus allowing 00 - 1FE addressing space, and requires 1 byte after the opcode.

Indirect indexed (long):

The pointer address is a byte, the pointer size is a word, thus allowing 64Kb addressing space, and requires 1 byte after the opcode.

Relative mode (direct, indirect):

This addressing mode is used to modify the PC register value, by adding an 8 bit signed offset to it.

Available Relative Direct/Indirect Instructions	Function
JRxx	Conditional Jump
CALLR	Call Relative

The relative addressing mode consists of two sub-modes:

Relative (direct):

The offset is following the opcode.

Relative (indirect):

The offset is defined in memory, which address follows the opcode.

6.2 Instruction groups

The ST7 family devices use an Instruction Set consisting of 63 instructions. The instructions may be subdivided into 13 main groups as illustrated in the following table:

Load and Transfer	LD	CLR						
Stack operation	PUSH	POP	RSP					
Increment/Decrement	INC	DEC						
Compare and Tests	CP	TNZ	BCP					
Logical operations	AND	OR	XOR	CPL	NEG			
Bit Operation	BSET	BRES						
Conditional Bit Test and Branch	BTJT	BTJF						
Arithmetic operations	ADC	ADD	SUB	SBC	MUL			
Shift and Rotates	SLL	SRL	SRA	RLC	RRC	SWAP	SLA	
Unconditional Jump or Call	JRA	JRT	JRF	JP	CALL	CALLR	NOP	RET
Conditional Branch	JRxx							
Interruption management	TRAP	WFI	HALT	IRET				
Code Condition Flag modification	SIM	RIM	SCF	RCF				

Using a pre-byte

The instructions are described with one to four opcodes.

In order to extend the number of available opcodes for an 8-bit CPU (256 opcodes), three different probate pockets are defined. These prebytes modify the meaning of the instruction they precede.

The whole instruction becomes:

PC-2 End of previous instruction

PC-1 Prebyte

PC opcode

PC+1 Additional word (0 to 2) according to the number of bytes required to compute the effective address

These prebytes enable instruction in Y as well as indirect addressing modes to be implemented. They precede the opcode of the instruction in X or the instruction using direct addressing mode. The prebytes are:

PDY 90 Replace an X based instruction using immediate, direct, indexed, or inherent addressing mode by a Y one.

PIX 92 Replace an instruction using direct, direct bit, or direct relative addressing mode to an instruction using the corresponding indirect addressing mode.
It also changes an instruction using X indexed addressing mode to an instruction using indirect X indexed addressing mode.

PIY 91 Replace an instruction using X indirect indexed addressing mode by a Y one.

Mnemo	Description	Function/Example	Dst	Src	H	I	N	Z	C
ADC	Add with Carry	A = A + M + C	A	M	H		N	Z	C
ADD	Addition	A = A + M	A	M	H		N	Z	C
AND	Logical And	A = A . M	A	M			N	Z	
BCP	Bit compare A, Memory	tst (A . M)	A	M			N	Z	
BRES	Bit Reset	bres Byte, #3	M						
BSET	Bit Set	bset Byte, #3	M						
BTJF	Jump if bit is false (0)	btjf Byte, #3, Jmp1	M						C
BTJT	Jump if bit is true (1)	btjt Byte, #3, Jmp1	M						C
CALL	Call subroutine								
CALLR	Call subroutine relative								
CLR	Clear		reg, M				0	1	
CP	Arithmetic Compare	tst(Reg - M)	reg	M			N	Z	C
CPL	One Complement	A = FFH-A	reg, M				N	Z	1
DEC	Decrement	dec Y	reg, M				N	Z	
HALT	Halt					0			
IRET	Interrupt routine return	Pop CC, A, X, PC			H	I	N	Z	C
INC	Increment	inc X	reg, M				N	Z	
JP	Absolute Jump	jp [TBL.w]							
JRA	Jump relative always								
JRT	Jump relative								
JRF	Never jump	jrf *							
JRIH	Jump if Port A INT pin = 1	(no Port A Interrupts)							
JRIL	Jump if Port A INT pin = 0	(Port A interrupt)							
JRH	Jump if H = 1	H = 1 ?							
JRNH	Jump if H = 0	H = 0 ?							
JRM	Jump if I = 1	I = 1 ?							
JRNM	Jump if I = 0	I = 0 ?							
JRMI	Jump if N = 1 (minus)	N = 1 ?							
JRPL	Jump if N = 0 (plus)	N = 0 ?							

Mnemo	Description	Function/Example	Dst	Src	H	I	N	Z	C
JREQ	Jump if Z = 1 (equal)	Z = 1 ?							
JRNE	Jump if Z = 0 (not equal)	Z = 0 ?							
JRC	Jump if C = 1	C = 1 ?							
JRNC	Jump if C = 0	C = 0 ?							
JRULT	Jump if C = 1	Unsigned <							
JRUGE	Jump if C = 0	Jmp if unsigned >=							
JRUGT	Jump if (C + Z = 0)	Unsigned >							
JRULE	Jump if (C + Z = 1)	Unsigned <=							
LD	Load	dst <= src	reg, M	M, reg			N	Z	
MUL	Multiply	X,A = X * A	A, X, Y	X, Y, A	0				0
NEG	Negate (2's compl)	neg $10	reg, M				N	Z	C
NOP	No Operation								
OR	OR operation	A = A + M	A	M			N	Z	
POP	Pop from the Stack	pop reg pop CC	reg CC	M M	H	I	N	Z	C
PUSH	Push onto the Stack	push Y	M	reg, CC					
RCF	Reset carry flag	C = 0							0
RET	Subroutine Return								
RIM	Enable Interrupts	I = 0				0			
RLC	Rotate left true C	C <= A <= C	reg, M				N	Z	C
RRC	Rotate right true C	C => A => C	reg, M				N	Z	C
RSP	Reset Stack Pointer	S = Max allowed							
SBC	Substract with Carry	A = A - M - C	A	M			N	Z	C
SCF	Set carry flag	C = 1							1
SIM	Disable Interrupts	I = 1				1			
SLA	Shift left Arithmetic	C <= A <= 0	reg, M				N	Z	C
SLL	Shift left Logic	C <= A <= 0	reg, M				N	Z	C
SRL	Shift right Logic	0 => A => C	reg, M				0	Z	C
SRA	Shift right Arithmetic	A7 => A => C	reg, M				N	Z	C
SUB	Substraction	A = A - M	A	M			N	Z	C
SWAP	SWAP nibbles	A7-A4 <=> A3-A0	reg, M				N	Z	
TNZ	Test for Neg & Zero	tnz lbl1					N	Z	
TRAP	S/W trap	S/W interrupt				1			
WFI	Wait for Interrupt					0			
XOR	Exclusive OR	A = A XOR M	A	M			N	Z	

7 Electrical characteristics

7.1 Absolute maximum ratings

This device contains circuitry to protect the inputs against damage due to high static voltage or electric fields. Nevertheless, it is recommended that normal precautions be observed in order to avoid subjecting this high-impedance circuit to voltage above those quoted in the Absolute Maximum Ratings. For proper operation, it is recommended that the input voltage V_{IN}, on the digital pins, be constrained within the range:

$$(GND - 0.3V) \leq V_{IN} \leq (V_{DD} + 0.3V)$$

To enhance reliability of operation, it is recommended to configure unused I/Os as inputs and to connect them to an appropriate logic voltage level such as GND or V_{DD}.

All the voltage in the following tables are referenced to GND.

Table 17. Absolute maximum ratings (voltage referenced to GND)

Symbol	Ratings	Value	Unit
VBR VBL = VB VB1	Operating supply voltage t = 10s ISO transients t = 400ms(1)	0 to 18 0 to 24 0 to 40	V
VBR=VBL	Dynamic destruction proof t<500µs (single pulse)	0 to -2	V
|AGND - GND|	Max. variations (Ground Line)	50	mV
T_{STG}	Storage temperature range	-55 to +150	°C
T_J	Junction temperature	150	°C
ESD	ESD susceptibility	2000	V
V_{LV}	Input voltage, low voltage pins	GND - 0.3 to V_{DD} + 0.3	V
V_{PWM}	Pin voltage, PWMI, PWMO pins	GND - 18 to VB	V
V_{CAN}	Pin voltage, CAN_H, CAN_L pins	GND - 18 to VB	V
I_{IN}	Input current (low voltage pins)	-25.....+25	mA

1. ISO transient must not reset the device

7.2 Power considerations

The average chip-junction temperature, T_J, in degrees Celsius, may be calculated using the following equation:

$$T_J = T_A + (P_D \times \theta_{JA}) \ (1)^*$$

Where:

- T_A is the Ambient Temperature in °C,
- θ_{JA} is the Package Junction-to-Ambient Thermal Resistance, in °C/W,
- P_D is the sum of P_{INT} and $P_{I/O}$,
- P_{INT} is the product of I_1 and VB, plus the power dissipated by the power bridge, expressed in Watts. This is the Chip Internal Power
- $P_{I/O}$ represents the Power Dissipation on Input and Output Pins; User Determined.

For most applications $P_{I/O} << P_{INT}$ and may be neglected. $P_{I/O}$ may be significant if the device is configured to drive Darlington bases or sink LED Loads.

An approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected) is given by:

$$P_D = K \div (T_J + 273°C) \ (2)$$

Therefore:

$$K = P_D \times (T_A + 273°C) + \theta_{JA} \times P_D^2 \ (3)$$

Where:

- K is a constant for the particular part, which may be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J may be obtained by solving equations (1) and (2) iteratively for any value of T_A.

Table 18. Thermal characteristics (V_B=18V, T_J = 150°C, I_{LOAD} = 2A)

Symbol	Description	Value	Unit
P_D	Dissipated Power	3	W

(*) Maximum chip dissipation can directly be obtained from T_j (max), θ_{JA} and T_A parameters.

Figure 53. HiQUAD-64: θ_{JA}

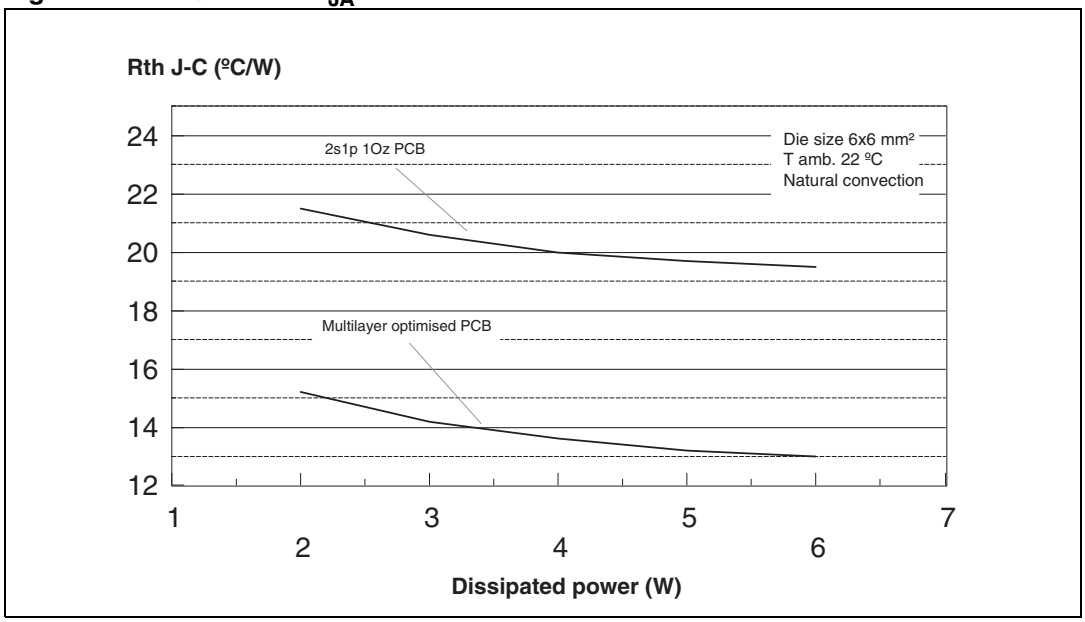
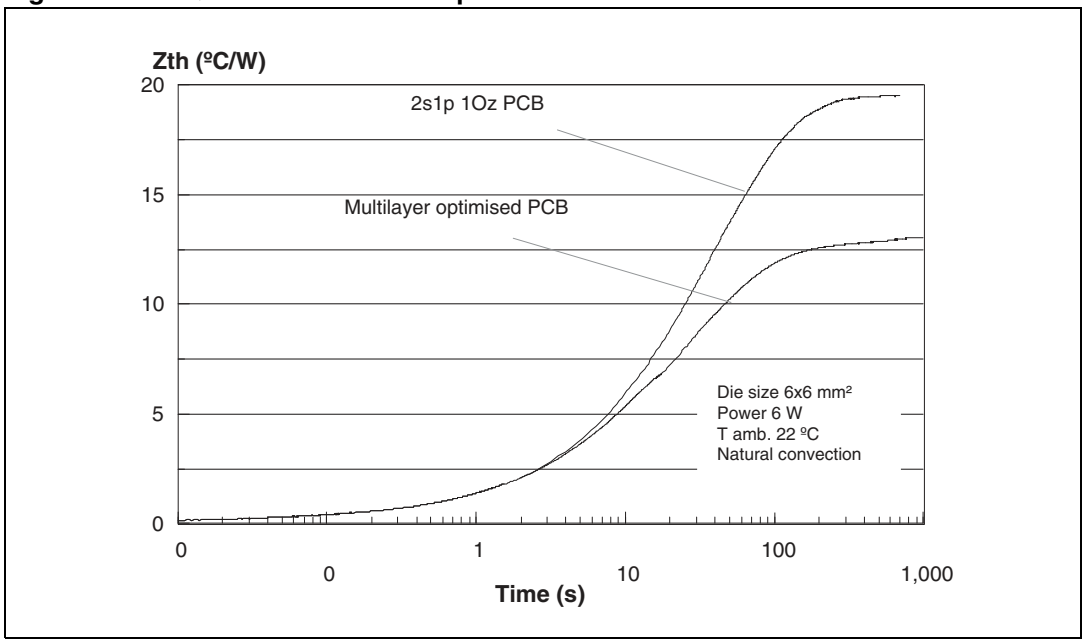
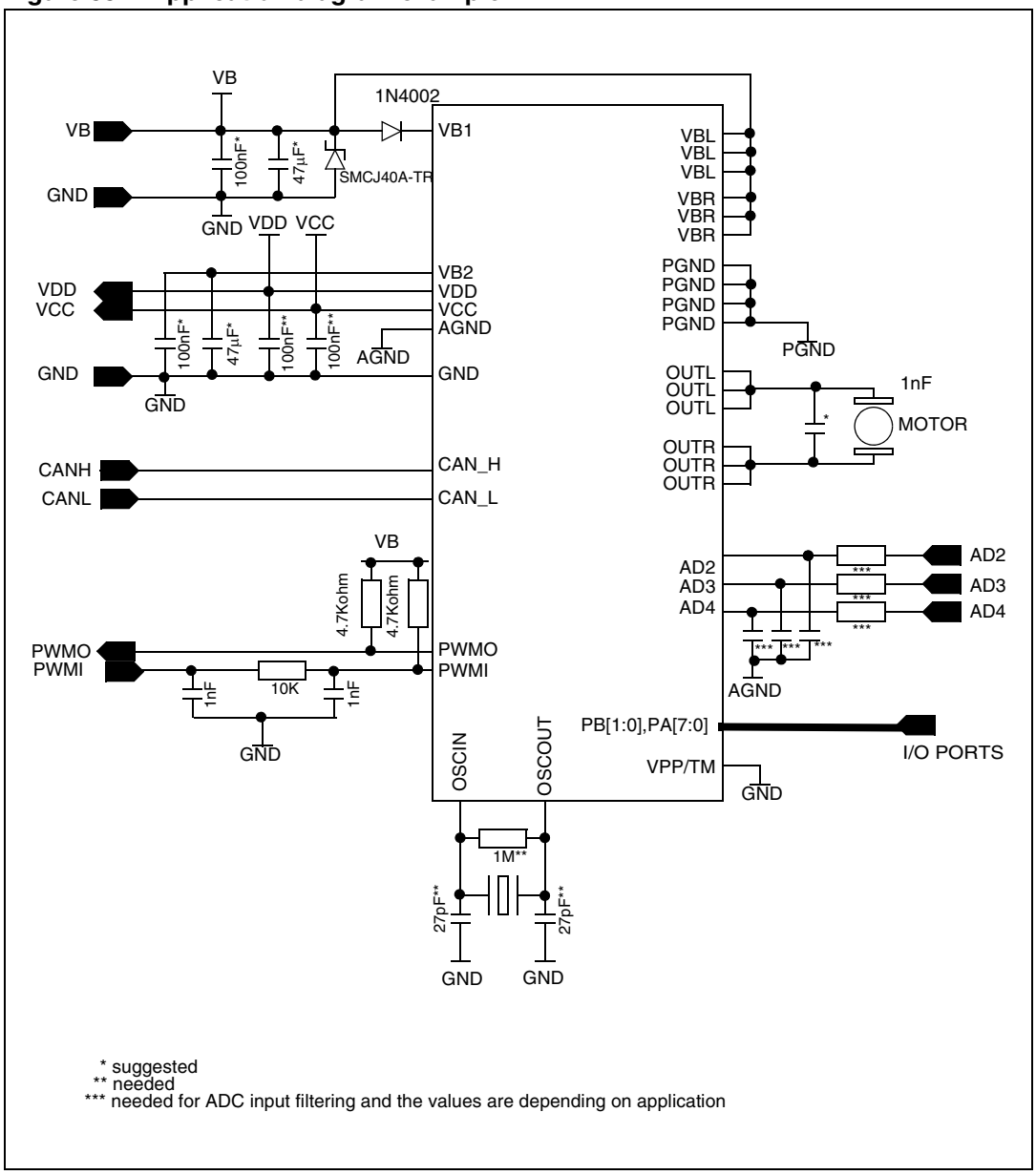


Figure 54. HiQUAD-64: Thermal impedance



7.3 Application diagram example

Figure 55. Application diagram example



7.4 DC electrical characteristics

(T_J = -40 to +150°C, VB=12V unless otherwise specified)

Table 19. DC electrical characteristics(1)

Symbol	Parameter	Conditions	Min.	Typ.	Max	Unit
GENERAL						
VB1	Supply Voltage	f_{OSC} = 16 MHz	6.4	12	18	V
VBR, VBL = VB	Power Supply Voltage	f_{OSC} = 16 MHz	7.1	12	18	V
I1	Supply Current from VB1	No external loads RUN mode WAIT mode Halt mode(2)		 24 21 16		 mA mA mA
I_{IN}	Input Current	Low voltage pins(3)	-5		5	mA
POWER SUPPLY						
VB2	Pre-regulated Voltage	VB1 = 12V	8	10	12	V
VDD	Regulated Voltage	VB1 = 12V	4.75	5	5.25	V
VDD	Regulated Voltage	VB1 = 3.. 6.4V	VB1 - 1.1			V
ΔVDD	Line Regulation	VB1 = 6.4..18V			50	mV
ΔVDD	Load Regulation	I_{VDD}=0..50mA			50	mV
VCC	Regulated Voltage		4.75	5	5.25	V
VCC	Regulated Voltage	VB1 = 3.. 6.4V	VB1 - 1.1			V
ΔVCC	Line Regulation	VB1 = 6.4..18V			50	mV
ΔVCC	Load Regulation	I_{VCC}=0..15mA			50	mV
I_{VDD}	Current sunk from VDD pin				50	mA
I_{VCC}	Current sunk from VCC pin				15	mA
I_{MAXVDD}	Current limit from VDD		150		400	mA
I_{MAXVCC}	Current limit from VCC		50		170	mA
C_{VDD}	External capacitor to be connected to VDD pin			100		nF
C_{VCC}	External capacitor to be connected to VCC pin			100		nF
STANDARD I/O PORT PINS						
V_{IL}	Input Low Level Voltage		-	-	$0.3xV_{DD}$	V
V_{IH}	Input High Level Voltage		$0.7xV_{DD}$	-	-	V

Table 19. DC electrical characteristics(1) (continued)

Symbol	Parameter	Conditions	Min.	Typ.	Max	Unit
V_{OL}	Output Low Level Voltage	I=-5mA	-	-	1.0	V
		I=-1.6mA	-	-	0.4	V
V_{OH}	Output High Level Voltage	I=5mA	3.1	-	-	V
		I=1.6mA	3.4	-	-	V
I_L	Input Leakage Current	$GND<V_{PIN}<V_{DD}$	-10	-	10	µA
I_{RPU}	Pull-up Equivalent Resistance	V_{IN}=GND	40	-	250	KΩ
T_{ohl}	Output H-L Fall Time	C_l=50pF	-	30	-	ns
T_{olh}	Output L-H Rise Time	C_l=50pF	-	30	-	ns

1. All voltage are referred to GND unless otherwise specified.
2. Halt mode is not allowed if Watchdog or Safeguard are enabled
3. A current of 5mA can be forced on each pin of the digital section without affecting the functional behaviour of the device.

7.5 Control timing

(Operating conditions T_j = -40 to +150°C, VB=12V unless otherwise specified)

Table 20. Control timing

Symbol	Parameter	Conditions	Value			Unit
			Min.	Typ.	Max	
f_{OSC}	Oscillator Frequency		0(1)		16	MHz
f_{CPU}	Operating Frequency		0(2)		8	MHz
t_{RL}	External RESET Input pulse Width		1.5			t_{CPU}
t_{PORL}	Internal Power Reset Duration		4096			t_{CPU}
T_{DOGL}	Watchdog or Safeguard RESET Output Pulse Width			500		ns
t_{DOG}	Watchdog Time-out	f_{cpu} = 8 MHz	12,288 1.54		786,432 98.3	t_{CPU} ms
t_{OXOV}	Crystal Oscillator Start-up Time				50	ms
t_{DDR}	Power up rise time				100	ms

1. With Safeguard disabled, A/D operations and Oscillator start-up are not garanteed below 1MHz
2. With Safeguard disabled, A/D operations and Oscillator start-up are not garanteed below 1MHz

7.6 Operating block electrical characteristics

These device-specific values take precedence over any generic values given elsewhere in the document.

(T_j = -40... +150°C, V_{DD} - GND = 5 V unless otherwise specified).

Table 21. A/D converter

A/D Converter						
Symbol	**Parameter**	**Conditions**	**Min.**	**Typ**	**Max**	**Unit**
V_{AL}	Resolution			10		bit
AE	Absolute Error		-2		2	LSB
FSC	Full Scale Error		-1		1	LSB
ZOE	Zero Offset Error		-1		1	LSB
NLE	Non Linearity Error		-2		2	LSB
DNLE	Differential Non Linearity Error		-1/2		1/2	LSB
tc	Conversion Time	f_{cpu} = 8MHz			20	µs
IL	Leakage current		-0.5		0.5	µA
Vin	Input Voltage		0		VCC	V
T_{SENS}	Temperature sensing range		-40		150	°C
T_{SENSR}	Temperature sensor resolution		1			LSB/°K
T_{SENSE}	Temperature sensor error (T in °K)				±2(1)	%

1. After trimming, being T_{TRIM} the trimming temperature, the specified precision can be achieved in the range T_{TRIM}-80, max[T_{TRIM}+80, 150°C]. Precision is related to the read temperature in Kelvin.

Table 22. POWER bridge

Symbol	Parameter	Conditions	Min.	Typ	Max	Unit
RdsON	Output Resistance	Measured on OUTL and OUTR.			160	mΩ
RdsON @ 25°C	Output Resistance	Measured on OUTL and OUTR.		90		mΩ
ISC	Short circuit current	Short to VBL,VBR, GND: load short	6	8	11	A
tSCPI	Short circuit protection intervention time			12		ms
Thw	Thermal shutdown threshold		165	175	185	°C
Thwh	Thermal shutdown threshold hysteresis			20		°C

Table 22. POWER bridge (continued)

Symbol	Parameter	Conditions	Min.	Typ	Max	Unit
trp	OUTL, OUTR rise time	measured from 10% to 90%		3		ms
tfp	OUTL, OUTR fall time	measured from 10% to 90%		1.5		ms

Table 23. EEPROM

Parameter	Min.	Max	Unit
Write time		4.0	ms
Write Erase Cycles	50000		Cycles
Data Retention	10		Years

Table 24. PWM output

Symbol	Parameter	Conditions	Min.	Typ	Max	Unit
V_{OH}	Output Voltage High	RL = 500Ω to VB	VB-0.2		VB	V
V_{SL}	Saturation Voltage Low	IO = 20mA	0		0.5	V
I_{IO}	Input Current	VB = 12V	0		25	µA
I_{PSC}	Short circuit current		30	60	100	mA

Table 25. PWM input

Symbol	Parameter	Conditions	Min.	Typ	Max	Unit
V_{TL}	Input state low	VB=VBR	-1		0.45*VB	V
V_{TH}	Input state high	VB=VBR	0.55VB		VB	V
V_{H}	Hysteresis	VB=VBR		0.025*VB	0.8	V
I_{II}	Input Current	VB = 12V	0.5		8	µA

Table 26. Oscillator safeguard

Symbol	Parameter	Conditions	Min.	Typ	Max	Unit
f_{low}	reset low frequency		0.6	1.1	1.7	MHz
f_{high}	reset high frequency		17	24	31	MHz

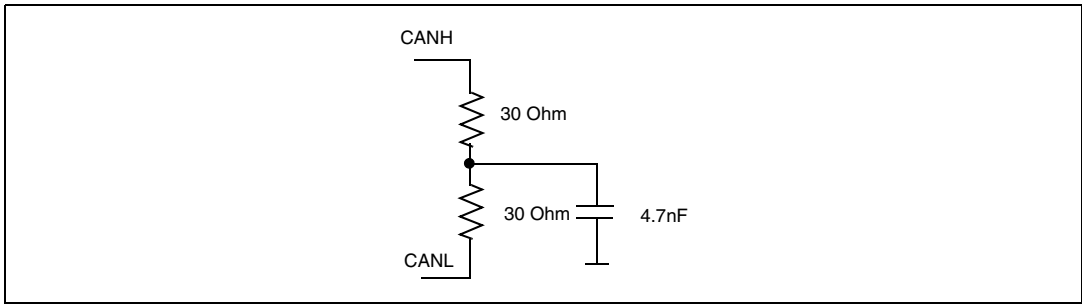
Table 27. CAN transceiver

Rl = 60 Ohm, see note 1, unless otherwise specified.						
Symbol	**Parameter**	**Conditions**	**Min.**	**Typ**	**Max**	**Unit**
V_{CANHL_R}	Recessive State CAN_H, CAN_L Output Voltage	TX=High Level V_{DD} = 5V I_{CANH} = I_{CANL} = 0	2.0	2.5	3.5	V
V_{DIFF_R}	Recessive State Differential Output Voltage (V_{DIFF} = V_{CANH} - V_{CANL})	TX=High Level I_{CANH} = I_{CANL} = 0	-500	0	50	mV
V_{CANH_D}	Dominant State CAN_H Output Voltage	TX = Low Level; VDD = 5V	2.75	3.5	4.5	V
V_{CANL_D}	Dominant State CAN_L Output Voltage	TX = Low Level; VDD = 5V	0.5	1.5	2.25	V
V_{DIFF_D}	Dominant State Differential Output Voltage (V_{DIFF} = V_{CANH} - V_{CANL})	TX = Low Level; VDD = 5V	1.5	2.0	3.0	V
I_{SC}	CAN_H, CAN_L Short Circuit Threshold Current		90		200	mA
V_{REC}	Differential Input Voltage for Recessive State (V_{DIFF} = V_{CANH} - V_{CANL})	V_{CANL} = -2V; TX = High Level V_{CANH} = 6.5V: TX = High Level			200	mV
V_{DOM}	Differential Input Voltage for Dominant State (V_{DIFF} = V_{CANH} - V_{CANL})	V_{CANL} = -2V; TX = High Level V_{CANH} = 6.1V: TX = High Level	900			mV
V_{DIFF_HYS}	Differential Input Voltage Hysteresis			150		mV
t_{TD}	Delay Time from TX to V_{DIFF} = V_{CANH} - V_{CANL}				50	ns
t_{DR}	Delay Time from V_{DIFF} to RX V_{DIFF} = V_{CANH} - V_{CANL}				150	ns
t_D	Disabled Transmission Time for Overcurrent Protection		1	5	10	us
SR_H	VCANH Slew Rate Between 20% and 80%		15		80	V/µs
SR_L	VCANL Slew Rate Between 20% and 80%		15		80	V/µs
t_{TR}	Delay Time from TX to RX				200	ns
$1/t_{bit}$	Transmission speed	non return to zero			1	Mb/s

Table 28. Power on/low voltage reset

Symbol	Parameter	Conditions	Min.	Typ	Max	Unit
$V_{Reset\ L}$	Input low level voltage	NRESET pin			0.3VDD	V
$V_{Reset\ H}$	Input high level voltage	NRESET pin	0.7VDD			V
$I_{Reset\ L, H}$	Input current	NRESET pin VDD = 5V Leakage current Internal reset by watchdog or POR Pull up current source	 30		 1 1 90	 µA mA µA
$V_{Reset\ UD}$	VDD for RESET undefined	Below this voltage RESET is not defined			2	V
$V_{Reset\ ON}$	VDD low level for RESET on		3.1	3.3		V
$V_{Reset\ OFF}$	VDD high level for RESET off			3.8	4.5	V

Note 1:



8 Package information

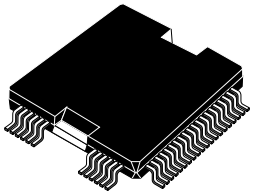
In order to meet environmental requirements, ST offers these devices in ECOPACK® packages. These packages have a Lead-free second level interconnect. The category of second Level Interconnect is marked on the package and on the inner box label, in compliance with JEDEC Standard JESD97. The maximum ratings related to soldering conditions are also marked on the inner box label. ECOPACK is an ST trademark. ECOPACK specifications are available at: www.st.com.

Figure 56. HiQUAD-64 mechanical data and package dimensions

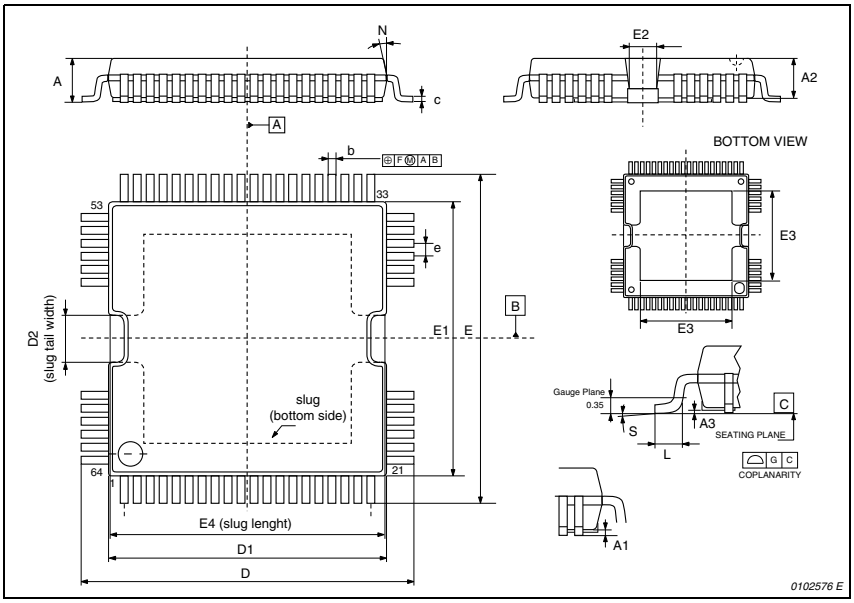
DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			3.15			0.124
A1	0		0.25	0		0.010
A2	2.50		2.90	0.10		0.114
A3	0		0.10	0		0.004
b	0.22		0.38	0.008		0.015
c	0.23		0.32	0.009		0.012
D	17.00		17.40	0.669		0.685
D1 (1)	13.90	14.00	14.10	0.547	0.551	0.555
D2	2.65	2.80	2.95	0.104	0.110	0.116
E	17.00		17.40	0.669		0.685
E1 (1)	13.90	14.00	14.10	0.547	0.551	0.555
e		0.65			0.025	
E2	2.35		2.65	0.092		0.104
E3	9.30	9.50	9.70	0.366	0.374	0.382
E4	13.30	13.50	13.70	0.523	0.531	0.539
F		0.12			0.005	
G		0.10			0.004	
L	0.80		1.10	0.031		0.043
N	10 (max.)					
S	0 (min.), 7 (max.)					

(1): "D1" and "E1" do not include mold flash or protusions
- Mold flash or protusions shall not exceed 0.15 mm (0.006 inch)

OUTLINE AND MECHANICAL DATA



HiQUAD-64



9 Revision history

Table 29. Document revision history

Date	Revision	Changes
29-May-2006	1	Initial release.
8-Jun-2006	2	Corrected typo errors.
24-Sep-2013	3	Updated disclaimer.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

ST PRODUCTS ARE NOT DESIGNED OR AUTHORIZED FOR USE IN: (A) SAFETY CRITICAL APPLICATIONS SUCH AS LIFE SUPPORTING, ACTIVE IMPLANTED DEVICES OR SYSTEMS WITH PRODUCT FUNCTIONAL SAFETY REQUIREMENTS; (B) AERONAUTIC APPLICATIONS; (C) AUTOMOTIVE APPLICATIONS OR ENVIRONMENTS, AND/OR (D) AEROSPACE APPLICATIONS OR ENVIRONMENTS. WHERE ST PRODUCTS ARE NOT DESIGNED FOR SUCH USE, THE PURCHASER SHALL USE PRODUCTS AT PURCHASER’S SOLE RISK, EVEN IF ST HAS BEEN INFORMED IN WRITING OF SUCH USAGE, UNLESS A PRODUCT IS EXPRESSLY DESIGNATED BY ST AS BEING INTENDED FOR “AUTOMOTIVE, AUTOMOTIVE SAFETY OR MEDICAL” INDUSTRY DOMAINS ACCORDING TO ST PRODUCT DESIGN SPECIFICATIONS. PRODUCTS FORMALLY ESCC, QML OR JAN QUALIFIED ARE DEEMED SUITABLE FOR USE IN AEROSPACE BY THE CORRESPONDING GOVERNMENTAL AGENCY.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.
Information in this document supersedes and replaces all information previously supplied.
The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com