AN1526
APPLICATION NOTE

ST7FLITE0 QUICK REFERENCE NOTE

by Microcontroller Division Applications

INTRODUCTION

The ST7FLITE0 extends the lower end of the ST7 range, designed to fit applications with state-of-the-art features in a very small package.

ST7FLITE0 8-bit MCUs have many cost-saving features and they come with low-cost tools, providing a complete package to reduce both design and final application costs to the absolute minimum.

This application note highlights some small but very important aspects of the ST7FLITE0 that users should not overlook when reading the datasheet.

Sections 2 and 3 contain helpful pointers and a table to help you start working with ST7FLITE0 and its related tools.

Table 1. ST7Lite0 Features

Program memory	1.5 Kbytes Flash (single voltage)
RAM	128 bytes
Data EEPROM	ST7FLITE05: no Data EEPROM ST7FLITE09: 128 bytes Data EEPROM
Package	SO16 (.150) or DIP16
Number of I/O pins	13 (including 6 high current pins)
Vdd range	2.4V - 5.5V
Temperature range	-40°C to +85°C
IDDmax in RUN mode	5mA
LVD	3 levels (Standard)
Clock sources	Internal RC (1MHz) +/- 1% PLLx4 (2.4V<V_{dd}<3.3V) PLLx8 (3.3V<V_{dd}<5.5V)
A/D	8 bits with fixed gain - Op-Amp 5 channels
Timers	Autoreload Timer: 1 12-bit PWM channel, Output Compare function Lite Timer: Watchdog, Real Time Clock, Input Capture functions
Communication peripheral	SPI
Special features	Read-out protection, In-Circuit Programmming (ICP), In Application Programming (IAP)

Table of Contents

1 ST7FLITE0 KEY ADVANTAGES

1.1 LOW-COST 8-BIT MICROCONTROLLER SOLUTION

The ST7FLITE0 is a small low-cost microcontroller, embedding many analog functions. It therefore allows you to save board area and the cost of external components. There is no need for:

- external ceramic resonator for accurate oscillators (see 3.2)

- external E^2data (see 3.4)

- external reset circuitry

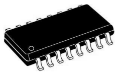
- external LVD circuitry

- PCB rework for SW update

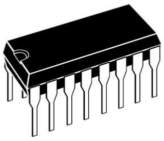
- large PCB

1.2 AREA OPTIMIZATION

ST7FLITE0 gives you the choice of two small 16-pin packages. You can take advantage of this to save area on your PCB.



SO16: 6x9.9mm



DIP16:10.92x19.18mm

ST7FLITE0 actual-size footprint of 16-Pin SO Package - Typical: 6mm wide, 9.9mm long

1.3 HIGH ACCURACY INTERNAL 1MHZ RC OSCILLATOR

The ST7Flite0 contains an internal RC oscillator with an accuracy of 1% for a given device, temperature and voltage. It must be calibrated to obtain the frequency required in the application. This is done by software writing a calibration value in the RCCR (RC Control Register).

Whenever the ST7FLITE0 microcontroller is reset, the RCCR returns to its default value (FFh), i.e. each time the device is reset, the calibration value must be loaded in the RCCR. Predefined calibration values are stored in E^{2}PROM for 3.0 and 5V VDD supply voltages at 25°C, as shown in the following table.

RCCR	Conditions	ST7FLITE09 Address	ST7FLITE05 Address
RCCR0	V_{DD} =5V T_A =25°C f_{RC} =1MHz	1000h and FFDEh	FFDEh
RCCR1	V_{DD} =3V T_A =25°C f_{RC} =700kHz	1001h and FFDFh	FFDFh

If the voltage or temperature conditions change in the application, the frequency may need to be recalibrated.

Refer to AN1324 “Calibrating The RC Oscillator of the ST7FLITE0 MCU using the Mains”

for information on how to calibrate the RC frequency using an external reference signal.

Decoupling capacitor for stable oscillator

Vdd and Gnd levels impact the stability of the 1% internal RC of the ST7FLITE0. To reach the most stable oscillation, use decoupling capacitors between Vdd and Gnd pins, at values recommended in the datasheet.

RCCR calibration locations can be Write Protected

The STVP or InDART tools allow you to prevent unintentional write access to the RCCR calibration locations.

1.4 8-BIT A/D CONVERTER WITH INPUT VOLTAGE AMPLIFIER (X8)

The 8-bit A/D converter allows up to 5 channels with multiplexed inputs.

You can take advantage of its internal fixed-gain amplifier (x8) for zooming on low voltage inputs. For Vdd=5V, the ADC can therefore convert from 0V to 250mV. In this case the ADC ideal resolution is 2.4mV, equivalent to a 11-bit resolution with input range from 0V to Vdd.

1.5 TRUE E^{2}PROM DATA

The ST7FLITE09 contains an Electrically Erasable Programmable Read-Only Memory which can be used as a non volatile back-up for storing data. It allows up to 300K Write/Erase cycles at 25°C.

Main Features

- Up to 32 Bytes programmed in the same cycle
- mono-voltage EEPROM (charge pump)
- Chained erase and programming cycles
- Internal control of the global programming cycle duration
- WAIT mode management
- Read-out protection against piracy

Useful Tips:	***ST7FLITE05: Emulate Data E^{2}PROM with program area*** For devices with no Data E^{2}PROM, you can emulate Data EEPROM with the XFlash Memory. For more information, refer to: AN1477: Emulated Data EEPROM with XFlash memory

1.6 SAFE PROTECTION AGAINST PIRACY ON DATA E2PROM AND FLASH

Data stored in the E^{2}PROM memory, as well as the program memory, are protected against read-out piracy. This is managed by the option bit “Read-out protection”. Removing this option by erasing the option byte will cause the whole memory to be erased first.

Flash program memory can also be protected against re-write operation by option bit “Flash Write protection”. When this option is selected, the program memory can never be erased or programmed again.

1.7 IN-CIRCUIT PROGRAMMING AND IN-APPLICATION PROGRAMMING CAPABILITIES

In addition to using a programming tool, the two following modes allow you to program your ST7LITE0 without removing it from your PCB.

ICP: In-Circuit-Programming: The ICP is the ability to program the Flash memory (FLASH sectors 0 and 1, option byte row and data EEPROM) of a microcontroller using ICC (In-Circuit Communication) protocol while the device is already plugged-in to the application, but application is not running.

IAP: In-Application-Programming: The IAP is the ability to re-program the FLASH memory (FLASH sector 1 and data EEPROM) of a microcontroller while the device is already plugged-in to the application and the application is running. As sector 0 contains the software driver to be able to re-program, it is write protected, therefore not reprogrammable.

In ICP minimum configuration, only 3 wires are needed (ICCCLK, ICCDATA, $\overline{\text{RESET}}$). The user may also use it as a basis to develop its own debugging tool.

Programming time:

ICP allows 1kbytes to be programmed/erased in 160ms.

Typical measurements: 1.5k Flash programming time: 315ms

For more information, refer to the Programming Manuals:

. ST7 Flash Programming Reference Manual

. ST7 ICC Protocol Reference Manual

Useful Tips:

Sector 0 size configurable by Option Byte

As sector 0 is not re-programmable in IAP programming mode, you may take advantage of configuring its size to optimize Program Memory.

By Option Byte, sector 0 size can be set at 0.5 Kbytes, 1 Kbytes, or 1.5 Kbytes.

2 ST7LITE0 DEVELOPMENT TOOLS

Table 2 shows a summary of available tools for each function.

Table 2. Available Tools

	SIMULATE	EMULATE	DEBUG & PROGRAM	GRAPHIC DESIGN & DEBUG	PROGRAM
SOFTWARE	STVD7 Simulator	STVD7	STVD7 (included in ST7FLITE0-INDART package)	ST-REALIZERII	STVP7
HARDWARE	no need	ST7MDT10-EMU3	ST7FLITE0-INDART	no need	ST7MDT10-EPB

Note: Third-party C-compiler tool chains can be used with STVD7 interface.

2.1 SOFTWARE TOOLS

2.1.1 ST7 Visual Debug IDE - Reference: STVD7

Visual interface for C or Assembler coding, compile, download and debug with ST7 Emulators or ST7FLITE0-INDART.

Web: http://mcu.st.com → ST7

2.1.2 STVD7 Simulator

Stand-alone tool which allows to write code, compile, and simulate an ST7FLITE0 with your PC your ST7FLITE0.

Web: http://mcu.st.com → ST7

2.1.3 C Compiler toolchains from Cosmic and Metrowerks

- Free evaluation version limited to 1K
- Low-cost lite C compiler limited to 8K

This compilers can be embedded in STVD7 IDE interface, or used through dedicated graphical interfaces.

Web: Cosmic Software Inc.: www.cosmic-software.com

Metrowerks: www.metrowerks.com

2.1.4 ST7 Visual Programmer - Reference: STVP7

Visual Interface allowing to program Flash, Data E^2PROM, and option bytes. This is the software part of the whole programming tool package. See 4.2.3 for more details.

2.2 HARDWARE TOOLS

2.2.1 In-Circuit Debugging Kit - Reference: ST7FLITE0-INDART

ST7FLITE0-INDART includes a full-featured experiment board, an in-circuit programming utility, and all tools required to develop custom embedded applications, at low cost. 100% of electrical characteristics are guaranteed by use of a standard chip, not bondout chip.

ST7FLITE0-INDART contains STVD7 graphical interface, C compiler and assembler.

Debugging capabilities: Real-time emulation, breakpoints, step capabilities, read/write memory and registers.

Programming capabilities: Blank-check, Program, Read, Verify Flash, E^{2}PROM memory and Option Bytes.

Notes:

1. It can be ordered from ST, or directly from Softec. Softec reference: inDART-ST7FLITE0

2. No power supply is delivered with this kit. A typical 5V may be used.

Web: http://www.softecmicro.com/indart-st7flite0.html

e-mail: info@softecmicro.com

2.2.2 ST Emulator - Reference: ST7MDT10-EMU3

Advanced development tool including:

- Real-time emulator (Advanced breakpoints management (through Bus Event Machine), 256K real-time trace recording, read/write on the fly through Watch and Memory windows, Performance analysis)
- STVD7 User Interface
- Set of probes for Lite family packages.
- Parallel interface cable to PC
- Power Supply

Note: the ST Emulator cannot program parts. Therefore, a separate device programmer is required (as ST7 Programming tool (see 4.2.3))

Web: http://mcu.st.com → ST7

2.2.3 ST Programming tool - Reference: ST7MDT10-EPB

Main features: Blank-check, Program, Read, Verify Flash, E^{2}PROM memory and Option Bytes.

ICP programming mode is supported. All Lite0 packages included.

This is done through **STVP7: ST7 Visual Programmer**. Visual Interface allowing to program Flash, Data E^{2}PROM, and option bytes.

Web: http://mcu.st.com → ST7

2.2.4 ST Graphic Design and Debug - Reference: STREALIZER-II

The ST Realizer allows you to graphically design applications for the ST7 microcontroller family, without any prerequisite knowledge in assembler programming. With a single click, using the powerful graphics editor and compiler, you can generate complete software applications for the ST7FLite0.

Main features: schematic-based Design, Analysis, Simulator for debugging

Web: http://mcu.st.com → ST7

3 START TODAY

1) Go to ST's website (http://mcu.st.com) and download:

- STVD7 software

and these items:

- ASM/LYN software
- Cosmic C compiler Demo software
- Metrowerks C compiler software
- ST Visual Programmer

2) Install all software and follow “Getting Started” in STVD7.

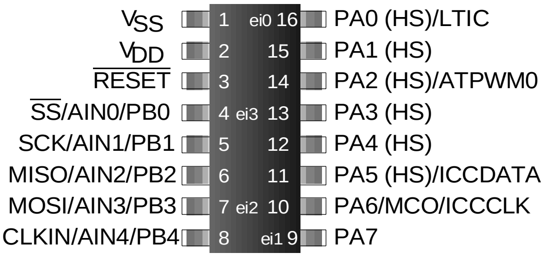
3) Copy certain files for the “Getting Started” example and create your own project in STVD7.

4) Order tools:

- EMU - ST's or inDART
- EPB
- C compiler - Cosmic or Metrowerks

4 ST7FLITE0 8-BIT MCU - EASY REFERENCE

Figure 1. Pinout


V_{SS} 1 ei0 16 PA0 (HS)/LTIC
V_{DD} 2 15 PA1 (HS)
$\overline{\text{RESET}}$ 3 14 PA2 (HS)/ATPWM0
$\overline{\text{SS}}$/AIN0/PB0 4 ei3 13 PA3 (HS)
SCK/AIN1/PB1 5 12 PA4 (HS)
MISO/AIN2/PB2 6 11 PA5 (HS)/ICCDATA
MOSI/AIN3/PB3 7 ei2 10 PA6/MCO/ICCCLK
CLKIN/AIN4/PB4 8 ei1 9 PA7

HS: 20 mA High Sink Capability
eix: Associated External Interrupt Vector

Table 3. Device Summary

Features	ST7FLITE09	ST7FLITE05
Program Memory (Bytes)	1.5K FLASH	1.5K FLASH
RAM/Stack (Bytes)	128/64	128/64
Data EEPROM (Bytes)	128	-
Peripherals	Lite Timer w/ Watchdog, Autoreload Timer w/ 1 PWM, SPI, 8-bit ADC w/ Op-Amp	
Operating Supply	2.4 V to 5.5 V	
CPU Frequency	1 MHz RC 1% PLL x4/8MHz	
Operating Temperature	-40 C to +85 C (-40 C to +105/125 C Optional)	

Table 4. Pin Chart

Pin	Name	Main Function	Alternate Function
1	V_{SS}	Ground	
2	V_{DD}	Main Power Supply	
3	$\overline{\text{RESET}}$	Top-Priority Non Maskable Interrupt (active low)	
4	$\overline{\text{SS}}$/AIN0/PB0	PORT B0	ADC Analog Input 0 or SPI Slave Select (active low)
5	SCK/AIN1/PB1	PORT B1	ADC Analog Input 1 or SPI Serial Clock
6	MISO/AIN2/PB2	PORT B2	ADC Analog Input 2 or SPI Master In/Slave Out Data
7	MOSI/AIN3/PB3	PORT B3	ADC Analog Input 3 or SPI Master Out/Slave In Data
8	CLKIN/AIN4/PB4	PORT B4	ADC Analog Input 4 or External Clock Input
9	PA7	PORT A7	
10	PA6 /MCO/ICCCLK	PORT A6	Main Clock Output or In-Circuit Communication Clock
11	PA5 (HS)/ICCDATA	PORT A5	In-Circuit Communication Data
12	PA4 (HS)	PORT A4	
13	PA3 (HS)	PORT A3	
14	PA2 (HS)/ATPWM0	PORT A2	Auto-Reload Timer PWM0
15	PA1 (HS)	PORT A1	
16	PA0 (HS)/LTIC	PORT A0	Lite Timer Input Capture

Table 5. Instruction Set

Mnemonic	Description	Operation	Dest.	Source	Flags
ADC d, s	Add with carry, s to d	d ⇐ d + s + C	A	mem	H, N, Z, C
ADD d, s	Add s to d	d ⇐ d + s	A	mem	H, N, Z, C
AND d, s	Logical AND (d with s)	d ⇐ d AND s	A	mem	N, Z
BCP s, d	Bit compare A, mem	{N, Z} ⇐ s AND d	A	mem	N, Z
BRES d, b	Bit reset d	d ⇐ d AND (2^b)	mem	-	-
BSET d, b	Bit set d	d ⇐ d OR (2^b)	mem	-	-
BTJF d, b, rel	Jump if bit is false (0)	PC ⇐ PC + rel IF (d AND (2^b)) = 0	mem	-	C
BTJT d, b, rel	Jump if bit is true (1)	PC ⇐ PC + rel IF (d AND (2^b)) ≠ 0	mem	-	C
CALL d	Call subroutine	PUSH (PC + length); PC ⇐ d	mem	-	-
CALLR d	Call subroutine relative	PUSH (PC + length); PC ⇐ PC + d	mem	-	-
CLR d	Clear d	d ⇐ 0	reg, mem	-	N = 0, Z = 1
CP d, s	Arithmetic compare	{N, Z, C} ⇐ TEST (d - s)	reg	mem	N, Z, C
CPL d	Logical complement of d	d ⇐ d XOR FFh	reg, mem	-	N, Z, C = 1
DEC d	Decrement d	d ⇐ d - 1	reg, mem	-	N, Z
HALT	Halt	I ⇐ 0	-	-	I = 0
INC d	Increment d	d ⇐ d + 1	reg, mem	-	N, Z
IRET	Interrupt routine return	POP CC, A, X, PC	-	-	H, I, N, Z, C
JP d	Absolute jump	PC ⇐ d	mem	-	-
JRA d	Jump relative always	PC ⇐ PC + d	mem	-	-
JRT d	Jump relative if true	PC ⇐ PC + d	mem	-	-
JRF d	Never jump	-	mem	-	-
JRIH d	Jump Relative if Port INT pin = 1	PC ⇐ PC + d IF interrupt line high	mem	-	-
JRIL d	Jump Relative if Port INT pin = 0	PC ⇐ PC + d IF interrupt line low	mem	-	-
JRH d	Jump Relative if H = 1	PC ⇐ PC + d IF H = 1	mem	-	-
JRNH d	Jump Relative if H = 0	PC ⇐ PC + d IF H = 0	mem	-	-
JRM d	Jump Relative if I = 1	PC ⇐ PC + d IF I = 1	mem	-	-
JRNM d	Jump Relative if I = 0	PC ⇐ PC + d IF I = 0	mem	-	-
JRMI d	Jump Relative if N = 1	PC ⇐ PC + d IF N = 1	mem	-	-
JRPL d	Jump Relative if N = 0	PC ⇐ PC + d IF N = 0	mem	-	-
JREQ d	Jump Relative if Z = 1	PC ⇐ PC + d IF Z = 1	mem	-	-
JRNE d	Jump Relative if Z = 0	PC ⇐ PC + d IF Z = 0	mem	-	-

Table 5. Instruction Set

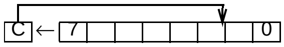
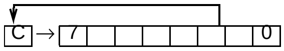
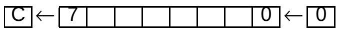
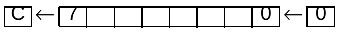
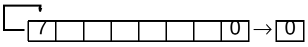
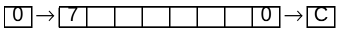
Mnemonic	Description	Operation	Dest.	Source	Flags
JRC d	Jump Relative if C = 1	PC ⇐ PC + d IF C = 1	mem	-	-
JRNC d	Jump Relative if C = 0	PC ⇐ PC + d IF C = 0	mem	-	-
JRULT d	Jump Relative if C = 1	PC ⇐ PC + d IF C = 1	mem	-	-
JRUGE d	Jump Relative if C = 0	PC ⇐ PC + d IF C = 0	mem	-	-
JRUGT d	Jump Relative if (C + Z) = 0	PC ⇐ PC + d IF (C OR Z) = 1	mem	-	-
JRULE d	Jump Relative if (C + Z) = 1	PC ⇐ PC + d IF (C OR Z) = 0	mem	-	-
LD d, s	Load s in d	d ⇐ s	reg, mem	mem, reg	N, Z
MUL d, s	Multiply d by s	d:s ⇐ d * s	A, X, Y	A, X, Y	H = 0, C = 0
NEG d	Negate d (logical 2-complement)	d ⇐ (d XOR FFh) + 1	reg, mem	-	N, Z, C
NOP	No operation	-	-	-	-
OR d, s	Logical OR (d with s)	d ⇐ d OR s	A	mem	N, Z
POP d	Pop from the Stack	d ⇐ (++SP)	reg, CC	-	H, I, N, Z, C
PUSH d	Push onto the Stack	(SP- -) ⇐ d	-	reg, CC	-
RCF	Reset carry flag	C ⇐ 0	-	-	C = 0
RET	Subroutine return	POP PC	-	-	-
RIM	Reset interrupt mask	I ⇐ 0	-	-	I = 0
RLC d	Rotate left through carry		reg, mem	-	N, Z, C
RRC d	Rotate right through carry		reg, mem	-	N, Z, C
RSP	Reset Stack pointer	SP ⇐ reset value	-	-	-
SBC d, s	Subtract s from d with carry	d ⇐ d - s - C	A	mem	N, Z, C
SCF	Set carry flag	C ⇐ 1	-	-	C = 1
SIM	Set interrupt mask	I ⇐ 1	-	-	I = 1
SLA d	Shift left arithmetic (equal to SLL 1)		reg, mem	-	N, Z, C
SLL d	Shift left logical		reg, mem	-	N, Z, C
SRA d	Shift right arithmetic (equal to SRL 1)		reg, mem	-	N, Z, C
SRL d	Shift right logical		reg, mem	-	N = 0, Z, C
SUB d, s	Subtract s from d	d ⇐ d - s	A	mem	N, Z, C
SWAP d	Swap nibbles	d (7:4) ⇔ d (3:0)	reg, mem	-	N, Z
TNZ d	Test for negative and zero	{N, Z} ⇐ TEST (d)	reg, mem	-	N, Z

Table 5. Instruction Set

Mnemonic	Description	Operation	Dest.	Source	Flags
TRAP	Software trap	PC ⇐ PC + 1; PUSH PC, X, A, CC; PC ⇐ trap vector	-	-	I = 1
WFI	Wait for interrupt	I ⇐ 0	-	-	I = 0
XOR d, s	Logical exclusive OR (d with s)	d ⇐ d XOR s	A	mem	N, Z

Figure 2. Block Diagram

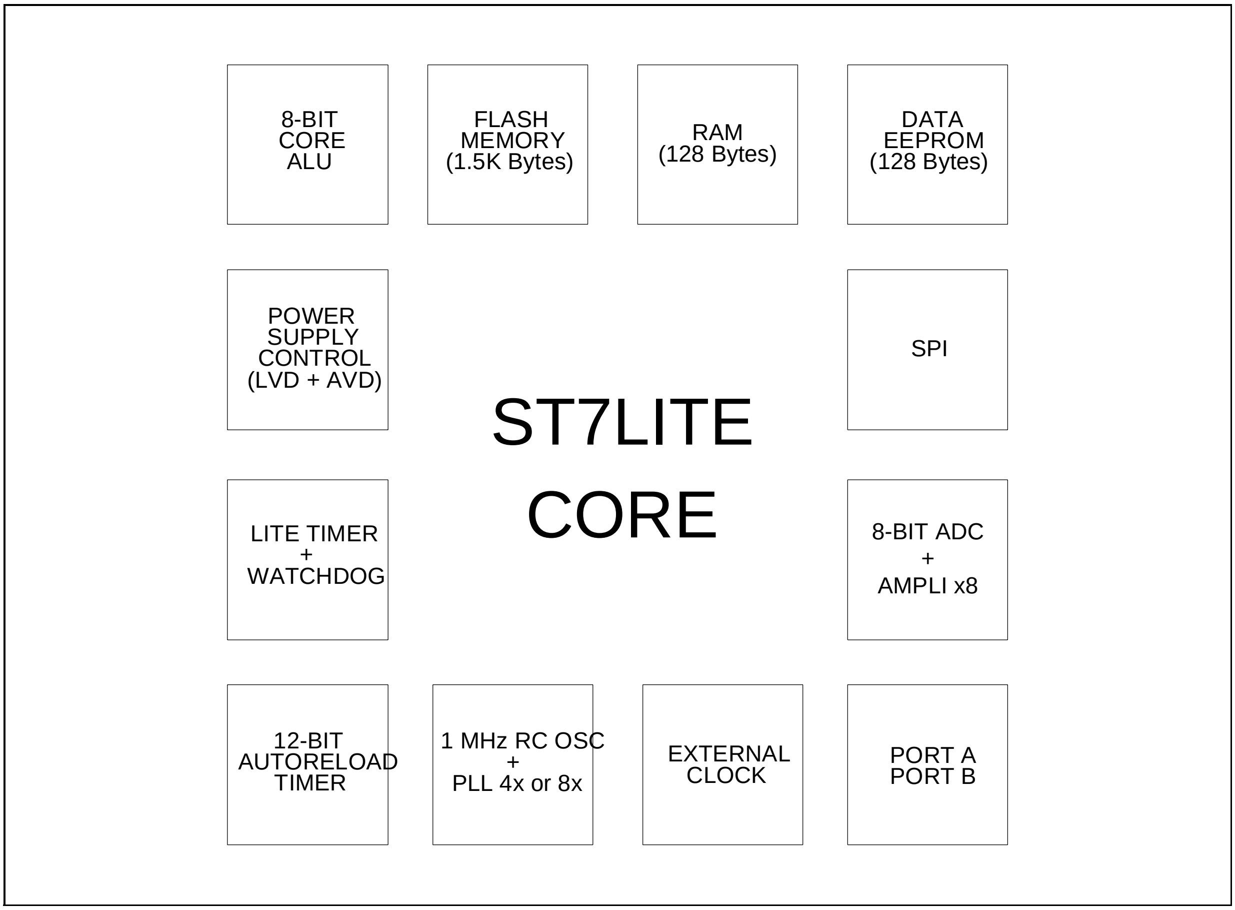
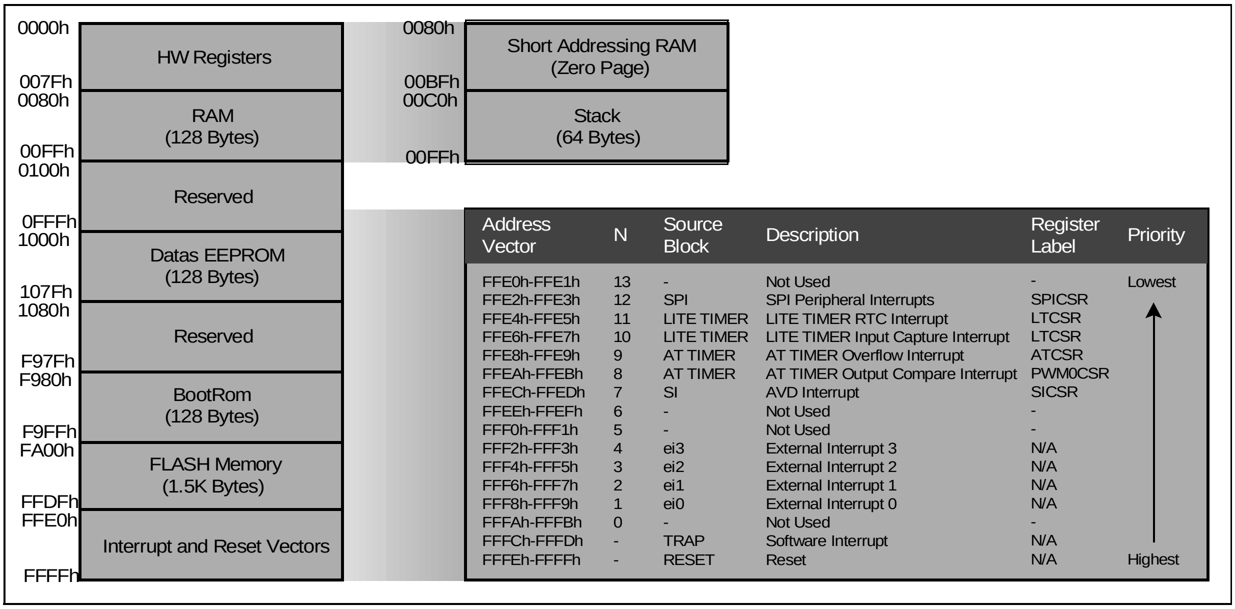
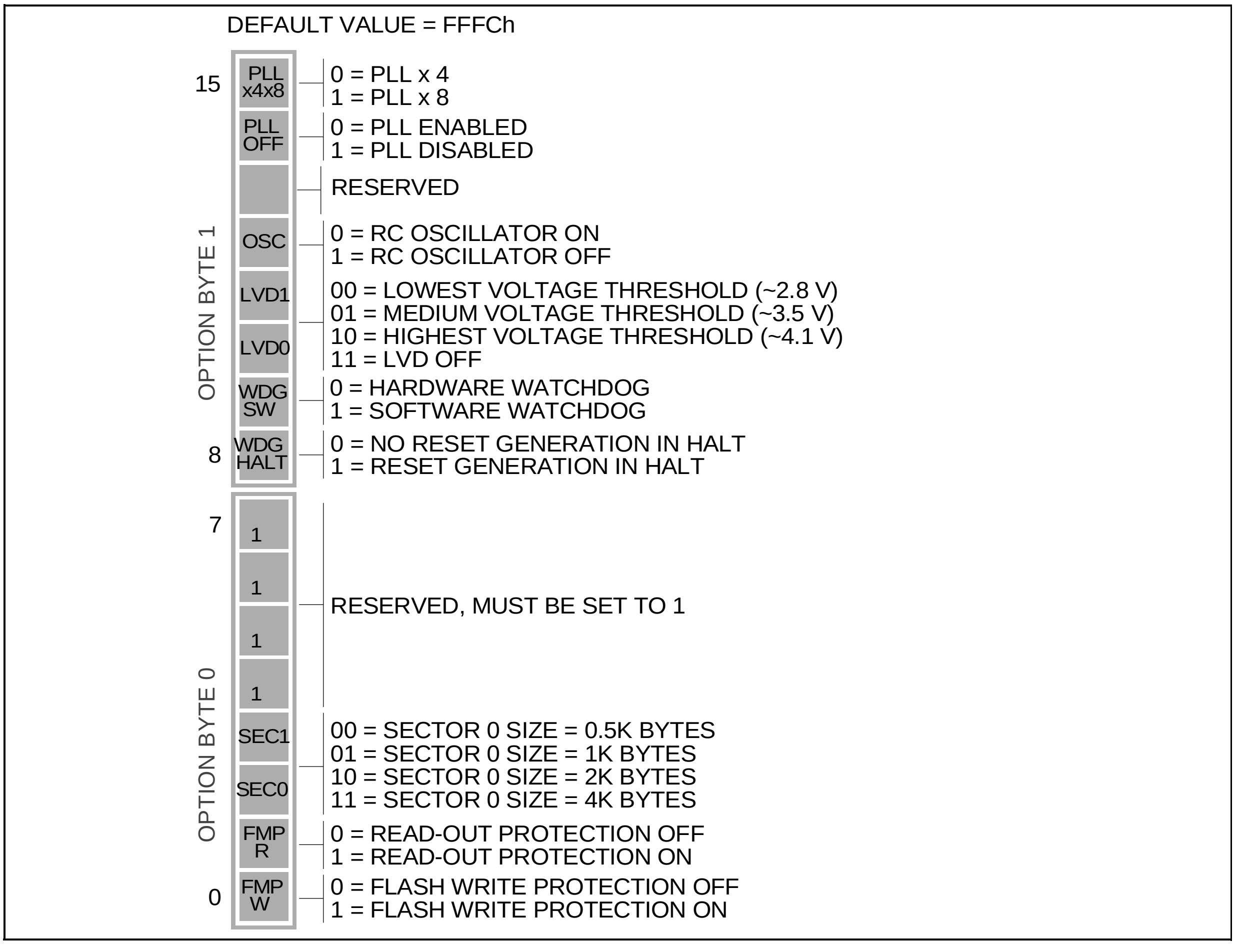


Figure 3. Memory Map



Address Vector	N	Source Block	Description	Register Label	Priority
FFE0h-FFE1h	13	-	Not Used	-	Lowest
FFE2h-FFE3h	12	SPI	SPI Peripheral Interrupts	SPICSR	↑
FFE4h-FFE5h	11	LITE TIMER	LITE TIMER RTC Interrupt	LTCSR	
FFE6h-FFE7h	10	LITE TIMER	LITE TIMER Input Capture Interrupt	LTCSR	
FFE8h-FFE9h	9	AT TIMER	AT TIMER Overflow Interrupt	ATCSR	
FFEAh-FFEBh	8	AT TIMER	AT TIMER Output Compare Interrupt	PWM0CSR	
FFECh-FFEDh	7	SI	AVD Interrupt	SICSR	
FFEEh-FFEFh	6	-	Not Used	-	
FFF0h-FFF1h	5	-	Not Used	-	
FFF2h-FFF3h	4	ei3	External Interrupt 3	N/A	
FFF4h-FFF5h	3	ei2	External Interrupt 2	N/A	
FFF6h-FFF7h	2	ei1	External Interrupt 1	N/A	
FFF8h-FFF9h	1	ei0	External Interrupt 0	N/A	
FFFAh-FFFBh	0	-	Not Used	-	
FFFCh-FFFDh	-	TRAP	Software Interrupt	N/A	
FFFEh-FFFFh	-	RESET	Reset	N/A	Highest

Figure 4. Option Bytes



OPTION BYTE 1	15 PLL x4x8	0 = PLL x 4 1 = PLL x 8
	PLL OFF	0 = PLL ENABLED 1 = PLL DISABLED
		RESERVED
	OSC	0 = RC OSCILLATOR ON 1 = RC OSCILLATOR OFF
	LVD1 LVD0	00 = LOWEST VOLTAGE THRESHOLD (~2.8 V) 01 = MEDIUM VOLTAGE THRESHOLD (~3.5 V) 10 = HIGHEST VOLTAGE THRESHOLD (~4.1 V) 11 = LVD OFF
	WDG SW	0 = HARDWARE WATCHDOG 1 = SOFTWARE WATCHDOG
	8 WDG HALT	0 = NO RESET GENERATION IN HALT 1 = RESET GENERATION IN HALT

OPTION BYTE 0	7 1
	1
	1
	1
		RESERVED, MUST BE SET TO 1

	SEC1 SEC0	00 = SECTOR 0 SIZE = 0.5K BYTES 01 = SECTOR 0 SIZE = 1K BYTES 10 = SECTOR 0 SIZE = 2K BYTES 11 = SECTOR 0 SIZE = 4K BYTES
	FMP R	0 = READ-OUT PROTECTION OFF 1 = READ-OUT PROTECTION ON
	0 FMP W	0 = FLASH WRITE PROTECTION OFF 1 = FLASH WRITE PROTECTION ON

Figure 5. CPU Registers

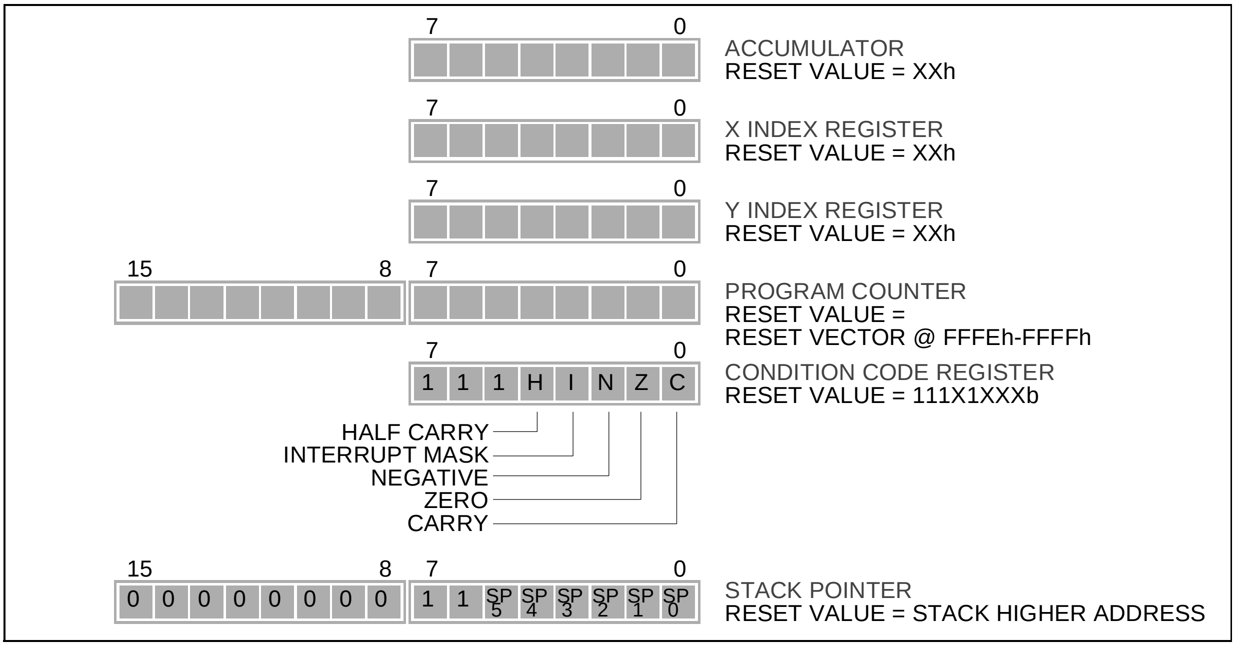


Figure 6. Typical In-Circuit Programming Interface

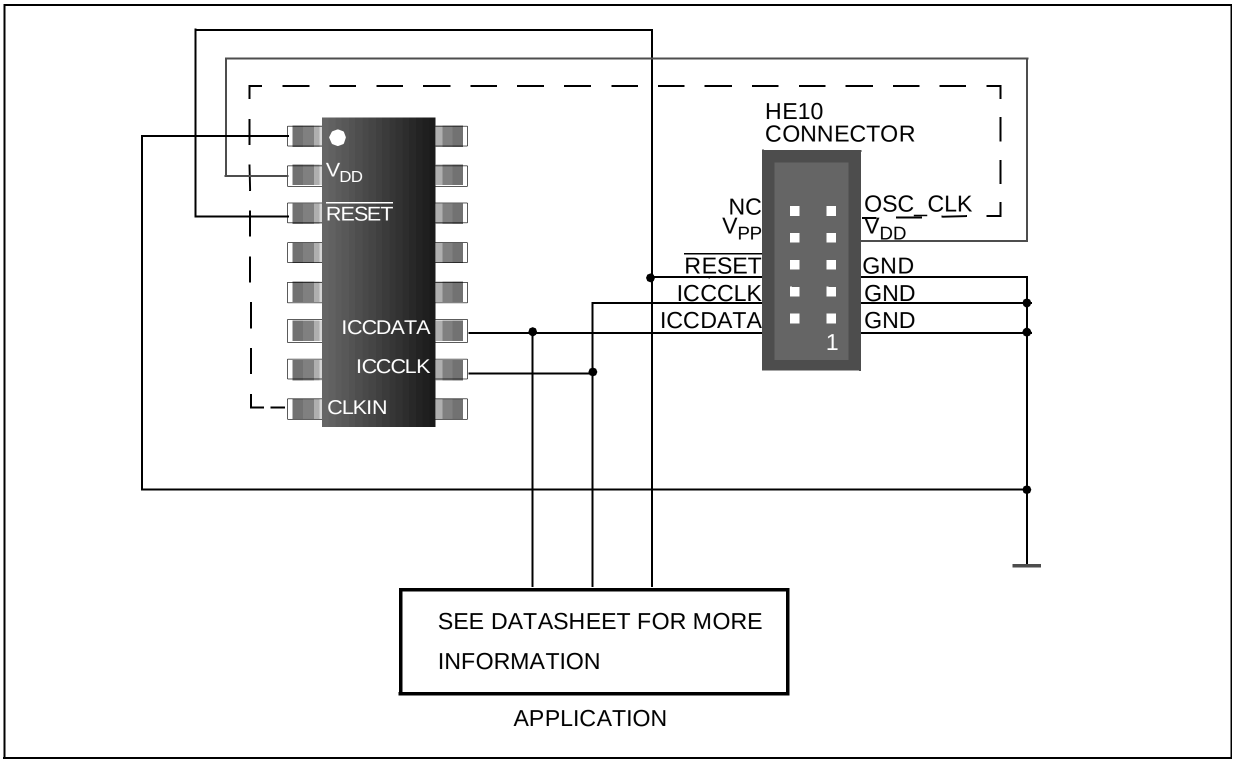


Table 6. Port Implementation

Standard Ports (PA6:1, PB4, PB2:0)			Interrupt Ports (PA7, PA0, PB3, PB0, with Pull-Ups)		
Mode	**DDR**	**OR**	**Mode**	**DDR**	**OR**
Floating Input	0	0	Floating Input	0	0
Pull-Up Input	0	1	Pull-Up Interrupt Input	0	1
Open Drain Output	1	0	Open Drain Output	1	0
Push-Pull Output	1	1	Push-Pull Output	1	1

Figure 7. Port A

7							0
D7	D6	D5	D4	D3	D2	D1	D0

PORT A DATA REGISTER
PADR (@ 0000h, Read/Write)
RESET VALUE = 00h

7							0
DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0

PORT A DATA DIRECTION REGISTER
PADDR (@ 0001h, Read/Write)
RESET VALUE = 00h

7							0
O7	O6	O5	O4	O3	O2	O1	O0

PORT A OPTION REGISTER
PADR (@ 0002h, Read/Write)
RESET VALUE = 40h

Figure 8. Port B

7							0
0	0	0	D4	D3	D2	D1	D0

PORT B DATA REGISTER
PBDR (@ 0003h, Read/Write)
RESET VALUE = 00h

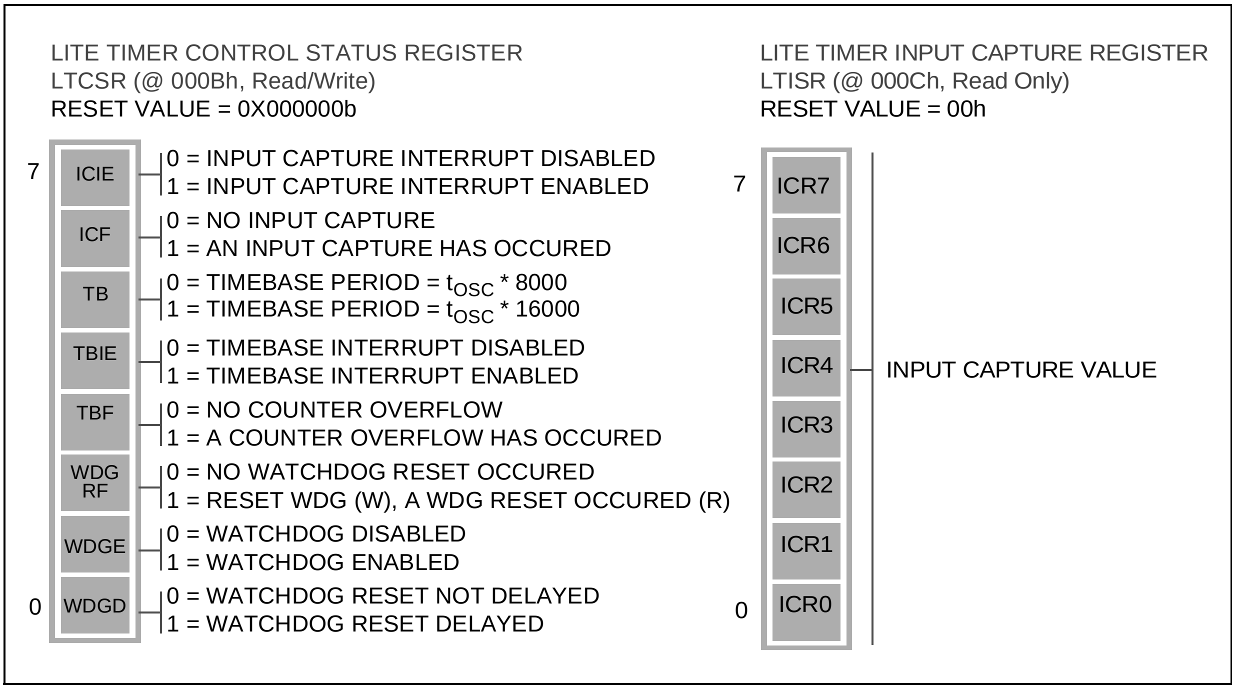
7							0
0	0	0	DD4	DD3	DD2	DD1	DD0

PORT B DATA DIRECTION REGISTER
PBDDR (@ 0004h, Read/Write)
RESET VALUE = 00h

7							0
0	0	0	O4	O3	O2	O1	O0

PORT B OPTION REGISTER
PBDR (@ 0005h, Read/Write)
RESET VALUE = 00h

Figure 9. Lite Timer



LITE TIMER CONTROL STATUS REGISTER
LTCSR (@ 000Bh, Read/Write)
RESET VALUE = 0X000000b

0 = INPUT CAPTURE INTERRUPT DISABLED
1 = INPUT CAPTURE INTERRUPT ENABLED

0 = NO INPUT CAPTURE
1 = AN INPUT CAPTURE HAS OCCURED

0 = TIMEBASE PERIOD = t_{OSC} * 8000
1 = TIMEBASE PERIOD = t_{OSC} * 16000

0 = TIMEBASE INTERRUPT DISABLED
1 = TIMEBASE INTERRUPT ENABLED

0 = NO COUNTER OVERFLOW
1 = A COUNTER OVERFLOW HAS OCCURED

0 = NO WATCHDOG RESET OCCURED
1 = RESET WDG (W), A WDG RESET OCCURED (R)

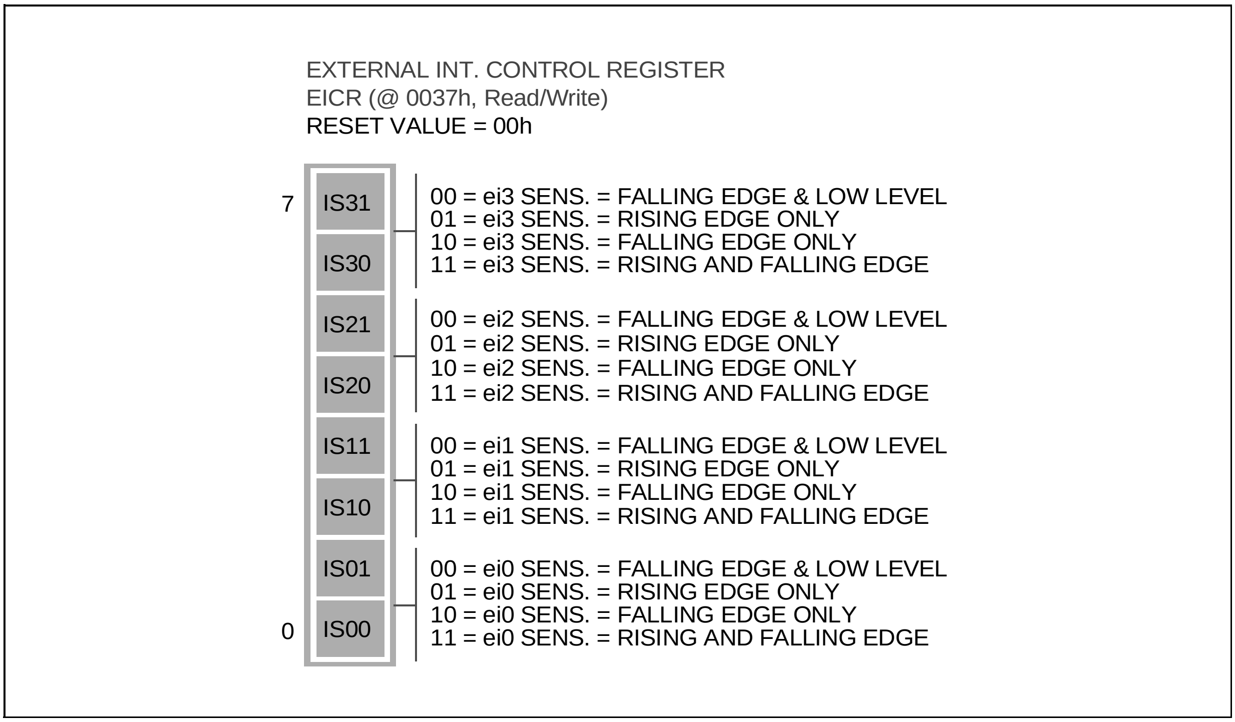
0 = WATCHDOG DISABLED
1 = WATCHDOG ENABLED

0 = WATCHDOG RESET NOT DELAYED
1 = WATCHDOG RESET DELAYED

LITE TIMER INPUT CAPTURE REGISTER
LTISR (@ 000Ch, Read Only)
RESET VALUE = 00h

INPUT CAPTURE VALUE

Figure 10. ITC



EXTERNAL INT. CONTROL REGISTER
EICR (@ 0037h, Read/Write)
RESET VALUE = 00h

00 = ei3 SENS. = FALLING EDGE & LOW LEVEL
01 = ei3 SENS. = RISING EDGE ONLY
10 = ei3 SENS. = FALLING EDGE ONLY
11 = ei3 SENS. = RISING AND FALLING EDGE

00 = ei2 SENS. = FALLING EDGE & LOW LEVEL
01 = ei2 SENS. = RISING EDGE ONLY
10 = ei2 SENS. = FALLING EDGE ONLY
11 = ei2 SENS. = RISING AND FALLING EDGE

00 = ei1 SENS. = FALLING EDGE & LOW LEVEL
01 = ei1 SENS. = RISING EDGE ONLY
10 = ei1 SENS. = FALLING EDGE ONLY
11 = ei1 SENS. = RISING AND FALLING EDGE

00 = ei0 SENS. = FALLING EDGE & LOW LEVEL
01 = ei0 SENS. = RISING EDGE ONLY
10 = ei0 SENS. = FALLING EDGE ONLY
11 = ei0 SENS. = RISING AND FALLING EDGE

Figure 11. Auto Reload Timer

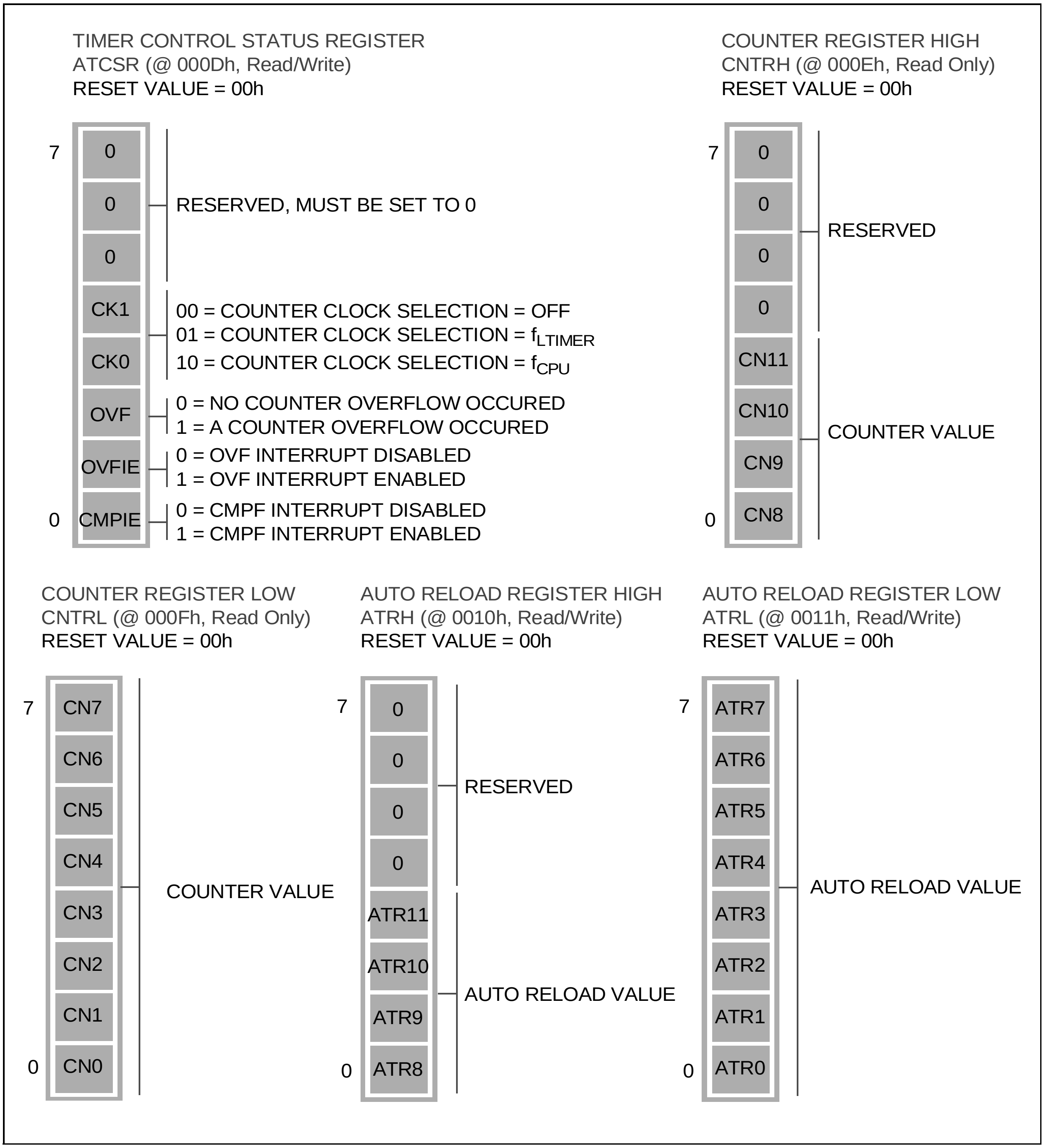
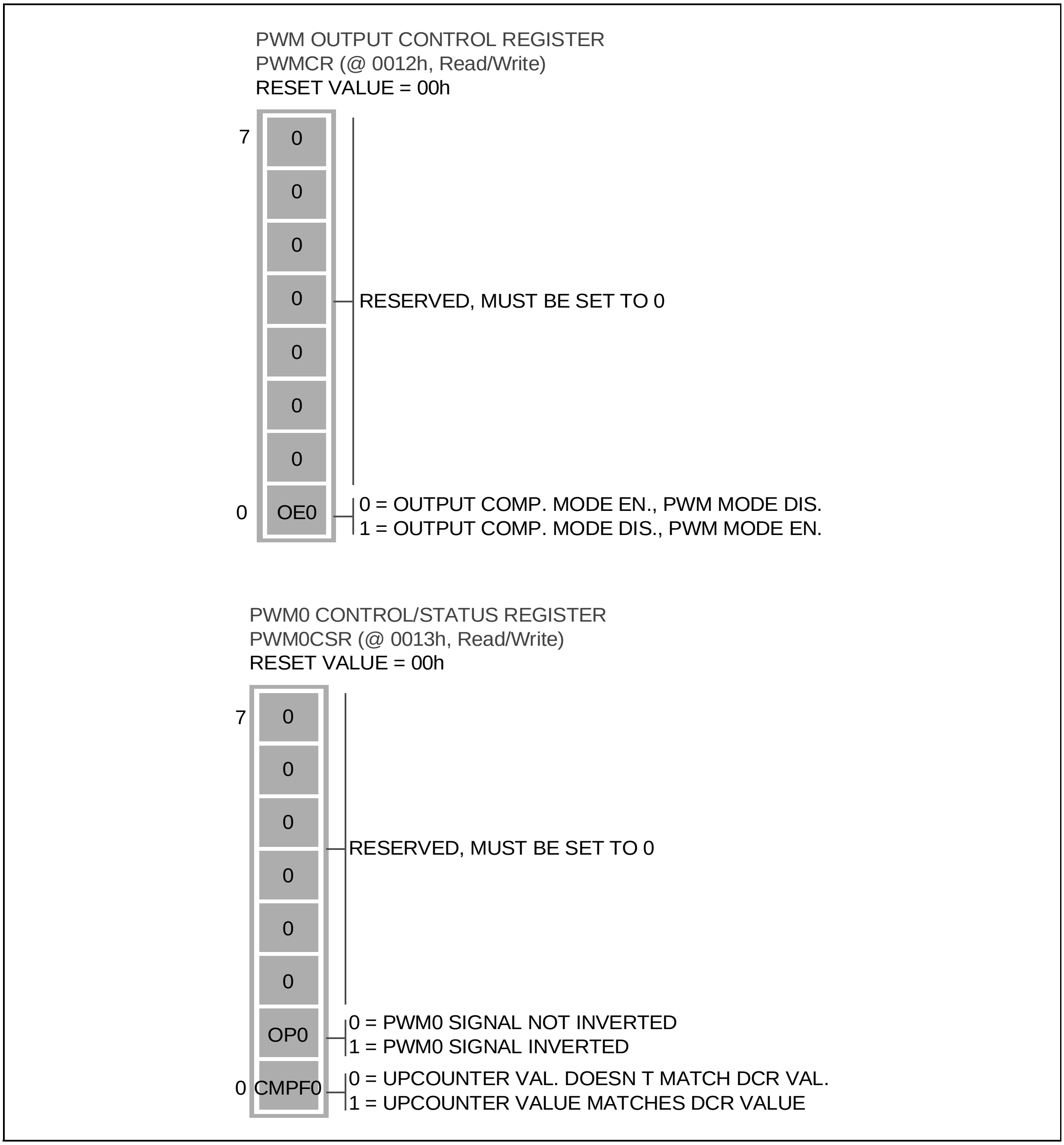

TIMER CONTROL STATUS REGISTER
ATCSR (@ 000Dh, Read/Write)
RESET VALUE = 00h
7
0
0
0
CK1
CK0
OVF
OVFIE
CMPIE
0
RESERVED, MUST BE SET TO 0
00 = COUNTER CLOCK SELECTION = OFF
01 = COUNTER CLOCK SELECTION = f_{LTIMER}
10 = COUNTER CLOCK SELECTION = f_{CPU}
0 = NO COUNTER OVERFLOW OCCURED
1 = A COUNTER OVERFLOW OCCURED
0 = OVF INTERRUPT DISABLED
1 = OVF INTERRUPT ENABLED
0 = CMPF INTERRUPT DISABLED
1 = CMPF INTERRUPT ENABLED
COUNTER REGISTER HIGH
CNTRH (@ 000Eh, Read Only)
RESET VALUE = 00h
7
0
0
0
0
CN11
CN10
CN9
CN8
0
RESERVED
COUNTER VALUE
COUNTER REGISTER LOW
CNTRL (@ 000Fh, Read Only)
RESET VALUE = 00h
7
CN7
CN6
CN5
CN4
CN3
CN2
CN1
CN0
0
COUNTER VALUE
AUTO RELOAD REGISTER HIGH
ATRH (@ 0010h, Read/Write)
RESET VALUE = 00h
7
0
0
0
0
ATR11
ATR10
ATR9
ATR8
0
RESERVED
AUTO RELOAD VALUE
AUTO RELOAD REGISTER LOW
ATRL (@ 0011h, Read/Write)
RESET VALUE = 00h
7
ATR7
ATR6
ATR5
ATR4
ATR3
ATR2
ATR1
ATR0
0
AUTO RELOAD VALUE

Figure 12. Auto Reload Timer (cont’d)



PWM OUTPUT CONTROL REGISTER
PWMCR (@ 0012h, Read/Write)
RESET VALUE = 00h

RESERVED, MUST BE SET TO 0

0 = OUTPUT COMP. MODE EN., PWM MODE DIS.
1 = OUTPUT COMP. MODE DIS., PWM MODE EN.

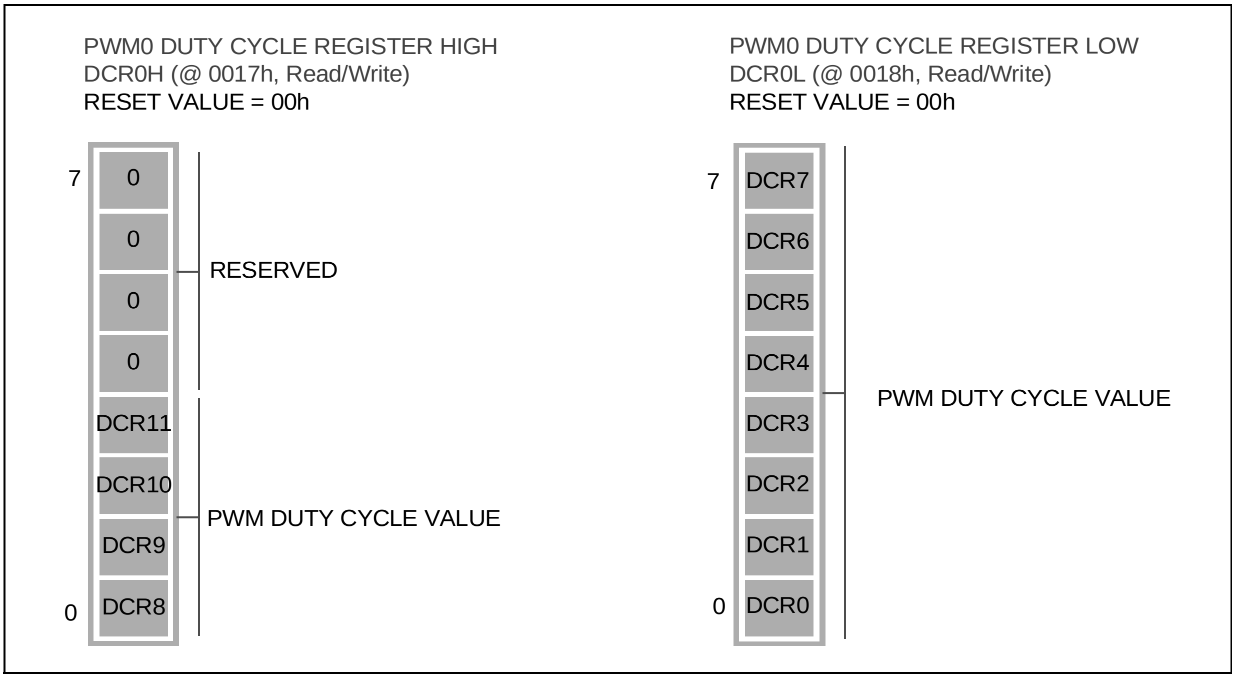
PWM0 CONTROL/STATUS REGISTER
PWM0CSR (@ 0013h, Read/Write)
RESET VALUE = 00h

RESERVED, MUST BE SET TO 0

0 = PWM0 SIGNAL NOT INVERTED
1 = PWM0 SIGNAL INVERTED

0 = UPCOUNTER VAL. DOESN T MATCH DCR VAL.
1 = UPCOUNTER VALUE MATCHES DCR VALUE

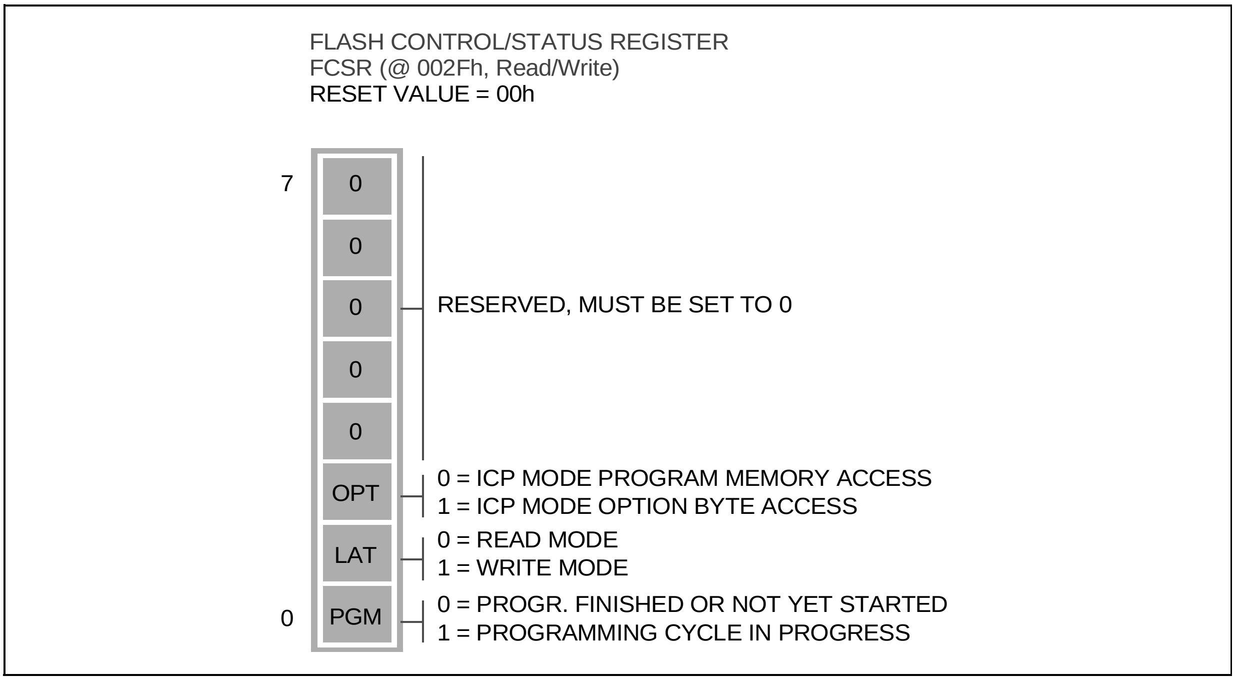
Figure 13. Auto Reload Timer (cont'd)



PWM0 DUTY CYCLE REGISTER HIGH
DCR0H (@ 0017h, Read/Write)
RESET VALUE = 00h

PWM0 DUTY CYCLE REGISTER LOW
DCR0L (@ 0018h, Read/Write)
RESET VALUE = 00h

Figure 14. FLASH



FLASH CONTROL/STATUS REGISTER
FCSR (@ 002Fh, Read/Write)
RESET VALUE = 00h

OPT: 0 = ICP MODE PROGRAM MEMORY ACCESS
1 = ICP MODE OPTION BYTE ACCESS

LAT: 0 = READ MODE
1 = WRITE MODE

PGM: 0 = PROGR. FINISHED OR NOT YET STARTED
1 = PROGRAMMING CYCLE IN PROGRESS

Figure 15. EEPROM

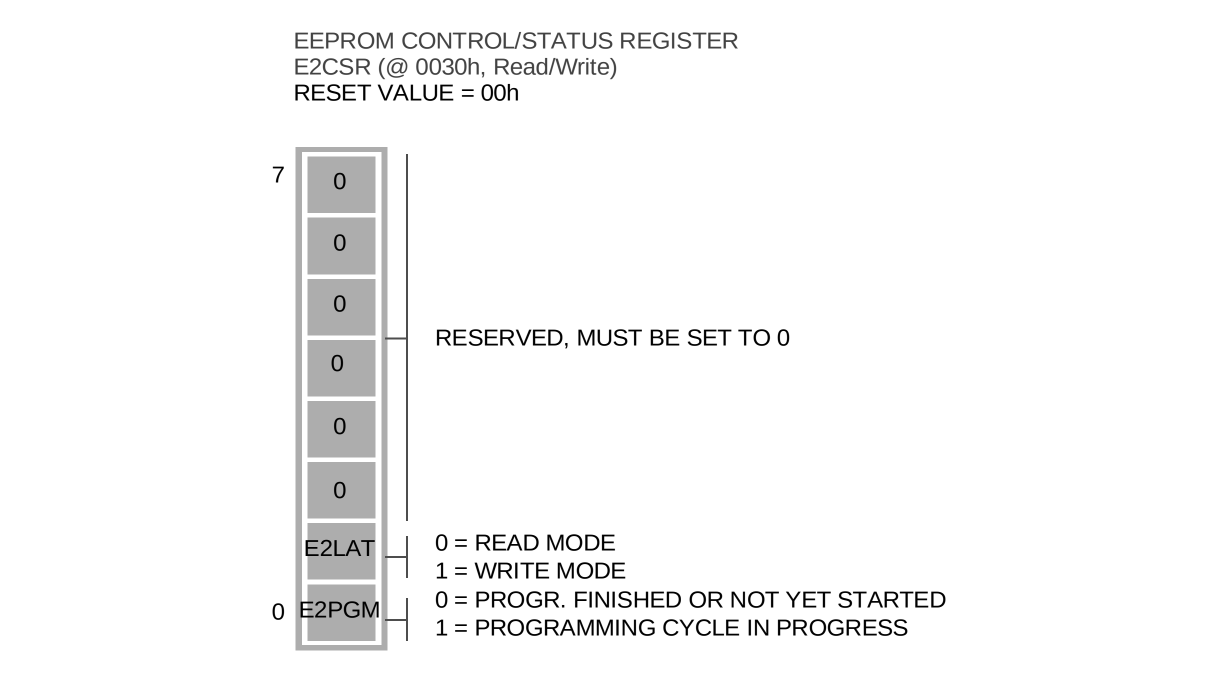
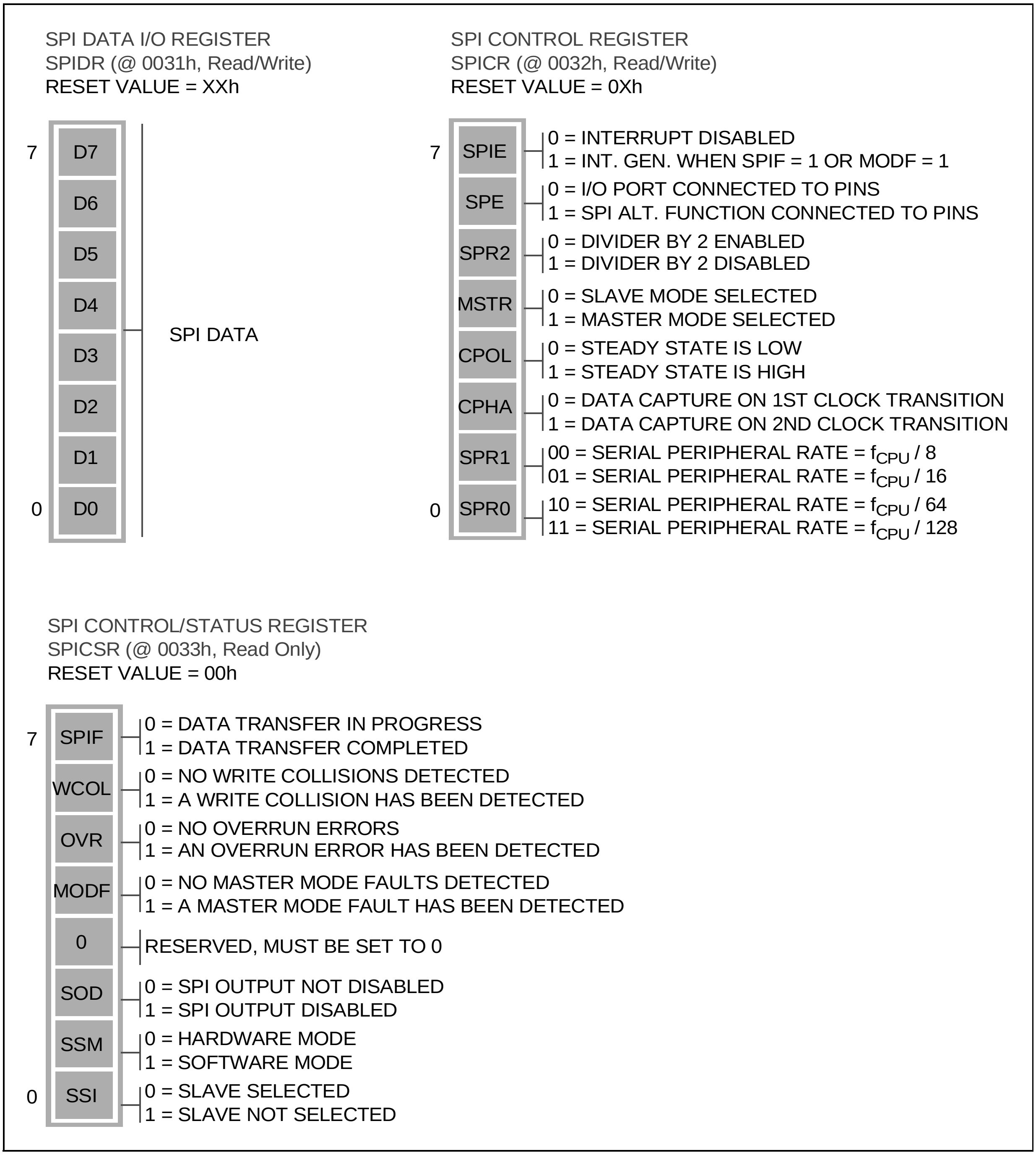

EEPROM CONTROL/STATUS REGISTER
E2CSR (@ 0030h, Read/Write)
RESET VALUE = 00h
7
0
0
0
0
0
0
RESERVED, MUST BE SET TO 0
E2LAT
0 = READ MODE
1 = WRITE MODE
0
E2PGM
0 = PROGR. FINISHED OR NOT YET STARTED
1 = PROGRAMMING CYCLE IN PROGRESS

Figure 16. SPI



SPI DATA I/O REGISTER
SPIDR (@ 0031h, Read/Write)
RESET VALUE = XXh

Bit	Name	Description
7	D7	SPI DATA
	D6	
	D5	
	D4	
	D3	
	D2	
	D1	
0	D0	

SPI CONTROL REGISTER
SPICR (@ 0032h, Read/Write)
RESET VALUE = 0Xh

Bit	Name	Description
7	SPIE	0 = INTERRUPT DISABLED 1 = INT. GEN. WHEN SPIF = 1 OR MODF = 1
	SPE	0 = I/O PORT CONNECTED TO PINS 1 = SPI ALT. FUNCTION CONNECTED TO PINS
	SPR2	0 = DIVIDER BY 2 ENABLED 1 = DIVIDER BY 2 DISABLED
	MSTR	0 = SLAVE MODE SELECTED 1 = MASTER MODE SELECTED
	CPOL	0 = STEADY STATE IS LOW 1 = STEADY STATE IS HIGH
	CPHA	0 = DATA CAPTURE ON 1ST CLOCK TRANSITION 1 = DATA CAPTURE ON 2ND CLOCK TRANSITION
	SPR1	00 = SERIAL PERIPHERAL RATE = f_{CPU} / 8 01 = SERIAL PERIPHERAL RATE = f_{CPU} / 16
0	SPR0	10 = SERIAL PERIPHERAL RATE = f_{CPU} / 64 11 = SERIAL PERIPHERAL RATE = f_{CPU} / 128

SPI CONTROL/STATUS REGISTER
SPICSR (@ 0033h, Read Only)
RESET VALUE = 00h

Bit	Name	Description
7	SPIF	0 = DATA TRANSFER IN PROGRESS 1 = DATA TRANSFER COMPLETED
	WCOL	0 = NO WRITE COLLISIONS DETECTED 1 = A WRITE COLLISION HAS BEEN DETECTED
	OVR	0 = NO OVERRUN ERRORS 1 = AN OVERRUN ERROR HAS BEEN DETECTED
	MODF	0 = NO MASTER MODE FAULTS DETECTED 1 = A MASTER MODE FAULT HAS BEEN DETECTED
	0	RESERVED, MUST BE SET TO 0
	SOD	0 = SPI OUTPUT NOT DISABLED 1 = SPI OUTPUT DISABLED
	SSM	0 = HARDWARE MODE 1 = SOFTWARE MODE
0	SSI	0 = SLAVE SELECTED 1 = SLAVE NOT SELECTED

Figure 17. MCC

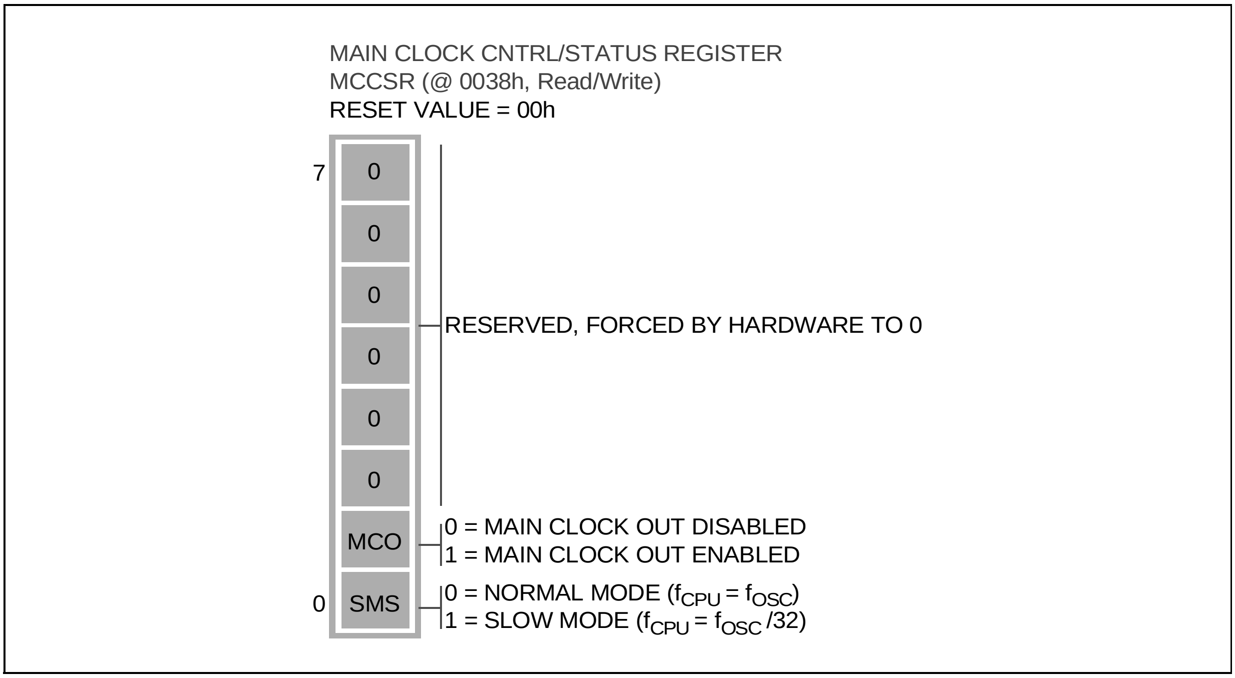

MAIN CLOCK CNTRL/STATUS REGISTER
MCCSR (@ 0038h, Read/Write)
RESET VALUE = 00h
7
0
0
0
0
0
0
MCO
0
SMS
RESERVED, FORCED BY HARDWARE TO 0
0 = MAIN CLOCK OUT DISABLED
1 = MAIN CLOCK OUT ENABLED
0 = NORMAL MODE ($f_{CPU} = f_{OSC}$)
1 = SLOW MODE ($f_{CPU} = f_{OSC}/32$)

Figure 18. Clock and Reset

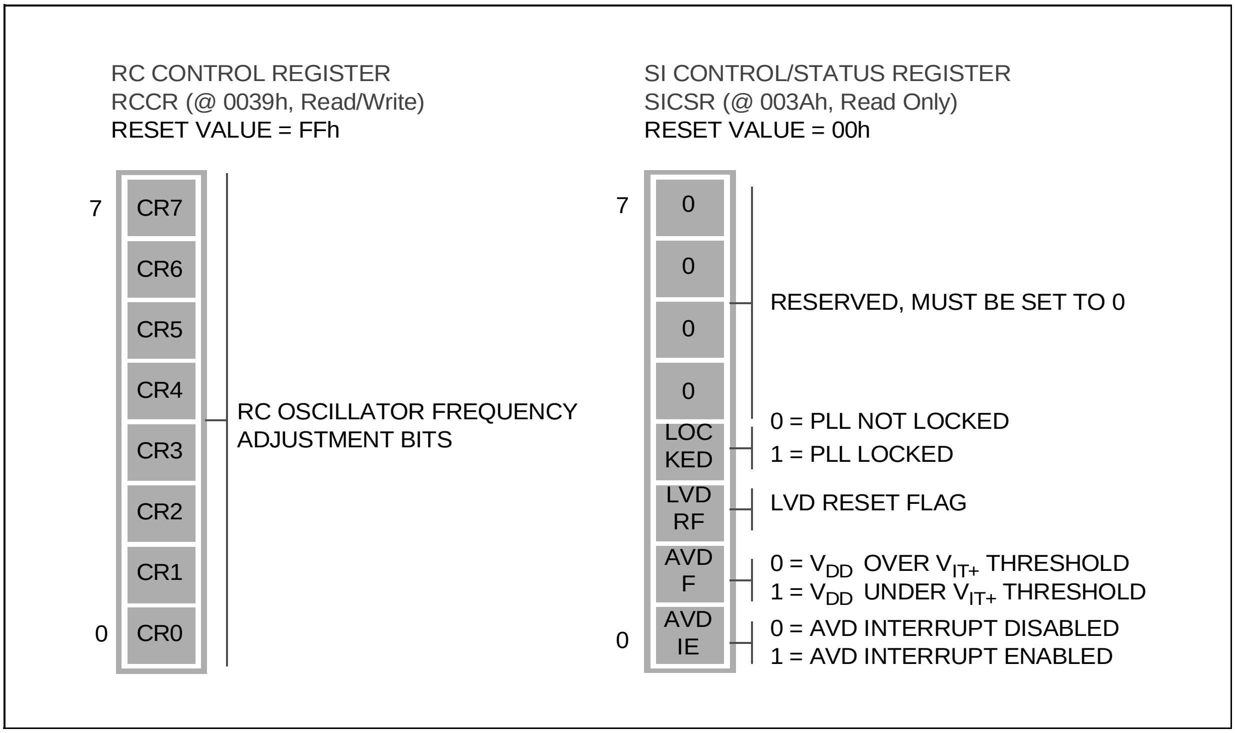
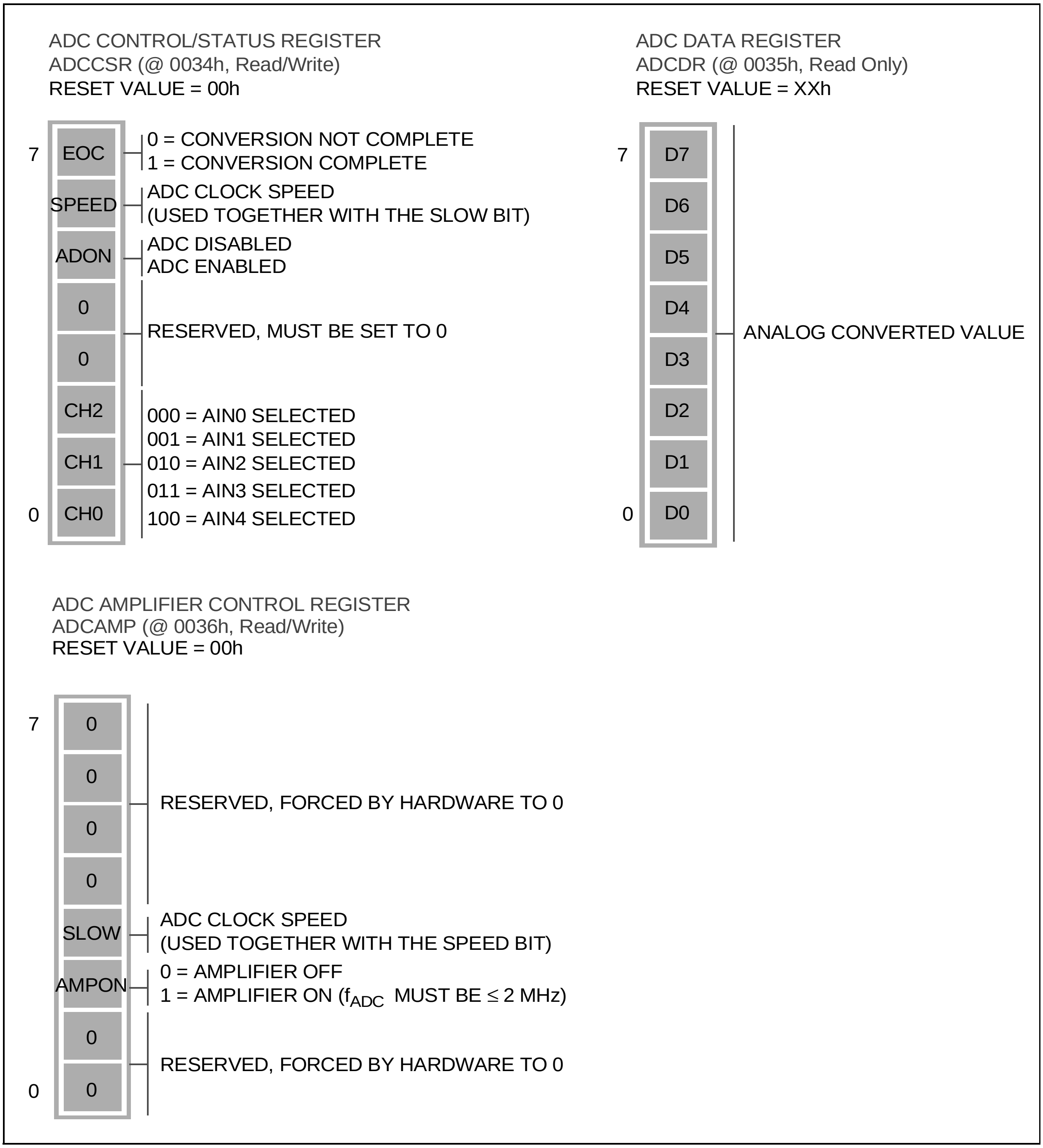

RC CONTROL REGISTER
RCCR (@ 0039h, Read/Write)
RESET VALUE = FFh
7
CR7
CR6
CR5
CR4
CR3
CR2
CR1
0
CR0
RC OSCILLATOR FREQUENCY ADJUSTMENT BITS
SI CONTROL/STATUS REGISTER
SICSR (@ 003Ah, Read Only)
RESET VALUE = 00h
7
0
0
0
0
LOC KED
LVD RF
AVD F
0
AVD IE
RESERVED, MUST BE SET TO 0
0 = PLL NOT LOCKED
1 = PLL LOCKED
LVD RESET FLAG
0 = V_{DD} OVER V_{IT+} THRESHOLD
1 = V_{DD} UNDER V_{IT+} THRESHOLD
0 = AVD INTERRUPT DISABLED
1 = AVD INTERRUPT ENABLED

Figure 19. ADC


ADC CONTROL/STATUS REGISTER
ADCCSR (@ 0034h, Read/Write)
RESET VALUE = 00h
7
EOC
SPEED
ADON
0
0
CH2
CH1
CH0
0
0 = CONVERSION NOT COMPLETE
1 = CONVERSION COMPLETE
ADC CLOCK SPEED
(USED TOGETHER WITH THE SLOW BIT)
ADC DISABLED
ADC ENABLED
RESERVED, MUST BE SET TO 0
000 = AIN0 SELECTED
001 = AIN1 SELECTED
010 = AIN2 SELECTED
011 = AIN3 SELECTED
100 = AIN4 SELECTED
ADC DATA REGISTER
ADCDR (@ 0035h, Read Only)
RESET VALUE = XXh
7
D7
D6
D5
D4
D3
D2
D1
D0
0
ANALOG CONVERTED VALUE
ADC AMPLIFIER CONTROL REGISTER
ADCAMP (@ 0036h, Read/Write)
RESET VALUE = 00h
7
0
0
0
0
SLOW
AMPON
0
0
0
RESERVED, FORCED BY HARDWARE TO 0
ADC CLOCK SPEED
(USED TOGETHER WITH THE SPEED BIT)
0 = AMPLIFIER OFF
1 = AMPLIFIER ON (f_{ADC} MUST BE ≤ 2 MHz)
RESERVED, FORCED BY HARDWARE TO 0

“THE PRESENT NOTE WHICH IS FOR GUIDANCE ONLY AIMS AT PROVIDING CUSTOMERS WITH INFORMATION REGARDING THEIR PRODUCTS IN ORDER FOR THEM TO SAVE TIME. AS A RESULT, STMICROELECTRONICS SHALL NOT BE HELD LIABLE FOR ANY DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS ARISING FROM THE CONTENT OF SUCH A NOTE AND/OR THE USE MADE BY CUSTOMERS OF THE INFORMATION CONTAINED HEREIN IN CONNEXION WITH THEIR PRODUCTS.”

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without the express written approval of STMicroelectronics.

STMicroelectronics Group of Companies
Australia - Brazil - Canada - China - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan
Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - U.S.A.

http://www.st.com