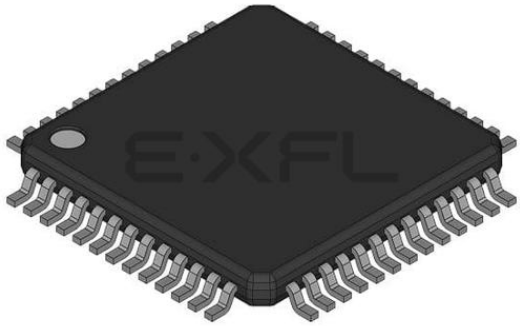

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	HCS12
Core Size	16-Bit
Speed	25MHz
Connectivity	EBI/EMI, SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	35
Program Memory Size	96KB (96K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.35V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	52-LQFP
Supplier Device Package	52-QFP (10x10)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=mc9s12gc96mpbe

Chapter 10
Freescale's Scalable Controller Area Network (S12MSCANV2)

Chapter 11
Oscillator (OSCV2) Block Description

Chapter 12
Pulse-Width Modulator (PWM8B6CV1) Block Description

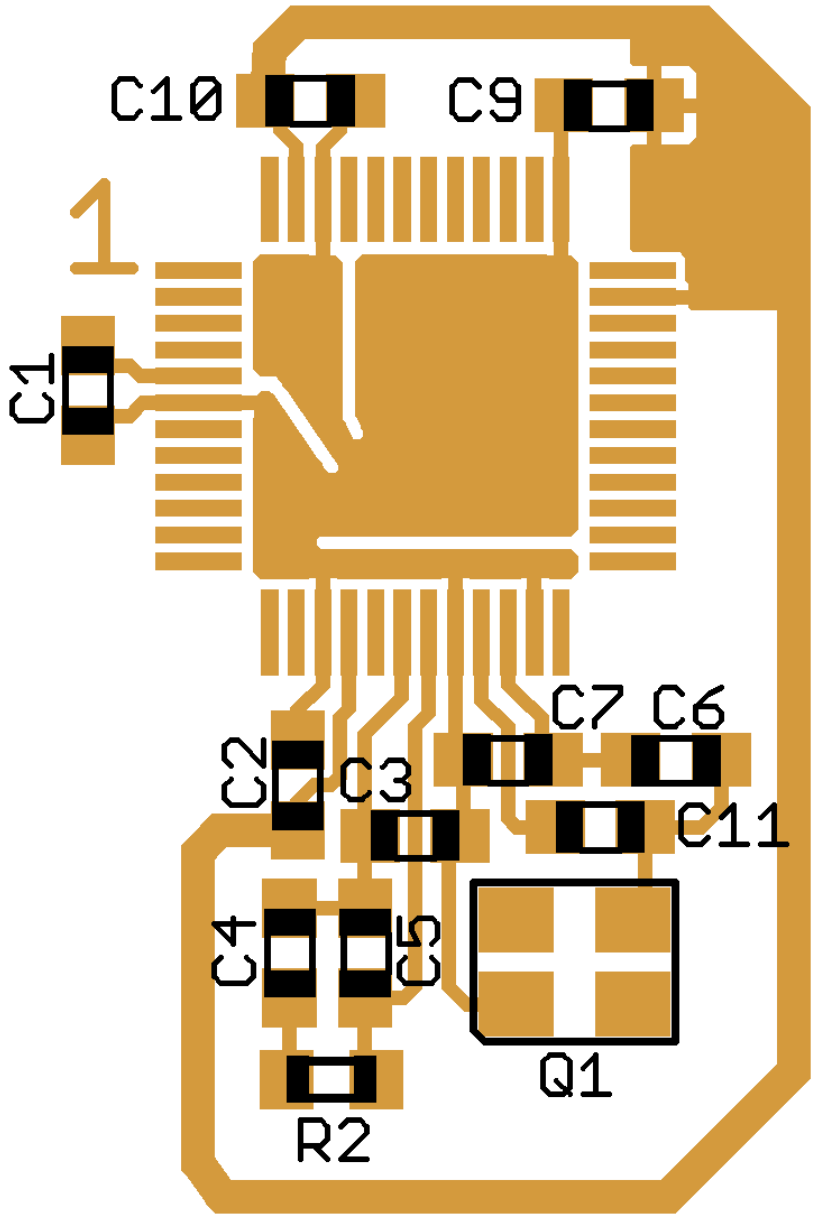

Figure 1-15. Recommended PCB Layout (48 LQFP) Colpitts Oscillator

when mapping PWM channels to Port T in an 80QFP option, the associated PWM channels are then mapped to both Port P and Port T.

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x001A	DDRP	R	DDRP7	DDRP6	DDRP5	DDRP4	DDRP3	DDRP2	DDRP1	DDRP0
		W								
0x001B	RDRP	R	RDRP7	RDRP6	RDRP5	RDRP4	RDRP3	RDRP2	RDRP1	RDRP0
		W								
0x001C	PERP	R	PERP7	PERP6	PERP5	PERP4	PERP3	PERP2	PERP1	PERP0
		W								
0x001D	PPSP	R	PPSP7	PPSP6	PPSP5	PPSP4	PPSP3	PPSP2	PPSP1	PPSP0
		W								
0x001E	PIEP	R	PIEP7	PIEP6	PIEP5	PIEP4	PIEP3	PIEP2	PIEP1	PIEP0
		W								
0x001F	PIFP	R	PIFP7	PIFP6	PIFP5	PIFP4	PIFP3	PIFP2	PIFP1	PIFP0
		W								
0x0020–0x0027	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0028	PTJ	R	PTJ7	PTJ6	0	0	0	0	0	0
		W								
0x0029	PTIJ	R	PTIJ7	PTIJ6	0	0	0	0	0	0
		W								
0x002A	DDRJ	R	DDRJ7	DDRJ6	0	0	0	0	0	0
		W								
0x002B	RDRJ	R	RDRJ7	RDRJ6	0	0	0	0	0	0
		W								
0x002C	PERJ	R	PERJ7	PERJ6	0	0	0	0	0	0
		W								
0x002D	PPSJ	R	PPSJ7	PPSJ6	0	0	0	0	0	0
		W								
0x002E	PIEJ	R	PIEJ7	PIEJ6	0	0	0	0	0	0
		W								
0x002F	PIFJ	R	PIFJ7	PIFJ6	0	0	0	0	0	0
		W								
0x0030	PTAD	R	PTAD7	PTAD6	PTAD5	PTAD4	PTAD3	PTAD2	PTAD1	PTAD0
		W								
0x0031	PTIAD	R	PTIAD7	PTIAD6	PTIAD5	PTIAD4	PTIAD3	PTIAD2	PTIAD1	PTIAD0
		W								
0x0032	DDRAD	R	DDRAD7	DDRAD6	DDRAD5	DDRAD4	DDRAD3	DDRAD2	DDRAD1	DDRAD0
		W								
0x0033	RDRAD	R	RDRAD7	RDRAD6	RDRAD5	RDRAD4	RDRAD3	RDRAD2	RDRAD1	RDRAD0
		W								
0x0034	PERAD	R	PERAD7	PERAD6	PERAD5	PERAD4	PERAD3	PERAD2	PERAD1	PERAD0
		W								
0x0035	PPSAD	R	PPSAD7	PPSAD6	PPSAD5	PPSAD4	PPSAD3	PPSAD2	PPSAD1	PPSAD0
		W								
0x0036–0x003F	Reserved	R	0	0	0	0	0	0	0	0
		W								

= Unimplemented or Reserved

Figure 2-2. Quick Reference to PIM Registers (Sheet 3 of 3)

There are two basic types of operating modes:

1. Normal modes: Some registers and bits are protected against accidental changes.
2. Special modes: Allow greater access to protected control registers and bits for special purposes such as testing.

A system development and debug feature, background debug mode (BDM), is available in all modes. In special single-chip mode, BDM is active immediately after reset.

Some aspects of Port E are not mode dependent. Bit 1 of Port E is a general purpose input or the $\overline{\text{IRQ}}$ interrupt input. $\overline{\text{IRQ}}$ can be enabled by bits in the CPU's condition codes register but it is inhibited at reset so this pin is initially configured as a simple input with a pull-up. Bit 0 of Port E is a general purpose input or the $\overline{\text{XIRQ}}$ interrupt input. $\overline{\text{XIRQ}}$ can be enabled by bits in the CPU's condition codes register but it is inhibited at reset so this pin is initially configured as a simple input with a pull-up. The ESTR bit in the EBICTL register is set to one by reset in any user mode. This assures that the reset vector can be fetched even if it is located in an external slow memory device. The PE6/MODB/IPIPE1 and PE5/MODA/IPIPE0 pins act as high-impedance mode select inputs during reset.

The following paragraphs discuss the default bus setup and describe which aspects of the bus can be changed after reset on a per mode basis.

4.4.3.1 Normal Operating Modes

These modes provide three operating configurations. Background debug is available in all three modes, but must first be enabled for some operations by means of a BDM background command, then activated.

4.4.3.1.1 Normal Single-Chip Mode

There is no external expansion bus in this mode. All pins of Ports A, B and E are configured as general purpose I/O pins Port E bits 1 and 0 are available as general purpose input only pins with internal pull resistors enabled. All other pins of Port E are bidirectional I/O pins that are initially configured as high-impedance inputs with internal pull resistors enabled. Ports A and B are configured as high-impedance inputs with their internal pull resistors disabled.

The pins associated with Port E bits 6, 5, 3, and 2 cannot be configured for their alternate functions IPIPE1, IPIPE0, $\overline{\text{LSTRB}}$, and R/$\overline{\text{W}}$ while the MCU is in single chip modes. In single chip modes, the associated control bits PIPOE, LSTRE, and RDWE are reset to zero. Writing the opposite state into them in single chip mode does not change the operation of the associated Port E pins.

In normal single chip mode, the MODE register is writable one time. This allows a user program to change the bus mode to narrow or wide expanded mode and/or turn on visibility of internal accesses.

Port E, bit 4 can be configured for a free-running E clock output by clearing NECLK=0. Typically the only use for an E clock output while the MCU is in single chip modes would be to get a constant speed clock for use in the external application system.

9.3.2.9 CRG COP Control Register (COPCTL)

This register controls the COP (computer operating properly) watchdog.

Module Base + 0x0008

	7	6	5	4	3	2	1	0
R	WCOP	RSBCK	0	0	0	CR2	CR1	CR0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 9-12. CRG COP Control Register (COPCTL)

Read: anytime

Write: WCOP, CR2, CR1, CR0: once in user mode, anytime in special mode

Write: RSBCK: once

Table 9-8. COPCTL Field Descriptions

Field	Description
7 WCOP	**Window COP Mode Bit** — When set, a write to the ARMCOP register must occur in the last 25% of the selected period. A write during the first 75% of the selected period will reset the part. As long as all writes occur during this window, 0x0055 can be written as often as desired. As soon as 0x00AA is written after the 0x0055, the time-out logic restarts and the user must wait until the next window before writing to ARMCOP. Table 9-9 shows the exact duration of this window for the seven available COP rates. 0 Normal COP operation 1 Window COP operation
6 RSBCK	COP and RTI Stop in Active BDM Mode Bit 0 Allows the COP and RTI to keep running in active BDM mode. 1 Stops the COP and RTI counters whenever the part is in active BDM mode.
2:0 CR[2:0]	**COP Watchdog Timer Rate Select** — These bits select the COP time-out rate (see Table 9-9). The COP time-out period is OSCCLK period divided by CR[2:0] value. Writing a nonzero value to CR[2:0] enables the COP counter and starts the time-out period. A COP counter time-out causes a system reset. This can be avoided by periodically (before time-out) reinitializing the COP counter via the ARMCOP register.

Table 9-9. COP Watchdog Rates[1]

CR2	CR1	CR0	OSCCLK Cycles to Time Out
0	0	0	COP disabled
0	0	1	2^{14}
0	1	0	2^{16}
0	1	1	2^{18}
1	0	0	2^{20}
1	0	1	2^{22}
1	1	0	2^{23}
1	1	1	2^{24}

1. OSCCLK cycles are referenced from the previous COP time-out reset (writing 0x0055/0x00AA to the ARMCOP register)

Module Base + 0x00X3

	7	6	5	4	3	2	1	0
R W	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR
Reset:	x	x	x	x	x	x	x	x

Figure 10-28. Identifier Register 3 (IDR3) — Extended Identifier Mapping

Table 10-27. IDR3 Register Field Descriptions — Extended

Field	Description
7:1 ID[6:0]	**Extended Format Identifier —** The identifiers consist of 29 bits (ID[28:0]) for the extended format. ID28 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number.
0 RTR	**Remote Transmission Request** — This flag reflects the status of the remote transmission request bit in the CAN frame. In the case of a receive buffer, it indicates the status of the received frame and supports the transmission of an answering frame in software. In the case of a transmit buffer, this flag defines the setting of the RTR bit to be sent. 0 Data frame 1 Remote frame

10.3.3.1.2 IDR0–IDR3 for Standard Identifier Mapping

Module Base + 0x00X0

	7	6	5	4	3	2	1	0
R W	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3
Reset:	x	x	x	x	x	x	x	x

Figure 10-29. Identifier Register 0 — Standard Mapping

Table 10-28. IDR0 Register Field Descriptions — Standard

Field	Description
7:0 ID[10:3]	**Standard Format Identifier —** The identifiers consist of 11 bits (ID[10:0]) for the standard format. ID10 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number. See also ID bits in Table 10-29.

The MSCAN is able to leave sleep mode (wake up) only when:

- CAN bus activity occurs and WUPE = 1

  or
- the CPU clears the SLPRQ bit

NOTE

The CPU cannot clear the SLPRQ bit before sleep mode (SLPRQ = 1 and SLPAK = 1) is active.

After wake-up, the MSCAN waits for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, if the MSCAN is woken-up by a CAN frame, this frame is not received.

The receive message buffers (RxFG and RxBG) contain messages if they were received before sleep mode was entered. All pending actions will be executed upon wake-up; copying of RxBG into RxFG, message aborts and message transmissions. If the MSCAN remains in bus-off state after sleep mode was exited, it continues counting the 128 occurrences of 11 consecutive recessive bits.

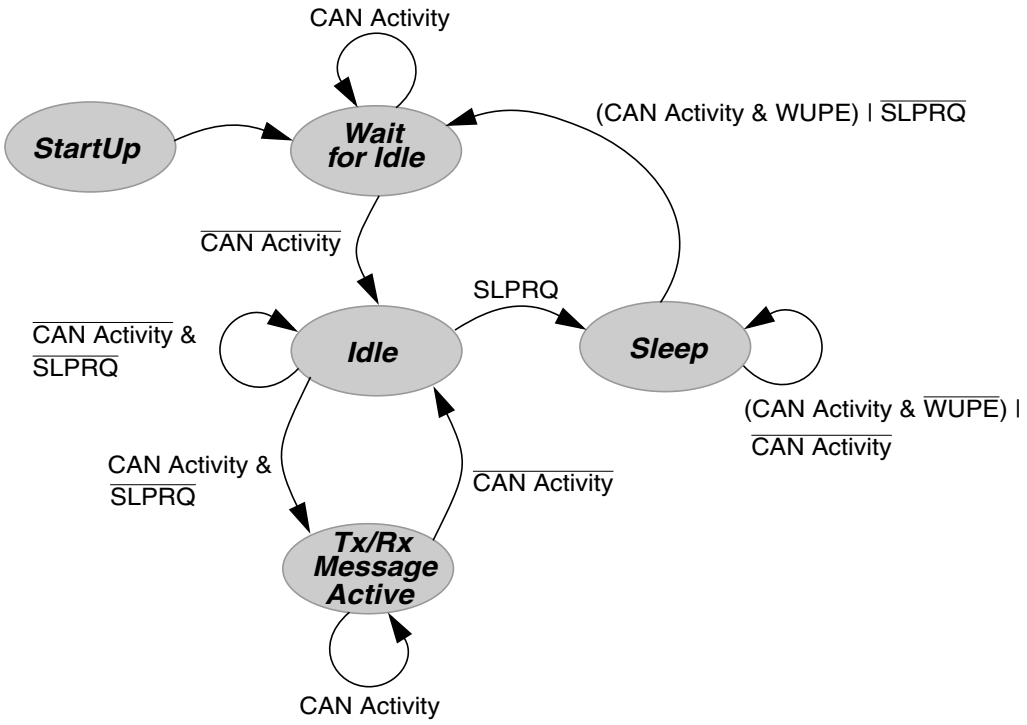

Figure 10-45. Simplified State Transitions for Entering/Leaving Sleep Mode

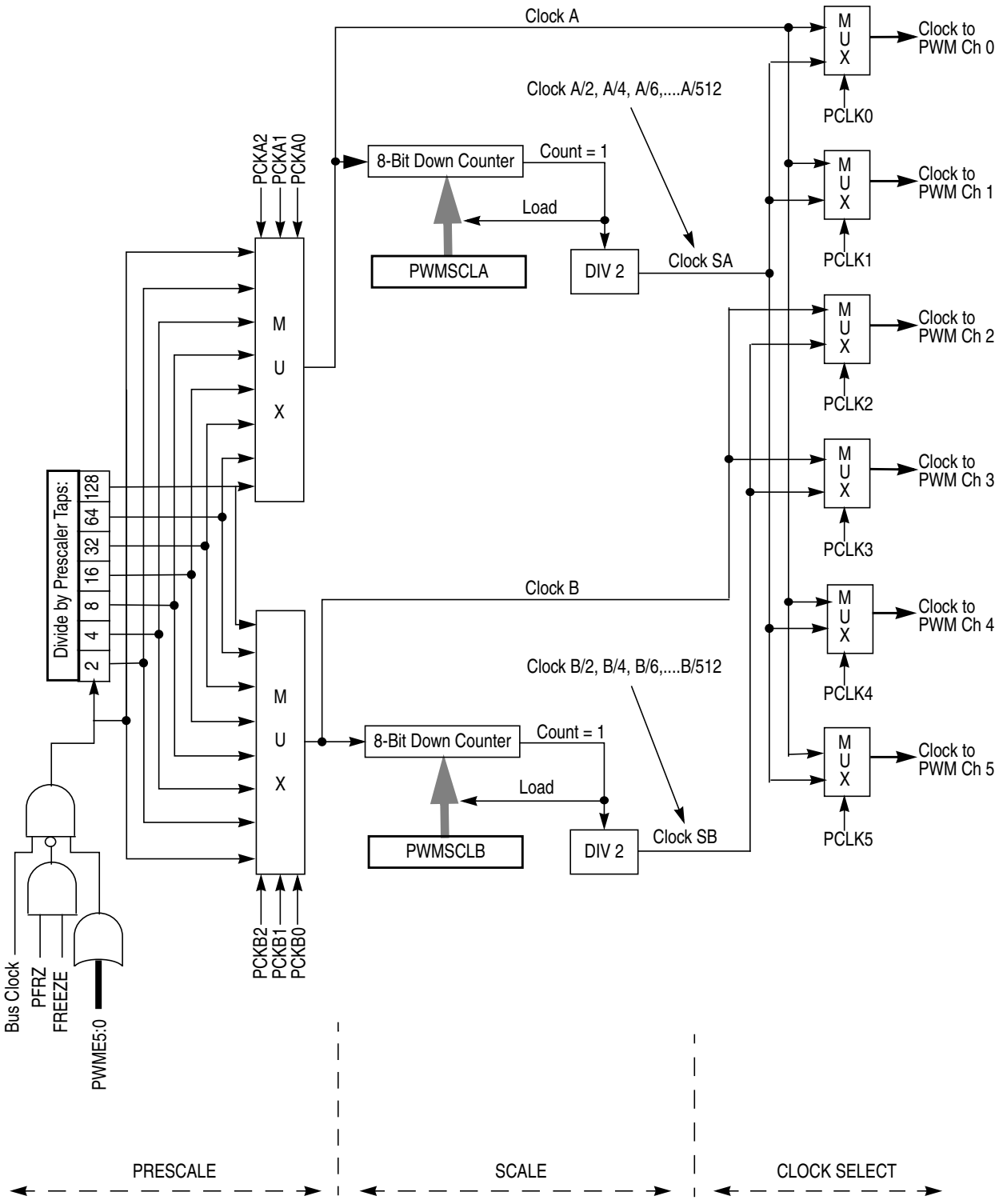

Figure 12-34. PWM Clock Select Block Diagram

Table 12-12 is used to summarize which channels are used to set the various control bits when in 16-bit mode.

Table 12-12. 16-bit Concatenation Mode Summary

CONxx	PWMEx	PPOLx	PCLKx	CAEx	PWMx Output
CON45	PWME5	PPOL5	PCLK5	CAE5	PWM5
CON23	PWME3	PPOL3	PCLK3	CAE3	PWM3
CON01	PWME1	PPOL1	PCLK1	CAE1	PWM1

12.4.2.8 PWM Boundary Cases

Table 12-13 summarizes the boundary conditions for the PWM regardless of the output mode (left aligned or center aligned) and 8-bit (normal) or 16-bit (concatenation):

Table 12-13. PWM Boundary Cases

PWMDTYx	PWMPERx	PPOLx	PWMx Output
0x0000 (indicates no duty)	>0x0000	1	Always Low
0x0000 (indicates no duty)	>0x0000	0	Always High
XX	0x0000(1) (indicates no period)	1	Always High
XX	0x0000[1] (indicates no period)	0	Always Low
>= PWMPERx	XX	1	Always High
>= PWMPERx	XX	0	Always Low

1. Counter = 0x0000 and does not count.

12.5 Resets

The reset state of each individual bit is listed within the register description section (see Section 12.3, “Memory Map and Register Definition,” which details the registers and their bit-fields. All special functions or modes which are initialized during or just following reset are described within this section.

- The 8-bit up/down counter is configured as an up counter out of reset.
- All the channels are disabled and all the counters don’t count.

12.6 Interrupts

The PWM8B6CV1 module has only one interrupt which is generated at the time of emergency shutdown, if the corresponding enable bit (PWMIE) is set. This bit is the enable for the interrupt. The interrupt flag PWMIF is set whenever the input level of the PWM5 channel changes while PWM5ENA=1 or when PWMENA is being asserted while the level at PWM5 is active.

A description of the registers involved and affected due to this interrupt is explained in Section 12.3.2.15, “PWM Shutdown Register (PWMSDN).”

writing another byte to the Transmitter buffer (SCIDRH/SCIDRL), while the shift register is still shifting out the first byte.

To initiate an SCI transmission:

1. Configure the SCI:
   a) Select a baud rate. Write this value to the SCI baud registers (SCIBDH/L) to begin the baud rate generator. Remember that the baud rate generator is disabled when the baud rate is zero. Writing to the SCIBDH has no effect without also writing to SCIBDL.
   b) Write to SCICR1 to configure word length, parity, and other configuration bits (LOOPS,RSRC,M,WAKE,ILT,PE,PT).
   c) Enable the transmitter, interrupts, receive, and wake up as required, by writing to the SCICR2 register bits (TIE,TCIE,RIE,ILIE,TE,RE,RWU,SBK). A preamble or idle character will now be shifted out of the transmitter shift register.
2. Transmit Procedure for Each Byte:
   a. Poll the TDRE flag by reading the SCISR1 or responding to the TDRE interrupt. Keep in mind that the TDRE bit resets to one.
   d) If the TDRE flag is set, write the data to be transmitted to SCIDRH/L, where the ninth bit is written to the T8 bit in SCIDRH if the SCI is in 9-bit data format. A new transmission will not result until the TDRE flag has been cleared.
3. Repeat step 2 for each subsequent transmission.

NOTE

The TDRE flag is set when the shift register is loaded with the next data to be transmitted from SCIDRH/L, which happens, generally speaking, a little over half-way through the stop bit of the previous frame. Specifically, this transfer occurs 9/16ths of a bit time AFTER the start of the stop bit of the previous frame.

Writing the TE bit from 0 to a 1 automatically loads the transmit shift register with a preamble of 10 logic 1s (if M = 0) or 11 logic 1s (if M = 1). After the preamble shifts out, control logic transfers the data from the SCI data register into the transmit shift register. A logic 0 start bit automatically goes into the least significant bit position of the transmit shift register. A logic 1 stop bit goes into the most significant bit position.

Hardware supports odd or even parity. When parity is enabled, the most significant bit (msb) of the data character is the parity bit.

The transmit data register empty flag, TDRE, in SCI status register 1 (SCISR1) becomes set when the SCI data register transfers a byte to the transmit shift register. The TDRE flag indicates that the SCI data register can accept new data from the internal data bus. If the transmit interrupt enable bit, TIE, in SCI control register 2 (SCICR2) is also set, the TDRE flag generates a transmitter interrupt request.

When the transmit shift register is not transmitting a frame, the **Tx output** signal goes to the idle condition, logic 1. If at any time software clears the TE bit in SCI control register 2 (SCICR2), the transmitter enable signal goes low and the transmit signal goes idle.

Table 15-5. OC7M Field Descriptions

Field	Description
7:0 OC7M[7:0]	**Output Compare 7 Mask** — Setting the OC7Mx (x ranges from 0 to 6) will set the corresponding port to be an output port when the corresponding TIOSx (x ranges from 0 to 6) bit is set to be an output compare. **Note:** A successful channel 7 output compare overrides any channel 6:0 compares. For each OC7M bit that is set, the output compare action reflects the corresponding OC7D bit.

15.3.2.4 Output Compare 7 Data Register (OC7D)

Module Base + 0x0003

	7	6	5	4	3	2	1	0
R W	OC7D7	OC7D6	OC7D5	OC7D4	OC7D3	OC7D2	OC7D1	OC7D0
Reset	0	0	0	0	0	0	0	0

Figure 15-9. Output Compare 7 Data Register (OC7D)

Read: Anytime

Write: Anytime

Table 15-6. OC7D Field Descriptions

Field	Description
7:0 OC7D[7:0]	**Output Compare 7 Data** — A channel 7 output compare can cause bits in the output compare 7 data register to transfer to the timer port data register depending on the output compare 7 mask register.

15.3.2.5 Timer Count Register (TCNT)

Module Base + 0x0004

	15	14	13	12	11	10	9	9
R W	TCNT15	TCNT14	TCNT13	TCNT12	TCNT11	TCNT10	TCNT9	TCNT8
Reset	0	0	0	0	0	0	0	0

Figure 15-10. Timer Count Register High (TCNTH)

Module Base + 0x0005

	7	6	5	4	3	2	1	0
R W	TCNT7	TCNT6	TCNT5	TCNT4	TCNT3	TCNT2	TCNT1	TCNT0
Reset	0	0	0	0	0	0	0	0

Figure 15-11. Timer Count Register Low (TCNTL)

16.2.3 V_{DD}, V_{SS} — Regulator Output1 (Core Logic)

Signals V_{DD}/V_{SS} are the primary outputs of VREG3V3V2 that provide the power supply for the core logic. These signals are connected to device pins to allow external decoupling capacitors (100 nF...220 nF, X7R ceramic).

In Shutdown Mode an external supply at V_{DD}/V_{SS} can replace the voltage regulator.

16.2.4 V_{DDPLL}, V_{SSPLL} — Regulator Output2 (PLL)

Signals V_{DDPLL}/V_{SSPLL} are the secondary outputs of VREG3V3V2 that provide the power supply for the PLL and oscillator. These signals are connected to device pins to allow external decoupling capacitors (100 nF...220 nF, X7R ceramic).

In Shutdown Mode an external supply at V_{DDPLL}/V_{SSPLL} can replace the voltage regulator.

16.2.5 V_{REGEN} — Optional Regulator Enable

This optional signal is used to shutdown VREG3V3V2. In that case V_{DD}/V_{SS} and V_{DDPLL}/V_{SSPLL} must be provided externally. Shutdown Mode is entered with V_{REGEN} being low. If V_{REGEN} is high, the VREG3V3V2 is either in Full Performance Mode or in Reduced Power Mode.

For the connectivity of V_{REGEN} see device overview chapter.

NOTE

Switching from FPM or RPM to shutdown of VREG3V3V2 and vice versa is not supported while the MCU is powered.

16.3 Memory Map and Register Definition

This subsection provides a detailed description of all registers accessible in VREG3V3V2.

16.3.1 Module Memory Map

Figure 16-2 provides an overview of all used registers.

Table 16-2. VREG3V3V2 Memory Map

Address Offset	Use	Access
0x0000	VREG3V3V2 Control Register (VREGCTRL)	R/W

17.4.1.3.2 Program Command

The program operation will program a previously erased word in the Flash array using an embedded algorithm.

An example flow to execute the program operation is shown in Figure 17-23. The program command write sequence is as follows:

1. Write to a Flash array address to start the command write sequence for the program command. The data written will be programmed to the Flash array address written.
2. Write the program command, 0x20, to the FCMD register.
3. Clear the CBEIF flag in the FSTAT register by writing a 1 to CBEIF to launch the program command.

If a word to be programmed is in a protected area of the Flash array, the PVIOL flag in the FSTAT register will set and the program command will not launch. Once the program command has successfully launched, the CCIF flag in the FSTAT register will set after the program operation has completed unless a new command write sequence has been buffered. By executing a new program command write sequence on sequential words after the CBEIF flag in the FSTAT register has been set, up to 55% faster programming time per word can be effectively achieved than by waiting for the CCIF flag to set after each program operation.

18.3 Memory Map and Registers

This section describes the FTS32K memory map and registers.

18.3.1 Module Memory Map

The FTS32K memory map is shown in Figure 18-2. The HCS12 architecture places the Flash array addresses between 0x4000 and 0xFFFF, which corresponds to three 16 Kbyte pages. The content of the HCS12 Core PPAGE register is used to map the logical middle page ranging from address 0x8000 to 0xBFFF to any physical 16K byte page in the Flash array memory.[1] The FPROT register (see Section 18.3.2.5) can be set to globally protect the entire Flash array. Three separate areas, one starting from the Flash array starting address (called lower) towards higher addresses, one growing downward from the Flash array end address (called higher), and the remaining addresses, can be activated for protection. The Flash array addresses covered by these protectable regions are shown in Figure 18-2. The higher address area is mainly targeted to hold the boot loader code since it covers the vector space. The lower address area can be used for EEPROM emulation in an MCU without an EEPROM module since it can be left unprotected while the remaining addresses are protected from program or erase. Default protection settings as well as security information that allows the MCU to restrict access to the Flash module are stored in the Flash configuration field described in Table 18-1.

Table 18-1. Flash Configuration Field

Flash Address	Size (bytes)	Description
0xFF00–0xFF07	8	Backdoor Key to unlock security
0xFF08–0xFF0C	5	Reserved
0xFF0D	1	Flash Protection byte Refer to Section 18.3.2.5, “Flash Protection Register (FPROT)”
0xFF0E	1	Reserved
0xFF0F	1	Flash Security/Options byte Refer to Section 18.3.2.2, “Flash Security Register (FSEC)”

1. By placing 0x3E/0x3F in the HCS12 Core PPAGE register, the bottom/top fixed 16 Kbyte pages can be seen twice in the MCU memory map.

Module Base + 0x0009

	7	6	5	4	3	2	1	0
R W	FABLO							
Reset	0	0	0	0	0	0	0	0

Figure 18-14. Flash Address Low Register (FADDRLO)

In normal modes, all FABHI and FABLO bits read 0 and are not writable. In special modes, the FABHI and FABLO bits are readable and writable. For sector erase, the MCU address bits [8:0] are ignored. For mass erase, any address within the Flash array is valid to start the command.

18.3.2.10 Flash Data Register (FDATA)

FDATAHI and FDATALO are the Flash data registers.

Module Base + 0x000A

	7	6	5	4	3	2	1	0
R W	FDHI							
Reset	0	0	0	0	0	0	0	0

Figure 18-15. Flash Data High Register (FDATAHI)

Module Base + 0x000B

	7	6	5	4	3	2	1	0
R W	FDLO							
Reset	0	0	0	0	0	0	0	0

Figure 18-16. Flash Data Low Register (FDATALO)

In normal modes, all FDATAHI and FDATALO bits read 0 and are not writable. In special modes, all FDATAHI and FDATALO bits are readable and writable when writing to an address within the Flash address range.

18.3.2.11 RESERVED3

This register is reserved for factory testing and is not accessible to the user.

19.1.3 Modes of Operation

See Section 19.4.2, “Operating Modes” for a description of the Flash module operating modes. For program and erase operations, refer to Section 19.4.1, “Flash Command Operations”.

19.1.4 Block Diagram

Figure 19-1Figure 19-2 shows a block diagram of the FTS128K1FTS64K module.

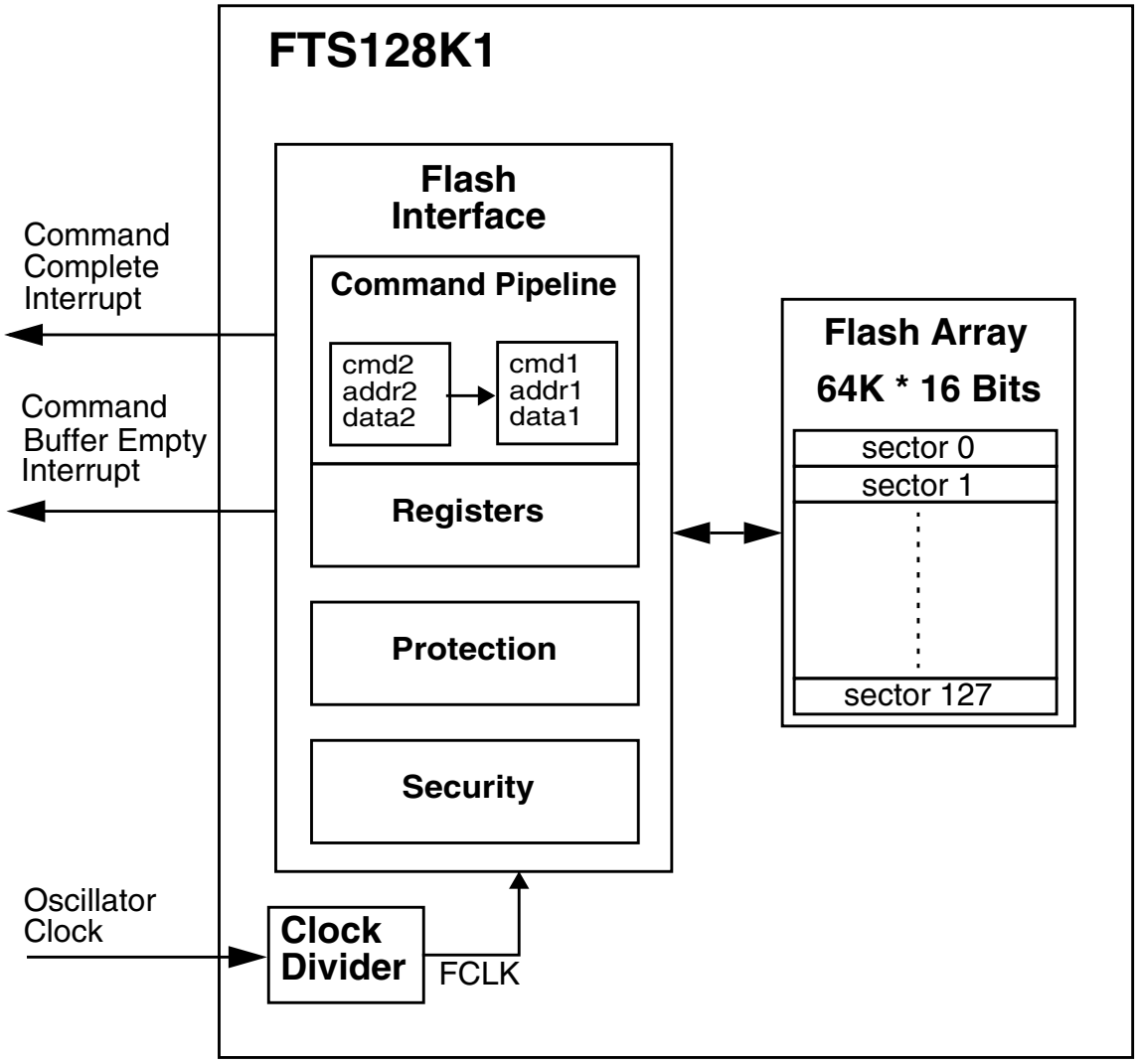

Figure 19-1. FTS128K1 Block Diagram

Table 19-15. FCMD Field Descriptions

Field	Description
6, 5, 2, 0 CMDB[6:5] CMDB[2] CMDB[0]	Valid Flash commands are shown in Table 19-16. An attempt to execute any command other than those listed in Table 19-16 will set the ACCERR bit in the FSTAT register (see Section 19.3.2.6).

Table 19-16. Valid Flash Command List

CMDB	NVM Command
0x05	Erase verify
0x20	Word program
0x40	Sector erase
0x41	Mass erase

19.3.2.8 RESERVED2

This register is reserved for factory testing and is not accessible to the user.

Module Base + 0x0007

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 19-14. RESERVED2

All bits read 0 and are not writable.

19.3.2.9 Flash Address Register (FADDR)

FADDRHI and FADDRLO are the Flash address registers.

Module Base + 0x0008

	7	6	5	4	3	2	1	0
R W	FABHI							
Reset	0	0	0	0	0	0	0	0

Figure 19-15. Flash Address High Register (FADDRHI)

Chapter 21
128 Kbyte Flash Module (S12FTS128K1V1)

21.1 Introduction

The FTS128K1 module implements a 128 Kbyte Flash (nonvolatile) memory. The Flash memory contains one array of 128 Kbytes organized as 1024 rows of 128 bytes with an erase sector size of eight rows (1024 bytes). The Flash array may be read as either bytes, aligned words, or misaligned words. Read access time is one bus cycle for byte and aligned word, and two bus cycles for misaligned words.

The Flash array is ideal for program and data storage for single-supply applications allowing for field reprogramming without requiring external voltage sources for program or erase. Program and erase functions are controlled by a command driven interface. The Flash module supports both mass erase and sector erase. An erased bit reads 1 and a programmed bit reads 0. The high voltage required to program and erase is generated internally. It is not possible to read from a Flash array while it is being erased or programmed.

CAUTION

A Flash word must be in the erased state before being programmed. Cumulative programming of bits within a Flash word is not allowed.

21.1.1 Glossary

Command Write Sequence — A three-step MCU instruction sequence to program, erase, or erase verify the Flash array memory.

21.1.2 Features

- 128 Kbytes of Flash memory comprised of one 128 Kbyte array divided into 128 sectors of 1024 bytes
- Automated program and erase algorithm
- Interrupts on Flash command completion and command buffer empty
- Fast sector erase and word program operation
- 2-stage command pipeline for faster multi-word program times
- Flexible protection scheme to prevent accidental program or erase
- Single power supply for Flash program and erase operations
- Security feature to prevent unauthorized access to the Flash array memory