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	Active
Core Processor	HCS12
Core Size	16-Bit
Speed	32MHz
Connectivity	CANbus, EBI/EMI, I²C, IrDA, LINbus, SCI, SPI
Peripherals	LCD, Motor control PWM, POR, PWM, WDT
Number of I/O	80
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/pro/item?MUrl=&PartUrl=s9s12hy64j0mll

- — Accumulator offsets using A, B, or D accumulators
- — Automatic index predecrement, preincrement, postdecrement, and postincrement (by –8 to +8)

1.3.2 On-Chip Flash with ECC

On-chip flash memory on the MC9S12HY/HA features the following:

- Up to 64 KB of program flash memory
  - — 32 data bits plus 7 syndrome ECC (error correction code) bits allow single bit error correction and double fault detection
  - — Erase sector size 512 bytes
  - — Automated program and erase algorithm
  - — User margin level setting for reads
  - — Protection scheme to prevent accidental program or erase
- 4 KB data flash space
  - — 16 data bits plus 6 syndrome ECC (error correction code) bits allow single bit error correction and double fault detection
  - — Erase sector size 256 bytes
  - — Automated program and erase algorithm
  - — User margin level setting for reads

1.3.3 On-Chip SRAM

- Up to 4 KB of general-purpose RAM, no single cycle misaligned access

1.3.4 Main External Oscillator (XOSC)

- Loop control Pierce oscillator using a 4 MHz to 16 MHz crystal
  - — Current gain control on amplitude output
  - — Signal with low harmonic distortion
  - — Low power
  - — Good noise immunity
  - — Eliminates need for external current limiting resistor
  - — Transconductance sized for optimum start-up margin for typical crystals

1.3.5 Internal RC Oscillator (IRC)

- Trimmable internal reference clock.
  - — Frequency: 1 MHz
  - — Trimmed accuracy over –40˚C to +125˚C ambient temperature range: ±2.0%
  - — Trimmed accuracy over –40˚C to +85˚C ambient temperature range: ±1.5%

is level sensitive and active low. As XIRQ is level sensitive, while this pin is low the MCU will not enter STOP mode. After Reset, the XIRQ default is not enabled.

1.7.3.9 PA0 / $\overline{\text{IRQ}}$ / FP[29]— Port A I/O Pin 0

PA0 is a general-purpose input or output pin. It can be configured as frontplane segment driver outputs FP[29].Tthe maskable interrupt request input that provides a means of applying asynchronous interrupt requests.

1.7.3.10 PB[7:4] / BP[3:0] — Port B I/O Pins [7:4]

PB[7:4] are a general-purpose input or output pins. They can be configured as backplane segment driver outputs BP[3:0].

1.7.3.11 PB[3:0] / FP[39:37,28] — Port B I/O Pins [3:0]

PB[3:0] are a general-purpose input or output pins. They can be configured as frontplane segment driver outputs FP[39:37,28].

1.7.3.12 PS7 / PWM3 / SDA / $\overline{\text{SS}}$ — Port S I/O Pin 7

PS7 is a general-purpose input or output pin. It can be configured as the slave selection pin $\overline{\text{SS}}$ for the serial peripheral interface (SPI). It can be configured as the serial data pin SDA as IIC module. It can be configured as PWM channel 3.

1.7.3.13 PS6 / PWM2 / SCK / KWS6 — Port S I/O Pin 6

PS6 is a general-purpose input or output pin. It can be configured as the serial clock SCK of the serial peripheral interface (SPI). It can be configured as PWM channel 2. It can be configured as keypad wakeup input.

1.7.3.14 PS5 / PWM1 / MOSI / KWS5 — Port S I/O Pin 5

PS5 is a general-purpose input or output pin. It can be configured as the master output (during master mode) or slave input pin (during slave mode) MOSI of the serial peripheral interface (SPI). It can be configured as PWM channel1. It can configured as keypad wakeup input.

1.7.3.15 PS4 / PWM0 / SCL / MISO — Port S I/O Pin 4

PS4 is a general-purpose input or output pin. It can be configured as the master input (during master mode) or slave output pin (during slave mode) MISO for the serial peripheral interface (SPI).It can be configured as the serial clock pin SCL as IIC module.It can be configured as PWM channel0

1.7.3.16 PS3 / TXCAN — Port S I/O Pin 3

PS3 is a general-purpose input or output pin. It can be configured as the transmit pin TXCAN of the scalable controller area network controller (CAN).

Table 2-57. PIFS Register Field Descriptions

Field	Description
6-5 PIFS	**Port S interrupt flag**— Each flag is set by an active edge on the associated input pin. This could be a rising or a falling edge based on the state of the PPSS register. To clear this flag, write logic level 1 to the corresponding bit in the PIFS register. Writing a 0 has no effect. 1 Active edge on the associated bit has occurred (an interrupt will occur if the associated enable bit is set). 0 No active edge pending.

2.3.69 Port AD Interrupt Enable Register (PIE1AD)

Address 0x028C Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PIE1AD7	PIE1AD6	PIE1AD5	PIE1AD4	PIE1AD3	PIE1AD2	PIE1AD1	PIE1AD0
Reset	0	0	0	0	0	0	0	0

Figure 2-67. Port AD Interrupt Enable Register (PIE1AD)

[1] Read: Anytime.
Write: Anytime.

Table 2-58. PIE1AD Register Field Descriptions

Field	Description
7-0 PIE1AD	**Port AD interrupt enable**— This register disables or enables on a per-pin basis the edge sensitive external interrupt associated with Port AD. 1 Interrupt is enabled. 0 Interrupt is disabled (interrupt flag masked).

2.3.70 Port AD Interrupt Flag Register (PIF1AD)

Address 0x028D Access: User read/write[1]

	7	6	5	4	3	2	1	0
R W	PIF1AD7	PIF1AD6	PIF1AD5	PIF1AD4	PIF1AD3	PIF1AD2	PIF1AD1	PIF1AD0
Reset	0	0	0	0	0	0	0	0

Figure 2-68. Port AD Interrupt Flag Register (PIF1AD)

[1] Read: Anytime.
Write: Anytime.

If the X bit maskable interrupt request is used to wake-up the MCU with the X bit in the CCR set, the associated ISR is not called. The CPU then resumes program execution with the instruction following the WAI or STOP instruction. This features works following the same rules like any interrupt request, that is care must be taken that the X interrupt request used for wake-up remains active at least until the system begins execution of the instruction following the WAI or STOP instruction; otherwise, wake-up may not occur.

1. The capability of the $\overline{\text{XIRQ}}$ pin to wake-up the MCU with the X bit set may not be available if, for example, the $\overline{\text{XIRQ}}$ pin is shared with other peripheral modules on the device. Please refer to the Device section of the MCU reference manual for details.

7.1.2.3 Stop Mode

This mode is entered by executing the CPU STOP instruction.

The voltage regulator is in Reduced Power Mode (RPM).

The API is available.

The Phase Locked Loop (PLL) is off.

The Internal Reference Clock (IRC1M) is off.

Core Clock, Bus Clock and BDM Clock are stopped.

Depending on the setting of the PSTP and the OSCE bit, Stop Mode can be differentiated between Full Stop Mode (PSTP = 0 or OSCE=0) and Pseudo Stop Mode (PSTP = 1 and OSCE=1).

- **Full Stop Mode (PSTP=0 or OSCE=0)**
  The external oscillator (OSCLCP) is disabled.
  After wake-up from Full Stop Mode the Core Clock and Bus Clock are running on PLLCLK (PLLSEL=1). After wake-up from Full Stop Mode COP and RTI are running on IRCCLK (COPOSCSEL=0, RTIOSCSEL=0).
- **Pseudo Stop Mode (PSTP=1 and OSCE=1)**
  The external oscillator (OSCLCP) continues torun. If the respective enable bits are set the COP and RTI will continue to run.
  The clock configuration bits PLLSEL, COPOSCSEL, RTIOSCSEL are unchanged.

NOTE

When starting up the external oscillator (either by programming OSCE bit to 1 or on exit from Full Stop Mode with OSCE bit already 1) the software must wait for a minimum time equivalent to the startup-time of the external oscillator t_{UPOSC} before entering Pseudo Stop Mode.

7.3.2.15 Autonomous Periodical Interrupt Control Register (CPMUAPICTL)

The CPMUAPICTL register allows the configuration of the autonomous periodical interrupt features.

0x02F2

	7	6	5	4	3	2	1	0
R	APICLK	0	0	APIES	APIEA	APIFE	APIE	APIF
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 7-18. Autonomous Periodical Interrupt Control Register (CPMUAPICTL)

Read: Anytime

Write: Anytime

Table 7-15. CPMUAPICTL Field Descriptions

Field	Description
7 APICLK	**Autonomous Periodical Interrupt Clock Select Bit** — Selects the clock source for the API. Writable only if APIFE = 0. APICLK cannot be changed if APIFE is set by the same write operation. 0 Autonomous periodical interrupt clock used as source. 1 Bus Clock used as source.
4 APIES	**Autonomous Periodical Interrupt External Select Bit** — Selects the waveform at the external pin API_EXTCLK as shown in Figure 7-19. See device level specification for connectivity of API_EXTCLK pin. 0 If APIEA and APIFE are set, at the external pin API_EXTCLK periodic high pulses are visible at the end of every selected period with the size of half of the minimum period (APIR=0x0000 in Table 7-19). 1 If APIEA and APIFE are set, at the external pin API_EXTCLK a clock is visible with 2 times the selected API Period.
3 APIEA	**Autonomous Periodical Interrupt External Access Enable Bit** — If set, the waveform selected by bit APIES can be accessed externally. See device level specification for connectivity. 0 Waveform selected by APIES can not be accessed externally. 1 Waveform selected by APIES can be accessed externally, if APIFE is set.
2 APIFE	**Autonomous Periodical Interrupt Feature Enable Bit** — Enables the API feature and starts the API timer when set. 0 Autonomous periodical interrupt is disabled. 1 Autonomous periodical interrupt is enabled and timer starts running.
1 APIE	**Autonomous Periodical Interrupt Enable Bit** 0 API interrupt request is disabled. 1 API interrupt will be requested whenever APIF is set.
0 APIF	**Autonomous Periodical Interrupt Flag** — APIF is set to 1 when the in the API configured time has elapsed. This flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (APIE = 1), APIF causes an interrupt request. 0 API time-out has not yet occurred. 1 API time-out has occurred.

Table 8-5. External Trigger Channel Select Coding

ETRIGSEL	ETRIGCH3	ETRIGCH2	ETRIGCH1	ETRIGCH0	External trigger source is
0	0	0	0	0	AN0
0	0	0	0	1	AN1
0	0	0	1	0	AN2
0	0	0	1	1	AN3
0	0	1	0	0	AN4
0	0	1	0	1	AN5
0	0	1	1	0	AN6
0	0	1	1	1	AN7
0	1	0	0	0	AN7
0	1	0	0	1	AN7
0	1	0	1	0	AN7
0	1	0	1	1	AN7
0	1	1	0	0	AN7
0	1	1	0	1	AN7
0	1	1	1	0	AN7
0	1	1	1	1	AN7
1	0	0	0	0	ETRIG0[1]
1	0	0	0	1	ETRIG1[1]
1	0	0	1	0	ETRIG2[1]
1	0	0	1	1	ETRIG3[1]
1	0	1	X	X	Reserved
1	1	X	X	X	Reserved

[1] Only if ETRIG3-0 input option is available (see device specification), else ETRISEL is ignored, that means external trigger source is still on one of the AD channels selected by ETRIGCH3-0

8.3.2.3 ATD Control Register 2 (ATDCTL2)

Writes to this register will abort current conversion sequence.

Module Base + 0x0002

	7	6	5	4	3	2	1	0
R	0	AFFC	ICLKSTP	ETRIGLE	ETRIGP	ETRIGE	ASCIE	ACMPIE
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 8-5. ATD Control Register 2 (ATDCTL2)

Read: Anytime

Write: Anytime

An exception to this is when channels are concatenated. Once concatenated mode is enabled (CONxx bits set in PWMCTL register), enabling/disabling the corresponding 16-bit PWM channel is controlled by the low order PWMEx bit.In this case, the high order bytes PWMEx bits have no effect and their corresponding PWM output lines are disabled.

While in run mode, if all eight PWM channels are disabled (PWME7–0 = 0), the prescaler counter shuts off for power savings.

Module Base + 0x0000

	7	6	5	4	3	2	1	0
R W	PWME7	PWME6	PWME5	PWME4	PWME3	PWME2	PWME1	PWME0
Reset	0	0	0	0	0	0	0	0

Figure 11-3. PWM Enable Register (PWME)

Read: Anytime

Write: Anytime

Table 11-1. PWME Field Descriptions

Field	Description
7 PWME7	**Pulse Width Channel 7 Enable** 0 Pulse width channel 7 is disabled. 1 Pulse width channel 7 is enabled. The pulse modulated signal becomes available at PWM output bit 7 when its clock source begins its next cycle.
6 PWME6	**Pulse Width Channel 6 Enable** 0 Pulse width channel 6 is disabled. 1 Pulse width channel 6 is enabled. The pulse modulated signal becomes available at PWM output bit6 when its clock source begins its next cycle. If CON67=1, then bit has no effect and PWM output line 6 is disabled.
5 PWME5	**Pulse Width Channel 5 Enable** 0 Pulse width channel 5 is disabled. 1 Pulse width channel 5 is enabled. The pulse modulated signal becomes available at PWM output bit 5 when its clock source begins its next cycle.
4 PWME4	**Pulse Width Channel 4 Enable** 0 Pulse width channel 4 is disabled. 1 Pulse width channel 4 is enabled. The pulse modulated signal becomes available at PWM, output bit 4 when its clock source begins its next cycle. If CON45 = 1, then bit has no effect and PWM output bit4 is disabled.
3 PWME3	**Pulse Width Channel 3 Enable** 0 Pulse width channel 3 is disabled. 1 Pulse width channel 3 is enabled. The pulse modulated signal becomes available at PWM, output bit 3 when its clock source begins its next cycle.
2 PWME2	**Pulse Width Channel 2 Enable** 0 Pulse width channel 2 is disabled. 1 Pulse width channel 2 is enabled. The pulse modulated signal becomes available at PWM, output bit 2 when its clock source begins its next cycle. If CON23 = 1, then bit has no effect and PWM output bit2 is disabled.

NOTE

Changing the PWM output mode from left aligned to center aligned output (or vice versa) while channels are operating can cause irregularities in the PWM output. It is recommended to program the output mode before enabling the PWM channel.

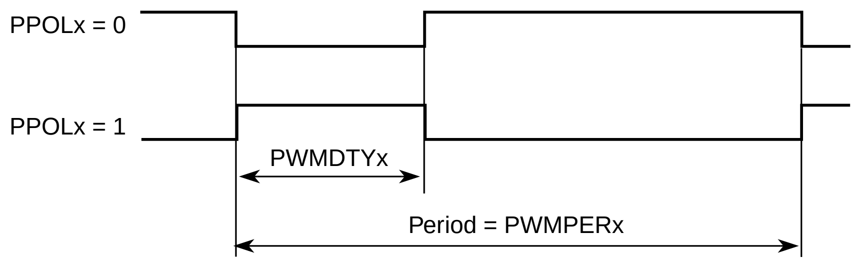

Figure 11-20. PWM Left Aligned Output Waveform

To calculate the output frequency in left aligned output mode for a particular channel, take the selected clock source frequency for the channel (A, B, SA, or SB) and divide it by the value in the period register for that channel.

- PWMx Frequency = Clock (A, B, SA, or SB) / PWMPERx
- PWMx Duty Cycle (high time as a% of period):
  — Polarity = 0 (PPOLx = 0)
- Duty Cycle = [(PWMPERx-PWMDTYx)/PWMPERx] * 100%
  — Polarity = 1 (PPOLx = 1)

Duty Cycle = [PWMDTYx / PWMPERx] * 100%

As an example of a left aligned output, consider the following case:

Clock Source = E, where E = 10 MHz (100 ns period)

PPOLx = 0

PWMPERx = 4

PWMDTYx = 1

PWMx Frequency = 10 MHz/4 = 2.5 MHz

PWMx Period = 400 ns

PWMx Duty Cycle = 3/4 *100% = 75%

The output waveform generated is shown in Figure 11-21.

12.1.4 Block Diagram

Figure 12-1 is a high level block diagram of the SCI module, showing the interaction of various function blocks.

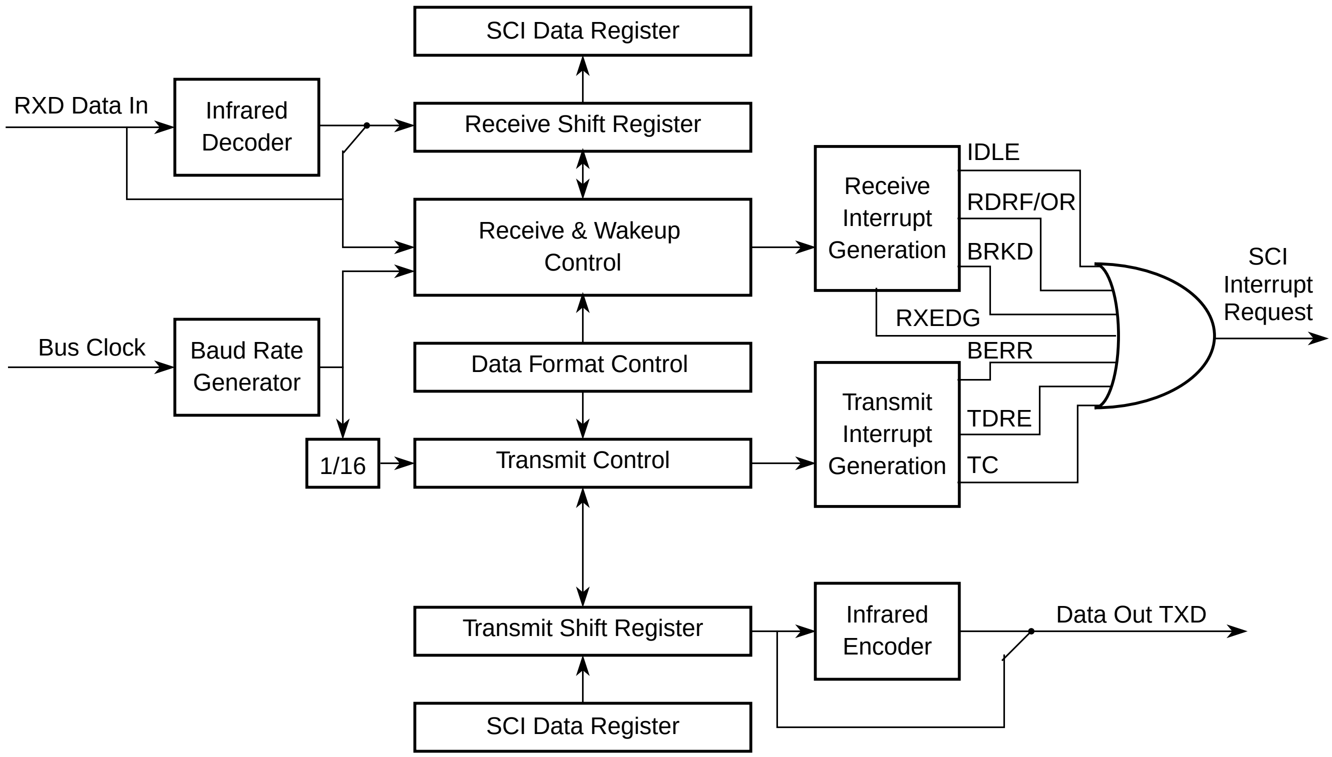

Figure 12-1. SCI Block Diagram

the SPI system is configured as a slave, the $\overline{SS}$ pin is a dedicated input pin. Mode fault error doesn't occur in slave mode.

If a mode fault error occurs, the SPI is switched to slave mode, with the exception that the slave output buffer is disabled. So SCK, MISO, and MOSI pins are forced to be high impedance inputs to avoid any possibility of conflict with another output driver. A transmission in progress is aborted and the SPI is forced into idle state.

If the mode fault error occurs in the bidirectional mode for a SPI system configured in master mode, output enable of the MOMI (MOSI in bidirectional mode) is cleared if it was set. No mode fault error occurs in the bidirectional mode for SPI system configured in slave mode.

The mode fault flag is cleared automatically by a read of the SPI status register (with MODF set) followed by a write to SPI control register 1. If the mode fault flag is cleared, the SPI becomes a normal master or slave again.

NOTE

If a mode fault error occurs and a received data byte is pending in the receive shift register, this data byte will be lost.

13.4.7 Low Power Mode Options

13.4.7.1 SPI in Run Mode

In run mode with the SPI system enable (SPE) bit in the SPI control register clear, the SPI system is in a low-power, disabled state. SPI registers remain accessible, but clocks to the core of this module are disabled.

13.4.7.2 SPI in Wait Mode

SPI operation in wait mode depends upon the state of the SPISWAI bit in SPI control register 2.

- If SPISWAI is clear, the SPI operates normally when the CPU is in wait mode
- If SPISWAI is set, SPI clock generation ceases and the SPI module enters a power conservation state when the CPU is in wait mode.
  - If SPISWAI is set and the SPI is configured for master, any transmission and reception in progress stops at wait mode entry. The transmission and reception resumes when the SPI exits wait mode.
  - If SPISWAI is set and the SPI is configured as a slave, any transmission and reception in progress continues if the SCK continues to be driven from the master. This keeps the slave synchronized to the master and the SCK.

    If the master transmits several bytes while the slave is in wait mode, the slave will continue to send out bytes consistent with the operation mode at the start of wait mode (i.e., if the slave is currently sending its SPIDR to the master, it will continue to send the same byte. Else if the slave is currently sending the last received byte from the master, it will continue to send each previous master byte).

Table 15-4. Program IFR Fields

Global Address	Size (Bytes)	Field Description
0x0_4000 – 0x0_4007	8	Reserved
0x0_4008 – 0x0_40B5	174	Reserved
0x0_40B6 – 0x0_40B7	2	Version ID[1]
0x0_40B8 – 0x0_40BF	8	Reserved
0x0_40C0 – 0x0_40FF	64	Program Once Field Refer to Section 15.4.5.6, “Program Once Command”

1 Used to track firmware patch versions, see Section 15.4.2

Table 15-5. D-Flash and Memory Controller Resource Fields

Global Address	Size (Bytes)	Description
0x0_4000 – 0x0_43FF	1,024	Reserved
0x0_4400 – 0x0_53FF	4,096	D-Flash Memory
0x0_5400 – 0x0_57FF	1,024	Reserved
0x0_5800 – 0x0_5AFF	768	Memory Controller Scratch RAM (RAMON[1] = 1)
0x0_5B00 – 0x0_5FFF	1,280	Reserved
0x0_6000 – 0x0_67FF	2,048	Reserved
0x0_6800 – 0x0_7FFF	6,144	Reserved

1 MMCCTL1 register bit

16.3.2.9.1 P-Flash Protection Restrictions

The general guideline is that P-Flash protection can only be added and not removed. Table 16-20 specifies all valid transitions between P-Flash protection scenarios. Any attempt to write an invalid scenario to the FPROT register will be ignored. The contents of the FPROT register reflect the active protection scenario. See the FPHS and FPLS bit descriptions for additional restrictions.

Table 16-20. P-Flash Protection Scenario Transitions

From Protection Scenario	To Protection Scenario[1]							
	0	1	2	3	4	5	6	7
0	X	X	X	X				
1		X		X				
2			X	X				
3				X				
4				X	X			
5			X	X	X	X		
6		X		X	X		X	
7	X	X	X	X	X	X	X	X

[1] Allowed transitions marked with X, see Figure 16-14 for a definition of the scenarios.

16.3.2.10 D-Flash Protection Register (DFPROT)

The DFPROT register defines which D-Flash sectors are protected against program and erase operations.

Offset Module Base + 0x0009

	7	6	5	4	3	2	1	0
R	DPOPEN	0	0	0	DPS[3:0]			
W								
Reset	F	0	0	0	F	F	F	F

= Unimplemented or Reserved

Figure 16-15. D-Flash Protection Register (DFPROT)

The (unreserved) bits of the DFPROT register are writable with the restriction that protection can be added but not removed. Writes must increase the DPS value and the DPOPEN bit can only be written from 1 (protection disabled) to 0 (protection enabled). If the DPOPEN bit is set, the state of the DPS bits is irrelevant.

During the reset sequence, the DFPROT register is loaded with the contents of the D-Flash protection byte in the Flash configuration field at global address 0x3_FF0D located in P-Flash memory (see Table 16-3) as indicated by reset condition F in Figure 16-15. To change the D-Flash protection that will be loaded during the reset sequence, the P-Flash sector containing the D-Flash protection byte must be unprotected, then the D-Flash protection byte must be programmed. If a double bit fault is detected while reading the

- Supports five LCD operation modes
- 40 frontplane drivers
- 4 backplane drivers
  - Each frontplane has an enable bit respectively
- Programmable frame clock generator
- Programmable bias voltage level selector
- On-chip generation of 4 different output voltage levels

18.1.2 Modes of Operation

The LCD40F4BV1 module supports five operation modes with different numbers of backplanes and different biasing levels. During wait mode the LCD operation can be suspended under software control. Depending on the state of internal bits, the LCD can operate normally or the LCD clock generation can be turned off and the LCD40F4BV1 module enters a power conservation state.

This is a high level description only, detailed descriptions of operating modes are contained in Section 18.4.2, “Operation in Wait Mode”, and Section 18.4.3, “Operation in Stop Mode”.

18.1.3 Block Diagram

Figure 18-1 is a block diagram of the LCD40F4BV1 module.

19.4.2 PWM Duty Cycle

The PWM duty cycle for the motor controller channel x can be determined by dividing the decimal representation of bits D[10:0] in MCDCx by the decimal representation of the bits P[10:0] in MCPER and multiplying the result by 100% as shown in the equation below:

$$\text{Effective PWM Channel X \% Duty Cycle} = \frac{\text{DUTY}}{\text{MCPER}} \cdot 100\%$$

NOTE

x = PWM Channel Number = 0, 1, 2, 3 ... 8. This equation is only valid if DUTY <= MCPER and MCPER is not equal to 0.

Whenever D[10:0] >= P[10:0], a constant low level (RECIRC = 0) or high level (RECIRC = 1) will be output.

19.4.3 Motor Controller Counter Clock Source

Figure 19-22 shows how the PWM motor controller timer counter clock source is selected.

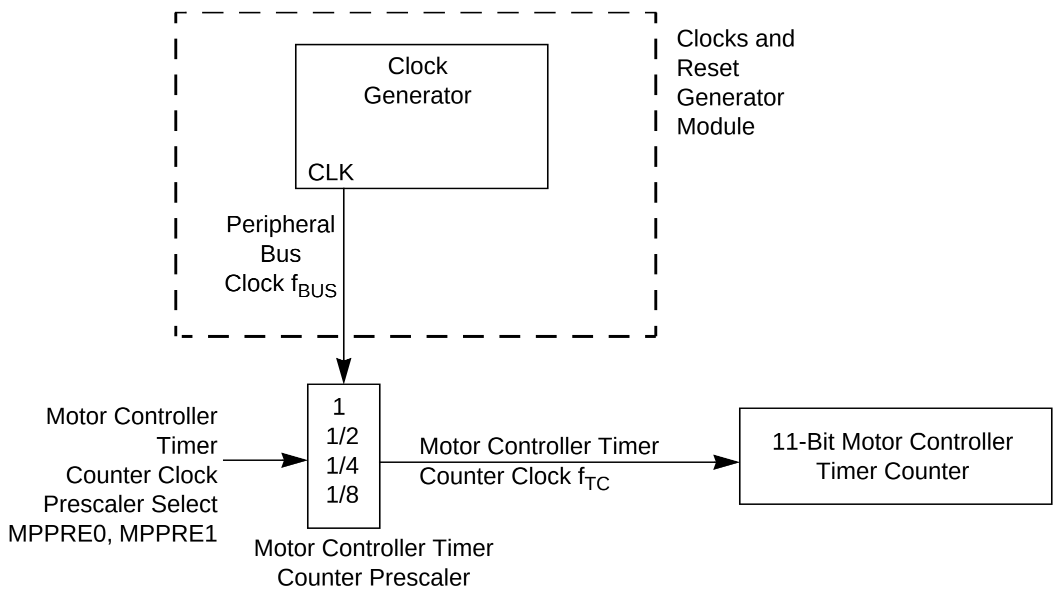

Figure 19-22. Motor Controller Counter Clock Selection

The peripheral bus clock is the source for the motor controller counter prescaler. The motor controller counter clock rate, f_{TC}, is set by selecting the appropriate prescaler value. The prescaler is selected with the MCPRE[1:0] bits in motor controller control register 0 (MCCTL0). The motor controller channel frequency of operation can be calculated using the following formula if DITH = 0:

$$\text{Motor Channel Frequency (Hz)} = \frac{f_{TC}}{\text{MCPER} \cdot \text{M}}$$

Table A-6. 5-V I/O Characteristics

18	D	Port T, S, R, AD interrupt input pulse passed ($\overline{\text{STOP}}$)	t_{PULSE}	4	—	—	tcyc
19	D	$\overline{\text{IRQ}}$ pulse width, edge-sensitive mode ($\overline{\text{STOP}}$)	PW_{IRQ}	1	—	—	tcyc

1. Maximum leakage current occurs at maximum operating temperature. Current decreases by approximately one-half for each 8 C to 12°C in the temperature range from 50°C to 125°C.
2. Refer to Section A.1.4, “Current Injection” for more details
3. Parameter only applies in stop or pseudo stop mode.

A.1.10 Supply Currents

This section describes the current consumption characteristics of the device as well as the conditions for the measurements.

A.1.10.1 Measurement Conditions

IDD value is measured on VDDR pin. It does not include the current to drive external loads. Unless otherwise noted the currents are measured in special single chip mode and the CPU code is executed from RAM. For Run and Wait current measurements PLL is on and the reference clock is the IRC trimmed to 1 MHz. The bus frequency is 32 MHz and the CPU frequency is 64 MHz. Table A-7, Table A-8 and Table A-9 show the configuration of the CPMU module and the peripherals for Run, Wait and Stop current measurement.

Table A-7. CPMU Configuration for Pseudo Stop Current Measurement

CPMU REGISTER	Bit settings/Conditions
CPMUCLKS	PLLSEL=0, PSTP=1, PRE=PCE=RTIOSCSEL=COPOSCSEL=1
CPMUOSC	OSCE=1, External Square wave on EXTAL f_{EXTAL}=16MHz, V_{IH}= 1.8V, V_{IL}=0V
CPMURTI	RTDEC=0, RTR[6:4]=111, RTR[3:0]=1111;
CPMUCOP	WCOP=1, CR[2:0]=111

Table A-8. CPUM Configuration for Run/Wait and Full Stop Current Measurement

CPMU REGISTER	Bit settings/Conditions
CPMUSYNR	VCOFRQ[1:0]=01,SYNDIV[5:0] = 32
CPMUPOSTDIV	POSTDIV[4:0]=0,
CPMUCLKS	PLLSEL=1
CPMUOSC	OSCE=0, Reference clock for PLL is f_{ref}=f_{irc1m} trimmed to 1MHz

In Table A-26 the timing characteristics for master mode are listed.

Table A-26. SPI Master Mode Timing Characteristics

Num	C	Characteristic	Symbol	Min	Typ	Max	Unit
1	D	SCK frequency	f_{sck}	$f_{bus}/2048$	—	MIN(16, $f_{bus}/2$)[1] MIN(10,$f_{bus}/2$)[2] MIN(0.8,$f_{bus}/2$)[3]	MHZ
1	D	SCK period	t_{sck}	MAX(62.5, 2*t_{bus})[1] MAX(100, 2*t_{bus})[2] MAX(1250, 2*t_{bus})[3]	—	$2048 * t_{bus}$	ns
2	D	Enable lead time	t_{lead}	—	1/2	—	t_{sck}
3	D	Enable lag time	t_{lag}	—	1/2	—	t_{sck}
4	D	Clock (SCK) high or low time	t_{wsck}	—	1/2	—	t_{sck}
5	D	Data setup time (inputs)	t_{su}	8[1,2]	—	—	ns
				220[3]	—	—	ns
6	D	Data hold time (inputs)	t_{hi}	8[1,2]	—	—	ns
				220[3]	—	—	ns
9	D	Data valid after SCK edge	t_{vsck}	—	—	15[1,2]	ns
						220[3]	ns
10	D	Data valid after $\overline{SS}$ fall (CPHA = 0)	t_{vss}	—	—	15	ns
11	D	Data hold time (outputs)	t_{ho}	20	—	—	ns
12	D	Rise and fall time inputs	t_{rfi}	—	—	8[1,2]	ns
						85[3]	ns
13	D	Rise and fall time outputs	t_{rfo}	—	—	8[1,2]	ns
						85[3]	ns

1. SPI on non-motor pad ports (Port S or Por t H)
2. SPI on Port V with slew rate control disable. All the SPI pins slew rate control should be disabled.
3. SPI on Port V with slew rate control enabled. All the SPI pins slew rate control should be enabled.
4. MIN(16, f_{bus}/2) means select minimum frequency value from 16MHZ and f_{bus}/2MHZ. same for the other MIN(X,Y)
5. MAX(62.5, 2*t_{bus}) means select the maximum period value from 62.5ns and 2*t_{bus} ns. same for the other MAX(X,Y)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x006F	Reserved	R	0	0	0	0	0	0	0	0
		W								

0x0070–0x009F Analog to Digital converter (ATD) Map

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0070	ATDCTL0	R	Reserved	0	0	0	WRAP3	WRAP2	WRAP1	WRAP0
		W								
0x0071	ATDCTL1	R	ETRIGSEL	SRES1	SRES0	SMP_DIS	ETRIGCH3	ETRIGCH2	ETRIGCH1	ETRIGCH0
		W								
0x0072	ATDCTL2	R	0	AFFC	ICLKSTP	ETRIGLE	ETRIGP	ETRIGE	ASCIE	ACMPIE
		W								
0x0073	ATDCTL3	R	DJM	S8C	S4C	S2C	S1C	FIFO	FRZ1	FRZ0
		W								
0x0074	ATDCTL4	R	SMP2	SMP1	SMP0	PRS[4:0]				
		W								
0x0075	ATDCTL5	R	0	SC	SCAN	MULT	CD	CC	CB	CA
		W								
0x0076	ATDSTAT0	R	SCF	0	ETORF	FIFOR	CC3	CC2	CC1	CC0
		W								
0x0077	Unimple-mented	R	0	0	0	0	0	0	0	0
		W								
0x0078	ATDCMPEH	R	0	0	0	0	0	0	0	0
		W								
0x0079	ATDCMPEL	R	CMPE[7:0]							
		W								
0x007A	ATDSTAT2H	R	0	0	0	0	0	0	0	0
		W								
0x007B	ATDSTAT2L	R	CCF[7:0]							
		W								
0x007C	ATDDIENH	R	0	0	0	0	0	0	0	0
		W								
0x007D	ATDDIENL	R	IEN[7:0]							
		W								
0x007E	ATDCMPHTH	R	0	0	0	0	0	0	0	0
		W								
0x007F	ATDCMPHTL	R	CMPHT[7:0]							
		W								
0x0080	ATDDR0	R	See Section 8.3.2.12.1, “Left Justified Result Data (DJM=0)” and Section 8.3.2.12.2, “Right Justified Result Data (DJM=1)”							
		W								

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00AA	PWMSCNTA	R	0	0	0	0	0	0	0	0
		W								
0x00AB	PWMSCNTB	R	0	0	0	0	0	0	0	0
		W								
0x00AC	PWMCNT0	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x00AD	PWMCNT1	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x00AE	PWMCNT2	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x00AF	PWMCNT3	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x00B0	PWMCNT4	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x00B1	PWMCNT5	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x00B2	PWMCNT6	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x00B3	PWMCNT7	R	Bit 7	6	5	4	3	2	1	Bit 0
		W	0	0	0	0	0	0	0	0
0x00B4	PWMPER0	R/W	Bit 7	6	5	4	3	2	1	Bit 0
0x00B5	PWMPER1	R/W	Bit 7	6	5	4	3	2	1	Bit 0
0x00B6	PWMPER2	R/W	Bit 7	6	5	4	3	2	1	Bit 0
0x00B7	PWMPER3	R/W	Bit 7	6	5	4	3	2	1	Bit 0
0x00B8	PWMPER4	R/W	Bit 7	6	5	4	3	2	1	Bit 0
0x00B9	PWMPER5	R/W	Bit 7	6	5	4	3	2	1	Bit 0
0x00BA	PWMPER6	R/W	Bit 7	6	5	4	3	2	1	Bit 0
0x00BB	PWMPER7	R/W	Bit 7	6	5	4	3	2	1	Bit 0
0x00BC	PWMDTY0	R/W	Bit 7	6	5	4	3	2	1	Bit 0
0x00BD	PWMDTY1	R/W	Bit 7	6	5	4	3	2	1	Bit 0