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	CPU12
Core Size	16-Bit
Speed	8MHz
Connectivity	SCI, SPI
Peripherals	POR, PWM, WDT
Number of I/O	63
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	768 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	4.5V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-QFP
Supplier Device Package	80-QFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68hc912b32cfu8

15.8.5 State Machine . . . 229
15.8.5.1 4X Mode . . . 229
15.8.5.2 Receiving a Message in Block Mode . . . 229
15.8.5.3 Transmitting a Message in Block Mode . . . 230
15.8.5.4 J1850 Bus Errors . . . 230
15.8.5.5 Summary . . . 231
15.9 BDLC Registers . . . 231
15.9.1 BDLC Control Register 1 . . . 231
15.9.2 BDLC Control Register 2 . . . 233
15.9.3 BDLC State Vector Register . . . 237
15.9.4 BDLC Data Register . . . 239
15.9.5 BDLC Analog Roundtrip Delay Register . . . 240
15.9.6 Port DLC Control Register . . . 241
15.9.7 Port DLC Data Register . . . 242
15.9.8 Port DLC Data Direction Register . . . 242

Chapter 16
msCAN12 Controller

16.1 Introduction . . . 243
16.2 External Pins . . . 243
16.3 Message Storage . . . 243
16.3.1 Background . . . 244
16.3.2 Receive Structures . . . 245
16.3.3 Transmit Structures . . . 246
16.4 Identifier Acceptance Filter . . . 247
16.5 Interrupts . . . 250
16.5.1 Interrupt Acknowledge . . . 250
16.5.2 Interrupt Vectors . . . 250
16.6 Protocol Violation Protection . . . 251
16.7 Low-Power Modes . . . 251
16.7.1 msCAN12 Sleep Mode . . . 252
16.7.2 msCAN12 Soft-Reset Mode . . . 253
16.7.3 msCAN12 Power-Down Mode . . . 254
16.7.4 Programmable Wakeup Function . . . 254
16.8 Timer Link . . . 254
16.9 Clock System . . . 254
16.10 Memory Map . . . 257
16.11 Programmer’s Model of Message Storage . . . 257
16.11.1 Message Buffer Organization . . . 257
16.11.2 Identifier Registers . . . 258
16.11.3 Data Length Register . . . 260
16.11.4 Data Segment Registers . . . 261
16.11.5 Transmit Buffer Priority Register . . . 261
16.12 Programmer’s Model of Control Registers . . . 262
16.12.1 msCAN12 Module Control Register 0 . . . 262
16.12.2 msCAN12 Module Control Register 1 . . . 263
16.12.3 msCAN12 Bus Timing Register 0 . . . 264

16.12.4 msCAN12 Bus Timing Register 1 . . . 265
16.12.5 msCAN12 Receiver Flag Register . . . 266
16.12.6 msCAN12 Receiver Interrupt Enable Register . . . 268
16.12.7 msCAN12 Transmitter Flag Register . . . 269
16.12.8 msCAN12 Transmitter Control Register . . . 270
16.12.9 msCAN12 Identifier Acceptance Control Register . . . 270
16.12.10 msCAN12 Receive Error Counter . . . 271
16.12.11 msCAN12 Transmit Error Counter . . . 272
16.12.12 msCAN12 Identifier Acceptance Registers . . . 272
16.12.13 msCAN12 Identifier Mask Registers . . . 273
16.12.14 msCAN12 Port CAN Control Register . . . 275
16.12.15 msCAN12 Port CAN Data Register . . . 275
16.12.16 msCAN12 Port CAN Data Direction Register . . . 276

Chapter 17
Analog-to-Digital Converter (ATD)

17.1 Introduction . . . 277
17.2 Functional Description . . . 277
17.3 ATD Registers . . . 279
17.3.1 ATD Control Register 0 . . . 279
17.3.2 ATD Control Register 1 . . . 279
17.3.3 ATD Control Register 2 . . . 279
17.3.4 ADT Control Register 3 . . . 280
17.3.5 ATD Control Register 4 . . . 281
17.3.6 ATD Control Register 5 . . . 282
17.3.7 ATD Status Registers . . . 284
17.3.8 ATD Test Registers . . . 285
17.3.9 Port AD Data Input Register . . . 286
17.3.10 ATD Result Registers . . . 286
17.4 ATD Mode Operation . . . 287
17.5 Using the ATD to Measure a Potentiometer Signal . . . 287
17.5.1 Equipment . . . 287
17.5.2 Code Listing . . . 287

Chapter 18
Development Support

18.1 Introduction . . . 289
18.2 Instruction Queue . . . 289
18.3 Background Debug Mode (BDM) . . . 290
18.3.1 BDM Serial Interface . . . 290
18.3.2 Enabling BDM Firmware Commands . . . 292
18.3.3 BDM Commands . . . 293
18.3.4 BDM Registers . . . 295
18.3.5 BDM Instruction Register . . . 296
18.3.5.1 Hardware Command . . . 296
18.3.5.2 Firmware Command . . . 297

18.3.6 BDM Status Register . . . 298
18.3.7 BDM Shifter Register . . . 298
18.3.8 BDM Address Register . . . 299
18.3.9 BDM CCR Holding Register . . . 299
18.4 Breakpoints . . . 299
18.4.1 Breakpoint Modes . . . 300
18.4.1.1 SWI Dual Address Mode . . . 300
18.4.1.2 BDM Full Breakpoint Mode . . . 300
18.4.1.3 BDM Dual Address Mode . . . 300
18.4.2 Breakpoint Registers . . . 301
18.4.2.1 Breakpoint Control Register 0 . . . 301
18.4.2.2 Breakpoint Control Register 1 . . . 302
18.4.2.3 Breakpoint Address Register High . . . 303
18.4.2.4 Breakpoint Address Register Low . . . 303
18.4.2.5 Breakpoint Data Register High . . . 304
18.4.2.6 Breakpoint Data Register Low Byte . . . 304
18.5 Instruction Tagging . . . 304

Chapter 19
Electrical Specifications

19.1 Introduction . . . 307
19.2 Maximum Ratings . . . 307
19.3 Functional Operating Range . . . 308
19.4 Thermal Characteristics . . . 308
19.5 5.0 Volt DC Electrical Characteristics . . . 309
19.6 Supply Current . . . 310
19.7 ATD Maximum Ratings . . . 310
19.8 ATD DC Electrical Characteristics . . . 311
19.9 Analog Converter Operating Characteristics . . . 312
19.10 ATD AC Operating Characteristics (Operating) . . . 313
19.11 EEPROM Characteristics . . . 313
19.12 FLASH EEPROM Characteristics . . . 314
19.12.1 Programming Voltage Supply Envelope . . . 315
19.12.2 Example V_{FP} Protection Circuitry . . . 316
19.13 Pulse-Width Modulator Characteristics . . . 319
19.14 Control Timing . . . 320
19.15 Peripheral Port Timing . . . 325
19.16 Multiplexed Expansion Bus Timing . . . 326
19.17 Serial Peripheral Interface (SPI) Timing . . . 328

Chapter 20
Mechanical Specifications

20.1 Introduction . . . 331
20.2 80-Pin Quad Flat Pack (Case 841B-02) . . . 332

Table 1-3. Signal Description Summary

Pin Name	Pin Number	Description
PW3–PW0	3–6	Pulse-width modulator channel outputs
ADDR7–ADDR0 DATA7–DATA0	25–18	External bus pins share function with general-purpose I/O ports A and B. In single-chip modes, the pins can be used for I/O. In expanded modes, the pins are used for the external buses.
ADDR15–ADDR8 DATA15–DATA8	46–39	
IOC7–IOC0	16–12, 9–7	Pins used for input capture and output compare in the timer and pulse accumulator subsystem
PAI	16	Pulse accumulator input
AN7–AN0	58–51	Analog inputs for the analog-to-digital conversion module
$\overline{\text{DBE}}$	26	Data bus control and, in expanded mode, enables the drive control of external buses during external reads
MODB, MODA	27, 28	State of mode select pins during reset determines the initial operating mode of the MCU. After reset, MODB and MODA can be configured as instruction queue tracking signals IPIPE1 and IPIPE0 or as general-purpose I/O pins.
IPIPE1, IPIPE0	27, 28	
ECLK	29	E-clock is the output connection for the external bus clock. ECLK is used as a timing reference and for address demultiplexing.
$\overline{\text{RESET}}$	32	An active low bidirectional control signal, $\overline{\text{RESET}}$ acts as an input to initialize the MCU to a known startup state and an output when COP or clock monitor causes a reset.
EXTAL	33	Crystal driver and external clock input pins. On reset all the device clocks are derived from the EXTAL input frequency. XTAL is the crystal output.
XTAL	34	
$\overline{\text{LSTRB}}$	35	Low byte strobe (0 = low byte valid), in all modes this pin can be used as I/O. The low strobe function is the exclusive-NOR of A0 and the internal $\overline{\text{SZ8}}$ signal. The $\overline{\text{SZ8}}$ internal signal indicates the size 16/8 access.
$\overline{\text{TAGLO}}$	35	Pin used in instruction tagging
R/$\overline{\text{W}}$	36	Indicates direction of data on expansion bus; shares function with general-purpose I/O; read/write in expanded modes
$\overline{\text{IRQ}}$	37	Maskable interrupt request input provides a means of applying asynchronous interrupt requests to the MCU. Either falling edge-sensitive triggering or level-sensitive triggering is program selectable (INTCR register).
$\overline{\text{XIRQ}}$	38	Provides a means of requesting asynchronous non-maskable interrupt requests after reset initialization
BKGD	17	Single-wire background interface pin is dedicated to the background debug function. During reset, this pin determines special or normal operating mode.
TAGHI	17	Pin used in instruction tagging
DLCRx/RxCAN(1)	76	BDLC receive pin
DLCTx/TxCAN(1)	75	BDLC transmit pin
$\overline{\text{CS}}$/$\overline{\text{SS}}$	68	Slave-select output for SPI master mode; input for slave mode or master mode
SCK	67	Serial clock for SPI system
SDO/MOSI	66	Master out/slave in pin for serial peripheral interface
SDI/MISO	65	Master in/slave out pin for serial peripheral interface
TxD0	62	SCI transmit pin
RxD0	61	SCI receive pin

1. The RxCAN and TxCAN designations are for the MC68HC(9)12BC32 only.

Addr.	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$011A	msCAN12 Identifier Acceptance Register 6 (CIDAR6)(3) See page 273.	Read: Write:	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
		Reset:	Unaffected by reset							
$011B	msCAN12 Identifier Acceptance Register 7 (CIDAR7)(3) See page 273.	Read: Write:	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
		Reset:	Unaffected by reset							
$011C	msCAN12 Identifier Mask Register 4 (CIDMR4)(3) See page 274.	Read: Write:	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
		Reset:	Unaffected by reset							
$011D	msCAN12 Identifier Mask Register 5 (CIDMR5)(3) See page 274.	Read: Write:	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
		Reset:	Unaffected by reset							
$011E	msCAN12 Identifier Mask Register 6 (CIDMR6)(3) See page 274.	Read: Write:	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
		Reset:	Unaffected by reset							
$011F	msCAN12 Identifier Mask Register 7 (CIDMR7)(3) See page 274.	Read: Write:	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
		Reset:	Unaffected by reset							
$0120	Reserved		R	R	R	R	R	R	R	R
↓										
$013C	Reserved		R	R	R	R	R	R	R	R
$013D	msCAN12 Port CAN Control Register (PCTLCAN)(3) See page 275.	Read: Write:	0	0	0	0	0	0	PUECAN	RDPCAN
		Reset:	0	0	0	0	0	0	0	0
$013E	msCAN12 Port CAN Data Register (PORTCAN)(3) See page 275.	Read: Write:	PCAN7	PCAN6	PCAN5	PCAN4	PCAN2	PCAN2	TxCAN	RxCAN
		Reset:	Unaffected by reset							
$013F	msCAN12 Port CAN Data Direction Register (DDRCAN)(3) See page 276.	Read: Write:	DDRCAN7	DDRCAN6	DDRCAN5	DDRCAN4	DDRCAN3	DDRCAN2	0	0
		Reset:	0	0	0	0	0	0	0	0
$0140 ↓ $014F	RECEIVE BUFFER (RxFG)(3) — SEE 16.3.2 Receive Structures									
$0150 ↓ $015F	TRANSMIT BUFFER 0 (Tx0)(3) — SEE 16.3.3 Transmit Structures									

= Unimplemented R = Reserved U = Unaffected

Notes:
1. Available only on MC68HC912B32 and MC68HC912BC32 devices.
2. Available only on MC68HC912B32 and MC68HC12BE32 devices.
3. Available only on MC68HC(9)12BC32 devices.

Figure 2-1. Register Map (Sheet 18 of 19)

4.5.2 Highest Priority I Interrupt Register

Address: $001F

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	1	1	PSEL5	PSEL4	PSEL3	PSEL2	PSEL1	0
Write:								
Reset:	1	1	1	1	0	0	1	0

Figure 4-2. Highest Priority I Interrupt Register (HPRIO)

Read: Anytime

Write: Only if I bit in CCR = 1 (interrupts inhibited)

To give a maskable interrupt source highest priority, write the low byte of the vector address to the HPRIO register. For example, writing $F0 to HPRIO assigns highest maskable interrupt priority to the real-time interrupt timer ($FFF0). If an unimplemented vector address or a non-I-masked vector address (a value higher than $F2) is written, then $\overline{\text{IRQ}}$ is the default highest priority interrupt.

4.6 Resets

There are four possible sources of reset. POR and external reset on the $\overline{\text{RESET}}$ pin share the normal reset vector. COP reset and the clock monitor reset each has a vector. Entry into reset is asynchronous and does not require a clock, but the MCU cannot sequence out of reset without a system clock.

4.6.1 Power-On Reset (POR)

A positive transition on V_{DD} causes a POR. An external voltage level detector or other external reset circuits are the usual source of reset in a system. The POR circuit only initializes internal circuitry during cold starts and cannot be used to force a reset as system voltage drops.

4.6.2 External Reset

The CPU distinguishes between internal and external reset conditions by sensing whether the reset pin rises to a logic 1 in less than eight E-clock cycles after an internal device releases reset. When a reset condition is sensed, the $\overline{\text{RESET}}$ pin is driven low by an internal device for about 16 E-clock cycles, then released. Eight E-clock cycles later it is sampled. If the pin is still held low, the CPU assumes that an external reset has occurred. If the pin is high, it indicates that the reset was initiated internally by either the COP system or the clock monitor.

To prevent a COP or clock monitor reset from being detected during an external reset, hold the reset pin low for at least 32 cycles. An external resistor-capacitor (RC) power-up delay circuit on the reset pin is not recommended because circuit charge time can cause the MCU to misinterpret the type of reset that has occurred.

4.6.3 Computer Operating Properly (COP) Reset

The MCU includes a COP system to help protect against software failures. When COP is enabled, software must write $55 and $AA (in this order) to the COPRST register to keep a watchdog timer from timing out. Other instructions may be executed between these writes. A write of any value other than $55 or $AA or software failing to execute the sequence properly causes a COP reset to occur.

6.3.9 Reduced Drive of I/O Lines

Address: $000D

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	0	0	0	0	RDPE	0	RDPB	RDPA
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 6-9. Reduced Drive of I/O Lines (RDRIV)

Read: Anytime, if register is in the map

Write: Once in normal modes; anytime, except the first time, in special modes

These bits select reduced drive for the associated port pins. This gives reduced power consumption and reduced radio frequency interference (RFI) with a slight increase in transition time (depending on loading). The reduced drive function is independent of which function is being used on a particular port. This register is not in the map in peripheral mode.

RDPE — Reduced Drive of Port E Bit

1 = Reduced drive for all port E output pins
0 = Full drive for all port E output pins

RDPB — Reduced Drive of Port B Bit

1 = Reduced drive for all port B output pins
0 = Full drive for all port B output pins

RDPA — Reduced Drive of Port A Bit

1 = Reduced drive for all port A output pins
0 = Full drive for all port A output pins

8.5 Programming the FLASH EEPROM

Programming the FLASH EEPROM is accomplished by this step-by-step procedure. The V_{FP} pin voltage must be at the proper level prior to executing step 4 the first time.

1. Apply program/erase voltage to the V_{FP} pin.
2. Clear ERAS and set the LAT bit in the FEECTL register to establish program mode and enable programming address and data latches.
3. Write data to a valid address. The address and data are latched. If BOOTP is asserted, an attempt to program an address in the boot block will be ignored.
4. Apply programming voltage by setting ENPE.
5. Delay for one programming pulse, t_{PPULSE}.
6. Remove programming voltage by clearing ENPE.
7. Delay while high voltage is turning off, t_{VPROG}.
8. Read the address location to verify that it has been programmed,
9. If the location is not programmed, repeat steps 4 through 7 until the location is programmed or until the specified maximum number of program pulses, n_{PP}, has been reached.
10. If the location is programmed, repeat the same number of pulses as required to program the location. This provides 100 percent program margin.
11. Read the address location to verify that it remains programmed.
12. Clear LAT.
13. If there are more locations to program, repeat steps 2 through 10.
14. Turn off V_{FP}. Reduce voltage on V_{FP} pin to V_{DD}.

The flowchart in Figure 8-5 demonstrates the recommended programming sequence.

10.7.2 Real-Time Interrupt Control Register

Address: $0014

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	RTIE	RSWAI	RSBCK	0	RTBYP	RTR2	RTR1	RTR0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 10-6. Real-Time Interrupt Control Register (RTICTL)

Read: Anytime

Write: Varies on a bit-by-bit basis

RTIE — Real-Time Interrupt Enable Bit

Write anytime.

0 = Interrupt requests from RTI are disabled.

1 = Interrupt is requested when RTI is set.

RSWAI — RTI and COP Stop While in Wait Bit

Write once in normal modes, anytime in special modes.

0 = Allows the RTI and COP to continue running in wait

1 = Disables both the RTI and COP when the part goes into wait

RSBCK — RTI and COP Stop While in Background Debug Mode Bit

Write once in normal modes, anytime in special modes.

0 = Allows the RTI and COP to continue running while in background mode

1 = Disables RTI and COP when the part is in background mode (useful for emulation)

RTBYP — Real-Time Interrupt Divider Chain Bypass Bit

Write is not allowed in normal modes, anytime in special modes.

0 = Divider chain functions normally.

1 = Divider chain is bypassed, allows faster testing. The divider chain is normally P divided by 2^{13}, when bypass becomes P divided by 4.

RTR2, RTR1, and RTR0 — Real-Time Interrupt Rate Select Bits

Write anytime.

Rate select for real-time interrupt. The E clock is used for this module.

Table 10-3. Real-Time Interrupt Rates

RTR2	RTR1	RTR0	Divide E By:	Timeout Period E = 4.0 MHz	Timeout Period E = 8.0 MHz
0	0	0	OFF	OFF	OFF
0	0	1	2^{13}	2.048 ms	1.024 ms
0	1	0	2^{14}	4.096 ms	2.048 ms
0	1	1	2^{15}	8.196 ms	4.096 ms
1	0	0	2^{16}	16.384 ms	8.196 ms
1	0	1	2^{17}	32.768 ms	16.384 ms
1	1	0	2^{18}	65.536 ms	32.768 ms
1	1	1	2^{19}	131.72 ms	65.536 ms

11.2.13 PWM Special Mode Register

Address: $0055

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	DISCR	DISCP	DISCAL	0	0	0	0	0
Write:								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Figure 11-25. PWM Special Mode Register (PWTST)

Read: Anytime

Write: Only in special mode (SMODN = 0)

These bits are available only in special mode and are reset in normal mode.

DISCR — Disable Channel Counter Reset Bit

This bit disables the normal operation of resetting the channel counter when the channel counter is written.

0 = Normal operation

1 = Write to PWM channel counter does not reset channel counter.

DISCP — Disable Compare Count Period Bit

0 = Normal operation

1 = In left-aligned output mode, match of the period does not reset the associated PWM counter register.

DISCAL — Disable Scale Counter Loading Bit

This bit disables the normal operation of loading scale counters on a write to the associated scale register.

0 = Normal operation

1 = Write to PWSCAL0 and PWSCAL1 does not load scale counters.

11.2.14 Port P Data Register

Address: $0056

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	PP7	PP6	PP5	PP4	PP3	PP2	PP1	PP0
Write:								
PWM					PWM3	PWM2	PWM1	PWM0
Reset:	Unaffected by reset							

Figure 11-26. Port P Data Register (PORTP)

Read: Anytime

Write: Anytime

PWM functions share port P pins 3 to 0 and take precedence over the general-purpose port when enabled. When configured as input, a read returns the pin level. When configured as output, a read returns the latched output data.

A write drives associated pins only if configured for output and the corresponding PWM channel is not enabled. After reset, all pins are general-purpose, high-impedance inputs.

12.3.15 Data Direction Register for Timer Port

Address: $00AF

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	DDT7	DDT6	DDT5	DDT4	DDT3	DDT2	DDT1	DDT0
Reset:	0	0	0	0	0	0	0	0

Figure 12-29. Data Direction Register for Timer Port (DDRT)

Read: Anytime

Write: Anytime

0 = Configures the corresponding I/O pin for input only
1 = Configures the corresponding I/O pin for output

The timer forces the I/O state to be an output for each timer port pin associated with an enabled output compare. In these cases the data direction bits will not be changed, but they have no affect on the direction of these pins. The DDRT will revert to controlling the I/O direction of a pin when the associated timer output compare is disabled. Input captures do not override the DDRT settings.

12.4 Timer Operation in Modes

Stop — Timer is off since both PCLK and ECLK are stopped.

BDM— Timer keeps running, unless TSBCK = 1.

Wait — Counters keep running, unless TSWAI = 1.

Normal — Timer keeps running, unless TEN = 0.

TEN = 0 —All timer operations are stopped, registers may be accessed.
Gated pulse accumulator ÷64 clock is also disabled.

PAEN = 0 —All pulse accumulator operations are stopped.
Registers may be accessed.

CLK1 and CLK0 — Clock Select Bits

CLK1	CLK0	Clock Source
0	0	Use timer prescaler clock as timer counter clock
0	1	Use PACLK as input to timer counter clock
1	0	Use PACLK/256 as timer counter clock frequency
1	1	Use PACLK/65,536 as timer counter clock frequency

If the pulse accumulator is disabled (PAEN = 0), the prescaler clock from the timer is always used as an input clock to the timer counter. The change from one selected clock to the other happens immediately after these bits are written.

PAOVI — Pulse Accumulator A Overflow Interrupt Enable Bit

0 = Interrupt inhibited
1 = Interrupt requested if PAOVF is set

PAI — Pulse Accumulator Input Interrupt Enable Bit

0 = Interrupt inhibited
1 = Interrupt requested if PAIF is set

13.4.12 Pulse Accumulator A Flag Register

Address: $00A1

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	0	0	0	0	0	0	PAOVF	PAIF
Reset:	0	0	0	0	0	0	0	0

Figure 13-30. Pulse Accumulator A Flag Register (PAFLG)

Read: Anytime

Write: Anytime

When the TFFCA bit in the TSCR register is set, any access to the PACNT register will clear all the flags in the PAFLG register.

PAOVF — Pulse Accumulator A Overflow Flag

Set when the 16-bit pulse accumulator A overflows from $FFFF to $0000 or when 8-bit pulse accumulator 3 (PAC3) overflows from $FF to $00. This bit is cleared automatically by a write to the PAFLG register with bit 1 set.

PAIF — Pulse Accumulator Input Edge Flag

Set when the selected edge is detected at the PT7 input pin. In event mode, the event edge triggers PAIF and, in gated time accumulation mode, the trailing edge of the gate signal at the PT7 input pin triggers PAIF. This bit is cleared by a write to the PAFLG register with bit 0 set. Any access to the PACN3 and PACN2 registers will clear all the flags in this register when TFFCA bit in register TSCR ($86) is set.

PUPS1 — Pullup Port S Enable PS3 and PS2 Bit

0 = No internal pullups on port S bits 3 and 2

1 = Port S input pins for bits 3 and 2 have an active pullup device. If a pin is programmed as output, the pullup device becomes inactive.

PUPS0 — Pullup Port S Enable PS1 and PS0 Bit

0 = No internal pullups on port S bits 1 and 0

1 = Port S input pins for bits 1 and 0 have an active pullup device. If a pin is programmed as output, the pullup device becomes inactive.

14.5 Serial Character Transmission using the SCI

Code is intended to use SCI1 to serially transmit characters using polling to the LCD display on the UDLP1 board: when the transmission data register is empty a flag will get set, which is telling us that SC1DR is ready so we can write another byte. The transmission is performed at a baud rate of 9600. Since the SCI1 is only being used for transmit data, the data register will not be used bidirectionally for received data.

14.5.1 Equipment

For this exercise, use the M68HC912B32EVB emulation board.

14.5.2 Code Listing

NOTE

A comment line is deliminted by a semi-colon. If there is no code before comment, an ";" must be placed in the first column to avoid assembly errors.

```
INCLUDE 'EQUATES.ASM'       ; Equates for registers

; User Variables

; Bit Equates

; ---------------------------------------------------------------------
;           MAIN PROGRAM
; ---------------------------------------------------------------------
        ORG     $7000             ; 16K On-Board RAM, User code data area,
;                                 ; start main program at $4000
MAIN:
        BSR     INIT              ; Subroutine to Initialize SCI0 registers
        BSR     TRANS             ; Subroutine to start transmission
DONE:   BRA     DONE              ; Always branch to DONE, convenient for breakpoint

; ---------------------------------------------------------------------
;         SUBROUTINE INIT:
; ---------------------------------------------------------------------

INIT:   TPA                       ; Transfer CCR to A accumulator
        ORAA    #$10              ; ORed A with #$10 to Set I bit
        TAP                       ; Transfer A to CCR

        MOVB    #$34,SC1BDL       ; Set BAUD =9600, in SCI1 Baud Rate Reg.

        MOVB    #$00,SC1CR1       ; Initialize for 8-bit Data format,
;                                 ; Loop Mode  and parity disabled,(SC1CR1)
```

16.5 Interrupts

The msCAN12 supports four interrupt vectors mapped onto 11 different interrupt sources, any of which can be individually masked. For details, see 16.12.5 msCAN12 Receiver Flag Register to 16.12.8 msCAN12 Transmitter Control Register.

1. *Transmit interrupt:* At least one of the three transmit buffers is empty (not scheduled) and can be loaded to schedule a message for transmission. The empty message buffers TXE flags are set.
2. *Receive interrupt:* A message has been successfully received and loaded into the foreground receive buffer. This interrupt will be emitted immediately after receiving the end of frame (EOF) symbol. The RXF flag is set.
3. *Wakeup interrupt:* An activity on the CAN bus occurred during msCAN12 internal sleep mode.
4. *Error interrupt:* An overrun, error, or warning condition occurred. The receiver flag register (CRFLG) will indicate one of these conditions:
   - *Overrun:* An overrun condition as described in **16.3.2 Receive Structures** has occurred.
   - *Receiver warning:* The receive error counter has reached the CPU warning limit of 96.
   - *Transmitter warning:* The transmit error counter has reached the CPU warning limit of 96.
   - *Receiver error passive:* The receive error counter has exceeded the error passive limit of 127 and msCAN12 has gone to error passive state.
   - *Transmitter error passive:* The transmit error counter has exceeded the error passive limit of 127 and msCAN12 has gone to error passive state.
   - *Bus off:* The transmit error counter has exceeded 255 and msCAN12 has gone to bus-off state.

16.5.1 Interrupt Acknowledge

Interrupts are associated directly with one or more status flags in either the msCAN12 receiver flag register (CRFLG) or the msCAN12 transmitter control register (CTCR). Interrupts are pending as long as one of the corresponding flags is set. The flags in the aforementioned registers must be reset within the interrupt handler to handshake the interrupt. The flags are reset through writing a 1 to the corresponding bit position. A flag cannot be cleared if the respective condition still prevails.

NOTE

Bit manipulation instructions (BSET) must not be used to clear interrupt flags.

16.5.2 Interrupt Vectors

The msCAN12 supports four interrupt vectors as shown in Table 16-1. The vector addresses are dependent on the chip integration and to be defined. The relative interrupt priority is also integration dependent and to be defined.

Address[1]	Register Name
01x0	IDENTIFIER REGISTER 0
01x1	IDENTIFIER REGISTER 1
01x2	IDENTIFIER REGISTER 2
01x3	IDENTIFIER REGISTER 3
01x4	DATA SEGMENT REGISTER 0
01x5	DATA SEGMENT REGISTER 1
01x6	DATA SEGMENT REGISTER 2
01x7	DATA SEGMENT REGISTER 3
01x8	DATA SEGMENT REGISTER 4
01x9	DATA SEGMENT REGISTER 5
01xA	DATA SEGMENT REGISTER 6
01xB	DATA SEGMENT REGISTER 7
01xC	DATA LENGTH REGISTER
01xD	TRANSMIT BUFFER PRIORITY REGISTER[2]
01xE	UNUSED
01xF	UNUSED

1. x is 4, 5, 6, or 7 depending on which buffer, RxFG, Tx0, Tx1, or Tx2, respectively.
2. Not applicable for receive buffers.

Figure 16-10. Message Buffer Organization

16.11.2 Identifier Registers

The Bosch CAN 2.0A and 2.0B protocol specifications allow for two different sizes of message identifiers. The Bosch CAN 2.0A specification requires an 11-bit identifier in the message buffer and the Bosch CAN 2.0B specification requires a 29-bit identifier in the message buffer. The resulting message buffers are referred to as standard and extended formats, respectively.

The identifier registers (IDRn) in the memory map can be configured to create either the 11-bit (IDR10–IDR0) identifier necessary for the standard format or the 29-bit (IDR28–IDR0) identifier necessary for the extended format. Figure 16-11 details the identifier structure used in the standard format while Figure 16-12 details the identifier structure used in the extended format. ID10/ID28 is the most significant bit and is transmitted first on the bus during the arbitration procedure. The priority of an identifier is defined to be the highest for the smallest binary number.

Addr.(1)	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$01x0	Identifier Register 0 (IDR0)	Read: Write:	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3
		Reset:	Undefined out of reset							
$01x1	Identifier Register 1 (IDR1)	Read: Write:	ID2	ID1	ID0	RTR	IDE			
		Reset:	Undefined out of reset							
$01x2	Identifier Register 2 (IDR2)	Read: Write:								
		Reset:	Undefined out of reset							
$01x3	Identifier Register 3 (IDR3)	Read: Write:								
		Reset:	Undefined out of reset							

Note 1. x is 4, 5, 6, or 7 depending on which buffer, RxFG, Tx0, Tx1, or Tx3, respectively. = Unimplemented

Figure 16-11. Identifier Mapping in the Standard Format

Addr.(1)	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
$01x0	Identifier Register 0 (IDR0)	Read: Write:	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
		Reset:	Undefined out of reset							
$01x1	Identifier Register 1 (IDR1)	Read: Write:	ID20	ID19	ID18	SRR	IDE	ID17	ID16	ID15
		Reset:	Undefined out of reset							
$01x2	Identifier Register 2 (IDR2)	Read: Write:	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
		Reset:	Undefined out of reset							
$01x3	Identifier Register 3 (IDR3)	Read: Write:	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR
		Reset:	Undefined out of reset							

Note 1. x is 4, 5, 6, or 7 depending on which buffer, RxFG, Tx0, Tx1, or Tx3, respectively.

Figure 16-12. Identifier Mapping in the Extended Format

SRR — Substitute Remote Request Bit

This fixed recessive bit is used only in extended format. It must be set to 1 by the user for transmission buffers and will be stored as received on the CAN bus for receive buffers.

IDE — ID Extended Flag

This flag indicates whether the extended or standard identifier format is applied in this buffer. In case of a receive buffer, the flag is set as received and indicates to the CPU how to process the buffer identifier registers. In case of a transmit buffer the flag indicates to the msCAN12 what type of identifier to send.

0 = Standard format (11 bit)
1 = Extended format (29 bit)

FRZ1 and FRZ0 — Background Debug (Freeze) Enable Bits

When debugging an application, it is useful in many cases to have the ATD pause when a breakpoint is encountered. These two bits determine how the ATD will respond when background debug mode becomes active. See Table 17-1.

Table 17-1. ATD Response to Background Debug Enable

FRZ1	FRZ0	ATD Response
0	0	Continue conversions in active background mode
0	1	Reserved
1	0	Finish current conversion, then freeze
1	1	Freeze when BDM is active

17.3.5 ATD Control Register 4

Address: $0064

	Bit 7	6	5	4	3	2	1	Bit 0
Read: Write:	S10BM	SMP1	SMP0	PRS4	PRS3	PRS2	PRS1	PRS0
Reset:	0	0	0	0	0	0	0	1

Figure 17-6. ATD Control Register 4 (ATDCTL4)

The ATD control register 4 (ATDCTL4) selects the clock source and sets up the prescaler. Writes to the ATD control registers initiate a new conversion sequence. If a write occurs while a conversion is in progress, the conversion is aborted and ATD activity halts until a write to ATDCTL5 occurs.

S10BM — ATD 10-Bit Mode Control Bit

0 = 8-bit operation
1 = 10-bit operation

SMP1 and SMP0 — Select Sample Time Bits

These bits are used to select one of four sample times after the buffered sample and transfer has occurred. See Table 17-2.

Table 17-2. Final Sample Time Selection

SMP1	SMP0	Final Sample Time	Total 8-Bit Conversion Time	Total 10-Bit Conversion Time
0	0	2 ATD clock periods	18 ATD clock periods	20 ATD clock periods
0	1	4 ATD clock periods	20 ATD clock periods	22 ATD clock periods
1	0	8 ATD clock periods	24 ATD clock periods	26 ATD clock periods
1	1	16 ATD clock periods	32 ATD clock periods	34 ATD clock periods

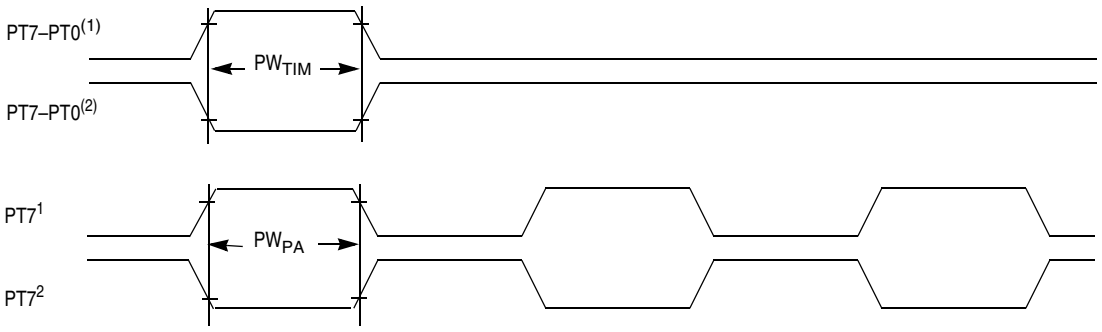
PRS4–PRS0 — Select Divide-By Factor for ATD P-Clock Prescaler Bits

The binary value written to these bits (1 to 31) selects the divide-by factor for the modulo counter-based prescaler. The P clock is divided by this value plus one, and then fed into a divide-by-two circuit to generate the ATD module clock. The divide-by-two circuit ensures symmetry of the output clock signal.

19.14 Control Timing

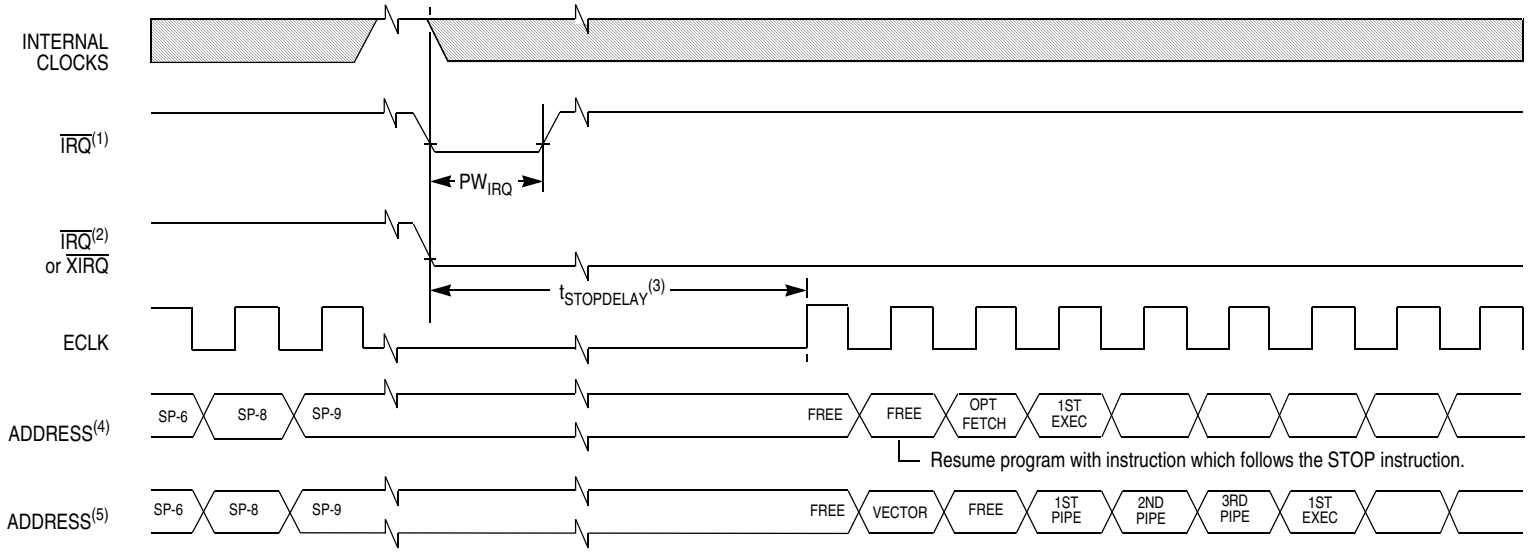
Characteristic	Symbol	8.0 MHz		Unit
		Min	Max	
Frequency of operation	f_o	dc	8.0	MHz
E-clock period	t_{cyc}	125	—	ns
Crystal frequency	f_{XTAL}	—	16.0	MHz
External oscillator frequency	$2\ f_o$	dc	16.0	MHz
Processor control setup time $t_{PCSU} = t_{cyc}/2 + 20$	t_{PCSU}	82	—	ns
Reset input pulse width(1) To guarantee external reset vector Minimum input time (can be pre-empted by internal reset)	PW_{RSTL}	 32 2	 — —	t_{cyc}
Mode programming setup time	t_{MPS}	4	—	t_{cyc}
Mode programming hold time	t_{MPH}	10	—	ns
Interrupt pulse width, $\overline{IRQ}$ edge-sensitive mode $PW_{IRQ} = 2\ t_{cyc} + 20$	PW_{IRQ}	270	—	ns
Wait recovery startup time $t_{WRS} = 4\ t_{cyc}$	t_{WRS}	—	4	t_{cyc}
Timer input capture pulse width $PW_{TIM} = 2\ t_{cyc} + 20$	PW_{TIM}	270	—	ns
Pulse accumulator pulse width	PW_{PA}	TBD	—	ns

1. $\overline{RESET}$ is recognized during the first clock cycle it is held low. Internal circuitry then drives the pin low for 16 clock cycles, releases the pin, and samples the pin level eight cycles later to determine the source of the interrupt.



Notes:
1. Rising edge sensitive input
2. Falling edge sensitive input

Figure 19-5. Timer Inputs



Notes:

1. Edge-sensitive $\overline{\text{IRQ}}$ pin (IRQE bit = 1)
2. Level-sensitive $\overline{\text{IRQ}}$ pin (IRQE bit = 0)
3. $t_{\overline{\text{STOP}}\text{DELAY}}$ = 4098 t_{cyc} if DLY bit = 1 or 2 t_{cyc} if DLY = 0.
4. $\overline{\text{XIRQ}}$ with X bit in CCR = 1.
5. $\overline{\text{IRQ}}$ or ($\overline{\text{XIRQ}}$ with X bit in CCR = 0)

Figure 19-7. Stop Recovery Timing Diagram