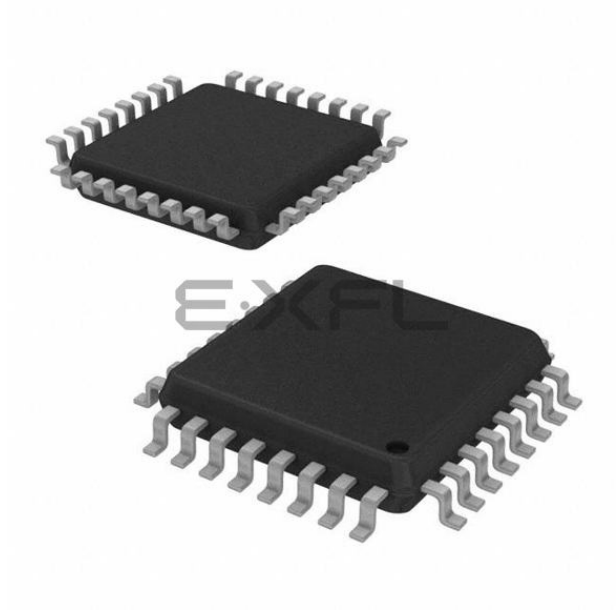

Welcome to E-XFL.COM

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	8051
Core Size	8-Bit
Speed	50MHz
Connectivity	SMBus (2-Wire/I²C), CANbus, LINbus, SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	25
Program Memory Size	96KB (96K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8.25K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.25V
Data Converters	A/D 25x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	32-LQFP
Supplier Device Package	32-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f586-iq

Table of Contents

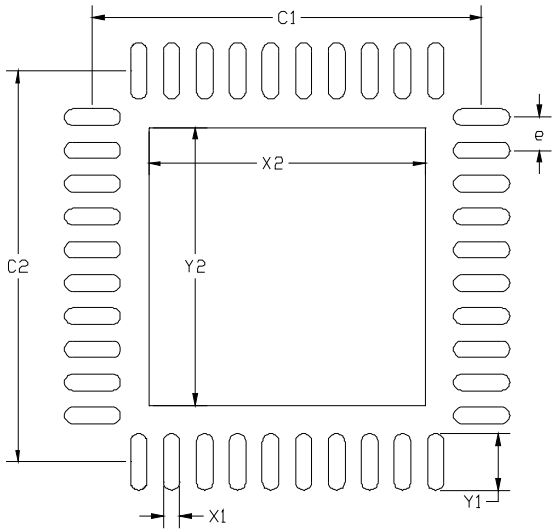

Figure 4.6. QFN-40 Landing Diagram

Table 4.6. QFN-40 Landing Diagram Dimensions

Dimension	Min	Max	Dimension	Min	Max
C1	5.80	5.90	X2	4.10	4.20
C2	5.80	5.90	Y1	0.75	0.85
e	0.50 BSC		Y2	4.10	4.20
X1	0.15	0.25			

Notes:

General

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimension and Tolerancing is per the ANSI Y14.5M-1994 specification.
3. This Land Pattern Design is based on the IPC-SM-7351 guidelines.
4. All dimensions shown are at Maximum Material Condition (MMC). Least Material Condition (LMC) is calculated based on a Fabrication Allowance of 0.05 mm.

Solder Mask Design

5. All metal pads are to be non-solder mask defined (NSMD). Clearance between the solder mask and the metal pad is to be 60 µm minimum, all the way around the pad.

Stencil Design

6. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
7. The stencil thickness should be 0.125 mm (5 mils).
8. The ratio of stencil aperture to land pad size should be 1:1 for all perimeter pads.
9. A 4x4 array of 0.80 mm square openings on a 1.05 mm pitch should be used for the center ground pad.

Card Assembly

10. A No-Clean, Type-3 solder paste is recommended.
11. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

SFR Definition 9.3. CPT1CN: Comparator1 Control

Bit	7	6	5	4	3	2	1	0
Name	CP1EN	CP1OUT	CP1RIF	CP1FIF	CP1HYP[1:0]		CP1HYN[1:0]	
Type	R/W	R	R/W	R/W	R/W		R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x9D; SFR Page = 0x00

Bit	Name	Function
7	CP1EN	**Comparator1 Enable Bit.** 0: Comparator1 Disabled. 1: Comparator1 Enabled.
6	CP1OUT	**Comparator1 Output State Flag.** 0: Voltage on CP1+ < CP1–. 1: Voltage on CP1+ > CP1–.
5	CP1RIF	**Comparator1 Rising-Edge Flag. Must be cleared by software.** 0: No Comparator1 Rising Edge has occurred since this flag was last cleared. 1: Comparator1 Rising Edge has occurred.
4	CP1FIF	**Comparator1 Falling-Edge Flag. Must be cleared by software.** 0: No Comparator1 Falling-Edge has occurred since this flag was last cleared. 1: Comparator1 Falling-Edge has occurred.
3:2	CP1HYP[1:0]	**Comparator1 Positive Hysteresis Control Bits.** 00: Positive Hysteresis Disabled. 01: Positive Hysteresis = 5 mV. 10: Positive Hysteresis = 10 mV. 11: Positive Hysteresis = 20 mV.
1:0	CP1HYN[1:0]	**Comparator1 Negative Hysteresis Control Bits.** 00: Negative Hysteresis Disabled. 01: Negative Hysteresis = 5 mV. 10: Negative Hysteresis = 10 mV. 11: Negative Hysteresis = 20 mV.

11. CIP-51 Microcontroller

The MCU system controller core is the CIP-51 microcontroller. The CIP-51 is fully compatible with the MCS-51™ instruction set; standard 803x/805x assemblers and compilers can be used to develop software. The MCU family has a superset of all the peripherals included with a standard 8051. The CIP-51 also includes on-chip debug hardware (see description in “C2 Interface” on page 351), and interfaces directly with the analog and digital subsystems providing a complete data acquisition or control-system solution in a single integrated circuit.

The CIP-51 Microcontroller core implements the standard 8051 organization and peripherals as well as additional custom peripherals and functions to extend its capability (see Figure 11.1 for a block diagram). The CIP-51 includes the following features:

- Fully Compatible with MCS-51 Instruction Set
- 50 MIPS Peak Throughput with 50 MHz Clock
- 0 to 50 MHz Clock Frequency
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- On-chip Debug Logic
- Program and Data Memory Security

11.1. Performance

The CIP-51 employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. In a standard 8051, all instructions except for MUL and DIV take 12 or 24 system clock cycles to execute, and usually have a maximum system clock of 12 MHz. By contrast, the CIP-51 core executes 70% of its instructions in one or two system clock cycles, with no instructions taking more than eight system clock cycles.

Table 13.3. Special Function Registers

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	Description	Page
ACC	0xE0	Accumulator	99
ADC0CF	0xBC	ADC0 Configuration	65
ADC0CN	0xE8	ADC0 Control	67
ADC0GTH	0xC4	ADC0 Greater-Than Compare High	69
ADC0GTL	0xC3	ADC0 Greater-Than Compare Low	69
ADC0H	0xBE	ADC0 High	66
ADC0L	0xBD	ADC0 Low	66
ADC0LTH	0xC6	ADC0 Less-Than Compare Word High	70
ADC0LTL	0xC5	ADC0 Less-Than Compare Word Low	70
ADC0MX	0xBB	ADC0 Mux Configuration	73
ADC0TK	0xBA	ADC0 Tracking Mode Select	68
B	0xF0	B Register	99
CCH0CN	0xE3	Cache Control	148
CKCON	0x8E	Clock Control	286
CLKMUL	0x97	Clock Multiplier	182
CLKSEL	0x8F	Clock Select	177
CPT0CN	0x9A	Comparator0 Control	79
CPT0MD	0x9B	Comparator0 Mode Selection	80
CPT0MX	0x9C	Comparator0 MUX Selection	86
CPT1CN	0x9D	Comparator1 Control	79
CPT1MD	0x9E	Comparator1 Mode Selection	80
CPT1MX	0x9F	Comparator1 MUX Selection	86
CPT2CN	0x9A	Comparator2 Control	83
CPT2MD	0x9B	Comparator2 Mode Selection	84
CPT2MX	0x9C	Comparator2 MUX Selection	88
DPH	0x83	Data Pointer High	98
DPL	0x82	Data Pointer Low	98
EIE1	0xE6	Extended Interrupt Enable 1	132
EIE2	0xE7	Extended Interrupt Enable 2	132
EIP1	0xF6	Extended Interrupt Priority 1	133
EIP2	0xF7	Extended Interrupt Priority 2	134
EMI0CF	0xB2	External Memory Interface Configuration	163
EMI0CN	0xAA	External Memory Interface Control	162
EMI0TC	0xAA	External Memory Interface Timing Control	168
FLKEY	0xB7	Flash Lock and Key	146

Table 13.3. Special Function Registers (Continued)

SFRs are listed in alphabetical order. All undefined SFR locations are reserved

Register	Address	Description	Page
PCA1CPL11	0xCE	PCA1 Capture 11 Low	350
PCA1CPM6	0xDA	PCA1 Module 6 Mode Register	348
PCA1CPM7	0xDB	PCA1 Module 7 Mode Register	348
PCA1CPM8	0xDC	PCA1 Module 8 Mode Register	348
PCA1CPM9	0xDD	PCA1 Module 9 Mode Register	348
PCA1CPM10	0xDE	PCA1 Module 10 Mode Register	348
PCA1CPM11	0xDF	PCA1 Module 11 Mode Register	348
PCA1H	0xFA	PCA1 Counter High	349
PCA1L	0xF9	PCA1 Counter Low	349
PCA1MD	0xD9	PCA1 Mode	346
PCA1PWM	0xDA	PCA1 PWM Configuration	347
PCON	0x87	Power Control	151
PSBANK	0xF5	Program Space Bank Select	104
PSCTL	0x8F	Program Store R/W Control	145
PSW	0xD0	Program Status Word	100
REF0CN	0xD1	Voltage Reference Control	76
REG0CN	0xD1	Voltage Regulator Control	90
RSTSRC	0xEF	Reset Source Configuration/Status	157
SBCON0	0xAB	UART0 Baud Rate Generator Control	263
SBRLH0	0xAD	UART0 Baud Rate Reload High Byte	264
SBRLL0	0xAC	UART0 Baud Rate Reload Low Byte	264
SBUF0	0x99	UART0 Data Buffer	263
SCON0	0x98	UART0 Control	261
SBUF1	0x99	UART1 Data Buffer	270
SCON1	0x98	UART1 Control	269
SFR0CN	0x84	SFR Page Control	113
SFRLAST	0x86	SFR Stack Last Page	116
SFRNEXT	0x85	SFR Stack Next Page	115
SFRPAGE	0xA7	SFR Page Select	114
SMB0CF	0xC1	SMBus0 Configuration	245
SMB0CN	0xC0	SMBus0 Control	247
SMB0DAT	0xC2	SMBus0 Data	249
SMOD0	0xA9	UART0 Mode	262
SN0	0xF9	Serial Number 0	101
SN1	0xFA	Serial Number 1	101
SN2	0xFB	Serial Number 2	101

Steps 5–7 must be repeated for each byte to be written. After Flash writes are complete, PSWE should be cleared so that MOVX instructions do not target program memory.

15.1.5. Flash Write Optimization

The Flash write procedure includes a block write option to optimize the time to perform consecutive byte writes. When block write is enabled by setting the CHBLKW bit (CCH0CN.0), writes to two consecutive bytes in Flash require the same amount of time as a single byte write. This is performed by caching the first byte that is written to Flash and then committing both bytes to Flash when the second byte is written. When block writes are enabled, if the second write does not occur, the first data byte written is not actually written to Flash. Flash bytes with block write enabled are programmed by software with the following sequence:

1. Disable interrupts (recommended).
2. If writing to an address in Banks 1, 2, or 3, set the COBANK[1:0] bits (register PSBANK) for the appropriate bank
3. Erase the 512-byte Flash page containing the target location, as described in Section 15.1.3.
4. Set the FLEWT bit (register FLSCL).
5. Set the CHBLKW bit (register CCH0CN).
6. Set the PSWE bit (register PSCTL).
7. Clear the PSEE bit (register PSCTL).
8. Write the first key code to FLKEY: 0xA5.
9. Write the second key code to FLKEY: 0xF1.
10. Using the MOVX instruction, write the first data byte to the desired location within the 512-byte sector.
11. Write the first key code to FLKEY: 0xA5.
12. Write the second key code to FLKEY: 0xF1.
13. Using the MOVX instruction, write the second data byte to the desired location within the 512-byte sector. The location of the second byte must be the next higher address from the first data byte.
14. Clear the PSWE bit.
15. Clear the CHBLKW bit.

18. External Data Memory Interface and On-Chip XRAM

For C8051F58x/F59x devices, 8 kB of RAM are included on-chip and mapped into the external data memory space (XRAM). Additionally, an External Memory Interface (EMIF) is available on the C8051F580/1/4/5 and C8051F588/9-F590/1 devices, which can be used to access off-chip data memories and memory-mapped devices connected to the GPIO ports. The external memory space may be accessed using the external move instruction (MOVX) and the data pointer (DPTR), or using the MOVX indirect addressing mode using R0 or R1. If the MOVX instruction is used with an 8-bit address operand (such as @R1), then the high byte of the 16-bit address is provided by the External Memory Interface Control Register (EMI0CN, shown in SFR Definition 18.1).

Note: The MOVX instruction can also be used for writing to the Flash memory. See Section "15. Flash Memory" on page 138 for details. The MOVX instruction accesses XRAM by default.

18.1. Accessing XRAM

The XRAM memory space is accessed using the MOVX instruction. The MOVX instruction has two forms, both of which use an indirect addressing method. The first method uses the Data Pointer, DPTR, a 16-bit register which contains the effective address of the XRAM location to be read from or written to. The second method uses R0 or R1 in combination with the EMI0CN register to generate the effective XRAM address. Examples of both of these methods are given below.

18.1.1. 16-Bit MOVX Example

The 16-bit form of the MOVX instruction accesses the memory location pointed to by the contents of the DPTR register. The following series of instructions reads the value of the byte at address 0x1234 into the accumulator A:

```
MOV    DPTR, #1234h        ; load DPTR with 16-bit address to read (0x1234)
MOVX   A, @DPTR            ; load contents of 0x1234 into accumulator A
```

The above example uses the 16-bit immediate MOV instruction to set the contents of DPTR. Alternately, the DPTR can be accessed through the SFR registers DPH, which contains the upper 8-bits of DPTR, and DPL, which contains the lower 8-bits of DPTR.

18.1.2. 8-Bit MOVX Example

The 8-bit form of the MOVX instruction uses the contents of the EMI0CN SFR to determine the upper 8-bits of the effective address to be accessed and the contents of R0 or R1 to determine the lower 8-bits of the effective address to be accessed. The following series of instructions read the contents of the byte at address 0x1234 into the accumulator A.

```
MOV    EMI0CN, #12h        ; load high byte of address into EMI0CN
MOV    R0, #34h            ; load low byte of address into R0 (or R1)
MOVX   a, @R0              ; load contents of 0x1234 into accumulator A
```

18.2. Configuring the External Memory Interface

Configuring the External Memory Interface consists of five steps:

1. Configure the Output Modes of the associated port pins as either push-pull or open-drain (push-pull is most common), and skip the associated pins in the crossbar.
2. Configure Port latches to “park” the EMIF pins in a dormant state (usually by setting them to logic 1).
3. Select Multiplexed mode or Non-multiplexed mode.
4. Select the memory mode (on-chip only, split mode without bank select, split mode with bank select, or off-chip only).
5. Set up timing to interface with off-chip memory or peripherals.

Each of these five steps is explained in detail in the following sections. The Port selection, Multiplexed mode selection, and Mode bits are located in the EMI0CF register shown in SFR Definition .

18.3. Port Configuration

The External Memory Interface appears on Ports 1, 2, 3, and 4 when it is used for off-chip memory access. When the EMIF is used, the Crossbar should be configured to skip over the /RD control line (P1.6) and the /WR control line (P1.7) using the P1SKIP register. When the EMIF is used in multiplexed mode, the Crossbar should also skip over the ALE control line (P1.5). For more information about configuring the Crossbar, see Section “20.6. Special Function Registers for Accessing and Configuring Port I/O” on page 204. The EMIF pinout is shown in Table 18.1 on page 160.

The External Memory Interface claims the associated Port pins for memory operations ONLY during the execution of an off-chip MOVX instruction. Once the MOVX instruction has completed, control of the Port pins reverts to the Port latches or to the Crossbar settings for those pins. See Section “20. Port Input/Output” on page 188 for more information about the Crossbar and Port operation and configuration. **The Port latches should be explicitly configured to “park” the External Memory Interface pins in a dormant state, most commonly by setting them to a logic 1**.

During the execution of the MOVX instruction, the External Memory Interface will explicitly disable the drivers on all Port pins that are acting as Inputs (Data[7:0] during a READ operation, for example). The Output mode of the Port pins (whether the pin is configured as Open-Drain or Push-Pull) is unaffected by the External Memory Interface operation, and remains controlled by the PnMDOUT registers. In most cases, the output modes of all EMIF pins should be configured for push-pull mode.

The C8051F580/1/4/5 devices support both the multiplexed and non-multiplexed modes and the C8051F588/9-F590/1 devices support only multiplexed modes. Accessing off-chip memory is not supported by the C8051F582/3/6/7 devices.

SFR Definition 19.6. OSCXCN: External Oscillator Control

Bit	7	6	5	4	3	2	1	0
Name	XTLVLD	XOSCMD[2:0]				XFCN[2:0]		
Type	R	R/W			R	R/W		
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x9F; SFR Page = 0x0F;

Bit	Name	Function
7	XTLVLD	**Crystal Oscillator Valid Flag.** (Read only when XOSCMD = 11x.) 0: Crystal Oscillator is unused or not yet stable. 1: Crystal Oscillator is running and stable.
6:4	XOSCMD[2:0]	**External Oscillator Mode Select.** 00x: External Oscillator circuit off. 010: External CMOS Clock Mode. 011: External CMOS Clock Mode with divide by 2 stage. 100: RC Oscillator Mode. 101: Capacitor Oscillator Mode. 110: Crystal Oscillator Mode. 111: Crystal Oscillator Mode with divide by 2 stage.
3	Unused	Read = 0b; Write =0b
2:0	XFCN[2:0]	**External Oscillator Frequency Control Bits.** Set according to the desired frequency for Crystal or RC mode. Set according to the desired K Factor for C mode.

XFCN	Crystal Mode	RC Mode	C Mode
000	f ≤ 32 kHz	f ≤ 25 kHz	K Factor = 0.87
001	32 kHz < f ≤ 84 kHz	25 kHz < f ≤ 50 kHz	K Factor = 2.6
010	84 kHz < f ≤ 225 kHz	50 kHz < f ≤ 100 kHz	K Factor = 7.7
011	225 kHz < f ≤ 590 kHz	100 kHz < f ≤ 200 kHz	K Factor = 22
100	590 kHz < f ≤ 1.5 MHz	200 kHz < f ≤ 400 kHz	K Factor = 65
101	1.5 MHz < f ≤ 4 MHz	400 kHz < f ≤ 800 kHz	K Factor = 180
110	4 MHz < f ≤ 10 MHz	800 kHz < f ≤ 1.6 MHz	K Factor = 664
111	10 MHz < f ≤ 30 MHz	1.6 MHz < f ≤ 3.2 MHz	K Factor = 1590

SFR Definition 20.3. XBR2: Port I/O Crossbar Register 2

Bit	7	6	5	4	3	2	1	0
Name	WEAKPUD	XBARE	Reserved		CP2AE	CP2E	URT1E	LIN0E
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC7; SFR Page = 0x0F

Bit	Name	Function
7	WEAKPUD	**Port I/O Weak Pullup Disable.** 0: Weak Pullups enabled (except for Ports whose I/O are configured for analog mode). 1: Weak Pullups disabled.
6	XBARE	**Crossbar Enable.** 0: Crossbar disabled. 1: Crossbar enabled.
5:4	Reserved	Always Write to 00b.
3	CP2AE	**Comparator2 Asynchronous Output Enable.** 0: Asynchronous CP2 unavailable at Port pin. 1: Asynchronous CP2 routed to Port pin.
2	CP2E	**Comparator2 Output Enable.** 0: CP2 unavailable at Port pin. 1: CP2 routed to Port pin.
1	URT1E	**UART1 I/O Output Enable.** 0: UART1 I/O unavailable at Port pin. 1: UART1 TX0, RX0 routed to Port pins.
0	LIN0E	**LIN I/O Output Enable.** 0: LIN I/O unavailable at Port pin. 1: LIN_TX, LIN_RX routed to Port pins.

SFR Definition 20.18. P1MDIN: Port 1 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	P1MDIN[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xF2; SFR Page = 0x0F

Bit	Name	Function
7:0	P1MDIN[7:0]	**Analog Configuration Bits for P1.7–P1.0 (respectively).** Port pins configured for analog mode have their weak pull-up and digital receiver disabled. For analog mode, the pin also needs to be configured for open-drain mode in the P1MDOUT register. 0: Corresponding P1.n pin is configured for analog mode. 1: Corresponding P1.n pin is not configured for analog mode.

SFR Definition 20.19. P1MDOUT: Port 1 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	P1MDOUT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xA5; SFR Page = 0x0F

Bit	Name	Function
7:0	P1MDOUT[7:0]	**Output Configuration Bits for P1.7–P1.0 (respectively).** These bits are ignored if the corresponding bit in register P1MDIN is logic 0. 0: Corresponding P1.n Output is open-drain. 1: Corresponding P1.n Output is push-pull.

22. Controller Area Network (CAN0)

Important Documentation Note: The Bosch CAN Controller is integrated in the C8051F580/2/4/6/8-F590 devices. This section of the data sheet gives a description of the CAN controller as an overview and offers a description of how the Silicon Labs CIP-51 MCU interfaces with the on-chip Bosch CAN controller. In order to use the CAN controller, refer to Bosch's C_CAN User's Manual as an accompanying manual to the Silicon Labs' data sheet.

The C8051F580/2/4/6/8-F590 devices feature a Control Area Network (CAN) controller that enables serial communication using the CAN protocol. Silicon Labs CAN facilitates communication on a CAN network in accordance with the Bosch specification 2.0A (basic CAN) and 2.0B (full CAN). The CAN controller consists of a CAN Core, Message RAM (separate from the CIP-51 RAM), a message handler state machine, and control registers. Silicon Labs CAN is a protocol controller and does not provide physical layer drivers (i.e., transceivers). Figure 22.1 shows an example typical configuration on a CAN bus.

Silicon Labs CAN operates at bit rates of up to 1 Mbit/second, though this can be limited by the physical layer chosen to transmit data on the CAN bus. The CAN processor has 32 Message Objects that can be configured to transmit or receive data. Incoming data, message objects and their identifier masks are stored in the CAN message RAM. All protocol functions for transmission of data and acceptance filtering is performed by the CAN controller and not by the CIP-51 MCU. In this way, minimal CPU bandwidth is needed to use CAN communication. The CIP-51 configures the CAN controller, accesses received data, and passes data for transmission via Special Function Registers (SFRs) in the CIP-51.

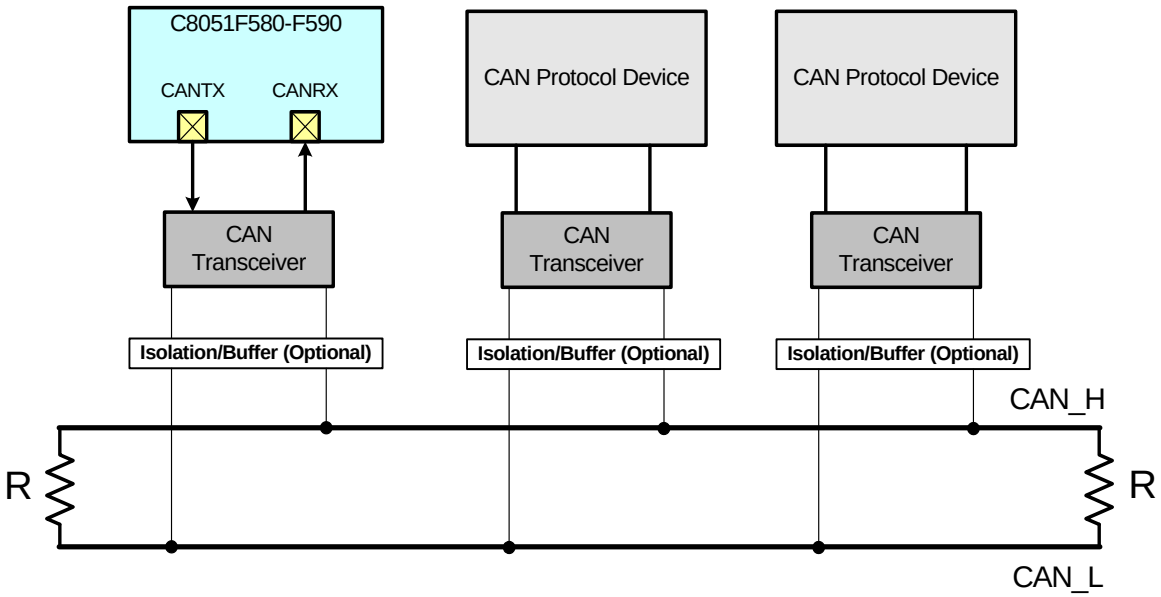

Figure 22.1. Typical CAN Bus Configuration

22.2. CAN Registers

CAN registers are classified as follows:

1. **CAN Controller Protocol Registers**: CAN control, interrupt, error control, bus status, test modes.
2. **Message Object Interface Registers**: Used to configure 32 Message Objects, send and receive data to and from Message Objects. The CIP-51 MCU accesses the CAN message RAM via the Message Object Interface Registers. Upon writing a message object number to an IF1 or IF2 Command Request Register, the contents of the associated Interface Registers (IF1 or IF2) will be transferred to or from the message object in CAN RAM.
3. **Message Handler Registers**: These read only registers are used to provide information to the CIP-51 MCU about the message objects (MSGVLD flags, Transmission Request Pending, New Data Flags) and Interrupts Pending (which Message Objects have caused an interrupt or status interrupt condition).

For the registers other than CAN0CFG, refer to the Bosch CAN User's Guide for information on the function and use of the CAN Control Protocol Registers.

22.2.1. CAN Controller Protocol Registers

The CAN Control Protocol Registers are used to configure the CAN controller, process interrupts, monitor bus status, and place the controller in test modes.

The registers are: CAN Control Register (CAN0CN), CAN Clock Configuration (CAN0CFG), CAN Status Register (CAN0STA), CAN Test Register (CAN0TST), Error Counter Register, Bit Timing Register, and the Baud Rate Prescaler (BRP) Extension Register.

22.2.2. Message Object Interface Registers

There are two sets of Message Object Interface Registers used to configure the 32 Message Objects that transmit and receive data to and from the CAN bus. Message objects can be configured for transmit or receive, and are assigned arbitration message identifiers for acceptance filtering by all CAN nodes.

Message Objects are stored in Message RAM, and are accessed and configured using the Message Object Interface Registers.

22.2.3. Message Handler Registers

The Message Handler Registers are read only registers. The message handler registers provide interrupt, error, transmit/receive requests, and new data information.

SFR Definition 22.1. CAN0CFG: CAN Clock Configuration

Bit	7	6	5	4	3	2	1	0
Name	Unused	Unused	Unused	Unused	Unused	Unused	SYSDIV[1:0]	
Type	R	R	R	R	R	R	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0x92; SFR Page = 0x0C

Bit	Name	Function
7:2	Unused	Read = 000000b; Write = Don't Care.
1:0	SYSDIV[1:0]	**CAN System Clock Divider Bits.** The CAN controller clock is derived from the CIP-51 system clock. The CAN controller clock must be less than or equal to 25 MHz. 00: CAN controller clock = System Clock/1. 01: CAN controller clock = System Clock/2. 10: CAN controller clock = System Clock/4. 11: CAN controller clock = System Clock/8.

24. UART0

UART0 is an asynchronous, full duplex serial port offering a variety of data formatting options. A dedicated baud rate generator with a 16-bit timer and selectable prescaler is included, which can generate a wide range of baud rates (details in Section “24.1. Baud Rate Generator” on page 256). A received data FIFO allows UART0 to receive up to three data bytes before data is lost and an overflow occurs.

UART0 has six associated SFRs. Three are used for the Baud Rate Generator (SBCON0, SBRLH0, and SBRLL0), two are used for data formatting, control, and status functions (SCON0, SMOD0), and one is used to send and receive data (SBUF0). The single SBUF0 location provides access to both transmit and receive registers. **Writes to SBUF0 always access the Transmit register. Reads of SBUF0 always access the buffered Receive register; it is not possible to read data from the Transmit register.**

With UART0 interrupts enabled, an interrupt is generated each time a transmit is completed (TI0 is set in SCON0), or a data byte has been received (RI0 is set in SCON0). The UART0 interrupt flags are not cleared by hardware when the CPU vectors to the interrupt service routine. They must be cleared manually by software, allowing software to determine the cause of the UART0 interrupt (transmit complete or receive complete). If additional bytes are available in the Receive FIFO, the RI0 bit cannot be cleared by software.

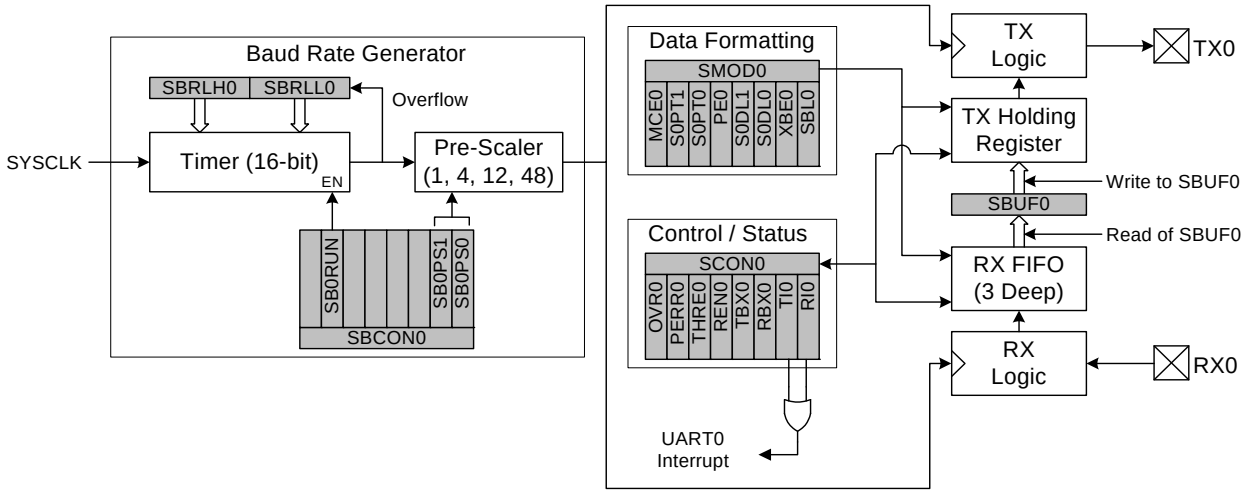

Figure 24.1. UART0 Block Diagram

24.1. Baud Rate Generator

The UART0 baud rate is generated by a dedicated 16-bit timer which runs from the controller’s core clock (SYSCLK) and has prescaler options of 1, 4, 12, or 48. The timer and prescaler options combined allow for a wide selection of baud rates over many clock frequencies.

The baud rate generator is configured using three registers: SBCON0, SBRLH0, and SBRLL0. The UART0 Baud Rate Generator Control Register (SBCON0, SFR Definition 24.4) enables or disables the baud rate generator, selects the clock source for the baud rate generator, and selects the prescaler value for the timer. The baud rate generator must be enabled for UART0 to function. Registers SBRLH0 and SBRLL0 contain a 16-bit reload value for the dedicated 16-bit timer. The internal timer counts up from the reload value on every clock tick. On timer overflows (0xFFFF to 0x0000), the timer is reloaded. The baud rate for UART0 is defined in Equation 24.1, where “BRG Clock” is the baud rate generator’s selected clock source. For reliable UART operation, it is recommended that the UART baud rate is not configured for baud rates faster than SYSCLK/16.

27.3. Timer 3

Timer 3 is a 16-bit timer formed by two 8-bit SFRs: TMR3L (low byte) and TMR3H (high byte). Timer 3 may operate in 16-bit auto-reload mode or (split) 8-bit auto-reload mode. The T3SPLIT bit (TMR3CN.3) defines the Timer 3 operation mode.

Timer 3 may be clocked by the system clock, the system clock divided by 12, or the external oscillator source divided by 8. The external clock mode is ideal for real-time clock (RTC) functionality, where the internal oscillator drives the system clock while Timer 3 (and/or the PCA) is clocked by an external precision oscillator. Note that the external oscillator source divided by 8 is synchronized with the system clock.

27.3.1. 16-bit Timer with Auto-Reload

When T3SPLIT (TMR3CN.3) is zero, Timer 3 operates as a 16-bit timer with auto-reload. Timer 3 can be clocked by SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. As the 16-bit timer register increments and overflows from 0xFFFF to 0x0000, the 16-bit value in the Timer 3 reload registers (TMR3RLH and TMR3RLL) is loaded into the Timer 3 register as shown in Figure 27.7, and the Timer 3 High Byte Overflow Flag (TMR3CN.7) is set. If Timer 3 interrupts are enabled, an interrupt will be generated on each Timer 3 overflow. Additionally, if Timer 3 interrupts are enabled and the TF3LEN bit is set (TMR3CN.5), an interrupt will be generated each time the lower 8 bits (TMR3L) overflow from 0xFF to 0x00.

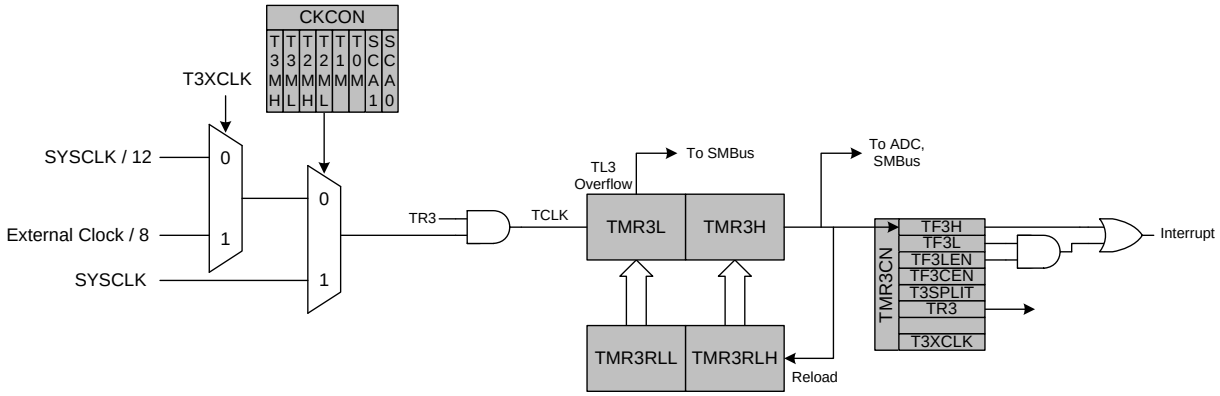

Figure 27.7. Timer 3 16-Bit Mode Block Diagram

27.3.2. 8-bit Timers with Auto-Reload

When T3SPLIT is set, Timer 3 operates as two 8-bit timers (TMR3H and TMR3L). Both 8-bit timers operate in auto-reload mode as shown in Figure 27.8. TMR3RLL holds the reload value for TMR3L; TMR3RLH holds the reload value for TMR3H. The TR3 bit in TMR3CN handles the run control for TMR3H. TMR3L is always running when configured for 8-bit Mode.

Each 8-bit timer may be configured to use SYSCLK, SYSCLK divided by 12, or the external oscillator clock source divided by 8. The Timer 3 Clock Select bits (T3MH and T3ML in CKCON) select either SYSCLK or the clock defined by the Timer 3 External Clock Select bit (T3XCLK in TMR3CN), as follows:

T3MH	T3XCLK	TMR3H Clock Source
0	0	SYSCLK/12
0	1	External Clock/8
1	X	SYSCLK

T3ML	T3XCLK	TMR3L Clock Source
0	0	SYSCLK/12
0	1	External Clock/8
1	X	SYSCLK

SFR Definition 28.4. PCA0CPMn: PCA0 Capture/Compare Mode

Bit	7	6	5	4	3	2	1	0
Name	PWM16n	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Addresses: PCA0CPM0 = 0xDA, PCA0CPM1 = 0xDB, PCA0CPM2 = 0xDC; PCA0CPM3 = 0xDD, PCA0CPM4 = 0xDE, PCA0CPM5 = 0xDF, SFR Page (all registers) = 0x00

Bit	Name	Function
7	PWM16n	**16-bit Pulse Width Modulation Enable.** This bit enables 16-bit mode when Pulse Width Modulation mode is enabled. 0: 8 to 11-bit PWM selected. 1: 16-bit PWM selected.
6	ECOMn	**Comparator Function Enable.** This bit enables the comparator function for PCA0 module n when set to 1.
5	CAPPn	**Capture Positive Function Enable.** This bit enables the positive edge capture for PCA0 module n when set to 1.
4	CAPNn	**Capture Negative Function Enable.** This bit enables the negative edge capture for PCA0 module n when set to 1.
3	MATn	**Match Function Enable.** This bit enables the match function for PCA0 module n when set to 1. When enabled, matches of the PCA0 counter with a module's capture/compare register cause the CCFn bit in PCA0MD register to be set to logic 1.
2	TOGn	**Toggle Function Enable.** This bit enables the toggle function for PCA0 module n when set to 1. When enabled, matches of the PCA0 counter with a module's capture/compare register cause the logic level on the CEXn pin to toggle. If the PWMn bit is also set to logic 1, the module operates in Frequency Output Mode.
1	PWMn	**Pulse Width Modulation Mode Enable.** This bit enables the PWM function for PCA0 module n when set to 1. When enabled, a pulse width modulated signal is output on the CEXn pin. 8 to 11-bit PWM is used if PWM16n is cleared; 16-bit mode is used if PWM16n is set to logic 1. If the TOGn bit is also set, the module operates in Frequency Output Mode.
0	ECCFn	**Capture/Compare Flag Interrupt Enable.** This bit sets the masking of the Capture/Compare Flag (CCFn) interrupt. 0: Disable CCFn interrupts. 1: Enable a Capture/Compare Flag interrupt request when CCFn is set.

Note: When the WDTE bit is set to 1, the PCA0CPM5 register cannot be modified, and module 5 acts as the watchdog timer. To change the contents of the PCA0CPM5 register or the function of module 5, the Watchdog Timer must be disabled.

SFR Definition 29.2. PCA1MD: PCA1 Mode

Bit	7	6	5	4	3	2	1	0
Name	CIDL1				CPS12	CPS11	CPS10	ECF1
Type	R/W	R	R/W	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xD9; SFR Page = 0x10

Bit	Name	Function
7	CIDL1	**PCA1 Counter/Timer Idle Control.** Specifies PCA1 behavior when CPU is in Idle Mode. 0: PCA1 continues to function normally while the system controller is in Idle Mode. 1: PCA1 operation is suspended while the system controller is in Idle Mode.
6:4	Unused	Read = 000b, Write = Don't care.
3:1	CPS1[2:0]	**PCA1 Counter/Timer Pulse Select.** These bits select the timebase source for the PCA1 counter 000: System clock divided by 12 001: System clock divided by 4 010: Timer 0 overflow 011: High-to-low transitions on ECI (max rate = system clock divided by 4) 100: System clock 101: External clock divided by 8 (synchronized with the system clock) 110: Timer 4 overflow 111: Timer 5 overflow
0	EC1F	**PCA1 Counter/Timer Overflow Interrupt Enable.** This bit sets the masking of the PCA1 Counter/Timer Overflow (CF1) interrupt. 0: Disable the CF1 interrupt. 1: Enable a PCA1 Counter/Timer Overflow interrupt request when CF1 (PCA1CN.7) is set.