



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Obsolete	
Core Processor	8051	
Core Size	8-Bit	
Speed	25MHz	
Connectivity	SMBus (2-Wire/I ² C), SPI, UART/USART	
Peripherals	POR, PWM, Temp Sensor, WDT	
Number of I/O	17	
Program Memory Size	8KB (8K x 8)	
Program Memory Type	FLASH	
EEPROM Size	-	
RAM Size	768 x 8	
Voltage - Supply (Vcc/Vdd)	cc/Vdd) 2.7V ~ 3.6V	
Data Converters	A/D 16x10b; D/A 1x10b	
Oscillator Type	Internal	
Operating Temperature	-40°C ~ 85°C (TA)	
Mounting Type	Through Hole	
Package / Case	20-DIP (0.300", 7.62mm)	
Supplier Device Package	20-DIP	
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f330-gp	

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

List of Figures

1.	System Overview	
	Figure 1.1. C8051F330 Block Diagram	19
	Figure 1.2. C8051F331 Block Diagram	19
	Figure 1.3. C8051F332 Block Diagram	20
	Figure 1.4. C8051F333 Block Diagram	20
	Figure 1.5. C8051F334 Block Diagram	21
	Figure 1.6. C8051F335 Block Diagram	21
	Figure 1.7. Comparison of Peak MCU Execution Speeds	22
	Figure 1.8. On-Chip Clock and Reset	23
	Figure 1.9. On-Board Memory Map	24
	Figure 1.10. Development/In-System Debug Diagram	25
	Figure 1.11. Digital Crossbar Diagram	26
	Figure 1.12. PCA Block Diagram	27
	Figure 1.13. PCA Block Diagram	27
	Figure 1.14. 10-Bit ADC Block Diagram	28
	Figure 1.15. Comparator0 Block Diagram	29
	Figure 1.16. IDA0 Functional Block Diagram	30
2.	Absolute Maximum Ratings	
3.	Global Electrical Characteristics	
4.	Pinout and Package Definitions	
	Figure 4.1. QFN-20 Pinout Diagram (Top View)	37
	Figure 4.2. PDIP-20 Pinout Diagram (Top View)	40
	Figure 4.3. PDIP-20 Package Drawing	41
5.	10-Bit ADC (ADC0, C8051F330/2/4 only)	
	Figure 5.1. ADC0 Functional Block Diagram	43
	Figure 5.2. Typical Temperature Sensor Transfer Function	45
	Figure 5.3. 10-Bit ADC Track and Conversion Example Timing	47
	Figure 5.4. ADC0 Equivalent Input Circuits	48
	Figure 5.5. ADC Window Compare Example: Right-Justified Single-Ended Data	55
	Figure 5.6. ADC Window Compare Example: Left-Justified Single-Ended Data	55
	Figure 5.7. ADC Window Compare Example: Right-Justified Differential Data	56
_	Figure 5.8. ADC Window Compare Example: Left-Justified Differential Data	56
6.	10-Bit Current Mode DAC (IDA0, C8051F330 only)	
	Figure 6.1. IDA0 Functional Block Diagram	59
_	Figure 6.2. IDA0 Data Word Mapping	60
7.	Voltage Reference (C8051F330/2/4 only)	~~
_	Figure 7.1. Voltage Reference Functional Block Diagram	63
8.	Comparator0	~-
	Figure 8.1. Comparator0 Functional Block Diagram	67
•	Figure 8.2. Comparator Hysteresis Plot	68
9.	CIP-51 Microcontroller	
	Figure 9.1. CIP-51 Block Diagram	/3
	Figure 9.2. Memory Map	79

C8051F330/1/2/3/4/5

	cillator
Table 16.2.	Timer Settings for Standard Baud Rates Using an External 25.0 MHz Os-
	cillator 162
Table 16.3.	Timer Settings for Standard Baud Rates Using an External 22.1184 MHz
	Oscillator 163
Table 16.4.	Timer Settings for Standard Baud Rates Using an External 18.432 MHz
	Oscillator 163
Table 16.5.	Timer Settings for Standard Baud Rates Using an External 11.0592 MHz
	Oscillator 164
Table 16.6.	Timer Settings for Standard Baud Rates Using an External 3.6864 MHz
	Oscillator 164
17. Enhanced	Serial Peripheral Interface (SPI0)
Table 17.1.	SPI Slave Timing Parameters 177
18. Timers	
19. Programm	able Counter Array
Table 19.1.	PCA Timebase Input Options 196
Table 19.2.	PCA0CPM Register Settings for PCA Capture/Compare Modules 197
Table 19.3.	Watchdog Timer Timeout Intervals1 206

5. 10-Bit ADC (ADC0, C8051F330/2/4 only)

The ADC0 subsystem for the C8051F330/2/4 consists of two analog multiplexers (referred to collectively as AMUX0) with 16 total input selections, and a 200 ksps, 10-bit successive-approximation-register ADC with integrated track-and-hold and programmable window detector. The AMUX0, data conversion modes, and window detector are all configurable under software control via the Special Function Registers shown in Figure 5.1. ADC0 operates in both Single-ended and Differential modes, and may be configured to measure Ports0-1, the Temperature Sensor output, or V_{DD} with respect to Ports0-1 or GND. The ADC0 subsystem is enabled only when the AD0EN bit in the ADC0 Control register (ADC0CN) is set to logic 1. The ADC0 subsystem is in low power shutdown when this bit is logic 0.





5.1. Analog Multiplexer

AMUX0 selects the positive and negative inputs to the ADC. Any of the following may be selected as the positive input: Ports0-1, the on-chip temperature sensor, or the positive power supply (V_{DD}). Any of the following may be selected as the negative input: Ports0-1, V_{REF} , or GND. When GND is selected as the negative input, ADC0 operates in Single-ended Mode; all other times, ADC0 operates in Differential Mode. The ADC0 input channels are selected in the AMX0P and AMX0N registers as described in SFR Definition 5.1 and SFR Definition 5.2.

The conversion code format differs between Single-ended and Differential modes. The registers ADC0H and ADC0L contain the high and low bytes of the output conversion code from the ADC at the completion of each conversion. Data can be right-justified or left-justified, depending on the setting of the AD0LJST. When in Single-ended Mode, conversion codes are represented as 10-bit unsigned integers. Inputs are

8. Comparator0

C8051F330/1/2/3/4/5 devices include an on-chip programmable voltage comparator, Comparator0, shown in Figure 8.1.

The Comparator offers programmable response time and hysteresis, an analog input multiplexer, and two outputs that are optionally available at the Port pins: a synchronous "latched" output (CP0), or an asynchronous "raw" output (CP0A). The asynchronous CP0A signal is available even when in when the system clock is not active. This allows the Comparator to operate and generate an output with the device in STOP mode. When assigned to a Port pin, the Comparator output may be configured as open drain or push-pull (see Section "14.2. Port I/O Initialization" on page 129). Comparator0 may also be used as a reset source (see Section "10.5. Comparator0 Reset" on page 102).

The Comparator0 inputs are selected in the CPT0MX register (SFR Definition 8.2). The CMX0P1– CMX0P0 bits select the Comparator0 positive input; the CMX0N1–CMX0N0 bits select the Comparator0 negative input. **Important Note About Comparator Inputs:** The Port pins selected as comparator inputs should be configured as analog inputs in their associated Port configuration register, and configured to be skipped by the Crossbar (for details on Port configuration, see **Section "14.3. General Purpose Port I/O" on page 131**).



Figure 8.1. Comparator0 Functional Block Diagram

9. 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. Included are four 16-bit counter/timers (see description in **Section 18**), an enhanced full-duplex UART (see description in **Section 16**), an Enhanced SPI (see description in **Section 17**), 256 bytes of internal RAM, 128 byte Special Function Register (SFR) address space (**Section 9.2.6**), and 17 Port I/O (see description in **Section 14**). The CIP-51 also includes on-chip debug hardware (see description in **Section 20**), 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 c

9.3.2. External Interrupts

The /INT0 and /INT1 external interrupt sources are configurable as active high or low, edge or level sensitive. The IN0PL (/INT0 Polarity) and IN1PL (/INT1 Polarity) bits in the IT01CF register select active high or active low; the IT0 and IT1 bits in TCON (**Section "18.1. Timer 0 and Timer 1" on page 179**) select level or edge sensitive. The table below lists the possible configurations.

	IT0	IN0PL	/INT0 Interrupt	
	1	0	Active low, edge sensitive	
	1	1	Active high, edge sensitive	
	0	0	Active low, level sensitive	
	0	1	Active high, level sensitive	

IT1	IN1PL	/INT1 Interrupt
1	0	Active low, edge sensitive
1	1	Active high, edge sensitive
0	0	Active low, level sensitive
0	1	Active high, level sensitive

/INT0 and /INT1 are assigned to Port pins as defined in the IT01CF register (see SFR Definition 9.11). Note that /INT0 and /INT0 Port pin assignments are independent of any Crossbar assignments. /INT0 and /INT1 will monitor their assigned Port pins without disturbing the peripheral that was assigned the Port pin via the Crossbar. To assign a Port pin only to /INT0 and/or /INT1, configure the Crossbar to skip the selected pin(s). This is accomplished by setting the associated bit in register XBR0 (see **Section** "**14.1. Priority Crossbar Decoder**" on page **127** for complete details on configuring the Crossbar).

IE0 (TCON.1) and IE1 (TCON.3) serve as the interrupt-pending flags for the /INT0 and /INT1 external interrupts, respectively. If an /INT0 or /INT1 external interrupt is configured as edge-sensitive, the corresponding interrupt-pending flag is automatically cleared by the hardware when the CPU vectors to the ISR. When configured as level sensitive, the interrupt-pending flag remains logic 1 while the input is active as defined by the corresponding polarity bit (INOPL or IN1PL); the flag remains logic 0 while the input is inactive. The external interrupt source must hold the input active until the interrupt request is recognized. It must then deactivate the interrupt request before execution of the ISR completes or another interrupt request will be generated.

9.3.3. Interrupt Priorities

Each interrupt source can be individually programmed to one of two priority levels: low or high. A low priority interrupt service routine can be preempted by a high priority interrupt. A high priority interrupt cannot be preempted. Each interrupt has an associated interrupt priority bit in an SFR (IP or EIP1) used to configure its priority level. Low priority is the default. If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate, given in Table 9.4.

9.3.4. Interrupt Latency

Interrupt response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 5 system clock cycles: 1 clock cycle to detect the interrupt and 4 clock cycles to complete the LCALL to the ISR. If an interrupt is pending when a RETI is executed, a single instruction is executed before an LCALL is made to service the pending interrupt. Therefore, the maximum response time for an interrupt (when no other interrupt is currently being serviced or the new interrupt is of greater priority) occurs when the CPU is performing an RETI instruction followed by a DIV as the next instruction. In this case, the response time is 18 system clock cycles: 1 clock cycle to detect the interrupt, 5 clock cycles to execute the RETI, 8 clock cycles to complete the DIV instruction and 4 clock cycles to execute the LCALL to the ISR. If the CPU is executing an ISR for an interrupt with equal or higher priority, the new interrupt will not be serviced until the current ISR completes, including the RETI and following instruction.

20.2. C2 Pin Sharing

The C2 protocol allows the C2 pins to be shared with user functions so that in-system debugging and Flash programming may be performed. This is possible because C2 communication is typically performed when the device is in the halt state, where all on-chip peripherals and user software are stalled. In this halted state, the C2 interface can safely 'borrow' the C2CK (RST) and C2D (P2.0) pins. In most applications, external resistors are required to isolate C2 interface traffic from the user application. A typical isolation configuration is shown in Figure 20.1.



Figure 20.1. Typical C2 Pin Sharing

The configuration in Figure 20.1 assumes the following:

- 1. The <u>user input</u> (b) cannot change state while the target device is halted.
- 2. The \overline{RST} pin on the target device is used as an input only.

Additional resistors may be necessary depending on the specific application.