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	25MHz
Connectivity	LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	16
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.25V
Data Converters	A/D 16x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	20-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	20-TSSOP
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f530-c-itr

1.6. Programmable Comparator

C8051F52x/F52xA/F53x/F53xA devices include a software-configurable voltage comparator with an input multiplexer. The comparator offers programmable response time and hysteresis and an output that is optionally available at the Port pins: a synchronous “latched” output (CP0). The comparator interrupt may be generated on rising, falling, or both edges. When in IDLE or SUSPEND mode, these interrupts may be used as a “wake-up” source for the processor. The Comparator may also be configured as a reset source. A block diagram of the comparator is shown in Figure 1.8.

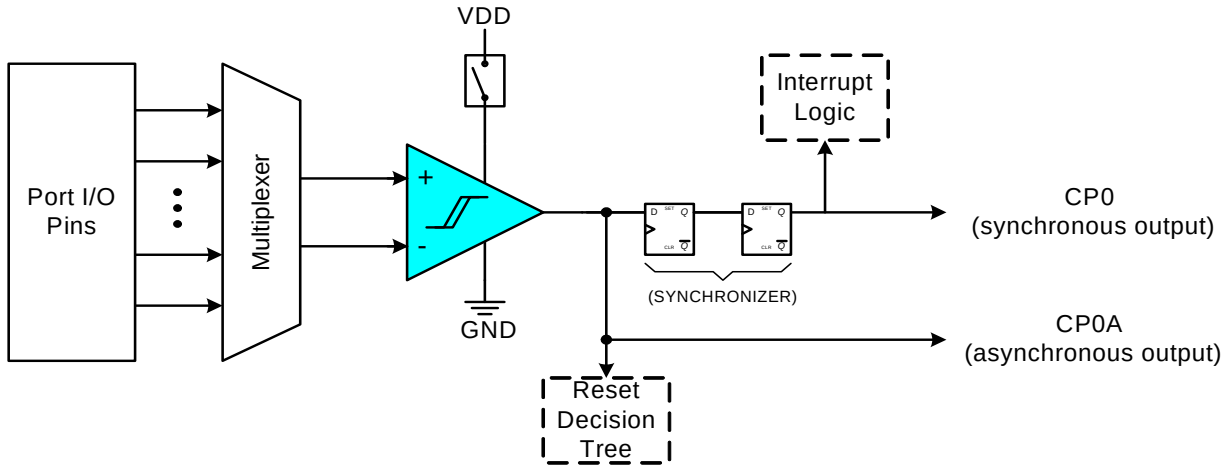

Figure 1.8. Comparator Block Diagram

1.7. Voltage Regulator

C8051F52x/F52xA/F53x/F53xA devices include an on-chip low dropout voltage regulator (REG0). The input to REG0 at the V_{REGIN} pin can be as high as 5.25 V. The output can be selected by software to 2.1 or 2.6 V. When enabled, the output of REG0 powers the device and drives the V_{DD} pin. The voltage regulator can be used to power external devices connected to V_{DD}.

1.8. Serial Port

The C8051F52x/F52xA/F53x/F53xA family includes a full-duplex UART with enhanced baud rate configuration, and an Enhanced SPI interface. Each of the serial buses is fully implemented in hardware and makes extensive use of the CIP-51's interrupts, thus requiring very little CPU intervention.

2.2. Electrical Characteristics

Table 2.2. Global DC Electrical Characteristics

–40 to +125 °C, 25 MHz System Clock unless otherwise specified. Typical values are given at 25 °C

Parameter	Conditions	Min	Typ	Max	Units
Supply Input Voltage (V_{REGIN})[1]	Output Current ≤ 1 mA				
	C8051F52x/53x	2.7	—	5.25	V
	C8051F52xA/53xA	1.8[1]	—	5.25	V
	C8051F52x-C/53x-C	2.0[1]	—	5.25	V
Digital Supply Voltage (V_{DD})	C8051F52x/53x	2.0	—	2.7	V
	C8051F52xA/53xA	1.8	—	2.7	V
	C8051F52x-C/53x-C	2.0	—	2.75	V
Core Supply RAM Data Retention Voltage		—	1.5	—	V
SYSCLK (System Clock)[2]		0	—	25	MHz
Specified Operating Temperature Range		–40	—	+125	°C
Digital Supply Current—CPU Active (Normal Mode, fetching instructions from Flash)					
I_{DD}[3,4]	**V_{DD} = 2.1 V:**				
	Clock = 32 kHz	—	13	—	µA
	Clock = 200 kHz	—	60	—	µA
	Clock = 1 MHz	—	0.28	—	mA
	Clock = 25 MHz	—	5.1	9	mA
	V_{DD} = 2.6 V:				
	Clock = 32 kHz	—	22	—	µA
	Clock = 200 kHz	—	105	—	µA
	Clock = 1 MHz	—	0.5	—	mA
	Clock = 25 MHz	—	7.3	13	mA
I_{DD} Frequency Sensitivity[3,5]	T = 25 °C:				
	V_{DD} = 2.1 V, F ≤ 12 MHz	—	0.276	—	mA/MHz
	V_{DD} = 2.1 V, F > 12 MHz	—	0.140	—	mA/MHz
	V_{DD} = 2.6 V, F ≤ 12 MHz	—	0.424	—	mA/MHz
	V_{DD} = 2.6 V, F > 12 MHz	—	0.184	—	mA/MHz

Notes:

1. For more information on V_{REGIN} characteristics, see Table 2.6 on page 30.
2. SYSCLK must be at least 32 kHz to enable debugging.
3. Based on device characterization data; Not production tested.
4. Does not include internal oscillator or internal regulator supply current.
5. I_{DD} can be estimated for frequencies <= 12 MHz by multiplying the frequency of interest by the frequency sensitivity number for that range. When using these numbers to estimate I_{DD} > 12 MHz, the estimate should be the current at 25 MHz minus the difference in current indicated by the frequency sensitivity number. For example: V_{DD} = 2.6 V; F= 20 MHz, I_{DD} = 7.3 mA – (25 MHz – 20 MHz) x 0.184 mA/MHz = 6.38 mA.
6. Idle I_{DD} can be estimated for frequencies <= 1 MHz by multiplying the frequency of interest by the frequency sensitivity number for that range. When using these numbers to estimate I_{DD} > 1 MHz, the estimate should be the current at 25 MHz minus the difference in current indicated by the frequency sensitivity number. For example: V_{DD} = 2.6 V; F= 5 MHz, Idle I_{DD} = 3 mA – (25 MHz– 5 MHz) x 118 µA/MHz = 0.64 mA.

Table 3.8. QFN-20 Package Diagram Dimensions

Dimension	MIN	NOM	MAX
A	0.80	0.90	1.00
A1	0.00	0.02	0.05
b	0.18	0.25	0.30
D	4.00 BSC.		
D2	2.55	2.70	2.85
e	0.50 BSC.		
E	4.00 BSC.		
E2	2.55	2.70	2.85
L	0.30	0.40	0.50
L1	0.00	—	0.15
aaa	—	—	0.15
bbb	—	—	0.10
ddd	—	—	0.05
eee	—	—	0.08
Z	—	0.43	—
Y	—	0.18	—

Notes:

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimensioning and Tolerancing per ANSI Y14.5M-1994.
3. This drawing conforms to JEDEC outline MO-220, variation VGGD except for custom features D2, E2, Z, Y, L, and L1, which are toleranced per supplier designation.
4. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

4.4.1. Calculating the Gain Value

The ADC0 selectable gain feature is controlled by 13 bits in three registers. ADC0GNH contains the 8 upper bits of the gain value and ADC0GNL contains the 4 lower bits of the gain value. The final GAINADD bit (ADC0GNA.0) controls an optional extra 1/64 (0.016) of gain that can be added in addition to the ADC0GNH and ADC0GNL gain. The ADC0GNA.0 bit is set to 1 after a power-on reset.

The equivalent gain for the ADC0GNH, ADC0GNL and ADC0GNA registers is:

$$gain = \left(\frac{GAIN}{4096}\right) + GAINADD \times \left(\frac{1}{64}\right)$$

Equation 4.2. Equivalent Gain from the ADC0GNH and ADC0GNL Registers

Where:
GAIN is the 12-bit word of ADC0GNH[7:0] and ADC0GNL[7:4]
GAINADD is the value of the GAINADD bit (ADC0GNA.0)
gain is the equivalent gain value from 0 to 1.016

For example, if ADC0GNH = 0xFC, ADC0GNL = 0x00, and GAINADD = '1', GAIN = 0xFC0 = 4032, and the resulting equation is:

$$gain = \left(\frac{4032}{4096}\right) + 1 \times \left(\frac{1}{64}\right) = 0.984 + 0.016 = 1.0$$

The table below equates values in the ADC0GNH, ADC0GNL, and ADC0GNA registers to the equivalent gain using this equation.

ADC0GNH Value	ADC0GNL Value	GAINADD Value	GAIN Value	Equivalent Gain
0xFC (default)	0x00 (default)	1 (default)	4032 + 64	1.0 (default)
0x7C	0x00	1	1984 + 64	0.5
0xBC	0x00	1	3008 + 64	0.75
0x3C	0x00	1	960 + 64	0.25
0xFF	0xF0	0	4095 + 0	~1.0
0xFF	0xF0	1	4095 + 64	1.016

For any desired gain value, the GAIN registers can be calculated by:

$$GAIN = \left(gain - GAINADD \times \left(\frac{1}{64}\right)\right) \times 4096$$

Equation 4.3. Calculating the ADC0GNH and ADC0GNL Values from the Desired Gain

Where:
GAIN is the 12-bit word of ADC0GNH[7:0] and ADC0GNL[7:4]
GAINADD is the value of the GAINADD bit (ADC0GNA.0)
gain is the equivalent gain value from 0 to 1.016

When calculating the value of GAIN to load into the ADC0GNH and ADC0GNL registers, the GAINADD bit can be turned on or off to reach a value closer to the desired gain value.

Table 8.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
ORL direct, #data	OR immediate to direct byte	3	3
XRL A, Rn	Exclusive-OR Register to A	1	1
XRL A, direct	Exclusive-OR direct byte to A	2	2
XRL A, @Ri	Exclusive-OR indirect RAM to A	1	2
XRL A, #data	Exclusive-OR immediate to A	2	2
XRL direct, A	Exclusive-OR A to direct byte	2	2
XRL direct, #data	Exclusive-OR immediate to direct byte	3	3
CLR A	Clear A	1	1
CPL A	Complement A	1	2
RL A	Rotate A left	1	1
RLC A	Rotate A left through Carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through Carry	1	1
SWAP A	Swap nibbles of A	1	1
Data Transfer			
MOV A, Rn	Move Register to A	1	1
MOV A, direct	Move direct byte to A	2	2
MOV A, @Ri	Move indirect RAM to A	1	2
MOV A, #data	Move immediate to A	2	2
MOV Rn, A	Move A to Register	1	1
MOV Rn, direct	Move direct byte to Register	2	2
MOV Rn, #data	Move immediate to Register	2	2
MOV direct, A	Move A to direct byte	2	2
MOV direct, Rn	Move Register to direct byte	2	2
MOV direct, direct	Move direct byte to direct byte	3	3
MOV direct, @Ri	Move indirect RAM to direct byte	2	2
MOV direct, #data	Move immediate to direct byte	3	3
MOV @Ri, A	Move A to indirect RAM	1	2
MOV @Ri, direct	Move direct byte to indirect RAM	2	2
MOV @Ri, #data	Move immediate to indirect RAM	2	2
MOV DPTR, #data16	Load DPTR with 16-bit constant	3	3
MOVC A, @A+DPTR	Move code byte relative DPTR to A	1	3
MOVC A, @A+PC	Move code byte relative PC to A	1	3
MOVX A, @Ri	Move external data (8-bit address) to A	1	3
MOVX @Ri, A	Move A to external data (8-bit address)	1	3
MOVX A, @DPTR	Move external data (16-bit address) to A	1	3
MOVX @DPTR, A	Move A to external data (16-bit address)	1	3
PUSH direct	Push direct byte onto stack	2	2
POP direct	Pop direct byte from stack	2	2
XCH A, Rn	Exchange Register with A	1	1
XCH A, direct	Exchange direct byte with A	2	2
XCH A, @Ri	Exchange indirect RAM with A	1	2
XCHD A, @Ri	Exchange low nibble of indirect RAM with A	1	2

SFR Definition 8.5. ACC: Accumulator

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xE0

Bits7–0: **ACC**: Accumulator.
This register is the accumulator for arithmetic operations.

SFR Definition 8.6. B: B Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xF0

Bits7–0: **B**: B Register.
This register serves as a second accumulator for certain arithmetic operations.

8.3. Power Management Modes

The CIP-51 core has two software programmable power management modes: Idle and Stop. Idle mode halts the CPU while leaving the peripherals and internal clocks active. In Stop mode, the CPU is halted, all interrupts and timers (except the Missing Clock Detector) are inactive, and the internal oscillator is stopped (analog peripherals remain in their selected states; the external oscillator is not affected). Since clocks are running in Idle mode, power consumption is dependent upon the system clock frequency and the number of peripherals left in active mode before entering Idle. Stop mode consumes the least power. SFR Definition 8.7 describes the Power Control Register (PCON) used to control the CIP-51's power management modes.

Although the CIP-51 has Idle and Stop modes built in (as with any standard 8051 architecture), power management of the entire MCU is better accomplished by enabling/disabling individual peripherals as needed. Each analog peripheral can be disabled when not in use and placed in low power mode. Digital peripherals, such as timers or serial buses, draw little power when they are not in use. Turning off the oscillators lowers power consumption considerably; however a reset is required to restart the MCU.

The C8051F52x/F52xA/F53x/F53xA devices feature a low-power SUSPEND mode, which stops the internal oscillator until a wakening event occurs. See Section “14.1.1. Internal Oscillator Suspend Mode” on page 136 for more information.

8.3.1. Idle Mode

Setting the Idle Mode Select bit (PCON.0) causes the CIP-51 to halt the CPU and enter Idle mode as soon as the instruction that sets the bit completes execution. All internal registers and memory maintain their original data. All analog and digital peripherals can remain active during Idle mode.

Idle mode is terminated when an enabled interrupt is asserted or a reset occurs. The assertion of an enabled interrupt will cause the Idle Mode Selection bit (PCON.0) to be cleared and the CPU to resume operation. The pending interrupt will be serviced and the next instruction to be executed after the return from interrupt (RETI) will be the instruction immediately following the one that set the Idle Mode Select bit. If Idle mode is terminated by an internal or external reset, the CIP-51 performs a normal reset sequence and begins program execution at address 0x0000.

If enabled, the Watchdog Timer (WDT) will eventually cause an internal watchdog reset and thereby terminate the Idle mode. This feature protects the system from an unintended permanent shutdown in the event of an inadvertent write to the PCON register. If this behavior is not desired, the WDT may be disabled by software prior to entering the Idle mode if the WDT was initially configured to allow this operation. This provides the opportunity for additional power savings, allowing the system to remain in the Idle mode indefinitely, waiting for an external stimulus to wake up the system.

8.3.2. Stop Mode

Setting the Stop Mode Select bit (PCON.1) causes the CIP-51 to enter Stop mode as soon as the instruction that sets the bit completes execution. In Stop mode the internal oscillator, CPU, and all digital peripherals are stopped; the state of the external oscillator circuit is not affected. Each analog peripheral (including the external oscillator circuit) may be shut down individually prior to entering Stop Mode. Stop mode can only be terminated by an internal or external reset. On reset, the CIP-51 performs the normal reset sequence and begins program execution at address 0x0000.

If enabled, the Missing Clock Detector will cause an internal reset and thereby terminate the Stop mode. The Missing Clock Detector should be disabled if the CPU is to be put to in STOP mode for longer than the MCD timeout period of 100 µs.

8.3.3. Suspend Mode

The C8051F52x/F52xA/F53x/F53xA devices feature a low-power Suspend mode, which stops the internal oscillator until a wakening event occurs. See Section Section “14.1.1. Internal Oscillator Suspend Mode” on page 136 for more information.

Note: When entering Suspend mode, firmware must set the ZTCEN bit in REF0CN (SFR Definition 5.1).

9.2. Data Memory

The C8051F52x/F52xA/F53x/F53xAincludes 256 bytes of internal RAM mapped into the data memory space from 0x00 through 0xFF. The lower 128 bytes of data memory are used for general purpose registers and scratch pad memory. Either direct or indirect addressing may be used to access the lower 128 bytes of data memory. Locations 0x00 through 0x1F are addressable as four banks of general purpose registers, each bank consisting of eight byte-wide registers. The next 16 bytes, locations 0x20 through 0x2F, may either be addressed as bytes or as 128 bit locations accessible with the direct addressing mode.

The upper 128 bytes of data memory are accessible only by indirect addressing. This region occupies the same address space as the Special Function Registers (SFRs) but is physically separate from the SFR space. The addressing mode used by an instruction when accessing locations above 0x7F determines whether the CPU accesses the upper 128 bytes of data memory space or the SFRs. Instructions that use direct addressing will access the SFR space. Instructions using indirect addressing above 0x7F access the upper 128 bytes of data memory. Figure 9.1 illustrates the data memory organization of the C8051F52x/F52xA/F53x/F53xA.

9.3. General Purpose Registers

The lower 32 bytes of data memory (locations 0x00 through 0x1F) may be addressed as four banks of general-purpose registers. Each bank consists of eight byte-wide registers designated R0 through R7. Only one of these banks may be enabled at a time. Two bits in the program status word, RS0 (PSW.3) and RS1 (PSW.4), select the active register bank (see description of the PSW in SFR Definition 8.4. PSW: Program Status Word). This allows fast context switching when entering subroutines and interrupt service routines. Indirect addressing modes use registers R0 and R1 as index registers.

9.4. Bit Addressable Locations

In addition to direct access to data memory organized as bytes, the sixteen data memory locations at 0x20 through 0x2F are also accessible as 128 individually addressable bits. Each bit has a bit address from 0x00 to 0x7F. Bit 0 of the byte at 0x20 has bit address 0x00 while bit 7 of the byte at 0x20 has bit address 0x07. Bit 7 of the byte at 0x2F has bit address 0x7F. A bit access is distinguished from a full byte access by the type of instruction used (bit source or destination operands as opposed to a byte source or destination).

The MCS-51™ assembly language allows an alternate notation for bit addressing of the form XX.B where XX is the byte address and B is the bit position within the byte. For example, the instruction:

```
MOV    C, 22.3h
```

moves the Boolean value at 0x13 (bit 3 of the byte at location 0x22) into the Carry flag.

9.5. Stack

A programmer's stack can be located anywhere in the 256-byte data memory. The stack area is designated using the Stack Pointer (SP, 0x81) SFR. The SP will point to the last location used. The next value pushed on the stack is placed at SP+1 and then SP is incremented. A reset initializes the stack pointer to location 0x07. Therefore, the first value pushed on the stack is placed at location 0x08, which is also the first register (R0) of register bank 1. Thus, if more than one register bank is to be used, the SP should be initialized to a location in the data memory not being used for data storage. The stack depth can extend up to 256 bytes.

9.6. Special Function Registers

The direct-access data memory locations from 0x80 to 0xFF constitute the special function registers (SFRs). The SFRs provide control and data exchange with the CIP-51's resources and peripherals. The CIP-51 duplicates the SFRs found in a typical 8051 implementation as well as implementing additional

SFR Definition 10.4. EIP1: Extended Interrupt Priority 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
PMAT	PREG0	PLIN	PCPR	PCPF	PPAC0	PREG0	PWADC0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xF6

Bit7: **PMAT**. Port Match Interrupt Priority Control.
This bit sets the priority of the Port Match interrupt.
0: Port Match interrupt set to low priority level.
1: Port Match interrupt set to high priority level.

Bit6: **PREG0**: Voltage Regulator Interrupt Priority Control.
This bit sets the priority of the Voltage Regulator interrupt.
0: Voltage Regulator interrupt set to low priority level.
1: Voltage Regulator interrupt set to high priority level.

Bit5: **PLIN**: LIN Interrupt Priority Control.
This bit sets the priority of the CP0 interrupt.
0: LIN interrupt set to low priority level.
1: LIN interrupt set to high priority level.

Bit4: **PCPR**: Comparator Rising Edge Interrupt Priority Control.
This bit sets the priority of the Rising Edge Comparator interrupt.
0: Comparator interrupt set to low priority level.
1: Comparator interrupt set to high priority level.

Bit3: **PCPF**: Comparator falling Edge Interrupt Priority Control.
This bit sets the priority of the Falling Edge Comparator interrupt.
0: Comparator interrupt set to low priority level.
1: Comparator interrupt set to high priority level.

Bit2: **PPAC0**: Programmable Counter Array (PCA0) Interrupt Priority Control.
This bit sets the priority of the PCA0 interrupt.
0: PCA0 interrupt set to low priority level.
1: PCA0 interrupt set to high priority level.

Bit1: **PREG0**: ADC0 Conversion Complete Interrupt Priority Control.
This bit sets the priority of the ADC0 Conversion Complete interrupt.
0: ADC0 Conversion Complete interrupt set to low priority level.
1: ADC0 Conversion Complete interrupt set to high priority level.

Bit0: **PWADC0**: ADC0 Window Comparison Interrupt Priority Control.
This bit sets the priority of the ADC0 Window Comparison interrupt.
0: ADC0 Window Comparison interrupt set to low priority level.
1: ADC0 Window Comparison interrupt set to high priority level.

10.5. External Interrupts

The $\overline{INT0}$ and $\overline{INT0}$ external interrupt sources are configurable as active high or low, edge or level sensitive. The IN0PL ($\overline{INT0}$ Polarity) and IN1PL ($\overline{INT0}$ 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 182) select level or edge sensitive. The table below lists the possible configurations.

IT0	IN0PL	$\overline{INT0}$ 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	$\overline{INT1}$ 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

$\overline{INT0}$ and $\overline{INT0}$ are assigned to Port pins as defined in the IT01CF register (see SFR Definition 10.5). Note that $\overline{INT0}$ and $\overline{INT0}$ Port pin assignments are independent of any Crossbar assignments. $\overline{INT0}$ and $\overline{INT0}$ 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 $\overline{INT0}$ and/or $\overline{INT0}$, configure the Crossbar to skip the selected pin(s). This is accomplished by setting the associated bit in register XBR0 (see Section “13.1. Priority Crossbar Decoder” on page 122 for complete details on configuring the Crossbar).

In the typical configuration, the external interrupt pins should be skipped in the crossbar and configured as open-drain with the pin latch set to 1. See Section “13. Port Input/Output” on page 120 for more information.

IE0 (TCON.1) and IE1 (TCON.3) serve as the interrupt-pending flags for the $\overline{INT0}$ and $\overline{INT0}$ external interrupts, respectively. If an $\overline{INT0}$ or $\overline{INT0}$ 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 (IN0PL 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.

11. Reset Sources

Reset circuitry allows the controller to be easily placed in a predefined default condition. On entry to this reset state, the following occur:

- CIP-51 halts program execution
- Special Function Registers (SFRs) are initialized to their defined reset values
- External Port pins are forced to a known state
- Interrupts and timers are disabled.

All SFRs are reset to the predefined values noted in the SFR detailed descriptions. The contents of internal data memory are unaffected during a reset; any previously stored data is preserved. However, since the stack pointer SFR is reset, the stack is effectively lost, even though the data on the stack is not altered.

The Port I/O latches are reset to 0xFF (all logic ones) in open-drain mode. Weak pullups are enabled during and after the reset. For V_{DD} Monitor and power-on resets, the $\overline{RST}$ pin is driven low until the device exits the reset state.

On exit from the reset state, the program counter (PC) is reset, and the system clock defaults to the internal oscillator. Refer to Section “14. Oscillators” on page 135 for information on selecting and configuring the system clock source. The Watchdog Timer is enabled with the system clock divided by 12 as its clock source (Section “19.3. Watchdog Timer Mode” on page 203 details the use of the Watchdog Timer). Program execution begins at location 0x0000.

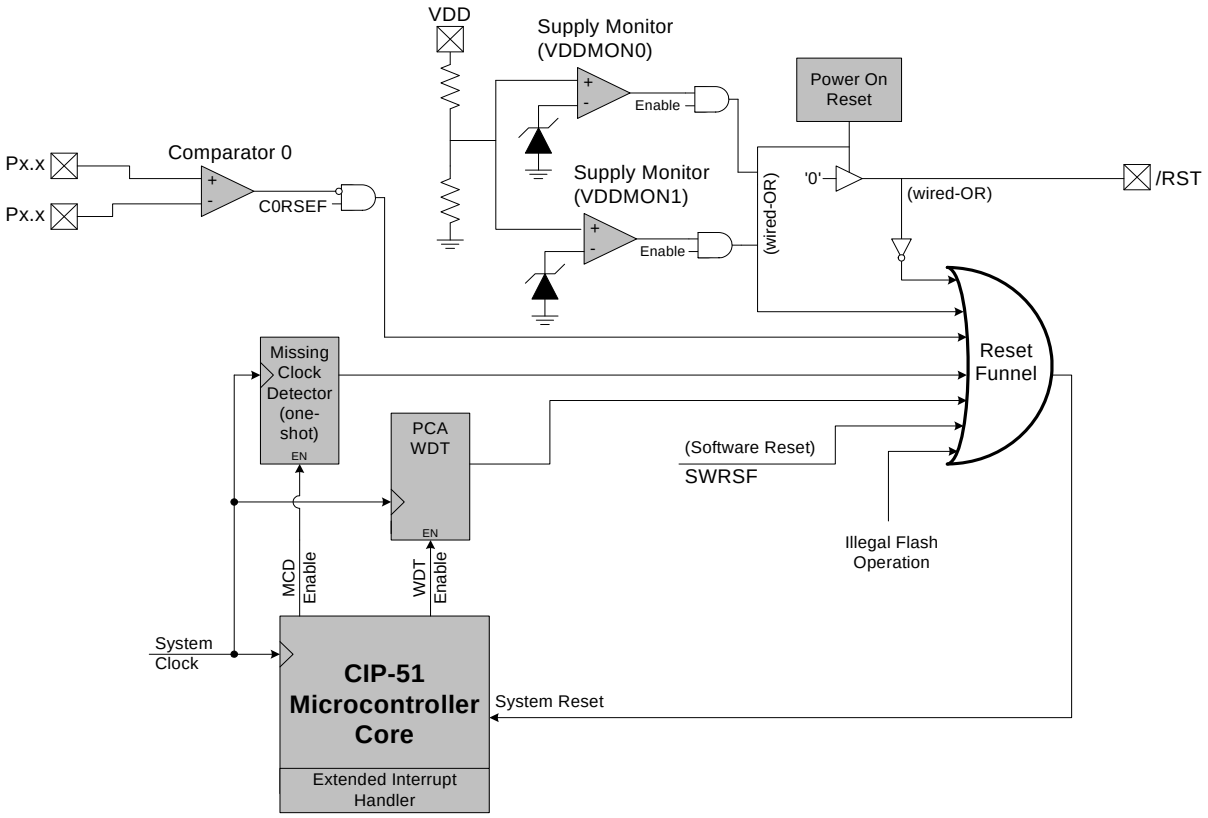

Figure 11.1. Reset Sources

SFR Definition 11.2. RSTSRC: Reset Source

R/W	R	R/W	R/W	R	R/W	R/W	R	Reset Value
—	FERROR	C0RSEF	SWRSF	WDTRSF	MCDRSF	PORSF	PINRSF	Variable
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
							SFR Address:	0xEF

Note: Software should avoid read modify write instructions when writing values to RSTSRC.

Bit7: **UNUSED**. Read = 1, Write = don't care.
Bit6: **FERROR**: Flash Error Indicator.
0: Source of last reset was not a Flash read/write/erase error.
1: Source of last reset was a Flash read/write/erase error.
Bit5: **C0RSEF**: Comparator0 Reset Enable and Flag.
0: **Read:** Source of last reset was not Comparator0.
Write: Comparator0 is not a reset source.
1: **Read:** Source of last reset was Comparator0.
Write: Comparator0 is a reset source (active-low).
Bit4: **SWRSF**: Software Reset Force and Flag.
0: **Read:** Source of last reset was not a write to the SWRSF bit.
Write: No Effect.
1: **Read:** Source of last reset was a write to the SWRSF bit.
Write: Forces a system reset.
Bit3: **WDTRSF**: Watchdog Timer Reset Flag.
0: Source of last reset was not a WDT timeout.
1: Source of last reset was a WDT timeout.
Bit2: **MCDRSF**: Missing Clock Detector Flag.
0: **Read:** Source of last reset was not a Missing Clock Detector timeout.
Write: Missing Clock Detector disabled.
1: **Read:** Source of last reset was a Missing Clock Detector timeout.
Write: Missing Clock Detector enabled; triggers a reset if a missing clock condition is detected.
Bit1: **PORSF**: Power-On Reset Force and Flag.
This bit is set anytime a power-on reset occurs. Writing this bit enables/disables the V_{DD} monitor (VDDMON0) as a reset source. **Note: writing 1 to this bit before the V_{DD} monitor is enabled and stabilized may cause a system reset.** See register VDDMON (SFR Definition 11.1)
0: **Read:** Last reset was not a power-on or V_{DD} monitor reset.
Write: V_{DD} monitor (VDDMON0) is not a reset source.
1: **Read:** Last reset was a power-on or V_{DD} monitor reset; all other reset flags indeterminate.
Write: V_{DD} monitor (VDDMON0) is a reset source.
Bit0: **PINRSF**: HW Pin Reset Flag.
0: Source of last reset was not $\overline{\text{RST}}$ pin.
1: Source of last reset was $\overline{\text{RST}}$ pin.

12.2. Flash Write and Erase Guidelines

Any system which contains routines which write or erase Flash memory from software involves some risk that the write or erase routines will execute unintentionally if the CPU is operating outside its specified operating range of V_{DD}, system clock frequency, or temperature. This accidental execution of Flash modifying code can result in alteration of Flash memory contents causing a system failure that is only recoverable by re-Flashing the code in the device.

The following guidelines are recommended for any system which contains routines which write or erase Flash from code.

12.2.1. V_{DD} Maintenance and the V_{DD} monitor

1. If the system power supply is subject to voltage or current "spikes," add sufficient transient protection devices to the power supply to ensure that the supply voltages listed in the Absolute Maximum Ratings table are not exceeded.
2. Make certain that the maximum V_{DD} ramp time specification (if applicable) is met. See Section 20.4 on page 211 for more details on V_{DD} ramp time. If the system cannot meet this ramp time specification, then add an external V_{DD} brownout circuit to the $\overline{RST}$ pin of the device that holds the device in reset until V_{DD} reaches the minimum specified V_{DD} and re-asserts $\overline{RST}$ if V_{DD} drops belowthat level. V_{DD} (min) is specified in Table 2.2 on page 26.
3. Enable the on-chip V_{DD} monitor (VDDMON0) and enable it as a reset source as early in code as possible. This should be the first set of instructions executed after the Reset Vector. For C-based systems, this will involve modifying the startup code added by the C compiler. See your compiler documentation for more details. Make certain that there are no delays in software between enabling the V_{DD} monitor (VDDMON0) and enabling it as a reset source. Code examples showing this can be found in “AN201: Writing to Flash from Firmware", available from the Silicon Laboratories web site.
4. As an added precaution, explicitly enable the V_{DD} monitor (VDDMON0) and enable the V_{DD} monitor as a reset source inside the functions that write and erase Flash memory. The V_{DD} monitor enable instructions should be placed just after the instruction to set PSWE to a 1, but before the Flash write or erase operation instruction.
5. Make certain that all writes to the RSTSRC (Reset Sources) register use direct assignment operators and explicitly DO NOT use the bit-wise operators (such as AND or OR). For example, "RSTSRC = 0x02" is correct. "RSTSRC |= 0x02" is incorrect.
6. Make certain that all writes to the RSTSRC register explicitly set the PORSF bit to a 1. Areas to check are initialization code which enables other reset sources, such as the Missing Clock Detector or Comparator, for example, and instructions which force a Software Reset. A global search on "RSTSRC" can quickly verify this.

12.2.2. PSWE Maintenance

1. Reduce the number of places in code where the PSWE bit (PSCTL.0) is set to a 1. There should be exactly one routine in code that sets PSWE to a 1 to write Flash bytes and one routine in code that sets PSWE and PSEE both to a 1 to erase Flash pages.
2. Minimize the number of variable accesses while PSWE is set to a 1. Handle pointer address updates and loop variable maintenance outside the "PSWE = 1;... PSWE = 0;" area. Code examples showing this can be found in “AN201: Writing to Flash from Firmware,” available from the Silicon Laboratories web site.
3. Disable interrupts prior to setting PSWE to a 1 and leave them disabled until after PSWE has been reset to '0'. Any interrupts posted during the Flash write or erase operation will be serviced in priority order after the Flash operation has been completed and interrupts have been re-enabled by software.
4. Make certain that the Flash write and erase pointer variables are not located in XRAM. See your compiler documentation for instructions regarding how to explicitly locate variables in different memory areas.

17.1. Software Interface with the LIN Peripheral

The selection of the mode (Master or Slave) and the automatic baud rate feature are done though the LIN0 Control Mode (LIN0CF) register. The other LIN registers are accessed indirectly through the two SFRs LIN0 Address (LINADDR) and LIN0 Data (LINDATA). The LINADDR register selects which LIN register is targeted by reads/writes of the LINDATA register. The full list of indirectly-accessible LIN register is given in Table 17.4 on page 174.

17.2. LIN Interface Setup and Operation

The hardware based LIN peripheral allows for the implementation of both Master and Slave nodes with minimal firmware overhead and complete control of the interface status while allowing for interrupt and polled mode operation.

The first step to use the peripheral is to define the basic characteristics of the node:

- Mode—Master or Slave
- Baud Rate—Either defined manually or using the autobaud feature (slave mode only).
- Checksum Type—Select between classic or enhanced checksum, both of which are implemented in hardware.

17.2.1. Mode Definition

Following the LIN specification, the peripheral implements both the Slave and Master operating modes in hardware. The mode is configured using the MODE bit (LIN0CF.6).

17.2.2. Baud Rate Options: Manual or Autobaud

The LIN peripheral can be selected to have its baud rate calculated manually or automatically. A master node must always have its baud rate set manually, but slave nodes can choose between a manual or automatic setup. The configuration is selected using the ABAUD bit (LIN0CF.5).

Both the manual and automatic baud rate configurations require additional setup. The following sections explain the different options available and their relation with the baud rate, along with the steps necessary to achieve the required baud rate.

17.2.3. Baud Rate Calculations—Manual Mode

The baud rate used by the peripheral is a function of the System Clock (SYSCLK) and the bit-timing Registers according to the following equation:

$$baud\_rate = \frac{SYSCLK}{2^{(\text{prescaler} + 1)} \times divider \times (multiplier + 1)}$$

The prescaler, divider and multiplier factors are part of the LIN0DIV and LIN0MUL registers and can assume values in the following range:

17.4. LIN Slave Mode Operation

When the device is configured for slave mode operation, it must wait for a command from a master node. Access from the firmware to data buffer and ID registers of the LIN peripheral is only possible when a data request is pending (DTREQ bit (LIN0ST.4) is 1) and also when the LIN bus is not active (ACTIVE bit (LIN0ST.7) is set to 0).

The LIN peripheral in slave mode detects the header of the message frame sent by the LIN master. If slave synchronization is enabled (autobaud), the slave synchronizes its internal bit time to the master bit time.

The LIN peripheral configured for slave mode will generated an interrupt in one of three situations:

1. After the reception of the IDENTIFIER FIELD.
2. When an error is detected.
3. When the message transfer is completed.

The application should perform the following steps when an interrupt is detected:

1. Check the status of the DTREQ bit (LIN0ST.4). This bit is set when the IDENTIFIER FIELD has been received.
2. If DTREQ (LIN0ST.4) is set, read the identifier from LIN0ID and process it. If DTREQ (LIN0ST.4) is not set, continue to step 7.
3. Set the TXRX bit (LIN0CTRL.5) to 1 if the current frame is a transmit operation for the slave and set to 0 if the current frame is a receive operation for the slave.
4. Load the data length into LIN0SIZE.
5. For a slave transmit operation, load the data to transmit into the data buffer.
6. Set the DTACK bit (LIN0CTRL.4). Continue to step 10.
7. If DTREQ (LIN0ST.4) is not set, check the DONE bit (LIN0ST.0). The transmission was successful if the DONE bit is set.
8. If the transmission was successful and the current frame was a receive operation for the slave, load the received data bytes from the data buffer.
9. If the transmission was not successful, check LIN0ERR to determine the nature of the error. Further error handling has to be done by the application.
10. Set the RSTINT (LIN0CTRL.3) and RSTERR bits (LIN0CTRL.2) to reset the interrupt request and the error flags.

In addition to these steps, the application should be aware of the following:

1. If the current frame is a transmit operation for the slave, steps 1 through 5 must be completed during the IN-FRAME RESPONSE SPACE. If it is not completed in time, a timeout will be detected by the master.
2. If the current frame is a receive operation for the slave, steps 1 through 5 have to be finished until the reception of the first byte after the IDENTIFIER FIELD. Otherwise, the internal receive buffer of the LIN peripheral will be overwritten and a timeout error will be detected in the LIN peripheral.
3. The LIN module does not directly support LIN Version 1.3 Extended Frames. If the application detects an unknown identifier (e.g. extended identifier), it has to write a 1 to the STOP bit (LIN0CTRL.7) instead of setting the DTACK (LIN0CTRL.4) bit. At that time, steps 2 through 5 can then be skipped. In this situation, the LIN peripheral stops the processing of the LIN communication until the next SYNC BREAK is received.
4. Changing the configuration of the checksum during a transaction will cause the interface to reset and the transaction to be lost. To prevent this, the checksum should not be configured while a transaction is

SFR Definition 17.14. LIN0ERR: LIN0 ERROR Register

R	R	R	R	R	R	R	R	Reset Value
			SYNCH	PRTY	TOUT	CHK	BITERR	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

Address: 0x0A (indirect)

Bits7–5: **UNUSED**. Read = 000b. Write = don't care.

Bit4: **SYNCH**: Synchronization Error Bit (**slave mode only**).
0: No error with the SYNCH FIELD has been detected.
1: Edges of the SYNCH FIELD are outside of the maximum tolerance.

Bit3: **PRTY**: Parity Error Bit (**slave mode only**).
0: No parity error has been detected.
1: A parity error has been detected.

Bit2: **TOUT**: Timeout Error Bit.
0: A timeout error has not been detected.
1: A timeout error has been detected. This error is detected whenever one of the following conditions is met:
•The master is expecting data from a slave and the slave does not respond.
•The slave is expecting data but no data is transmitted on the bus.
•A frame is not finished within the maximum frame length.
•The application does not set the DTACK bit (LIN0CTRL.4) or STOP bit (LIN0CTRL.7) until the end of the reception of the first byte after the identifier.

Bit1: **CHK**: Checksum Error Bit.
0: Checksum error has not been detected.
1: Checksum error has been detected.

Bit0: **BITERR**: Bit Transmission Error Bit.
0: No error in transmission has been detected.
1: The bit value monitored during transmission is different than the bit value sent.

19.2.2. Software Timer (Compare) Mode

In Software Timer mode, the PCA counter/timer value is compared to the module's 16-bit capture/compare register (PCA0CPHn and PCA0CPLn). When a match occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1 and an interrupt request is generated if CCF interrupts are enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. Setting the ECOMn and MATn bits in the PCA0CPMn register enables Software Timer mode.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to 0; writing to PCA0CPHn sets ECOMn to 1.

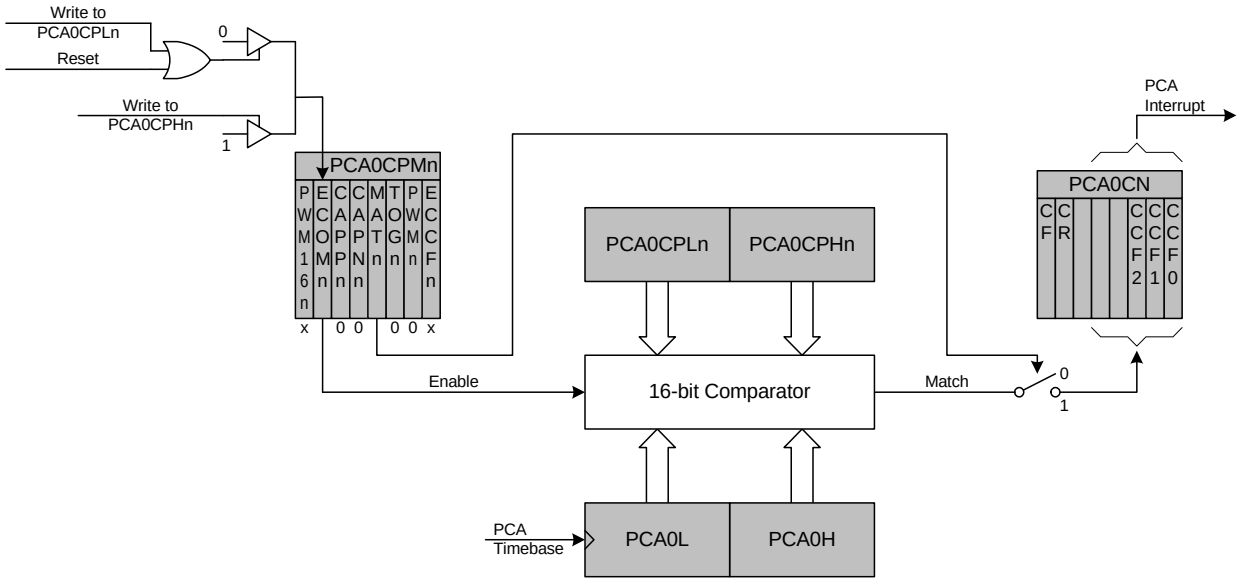

Figure 19.5. PCA Software Timer Mode Diagram

SFR Definition 19.2. PCA0MD: PCA Mode

R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	Reset Value
CIDL	WDTE	WDLCK	-	CPS2	CPS1	CPS0	ECF	01000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xD9

Bit7: **CIDL**: PCA Counter/Timer Idle Control.
Specifies PCA behavior when CPU is in Idle Mode.
0: PCA continues to function normally while the system controller is in Idle Mode.
1: PCA operation is suspended while the system controller is in Idle Mode.

Bit6: **WDTE**: Watchdog Timer Enable
If this bit is set, PCA Module 2 is used as the watchdog timer.
0: Watchdog Timer disabled.
1: PCA Module 2 enabled as Watchdog Timer.

Bit5: **WDLCK**: Watchdog Timer Lock
This bit locks/unlocks the Watchdog Timer Enable. When WDLCK is set, the Watchdog Timer may not be disabled until the next system reset.
0: Watchdog Timer Enable unlocked.
1: Watchdog Timer Enable locked.

Bit4: **UNUSED**. Read = 0b, Write = don't care.

Bits3–1: **CPS2–CPS0**: PCA Counter/Timer Pulse Select.
These bits select the timebase source for the PCA counter.

CPS2	CPS1	CPS0	Timebase
0	0	0	System clock divided by 12
0	0	1	System clock divided by 4
0	1	0	Timer 0 overflow
0	1	1	High-to-low transitions on ECI (max rate = system clock divided by 4)
1	0	0	System clock
1	0	1	External clock divided by 8*
1	1	0	Reserved
1	1	1	Reserved

Note: External clock divided by 8 is synchronized with the system clock.

Bit0: **ECF**: PCA Counter/Timer Overflow Interrupt Enable.
This bit sets the masking of the PCA Counter/Timer Overflow (CF) interrupt.
0: Disable the CF interrupt.
1: Enable a PCA Counter/Timer Overflow interrupt request when CF (PCA0CN.7) is set.

Note: When the WDTE bit is set to 1, the PCA0MD register cannot be modified. To change the contents of the PCA0MD register, the Watchdog Timer must first be disabled.

21. C2 Interface

C8051F52x/F52xA/F53x/F53xA devices include an on-chip Silicon Laboratories 2-Wire (C2) debug interface to allow Flash programming and in-system debugging with the production part installed in the end application. The C2 interface uses a clock signal (C2CK) and a bi-directional C2 data signal (C2D) to transfer information between the device and a host system. See the C2 Interface Specification for details on the C2 protocol.

21.1. C2 Interface Registers

The following describes the C2 registers necessary to perform Flash programming functions through the C2 interface. All C2 registers are accessed through the C2 interface as described in the C2 Interface Specification.

C2 Register Definition 21.1. C2ADD: C2 Address

Reset Value: 00000000

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bits7–0: The C2ADD register is accessed via the C2 interface to select the target Data register for C2 Data Read and Data Write commands.

Address	Description
0x00	Selects the Device ID register for Data Read instructions (DEVICEID)
0x01	Selects the Revision ID register for Data Read instructions (REVID)
0x02	Selects the C2 Flash Programming Control register for Data Read/Write instructions (FPCTL)
0xB4	Selects the C2 Flash Programming Data register for Data Read/Write instructions (FPDAT)

C2 Register Definition 21.2. DEVICEID: C2 Device ID

Reset Value: 00010001

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

This read-only register returns the 8-bit device ID: 0x11 (C8051F52x/F52xA/F53x/F53xA).