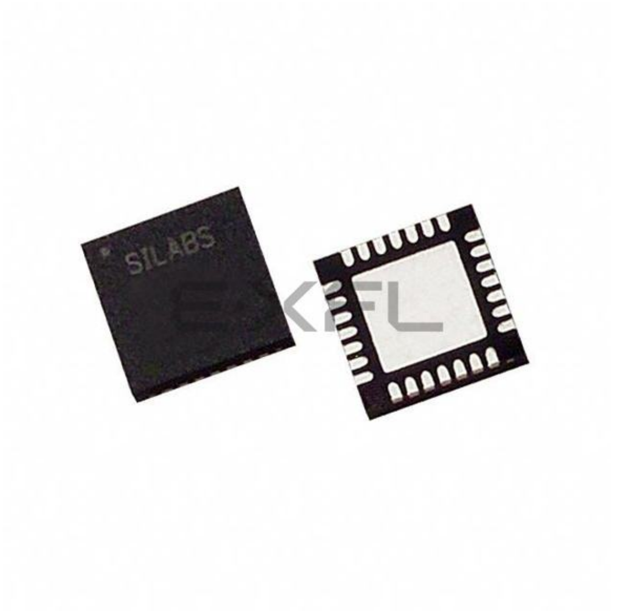
Silicon Labs - C8051F411-GMR Datasheet



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), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	20
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.25V
Data Converters	A/D 20x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	28-VFQFN Exposed Pad
Supplier Device Package	28-QFN (5x5)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f411-gmr

NOTES:

SFR Definition 5.9. ADC0LTH: ADC0 Less-Than Data High Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xC6

Bits7–0: High byte of ADC0 Less-Than Data Word.

SFR Definition 5.10. ADC0LTL: ADC0 Less-Than Data Low Byte

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address:
								0xC5

Bits7–0: Low byte of ADC0 Less-Than Data Word.

SFR Definition 9.2. CPT0MX: Comparator0 MUX Selection

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
CMX0N3	CMX0N2	CMX0N1	CMX0N0	CMX0P3	CMX0P2	CMX0P1	CMX0P0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	SFR Address: 0x9F

Bits7–4: CMX0N3–CMX0N0: Comparator0 Negative Input MUX Select.
These bits select which Port pin is used as the Comparator0 negative input.

CMX0N3	CMX0N2	CMX0N1	CMX0N0	Negative Input
0	0	0	0	P0.1
0	0	0	1	P0.3
0	0	1	0	P0.5
0	0	1	1	P0.7
0	1	0	0	P1.1
0	1	0	1	P1.3
0	1	1	0	P1.5
0	1	1	1	P1.7
1	0	0	0	P2.1
1	0	0	1	P2.3*
1	0	1	0	P2.5*
1	0	1	1	P2.7
1	1	x	x	Reserved
***Note:** *Available only on the C8051F410/2.*				

Bits1–0: CMX0P3–CMX0P0: Comparator0 Positive Input MUX Select.
These bits select which Port pin is used as the Comparator0 positive input.

CMX0P3	CMX0P2	CMX0P1	CMX0P0	Positive Input
0	0	0	0	P0.0
0	0	0	1	P0.2
0	0	1	0	P0.4
0	0	1	1	P0.6
0	1	0	0	P1.0
0	1	0	1	P1.2
0	1	1	0	P1.4
0	1	1	1	P1.6
1	0	0	0	P2.0
1	0	0	1	P2.2
1	0	1	0	P2.4*
1	0	1	1	P2.6*
1	1	x	x	Reserved
***Note:** *Available only on the C8051F410/2.*				

Table 10.1. CIP-51 Instruction Set Summary[1] (Continued)

Mnemonic	Description	Bytes	Clock Cycles
ORL A, @Ri	OR indirect RAM to A	1	2
ORL A, #data	OR immediate to A	2	2
ORL direct, A	OR A to direct byte	2	2
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	1
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	4 to 7[2]
MOVC A, @A+PC	Move code byte relative PC to A	1	4 to 7[2]
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

Notes:

1. Assumes PFEN = 1 for all instruction timing.
2. MOVC instructions take 4 to 7 clock cycles depending on instruction alignment and the FLRT setting (SFR Definition 16.3. FLSCL: Flash Scale).

SFR Definition 12.3. EIE1: Extended Interrupt Enable 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
ET3	ECP1	ECP0	EPCA0	EADC0	EWADC0	ERTC0	ESMB0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xE6

Bit 7: ET3: Enable Timer 3 Interrupt.
This bit sets the masking of the Timer 3 interrupt.
0: Disable Timer 3 interrupts.
1: Enable interrupt requests generated by the TF3L or TF3H flags.

Bit 6: ECP1: Enable Comparator1 (CP1) Interrupt.
This bit sets the masking of the CP1 interrupt.
0: Disable CP1 interrupts.
1: Enable interrupt requests generated by the CP1RIF or CP1FIF flags.

Bit 5: ECP0: Enable Comparator0 (CP0) Interrupt.
This bit sets the masking of the CP0 interrupt.
0: Disable CP0 interrupts.
1: Enable interrupt requests generated by the CP0RIF or CP0FIF flags.

Bit 4: EPCA0: Enable Programmable Counter Array (PCA0) Interrupt.
This bit sets the masking of the PCA0 interrupts.
0: Disable all PCA0 interrupts.
1: Enable interrupt requests generated by PCA0.

Bit 3: EADC0: Enable ADC0 Conversion Complete Interrupt.
This bit sets the masking of the ADC0 Conversion Complete interrupt.
0: Disable ADC0 Conversion Complete interrupt.
1: Enable interrupt requests generated by the AD0INT flag.

Bit 2: EWADC0: Enable ADC0 Window Comparison Interrupt.
This bit sets the masking of the ADC0 Window Comparison interrupt.
0: Disable ADC0 Window Comparison interrupt.
1: Enable interrupt requests generated by the AD0WINT flag.

Bit 1: ERTC0: Enable smaRTClock Interrupt.
This bit sets the masking of the smaRTClock interrupt.
0: Disable smaRTClock interrupts.
1: Enable interrupt requests generated by the ALRM and OSCFAIL flag.

Bit 0: ESMB0: Enable SMBus (SMB0) Interrupt.
This bit sets the masking of the SMB0 interrupt.
0: Disable all SMB0 interrupts.
1: Enable interrupt requests generated by SMB0.

SFR Definition 12.7. IT01CF: INT0/INT1 Configuration

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
IN1PL	IN1SL2	IN1SL1	IN1SL0	IN0PL	IN0SL2	IN0SL1	IN0SL0	00000001
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xE4

Note: Refer to SFR Definition 24.1. “TCON: Timer Control” on page 235 for INT0/1 edge- or level-sensitive interrupt selection.

Bit 7: IN1PL: /INT1 Polarity
0: /INT1 input is active low.
1: /INT1 input is active high.

Bits 6–4: IN1SL2–0: /INT1 Port Pin Selection Bits
These bits select which Port pin is assigned to /INT1. Note that this pin assignment is independent of the Crossbar; /INT1 will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin (accomplished by setting to ‘1’ the corresponding bit in register P0SKIP).

IN1SL2–0	/INT1 Port Pin
000	P0.0
001	P0.1
010	P0.2
011	P0.3
100	P0.4
101	P0.5
110	P0.6
111	P0.7

Bit 3: IN0PL: /INT0 Polarity
0: /INT0 interrupt is active low.
1: /INT0 interrupt is active high.

Bits 2–0: INT0SL2–0: /INT0 Port Pin Selection Bits
These bits select which Port pin is assigned to /INT0. Note that this pin assignment is independent of the Crossbar. /INT0 will monitor the assigned Port pin without disturbing the peripheral that has been assigned the Port pin via the Crossbar. The Crossbar will not assign the Port pin to a peripheral if it is configured to skip the selected pin (accomplished by setting to ‘1’ the corresponding bit in register P0SKIP).

IN0SL2–0	/INT0 Port Pin
000	P0.0
001	P0.1
010	P0.2
011	P0.3
100	P0.4
101	P0.5
110	P0.6
111	P0.7

For example, the 16-bit 'F41x CRC algorithm can be described by the following code:

```
unsigned short UpdateCRC (unsigned short CRC_acc, unsigned char CRC_input)
{
   unsigned char i;                    // loop counter

   #define POLY 0x1021

   // Create the CRC "dividend" for polynomial arithmetic (binary arithmetic
   // with no carries)
   CRC_acc = CRC_acc ^ (CRC_input << 8);

   // "Divide" the poly into the dividend using CRC XOR subtraction
   // CRC_acc holds the "remainder" of each divide
   //
   // Only complete this division for 8 bits since input is 1 byte
   for (i = 0; i < 8; i++)
   {
      // Check if the MSB is set (if MSB is 1, then the POLY can "divide"
      // into the "dividend")
      if ((CRC_acc & 0x8000) == 0x8000)
      {
         // if so, shift the CRC value, and XOR "subtract" the poly
         CRC_acc = CRC_acc << 1;
         CRC_acc ^= POLY;
      }
      else
      {
         // if not, just shift the CRC value
         CRC_acc = CRC_acc << 1;
      }
   }

   // Return the final remainder (CRC value)
   return CRC_acc;
}
```

The following table lists several input values and the associated outputs using the 16-bit 'F41x CRC algorithm (an initial value of 0xFFFF is used):

Table 14.1. Example 16-bit CRC Outputs

Input	Output
0x63	0xBD35
0x8C	0xB1F4
0x7D	0x4ECA
0xAA, 0xBB, 0xCC	0x6CF6
0x00, 0x00, 0xAA, 0xBB, 0xCC	0xB166

16. Flash Memory

On-chip, re-programmable Flash memory is included for program code and non-volatile data storage. The Flash memory can be programmed in-system through the C2 interface or by software using the MOVX write instruction. Once cleared to logic 0, a Flash bit must be erased to set it back to logic 1. Flash bytes would typically be erased (set to 0xFF) before being reprogrammed. The write and erase operations are automatically timed by hardware for proper execution; data polling to determine the end of the write/erase operations is not required. Code execution is stalled during Flash write/erase operations. Refer to Table 16.2 for complete Flash memory electrical characteristics.

16.1. Programming The Flash Memory

The simplest means of programming the Flash memory is through the C2 interface using programming tools provided by Silicon Laboratories or a third party vendor. This is the only means for programming a non-initialized device. For details on the C2 commands to program Flash memory, see **Section "26. C2 Interface" on page 265**. For detailed guidelines on writing or erasing Flash from firmware, please see **Section "16.4. Flash Write and Erase Guidelines" on page 139**.

To ensure the integrity of the Flash contents, the on-chip VDD Monitor must be enabled to the higher setting (VDMLVL = '1') in any system that includes code that writes and/or erases Flash memory from software. Furthermore, there should be no delay between enabling the V_{DD} Monitor and enabling the V_{DD} Monitor as a reset source. Any attempt to write or erase Flash memory while the V_{DD} Monitor disabled will cause a Flash Error device reset.

16.1.1. Flash Lock and Key Functions

Flash writes and erases by user software are protected with a lock and key function. The Flash Lock and Key Register (FLKEY) must be written with the correct key codes, in sequence, before Flash operations may be performed. The key codes are: 0xA5, 0xF1. The timing does not matter, but the codes must be written in order. If the key codes are written out of order, or the wrong codes are written, Flash writes and erases will be disabled until the next system reset. Flash writes and erases will also be disabled if a Flash write or erase is attempted before the key codes have been written properly. The Flash lock resets after each write or erase; the key codes must be written again before a following Flash operation can be performed. The FLKEY register is detailed in SFR Definition 16.2.

16.1.2. Flash Erase Procedure

The Flash memory can be programmed by software using the MOVX write instruction with the address and data byte to be programmed provided as normal operands. Before writing to Flash memory using MOVX, Flash write operations must be enabled by: (1) setting the PSWE Program Store Write Enable bit (PSCTL.0) to logic 1 (this directs the MOVX writes to target Flash memory); and (2) Writing the Flash key codes in sequence to the Flash Lock register (FLKEY). The PSWE bit remains set until cleared by software.

A write to Flash memory can clear bits to logic 0 but cannot set them; only an erase operation can set bits to logic 1 in Flash. **A byte location to be programmed should be erased before a new value is written.** The Flash memory is organized in 512-byte pages. The erase operation applies to an entire page (setting all bytes in the page to 0xFF). To erase an entire 512-byte page, perform the following steps:

Step 1. Disable interrupts (recommended).
Step 2. Write the first key code to FLKEY: 0xA5.
Step 3. Write the second key code to FLKEY: 0xF1.
Step 4. Set the PSEE bit (register PSCTL).
Step 5. Set the PSWE bit (register PSCTL).
Step 6. Using the MOVX instruction, write a data byte to any location within the 512-byte page to be erased.
Step 7. Clear the PSWE and PSEE bits.
Step 8. Re-enable interrupts.

SFR Definition 16.4. ONESHOT: Flash Oneshot Period

R	R	R	R	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	PERIOD				00001111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xAF

Bits7–4: UNUSED. Read = 0000b. Write = don't care.

Bits3–0: PERIOD: Oneshot Period Control Bits.
These bits limit the internal Flash read strobe width as follows. When the Flash read strobe is de-asserted, the Flash memory enters a low-power state for the remainder of the system clock cycle. These bits have no effect when the system clocks is greater than 12.5 MHz and FLRT = 0.

$$FLASH_{RDMAX} = 5ns + (PERIOD \times 5ns)$$

Table 16.2. Flash Electrical Characteristics

V_{DD} = 2.0 to 2.75 V; –40 to +85 ºC unless otherwise specified. Typical values are given at 25 ºC.

Parameter	Conditions	Min	Typ	Max	Units
Flash Size	C8051F410/1 C8051F412/3	32768* 16384	—	—	bytes
Endurance	V_{DD} is 2.2 V or greater	20 k	90 k	—	Erase/Write
Erase Cycle Time	FLSCL.3–0 written to '0000'	16	20	24	ms
Write Cycle Time	FLSCL.3–0 written to '0000'	38	46	57	µs
Read Cycle Time		40	—	—	ns
V_{DD}	Write/Erase Operations	2.25	—	—	V
***Note:** 512 bytes at addresses 0x7E00 to 0x7FFF are reserved.					

(P1MATCH & P1MASK). This allows Software to be notified if a certain change or pattern occurs on P0 or P1 input pins regardless of the XBRn settings. A port match event can cause an interrupt if EMAT (EIE2.1) is set to '1' or cause the internal oscillator to awaken from SUSPEND mode. See Section “**19.1.1. Internal Oscillator Suspend Mode**” on page **166** for more information.

SFR Definition 18.3. P0: Port0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0x80

Bits7–0: P0.[7:0]
Write - Output appears on I/O pins per Crossbar Registers.
0: Logic Low Output.
1: Logic High Output (high impedance if corresponding P0MDOUT.n bit = 0).
Read - Always reads ‘0’ if selected as analog input in register P0MDIN. Directly reads Port pin when configured as digital input.
0: P0.n pin is logic low.
1: P0.n pin is logic high.

SFR Definition 18.4. P0MDIN: Port0 Input Mode

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
								11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xF1

Bits7–0: Analog Input Configuration Bits for P0.7–P0.0 (respectively).
Port pins configured as analog inputs have their weak pullup, digital driver, and digital receiver disabled.
0: Corresponding P0.n pin is configured as an analog input. **In order for the P0.n pin to be in analog input mode, there MUST be a '1' in the Port Latch register corresponding to that pin.**
1: Corresponding P0.n pin is not configured as an analog input.

SFR Definition 19.4. CLKMUL: Clock Multiplier Control

R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	Reset Value
MULEN	MULINIT	MULRDY	MULDIV			MULSEL		00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xAB

Note: The maximum SYSCLK is 50 MHz, so the Clock Multiplier output should be scaled accordingly.

Bit7: MULEN: Clock Multiplier Enable
0: Clock Multiplier disabled.
1: Clock Multiplier enabled.

Bit6: MULINIT: Clock Multiplier Initialize
This bit should be a ‘0’ when the Clock Multiplier is enabled. Once enabled, writing a ‘1’ to this bit will initialize the Clock Multiplier. The MULRDY bit reads ‘1’ when the Clock Multiplier is stabilized.

Bit5: MULRDY: Clock Multiplier Ready
This read-only bit indicates the status of the Clock Multiplier.
0: Clock Multiplier not ready.
1: Clock Multiplier ready (locked).

Bits4–2: MULDIV: Clock Multiplier Output Scaling Factor
These bits scale the Clock Multiplier output.
000: Clock Multiplier Output scaled by a factor of 1.
001: Clock Multiplier Output scaled by a factor of 1.
010: Clock Multiplier Output scaled by a factor of 1.
011: Clock Multiplier Output scaled by a factor of 2/3*.
100: Clock Multiplier Output scaled by a factor of 2/4 (or 1/2).
101: Clock Multiplier Output scaled by a factor of 2/5*.
110: Clock Multiplier Output scaled by a factor of 2/6 (or 1/3).
111: Clock Multiplier Output scaled by a factor of 2/7*.
***Note:** The Clock Multiplier Output duty cycle is not 50% for these settings.

Bits1–0: MULSEL: Clock Multiplier Input Select
These bits select the clock supplied to the Clock Multiplier.

MULSEL	Selected Input Clock	Clock Multiplier Output for MULDIV = 000b
00	Internal Oscillator / 2	Internal Oscillator x 2
01	External Oscillator	External Oscillator x 4
10	External Oscillator / 2	External Oscillator x 2
11	Internal Oscillator	Internal Oscillator x 4

20.1. smaRTClock Interface

The smaRTClock Interface consists of three registers: RTC0KEY, RTC0ADR, and RTC0DAT. These interface registers are located on the CIP-51's SFR map and provide access to the smaRTClock internal registers listed in Table 20.1. The smaRTClock internal registers can only be accessed indirectly through the smaRTClock Interface.

20.1.1. smaRTClock Lock and Key Functions

The smaRTClock Interface is protected with a lock and key function. The smaRTClock Lock and Key Register (RTC0KEY) must be written with the correct key codes, in sequence, before writes and reads to RTC0ADR and RTC0DAT may be performed. The key codes are: 0xA5, 0xF1. There are no timing restrictions, but the key codes must be written in order. If the key codes are written out of order, the wrong codes are written, or an invalid read or write is attempted, further writes and reads to RTC0ADR and RTC0DAT will be disabled until the next system reset. Once the smaRTClock interface is unlocked, software may perform accesses of the smaRTClock registers until an invalid access, the interface is locked, or a system reset.

Reading the RTC0KEY register at any time will provide the smaRTClock Interface status and will not interfere with the sequence that is being written. The RTC0KEY register description in SFR Definition 20.1 lists the definition of each status code.

20.1.2. Using RTC0ADR and RTC0DAT to Access smaRTClock Internal Registers

The smaRTClock internal registers can be read and written using RTC0ADR and RTC0DAT. The RTC0ADR register selects the smaRTClock internal register that will be targeted by subsequent reads or writes. Prior to each read or write, BUSY (RTC0ADR.7) should be checked to make sure the smaRTClock Interface is not busy performing another read or write operation. A smaRTClock Write operation is initiated by writing to the RTC0DAT register. Below is an example of writing to a smaRTClock internal register.

Step 1. Poll BUSY (RTC0ADR.7) until it returns a '0'.
Step 2. Write 0x06 to RTC0ADR. This selects the internal RTC0CN register at smaRTClock Address 0x06.
Step 3. Write 0x00 to RTC0DAT. This operation writes 0x00 to the internal RTC0CN register.

An smaRTClock Read operation is initiated by setting the smaRTClock Interface Busy bit. This transfers the contents of the internal register selected by RTC0ADR to RTC0DAT. The transferred data will remain in RTC0DAT until the next read or write operation. Below is an example of reading a smaRTClock internal register.

Step 1. Poll BUSY (RTC0ADR.7) until it returns a '0'.
Step 2. Write 0x06 to RTC0ADR. This selects the internal RTC0CN register at smaRTClock Address 0x06.
Step 3. Write '1' to BUSY. This initiates the transfer of data from RTC0CN to RTC0DAT.
Step 4. Poll BUSY (RTC0ADR.7) until it returns a '0'.
Step 5. Read data from RTC0DAT. This data is a copy of the RTC0CN register.

Note: The RTC0ADR and RTC0DAT registers will retain their state upon a device reset.

20.1.3. smaRTClock Interface Autoread Feature

When Autoread is enabled, each read from RTC0DAT initiates the next indirect read operation on the smaRTClock internal register selected by RTC0ADR. Software should set the BUSY bit once at the begin-

Internal Register Definition 20.4. RTC0CN: smaRTClock Control

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
RTC0EN	MCLKEN	OSCFAIL	RTC0TR	RTC0AEN	ALRM	RTC0SET	RTC0CAP	Variable
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	smaRTClock Address:
Note: This register is not an SFR. It can only be accessed indirectly through RTC0ADR and RTC0DAT.								0x06

Bit 7: RTC0EN: smaRTClock Enable Bit.
0: smaRTClock bias and crystal oscillator disabled. smaRTClock is powered from V_{DD} only.
1: smaRTClock bias and crystal oscillator enabled. smaRTClock can switch to the backup battery if V_{DD} fails.

Bit 6: MCLKEN: smaRTClock Missing Clock Detector Enable Bit.
When enabled, the smaRTClock missing clock detector sets the OSCFAIL bit if the smaRTClock clock frequency falls below approximately 20 kHz.
0: smaRTClock missing clock detector disabled.
1: smaRTClock missing clock detector enabled.

Bit 5: OSCFAIL: smaRTClock Clock Fail Flag.
Set by hardware when a missing clock detector timeout occurs. When the smaRTClock Interrupt is enabled, setting this bit causes the CPU to vector to the smaRTClock interrupt service routine. This bit is not automatically cleared by hardware.

Bit 4: RTC0TR: smaRTClock Timer Run Control.
0: smaRTClock timer holds its current value.
1: smaRTClock timer increments every smaRTClock clock period.

Bit 3: RTC0AEN: smaRTClock Alarm Enable.
0: smaRTClock alarm events disabled.
1: smaRTClock alarm events enabled.

Bit 2: ALRM: smaRTClock Alarm Event Flag.
Set by hardware when the smaRTClock timer value is **greater than or equal to** the value of the ALARMn registers. When the smaRTClock Interrupt is enabled, setting this bit causes the CPU to vector to the smaRTClock interrupt service routine. This bit is not automatically cleared by hardware.

Bit 1: RTC0SET: smaRTClock Set Bit.
Writing a ‘1’ to this bit causes the 47-bit value in CAPTUREn registers to be transferred to the smaRTClock timer. This bit is automatically cleared by hardware once the transfer is complete.

Bit 0: RTC0CAP: smaRTClock Capture Bit.
Writing a ‘1’ to this bit causes the 47-bit smaRTClock timer value to be transferred to the CAPTUREn registers. This bit is automatically cleared by hardware once the transfer is complete.

SFR Definition 22.1. SCON0: Serial Port 0 Control

R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
S0MODE	-	MCE0	REN0	TB80	RB80	TI0	RI0	01000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0x98

Bit7: S0MODE: Serial Port 0 Operation Mode.
This bit selects the UART0 Operation Mode.
0: 8-bit UART with Variable Baud Rate.
1: 9-bit UART with Variable Baud Rate.

Bit6: UNUSED. Read = 1b. Write = don't care.

Bit5: MCE0: Multiprocessor Communication Enable.
The function of this bit is dependent on the Serial Port 0 Operation Mode.
S0MODE = 0: Checks for valid stop bit.
0: Logic level of stop bit is ignored.
1: RI0 will only be activated if stop bit is logic level 1.
S0MODE = 1: Multiprocessor Communications Enable.
0: Logic level of ninth bit is ignored.
1: RI0 is set and an interrupt is generated only when the ninth bit is logic 1.

Bit4: REN0: Receive Enable.
This bit enables/disables the UART receiver.
0: UART0 reception disabled.
1: UART0 reception enabled.

Bit3: TB80: Ninth Transmission Bit.
The logic level of this bit will be assigned to the ninth transmission bit in 9-bit UART Mode. It is not used in 8-bit UART Mode. Set or cleared by software as required.

Bit2: RB80: Ninth Receive Bit.
RB80 is assigned the value of the STOP bit in Mode 0; it is assigned the value of the 9th data bit in Mode 1.

Bit1: TI0: Transmit Interrupt Flag.
Set by hardware when a byte of data has been transmitted by UART0 (after the 8th bit in 8-bit UART Mode, or at the beginning of the STOP bit in 9-bit UART Mode). When the UART0 interrupt is enabled, setting this bit causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.

Bit0: RI0: Receive Interrupt Flag.
Set to '1' by hardware when a byte of data has been received by UART0 (set at the STOP bit sampling time). When the UART0 interrupt is enabled, setting this bit to '1' causes the CPU to vector to the UART0 interrupt service routine. This bit must be cleared manually by software.

25.2.4. Frequency Output Mode

Frequency Output Mode produces a programmable-frequency square wave on the module's associated CEXn pin. The capture/compare module high byte holds the number of PCA clocks to count before the output is toggled. The frequency of the square wave is then defined by Equation 25.1.

$$F_{CEXn} = \frac{F_{PCA}}{2 \times PCA0CPHn}$$

Note: A value of 0x00 in the PCA0CPHn register is equal to 256 for this equation.

Equation 25.1. Square Wave Frequency Output

Where F_{PCA} is the frequency of the clock selected by the CPS2-0 bits in the PCA mode register, PCA0MD. The lower byte of the capture/compare module is compared to the PCA counter low byte; on a match, CEXn is toggled and the offset held in the high byte is added to the matched value in PCA0CPLn. Frequency Output Mode is enabled by setting the ECOMn, TOGn, and PWMn bits in the PCA0CPMn register.

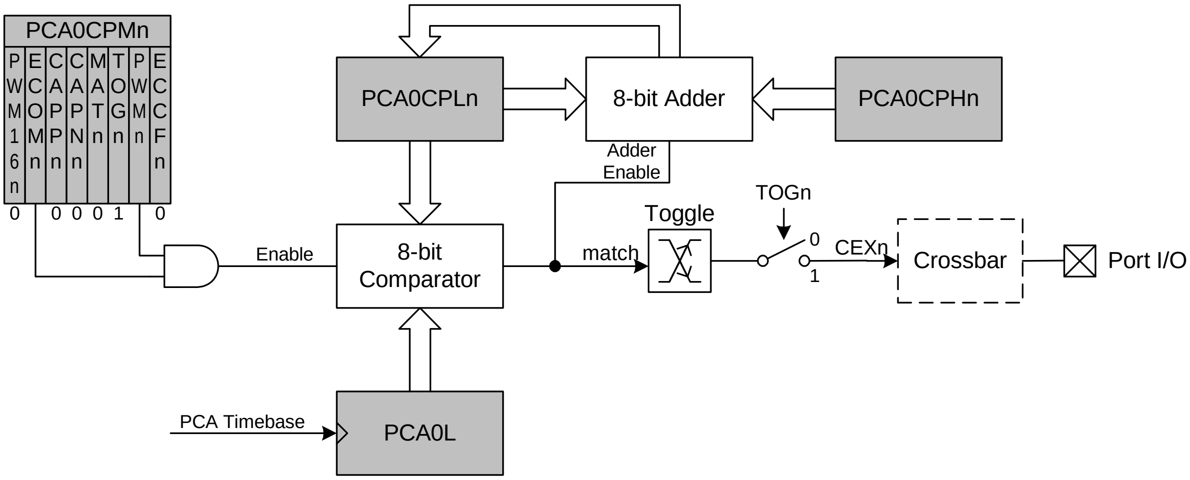


Figure 25.7. PCA Frequency Output Mode

Table 25.3. Watchdog Timer Timeout Intervals[1]

System Clock (Hz)	PCA0CPL5	Timeout Interval (ms)
24,500,000	255	32.1
24,500,000	128	16.2
24,500,000	32	4.1
18,432,000	255	42.7
18,432,000	128	21.5
18,432,000	32	5.5
11,059,200	255	71.1
11,059,200	128	35.8
11,059,200	32	9.2
3,062,500	255	257
3,062,500	128	129.5
3,062,500	32	33.1
191,406[2]	255	4109
191,406[2]	128	2070
191,406[2]	32	530
32,000	255	24576
32,000	128	12384
32,000	32	3168

Notes:

1. Assumes SYSCLK / 12 as the PCA clock source, and a PCA0L value of 0x00 at the update time.
2. Internal oscillator reset frequency.

SFR Definition 25.2. 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 5 is used as the watchdog timer.
0: Watchdog Timer disabled.
1: PCA Module 5 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	smaRTClock clock divided by 8*
1	1	1	Reserved

***Note:** External clock divided by 8 and smaRTClock clock divided by 8 are 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.

Document Change List

Revision 0.7 to Revision 0.8

- Updated specification tables with most recently available characterization data.
- Corrected references to configuring pins for Analog Mode - Port Latch must contain a '1'.
- SFR Definition 5.6: Address correction to 0xBA.
- Added Figure 8.2 showing power connection diagram without using on-chip regulator.
- Section 9 : Removed references to "High Speed Analog Mode".
- Table 11.2 : Corrected SFR Name P2MDIN on location 0xF3.
- Section 14 : Corrected operational description of CRC engine.
- Section 18, Important Note on page 151 : Added "and have the same behavior as P0 in Normal Mode." to last sentence.
- Section 19.2.2 : Inserted Step 3 "Release the crystal pins by writing ‘1's to the port latch."
- Section 19.3 : Added Figure 19.3 and text to describe behavior of clock multiplier with slower input frequencies.
- Section 21: Corrected SMBus maximum rate to 1/20th system clock.
- Table 21.4 : Made corrections to SMBus state descriptions.
- Figure 24.6 : Corrected T2RCLK Mux selection options.
- Figure 24.9 : Corrected T3RCLK Mux selection options.
- C2 Register Definition 26.2 : Corrected DEVICEID value to 0x0C.

Revision 0.8 to Revision 1.0

- Updated specification tables with full characterization data.
- Updated Flash write and erase procedures to include a write to FLSCL.3-0.
- Changed /RST pin comments in Table 4.1, “Pin Definitions for the C8051F41x,” on page 41 for the recommended pull-up resistor.
- Changed the reset value of the SFR Definition 16.3. FLSCL: Flash Scale.
- Removed the "Optional GND Connection" from Figure 4.5. ’Typical QFN-28 Landing Diagram’ on page 48.
- Added a note regarding the maximum SYSCLK frequency to SFR Definition 19.4. CLKMUL: Clock Multiplier Control.

Revision 1.0 to Revision 1.1

- Updated Figure 4.3. ’LQFP-32 Package Diagram’ on page 46, Figure 4.5. ’QFN-28 Package Drawing’ on page 48, and Figure 4.6. ’QFN-28 Recommended PCB Land Pattern’ on page 49.
- Added note that VIO must be ≥ VDD in Table 3.1, “Global DC Electrical Characteristics,” on page 36.
- Added information about ADC0 output register auto-clearing in SFR Definition 5.2.
- Corrected ADC0 Tracking time equation in SFR Definition 5.6.
- Clarified Voltage Regulator Electrical Specifications in Table 8.1 on page 82.
- Added information about 16-bit and 32-bit CRC algorithms in Section 14.