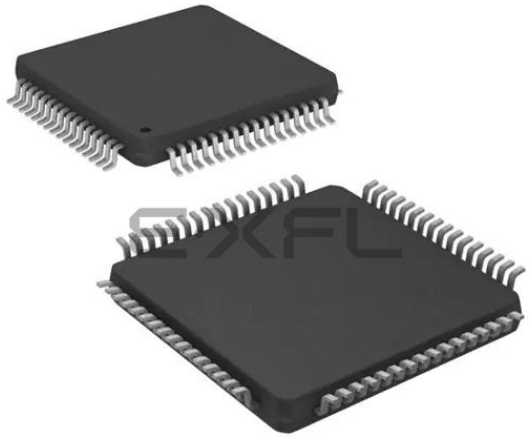


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	8051
Core Size	8-Bit
Speed	25MHz
Connectivity	SMBus (2-Wire/I²C), SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	24
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	4.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 2x16b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f067

Figure 5.11. REF0CN: Reference Control Register 0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	-	-	BIASE0	REFBE0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xD1
SFR Page: 0

Bits7-2: RESERVED. Read = 000000b; Write = 000000b.
Bit1: BIASE0: ADC0 Bias Generator Enable Bit. (Must be '1' if using ADC0).
0: ADC0 Internal Bias Generator Off.
1: ADC0 Internal Bias Generator On.
Bit0: REFBE0: Internal Reference Buffer for ADC0 Enable Bit.
0: Internal Reference Buffer for ADC0 Off. External voltage reference can be used.
1: Internal Reference Buffer for ADC0 On. Internal voltage reference is driven on the VREF0 pin.

Figure 5.12. REF1CN: Reference Control Register 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
-	-	-	-	-	-	BIASE1	REFBE1	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xD1
SFR Page: 1

Bits7-2: RESERVED. Read = 000000b; Write = 000000b.
Bit1: BIASE1: ADC1 Bias Generator Enable Bit. (Must be '1' if using ADC1).
0: ADC1 Internal Bias Generator Off.
1: ADC1 Internal Bias Generator On.
Bit0: REFBE1: Internal Reference Buffer for ADC1 Enable Bit.
0: Internal Reference Buffer for ADC1 Off. External voltage reference can be used.
1: Internal Reference Buffer for ADC1 On. Internal voltage reference is driven on the VREF1 pin.

Figure 6.14. DMA0CTH: DMA0 Repeat Counter Limit MSB Register

SFR Page: 3
SFR Address: 0xFA

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	

Bits 7-0: DMA0 Repeat Counter Limit High-Order Bits.

Figure 6.15. DMA0CTL: DMA0 Repeat Counter Limit LSB Register

SFR Page: 3
SFR Address: 0xF9

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	

Bits 7-0: DMA0 Repeat Counter Limit Low-Order Bits.

Figure 6.16. DMA0CSH: DMA0 Repeat Counter MSB Register

SFR Page: 3
SFR Address: 0xFC

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	

Bits 7-0: DMA0 Repeat Counter High-Order Bits.

Figure 6.17. DMA0CSL: DMA0 Repeat Counter LSB Register

SFR Page: 3
SFR Address: 0xFB

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	

Bits 7-0: DMA0 Repeat Counter Low-Order Bits.

7.1. Analog Multiplexer

The analog multiplexer (AMUX2) selects the inputs to the ADC, allowing any of the pins on Port 1 to be measured in single-ended mode, or as a differential pair. Additionally, the on-chip temperature sensor may be selected as a single-ended input. The ADC2 input channels are configured and selected in the AMX-2CF and AMX2SL registers as described in Figure 7.5 and Figure 7.6, respectively. In Single-ended Mode, the selected pin is measured with respect to AGND. In Differential Mode, the selected differential pair is measured with respect to one another. The polarity of the differential measurement depends on the setting of the AMX2AD3-0 bits in the AMX2SL register. For example, if pins AIN2.0 and AIN2.1 are configured for differential measurement (AIN01IC = 1), and AMX2AD3-0 = 0000b, the ADC will measure the voltage (AIN2.0 - AIN2.1). If AMX2AD3-0 is changed to 0001b, the ADC will measure the same voltage, with opposite polarity (AIN2.1 - AIN2.0).

The conversion code format differs between Single-ended and Differential modes. The registers ADC2H and ADC2L 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 AD2LJST bit (ADC2CN.0). When in Single-ended Mode, conversion codes are represented as 10-bit unsigned integers. Inputs are measured from ‘0’ to VREF * 1023/1024. Example codes are shown below for both right-justified and left-justified data. Unused bits in the ADC2H and ADC2L registers are set to ‘0’.

Input Voltage	Right-Justified ADC2H:ADC2L (AD2LJST = 0)	Left-Justified ADC2H:ADC2L (AD2LJST = 1)
VREF * 1023/1024	0x03FF	0xFFC0
VREF * 512/1024	0x0200	0x8000
VREF * 256/1024	0x0100	0x4000
0	0x0000	0x0000

When in Differential Mode, conversion codes are represented as 10-bit signed 2’s complement numbers. Inputs are measured from -VREF to VREF * 511/512. Example codes are shown below for both right-justified and left-justified data. For right-justified data, the unused MSBs of ADC2H are a sign-extension of the data word. For left-justified data, the unused LSBs in the ADC2L register are set to ‘0’.

Input Voltage	Right-Justified ADC2H:ADC2L (AD2LJST = 0)	Left-Justified ADC2H:ADC2L (AD2LJST = 1)
VREF * 511/512	0x01FF	0x7FC0
VREF * 256/512	0x0100	0x4000
0	0x0000	0x0000
-VREF * 256/512	0xFF00	0xC000
- VREF	0xFE00	0x8000

Important Note About ADC2 Input Configuration: Port 1 pins selected as ADC2 inputs should be configured as analog inputs. To configure a Port 1 pin for analog input, set to ‘1’ the corresponding bit in register P1MDIN. Port 1 pins used as ADC2 inputs will be skipped by the crossbar for peripheral assignments. See Section “18. Port Input/Output” on page 203 for more Port I/O configuration details.

The Temperature Sensor transfer function is shown in Figure 7.2 on Page 89. The output voltage (V_{TEMP}) is a single-ended input to ADC2 when the Temperature Sensor is selected by bits AMX2AD3-0 in register AMX2SL. Typical values for the Slope and Offset parameters can be found in Table 7.1.

7.3. Programmable Window Detector

The ADC Programmable Window Detector continuously compares the ADC2 output registers to user-programmed limits, and notifies the system when a desired condition is detected. This is especially effective in an interrupt-driven system, saving code space and CPU bandwidth while delivering faster system response times. The window detector interrupt flag (AD2WINT in register ADC2CN) can also be used in polled mode. The ADC2 Greater-Than (ADC2GTH, ADC2GTL) and Less-Than (ADC2LTH, ADC2LTL) registers hold the comparison values. The window detector flag can be programmed to indicate when measured data is inside or outside of the user-programmed limits, depending on the contents of the ADC2 Less-Than and ADC2 Greater-Than registers.

Figure 7.11. ADC2GTH: ADC2 Greater-Than Data High Byte Register

SFR Page: 2
SFR Address: 0xC5

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	

Bits7-0: High byte of ADC2 Greater-Than Data Word.

Figure 7.12. ADC2GTL: ADC2 Greater-Than Data Low Byte Register

SFR Page: 2
SFR Address: 0xC4

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	

Bits7-0: Low byte of ADC2 Greater-Than Data Word.

7.3.1. Window Detector In Single-Ended Mode

Figure 7.15 shows two example window comparisons for right-justified, single-ended data, with ADC2LTH:ADC2LTL = 0x0080 (128d) and ADC2GTH:ADC2GTL = 0x0040 (64d). In single-ended mode, the input voltage can range from '0' to VREF * (1023/1024) with respect to AGND, and is represented by a 10-bit unsigned integer value. In the left example, an AD2WINT interrupt will be generated if the ADC2 conversion word (ADC2H:ADC2L) is within the range defined by ADC2GTH:ADC2GTL and ADC2LTH:ADC2LTL (if 0x0040 < ADC2H:ADC2L < 0x0080). In the right example, and AD2WINT interrupt will be generated if the ADC2 conversion word is outside of the range defined by the ADC2GT and ADC2LT registers (if ADC2H:ADC2L < 0x0040 or ADC2H:ADC2L > 0x0080). Figure 7.16 shows an example using left-justified data with the same comparison values.

Figure 7.15. ADC Window Compare Example: Right-Justified Single-Ended Data

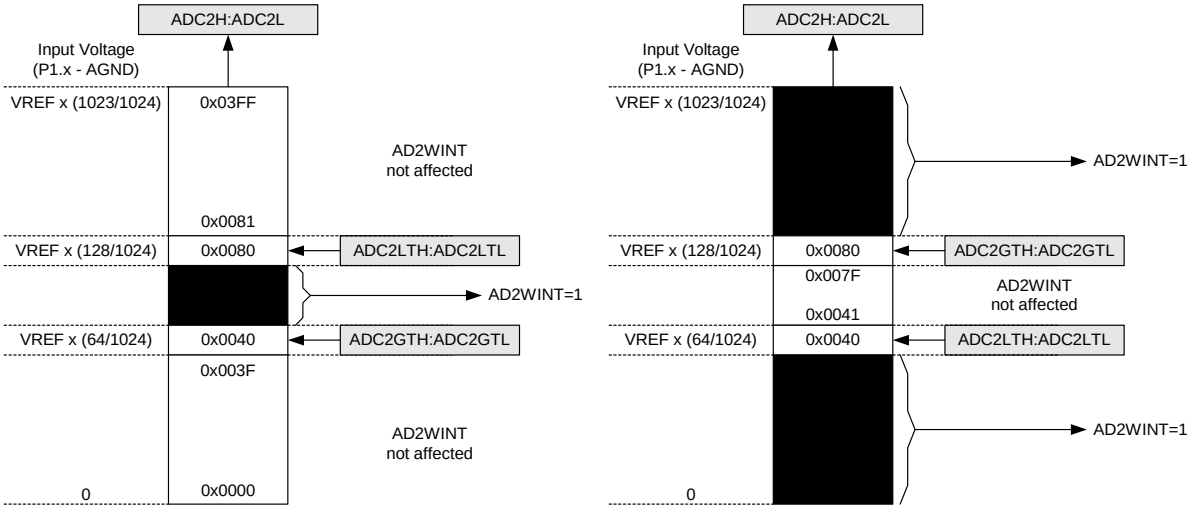


Figure 7.16. ADC Window Compare Example: Left-Justified Single-Ended Data

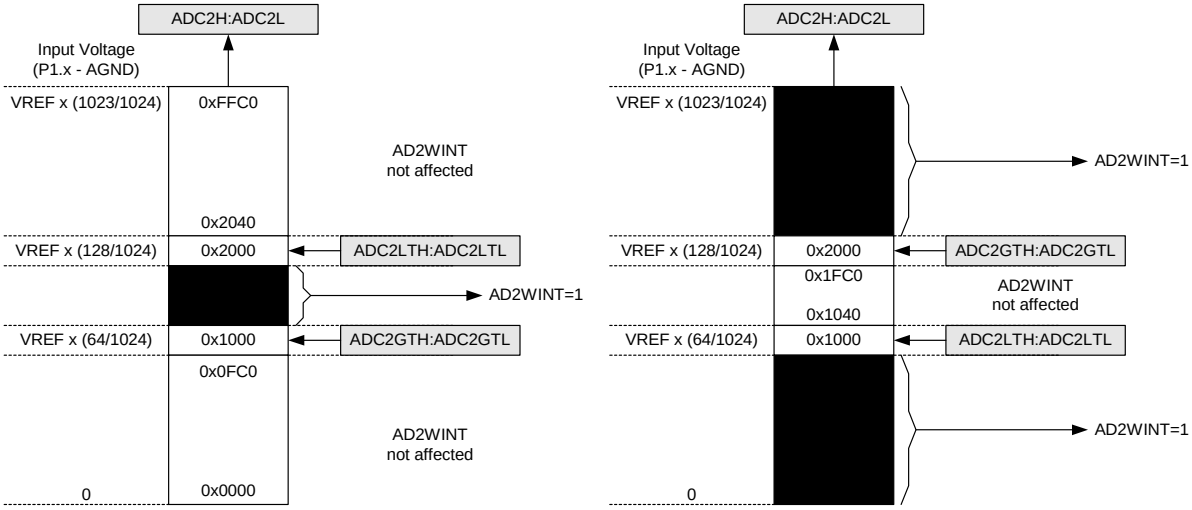


Table 7.1. ADC2 Electrical Characteristics

VDD = 3.0 V, VREF = 2.40 V (REFSL=0), PGA Gain = 1, -40°C to +85°C unless otherwise specified

Parameter	Conditions	Min	Typ	Max	Units
DC Accuracy					
Resolution			10		bits
Integral Nonlinearity			±0.5	±1	LSB
Differential Nonlinearity	Guaranteed Monotonic		±0.5	±1	LSB
Offset Error		-12	1	12	LSB
Full Scale Error	Differential mode	-15	-5	5	LSB
Offset Temperature Coefficient			3.6		ppm/°C
DYNAMIC PERFORMANCE (10 kHz sine-wave Differential input, 1 dB below Full Scale, 200 ksps)					
Signal-to-Noise Plus Distortion		53	55.5		dB
Total Harmonic Distortion	Up to the 5th harmonic		-67		dB
Spurious-Free Dynamic Range			78		dB
Conversion Rate					
SAR Conversion Clock				3	MHz
Conversion Time in SAR Clocks		10			clocks
Track/Hold Acquisition Time		300			ns
Throughput Rate				200	ksps
Analog Inputs					
ADC Input Voltage Range	Single Ended (AIN+ - AGND) Differential (AIN+ - AIN-)	0 -VREF		VREF VREF	V V
Absolute Pin Voltage with respect to AGND	Single Ended or Differential	0		AV+	V
Input Capacitance			5		pF
Temperature Sensor					
Linearity			±0.2		°C
Offset	Temp = 0 °C		776		mV
Offset Error (Note 1)	Temp = 0 °C		±8.9		mV
Slope			2.89		mV/°C
Slope Error (Note 1)			±63		µV/°C
Power Specifications					
Power Supply Current (VDD supplied to ADC2)	Operating Mode, 200 ksps		400	900	µA
Power Supply Rejection			±0.3		mV/V
Note 1: Represents one standard deviation from the mean value.					

Table 13.1. CIP-51 Instruction Set Summary (Continued)

Mnemonic	Description	Bytes	Clock Cycles
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	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
Boolean Manipulation			
CLR C	Clear Carry	1	1
CLR bit	Clear direct bit	2	2
SETB C	Set Carry	1	1
SETB bit	Set direct bit	2	2
CPL C	Complement Carry	1	1
CPL bit	Complement direct bit	2	2
ANL C, bit	AND direct bit to Carry	2	2

13.2.2. Data Memory

The CIP-51 implements 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 above 0x7F will access the SFR space. Instructions using indirect addressing above 0x7F access the upper 128 bytes of data memory. Figure 13.2 illustrates the data memory organization of the CIP-51.

13.2.3. 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 Figure 13.16). This allows fast context switching when entering subroutines and interrupt service routines. Indirect addressing modes use registers R0 and R1 as index registers.

13.2.4. 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 (a bit source or destination operand 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.

13.2.5. 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, address 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.

The MCUs also have built-in hardware for a stack record which is accessed by the debug logic. The stack record is a 32-bit shift register, where each PUSH or increment SP pushes one record bit onto the register,

Figure 13.21. EIE1: Extended Interrupt Enable 1

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
EADC0	CP2IE	CP1IE	CP0IE	EPCA0	EWADC0	ESMB0	ESPI0	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: 0xE6
SFR Page: All Pages

Bit7: EADC0: Enable ADC0 End of Conversion Interrupt.
This bit sets the masking of the ADC0 End of Conversion Interrupt.
0: Disable ADC0 Conversion Interrupt.
1: Enable interrupt requests generated by the ADC1 Conversion Interrupt.

Bit6: CP2IE: Enable Comparator (CP2) Interrupt.
This bit sets the masking of the CP2 interrupt.
0: Disable CP2 interrupts.
1: Enable interrupt requests generated by the CP2IF flag.

Bit6: CP1IE: Enable Comparator (CP1) Interrupt.
This bit sets the masking of the CP1 interrupt.
0: Disable CP1 interrupts.
1: Enable interrupt requests generated by the CP1IF flag.

Bit6: CP0IE: Enable Comparator (CP0) Interrupt.
This bit sets the masking of the CP0 interrupt.
0: Disable CP0 interrupts.
1: Enable interrupt requests generated by the CP0IF flag.

Bit3: 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.

Bit2: EWADC0: Enable Window Comparison ADC0 Interrupt.
This bit sets the masking of ADC0 Window Comparison interrupt.
0: Disable ADC0 Window Comparison Interrupt.
1: Enable Interrupt requests generated by ADC0 Window Comparisons.

Bit1: ESMB0: Enable System Management Bus (SMBus0) Interrupt.
This bit sets the masking of the SMBus interrupt.
0: Disable all SMBus interrupts.
1: Enable interrupt requests generated by the SI flag.

Bit0: ESPI0: Enable Serial Peripheral Interface (SPI0) Interrupt.
This bit sets the masking of SPI0 interrupt.
0: Disable all SPI0 interrupts.
1: Enable Interrupt requests generated by the SPI0 flag.

Step 8. Clear the PSWE bit to redirect MOVX write commands to the XRAM data space.

Step 9. Re-enable interrupts.

Write/Erase timing is automatically controlled by hardware. Note that code execution in the 8051 is stalled while the Flash is being programmed or erased.

Table 16.1. Flash Electrical Characteristics

Parameter	Conditions	Min	Typ	Max	Units
Flash Size *	C8051F060/1/2/3/4/5		65664 †		Bytes
Flash Size *	C8051F066/7		32896		Bytes
Endurance		20 k	100 k		Erase/Write
Erase Cycle Time		10	12	14	ms
Write Cycle Time		40	50	60	µs

* Includes 128-byte Scratch Pad Area

† 1024 Bytes at location 0xFC00 to 0xFFFF are reserved.

16.2. Non-volatile Data Storage

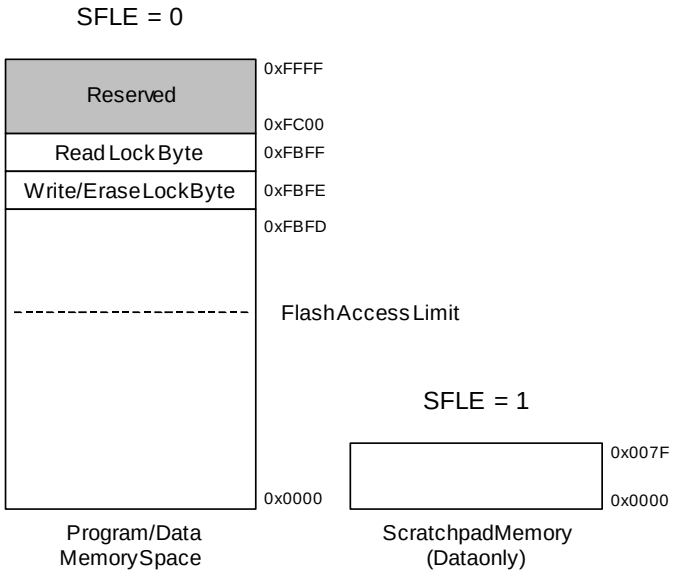
The Flash memory can be used for non-volatile data storage as well as program code. This allows data such as calibration coefficients to be calculated and stored at run time. Data is written using the MOVX write instruction (as described in the previous section) and read using the MOVC instruction.

An additional 128-byte sector of Flash memory is included for non-volatile data storage. Its smaller sector size makes it particularly well suited as general purpose, non-volatile scratchpad memory. Even though Flash memory can be written a single byte at a time, an entire sector must be erased first. In order to change a single byte of a multi-byte data set, the data must be moved to temporary storage. The 128-byte sector size facilitates updating data without wasting program memory or RAM space. The 128-byte sector is double-mapped over the normal Flash memory area; its address ranges from 0x00 to 0x7F (see Figure 16.1 and Figure 16.2). To access this 128-byte sector, the SFLE bit in PSCTL must be set to logic 1. Code execution from this 128-byte scratchpad sector is not supported.

Figure 16.1. C8051F060/1/2/3/4/5 Flash Program Memory Map and Security Bytes

Read and Write/Erase Security Bits
(Bit 7 is MSB)

Bit	Memory Block
7	0xE000-0xFBFD
6	0xC000-0xDFFF
5	0xA000-0xBFFF
4	0x8000-0x9FFF
3	0x6000-0x7FFF
2	0x4000-0x5FFF
1	0x2000-0x3FFF
0	0x0000-0x1FFF



Flash Read Lock Byte

Bits7-0: Each bit locks a corresponding block of memory. (Bit7 is MSB).

0: Read operations are locked (disabled) for corresponding block across the JTAG interface.

1: Read operations are unlocked (enabled) for corresponding block across the JTAG interface.

Flash Write/Erase Lock Byte

Bits7-0: Each bit locks a corresponding block of memory.

0: Write/Erase operations are locked (disabled) for corresponding block across the JTAG interface.

1: Write/Erase operations are unlocked (enabled) for corresponding block across the JTAG interface.

NOTE: When the block containing the security bytes is locked, the security bytes may be written but not erased.

Flash Access Limit

The Flash Access Limit is defined by the setting of the FLACL register, as described in Figure 16.3. Firmware running at or above this address is prohibited from using the MOVX and MOVC instructions to read, write, or erase Flash locations below this address.

eral's enable bits are not set to a logic 1, then its ports are not accessible at the Port pins of the device. Also note that the Crossbar assigns pins to all associated functions when the SMBus, UART0 or UART1 are selected (i.e. SMBus, SPI, UART). It would be impossible, for example, to assign TX0 to a Port pin without assigning RX0 as well. The SPI can operate in 3 or 4-wire mode (with or without NSS). Each combination of enabled peripherals results in a unique device pinout.

All Port pins on Ports 0 through 3 that are not allocated by the Crossbar can be accessed as General-Purpose I/O (GPIO) pins by reading and writing the associated Port Data registers (See Figure 18.9, Figure 18.11, Figure 18.14, and Figure 18.17), a set of SFRs which are both byte- and bit-addressable. The output states of Port pins that are allocated by the Crossbar are controlled by the digital peripheral that is mapped to those pins. Writes to the Port Data registers (or associated Port bits) will have no effect on the states of these pins.

A Read of a Port Data register (or Port bit) will always return the logic state present at the pin itself, regardless of whether the Crossbar has allocated the pin for peripheral use or not. An exception to this occurs during the execution of a *read-modify-write* instruction (ANL, ORL, XRL, CPL, INC, DEC, DJNZ, JBC, CLR, SETB, and the bitwise MOV write operation). During the *read* cycle of the *read-modify-write* instruction, it is the contents of the Port Data register, not the state of the Port pins themselves, which is read.

Because the Crossbar registers affect the pinout of the peripherals of the device, they are typically configured in the initialization code of the system before the peripherals themselves are configured. Once configured, the Crossbar registers are typically left alone.

Once the Crossbar registers have been properly configured, the Crossbar is enabled by setting XBARE (XBR2.4) to a logic 1. **Until XBARE is set to a logic 1, the output drivers on Ports 0 through 3 are explicitly disabled in order to prevent possible contention on the Port pins while the Crossbar registers and other registers which can affect the device pinout are being written**.

The output drivers on Crossbar-assigned input signals (like RX0, for example) are explicitly disabled; thus the values of the Port Data registers and the PnMDOUT registers have no effect on the states of these pins.

18.1.2. Configuring the Output Modes of the Port Pins

The output drivers on Ports 0 through 3 remain disabled until the Crossbar is enabled by setting XBARE (XBR2.4) to a logic 1.

The output mode of each port pin can be configured to be either Open-Drain or Push-Pull. In the Push-Pull configuration, writing a logic 0 to the associated bit in the Port Data register will cause the Port pin to be driven to GND, and writing a logic 1 will cause the Port pin to be driven to VDD. In the Open-Drain configuration, writing a logic 0 to the associated bit in the Port Data register will cause the Port pin to be driven to GND, and a logic 1 will cause the Port pin to assume a high-impedance state. The Open-Drain configuration is useful to prevent contention between devices in systems where the Port pin participates in a shared interconnection in which multiple outputs are connected to the same physical wire (like the SDA signal on an SMBus connection).

The output modes of the Port pins on Ports 0 through 3 are determined by the bits in the associated PnMDOUT registers (See Figure 18.10, Figure 18.13, Figure 18.16, and Figure 18.18). For example, a logic 1 in P3MDOUT.7 will configure the output mode of P3.7 to Push-Pull; a logic 0 in P3MDOUT.7 will configure the output mode of P3.7 to Open-Drain. All Port pins default to Open-Drain output.

Figure 18.25. P7: Port7 Data Register

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
P7.7	P7.6	P7.5	P7.4	P7.3	P7.2	P7.1	P7.0	11111111
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Bit Addressable

SFR Address: 0xF8
SFR Page: F

Bits7-0: P7.[7:0]: Port7 Output Latch Bits.
Write - Output appears on I/O pins.
0: Logic Low Output.
1: Logic High Output (open, if corresponding P7MDOUT bit = 0). See Figure 18.26.
Read - Returns states of I/O pins.
0: P7.n pin is logic low.
1: P7.n pin is logic high.

Note: P7.[7:0] can be driven by the External Data Memory Interface (as AD[7:0] in Multiplexed mode, or as D[7:0] in Non-multiplexed mode). See Section “17. External Data Memory Interface and On-Chip XRAM” on page 187 for more information about the External Memory Interface.

Figure 18.26. P7MDOUT: Port7 Output Mode Register

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: 0x9F
SFR Page: F

Bits7-0: P7MDOUT.[7:0]: Port7 Output Mode Bits.
0: Port Pin output mode is configured as Open-Drain.
1: Port Pin output mode is configured as Push-Pull.

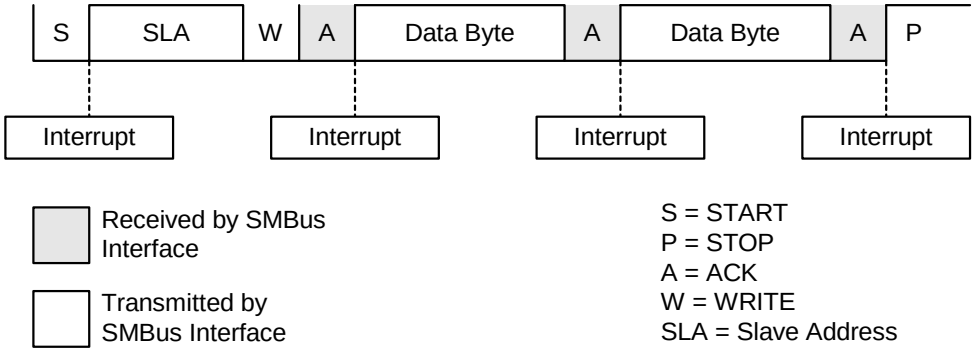
20.3. SMBus Transfer Modes

The SMBus0 interface may be configured to operate as a master and/or a slave. At any particular time, the interface will be operating in one of the following modes: Master Transmitter, Master Receiver, Slave Transmitter, or Slave Receiver. See Table 20.1 for transfer mode status decoding using the SMB0STA status register. The following mode descriptions illustrate an interrupt-driven SMBus0 application; SMBus0 may alternatively be operated in polled mode.

20.3.1. Master Transmitter Mode

Serial data is transmitted on SDA while the serial clock is output on SCL. SMBus0 generates a START condition and then transmits the first byte containing the address of the target slave device and the data direction bit. In this case the data direction bit (R/W) will be logic 0 to indicate a "WRITE" operation. The SMBus0 interface transmits one or more bytes of serial data, waiting for an acknowledge (ACK) from the slave after each byte. To indicate the end of the serial transfer, SMBus0 generates a STOP condition.

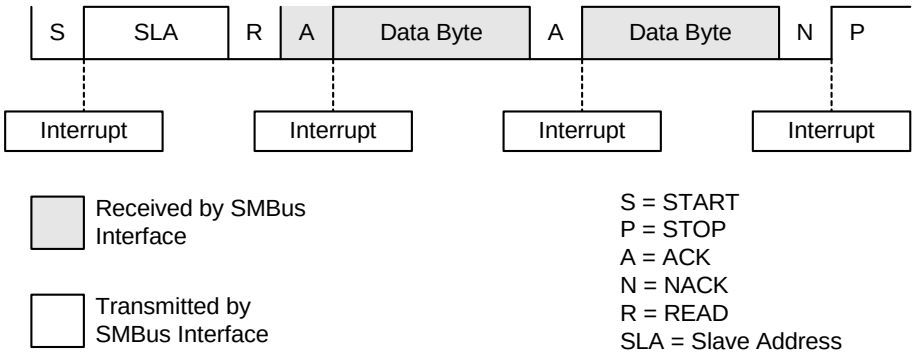
Figure 20.4. Typical Master Transmitter Sequence



20.3.2. Master Receiver Mode

Serial data is received on SDA while the serial clock is output on SCL. The SMBus0 interface generates a START followed by the first data byte containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic 1 to indicate a "READ" operation. The SMBus0 interface receives serial data from the slave and generates the clock on SCL. After each byte is received, SMBus0 generates an ACK or NACK depending on the state of the AA bit in register SMB0CN. SMBus0 generates a STOP condition to indicate the end of the serial transfer.

Figure 20.5. Typical Master Receiver Sequence



21.1. Signal Descriptions

The four signals used by SPI0 (MOSI, MISO, SCK, NSS) are described below.

21.1.1. Master Out, Slave In (MOSI)

The master-out, slave-in (MOSI) signal is an output from a master device and an input to slave devices. It is used to serially transfer data from the master to the slave. This signal is an output when SPI0 is operating as a master and an input when SPI0 is operating as a slave. Data is transferred most-significant bit first. When configured as a master, MOSI is driven by the MSB of the shift register in both 3- and 4-wire mode.

21.1.2. Master In, Slave Out (MISO)

The master-in, slave-out (MISO) signal is an output from a slave device and an input to the master device. It is used to serially transfer data from the slave to the master. This signal is an input when SPI0 is operating as a master and an output when SPI0 is operating as a slave. Data is transferred most-significant bit first. The MISO pin is placed in a high-impedance state when the SPI module is disabled and when the SPI operates in 4-wire mode as a slave that is not selected. When acting as a slave in 3-wire mode, MISO is always driven by the MSB of the shift register.

21.1.3. Serial Clock (SCK)

The serial clock (SCK) signal is an output from the master device and an input to slave devices. It is used to synchronize the transfer of data between the master and slave on the MOSI and MISO lines. SPI0 generates this signal when operating as a master. The SCK signal is ignored by a SPI slave when the slave is not selected (NSS = 1) in 4-wire slave mode.

21.1.4. Slave Select (NSS)

The function of the slave-select (NSS) signal is dependent on the setting of the NSSMD1 and NSSMD0 bits in the SPI0CN register. There are three possible modes that can be selected with these bits:

1. NSSMD[1:0] = 00: 3-Wire Master or 3-Wire Slave Mode: SPI0 operates in 3-wire mode, and NSS is disabled. When operating as a slave device, SPI0 is always selected in 3-wire mode. Since no select signal is present, SPI0 must be the only slave on the bus in 3-wire mode. This is intended for point-to-point communication between a master and one slave.
2. NSSMD[1:0] = 01: 4-Wire Slave or Multi-Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an input. When operating as a slave, NSS selects the SPI0 device. When operating as a master, a 1-to-0 transition of the NSS signal disables the master function of SPI0 so that multiple master devices can be used on the same SPI bus.
3. NSSMD[1:0] = 1x: 4-Wire Master Mode: SPI0 operates in 4-wire mode, and NSS is enabled as an output. The setting of NSSMD0 determines what logic level the NSS pin will output. This configuration should only be used when operating SPI0 as a master device.

See Figure 21.2, Figure 21.3, and Figure 21.4 for typical connection diagrams of the various operational modes. **Note that the setting of NSSMD bits affects the pinout of the device.** When in 3-wire master or 3-wire slave mode, the NSS pin will not be mapped by the crossbar. In all other modes, the NSS signal will be mapped to a pin on the device. See Section “18. Port Input/Output” on page 203 for general purpose port I/O and crossbar information.

24.2. Timer 2, Timer 3, and Timer 4

Timers 2, 3, and 4 are 16-bit counter/timers, each formed by two 8-bit SFRs: TMRnL (low byte) and TMRnH (high byte) where n = 2, 3, and 4 for timers 2, 3, and 4 respectively. These timers feature auto-reload, capture, and toggle output modes with the ability to count up or down. Capture Mode and Auto-reload mode are selected using bits in the Timer 2, 3, and 4 Control registers (TMRnCN). Toggle output mode is selected using the Timer 2, 3, and 4 Configuration registers (TMRnCF). These timers may also be used to generate a square-wave at an external pin. Timers 2, 3, and 4 can use either the system clock (divided by one, two, or twelve), external clock (divided by eight) or transitions on an external input pin as its clock source. Timer 2 and 3 can be used to start an ADC Data Conversion and Timers 2, 3, and 4 can schedule DAC outputs. Timers 1, 2, 3, or 4 may be used to generate baud rates for UART 0. Only Timer 1 can be used to generate baud rates for UART 1.

The Counter/Timer Select bit C/Tn bit (TMRnCN.1) configures the peripheral as a counter or timer. Clearing C/Tn configures the Timer to be in a timer mode (i.e., the selected timer clock source as the input for the timer). When C/Tn is set to 1, the timer is configured as a counter (i.e., high-to-low transitions at the Tn input pin increment (or decrement) the counter/timer register. Refer to Section “18.1. Ports 0 through 3 and the Priority Crossbar Decoder” on page 205 for information on selecting and configuring external I/O pins for digital peripherals, such as the Tn pin.

Timer 2, 3, and 4 can use either SYSCLK, SYSCLK divided by 2, SYSCLK divided by 12, an external clock divided by 8, or high-to-low transitions on the Tn input pin as its clock source when operating in Counter/Timer with Capture mode. Clearing the C/Tn bit (TnCON.1) selects the system clock/external clock as the input for the timer. The Timer Clock Select bits TnM0 and TnM1 in TMRnCF can be used to select the system clock undivided, system clock divided by two, system clock divided by 12, or an external clock provided at the XTAL1/XTAL2 pins divided by 8 (see Figure 24.14). When C/Tn is set to logic 1, a high-to-low transition at the Tn input pin increments the counter/timer register (i.e., configured as a counter).

24.2.1. Configuring Timer 2, 3, and 4 to Count Down

Timers 2, 3, and 4 have the ability to count down. When the timer’s respective Decrement Enable Bit (DCENn) in the Timer Configuration Register (See Figure 24.14) is set to ‘1’, the timer can then count *up* or *down*. When DCENn = 1, the direction of the timer’s count is controlled by the TnEX pin’s logic level. When TnEX = 1, the counter/timer will count up; when TnEX = 0, the counter/timer will count down. To use this feature, TnEX must be enabled in the digital crossbar and configured as a digital input.

Note: When DCENn = 1, other functions of the TnEX input (i.e., capture and auto-reload) are not available. TnEX will only control the direction of the timer when DCENn = 1.

Figure 25.12. PCA0CPMn: PCA0 Capture/Compare Mode Registers

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	Reset Value
PWM16n	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn	00000000
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	

SFR Address: PCA0CPM0: 0xDA, PCA0CPM1: 0xDB, PCA0CPM2: 0xDC, PCA0CPM3: 0xDD, PCA0CPM4: 0xDE, PCA0CPM5: 0xDF

SFR Page: PCA0CPM0: page 0, PCA0CPM1: page 0, PCA0CPM2: page 0, PCA0CPM3: 0, PCA0CPM4: page 0, PCA0CPM5: page 0

Bit7: PWM16n: 16-bit Pulse Width Modulation Enable.
This bit selects 16-bit mode when Pulse Width Modulation mode is enabled (PWMn = 1).
0: 8-bit PWM selected.
1: 16-bit PWM selected.

Bit6: ECOMn: Comparator Function Enable.
This bit enables/disables the comparator function for PCA0 module n.
0: Disabled.
1: Enabled.

Bit5: CAPPn: Capture Positive Function Enable.
This bit enables/disables the positive edge capture for PCA0 module n.
0: Disabled.
1: Enabled.

Bit4: CAPNn: Capture Negative Function Enable.
This bit enables/disables the negative edge capture for PCA0 module n.
0: Disabled.
1: Enabled.

Bit3: MATn: Match Function Enable.
This bit enables/disables the match function for PCA0 module n. 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.
0: Disabled.
1: Enabled.

Bit2: TOGn: Toggle Function Enable.
This bit enables/disables the toggle function for PCA0 module n. 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.
0: Disabled.
1: Enabled.

Bit1: PWMn: Pulse Width Modulation Mode Enable.
This bit enables/disables the PWM function for PCA0 module n. When enabled, a pulse width modulated signal is output on the CEXn pin. 8-bit PWM is used if PWM16n is logic 0; 16-bit mode is used if PWM16n logic 1. If the TOGn bit is also set, the module operates in Frequency Output Mode.
0: Disabled.
1: Enabled.

Bit0: 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.

Table 26.2. Boundary Data Register Bit Definitions (C8051F061/3/5/7)

EXTEST provides access to both capture and update actions, while Sample only performs a capture.

Bit	Action	Target
0	Capture	Not used
	Update	Not used
1	Capture	Not used
	Update	Not used
2	Capture	CAN RX Output Enable to pin
	Update	CAN RX Output Enable to pin
3	Capture	CAN RX Input from pin
	Update	CAN RX Output to pin
4	Capture	CAN TX Output Enable to pin
	Update	CAN TX Output Enable to pin
5	Capture	CAN TX Input from pin
	Update	CAN TX Output to pin
6	Capture	External Clock from XTAL1 pin
	Update	Not used
7	Capture	Weak Pullup Enable from MCU
	Update	Weak Pullup Enable to Port Pins
8, 10, 12, 14, 16, 18, 20, 22	Capture	P0.n output enable from MCU (e.g. Bit 8 = P0.0, Bit 10 = P0.1, etc.)
	Update	P0.n output enable to pin (e.g. Bit 8 = P0.0oe, Bit 10 = P0.1oe, etc.)
9, 11, 13, 15, 17, 19, 21, 23	Capture	P0.n input from pin (e.g. Bit 9 = P0.0, Bit 11 = P0.1, etc.)
	Update	P0.n output to pin (e.g. Bit 9 = P0.0, Bit 11 = P0.1, etc.)
24, 26, 28, 30, 32, 34, 36, 38	Capture	P1.n output enable from MCU (follows P0.n numbering scheme)
	Update	P1.n output enable to pin (follows P0.n numbering scheme)
25, 27, 29, 31, 33, 35, 37, 39	Capture	P1.n input from pin (follows P0.n numbering scheme)
	Update	P1.n output to pin (follows P0.n numbering scheme)
40, 42, 44, 46, 48, 50, 52, 54	Capture	P2.n output enable from MCU (follows P0.n numbering scheme)
	Update	P2.n output enable to pin (follows P0.n numbering scheme)
41, 43, 45, 47, 49, 51, 53, 55	Capture	P2.n input from pin (follows P0.n numbering scheme)
	Update	P2.n output to pin (follows P0.n numbering scheme)
56, 58, 60, 62, 64, 66, 68, 70	Capture	P3.n output enable from MCU (follows P0.n numbering scheme)
	Update	P3.n output enable to pin (follows P0.n numbering scheme)
57, 59, 61, 63, 65, 67, 69, 71	Capture	P3.n input from pin (follows P0.n numbering scheme)
	Update	P3.n output to pin (follows P0.n numbering scheme)
72	Capture	Reset Enable from MCU
	Update	Reset Enable to /RST pin
73	Capture	Reset Input from /RST pin
	Update	Not used
74, 76, 78, 80, 82, 84	Capture	P5.0, P5.1, P5.2, P5.3, P5.5, P5.7 (respectively) output enable from MCU†
	Update	P5.0, P5.1, P5.2, P5.3, P5.5, P5.7 (respectively) output enable to pin†
75, 77, 79, 81, 83, 85	Capture	P5.0, P5.1, P5.2, P5.3, P5.5, P5.7 (respectively) input from pin†
	Update	P5.0, P5.1, P5.2, P5.3, P5.5, P5.7 (respectively) output to pin†
86, 88, 90, 92, 94, 96, 98, 100	Capture	P6.n output enable from MCU (follows P0.n numbering scheme)†
	Update	P6.n output enable to pin (follows P0.n numbering scheme)†