
# TM52F2384

## *DATA SHEET*

## *Rev 1.4*

# AMENDMENT HISTORY

| Version | Date | Description |
|---|---|---|
| V1.0 | 2023/8/15 | New release. |
| V1.1 | 2023/9/22 | Operation voltage modify<br>Other details |
| V1.2 | 2024/5/30 | 1. Operation voltage modify (p22)<br>2. FW must force ADCSRV=0 before enter Stop/Halt/Idle mode.(p69) |
| V1.3 | 2024/6/14 | Add QFN-48 package type |
| V1.4 | 2024/7/3 | LCD pump start-up note |

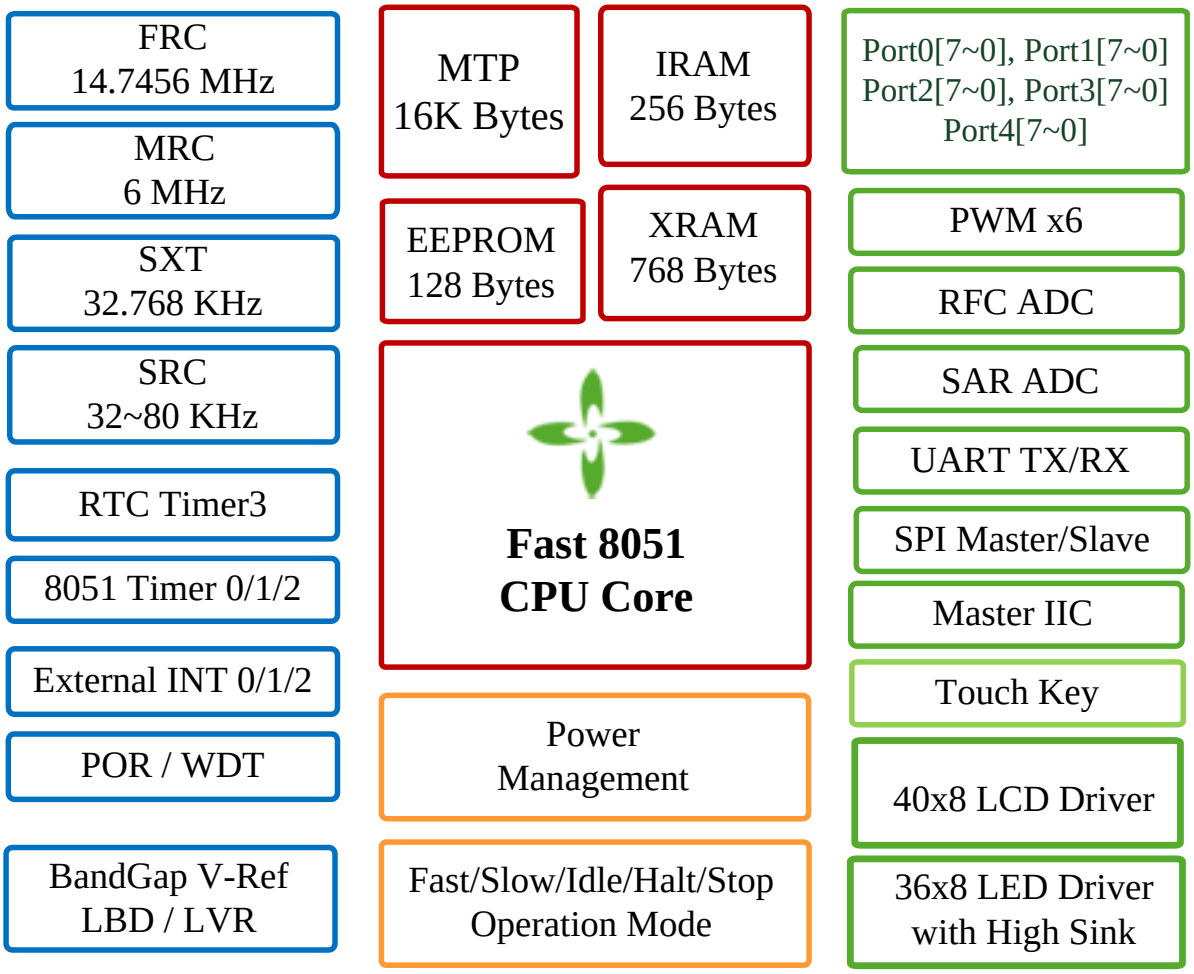
# CONTENTS

## GENERAL DESCRIPTION

**TM52-F2384** is a version of a new, fast 8051 architecture for an 8-bit microcontroller single chip with an instruction set fully compatible with industry standard 8051, C language development platform, and retains most 8051 peripheral's functional block. Typically, the **TM52-F2384** executes instructions six times faster than the standard 8051 architecture.

The **TM52-F2384** provides improved performance, lower cost and fast time-to-market by integrating features on the chip, including 16K Bytes MTP program memory, 128 Bytes EEPROM, 1024 Bytes SRAM, Low Voltage Reset (LVR), Low Battery Detector (LBD), dual clock power saving operation mode, SPI Interface, Master I2C Interface, 8051 standard UART and Timer0/1/2, adjustable real time clock Timer3, LCD/LED Driver, Touch Key, 12-bit SAR ADC, 6 set 8-bit PWM, Resistance to Frequency Converter (RFC) and Watchdog Timer. Its high reliability and low power consumption feature can be widely applied in consumer, industry and home appliance products.

## BLOCK DIAGRAM




**TM52F2384**

# FEATURES

1. **Standard 8051 Instruction set, fast machine cycle**
   - Executes instructions six times faster than the standard 8051.
2. **16K Bytes MTP Program Memory**
   - Support “In Circuit Programming” (ICP) or “In System Programming” (ISP) for the MTP code
   - Support Byte Write “In Application Programming” (IAP) mode.
   - 10000 write cycles & 10 years data retention
3. **128 Bytes EEPROM**
   - 50,000 write cycles & 10 years data retention
4. **Total 1280 Bytes SRAM (IRAM + XRAM)**
   - 256 Bytes IRAM in the 8051 internal data memory area
   - 1024 Bytes XRAM in the 8051 external data memory area (accessed by MOVX Instruction)
5. **Five System Clock type Selections**
   - Fast clock from Internal Fast RC (FRC, 14.7456 MHz)
   - Fast clock from Internal Medium RC (MRC, 6MHz @$V_{DD}$ = 3V, 2.3MHz @$V_{DD}$ = 1.5V)
   - Fast clock from External RC (RFC)
   - Slow clock from Slow Crystal (SXT, 32768Hz)
   - Slow clock from Internal Slow RC (SRC, 75KHz @$V_{DD}$ = 3V, 35KHz @$V_{DD}$ = 1.5V)
   - System Clock can be divided by 1/2/4/16 option
6. **8051 Standard Timer – Timer0 / 1 / 2**
   - 16-bit Timer0, also supports RFC or SXT/16 clock input counting
   - 16-bit Timer1, also supports SXT/16 clock input counting
   - 16-bit Timer2, also supports SXT/16 clock input counting
7. **23-bit Timer3 used for Real Time 32768Hz Crystal counting**
   - ± 0.5 ppm ~ 61 ppm interrupt rate adjustable
   - MSB 8-bit overflow auto-reload
   - 16ms ~1.0 sec or overflow Interrupt
8. **10-Channel Touch Key**
9. **12-bit SAR ADC**
10. **Resistance to Frequency Converter (RFC)**
    - RFC can be used for Temperature or Humidity sensor
    - RFC clock can be used as System clock source

**11. 8051 Standard UART**

- Support One Wire UART
- Extra Baud rate generator
- Can use P3.0/P3.1 or P1.2/1.3 pins

**12. SPI Interface**

- Master or Slave mode selectable
- Programmable transmit bit rate
- Serial clock phase and polarity options
- MSB-first or LSB-first selectable

**13. Master I2C Interface**

**14. 14-Sources, 4-level priority Interrupt**

- Timer0 / Timer1 / Timer2 / Timer3 Interrupt
- INT0 / INT1 Falling-Edge / Low-Level Interrupt
- Pin Change Interrupt
- P2.7 (INT2) Interrupt
- SPI / I2C / UART Interrupt
- Touch Key / ADC Interrupt
- PWM5 Interrupt
- LBD Interrupt

**15. Pin Interrupt can Wake up CPU from Power-Down (Stop) mode**

- P3.2 / P3.3 / P2.7 (INT0 / INT1 / INT2) Interrupt & Wake-up
- Port1/2/3 pin can be defined as Interrupt & Wake-up pin (by pin change)

**16. Max. 40 Programmable I/O pins**

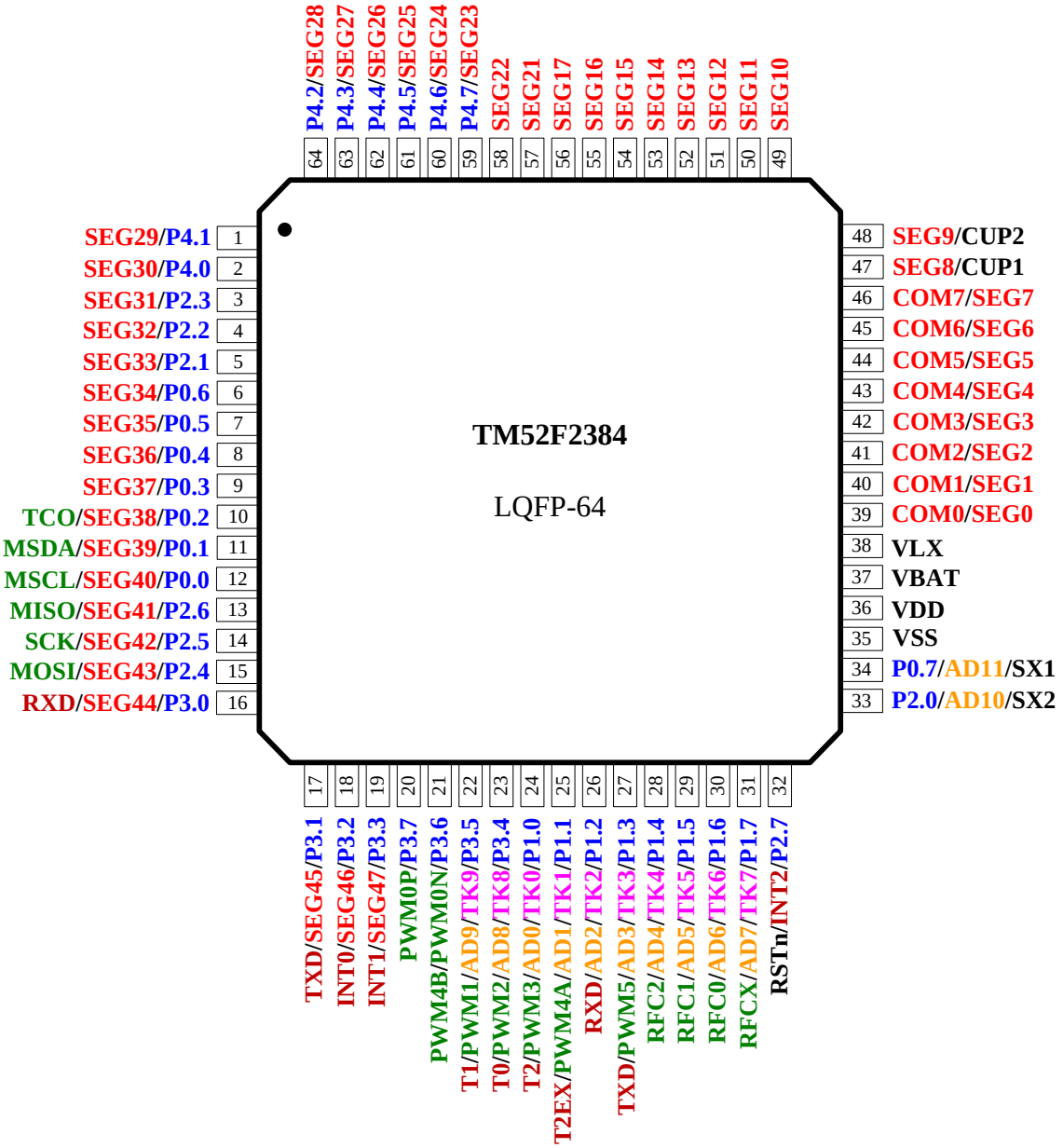
- CMOS Output
- Pseudo-Open-Drain, or Open-Drain Output
- Schmitt Trigger Input
- Pin Pull-up can be Enabled or Disabled

**17. LCD Controller / Driver**

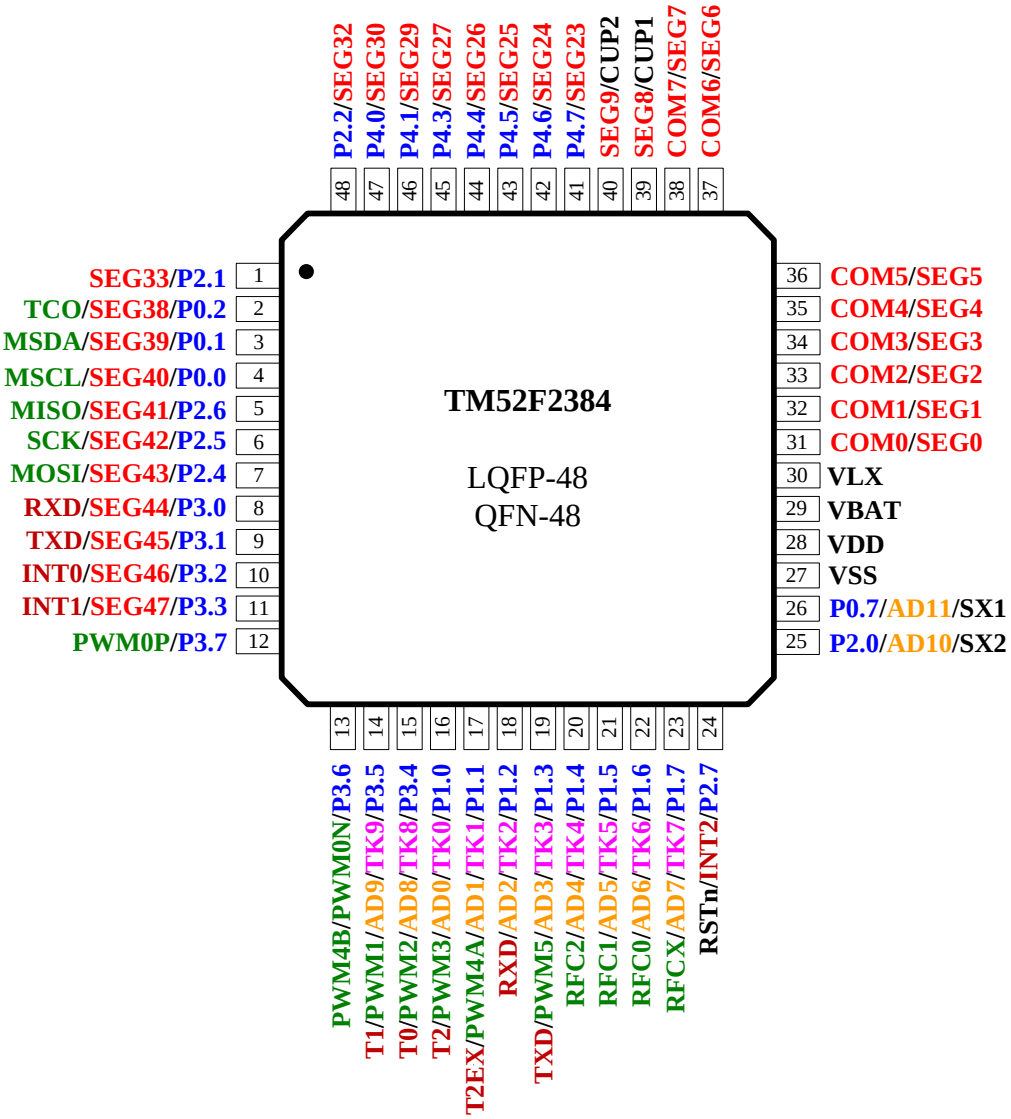
- 1/3 ~ 1/8 Duty
- 4 COM x 44 SEG ~ 8 COM x 40 SEG selectable
- 1/3 LCD Bias voltage, VL1 = VLCD/3, VL2 = VLCD*2/3, VL3 = VLCD
  - PUMP=0: VLCD (VL3) = $V_{BAT}$*3/5 ~ $V_{BAT}$*5/5 (16 steps Brightness level)
  - PUMP=1: VLCD (VL3) = $V_{BAT}$*1.2 ~ $V_{BAT}$*2 (16 steps Brightness level)
- 1/2 LCD Bias voltage, PUMP=1: VL1 = $V_{BAT}$, VL2 = VLCD = $V_{BAT}$*2
- Frame Rate: 40~90Hz

18. **LED Controller / Driver**
    - Max. 8 COM x 36 SEG
    - 60 mA High Sink COM, Active Low
    - Dot Matrix Mode (DMX), up to 8 x 7 = 56 dots
19. **BandGap Voltage Reference for Low Battery Detection (LBD)**
    - Detect $V_{BAT}$ voltage level from 1.8V to 3.7V
20. **Built-in tiny current LDO Regulator for chip internal power supply ($V_{DD}$)**
    - $V_{DD}$ voltage level can be set to 0.375*$V_{BAT}$ ~ 0.725*$V_{BAT}$ for power saving
    - Must set $V_{DD}$ > 1.4V
21. **Watch Dog Timer based on Slow Clock**
22. **CRC Code check**
23. **6 set 8-bit PWM**
    - Adjustable Period & Clock Pre-scale
    - PWM0P / PWM0N support Pump Voltage Drive
    - PWM1 with 300mA sink current capability
    - PWM5 can generate Interrupt
24. **Five types Reset**
    - Power on Reset (1.1V or 1.7V)
    - Selectable External Pin Reset
    - Selectable Watch Dog Reset
    - Software Command Reset
    - Low Voltage Reset (LVR, 1.7V ~ 3.6V)
25. **Five types Operation Mode**
    - Fast / Slow / Idle / Halt / Stop mode
26. **On-chip Debug / ICE interface**
    - Use P1.2 / P1.3 pin, share with ICP programming pin
27. **Operating Voltage and Current**
    - $V_{BAT}$ = 1.7V ~ 5.5V (25℃) (> 2V for ADC, >3V for EEPROM)
    - Total 3.2uA Halt mode Current with LCD on @$V_{BAT}$ = 3V, $V_{DD}$ = 1.5V
28. **Operating Temperature Range**
    - –40℃ ~ +105℃
29. **64/48 pin LQFP Package, 48-QFN Package**

# PIN ASSIGNMENT

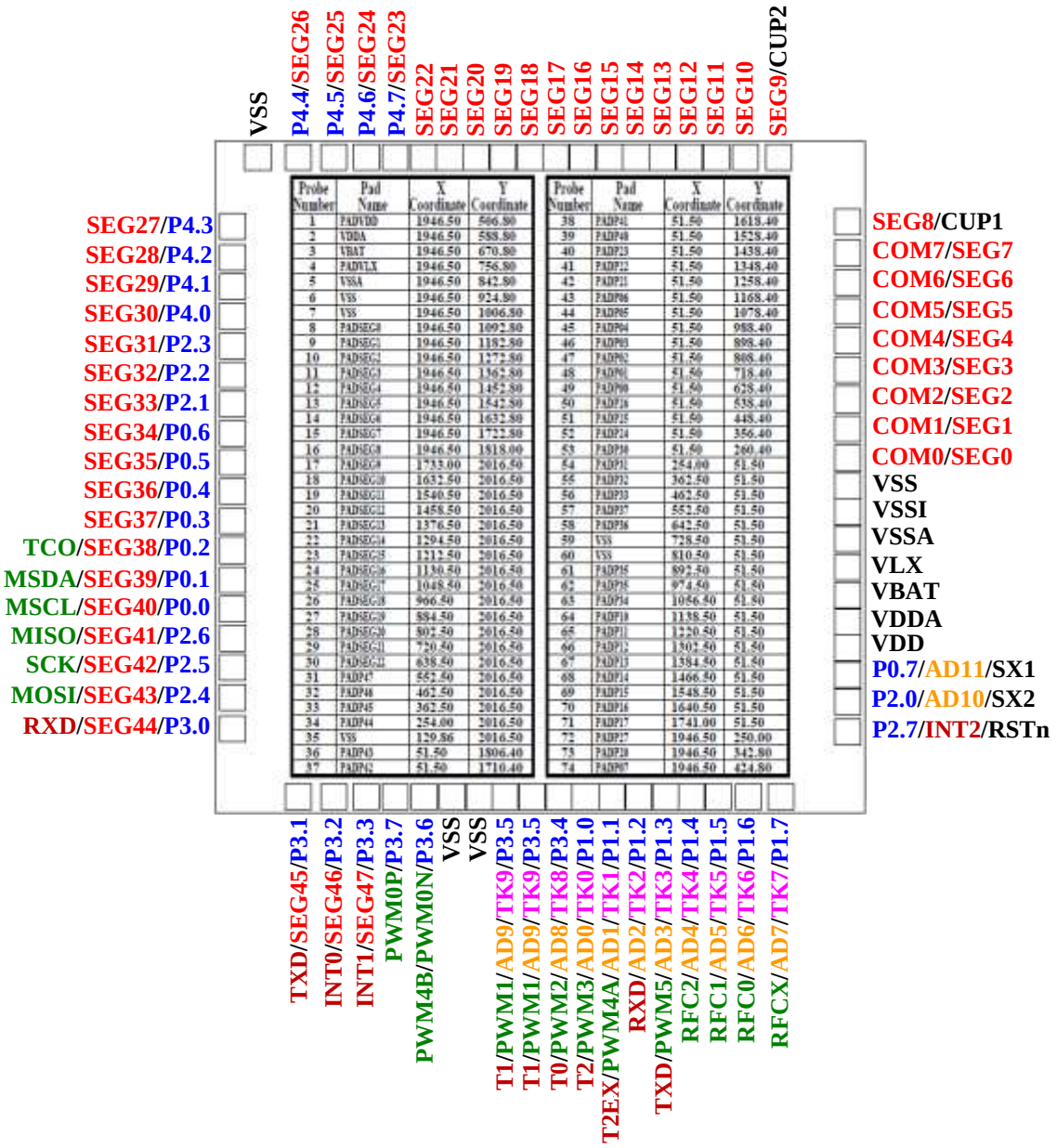



***Note***: SEG44~47 only support LCD mode, does not support LED mode.
***Note***: For low power application, all digital IOs (including unbounded or unused) should avoid high-impedance settings.




***Note***: SEG44~47 only support LCD mode, does not support LED mode.
***Note***: For low power application, all digital IOs (including unbounded or unused) should avoid high-impedance settings.

# DIE PAD LIST




| Probe Number | Pad Name | X Coordinate | Y Coordinate |
|---|---|---|---|
| 1 | PADVDD | 1946.50 | 606.80 |
| 2 | VDDA | 1946.50 | 588.80 |
| 3 | VBAT | 1946.50 | 670.80 |
| 4 | PADVLX | 1946.50 | 756.80 |
| 5 | VSSA | 1946.50 | 842.80 |
| 6 | VSS | 1946.50 | 924.80 |
| 7 | VSS | 1946.50 | 1006.80 |
| 8 | PADSEG0 | 1946.50 | 1092.80 |
| 9 | PADSEG1 | 1946.50 | 1182.80 |
| 10 | PADSEG2 | 1946.50 | 1272.80 |
| 11 | PADSEG3 | 1946.50 | 1362.80 |
| 12 | PADSEG4 | 1946.50 | 1452.80 |
| 13 | PADSEG5 | 1946.50 | 1542.80 |
| 14 | PADSEG6 | 1946.50 | 1632.80 |
| 15 | PADSEG7 | 1946.50 | 1722.80 |
| 16 | PADSEG8 | 1946.50 | 1818.00 |
| 17 | PADSEG9 | 1733.00 | 2016.50 |
| 18 | PADSEG10 | 1632.50 | 2016.50 |
| 19 | PADSEG11 | 1540.50 | 2016.50 |
| 20 | PADSEG12 | 1458.50 | 2016.50 |
| 21 | PADSEG13 | 1376.50 | 2016.50 |
| 22 | PADSEG14 | 1294.50 | 2016.50 |
| 23 | PADSEG15 | 1212.50 | 2016.50 |
| 24 | PADSEG16 | 1130.50 | 2016.50 |
| 25 | PADSEG17 | 1048.50 | 2016.50 |
| 26 | PADSEG18 | 966.50 | 2016.50 |
| 27 | PADSEG19 | 884.50 | 2016.50 |
| 28 | PADSEG20 | 802.50 | 2016.50 |
| 29 | PADSEG21 | 720.50 | 2016.50 |
| 30 | PADSEG22 | 638.50 | 2016.50 |
| 31 | PADP47 | 552.50 | 2016.50 |
| 32 | PADP46 | 462.50 | 2016.50 |
| 33 | PADP45 | 362.50 | 2016.50 |
| 34 | PADP44 | 254.00 | 2016.50 |
| 35 | VSS | 129.86 | 2016.50 |
| 36 | PADP43 | 51.50 | 1806.40 |
| 37 | PADP42 | 51.50 | 1710.40 |
| 38 | PADP41 | 51.50 | 1618.40 |
| 39 | PADP40 | 51.50 | 1528.40 |
| 40 | PADP23 | 51.50 | 1438.40 |
| 41 | PADP22 | 51.50 | 1348.40 |
| 42 | PADP21 | 51.50 | 1258.40 |
| 43 | PADP06 | 51.50 | 1168.40 |
| 44 | PADP05 | 51.50 | 1078.40 |
| 45 | PADP04 | 51.50 | 988.40 |
| 46 | PADP03 | 51.50 | 898.40 |
| 47 | PADP02 | 51.50 | 808.40 |
| 48 | PADP01 | 51.50 | 718.40 |
| 49 | PADP00 | 51.50 | 628.40 |
| 50 | PADP26 | 51.50 | 538.40 |
| 51 | PADP25 | 51.50 | 448.40 |
| 52 | PADP24 | 51.50 | 356.40 |
| 53 | PADP30 | 51.50 | 260.40 |
| 54 | PADP31 | 254.00 | 51.50 |
| 55 | PADP32 | 362.50 | 51.50 |
| 56 | PADP33 | 462.50 | 51.50 |
| 57 | PADP37 | 552.50 | 51.50 |
| 58 | PADP36 | 642.50 | 51.50 |
| 59 | VSS | 728.50 | 51.50 |
| 60 | VSS | 810.50 | 51.50 |
| 61 | PADP35 | 892.50 | 51.50 |
| 62 | PADP35 | 974.50 | 51.50 |
| 63 | PADP34 | 1056.50 | 51.50 |
| 64 | PADP10 | 1138.50 | 51.50 |
| 65 | PADP11 | 1220.50 | 51.50 |
| 66 | PADP12 | 1302.50 | 51.50 |
| 67 | PADP13 | 1384.50 | 51.50 |
| 68 | PADP14 | 1466.50 | 51.50 |
| 69 | PADP15 | 1548.50 | 51.50 |
| 70 | PADP16 | 1640.50 | 51.50 |
| 71 | PADP17 | 1741.00 | 51.50 |
| 72 | PADP27 | 1946.50 | 250.00 |
| 73 | PADP28 | 1946.50 | 342.80 |
| 74 | PADP07 | 1946.50 | 424.80 |

***Note***: VDDA and VBAT need bonding together
***Note***: VSSI, VSSA and VSS need bonding together
***Note***: The two P35 PADs are connected inside chip, double bond for high sink application

# PIN DESCRIPTION

| Name | In/Out | Pin Description |
|---|---|---|
| P1.0~P1.7<br>P3.3~P3.7 | I/O | Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output or "open-drain" output. Pull-up resistors are assignable by software. |
| P3.0~P3.2 | I/O | Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output or "pseudo open drain" output. Pull-up resistors are assignable by software. |
| P0.0~P0.7<br>P2.0~P2.7<br>P4.0~P4.7 | I/O | Bit-programmable I/O port for Schmitt-trigger input or CMOS push-pull output. Pull-up resistors are assignable by software. |
| INT0, INT1 | I | External low level or falling edge Interrupt input, Idle/Halt/Stop mode wake up input |
| INT2 | I | External falling edge Interrupt input, Idle/Halt/Stop mode wake up input |
| RXD | I/O | UART Mode0 transmit & receive data, Mode1/2/3 receive data |
| TXD | I/O | UART Mode0 transmit clock, Mode1/2/3 transmit data. In One Wire UART mode, this pin transmits and receives serial data. |
| MISO | I/O | SPI data input for Master mode, data output for Slave mode |
| MOSI | I/O | SPI data output for Master mode, data input for Slave mode |
| SCK | I/O | SPI clock output for Master or clock input for Slave mode |
| MSCL | O | Master I2C SCL |
| MSDA | I/O | Master I2C SDA |
| T0, T1, T2 | I | Timer0, Timer1, Timer2 event count pin input |
| T2EX | I | Timer2 external trigger input |
| TCO | O | System clock divided by 2 output |
| PWM0P, PWM0N | O | Positive and Negative PWM0 output, support pump voltage drive |
| PWM1~PWM3<br>PWM4A, PWM4B<br>PWM5 | O | PWM outputs<br>PWM1 with 300mA sink capability |
| RFC0R~RFC2R | O | RFC resistor connection pin |
| RFCX | I | RFC clock input pin |
| SEG0~SEG2 | O | LED segment output (for DC high/low voltage output) |
| SEG3~SEG43 | O | LCD / LED segment output |
| SEG44~SEG47 | O | LCD segment output |
| COM0~COM7 | O | LCD / LED common output |
| VLX | – | Add 1uF capacitor to VSS for LCD pump; otherwise, connect this pin to VBAT. |
| AD0~AD11 | I | 12-bit ADC channel input |
| TK0~TK9 | I | Touch Key Input |
| RSTn | I | External active low reset input |
| SX1, SX2 | – | 32768 Crystal / Resonator oscillator connection for System Clock (SXT) |
| VDD | – | LDO Regulator output and internal power supply. Add 1uF capacitor to VSS for 5V/3V application. Connect to VBAT for 1.5V application. |
| VBAT, VSS | P | Power input pin and ground, VBAT is the I/O pin power supply |

***Note***: Digital I/O pins voltage swing from $V_{SS}$ to $V_{BAT}$.

***Note***: P1.0~P1.7, P2.4~P2.5, P3.0~P3.1 and P3.4~P3.7 support Pin Change Interrupt & Wake-up

# FUNCTIONAL DESCRIPTION

## 1. CPU Core

In the 8051 architecture, the C programming language is used as a development platform. The TM52 device features a fast 8051 core in a highly integrated microcontroller, allowing designers to be able to achieve improved performance compared to a classic 8051 device. TM52 series microcontrollers provide a complete binary code with standard 8051 instruction set compatibility, ensuring an easy migration path to accelerate the development speed of system products. The CPU core includes an ALU, a program status word (PSW), an accumulator (ACC), a B register, a stack point (SP), DPTRs, a program counter, an instruction decoder, and core special function registers (SFRs).

### 1.1 Accumulator (ACC)

This register provides one of the operands for most ALU operations. Accumulators are generally referred to as A or Acc and sometimes referred to as Register A. In this document, the accumulator is represented as “A” or “ACC,” including the instruction table. The accumulator, as its name suggests, is used as a general register to accumulate the intermediate results of a large number of instructions. The accumulator is the most important and frequently used register to complete arithmetic and logical operations. It holds the intermediate results of most arithmetic and logic operations and assists in data transportation.

| SFR **E0h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **ACC** | ACC.7 | ACC.6 | ACC.5 | ACC.4 | ACC.3 | ACC.2 | ACC.1 | ACC.0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

E0h.7~0 **ACC:** Accumulator

### 1.2 B Register (B)

The “B” register is very similar to the ACC and may hold a 1 Byte value. This register provides the second operand for multiply or divide instructions. Otherwise, it may be used as a scratch pad register. The B register is only used by two 8051 instructions, MUL and DIV. When A is to be multiplied or divided by another number, the other number is stored in B. For MUL and DIV instructions, it is necessary that the two operands be in A and B.

| SFR **F0h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **B** | B.7 | B.6 | B.5 | B.4 | B.3 | B.2 | B.1 | B.0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

F0h.7~0 **B:** B register

### 1.3 Stack Pointer (SP)

The SP register contains the Stack Pointer. The Stack Pointer is used to load the program counter into memory during LCALL and ACALL instructions and is used to retrieve the program counter from memory in RET and RETI instructions. The stack may also be saved or loaded using PUSH and POP instructions, which also increment and decrement the Stack Pointer. The Stack Pointer points to the top location of the stack.

| SFR **81h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SP** | SP | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

81h.7~0 **SP:** Stack Point

### 1.4 Dual Data Pointer (DPTRs)

TM52 device has two DPTRs, which share the same SFR address. Each DPTR is 16 bits in size and consists of two registers: the DPTR high byte (DPH) and the DPTR low byte (DPL). The DPTR is used for 16-bit-address external memory accesses, for offset code byte fetches, and for offset program jumps. Setting the DPSEL control bit allows the program code to switch between the two physical DPTRs.

| SFR **82h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **DPL** | DPL | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

82h.7~0 **DPL:** Data Point low byte

| SFR **83h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **DPH** | DPH | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

83h.7~0 **DPH:** Data Point high byte

| SFR **F8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX1** | – | – | TKSOC | ADSOC | CLRWDT | CLRTM3 | STPRFC | DPSEL |
| R/W | – | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | – | – | 0 | 0 | 0 | 0 | 0 | 0 |

F8h.0 **DPSEL:** Active DPTR Select

### 1.5 Program Status Word (PSW)

This register contains status information resulting from CPU and ALU operations. The instructions that affect the PSW are listed below.

| **Instruction** | **Flag** | | |
|---|---|---|---|
| | **C** | **OV** | **AC** |
| ADD | X | X | X |
| ADDC | X | X | X |
| SUBB | X | X | X |
| MUL | 0 | X | |
| DIV | 0 | X | |
| DA | X | | |
| RRC | X | | |
| RLC | X | | |
| SETB C | 1 | | |

| **Instruction** | **Flag** | | |
|---|---|---|---|
| | **C** | **OV** | **AC** |
| CLR C | 0 | | |
| CPL C | X | | |
| ANL C, bit | X | | |
| ANL C, /bit | X | | |
| ORL C, bit | X | | |
| ORL C, /bit | X | | |
| MOV C, bit | X | | |
| CJNE | X | | |
| | | | |

A “0” means the flag is always cleared, a “1” means the flag is always set and an “X” means that the state of the flag depends on the result of the operation.

| SFR **D0h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PSW** | CY | AC | F0 | RS1 | RS0 | OV | F1 | P |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

D0h.7 **CY:** ALU carry flag

D0h.6 **AC:** ALU auxiliary carry flag

D0h.5 **F0:** General purpose user-definable flag

D0h.4~3 **RS1, RS0:** The contents of (RS1, RS0) enable the working register banks as:

00: Bank 0 (00h~07h)

01: Bank 1 (08h~0Fh)

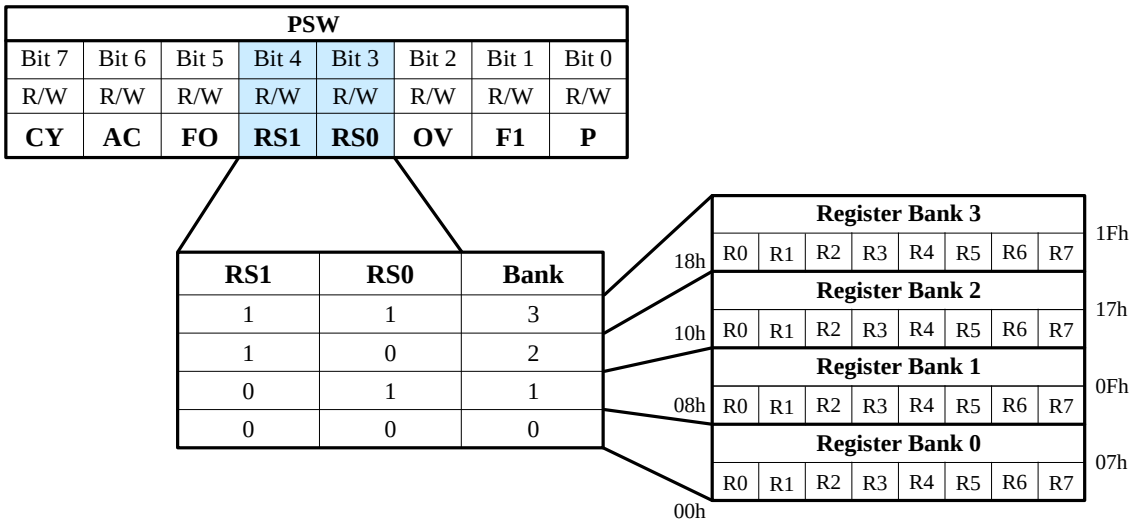
10: Bank 2 (10h~17h)

11: Bank 3 (18h~1Fh)

D0h.2 **OV:** ALU overflow flag

D0h.1 **F1:** General purpose user-definable flag

D0h.0 **P:** Parity flag. Set/cleared by hardware each instruction cycle to indicate odd/even number of "one" bits in the accumulator.



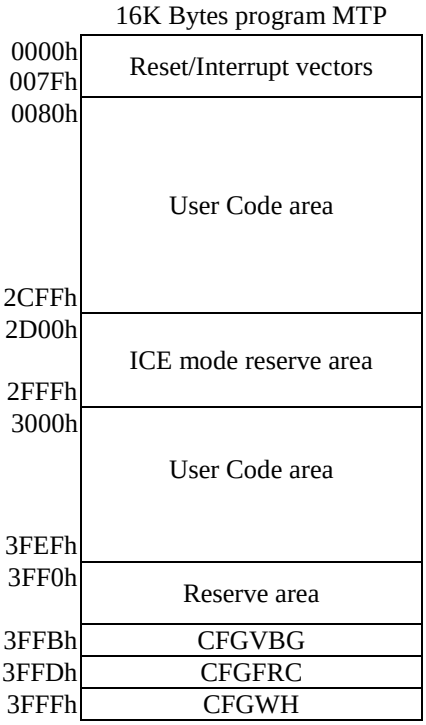
# 2. Memory

## 2.1 Program Memory

The chip has a 16K Bytes MTP program memory, which can support In Circuit Programming (ICP), In Application Programming (IAP) and In System Programming (ISP) function modes. The MTP write endurance is at least 10000 cycles. The program memory address continuous space (0000h~3FFFh) is partitioned to several sectors for device operation.

### 2.1.1 Program Memory Functional Partition

The last 16 bytes (3FF0h~3FFFh) of program memory is defined as reserve area or chip Configuration Word (CFGWs). Three of them are loaded into the device control registers upon power on reset (POR). The 0000h~007Fh is occupied by Reset/Interrupt vectors as standard 8051 definition. In the in-circuit emulation (ICE) mode, address space 2D00h~2FFFh is reserved for ICE System communication.




### 2.1.2 MTP ICP Mode

The MTP memory can be programmed by the tenx proprietary writer (TWR99/TWR100), which needs at least four wires (VBAT, VSS, P1.2, and P1.3 pins) to connect to this chip. If the user wants to program the MTP memory on the target circuit board (In Circuit Program, ICP), these pins must be reserved sufficient freedom to be connected to the Writer.

### 2.1.3 MTP IAP Mode

The chip has “In Application Program” (IAP) capability, which allows software to read/write data from/to the MTP memory during CPU run time as conveniently as data EEPROM access. The IAP function is byte writable, meaning that the chip does not need to erase one MTP page before write. The full MTP space is available for IAP access, except the CFGWH (address 3fffh).

### 2.1.4 IAP Mode Access Routines

**MTP IAP Write** is simply achieved by a "MOVX @DPTR, A" instruction while the DPTR contains the target MTP address (0~3FFEh), and the ACC contains the data being written. The chip accepts MTP write command only when MTPWE=1. MTP IAP writing requires approximately 1ms. Meanwhile, the CPU stays in a waiting state, but all peripheral modules (Timers, LCD, and others) continue running during the writing time. The software must handle the pending interrupts after an IAP write. The chip has a build-in IAP Time-out function for escaping write fail state.

MTP IAP writing needs higher $V_{DD}$ voltage and lower $F_{SYSCLK}$, typically $V_{DD}$>4.0V and $F_{SYSCLK}$ < 8MHz. Besides, S/W must disable WDT and enable LVR before IAP Write. Be careful to avoid the IAP write during $V_{DD}$ drops. It is recommended to insert at least 20uS delay between each write for the consecutive writing.

Because the Program memory and the IAP data space share the same entity, a **MTP IAP Read** can be performed by the "MOVX A, @DPTR" or "MOVC" instruction as well. A MTP IAP read does not require extra CPU wait time.

```
; IAP example code,  need VDD > 4.0V & WDT disable
MOV     DPTR, #3000h        ; DPTR = 3000h = target IAP address
MOV     A, #5Ah             ; A = 5Ah = target IAP write data
MOV     AUX2, #02h          ; IAP Time-Out function enable
MOV     97h, #65h           ; MTPALL=1, enable IAP
MOV     0C9h, #47h          ; MTPWE=1
MOVX    @DPTR, A            ; MTP[3000h] = 5Ah, after IAP write
                            ; 1ms H/W writing time, CPU wait
MOV     0C9h, #00h          ; MTPWE=0 immediately after IAP write
CLR     A                   ; A = 0
MOVX    A, @DPTR            ; A = 5Ah
CLR     A                   ; A = 0
MOVC    A, @A+DPTR          ; A = 5Ah
```

## 2.2 EEPROM

The chip contains 128 bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 50K write cycles.

EEPROM Memory

| | |
|---|---|
| EE00h | EEPROM[0] |
| EE02h | EEPROM[1] |
| EE04h | . . . |
| EEFCh | EEPROM[126] |
| EEFEh | EEPROM[127] |

(Only even addresses can be used, odd addresses are invalid)

**The EEPROM Write** is similar to the MTP IAP mode. It is simply achieved by a "MOVX @DPTR, A" instruction while the DPTR contains the target EEPROM address (EE00h~EEFEh, ADDR=ADDR+2), and the ACC contains the data being written. EEPROM writing requires approximately 2 ms @$V_{BAT}$=3V, 1 ms @$V_{BAT}$=5V. Meanwhile, the CPU stays in a waiting state, but all peripheral modules (Timers, LED, and others) continue running during the writing time. The software must handle the pending interrupts after an EEPROM write. The chip has a build-in EEPROM Time-out function shared with MTP IAP for escaping write fail state.

EEPROM writing needs higher $V_{DD}$ voltage, typically $V_{DD}$ > 3.0V. Besides, S/W must disable WDT and enable LVR before EEPROM Write. Be careful to avoid the EEPROM write during $V_{DD}$ drops. It is recommended to insert at least 20uS delay between each write for the consecutive writing.

**The EEPROM Read** can be performed by the "MOVX A, @DPTR" instruction as long as the target address points to the EE00h~EEFEh area. The EEPROM read require approximately 300ns.

```
; EEPROM example code, need VDD > 3.0V & WDT disable
ANL     WDTCON, #0F3h       ; Disable WDT
MOV     DPTR, #0EE00h       ; DPTR=EE00h=target EEPROM[0] address
MOV     A, #0A5h            ; A=A5h=target EEPROM[0] write data
MOV     0C9h, #0E2h         ; EEPROM write enable
MOV     AUX2, #02h          ; EEPROM Time-Out function enable
MOVX    @DPTR, A            ; EEPROM[0]=0A5h, after EEPROM write
                            ; 1ms H/W writing time, CPU wait
MOV     0C9h, #00h          ; EEPROM write disable, immediately after EEPROM write
CLR     A                   ; A=0
MOVX    A, @DPTR            ; A=0A5h
```

| SFR **97h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SWCMD** | MTPALL / SWRST | | | | | | | |
| R/W | W | | | | | | | R/W |
| Reset | – | | | | | | | 0 |

97h.7~0 **MTPALL (W):** Write 65h to set MTPALL flag and enable MTP IAP; Write other value to clear MTPALL flag and disable IAP. It is recommended to clear it immediately after IAP access.

97h.0 **MTPALL (R):** Flag indicates MTP memory can be accessed by IAP or not.

| SFR **C9h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **IAPWE** | MTPWE / EEPWE | | | | | | | |
| | MTPWE | IAPTO | EEPWE | | | | | |
| R/W | R/W | R/W | R/W | W | | | | |
| Reset | 0 | 0 | 0 | – | | | | |

C9h.7~0 **IAPWE (W):** Write 47h to set MTPWE control flag; Write E2h to set EEPWE control flag; Write other value to clear MTPWE and EEPWE flag. It is recommended to clear it immediately after MTP or EEPROM write.

C9h.7 **MTPWE (R):** Flag indicates MTP memory can be written by IAP or not, 1 = MTP write enable.

C9h.6 **IAPTO (R):** MTP (or EEPROM) write Time-Out flag, Set by H/W when MTP (or EEPROM) write Time-out occurs. Cleared by H/W when MTPWE=0 (or EEPWE=0).

C9h.5 **EEPWE (R):** Flag indicates EEPROM memory can be written or not, 1 = EEPROM write enable.

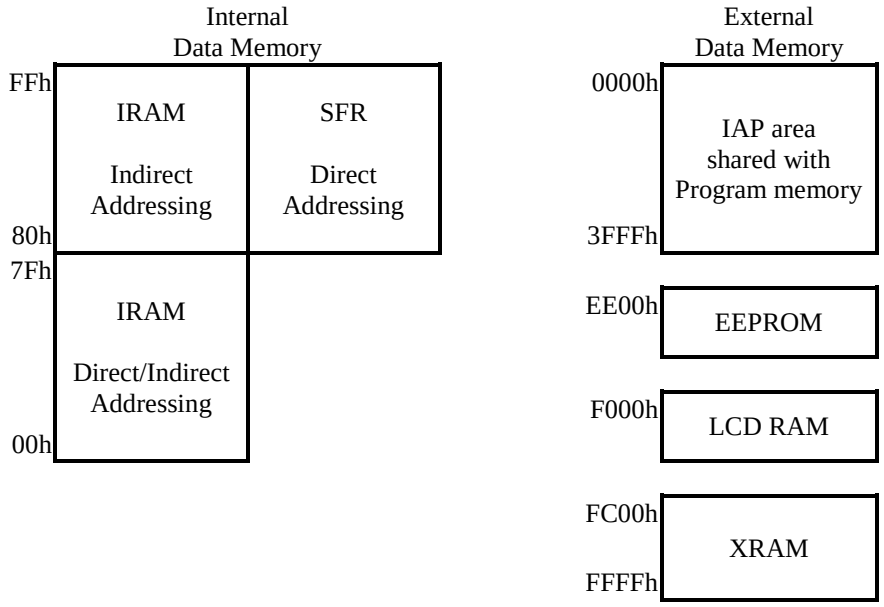
| SFR **D3h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX2** | P07ADC | P20ADC | P02TCO | LBDEDGE | VBGE | VBGOUT | IAPTE | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

D3h.1~0 **IAPTE:** MTP (or EEPROM) write time-out enable.

00: Disable
01: wait 1ms to trigger time-out flag, and escape the write fail state
10: wait 4ms to trigger time-out flag, and escape the write fail state
11: wait 8ms to trigger time-out flag, and escape the write fail state

## 2.3 Data Memory

As the standard 8051, the chip has both Internal and External Data Memory space. The Internal Data Memory space consists of 256 Bytes IRAM and SFRs, which are accessible through a rich instruction set. The External Data Memory space consists of 1024 Bytes XRAM, LCDRAM and IAP area, which can be only accessed by MOVX instruction.




### 2.3.1 IRAM

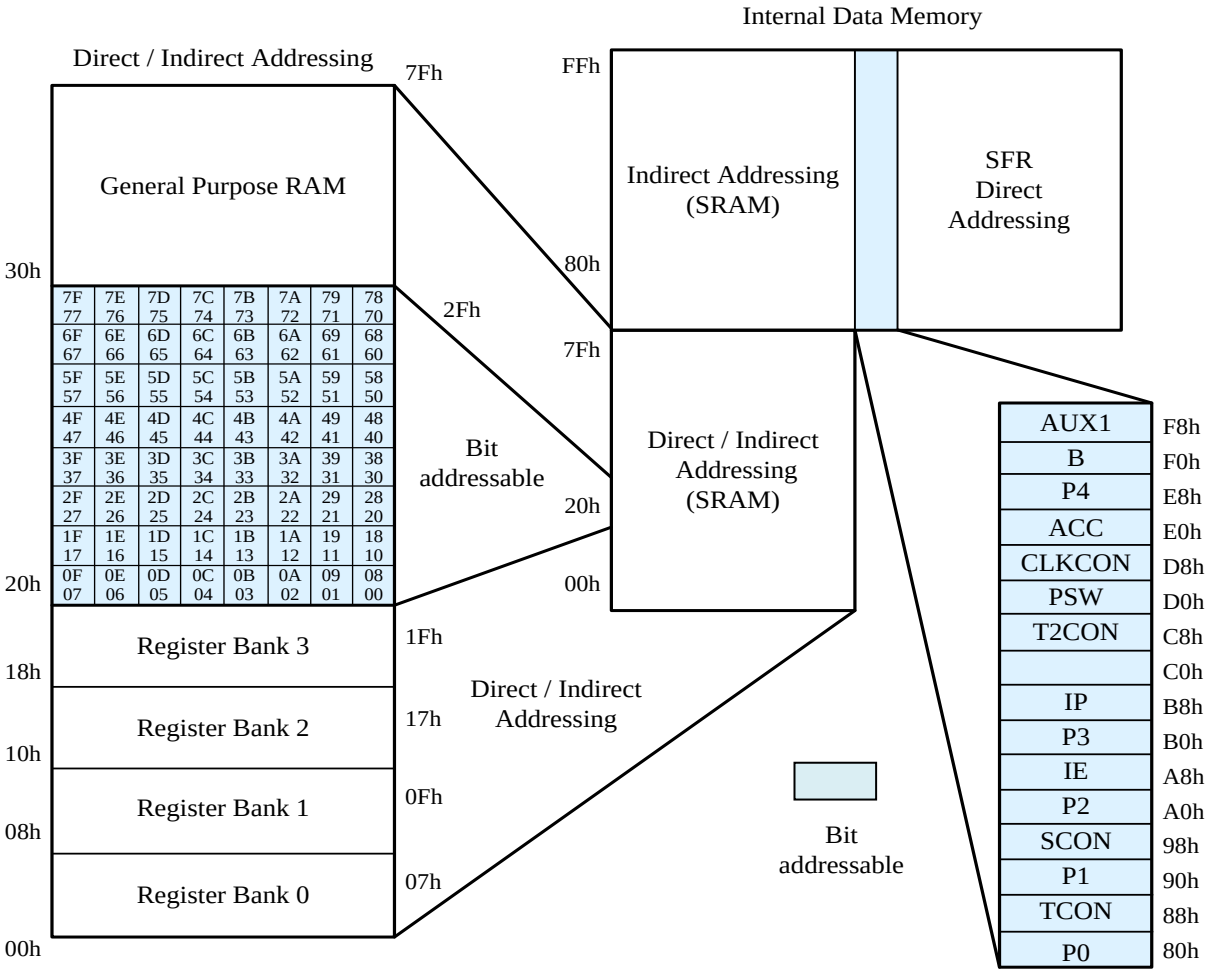
IRAM is located in the 8051 internal data memory space. The whole 256 Bytes IRAM are accessible using indirect addressing but only the lower 128 Bytes are accessible using direct addressing. There are four directly addressable register banks (switching by PSW), which occupy IRAM space from 00h to 1Fh. The address 20h to 2Fh 16 Bytes IRAM space is bit-addressable. IRAM can be used as scratch pad registers or program stack.

### 2.3.2 XRAM

XRAM is located in the 8051 external data memory space (address from FC00h to FFFFh). The 1024 Bytes XRAM can be only accessed by “MOVX” instruction.

### 2.3.3 SFRs

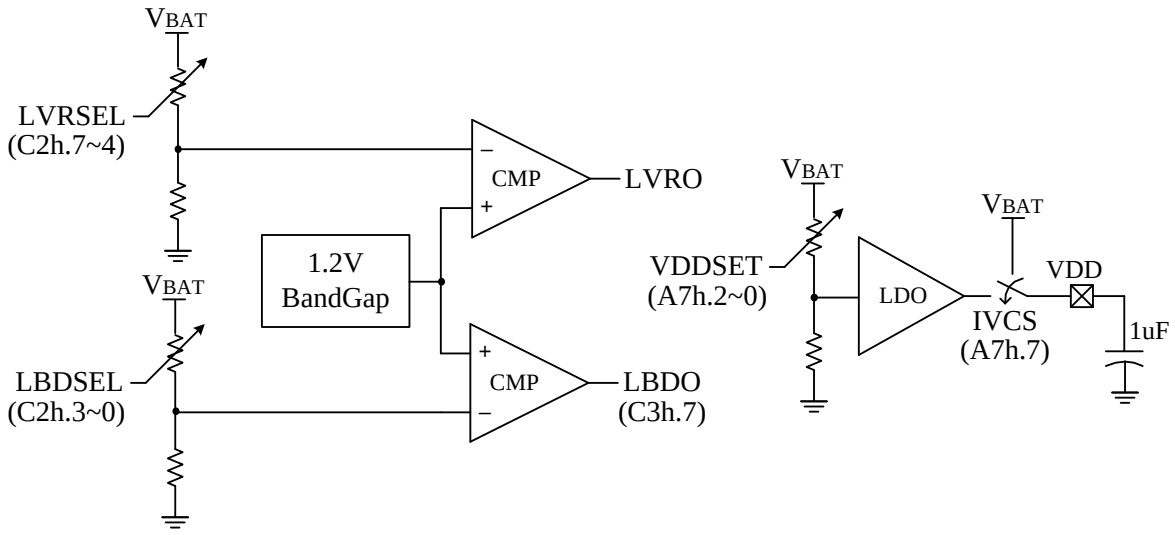
All peripheral functional modules such as I/O ports, Timers and UART operations for the device are accessed via Special Function Registers (SFRs). These registers occupy upper 128 Bytes of direct Data Memory space locations in the range 80h to FFh. There are 15 bit-addressable SFRs (which means that eight individual bits inside a single byte are addressable), such as ACC, B register, PSW, TCON, SCON, and others. The remaining SFRs are only byte addressable. SFRs provide control and data exchange with the resources and peripherals of the device. The TM52 series of microcontrollers provides complete binary code with standard 8051 instruction set compatibility. Beside the standard 8051 SFRs, the chip implements additional SFRs used to configure and access subsystems such as the SPI/LCD, which are unique to the chip.




| | 8/0 | 9/1 | A/2 | B/3 | C/4 | D/5 | E/6 | F/7 |
|---|---|---|---|---|---|---|---|---|
| F8h | AUX1 | | | | | | | |
| F0h | B | CRCDL | CRCDH | CRCIN | | CFGVBG | | CFGFRC |
| E8h | P4 | | | | | | | |
| E0h | ACC | MICON | MIDAT | | | | | |
| D8h | CLKCON | | | | | | | |
| D0h | PSW | | WDTCON | AUX2 | PWM4DTY | PWM5DTY | PWM5PRD | PWMOE |
| C8h | T2CON | IAPWE | RCP2L | RCP2H | TL2 | TH2 | PWMCON | PWMCON2 |
| C0h | | | LVSET | ADDTL | ADCON | ADDTH | XBAUD | EFTCON |
| B8h | IP | IPH | IP1 | IP1H | SPCON | SPSTA | SPDAT | |
| B0h | P3 | LCON | LCON2 | TM3SEC | TM3DL | TM3DH | TM3RLD | TM3ADJ |
| A8h | IE | INTE1 | | TKDH | TKDL | TKCON | TKCON2 | RFCON |
| A0h | P2 | P3WKUP | P1MODL | P1MODH | P3MODL | P3MODH | P4OE | VCON |
| 98h | SCON | SBUF | PWM0PRD | PWM0DTY | PWM1PRD | PWM1DTY | PWM2DTY | PWM3DTY |
| 90h | P1 | P0OE | PINMODE | P2OE | OPTION | INTFLG | P1WKUP | SWCMD |
| 88h | TCON | TMOD | TL0 | TL1 | TH0 | TH1 | | |
| 80h | P0 | SP | DPL | DPH | | | | PCON |

## 3. Power Management

VBAT pin is the power supply of this chip. It provides voltage source to the built-in tiny current LDO Regulator for device internal operation. The VDD pin is the LDO output, which needs an external 1uF capacitor connection to VSS for voltage level stability. If IVCS=0, LDO is disable and VDD is shorted to VBAT. If IVCS=1, LDO is enable and the $V_{DD}$ voltage level is defined by VDDSET SFR. The $V_{DD}$ range can be set as $V_{BAT}$*0.375~$V_{BAT}$*0.725. The lower $V_{DD}$ voltage level causes lower chip current consumption, but user must also consider the System clock rate. Higher clock rate requires higher $V_{DD}$ voltage level. User must keep $V_{DD}$>1.4V(25℃) for the chip's proper operation. In EEPROM write mode, user also needs to set $V_{DD}$>3V(25℃).




LDO Regulator & LVR/LBD

The 1.2V BandGap Voltage Reference module supports for Low Battery Detection (LBD) and LVR. User can refer to the $V_{BAT}$ voltage level for setting the $V_{DD}$ level by VDDSET SFR. The BandGap and LBD consume un-neglect current, so user should not use them too often. Since $V_{BAT}$ voltage level changes very slowly, user can detect it once an hour or once a day to reduce current consumption.

| SFR **A7h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **VCON** | IVCS | PWRSAV | PORPD | LBDPD | LVRPD | VDDSET | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

A7h.7 **IVCS:** Chip internal LDO Regulator enable control
0: LDO disable, $V_{DD}$ = $V_{BAT}$
1: LDO enable, $V_{DD}$ = LDO Regulator output

A7h.6 **PWRSAV:** Power saving mode control
0: No power saving
1: Power saving, disable POR in Halt mode, disable LVR/LBD in Idle/Halt/Stop mode, POR enable time is 1/16 duty.

A7h.5 **PORPD:** POR control, 1=force POR disable

A7h.4 **LBDPD:** LBD control, 1=force LBD disable

A7h.3 **LVRPD:** LVR control, 1=force LVR disable

A7h.2~0 **VDDSET:** $V_{DD}$ voltage setting while IVCS=1.

000: $V_{DD}$ =$V_{BAT}$*0.375; 001: $V_{DD}$ =$V_{BAT}$*0.425;
010: $V_{DD}$ =$V_{BAT}$*0.475; 011: $V_{DD}$ =$V_{BAT}$*0.525;
100: $V_{DD}$ =$V_{BAT}$*0.575; 101: $V_{DD}$ =$V_{BAT}$*0.625;
110: $V_{DD}$ =$V_{BAT}$*0.675; 111: $V_{DD}$ =$V_{BAT}$*0.725;

| Mode | PWRSAV | POR (PORPD=0) | LVR/LBD (LVRPD=LBDPD=0) |
|---|---|---|---|
| STOP | 0 | Off | On |
| HALT | | On, Full Duty | |
| IDLE | | | |
| FAST / SLOW | | | |
| STOP | 1 | Off | Off |
| HALT | | Off | |
| IDLE | | AGMOD=0: On, 1/16 Duty | |
| FAST / SLOW | | AGMOD=1: On, Full Duty | On |

| SFR**CFh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWMCON2** | PWRSAV2 | PWM5CLR | PWM0VX2 | PWM1SNK | PWM5CKS | PWM5PSC | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

CFh.7 **PWRSAV2:** Power saving mode control
0: No power saving
1: Reduce Slow mode current consumption

| SFR **C2h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **LVSET** | LVRSEL | | | | LBDSEL | | | |
| R/W | R/W | | | | R/W | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

C2h.7~4 **LVRSEL:** Low Voltage Reset select

| | |
|---|---|
| 0000: LVR=1.73V | 0001: LVR=1.85V |
| 0010: LVR=1.98V | 0011: LVR=2.10V |
| 0100: LVR=2.22V | 0101: LVR=2.34V |
| 0110: LVR=2.46V | 0111: LVR=2.59V |
| 1000: LVR=2.71V | 1001: LVR=2.83V |
| 1010: LVR=2.96V | 1011: LVR=3.09V |
| 1100: LVR=3.21V | 1101: LVR=3.33V |
| 1110: LVR=3.46V | 1111: LVR=3.58V |

C2h.3~0 **LBDSEL:** Low Battery Detector select

| | |
|---|---|
| 0000: LBD=1.73V | 0001: LBD=1.85V |
| 0010: LBD=1.98V | 0011: LBD=2.10V |
| 0100: LBD=2.22V | 0101: LBD=2.34V |
| 0110: LBD=2.46V | 0111: LBD=2.59V |
| 1000: LBD=2.71V | 1001: LBD=2.83V |
| 1010: LBD=2.96V | 1011: LBD=3.09V |
| 1100: LBD=3.21V | 1101: LBD=3.33V |
| 1110: LBD=3.46V | 1111: LBD=3.58V |

| SFR **C3h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **ADDTL** | LBDO | – | – | ADEOC | ADCDTL | | | |
| R/W | R | – | – | R | R | | | |
| Reset | – | – | – | – | – | – | – | – |

C3h.7 **LBDO:** Low Battery Detector flag
If $V_{BAT}$ < LBDSEL's setting voltage, LBDO=1; otherwise LBDO=0.

| SFR **95h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTFLG** | LBDIF | – | TKIF | ADIF | PWMIF | IE2 | PNCIF | TF3 |
| R/W | R/W | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |

95h.7 **LBDIF:** LBD Interrupt Flag

Set by H/W at LBDO's rising or falling edge. Cleared by H/W when CPU vectors into the interrupt service routine. S/W writes 7Fh to INTFLG to clear this flag.

| SFR **D3h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX2** | P07ADC | P20ADC | P02TCO | LBDEDGE | VBGE | VBGOUT | IAPTE | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

D3h.4 **LBDEDGE:** LBDIF trigger condition

0: LBDIF trigger by LBDO's rising edge. (when $V_{BAT}$ falling)

1: LBDIF trigger by LBDO's falling edge. (when $V_{BAT}$ rising)

| SFR **F5h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CFGVBG** | – | – | – | VBGTRIM | | | | |
| R/W | – | – | – | R/W | | | | |
| Reset | – | – | – | – | – | – | – | – |

F5h.4~0 **VBGTRIM:** VBG adjustment. It is automatically loaded with MTP's 3FFBh data at power on reset and can be read/written as any other SFR register in normal mode. (00h=lowest)

# 4. Reset

The chip has five types of reset methods. The CFGW and SFRs control the Reset functionality.

## 4.1 Power on Reset (POR)

After Power on Reset, the chip stays on Reset state for 20ms as warm up time, then downloads the CFGWs register from MTP's last three words (other Reset dose not reload the CFGWs). The Power on Reset needs $V_{BAT}$ voltage first discharge to near $V_{SS}$ level, then rise beyond 1.0V or 1.7V, which is determined by the CFGWH. POR is disabled in Stop mode and enabled in others mode by VCON SFR control.

## 4.2 External Pin Reset

External Pin Reset is active low. The RSTn pin needs to keep at least 2 SRC clock cycle long to be sampled by the chip. Pin Reset can be disabled or enabled by CFGWH.

## 4.3 Software Reset

Software Reset is activated by writing the SFR 97h with data 56h.

## 4.4 Watch Dog Timer Reset

WDT overflow Reset is disabled or enable by WDTCON SFR. The WDT uses slow clock as its counting time base. WDT overflow speed is defined by WDTPSC SFR. WDT is cleared by the chip Reset or CLRWDT SFR bit.

## 4.5 Low Voltage Reset (LVR)

LVR is disabled or enable by VCON SFR. There are 16-level LVR can be selected by LVRSEL.

| MTP **3FFFh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CFGWH** | PROT | XRSTE | – | – | – | AGMOD | IAPHVS | – |

3FFFh.6 **XRSTE:** Pin Reset enable, 1=enable.

3FFFh.2 **AGMOD:** Power on reset level select.
0: POR is 1.7V
1: POR is 1.1V

| SFR **97h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SWCMD** | MTPALL / SWRST | | | | | | | |
| R/W | W | | | | | | | R/W |
| Reset | – | | | | | | | 0 |

97h.7~0 **SWRST (W):** Write 56h to generate S/W Reset.

| SFR **F8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX1** | – | – | TKSOC | ADSOC | CLRWDT | CLRTM3 | STPRFC | DPSEL |
| R/W | – | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | – | – | 0 | 0 | 0 | 0 | 0 | 0 |

F8h.3 **CLRWDT:** Set to 1 to clear Watch Dog Timer.

| SFR **D2h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **WDTCON** | IAPHTW | TM3PSC | | | WDTMOD | | WDTPSC | |
| R/W | R/W | R/W | | | R/W | | R/W | |
| Reset | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

D2h.3~2 **WDTMOD:** WDT control
- 00: WDT disable
- 01: WDT disable in Halt / Stop mode, enable in Idle / Slow / Fast mode
- 10: WDT disable in Idle / Halt / Stop mode, enable in Slow / Fast mode
- 11: WDT disable in Stop mode, enable in Halt / Idle / Slow / Fast mode

D2h.1~0 **WDTPSC:** WDT pre-scalar time select
- 00: WDT overflow is 2048 Slow clock cycle (64ms @SXT=32K)
- 01: WDT overflow is 4096 Slow clock cycle (128ms @SXT=32K)
- 10: WDT overflow is 8192 Slow clock cycle (256ms @SXT=32K)
- 11: WDT overflow is 16384 Slow clock cycle (512ms @SXT=32K)

| SFR **A7h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **VCON** | IVCS | PWRSAV | PORPD | LBDPD | LVRPD | VDDSET | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

A7h.6 **PWRSAV:** Power saving mode control
- 0: No power saving
- 1: Power saving, disable POR in Halt mode, disable LVR/LBD in Idle/Halt/Stop mode, POR enable time is 1/16 duty.

A7h.5 **PORPD:** POR control, 1=force POR disable

A7h.4 **LBDPD:** LBD control, 1=force LBD disable

A7h.3 **LVRPD:** LVR control, 1=force LVR disable

| SFR **C2h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **LVSET** | LVRSEL | | | | LBDSEL | | | |
| R/W | R/W | | | | R/W | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

C2h.7~4 **LVRSEL:** Low Voltage Reset select

| | |
|---|---|
| 0000: LVR=1.73V | 0001: LVR=1.85V |
| 0010: LVR=1.98V | 0011: LVR=2.10V |
| 0100: LVR=2.22V | 0101: LVR=2.34V |
| 0110: LVR=2.46V | 0111: LVR=2.59V |
| 1000: LVR=2.71V | 1001: LVR=2.83V |
| 1010: LVR=2.96V | 1011: LVR=3.09V |
| 1100: LVR=3.21V | 1101: LVR=3.33V |
| 1110: LVR=3.46V | 1111: LVR=3.58V |

C2h.3~0 **LBDSEL:** Low Battery Detector select

| | |
|---|---|
| 0000: LBD=1.73V | 0001: LBD=1.85V |
| 0010: LBD=1.98V | 0011: LBD=2.10V |
| 0100: LBD=2.22V | 0101: LBD=2.34V |
| 0110: LBD=2.46V | 0111: LBD=2.59V |
| 1000: LBD=2.71V | 1001: LBD=2.83V |
| 1010: LBD=2.96V | 1011: LBD=3.09V |
| 1100: LBD=3.21V | 1101: LBD=3.33V |
| 1110: LBD=3.46V | 1111: LBD=3.58V |

# 5. Clock Circuitry & Operation Mode

## 5.1 System Clock

The chip is designed with dual-clock system. During runtime, user can directly switch the System clock from fast to slow or from slow to fast. It also can directly select a clock divider of 1, 2, 4 or 16. The Fast clock consists of FRC, MRC and RFC. The Slow clock can be selected as SXT or SRC. Fast mode and Slow mode are defined as the CPU running at Fast and Slow clock speeds. The five System Clock sources are list below.

**FRC** (Internal Fast RC, 14.7456 MHz @$V_{BAT}$ = 2.5V~5.5V): FRC is the default Fast clock type. Its frequency is controlled by FRCF SFR, which is automatically loaded with CFGW data at power on reset. The FRC is trimmed to 14.7456 MHz in chip manufacturing. FRC can maintain stable frequency when Temperature and $V_{BAT}$ voltage change, but it needs higher $V_{DD}$ voltage and consumes higher current.

**MRC** (Internal Medium RC, 6MHz @$V_{DD}$ = 3V, 2.3MHz @$V_{DD}$ = 1.5V): MRC frequency depends on $V_{DD}$ voltage and differs chip by chip. The advantage of MRC is being able to work in lower $V_{DD}$ voltage and consume lower current.

**RFC** (Resistance to Frequency Convert, External RC): RFC is usually used for RFC ADC measuring mode. Its frequency depends on External RC and $V_{BAT}$.

**SRC** (Internal Slow RC, 75KHz @$V_{DD}$ = 3V, 35KHz @$V_{DD}$ = 1.5V): After Reset, the chip is running at Slow mode with SRC clock. SRC can work in very low $V_{DD}$ voltage and consumes very low current.

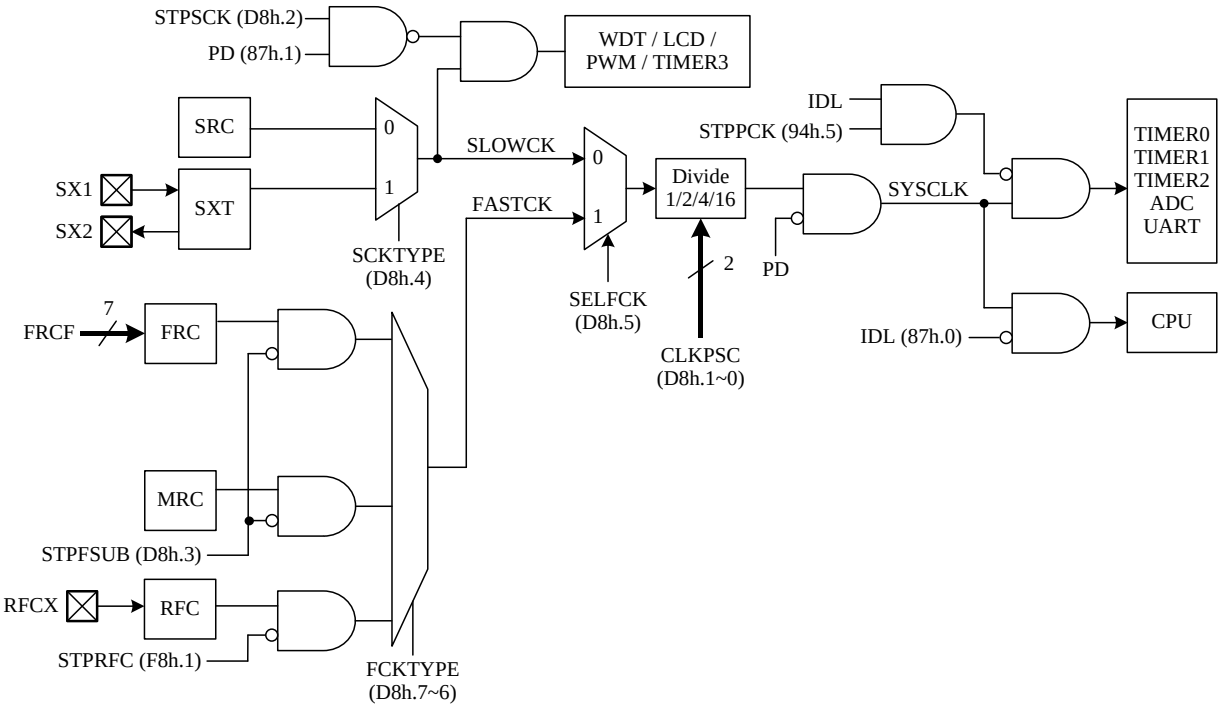
**SXT** (Slow Crystal, 32768Hz): SXT provides accurate real time base. It can work in very low $V_{DD}$ voltage and consumes very low current.

The SXTKICK control bit can accelerate the Crystal start-up oscillating while $V_{BAT}$ < 1.5V. To use this function, F/W needs to setup the LCD pump environment, which includes LCDPUMP=1, LCDCLK=10 (FASTCLK/128) and DSPON=1. After the Crystal oscillating becoming smooth, F/W must clear SXTKICK to reduce current consumption.

Before entering the Fast mode, S/W must select the Fast clock type in advance. If RFC is used as the Fast clock source, S/W also has to setup the pin mode and RFC related SFRs in advance.

Since Fast clock is useless in Slow mode, S/W can set STPFSUB=1 or STPRFC=1 to stop Fast Clock to reduce chip's current consumption. Before the chip switches to FRC, S/W must also consider the $V_{DD}$ voltage level for chip operation safe range. The higher $V_{DD}$ allows the chip to run at higher System Clock frequency. In typical condition, 16 MHz System Clock rate requires $V_{DD}$>2.5V.

The CLKCON SFR controls the System clock operating. H/W automatically blocks the S/W abnormally setting for this register. S/W can only change the Slow clock type in Fast mode, and change the Fast clock type in Slow mode. Never to write both STPFSUB=1 & SELFCK=1 in FRC/MRC mode. It is recommended to write this register bit by bit.




Clock Structure

| MTP **3FFDh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CFGFRC** | – | FRCF | | | | | | |

3FFDh.6~0 **FRCF:** FRC frequency adjustment.
FRC is trimmed to 14.7456 MHz in chip manufacturing. FRCF records the trimming data.

| SFR **F7h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CFGFRC** | – | FRCF | | | | | | |
| R/W | – | R/W | | | | | | |
| Reset | – | – | – | – | – | – | – | – |

F7h.6~0 **FRCF:** FRC frequency adjustment. It is automatically loaded with MTP's 3FFDh data at power on reset and can be read/written as any other SFR register in normal mode. So the FRC clock speed can be changed on CPU run time by S/W. (00h=lowest)

| SFR **94h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **OPTION** | SXTGAIN | | STPPCK | SXTKICK | UART1W | UARTP1 | T2SEL | T1SEL |
| R/W | R/W | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

94h.7~6 **SXTGAIN:** 32768 SXT oscillator gain, 3=Highest gain, 0=Lowest gain. Higher gain can shorten the Crystal oscillation warm-up time. Lower gain can reduce oscillation current.

94h.5 **STPPCK:** Set 1 to stop UART/Timer0/Timer1/Timer2 clock in Idle mode for current reducing.

94h.4 **SXTKICK:** Set 1 to kick SXT by LCD pump voltage, for crystal start up @$V_{BAT}$ < 1.5V

| SFR **F8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX1** | – | – | TKSOC | ADSOC | CLRWDT | CLRTM3 | STPRFC | DPSEL |
| R/W | – | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | – | – | 0 | 0 | 0 | 0 | 0 | 0 |

F8h.1 **STPRFC:** Set 1 to stop RFC clock oscillating

| SFR **D8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CLKCON** | FCKTYPE | | SELFCK | SCKTYPE | STPFSUB | STPSCK | CLKPSC | |
| R/W | R/W | | R/W | R/W | R/W | R/W | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

D8h.7~6 **FCKTYPE:** Fast clock type select, These bits can be changed only in Slow mode (SELFCK=0)
00: Fast clock is FRC
10: Fast clock is MRC
11: Fast clock is RFC, S/W must setup RFC oscillating circuitry before this setting.

D8h.5 **SELFCK:** System clock select. This bit can be changed only when STPFSUB=0 or FCKTYPE=3.
0: Slow clock (SRC / SXT)
1: Fast clock (FRC / MRC / RFC)

D8h.4 **SCKTYPE:** Slow clock Type. This bit can be changed only in Fast mode (SELFCK=1).
0: SRC
1: SXT, P0.7 and P2.0 are crystal oscillator pins

D8h.3 **STPFSUB:** FRC / MRC clock stop control. This bit can be changed only when SELFCK=0 or FCKTYPE=3.
0: FRC / MRC clock running
1: Stop FRC / MRC clock for power saving in Slow / Idle mode.

D8h.2 **STPSCK:** Set 1 to stop Slow clock after PD=1 (Halt / Stop mode entry control)

D8h.1~0 **CLKPSC:** System clock prescaler. Effective after 16 clock cycles (Max.) delay.
00: System clock is Fast/Slow clock divided by 16
01: System clock is Fast/Slow clock divided by 4
10: System clock is Fast/Slow clock divided by 2
11: System clock is Fast/Slow clock divided by 1

***Note:*** In crystal mode, user should set the P0.7/P2.0 (SXT) pins as Input with Pull-up (section7).

***Note:*** In the Power on stage, FW must wait until $V_{DD}$ > 2.2V, before switch to FRC/1.

| SYSCLK | CLKCON (D8h) | | | |
|---|---|---|---|---|
| | bit7~6 FCKTYPE | bit5 SELFCK | bit4 SCKTYPE | bit3 STPFSUB |
| Fast RFC (*1) | 11 | 1 | X | X |
| Fast MRC | 10 | 1 | X | 0 |
| Fast FRC | 00 | 1 | X | 0 |
| Slow SXT | XX | 0 | 1 | X |
| Slow SRC | XX | 0 | 0 | X |
| Fast type change | AB ← → CD | 0 | X | X |
| Slow type change | 00, 01, 10 | 1 | 0 ← → 1 | 0 |
| Slow type change | 11 (RFC mode) | 1 | 0 ← → 1 | X |
| Stop FRC/MRC | 00, 01, 10 | 0 | X | 0 → 1 |
| Stop FRC/MRC | 11 (RFC mode) | X | X | 0 → 1 |
| Start FRC/MRC | 00, 01, 10 | 0 | X | 1 → 0 |
| Start FRC/MRC | 11 (RFC mode) | X | X | 1 → 0 |
| Switch to FRC/MRC | 00, 01, 10 | 0 → 1 | X | 0 |
| Switch to RFC (*1) | 11 (RFC mode) | 0 → 1 | X | X |
| Switch to SRC/SXT | 00, 01, 10 | 1 → 0 | X | 0 |
| Switch to SRC/SXT | 11 (RFC mode) | 1 → 0 | X | X |

(*1) also need RFC related SFRs proper setting

## 5.2 Operation Modes

There are five operation modes for this chip. **Fast Mode** is defined as the CPU running at Fast clock speed. **Slow Mode** is defined as the CPU running at Slow clock speed. When the System clock speed is lower, the power consumption is lower.

**Idle Mode** is entered by setting the IDL bit in PCON SFR. Both Fast and Slow clock can be set as the System clock source in Idle Mode, but Slow clock is better for power saving. In Idle mode, the CPU puts itself to sleep while the on-chip peripherals stay active. The STPPCK bit can be set to furthermore reduce Idle mode current. If STPPCK=1, Timer0/1/2 and UART are stopped in Idle mode. The slower System clock rate also helps current saving. It can be achieved by setup the CLKPSC SFR to divide System clock frequency. Idle mode is terminated by Reset or enabled Interrupts wake up.

**Halt Mode** is entered by setting the PD bit in PCON SFR while STPSCK is cleared. Both Fast and Slow mode can switch to Halt mode. In Halt mode, all clocks stop except the Timer3, WDT, PWMs and LCD could be alive if they are enabled with Slow clock source. Halt mode is terminated by Reset, pin wake up or Timer3/PWM interrupt.

**Stop Mode** is entered by setting the PD bit in PCON SFR while STPSCK is set. This mode is the so-called "Power Down" mode in standard 8051. In Stop mode, all clocks stop. Stop Mode can be terminated by Reset or pin wake up.

***Note:*** Chip cannot enter Stop/Halt mode if INTn pin is low and wakeup is enable. (INTn=0 and EXn=1, n=0,1,2)

| SFR **87h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PCON** | SMOD | – | – | – | GF1 | GF0 | PD | IDL |
| R/W | R/W | – | – | – | R/W | R/W | R/W | R/W |
| Reset | 0 | – | – | – | 0 | 0 | 0 | 0 |

87h.1 **PD:** Power down control bit, set 1 to enter Stop/Halt mode.

87h.0 **IDL:** Idle mode control bit, set 1 to enter Idle mode.

## 6. Interrupt & Wake-up

The chip has an 14-source four-level priority interrupt structure. All enabled Interrupts can wake up CPU from Idle mode, but only the Pin Interrupts can wake up CPU from Stop mode. The Halt mode can be waked up by Time3, PWM and Pin Interrupts. Each interrupt source has its own enable control bit. An interrupt event will set its individual Interrupt Flag, no matter whether its interrupt enable control bit is 0 or 1. The Interrupt vectors and flags are list below.

| Vector | Flag | Description |
|---|---|---|
| 0003 | IE0 | INT0 external pin Interrupt (can wake up Stop/Halt mode) |
| 000B | TF0 | Timer0 Interrupt |
| 0013 | IE1 | INT1 external pin Interrupt (can wake up Stop/Halt mode) |
| 001B | TF1 | Timer1 Interrupt |
| 0023 | RI+TI | Serial Port (UART) Interrupt |
| 002B | TF2+EXF2 | Timer2 Interrupt |
| 0033 | | Reserved for ICE mode use |
| 003B | TF3 | Timer3 Interrupt (can wake up Halt mode) |
| 0043 | PNCIF | Pin change Interrupt (can wake up Stop/Halt mode) |
| 004B | IE2 | INT2 external pin Interrupt (can wake up Stop/Halt mode) |
| 0053 | TKIF+ADIF | Touch Key / ADC Interrupt |
| 005B | SPIF+WCOL | SPI Interrupt |
| 0063 | LBDIF | LBD Interrupt |
| 006B | PWMIF | PWM Interrupt (can wake up Halt mode) |
| 0073 | I2CIF | Master I2C Interrupt |

Interrupt Vector & Flag

### 6.1 Interrupt Enable and Priority Control

The IE and INTE1 SFRs decide whether the pending interrupt is serviced by CPU. The P1WKUP and P3WKUP controls the individual Port1~3 pin's wake-up and interrupt capability. The IP, IPH, IP1 and IP1H SFRs decide the interrupt priority. An interrupt will be serviced as long as an interrupt of equal or higher priority is not already being serviced. If an interrupt of equal or higher level priority is being serviced, the new interrupt will wait until it is finished before being serviced. If a lower priority level interrupt is being serviced, it will be stopped and the new interrupt serviced. When the new interrupt is finished, the lower priority level interrupt that was stopped will be completed.

| SFR **96h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P1WKUP** | P1WKUP | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

96h.7~0 **P1WKUP:** P1.7~P1.0 pin individual Wake-up / Interrupt enable control. 1=Enable

| SFR **A1h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P3WKUP** | P37WK | P36WK | P35WK | P34WK | P25WK | P24WK | P31WK | P30WK |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

A1h.7~0 **P3WKUP:** P3.7~4, P2.5~4, P3.1~0 pin individual Wake-up / Interrupt enable control. 1=Enable

| SFR **A8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **IE** | EA | – | ET2 | ES | ET1 | EX1 | ET0 | EX0 |
| R/W | R/W | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |

A8h.7 **EA:** Global interrupt enable control.
0: Disable all Interrupts.
1: Each interrupt is enabled or disabled by its individual interrupt control bit

A8h.5 **ET2:** Timer2 interrupt enable. 1= Enable

A8h.4 **ES:** Serial Port (UART) interrupt enable. 1= Enable

A8h.3 **ET1:** Timer1 interrupt enable. 1= Enable

A8h.2 **EX1:** External INT1 pin Interrupt enable and Stop/Halt mode wake up enable
0: Disable INT1 pin Interrupt and Stop/Halt mode wake up
1: Enable INT1 pin Interrupt and Stop/Halt mode wake up, it can wake up CPU from Stop/Halt mode no matter EA is 0 or 1.

A8h.1 **ET0:** Timer0 interrupt enable. 1= Enable

A8h.0 **EX0:** External INT0 pin Interrupt enable and Stop/Halt mode wake up enable
0: Disable INT0 pin Interrupt and Stop/Halt mode wake up
1: Enable INT0 pin Interrupt and Stop/Halt mode wake up, it can wake up CPU from Stop/Halt mode no matter EA is 0 or 1.

| SFR **A9h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTE1** | I2CIE | PWMIE | LBDIE | SPIE | ADTKIE | EX2 | PNCIE | ET3 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

A9h.7 **I2CIE:** Master I2C interrupt enable. 1= Enable

A9h.6 **PWMIE:** PWM Interrupt enable and Halt mode wake up enable
0: Disable PWM Interrupt and Halt mode wake up
1: Enable PWM Interrupt and Halt mode wake up, it can wake up CPU from Halt mode no matter EA is 0 or 1.

A9h.5 **LBDIE:** LBD interrupt enable. 1= Enable

A9h.4 **SPIE:** SPI interrupt enable. 1= Enable

A9h.3 **ADTKIE:** ADC / Touch Key interrupt enable. 1= Enable

A9h.2 **EX2:** External INT2 pin Interrupt enable and Stop/Halt mode wake up enable
0: Disable INT2 pin Interrupt and Stop/Halt mode wake up
1: Enable INT2 pin Interrupt and Stop/Halt mode wake up, it can wake up CPU from Stop/Halt mode no matter EA is 0 or 1.

A9h.1 **PNCIE:** Pin change interrupt enable. This bit does not affect the Port1~3 pin's Stop/Halt mode wake up capability.
0: Disable Port1~3 pin change interrupt
1: Enable Port1~3 pin change interrupt

A9h.0 **ET3:** Timer3 Interrupt enable and Halt mode wake up enable
0: Disable Timer3 Interrupt and Halt mode wake up
1: Enable Timer3 Interrupt and Halt mode wake up, it can wake up CPU from Halt mode no matter EA is 0 or 1.

| SFR **B9h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **IPH** | – | – | PT2H | PSH | PT1H | PX1H | PT0H | PX0H |
| R/W | – | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | – | – | 0 | 0 | 0 | 0 | 0 | 0 |

| SFR **B8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **IP** | – | – | PT2 | PS | PT1 | PX1 | PT0 | PX0 |
| R/W | – | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | – | – | 0 | 0 | 0 | 0 | 0 | 0 |

B9h.5, B8h.5 **PT2H, PT2 :** Timer2 Interrupt Priority control. (PT2H, PT2)=
11: Level 3 (highest priority)
10: Level 2
01: Level 1
00: Level 0 (lowest priority)

B9h.4, B8h.4 **PSH, PS :** Serial Port (UART) Interrupt Priority control. Definition as above.

B9h.3, B8h.3 **PT1H, PT1 :** Timer1 Interrupt Priority control. Definition as above.

B9h.2, B8h.2 **PX1H, PX1 :** External INT1 pin Interrupt Priority control. Definition as above.

B9h.1, B8h.1 **PT0H, PT0 :** Timer0 Interrupt Priority control. Definition as above.

B9h.0, B8h.0 **PX0H, PX0 :** External INT0 pin Interrupt Priority control. Definition as above.

| SFR **BBh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **IP1H** | PI2CH | PPWMH | PLBDH | PSPIH | PADTKH | PX2H | PPNCH | PT3H |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| SFR **BAh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **IP1** | PI2C | PPWM | PLBD | PSPI | PADTK | PX2 | PPNC | PT3 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BBh.7, BAh.7 **PI2CH, PI2C :** I2C Interrupt Priority control. Definition as above.

BBh.6, BAh.6 **PPWMH, PPWM :** PWM Interrupt Priority control. Definition as above.

BBh.5, BAh.5 **PLBDH, PLBD :** LBD Interrupt Priority control. Definition as above.

BBh.4, BAh.4 **PSPIH, PSPI :** SPI Interrupt Priority control. Definition as above.

BBh.3, BAh.3 **PADTKH, PADTK :** ADC / Touch Key Interrupt Priority control. Definition as above.

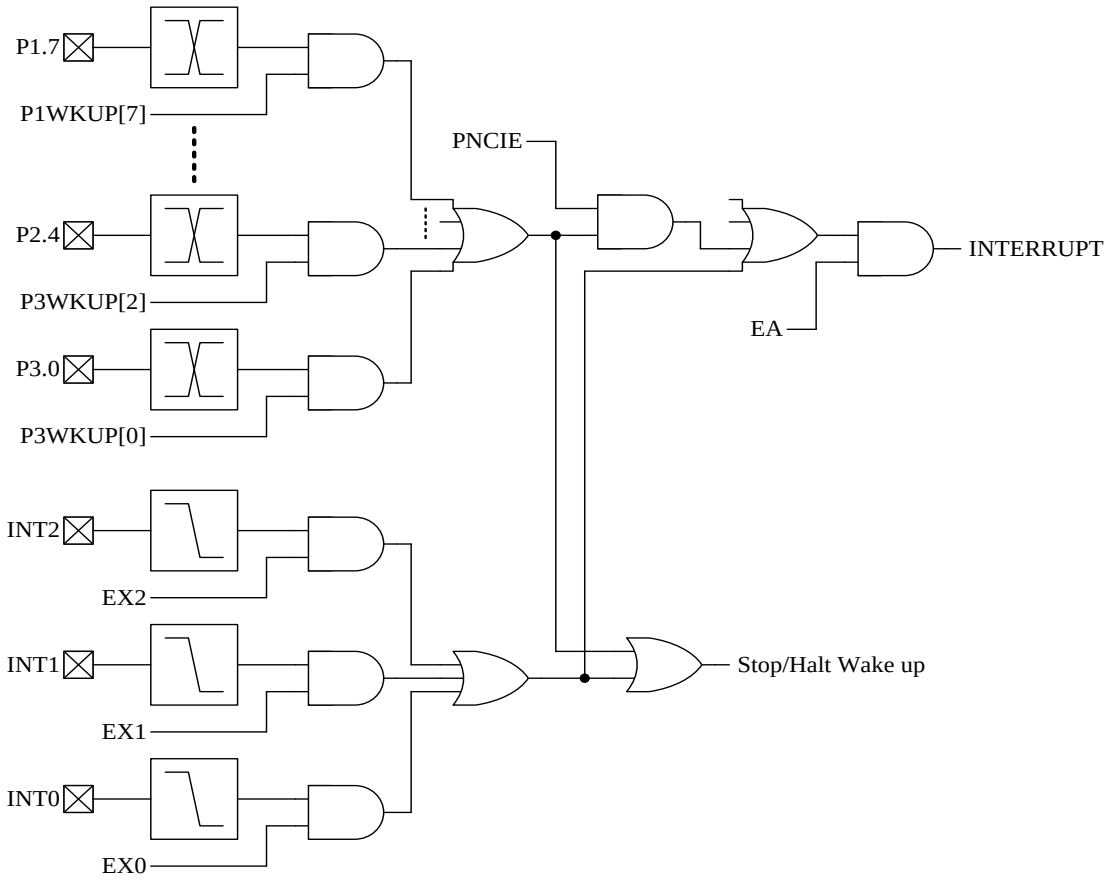
BBh.2, BAh.2 **PX2H, PX2 :** External INT2 pin Interrupt Priority control. Definition as above.

BBh.1, BAh.1 **PPNCH, PPNC :** Pin Change Interrupt Priority control. Definition as above.

BBh.0, BAh.0 **PT3H, PT3 :** Timer3 Interrupt Priority control. Definition as above.

## 6.2 Pin Interrupt & Wake up

Pin Interrupts include INT0 (P3.2), INT1 (P3.3), INT2 (P2.7) and Pin Change Interrupt. These pins also have the Stop/Halt mode wake up capability. INT0 and INT1 are falling edge or low level triggered as the 8051 standard. INT2 is falling edge triggered. Pin Change Interrupt is triggered by P1.7~0 / P3.7~4 / P3.1~0 / P2.5~4 pin state change.




Pin Interrupt & Wake up

| SFR **88h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TCON** | TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

88h.3 **IE1:** External Interrupt 1 (INT1 pin) edge flag.
Set by H/W when an INT1 pin falling edge is detected, no matter the EX1 is 0 or 1.
It is cleared automatically when the program performs the interrupt service routine.

88h.2 **IT1:** External Interrupt 1 control bit
0: Low level active (level triggered) for INT1 pin
1: Falling edge active (edge triggered) for INT1 pin

88h.1 **IE0:** External Interrupt 0 (INT0 pin) edge flag
Set by H/W when an INT0 pin falling edge is detected, no matter the EX0 is 0 or 1.
It is cleared automatically when the program performs the interrupt service routine.

88h.0 **IT0:** External Interrupt 0 control bit
0: Low level active (level triggered) for INT0 pin
1: Falling edge active (edge triggered) for INT0 pin

| SFR **95h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTFLG** | LBDIF | – | TKIF | ADIF | PWMIF | IE2 | PNCIF | TF3 |
| R/W | R/W | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |

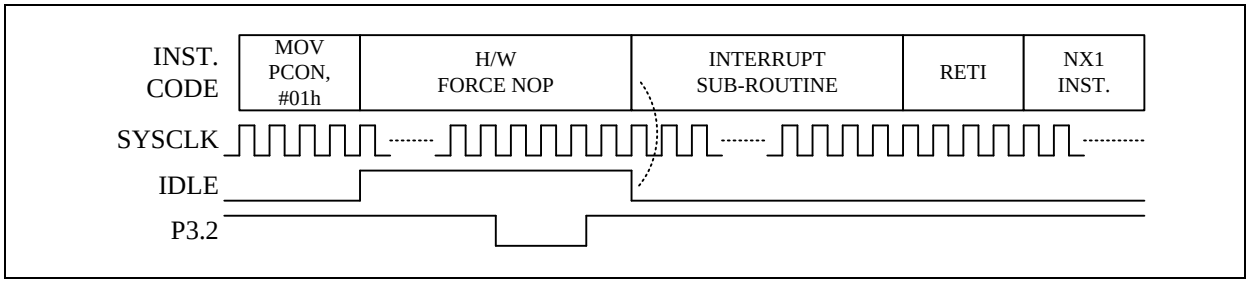
95h.2 **IE2:** External Interrupt 2 (INT2 pin) edge flag
Set by H/W when a falling edge is detected on the INT2 pin, no matter the EX2 is 0 or 1.
It is cleared automatically when the program performs the interrupt service routine.
S/W can write FBh to INTFLG to clear this bit.

95h.1 **PNCIF:** Pin change interrupt flag
Set by H/W when a Port1~3 pin state change is detected and its interrupt enable bit is set (P1WKUP / P3WKUP). PNCIE does not affect this flag's setting.
It is cleared automatically when the program performs the interrupt service routine.
S/W can write FDh to INTFLG to clear this bit.

***Note2:*** S/W can write 0 to clear a flag in the INTFLG, but writing 1 has no effect.

## 6.3 Idle mode Wake up and Interrupt

Idle mode is waked up by enabled Interrupts, which means individual interrupt enable bit (ex: EX0) and EA bit must be both set to 1 to establish Idle mode wake up capability. All enabled Interrupts (Pins, Timers, TK, SPI and UART) can wake up CPU from Idle mode. Upon Idle wake-up, Interrupt service routine is entered immediately. "The first instruction behind IDL (PCON.0) setting" is executed after interrupt service routine return.




EA=EX0=1, Idle mode wake-up and Interrupt by P3.2 (INT0)

| SFR **87h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PCON** | SMOD | – | – | – | GF1 | GF0 | PD | IDL |
| R/W | R/W | – | – | – | R/W | R/W | R/W | R/W |
| Reset | 0 | – | – | – | 0 | 0 | 0 | 0 |

87h.1 **PD:** Power down control bit, set 1 to enter Stop/Halt mode.

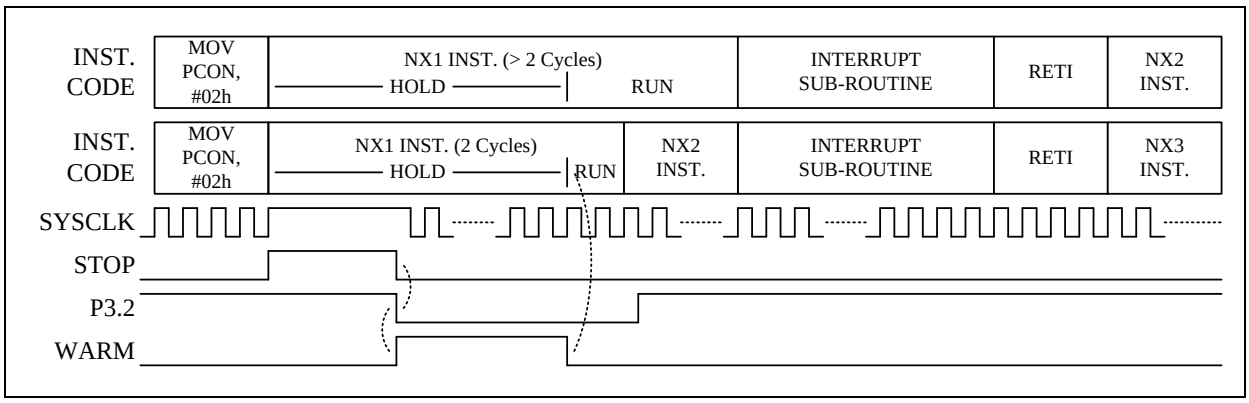
87h.0 **IDL:** Idle mode control bit, set 1 to enter Idle mode.

## 6.4 Halt/Stop mode Wake up and Interrupt

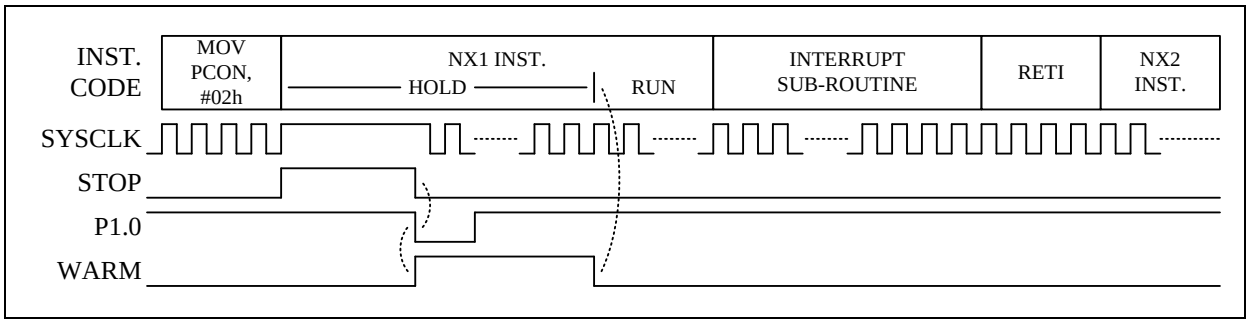
Halt/Stop mode wake up is simple, as long as the individual pin interrupt enable bit (ex: EX0) is set, the pin wake up capability is asserted. Set EX0/EX1/EX2 can enable INT0/INT1/INT2 pins' Halt/Stop mode wake up capability. Set P1WKUP/P3WKUP can enable Port1~3's Halt/Stop mode wake up capability. Upon Halt/Stop wake up, "the first instruction behind PD setting (PCON.1)" is executed immediately before Interrupt service. Interrupt entry requires EA=1 (Pin change also needs PNCIE=1) and trigger state of the pin staying sufficiently long to be sampled by the System clock. This feature allows CPU to enter or not enter Interrupt sub-routine after Halt/Stop mode wake up. Besides pin wakeup, PWM and Timer3 can also wakeup Halt mode if PWMIE/ET3 is set.

***Note:*** Chip cannot enter Stop/Halt mode if INTn pin is low and wakeup is enable. (INTn=0 and EXn=1, n=0,1,2)

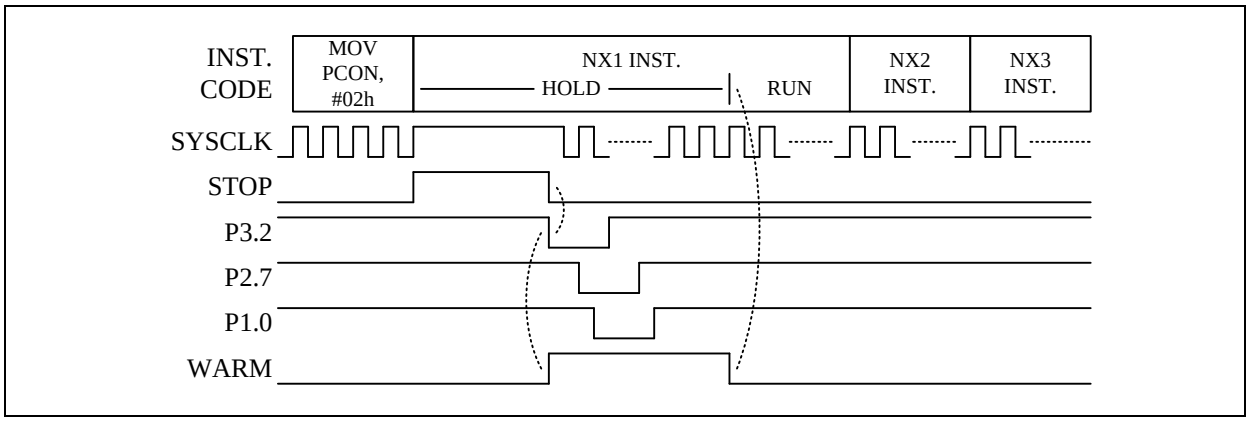
***Note:*** It is recommended to place the NX1/NX2 with NOP Instruction in figures below.



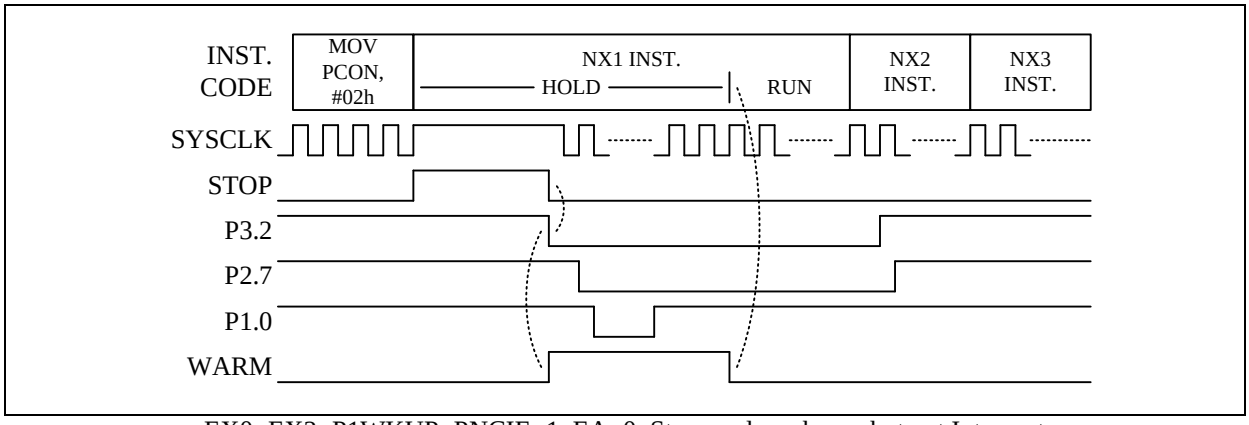

EA=EX0=1, P3.2 (INT0) is sampled after warm-up, Stop mode wake-up and Interrupt




EA=PNCIE=P1WKUP=1, P1.0 change (not need clock sample), Stop mode wake-up and Interrupt




EA=EX0=EX2=P1WKUP=1, PNCIE=0, Stop mode wake-up but not Interrupt. P3.2/P2.7 pulse too narrow




EX0=EX2=P1WKUP=PNCIE=1, EA=0, Stop mode wake-up but not Interrupt

# 7. I/O Ports

The chip has total 40 multi-function I/O pins. All I/O pins follow the standard 8051 “Read-Modify-Write” feature. The instructions that read the SFR rather than the Pin State are the ones that read a port or port bit value, possibly change it, and then rewrite it to the SFR. (ex: ANL P1, A; INC P2; CPL P3.0)

## 7.1 Port1 & Port3

These pins can operate in four different modes as below.

| Mode | Port1, Port3 pin function | | P1.n / P3.n SFR data | Pin State | Resistor Pull-up | Digital Input |
|---|---|---|---|---|---|---|
| | P3.0~P3.2 | Others | | | | |
| **Mode 0** | Pseudo Open Drain | Open Drain | 0 | Drive Low | N | N |
| | | | 1 | Pull-up | Y | Y |
| **Mode 1** | Pseudo Open Drain | Open Drain | 0 | Drive Low | N | N |
| | | | 1 | Hi-Z | N | Y |
| **Mode 2** | CMOS Output | | 0 | Drive Low | N | N |
| | | | 1 | Drive High | N | N |
| **Mode 3** | Alternative Function, such as LCD and ADC | | X (don’t care) | – | N | N |

Port1, Port3 I/O Pin Function Table

If a Port1 or Port3 pin is used for Schmitt-trigger input, S/W must set the I/O pin to Mode0 or Mode1 and set the corresponding Port Data SFR to 1 to disable the pin’s output driving circuitry.

Beside I/O port function, each Port1 and Port3 pin has one or more alternative functions, such as Touch Key, ADC, LCD, PWM and RFC. Port1/Port3 pins also have standard 8051 auxiliary definition such as INT0/1, T0/1/2, or RXD/TXD. These pin functions need to set the pin mode SFR to Mode0 or Mode1 and keep the P1.n / P3.n SFR at 1.

| Pin Name | 8051 | Wake-up | PWM | RFC | ADC | TK | LCD | Mode3 |
|---|---|---|---|---|---|---|---|---|
| P1.0 | T2 | Y | PWM3 | | AD0 | TK0 | | AD0 |
| P1.1 | T2EX | Y | PWM4A | | AD1 | TK1 | | AD1 |
| P1.2 | RXD | Y | | | AD2 | TK2 | | AD2 |
| P1.3 | TXD | Y | PWM5 | | AD3 | TK3 | | AD3 |
| P1.4 | | Y | | RFC2 | AD4 | TK4 | | AD4 |
| P1.5 | | Y | | RFC1 | AD5 | TK5 | | AD5 |
| P1.6 | | Y | | RFC0 | AD6 | TK6 | | AD6 |
| P1.7 | | Y | | RFCX | AD7 | TK7 | | AD7 |
| P3.0 | RXD | Y | | | | | SEG44 | SEG44 |
| P3.1 | TXD | Y | | | | | SEG45 | SEG45 |
| P3.2 | INT0 | Y | | | | | SEG46 | SEG46 |
| P3.3 | INT1 | Y | | | | | SEG47 | SEG47 |
| P3.4 | T0 | Y | PWM2 | | AD8 | TK8 | | AD8 |
| P3.5 | T1 | Y | PWM1 | | AD9 | TK9 | | AD9 |
| P3.6 | | Y | PWM0N/4B | | | | | |
| P3.7 | | Y | PWM0P | | | | | |

Port1, Port3 multi-function Table

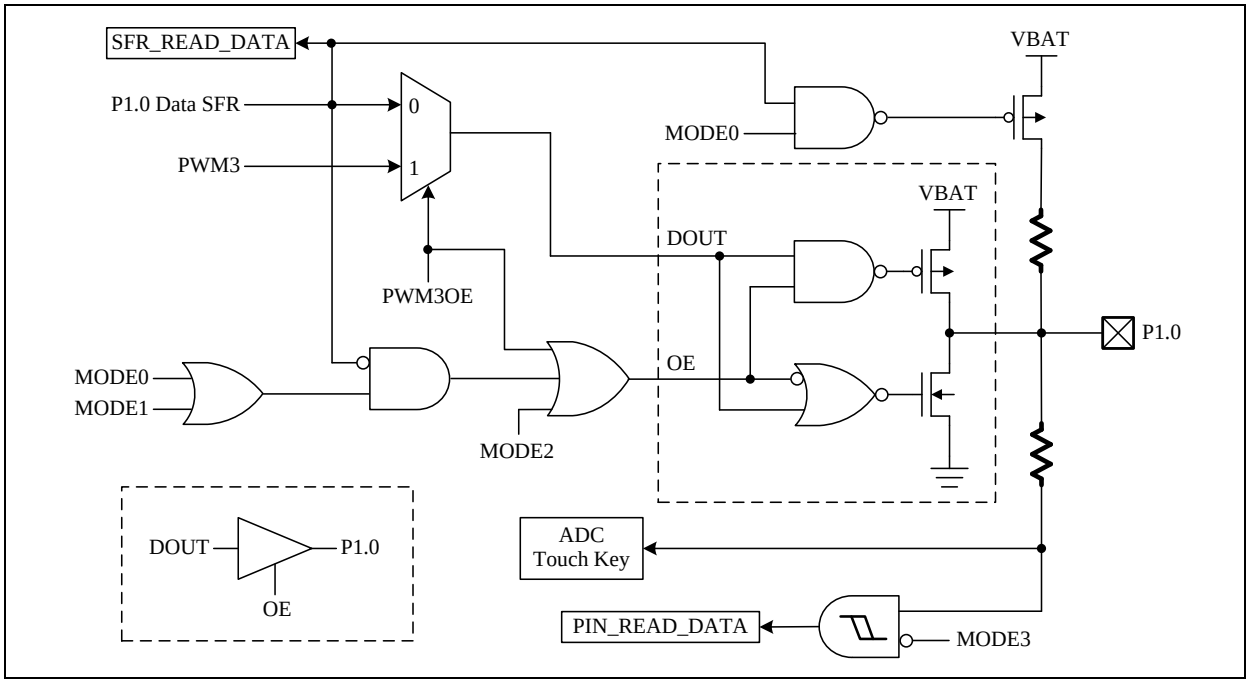
The necessary SFR setting for Port1/Port3 pin's alternative functions is list below.

| Alternative Function | Mode | P1.n / P3.n SFR data | Pin State |
|---|---|---|---|
| T0, T1, T2, T2EX, INT0, INT1 | **0** | 1 | Input with Pull-up |
| | **1** | 1 | Input |
| RXD, TXD | **0** | 1 | Input with Pull-up / (Pseudo) Open Drain Output |
| | **1** | 1 | Input / (Pseudo) Open Drain Output |
| RFCX, RFC0~2 | **0** | 1 | RFC clock oscillation |
| PWM0~5 | **2** | X | PWM Output (CMOS Push-Pull) |
| SEG44~SEG47 | **3** | X | LCD Waveform Output |
| TK0~TK9 | **2** | 0 | Touch Key Idling or Scanning |
| AD0~9 | **3** | X | ADC analog Input |

For tables above, a "**CMOS Output**" pin means it can sink and drive at least 4mA current. It is not recommended to use such pin as input function.

An "**Open Drain**" pin means it can sink at least 4mA current but only drive a small current (< 20uA). It can be used as input or output function and typically needs an external pull up resistor.

An 8051 standard pin is a "**Pseudo Open Drain**" pin. It can sink at least 4mA current when output is at low level, and drives at least 4mA current for 1~2 clock cycle when output transits from low to high, then keeps driving a small current (< 20uA) to maintain the pin at high level. It can be used as input or output function.




P1.0 Pin Structure

| SFR **90h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P1** | P1.7 | P1.6 | P1.5 | P1.4 | P1.3 | P1.2 | P1.1 | P1.0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

90h.7~0 **P1:** Port1 data

| SFR **A2h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P1MODL** | P1MOD3 | | P1MOD2 | | P1MOD1 | | P1MOD0 | |
| R/W | R/W | | R/W | | R/W | | R/W | |
| Reset | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |

A2h.7~6 **P1MOD3:** P1.3 pin control.
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P1.3 is ADC input.

A2h.5~4 **P1MOD2:** P1.2 pin control.
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P1.2 is ADC input.

A2h.3~2 **P1MOD1:** P1.1 pin control.
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P1.1 is ADC input.

A2h.1~0 **P1MOD0:** P1.0 pin control.
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P1.0 is ADC input.

| SFR **A3h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P1MODH** | P1MOD7 | | P1MOD6 | | P1MOD5 | | P1MOD4 | |
| R/W | R/W | | R/W | | R/W | | R/W | |
| Reset | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

A3h.7~6 **P1MOD7:** P1.7 pin control.
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P1.7 is ADC input.

A3h.5~4 **P1MOD6:** P1.6 pin control.
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P1.6 is ADC input.

A3h.3~2 **P1MOD5:** P1.5 pin control.
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P1.5 is ADC input.

A3h.1~0 **P1MOD4:** P1.4 pin control.
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P1.4 is ADC input.

| SFR **AFh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **RFCON** | P1RFC | | T0SEL | | RFCPSC | | RFCS | |
| R/W | R/W | | R/W | | R/W | | R/W | |
| Reset | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |

AFh.7~6 **P1RFC:** P1.7~P1.4 pin RFC mode control.
00: P1.7~P1.4 are not RFC pins
01: P1.7 and P1.6 are RFC pins, P1.5 and P1.4 are not RFC pins
10: P1.7~P1.5 are RFC pins, P1.4 is not RFC pin
11: P1.7~P1.4 are RFC pins

| SFR **94h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **OPTION** | SXTGAIN | | STPPCK | SXTKICK | UART1W | UARTP1 | T2SEL | T1SEL |
| R/W | R/W | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

94h.2 **UARTP1:** UART pin select
0: P3.0 / P3.1 is UART RXD / TXD
1: P1.2 / P1.3 is UART RXD / TXD

| SFR **B0h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P3** | P3.7 | P3.6 | P3.5 | P3.4 | P3.3 | P3.2 | P3.1 | P3.0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

B0h.7~0 **P3:** Port3 data

| SFR **A4h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P3MODL** | P3MOD3 | | P3MOD2 | | P3MOD1 | | P3MOD0 | |
| R/W | R/W | | R/W | | R/W | | R/W | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

A4h.7~6 **P3MOD3:** P3.3 pin control.
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P3.3 is LCD Segment output.

A4h.5~4 **P3MOD2:** P3.2 pin control.
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P3.2 is LCD Segment output.

A4h.3~2 **P3MOD1:** P3.1 pin control.
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P3.1 is LCD Segment output.

A4h.1~0 **P3MOD0:** P3.0 pin control.
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P3.0 is LCD Segment output.

| SFR **A5h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P3MODH** | P3MOD7 | | P3MOD6 | | P3MOD5 | | P3MOD4 | |
| R/W | R/W | | R/W | | R/W | | R/W | |
| Reset | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

A5h.7~6 **P3MOD7:** P3.7 pin control.
00: Mode0
01: Mode1
10: Mode2
11: Not defined

A5h.5~4 **P3MOD6:** P3.6 pin control.
00: Mode0
01: Mode1
10: Mode2
11: Not defined

A5h.3~2 **P3MOD5:** P3.5 pin control.
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P3.5 is ADC input

A5h.1~0 **P3MOD4:** P3.4 pin control.
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P3.4 is ADC input

| SFR **D7h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWMOE** | PWM5OE | PWM4BOE | PWM4AOE | PWM3OE | PWM2OE | PWM1OE | PWM0POE | PWM0NOE |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

D7h.7 **PWM5OE:** PWM5 output to P1.3
D7h.6 **PWM4BOE:** PWM4 output to P3.6
D7h.5 **PWM4AOE:** PWM4 output to P1.1
D7h.4 **PWM3OE:** PWM3 output to P1.0
D7h.3 **PWM2OE:** PWM2 output to P3.4
D7h.2 **PWM1OE:** PWM1 output to P3.5
D7h.1 **PWM0POE:** PWM0P output to P3.7
D7h.0 **PWM0NOE:** PWM0N output to P3.6

### 7.2 Port0, Port2 & Port4

These pins are shared with LCD, LED, I2C, SPI and crystal oscillator. If a Port0/2/4 pin is defined as I/O pin, it can be used as CMOS push-pull output or Schmitt-trigger input. The pin's pull up function is enable while SFR bit PxOE.n=0 and Px.n=1.

| Port0 / Port2 / Port4 pin function | PxOE.n | Px.n SFR data | Pin State | Resistor Pull-up | Digital Input |
|---|---|---|---|---|---|
| Input | 0 | 0 | Hi-Z | N | Y |
| | 0 | 1 | Pull-up | Y | Y |
| CMOS Output | 1 | 0 | Drive Low | N | N |
| | 1 | 1 | Drive High | N | N |

P2.6~P2.0 & Port0 I/O Pin Function Table

| Pin Name | Wake-up | ADC | SPI / I2C | SXT | LCD/LED | Others |
|---|---|---|---|---|---|---|
| P0.0 | | | MSCL | | SEG40 | |
| P0.1 | | | MSDA | | SEG39 | |
| P0.2 | | | | | SEG38 | TCO |
| P0.3 | | | | | SEG37 | |
| P0.4 | | | | | SEG36 | |
| P0.5 | | | | | SEG35 | |
| P0.6 | | | | | SEG34 | |
| P0.7 | | AD11 | | SX1 | | |
| P2.0 | | AD10 | | SX2 | | |
| P2.1 | | | | | SEG33 | |
| P2.2 | | | | | SEG32 | |
| P2.3 | | | | | SEG31 | |
| P2.4 | | | MOSI | | SEG43 | |
| P2.5 | | | SCK | | SEG42 | |
| P2.6 | | | MISO | | SEG41 | |
| P2.7 | Y | | | | | INT2, RSTn |
| P4.0 | | | | | SEG30 | |
| P4.1 | | | | | SEG29 | |
| P4.2 | | | | | SEG28 | |
| P4.3 | | | | | SEG27 | |
| P4.4 | | | | | SEG26 | |
| P4.5 | | | | | SEG25 | |
| P4.6 | | | | | SEG24 | |
| P4.7 | | | | | SEG23 | |

Port0, Port2 & Port4 multi-function Table

The necessary SFR setting for Port0/Port2/Port4 pin's alternative functions is list below.

| Alternative Function | PxOE.n | Px.n | Pin State |
|---|---|---|---|
| MSCL, MSDA, MOSI, SCK, MISO, TCO | 0 | 0 | I2C/SPI Communicate |
| AD10~AD11 | 0 | 0 | ADC analog Input |
| SX1, SX2 | 0 | 1 | Crystal oscillation |
| SEG23~SEG43 | 0 | X | LCD/LED Output |

| SFR **80h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P0** | P0.7 | P0.6 | P0.5 | P0.4 | P0.3 | P0.2 | P0.1 | P0.0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

80h.7~0 **P0:** Port0 data, also controls the P0.n pin's pull-up function. If the P0.n SFR data is "1" and the corresponding P0OE.n=0 (input mode), the pull-up is enabled.

| SFR **A0h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P2** | P2.7 | P2.6 | P2.5 | P2.4 | P2.3 | P2.2 | P2.1 | P2.0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

A0h.7~0 **P2:** Port2 data, also controls the P2.n pin's pull-up function. If the P2.n SFR data is "1" and the corresponding P2OE.n=0 (input mode), the pull-up is enabled.

| SFR **E8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P4** | P4.7 | P4.6 | P4.5 | P4.4 | P4.3 | P4.2 | P4.1 | P4.0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

E8h.7~0 **P4:** Port4 data, also controls the P4.n pin's pull-up function. If the P4.n SFR data is "1" and the corresponding P4OE.n=0 (input mode), the pull-up is enabled.

| SFR **91h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P0OE** | P0OE | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

91h.7~0 **P0OE:** Port0 CMOS Push-Pull output enable control, 1=Enable.

| SFR **93h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P2OE** | P2OE | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

93h.7~0 **P2OE:** Port2 CMOS Push-Pull output enable control, 1=Enable.

| SFR **A6h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **P4OE** | P4OE | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

A6h.7~0 **P4OE:** Port4 CMOS Push-Pull output enable control, 1=Enable.

| SFR **D8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CLKCON** | FCKTYPE | | SELFCK | SCKTYPE | STPFSUB | STPSCK | CLKPSC | |
| R/W | R/W | | R/W | R/W | R/W | R/W | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

D8h.4 **SCKTYPE:** Set 1 to enable P0.7 and P2.0 pin's SXT oscillation mode

***Note:*** In crystal mode, user should set the P0.7/P2.0 (SXT) pins as Input with Pull-up.

| SFR **92h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PINMODE** | – | P2HSEG | | P2LSEG | | P0SEG | | |
| R/W | – | R/W | | R/W | | R/W | | |
| Reset | – | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

92h.6~5 **P2HSEG:** P2.4~P2.6 pin LCD/LED mode control.
- 00: P2.4~P2.6 are I/O pins
- 01: P2.4 and P2.5 are I/O pins, P2.6 is LCD/LED Segment pin
- 10: P2.4 is I/O pin, P2.5 and P2.6 are LCD/LED Segment pins
- 11: P2.4~P2.6 are LCD/LED Segment pins

92h.4~3 **P2LSEG:** P2.1~P2.3 pin LCD/LED mode control.
- 00: P2.1~P2.3 are I/O pins
- 01: P2.1 and P2.2 are I/O pins, P2.3 is LCD/LED Segment pin
- 10: P2.1 is I/O pin, P2.2 and P2.3 are LCD/LED Segment pins
- 11: P2.1~P2.3 are LCD/LED Segment pins

92h.2~0 **P0SEG:** Port0 LCD/LED mode control.
- 000: P0.0~P0.6 are I/O pins
- 001: P0.0~P0.5 are I/O pins, P0.6 is LCD/LED Segment pin
- 010: P0.0~P0.4 are I/O pins, P0.5~P0.6 are LCD/LED Segment pins
- 011: P0.0~P0.3 are I/O pins, P0.4~P0.6 are LCD/LED Segment pins
- 100: P0.0~P0.2 are I/O pins, P0.3~P0.6 are LCD/LED Segment pins
- 101: P0.0~P0.1 are I/O pins, P0.2~P0.6 are LCD/LED Segment pins
- 110: P0.0 is I/O pin, P0.1~P0.6 are LCD/LED Segment pins
- 111: P0.0~P0.6 are LCD/LED Segment pins

| SFR **AEh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TKCON2** | P4SEG | | | | TKREFC | | | |
| R/W | R/W | | | | R/W | | | |
| Reset | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

AEh.7~4 **P4SEG:** Port4 LCD/LED mode control.
- 0000: P4.0~P4.7 are I/O pins
- 0001: P4.0~P4.6 are I/O pins, P4.7 is LCD/LED Segment pin
- 0010: P4.0~P4.5 are I/O pins, P4.6~P4.7 are LCD/LED Segment pins
- 0011: P4.0~P4.4 are I/O pins, P4.5~P4.7 are LCD/LED Segment pins
- 0100: P4.0~P4.3 are I/O pins, P4.4~P4.7 are LCD/LED Segment pins
- 0101: P4.0~P4.2 are I/O pins, P4.3~P4.7 are LCD/LED Segment pins
- 0110: P4.0~P4.1 are I/O pins, P4.2~P4.7 are LCD/LED Segment pins
- 0111: P4.0 is I/O pin, P4.1~P4.7 are LCD/LED Segment pins
- 1000: P4.0~P4.7 are LCD/LED Segment pins

| SFR **D3h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX2** | P07ADC | P20ADC | P02TCO | LBDEDGE | VBGE | VBGOUT | IAPTE | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

D3h.7 **P07ADC:** P0.7 ADC pin select. 1=Select P0.7 as ADC input

D3h.6 **P20ADC:** P2.0 ADC pin select. 1=Select P2.0 as ADC input

D3h.5 **P02TCO:** P0.2 TCO pin select. 1=Select P0.2 as $F_{SYSCLK}/2$ output

D3h.2 **VBGOUT:** P1.1 VBG pin select. 1=Select P1.1 as VBG output

| SFR **BCh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SPCON** | SPEN | MSTR | CPOL | CPHA | – | LSBF | SPCR | |
| R/W | R/W | R/W | R/W | R/W | – | R/W | R/W | |
| Reset | 0 | 0 | 0 | 0 | – | 0 | 0 | 0 |

BCh.7 **SPEN:** SPI Enable.
0: SPI Disable
1: SPI Enable, P2.4~P2.6 are SPI functional pins.

| SFR **E1h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **MICON** | MIEN | MIACKO | MIIF | MIACKI | MISTART | MISTOP | MICR | |
| R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

E1h.7 **MIEN**:Master I2C enable
0: I2C Disable
1: I2C Enable, P0.0~P0.1 are I2C functional pins.

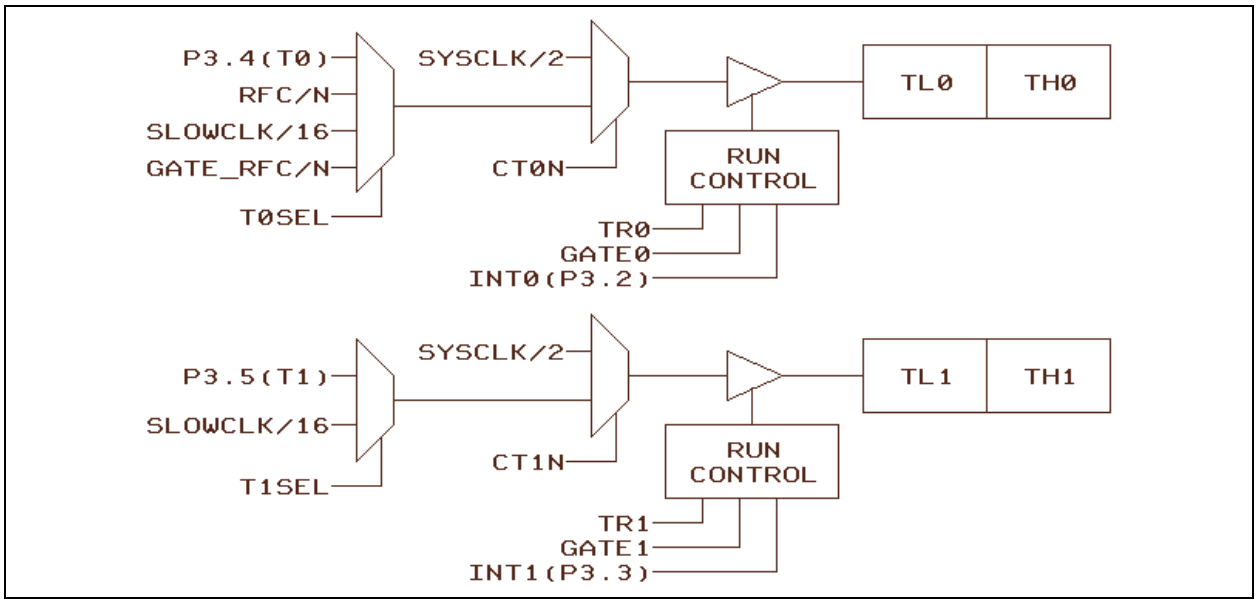
# 8. Timers

Timer0, Timer1 and Timer2 are provided as standard 8051 compatible timer/counter. Timer3 is provided for a real-time clock count.

## 8.1 Timer0 / Timer1 / Timer2

Compare to the traditional 12T 8051, the chip's Timer0/1/2 use 2 System clock cycle as the time base unit. That is, in timer mode, these timers increase at every "2 System clock" rate; in counter mode, T0, T1 and T2 pins' input pulse must be wider than 2 System clock to be sampled by this chip.

In addition to standard 8051 timers function, SLOWCLK/16 can replace P3.4(T0), P3.5(T1) and P1.0(T2) pins as the Timer0, Timer1 and Timer2 counter mode input. Timer0 also supports RFC counting. The RFC clock divided/gated signal can also replace T0 pin as the Timer0's event count input.




Timer0 and Timer1 structure

TCON and TMOD set the operation mode and control the running and interrupt generation of Timer0/1, with the timer/counter values stored in two pairs of 8-bit registers (TL0, TH0, and TL1, TH1).

| SFR **88h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TCON** | TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

88h.7 **TF1:** Timer1 overflow flag
Set by H/W when Timer/Counter 1 overflows
Cleared by H/W when CPU vectors into the interrupt service routine.

88h.6 **TR1:** Timer1 run control
0: Timer1 stops
1: Timer1 runs

88h.5 **TF0:** Timer0 overflow flag
Set by H/W when Timer/Counter 0 overflows
Cleared by H/W when CPU vectors into the interrupt service routine.

88h.4 **TR0:** Timer0 run control
0: Timer0 stops
1: Timer0 runs

| SFR **89h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TMOD** | GATE1 | CT1N | TMOD1 | | GATE0 | CT0N | TMOD0 | |
| R/W | R/W | R/W | R/W | | R/W | R/W | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

89h.7 **GATE1:** Timer1 gating control bit
0: Timer1 enable when TR1 bit is set
1: Timer1 enable only while the INT1 pin is high and TR1 bit is set

89h.6 **CT1N:** Timer1 Counter/Timer select bit
0: Timer mode, Timer1 data increases at 2 System clock cycle rate
1: Counter mode, Timer1 data increases at T1 pin or SLOWCLK/16 falling edge

89h.5~4 **TMOD1:** Timer1 mode select
00: 8-bit timer/counter (TH1) and 5-bit prescaler (TL1)
01: 16-bit timer/counter
10: 8-bit auto-reload timer/counter (TL1). Reloaded from TH1 at overflow.
11: Timer1 stops

89h.3 **GATE0:** Timer0 gating control bit
0: Timer0 enable when TR0 bit is set
1: Timer0 enable only while the INT0 pin is high and TR0 bit is set

89h.2 **CT0N:** Timer0 Counter/Timer select bit
0: Timer mode, Timer0 data increases at 2 System clock cycle rate
1: Counter mode, Timer0 data increases at T0 pin, SLOWCLK/16 or RFC/N falling edge

89h.1~0 **TMOD0:** Timer0 mode select
00: 8-bit timer/counter (TH0) and 5-bit prescaler (TL0)
01: 16-bit timer/counter
10: 8-bit auto-reload timer/counter (TL0). Reloaded from TH0 at overflow.
11: TL0 is an 8-bit timer/counter. TH0 is an 8-bit timer/counter using Timer1's TR1 and TF1 bits.

| SFR **8Ah** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TL0** | TL0 | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

8Ah.7~0 **TL0:** Timer0 data low byte

| SFR **8Bh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TL1** | TL1 | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

8Bh.7~0 **TL1:** Timer1 data low byte

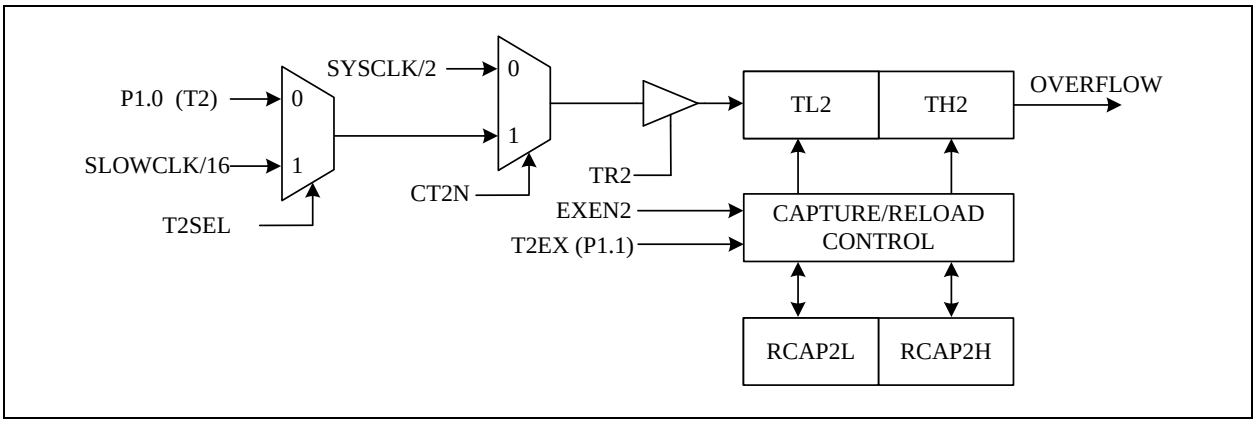
| SFR **8Ch** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TH0** | TH0 | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

8Ch.7~0 **TH0:** Timer0 data high byte

| SFR **8Dh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TH1** | TH1 | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

8Dh.7~0 **TH1:** Timer1 data high byte

Timer2 is controlled through the TCON2 register with the low and high bytes of Timer/Counter2 stored in TL2 and TH2 and the low and high bytes of the Timer2 reload/capture registers stored in RCAP2L and RCAP2H. Timer2 also supports SLOWCLK/16 event count mode.




Timer2 structure

| SFR **C8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **T2CON** | TF2 | EXF2 | RCLK | TCLK | EXEN2 | TR2 | CT2N | CPRL2N |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

C8h.7 **TF2:** Timer2 overflow flag
Set by H/W when Timer/Counter 2 overflows unless RCLK=1 or TCLK=1. This bit must be cleared by S/W.

C8h.6 **EXF2:** T2EX interrupt pin falling edge flag
Set when a capture or a reload is caused by a negative transition on T2EX pin if EXEN2=1. This bit must be cleared by S/W.

C8h.5 **RCLK:** UART receive clock control bit
0: Use Timer1 overflow as receive clock for serial port in mode 1 or 3
1: Use Timer2 overflow as receive clock for serial port in mode 1 or 3

C8h.4 **TCLK:** UART transmit clock control bit
0: Use Timer1 overflow as transmit clock for serial port in mode 1 or 3
1: Use Timer2 overflow as transmit clock for serial port in mode 1 or 3

C8h.3 **EXEN2:** T2EX pin enable
0: T2EX pin disable
1: T2EX pin enable, it cause a capture or reload when a negative transition on T2EX pin is detected if RCLK=TCLK=0

C8h.2 **TR2:** Timer2 run control
0: Timer2 stops
1: Timer2 runs

C8h.1 **CT2N:** Timer2 Counter/Timer select bit
0: Timer mode, Timer2 data increases at 2 System clock cycle rate
1: Counter mode, Timer2 data increases at T2 pin or SLOWCLK/16 falling edge

C8h.0 **CPRL2N:** Timer2 Capture/Reload control bit
0: Reload mode, auto-reload on Timer2 overflows or negative transitions on T2EX pin if EXEN2=1.
1: Capture mode, capture on negative transitions on T2EX pin if EXEN2=1.
If RCLK=1 or TCLK=1, CPRL2N is ignored and timer is forced to auto-reload on Timer2 overflow.

| SFR **CAh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **RCP2L** | RCP2L | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

CAh.7~0 **RCP2L:** Timer2 reload/capture data low byte

| SFR **CBh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **RCP2H** | RCP2H | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

CBh.7~0 **RCP2H:** Timer2 reload/capture data high byte

| SFR **CCh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TL2** | TL2 | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

CCh.7~0 **TL2:** Timer2 data low byte

| SFR **CDh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TH2** | TH2 | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

CDh.7~0 **TH2:** Timer2 data high byte

| SFR **AFh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **RFCON** | P1RFC | | T0SEL | | RFCPSC | | RFCS | |
| R/W | R/W | | R/W | | R/W | | R/W | |
| Reset | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |

AFh.5~4 **T0SEL:** Timer0 Counter mode (CT0N=1) input select
00: P3.4 pin (8051 standard)
01: RFC clock divided by 1/4/16/64
10: Slow clock divided by 16 (SLOWCLK/16)
11: RFC clock divided by 1/4/16/64 gated by Timer2 overflow

AFh.3~2 **RFCPSC:** RFC clock divider to Timer0
00: divided by 64
01: divided by 16
10: divided by 4
11: divided by 1

| SFR **94h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **OPTION** | SXTGAIN | | STPPCK | SXTKICK | UART1W | UARTP1 | T2SEL | T1SEL |
| R/W | R/W | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

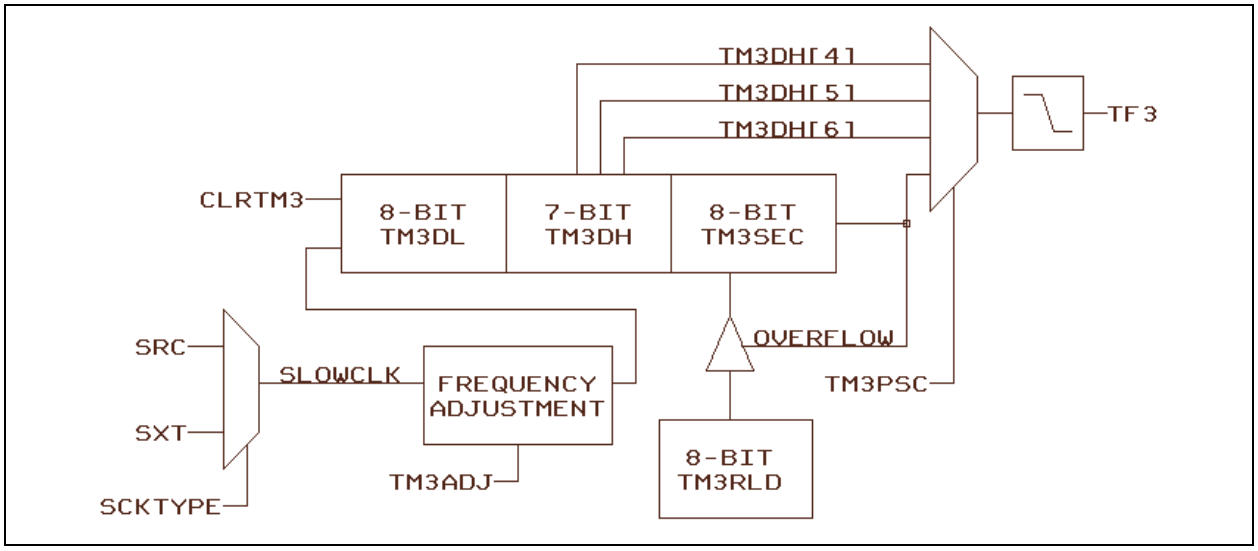
94h.1 **T2SEL:** Timer2 Counter mode (CT2N=1) input select
0: P1.0 pin (8051 standard)
1: Slow clock divided by 16 (SLOWCLK/16)

94h.0 **T1SEL:** Timer1 Counter mode (CT1N=1) input select
0: P3.5 pin (8051 standard)
1: Slow clock divided by 16 (SLOWCLK/16)

***Note:*** for SLOWCLK/16 sampling, System clock must not be slower than SLOWCLK/4.

## 8.2 Timer3

The 23-bit wide Timer3 is reloadable for its 8-bit MSB when overflow. Its time base is Slow clock (SRC or SXT). Timer3 can generate interrupt periodically at different rate, and its counting data can be read out by CPU. It is recommended to read Timer3 data in Slow mode. While CPU clock is switched to Fast clock, the clock source of CPU and Timer3 are different, CPU may read a "under changing Timer3 data". User F/W must have some filter mechanism to avoid such kind un-stability. On the contrast, Timer3 interrupt has no ambiguous behavior no matter what the CPU clock source is.




Timer3 Structure

Timer3 can control its counting rate by the TM3ADJ SFR. This feature compensates the 32768 SXT crystal's in-accuracy. While TM3ADJ=0, Timer3 increase its data count normally at each Slow clock cycle. If TM3ADJ is set to positive adjustment, Timer3 increase its data count by 2 in particular Slow clock cycles, resulting a faster counting rate. If TM3ADJ is set to negative adjustment, Timer3 stop increase in particular Slow clock cycles, resulting a slower counting rate. The adjustment is 0.477ppm per step, and the total adjustable range is ± 61ppm.

| SFR **F8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX1** | – | – | TKSOC | ADSOC | CLRWDT | CLRTM3 | STPRFC | DPSEL |
| R/W | – | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | – | – | 0 | 0 | 0 | 0 | 0 | 0 |

F8h.2 **CLRTM3:** Set 1 to Clear Timer3 and force TM3SEC reload

| SFR **D2h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **WDTCON** | IAPHTW | TM3PSC | | | WDTMOD | | WDTPSC | |
| R/W | R/W | R/W | | | R/W | | R/W | |
| Reset | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

D2h.6~4 **TM3PSC:** Timer3 Interrupt rate

000: Timer3 interrupt occurs when 23 bit count data overflow
001: Timer3 interrupt rate is 32768 Slow clock cycles (1.0 second for SXT)
010: Timer3 interrupt rate is 16384 Slow clock cycles (0.5 second for SXT)
011: Timer3 interrupt rate is 8192 Slow clock cycles (0.25 second for SXT)
100: Timer3 interrupt rate is 4096 Slow clock cycles (0.125 second for SXT)
101: Timer3 interrupt rate is 2048 Slow clock cycles (62.5 ms for SXT)
110: Timer3 interrupt rate is 1024 Slow clock cycles (31.2 ms for SXT)
111: Timer3 interrupt rate is 512 Slow clock cycles (15.6 ms for SXT)

| SFR **95h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTFLG** | LBDIF | – | TKIF | ADIF | PWMIF | IE2 | PNCIF | TF3 |
| R/W | R/W | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |

95h.0 **TF3:** Timer3 Interrupt Flag
Set by H/W when Timer3 reaches TM3PSC setting cycles. Cleared automatically when the program performs the interrupt service routine. S/W can write FEh to INTFLG to clear this bit. (***Note2***)

| SFR **B3h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TM3SEC** | TM3SEC | | | | | | | |
| R/W | R | | | | | | | |
| Reset | – | – | – | – | – | – | – | – |

B3h.7~0 **TM3SEC:** Timer3 count data bit 22~15

| SFR **B4h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TM3DL** | TM3DL | | | | | | | |
| R/W | R | | | | | | | |
| Reset | – | – | – | – | – | – | – | – |

B4h.7~0 **TM3DL:** Timer3 count data bit 7~0

| SFR **B5h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TM3DH** | – | TM3DH | | | | | | |
| R/W | – | R | | | | | | |
| Reset | – | – | – | – | – | – | – | – |

B5h.6~0 **TM3DH:** Timer3 count data bit 14~8

| SFR **B6h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TM3RLD** | TM3RLD | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

B6h.7~0 **TM3RLD:** Timer3 overflow reload data for Timer3 bit 22~15 (TM3SEC)

| SFR **B7h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TM3ADJ** | TM3ADJS | TM3ADJ | | | | | | |
| R/W | R/W | R/W | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

B7h.7 **TM3ADJS:** Timer3 adjustment sign
0: Timer3 positive adjust, to increase Timer3 counting rate
1: Timer3 negative adjust, to decrease Timer3 counting rate

B7h.6~0 **TM3ADJ:** Timer3 adjust magnitude, 0.477 ppm per LSB.
The adjustment is calculated as ±TM3ADJ*0.477ppm. The total adjustable range is ± 61ppm.

***Note6:*** also refer to Section 6 for more information about Timer0/1/2/3 Interrupt enable and priority.

## 9. UART

The UART uses SCON and SBUF SFRs. SCON is the control register, SBUF is the data register. Data is written to SBUF for transmission and SBUF is read to obtain received data. The received data and transmitted data registers are completely independent. In addition to standard 8051's full duplex mode, this chip also provides one wire mode. If the UART1W bit is set, both transmit and receive data use P3.1 pin. The chip also provide extra baud rate generator to save Timers loading. The RXD/TXD can be assigned to P3.0/P3.1 or P1.2/P1.3 pins.

| SFR **94h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **OPTION** | SXTGAIN | | STPPCK | SXTKICK | UART1W | UARTP1 | T2SEL | T1SEL |
| R/W | R/W | | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

94h.3 **UART1W:** One wire UART mode enable, both TXD/RXD use P3.1 pin
0: Disable one wire UART mode
1: Enable one wire UART mode

94h.2 **UARTP1:** UART pin select
0: P3.0 / P3.1 is UART RXD / TXD
1: P1.2 / P1.3 is UART RXD / TXD

| SFR **98h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SCON** | SM0 | SM1 | SM2 | REN | TB8 | RB8 | TI | RI |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

98h.7~6 **SM0,SM1:** Serial port mode select bit 0,1
00: Mode0: 8 bit shift register, Baud Rate = $F_{SYSCLK}$ / 2
01: Mode1: 8 bit UART, Baud Rate is variable
10: Mode2: 9 bit UART, Baud Rate = $F_{SYSCLK}$ / 32 or / 64
11: Mode3: 9 bit UART, Baud Rate is variable

98h.5 **SM2:** Serial port mode select bit 2
SM2 enables multiprocessor communication over a single serial line and modifies the above as follows. In Modes 2 & 3, if SM2 is set then the received interrupt will not be generated if the received ninth data bit is 0. In Mode 1, the received interrupt will not be generated unless a valid stop bit is received. In Mode 0, SM2 should be 0.

98h.4 **REN:** UART reception enable
0: Disable reception
1: Enable reception

98h.3 **TB8:** Transmit Bit 8, the ninth bit to be transmitted in Mode 2 and 3

98h.2 **RB8:** Receive Bit 8, contains the ninth bit that was received in Mode 2 and 3 or the stop bit in Mode 1 if SM2=0

98h.1 **TI:** Transmit interrupt flag
Set by H/W at the end of the eighth bit in Mode 0, or at the beginning of the stop bit in other modes. Must be cleared by S/W.

98h.0 **RI:** Receive interrupt flag
Set by H/W at the end of the eighth bit in Mode 0, or at the sampling point of the stop bit in other modes. Must be cleared by S/W.

| SFR **99h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SBUF** | SBUF | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | – | – | – | – | – | – | – | – |

99h.7~0 **SBUF:** UART transmit and receive data. Transmit data is written to this location and receive data is read from this location, but the paths are independent.

| SFR **87h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PCON** | SMOD | – | – | – | GF1 | GF0 | PD | IDL |
| R/W | R/W | – | – | – | R/W | R/W | R/W | R/W |
| Reset | 0 | – | – | – | 0 | 0 | 0 | 0 |

87h.7 **SMOD:** UART double baud rate control bit
0: Disable UART double baud rate
1: Enable UART double baud rate

| SFR **C6h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **XBAUD** | XBAUDS | BAUDRT | | | | | | |
| R/W | R/W | R/W | | | | | | |
| Reset | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |

C6h.7 **XBAUDS:** select UART extra baud rate generator
0: Baud rate uses Timer1/Timer2 overflow
1: Baud rate uses BAUDRT

C6h.6~0 **BAUDRT:** Extra baud rate

$F_{SYSCLK}$ denotes System clock frequency, the UART baud rate is calculated as below.

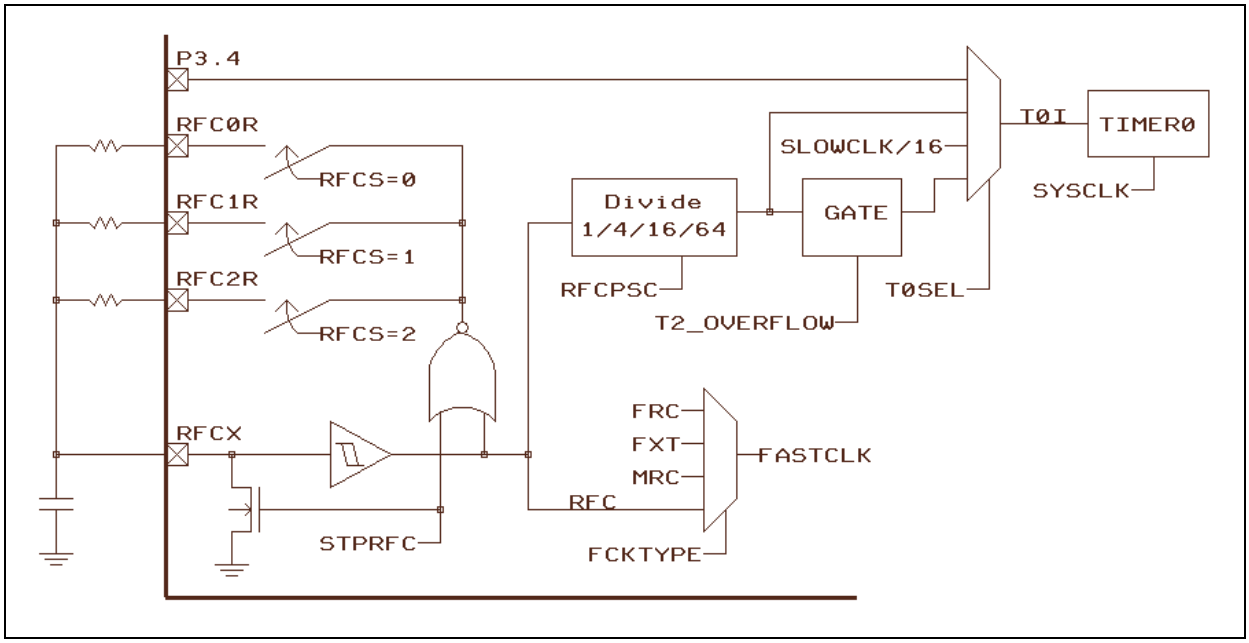
- **Mode 0:**
  Baud Rate = $F_{SYSCLK}$ / 2
- **Mode 1, 3:** if using Timer1 auto reload mode
  Baud Rate = (SMOD + 1) x $F_{SYSCLK}$ / (32 x 2 x (256 – TH1))
- **Mode 1, 3:** if using Timer2
  Baud Rate = Timer2 overflow rate / 16 = $F_{SYSCLK}$ / (32 x (65536 – RCP2H, RCP2L))
- **Mode 1, 3:** if using BAUDRT
  Baud Rate = $F_{SYSCLK}$ / (32 x BAUDRT)
- **Mode 2:**
  Baud Rate = (SMOD + 1) x $F_{SYSCLK}$ / 64

***Note6:*** also refer to Section 6 for more information about UART Interrupt enable and priority.

***Note8:*** also refer to Section 8 for more information about how Timer2 controls UART clock.

## 10. Resistance to Frequency Converter (RFC)

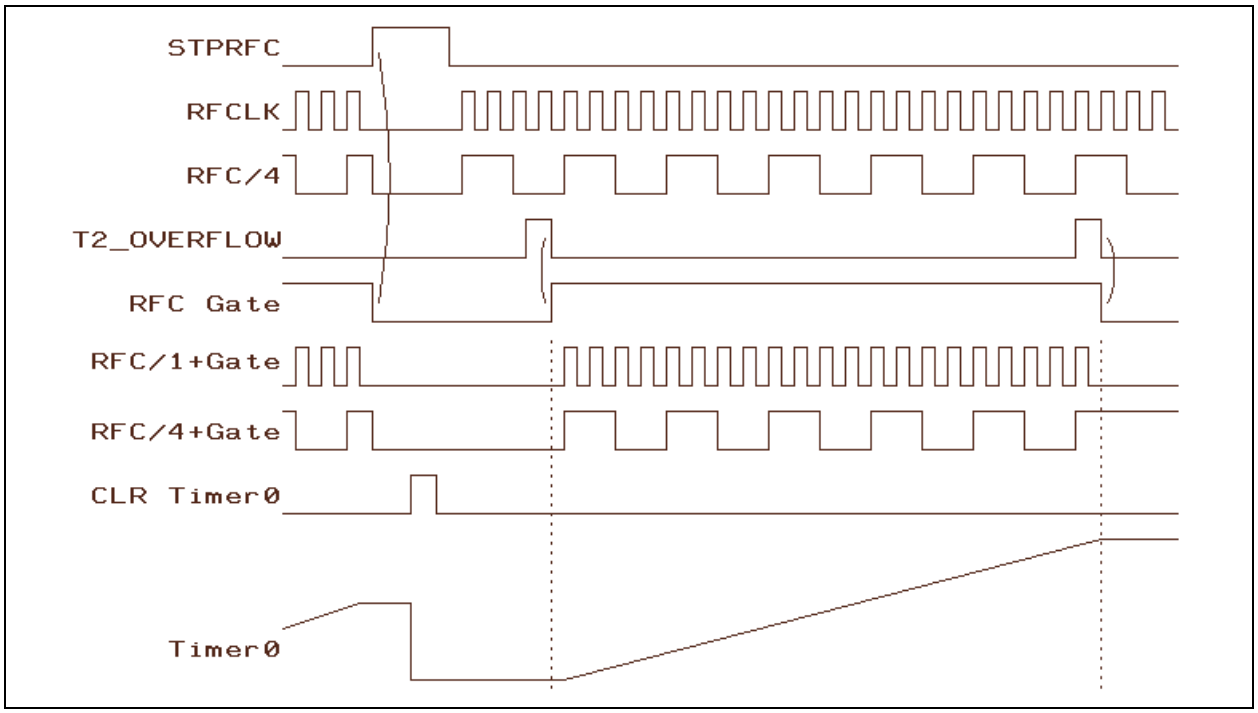
The RFC module can build the RC oscillation circuitry with RFCX pin and RFC0R, RFC1R or RFC2R pins. Only one RC oscillation circuitry is active at a time. There are 2 methods to measure the RFC clock frequency. One is to set the RFC as the Timer0 Counter mode input, the other one is to set RFC as the System clock. Since SXT is a precise timing source, user can derive the RFC frequency by comparing the Timer's count data which running by RFC and SXT.



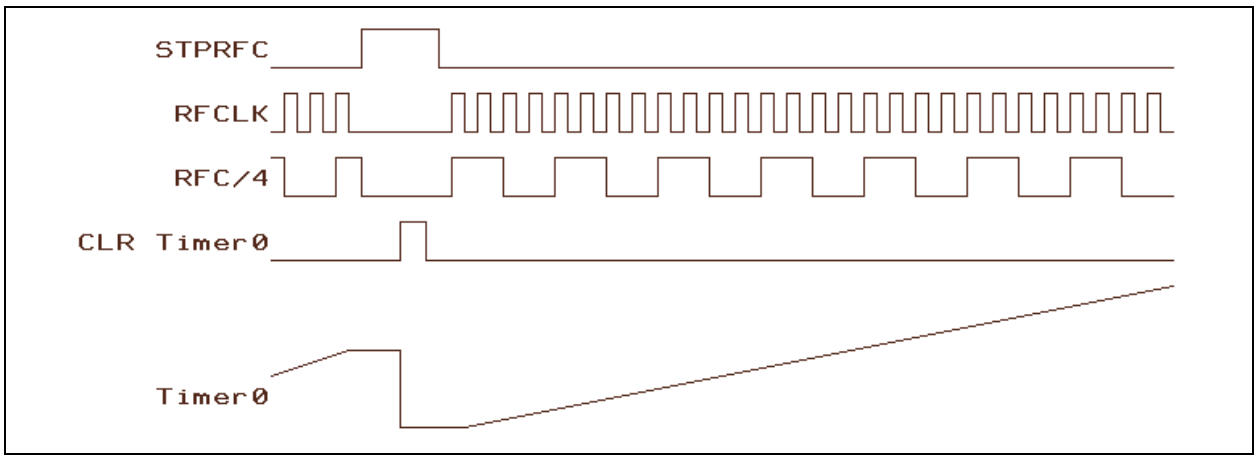

RFC Structure

The Timer0's event count input can be selected by T0SEL SFR. When T0SEL=3, the RFC clock is gated by Timer2's overflow period then go into the Timer0 for event counting. This function helps Timer0 to count the RFC clock with more accuracy by H/W automatically start and stop gating the RFC clock. The steps of this usage are described below.

1. Proper setting the PINMODE/RFCON SFR to setup the RFC oscillation circuitry.
2. CT0N=1 (Timer0 counter mode), CT2N=0 (Timer2 timer mode), T0SEL=3.
3. STPRFC=1, RFC gating is cleared and waiting for next Timer2 overflow to start
4. Clear Timer0, write TH2/TL2 with a data to accelerate Timer2 overflow (ex: FF00)
5. STPRFC=0, RFC starts, wait for next two Timer2 overflows.
6. The Timer0 counting the RFC clock only in between the two Timer2 overflows time slot.




RFC clock to Timer0, T0SEL=3




RFC clock to Timer0, T0SEL=1

| SFR **AFh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **RFCON** | P1RFC | | T0SEL | | RFCPSC | | RFCS | |
| R/W | R/W | | R/W | | R/W | | R/W | |
| Reset | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |

AFh.7~6 **P1RFC:** P1.7~P1.4 pin RFC mode control.
00: P1.7~P1.4 are not RFC pins
01: P1.7 and P1.6 are RFC pins, P1.5 and P1.4 are not RFC pins
10: P1.7~P1.5 are RFC pins, P1.4 is not RFC pin
11: P1.7~P1.4 are RFC pins

AFh.5~4 **T0SEL:** Timer0 Counter mode (CT0N=1) input select
00: P3.4 pin (8051 standard)
01: RFC clock divided by 1/4/16/64
10: Slow clock divided by 16 (SLOWCLK/16)
11: RFC clock divided by 1/4/16/64 gated by Timer2 overflow

AFh.3~2 **RFCPSC:** RFC clock divider to Timer0
00: divided by 64
01: divided by 16
10: divided by 4
11: divided by 1

AFh.1~0 **RFCS:** Select RFC convert channel.
00: RFC0R (P1.6)
01: RFC1R (P1.5)
10: RFC2R (P1.4)

| SFR **F8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX1** | – | – | TKSOC | ADSOC | CLRWDT | CLRTM3 | STPRFC | DPSEL |
| R/W | – | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | – | – | 0 | 0 | 0 | 0 | 0 | 0 |

F8h.1 **STPRFC:** Set 1 to stop RFC clock oscillating

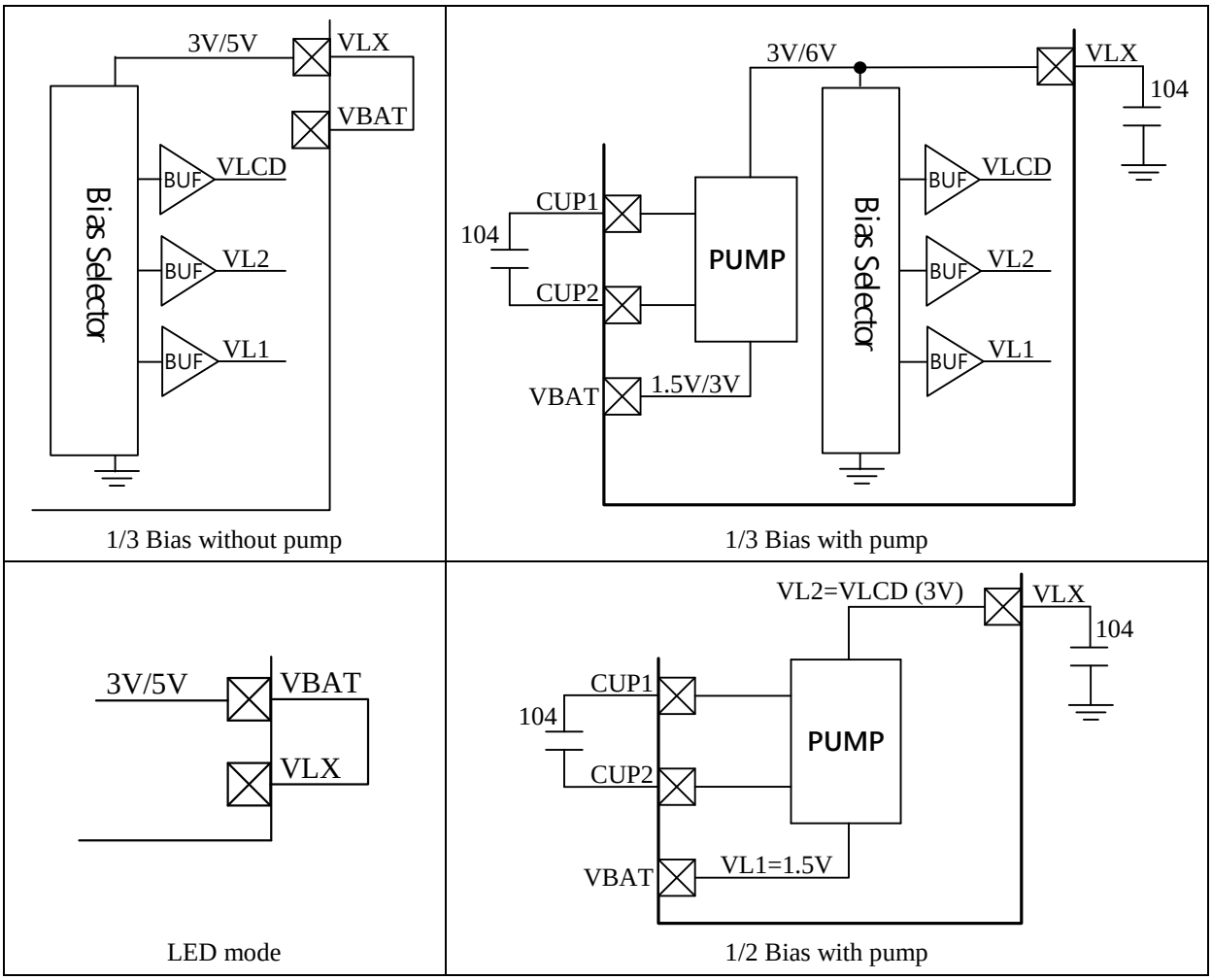
| SFR **D8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CLKCON** | FCKTYPE | | SELFCK | SCKTYPE | STPFSUB | STPSCK | CLKPSC | |
| R/W | R/W | | R/W | R/W | R/W | R/W | R/W | |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

D8h.7~6 **FCKTYPE:** Fast clock type select, These bits can be changed only in Slow mode (SELFCK=0)
00: Fast clock is FRC
10: Fast clock is MRC
11: Fast clock is RFC, S/W must setup RFC oscillating circuitry before this setting.

## 11. LCD / LED Driver

The **LCD Driver** is capable of driving the LCD panel with 3~8 Commons and maximum 45 Segments. The module can operate with or without pump. If LCDPUMP=0, no external component is required, VBAT and VLX pin should be tied together. If LCDPUMP=1, two 0.1uF capacitor should be placed at CUP1, CUP2 and VLX pin as the diagram below. In 1/3 Bias mode, the VCLD voltage has 16 brightness levels, which is controlled by LCDBV SFR. The VL1 and VL2 voltage level are divided from VLCD. So VL1=VLCD/3, VL2=VLCD*2/3 and VL3=VLCD. In 1/2 Bias mode, VL1=$V_{BAT}$=VLCD/2 and VL2=VLCD.

The VL1, VL2 and VLCD (VL3) LCD 1/3 bias voltage are generated by tenx's unique tiny current LCD Buffer technology, which can drive very big LCD panel without waveform distortion, but the Driver itself only consumes small current (1.6uA @$V_{BAT}$=3V). This technique also reduce external component and pin connection for package/PCB cost reduction.




***Note:*** User must force LCDBV=1111b at least 100ms for 1/3 Bias pump mode start-up.

Table below illustrates VLCD and VL1 voltage for 1/3 Bias mode, with or without pump. User can detect the $V_{BAT}$ voltage level by LBD, and accordingly set the LCDBV for VLCD voltage (brightness level).

| BIAS2=0 | LCDPUMP=0 | | | LCDPUMP=1 | | |
|---|---|---|---|---|---|---|
| LCDBV | VLCD | VL1 (V) | | VLCD | VL1 (V) | |
| | | $V_{BAT}$=3V | $V_{BAT}$=5V | | $V_{BAT}$=1.5V | $V_{BAT}$=3V |
| 0 | $V_{BAT}$ * 24/40 | 0.600 | 1.000 | $V_{BAT}$ * 48/40 | 0.600 | 1.200 |
| 1 | $V_{BAT}$ * 24/38 | 0.632 | 1.053 | $V_{BAT}$ * 48/38 | 0.632 | 1.263 |
| 2 | $V_{BAT}$ * 24/37 | 0.649 | 1.081 | $V_{BAT}$ * 48/37 | 0.649 | 1.297 |
| 3 | $V_{BAT}$ * 24/36 | 0.667 | 1.111 | $V_{BAT}$ * 48/36 | 0.667 | 1.333 |
| 4 | $V_{BAT}$ * 24/35 | 0.686 | 1.143 | $V_{BAT}$ * 48/35 | 0.686 | 1.371 |
| 5 | $V_{BAT}$ * 24/34 | 0.706 | 1.176 | $V_{BAT}$ * 48/34 | 0.706 | 1.412 |
| 6 | $V_{BAT}$ * 24/33 | 0.727 | 1.212 | $V_{BAT}$ * 48/33 | 0.727 | 1.455 |
| 7 | $V_{BAT}$ * 24/32 | 0.750 | 1.250 | $V_{BAT}$ * 48/32 | 0.750 | 1.500 |
| 8 | $V_{BAT}$ * 24/31 | 0.774 | 1.290 | $V_{BAT}$ * 48/31 | 0.774 | 1.548 |
| 9 | $V_{BAT}$ * 24/30 | 0.800 | 1.333 | $V_{BAT}$ * 48/30 | 0.800 | 1.600 |
| 10 | $V_{BAT}$ * 24/29 | 0.828 | 1.379 | $V_{BAT}$ * 48/29 | 0.828 | 1.655 |
| 11 | $V_{BAT}$ * 24/28 | 0.857 | 1.429 | $V_{BAT}$ * 48/28 | 0.857 | 1.714 |
| 12 | $V_{BAT}$ * 24/27 | 0.889 | 1.481 | $V_{BAT}$ * 48/27 | 0.889 | 1.778 |
| 13 | $V_{BAT}$ * 24/26 | 0.923 | 1.538 | $V_{BAT}$ * 48/26 | 0.923 | 1.846 |
| 14 | $V_{BAT}$ * 24/25 | 0.960 | 1.600 | $V_{BAT}$ * 48/25 | 0.960 | 1.920 |
| 15 | $V_{BAT}$ * 24/24 | 1.000 | 1.667 | $V_{BAT}$ * 48/24 | 1.000 | 2.000 |

LCD Brightness level setting by LCDBV

The LCD clock can be driven by Slow clock or Fast clock. If SXT is the clock source, the LCD frame rate ranges from 43 Hz to 98 Hz according to LCD Duty and LCDFRM. If the LCD clock comes from other clock source, the Frame rate varies proportionally to the clock frequency. The frame rate of LED mode is double of LCD mode in the same setting. The LED and LCD module share the same LCD RAM and several common SFR.

| LCD Frame Rate (Hz) | LCDFMR (SFR B1h.1~0) | | | |
|---|---|---|---|---|
| | 00 | 01 | 10 | 11 |
| 1/3 Duty | 57 | 68 | 85 | 98 |
| 1/4 Duty | 43 | 51 | 64 | 73 |
| 1/5 Duty | 46 | 59 | 68 | 82 |
| 1/6 Duty | 57 | 68 | 85 | 98 |
| 1/7 Duty | 49 | 59 | 73 | 84 |
| 1/8 Duty | 43 | 51 | 64 | 73 |

LCD Frame Rate when LCDCLK=SXT

In **LED Normal mode**, the chip provides maximum 8COM x 36SEG driver. For LED application, the COM pin is active low with dead time control and the Segment pin is active high. Each COM pin can sink 70mA current when $V_{BAT}$=5V. The chip support All LED Segment mode for DC output. In such application, user set LCDUTY=7 and fill the LCDRAM SEG bit with same data. For example, write 0xF001 with 0x00 for SEG1's low level output; write 0xF009 with 0xFF for SEG9's high level output.

| SFR **B1h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **LCON** | DSPON | LCDUTY | | | LCDCLK | | LCDFMR | |
| R/W | R/W | R/W | | | R/W | | R/W | |
| Reset | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |

B1h.7 **DSPON:** LCD / LED display enable control
0: LCD / LED disable
1: LCD / LED enable

B1h.6~4 **LCDUTY:** LCD / LED duty control
000: 1/3 duty
001: 1/4 duty
010: 1/5 duty
011: 1/6 duty
100: 1/7 duty
101: 1/8 duty
111: All LED Segment DC output mode, SEG0~2 replace the COM0~2 output.

B1h.3~2 **LCDCLK:** LCD / LED clock source
00: SLOWCLK
01: SLOWCLK/2
10: FASTCLK/128
11: FASTCLK/256

B1h.1~0 **LCDFMR:** LCD / LED Frame rate control. If SXT is the LCD clock source, the accurate LCD frame rate is listed in the table above. If others clock is the LCD clock source, the Frame rate can be derived by the clock frequency proportional to 32KHz.

| SFR **B2h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **LCON2** | LEDMOD | LCDPUMP | BIAS2 | LEDBLC | LCDBV | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

B2h.7 **LEDMOD:** LCD / LED mode select for COM and SEG pins
0: LCD mode
1: LED mode

B2h.6 **LCDPUMP:** LCD pump / LED DMX mode select
0: LCD no pump / LED Normal mode
1: LCD Pump / LED DMX mode

B2h.5 **BIAS2:** LCD Bias select
0: 1/3 Bias
1=1/2 Bias

B2h.4 **LEDBLC:** LED brightness balance
0: LED Normal Brightness
1: LED Balanced Brightness

B2h.3~0 **LCDBV:** LCD Brightness, VLCD Voltage level control
0000: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/40
0001: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/38
0010: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/37
0011: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/36
0100: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/35
0101: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/34
0110: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/33
0111: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/32
1000: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/31
1001: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/30
1010: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/29
1011: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/28
1100: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/27
1101: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/26
1110: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/25
1111: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/24

***Note:*** User must force LCDBV=1111b at least 100ms for 1/3 Bias pump mode start-up.

| SFR **92h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PINMODE** | – | P2HSEG | | P2LSEG | | P0SEG | | |
| R/W | – | R/W | | R/W | | R/W | | |
| Reset | – | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

92h.6~5 **P2HSEG:** P2.4~P2.6 pin LCD/LED mode control.
00: P2.4~P2.6 are I/O pins
01: P2.4 and P2.5 are I/O pins, P2.6 is LCD/LED Segment pin
10: P2.4 is I/O pin, P2.5 and P2.6 are LCD/LED Segment pins
11: P2.4~P2.6 are LCD/LED Segment pins

92h.4~3 **P2LSEG:** P2.1~P2.3 pin LCD/LED mode control.
00: P2.1~P2.3 are I/O pins
01: P2.1 and P2.2 are I/O pins, P2.3 is LCD/LED Segment pin
10: P2.1 is I/O pin, P2.2 and P2.3 are LCD/LED Segment pins
11: P2.1~P2.3 are LCD/LED Segment pins

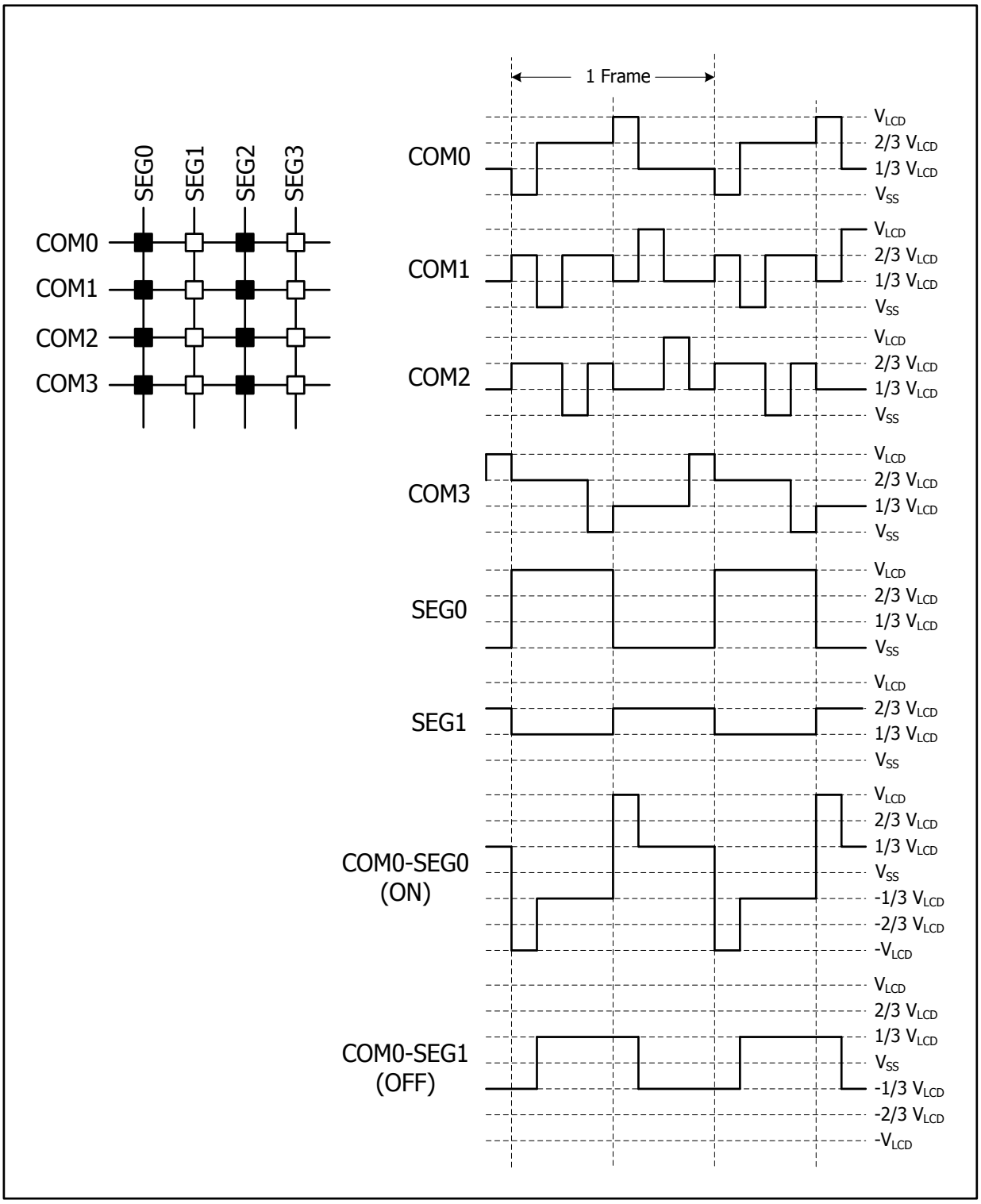
92h.2~0 **P0SEG:** Port0 LCD/LED mode control.
000: P0.0~P0.6 are I/O pins
001: P0.0~P0.5 are I/O pins, P0.6 is LCD/LED Segment pin
010: P0.0~P0.4 are I/O pins, P0.5~P0.6 are LCD/LED Segment pins
011: P0.0~P0.3 are I/O pins, P0.4~P0.6 are LCD/LED Segment pins
100: P0.0~P0.2 are I/O pins, P0.3~P0.6 are LCD/LED Segment pins
101: P0.0~P0.1 are I/O pins, P0.2~P0.6 are LCD/LED Segment pins
110: P0.0 is I/O pin, P0.1~P0.6 are LCD/LED Segment pins
111: P0.0~P0.6 are LCD/LED Segment pins

| SFR **AEh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TKCON2** | P4SEG | | | | TKREFC | | | |
| R/W | R/W | | | | R/W | | | |
| Reset | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

AEh.7~4 **P4SEG:** Port4 LCD/LED mode control.
0000: P4.0~P4.7 are I/O pins
0001: P4.0~P4.6 are I/O pins, P4.7 is LCD/LED Segment pin
0010: P4.0~P4.5 are I/O pins, P4.6~P4.7 are LCD/LED Segment pins
0011: P4.0~P4.4 are I/O pins, P4.5~P4.7 are LCD/LED Segment pins
0100: P4.0~P4.3 are I/O pins, P4.4~P4.7 are LCD/LED Segment pins
0101: P4.0~P4.2 are I/O pins, P4.3~P4.7 are LCD/LED Segment pins
0110: P4.0~P4.1 are I/O pins, P4.2~P4.7 are LCD/LED Segment pins
0111: P4.0 is I/O pin, P4.1~P4.7 are LCD/LED Segment pins
1000: P4.0~P4.7 are LCD/LED Segment pins

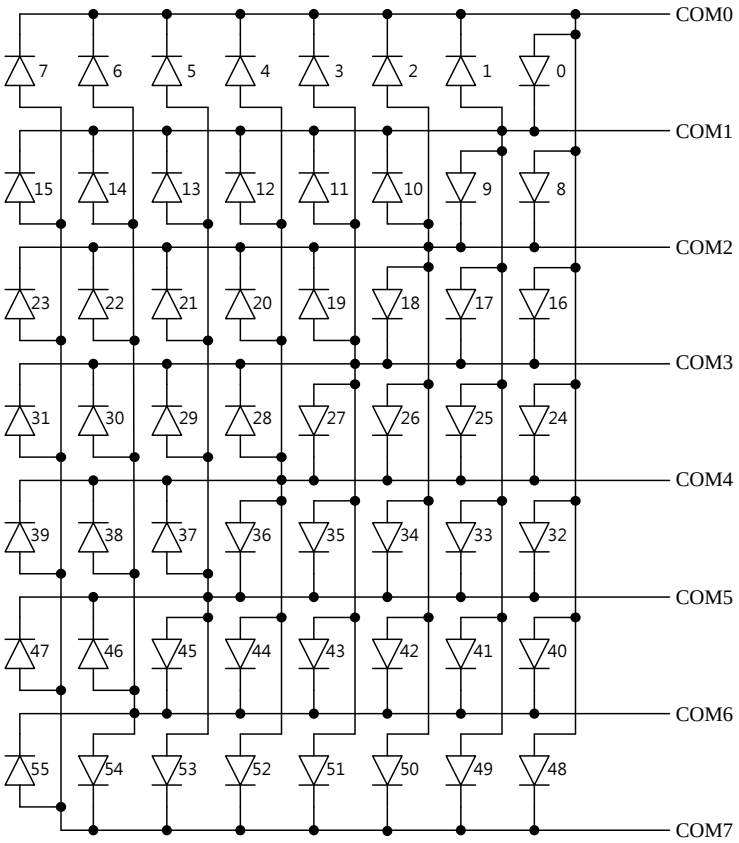
| | COM7 | COM6 | COM5 | COM4 | COM3 | COM2 | COM1 | COM0 |
|---|---|---|---|---|---|---|---|---|
| Adr | bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
| F000 | – | – | – | – | – | SEG0 | SEG0 | SEG0 |
| F001 | – | – | – | – | – | SEG1 | SEG1 | SEG1 |
| F002 | – | – | – | – | – | SEG2 | SEG2 | SEG2 |
| F003 | – | – | – | – | – | SEG3 | SEG3 | SEG3 |
| F004 | – | – | – | – | SEG4 | SEG4 | SEG4 | SEG4 |
| F005 | – | – | – | SEG5 | SEG5 | SEG5 | SEG5 | SEG5 |
| F006 | – | – | SEG6 | SEG6 | SEG6 | SEG6 | SEG6 | SEG6 |
| F007 | – | SEG7 | SEG7 | SEG7 | SEG7 | SEG7 | SEG7 | SEG7 |
| F008 | SEG8 | SEG8 | SEG8 | SEG8 | SEG8 | SEG8 | SEG8 | SEG8 |
| F009 | SEG9 | SEG9 | SEG9 | SEG9 | SEG9 | SEG9 | SEG9 | SEG9 |
| F00A~F02D | SEG10~SEG45 | | | | | | | |
| F02E | SEG46 | SEG46 | SEG46 | SEG46 | SEG46 | SEG46 | SEG46 | SEG46 |
| F02F | SEG47 | SEG47 | SEG47 | SEG47 | SEG47 | SEG47 | SEG47 | SEG47 |

LCD / LED Normal mode RAM Mapping (8051's External Data Memory space)




LCD Waveform, 1/3 Bias, 1/4 Duty, (VLCD=3*VL1)

The chip also provides **LED DMX mode** (Dot Matrix mode) using COM0~COM7 pins, up to 7 * 8 = 56 LED points can be configured to drive. This mode is enabled by set LCDPUMP=1. The corresponding LED dot matrix position is marked in the figure below. The relationship between LRAM's bit and LED lighting map is also shown as below table.



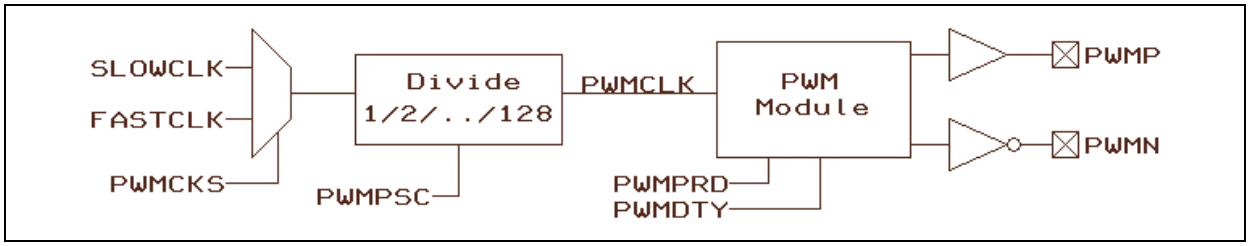

LED 7*8 DMX

| Adr | bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|---|---|---|---|---|---|---|---|---|
| F000 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | – |
| F001 | 15 | 14 | 13 | 12 | 11 | 10 | – | 0 |
| F002 | 23 | 22 | 21 | 20 | 19 | – | 9 | 8 |
| F003 | 31 | 30 | 29 | 28 | – | 18 | 17 | 16 |
| F004 | 39 | 38 | 37 | – | 27 | 26 | 25 | 24 |
| F005 | 47 | 46 | – | 36 | 35 | 34 | 33 | 32 |
| F006 | 55 | – | 45 | 44 | 43 | 42 | 41 | 40 |
| F007 | – | 54 | 53 | 52 | 51 | 50 | 49 | 48 |

LED DMX mode bit mapping

## 12. PWM

The chip has 6 channel CMOS output PWMs. Each PWM can select Fast clock or Slow clock as its clock source, with divided by 1~128 prescaler. The PWM period is adjustable by PWMnPRD SFR and its 256 duty cycle controlled by PWMnDTY SFR. The PWM0P and PWM0N are positive and negative pairs, which support pump voltage drive. The PWM1 can sink maximum 300mA for IR application. The PWM5 can generate interrupt and wake-up CPU from Idle/Halt mode.




PWM Structure

| SFR**CEh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWMCON** | PWM1CKS | PWM1PSC | | | PWM0CKS | PWM0PSC | | |
| R/W | R/W | R/W | | | R/W | R/W | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

CEh.7 **PWM1CKS:** PWM1~4 clock source select
0: Slow clock
1: Fast clock

CEh.6~4 **PWM1PSC:** PWM1~4 clock prescaler
000: PWM clock is Slow/Fast clock divided by 128
001: PWM clock is Slow/Fast clock divided by 64
010: PWM clock is Slow/Fast clock divided by 32
011: PWM clock is Slow/Fast clock divided by 16
100: PWM clock is Slow/Fast clock divided by 8
101: PWM clock is Slow/Fast clock divided by 4
110: PWM clock is Slow/Fast clock divided by 2
111: PWM clock is Slow/Fast clock divided by 1

CEh.3 **PWM0CKS:** PWM0 clock source select
0: Slow clock
1: Fast clock

CEh.2~0 **PWM0PSC:** PWM0 clock prescaler
000: PWM clock is Slow/Fast clock divided by 128
001: PWM clock is Slow/Fast clock divided by 64
010: PWM clock is Slow/Fast clock divided by 32
011: PWM clock is Slow/Fast clock divided by 16
100: PWM clock is Slow/Fast clock divided by 8
101: PWM clock is Slow/Fast clock divided by 4
110: PWM clock is Slow/Fast clock divided by 2
111: PWM clock is Slow/Fast clock divided by 1

| SFR**CFh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWMCON2** | PWRSAV2 | PWM5CLR | PWM0VX2 | PWM1SNK | PWM5CKS | PWM5PSC | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | | |
| Reset | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

CFh.6 **PWM5CLR:** Clear PWM5 Period counter
0: PWM5 run
1: PWM5 clear and hold

CFh.5 **PWM0VX2:** PWM0P / PWM0N pump drive select
0: PWM0P / PWM0N normal drive
1: PWM0P / PWM0N pump drive (high level = $V_{BAT}$ * 2, need LCD pump)

CFh.4 **PWM1SNK:** PWM1 high sink select
0: PWM1 normal sink
1: PWM1 high sink (300mA)

CFh.3 **PWM5CKS:** PWM5 clock source select
0: Slow clock
1: Fast clock

CFh.2~0 **PWM5PSC:** PWM5 clock prescaler
000: PWM clock is Slow/Fast clock divided by 128
001: PWM clock is Slow/Fast clock divided by 64
010: PWM clock is Slow/Fast clock divided by 32
011: PWM clock is Slow/Fast clock divided by 16
100: PWM clock is Slow/Fast clock divided by 8
101: PWM clock is Slow/Fast clock divided by 4
110: PWM clock is Slow/Fast clock divided by 2
111: PWM clock is Slow/Fast clock divided by 1

| SFR **9Ah** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM0PRD** | PWM0PRD | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

9Ah.7~0 **PWM0PRD:** PWM0 Period, FFh=256 PWMCLK, 7Fh=128 PWMCLK

| SFR **9Bh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM0DTY** | PWM0DTY | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

9Bh.7~0 **PWM0DTY:** PWM0 Duty, 0=0 PWMCLK, 80h=128 PWMCLK, FFh=255 PWMCLK

| SFR **9Ch** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM1PRD** | PWM1PRD | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

9Ch.7~0 **PWM1PRD:** PWM1~4 Period, FFh=256 PWMCLK, 7Fh=128 PWMCLK

| SFR **9Dh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM1DTY** | PWM1DTY | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

9Dh.7~0 **PWM1DTY:** PWM1 Duty, 0=0 PWMCLK, 80h=128 PWMCLK, FFh=255 PWMCLK

| SFR **9Eh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM2DTY** | PWM2DTY | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

9Eh.7~0 **PWM2DTY:** PWM2 Duty, 0=0 PWMCLK, 80h=128 PWMCLK, FFh=255 PWMCLK

| SFR **9Fh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM3DTY** | PWM3DTY | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

9Fh.7~0 **PWM3DTY:** PWM3 Duty, 0=0 PWMCLK, 80h=128 PWMCLK, FFh=255 PWMCLK

| SFR **D4h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM4DTY** | PWM4DTY | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

D4h.7~0 **PWM4DTY:** PWM4 Duty, 0=0 PWMCLK, 80h=128 PWMCLK, FFh=255 PWMCLK

| SFR **D5h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM5DTY** | PWM5DTY | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

D5h.7~0 **PWM5DTY:** PWM5 Duty, 0=0 PWMCLK, 80h=128 PWMCLK, FFh=255 PWMCLK

| SFR **D6h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWM5PRD** | PWM5PRD | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

D6h.7~0 **PWM5PRD:** PWM5 Period, FFh=256 PWMCLK, 7Fh=128 PWMCLK

| SFR **D7h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **PWMOE** | PWM5OE | PWM4BOE | PWM4AOE | PWM3OE | PWM2OE | PWM1OE | PWM0POE | PWM0NOE |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

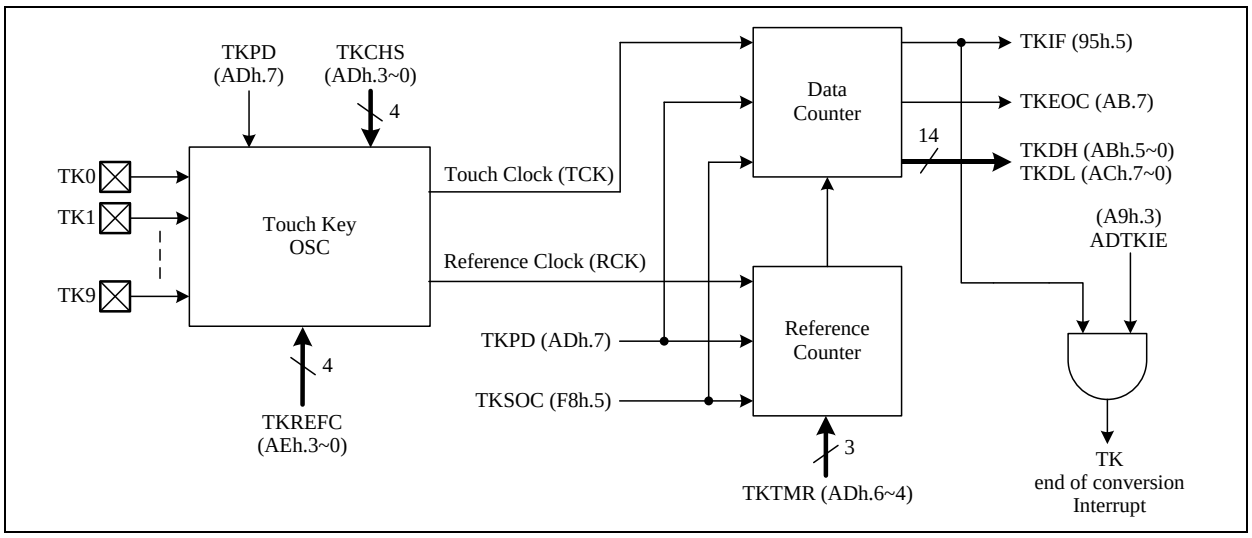
D7h.7 **PWM5OE:** PWM5 output to P1.3
D7h.6 **PWM4BOE:** PWM4 output to P3.6
D7h.5 **PWM4AOE:** PWM4 output to P1.1
D7h.4 **PWM3OE:** PWM3 output to P1.0
D7h.3 **PWM2OE:** PWM2 output to P3.4
D7h.2 **PWM1OE:** PWM1 output to P3.5
D7h.1 **PWM0POE:** PWM0P output to P3.7
D7h.0 **PWM0NOE:** PWM0N output to P3.6

| SFR **95h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTFLG** | LBDIF | – | TKIF | ADIF | PWMIF | IE2 | PNCIF | TF3 |
| R/W | R/W | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |

95h.3 **PWMIF:** PWM5 period counter full interrupt flag.
Set by H/W when PWM5 period counter full. Cleared automatically when the program performs the interrupt service routine. S/W can write F7h to INTFLG to clear this bit. (***Note2***)

## 13. Touch Key

The Touch Key module offers an easy, simple and reliable method to implement finger touch detection. The chip support 10 channels touch key detection.




Touch Key Structure

While a TK pin is under scanning, the module automatically disables the pin's CMOS output path. Therefore, user can set the scan TK pin's mode as Mode2. After TK scan, user must set TKPD=1 to disconnect the TK module and IO pins.

| TK0~TK9 | P1.n / P3.n I/O pin setting |
|---|---|
| Pin is Touch Key, Idling | Drive Low (Mode2) |
| Pin is Touch Key, Scanning | |

To start a TK scan, user assigns TKPD=0, then set the TKSOC bit to start touch key conversion. After the end of conversion, H/W clears the TKSOC bit and set the TKIF interrupt flag. However, if the SYSCLK is too slow, H/W might fail to clear TKSOC due to clock sampling rate. TKEOC=0 means conversion is in process. TKEOC=1 means the conversion is finish, and the touch key counting result is stored into the 14 bits TK Data Counter TKDH and TKDL. The larger TK pin capacitance is, the smaller TK Data counter is.

The Touch Key unit has an internal built-in reference capacitor to simulate the KEY behavior. Set TKCHS=15 and start a scan can get the TK Data count of this capacitor. Since the internal capacitor would not be affected by water or mobile phone, it is useful for comparing the environment background noise.

| SFR **F8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX1** | – | – | TKSOC | ADSOC | CLRWDT | CLRTM3 | STPRFC | DPSEL |
| R/W | – | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | – | – | 0 | 0 | 0 | 0 | 0 | 0 |

F8h.5 **TKSOC:** Rising edge of this bit will trigger a Touch Key conversion. Basically, this bit is automatically cleared by H/W after end of conversion. However, if the SYSCLK is too slow, H/W might fail to clear TKSOC due to clock sampling rate issue.

| SFR **ADh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TKCON** | TKPD | TKTMR | | | TKCHS | | | |
| R/W | R/W | R/W | | | R/W | | | |
| Reset | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |

ADh.7 **TKPD:** Touch Key Power Down
0: Touch Key enable
1: Touch Key disable

ADh.6~4 **TKTMR:** Touch Key Conversion Time
000: Conversion time shortest
…
111: Conversion time longest

ADh.3~0 **TKCHS:** Touch Key Channel Select
0000: TK0 (P1.0)
0001: TK1 (P1.1)
0010: TK2 (P1.2)
0011: TK3 (P1.3)
0100: TK4 (P1.4)
0101: TK5 (P1.5)
0110: TK6 (P1.6)
0111: TK7 (P1.7)
1000: TK8 (P3.4)
1001: TK9 (P3.5)
1111: TK15 (Internal reference)

| SFR **AEh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TKCON2** | P4SEG | | | | TKREFC | | | |
| R/W | R/W | | | | R/W | | | |
| Reset | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

AEh.3~0 **TKREFC:** Touch Key reference clock capacitor select
0000: Smallest (conversion time shortest)
…
1111: Biggest (conversion time longest)

| SFR **ABh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TKDH** | TKEOC | – | TKDTH | | | | | |
| R/W | R | – | R | | | | | |
| Reset | – | – | – | – | – | – | – | – |

ABh.7 **TKEOC:** Touch Key End of Conversion, 1=EOC. TKEOC may have 3uS delay after TKSOC=1.

ABh.5~0 **TKDTH:** Touch Key Counter Data 13~8

| SFR **ACh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **TKDL** | TKDL | | | | | | | |
| R/W | R | | | | | | | |
| Reset | – | – | – | – | – | – | – | – |

ACh.7~0 **TKDL:** Touch Key Counter Data 7~0

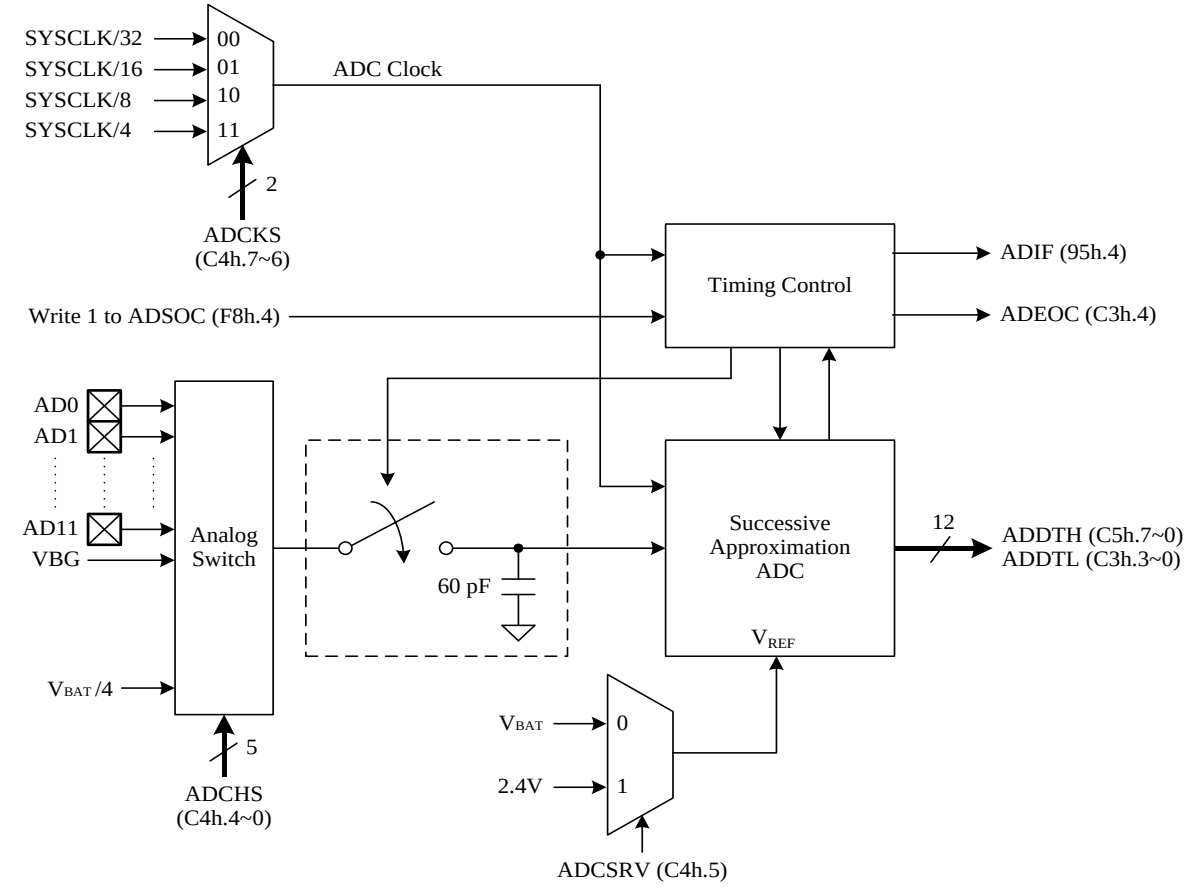
| SFR **95h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTFLG** | LBDIF | – | TKIF | ADIF | PWMIF | IE2 | PNCIF | TF3 |
| R/W | R/W | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |

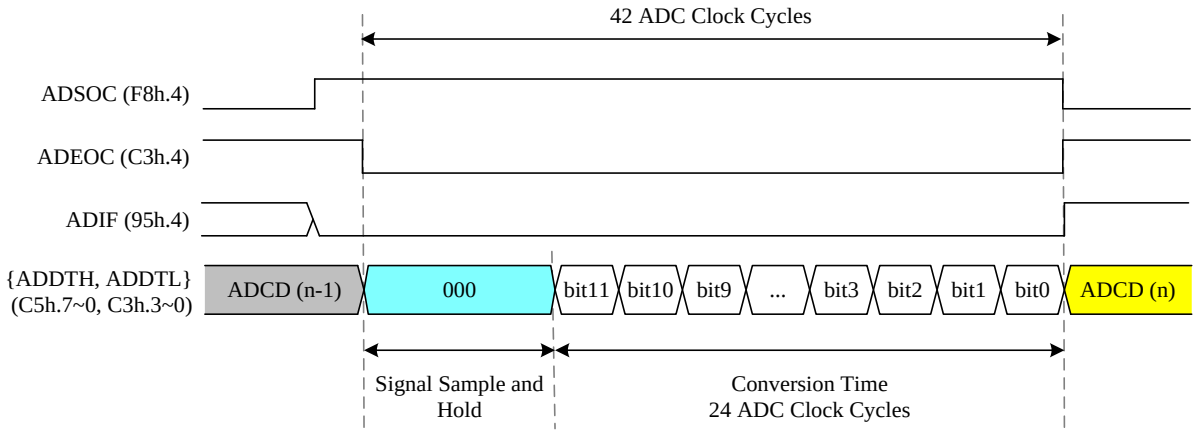
95h.5 **TKIF:** Touch Key Interrupt Flag
Set by H/W when TK end of conversion. S/W can write DFh to INTFLG to clear this bit.

***Note6:*** also refer to Section 6 for more information about Touch Key Interrupt enable and priority.

## 14. 12-bit SAR ADC

The chip offers a 12-bit ADC consisting of analog input multiplexer, control register, clock generator, 12-bit successive approximation register, and output data register. To use the ADC, set the ADCKS bit first to choose a proper ADC clock frequency, which must be less than 2 MHz. Then, launch the ADC conversion by setting the ADSOC bit, and H/W will automatic clear it at the end of the conversion. After the end of the conversion, H/W will set the ADIF bit and generate an interrupt if an ADC interrupt is enabled. The ADIF bit can be cleared by writing 0 to this bit or 1 to the ADSOC bit. The $V_{REF}$ of the ADC can be selected $V_{BAT}$ or 2.4V.



| SFR **F8h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **AUX1** | – | – | TKSOC | ADSOC | CLRWDT | CLRTM3 | STPRFC | DPSEL |
| R/W | – | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | – | – | 0 | 0 | 0 | 0 | 0 | 0 |

F8h.4 **ADSOC:** Rising edge of this bit will trigger an ADC conversion. This bit is automatically cleared by H/W after end of conversion. S/W can also write 0 to clear this flag.

| SFR **C4h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **ADCON** | ADCKS | | ADCSRV | ADCHS | | | | |
| R/W | R/W | | R/W | R/W | | | | |
| Reset | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |

C4h.7~6 **ADCKS:** ADC clock rate select
00: $F_{SYSCLK}$/32
01: $F_{SYSCLK}$/16
10: $F_{SYSCLK}$/8
11: $F_{SYSCLK}$/4

C4h.5 **ADCSRV:** ADC reference voltage select
0: $V_{BAT}$
1: 2.4V

C4h.4~0 **ADCHS:** ADC channel select
00000: AD0 (P1.0)
00001: AD1 (P1.1)
00010: AD2 (P1.2)
00011: AD3 (P1.3)
00100: AD4 (P1.4)
00101: AD5 (P1.5)
00110: AD6 (P1.6)
00111: AD7 (P1.7)
01000: AD8 (P3.4)
01001: AD9 (P3.5)
01010: AD10 (P2.0)
01011: AD11 (P0.7)
01100: VBG (ADCSRV=0)
10111: $V_{BAT}$ /4

| SFR **C5h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **ADDTH** | ADDTH | | | | | | | |
| R/W | R | | | | | | | |
| Reset | – | – | – | – | – | – | – | – |

C5h.7~0 **ADDTH:** ADC data bit 11~4

| SFR **C3h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **ADDTL** | LBDO | – | – | ADEOC | ADCDTL | | | |
| R/W | R | – | – | R | R | | | |
| Reset | – | – | – | – | – | – | – | – |

C3h.4 **ADEOC:** ADC end of conversion. 1=end

C3h.3~0 **ADCDTL:** ADC data bit 3~0

| SFR **95h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **INTFLG** | LBDIF | – | TKIF | ADIF | PWMIF | IE2 | PNCIF | TF3 |
| R/W | R/W | – | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | 0 | – | 0 | 0 | 0 | 0 | 0 | 0 |

95h.4 **ADIF:** ADC Interrupt Flag
Set by H/W when ADC end of conversion. S/W can write EFh to INTFLG to clear this bit.

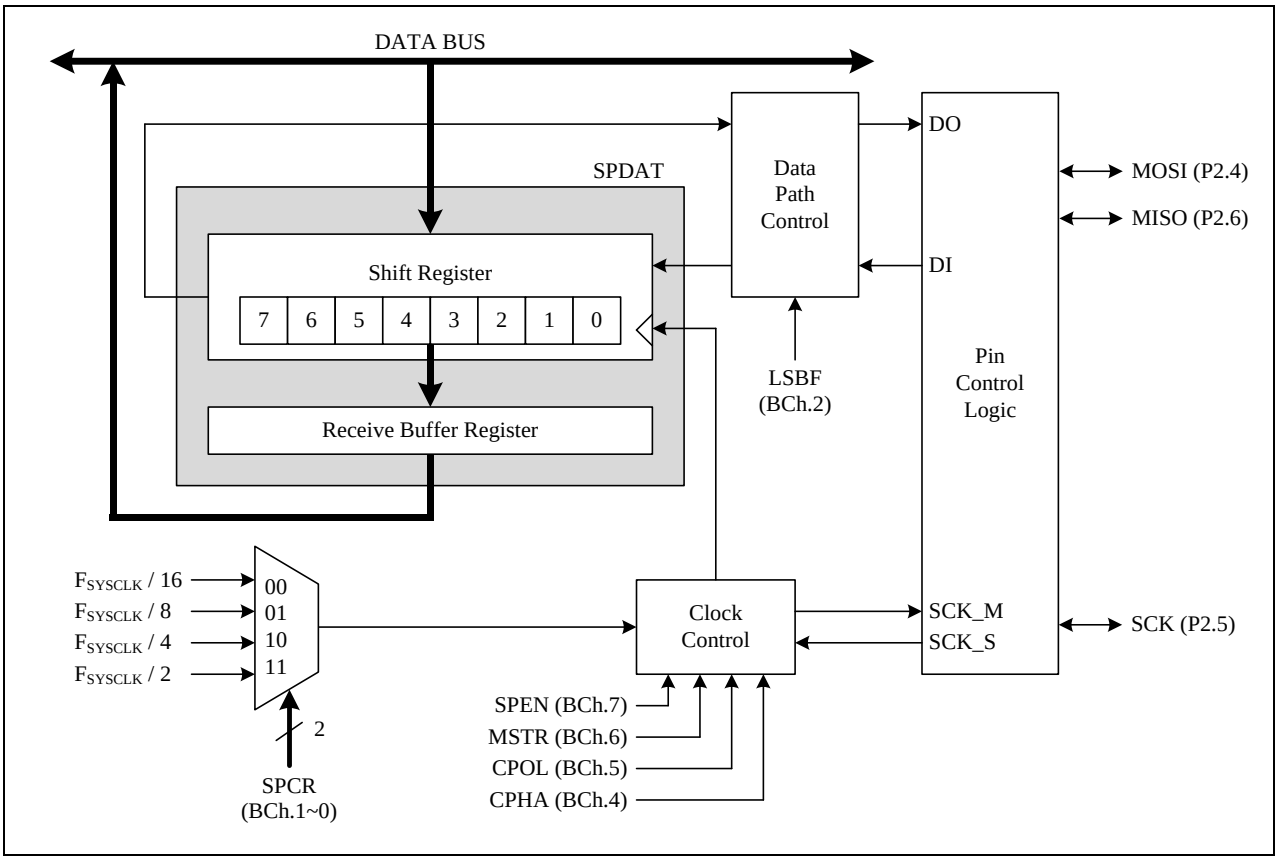
***Note:*** FW must force ADCSRV=0 before enter Stop/Halt/Idle mode.

## 15. Serial Peripheral Interface (SPI)

The SPI module is capable of full-duplex, synchronous, serial communication between the chip and peripheral devices. The peripheral devices can be other MCUs, A/D converter, sensors, or MTP memory, etc. The SPI runs at a baud rate up to the system clock divided by two. Firmware can read the status flags, or the operation can be interrupt driven.

The features of the SPI module include:

- Master or Slave mode operation
- 3-wire mode operation
- Full-duplex operation
- Programmable transmit bit rate
- Single Buffer receive
- Serial clock phase and polarity options
- MSB-first or LSB-first shifting selectable




SPI System Block Diagram

The MOSI (P2.4) signal is an output when SPI is operating in Master mode and an input when SPI is operating in Slave mode. The MISO (P2.6) signal is an input when SPI is operating in Master mode and an output when SPI is operating in Slave mode. Data is transferred MSB or LSB first by setting the LSBF bit. The SCK (P2.5) signal is an output from a Master device and an input to Slave devices. It is used to synchronize the data on the MOSI and MISO lines of Master and Slave. SPI generates the signal with eight programmable clock rates in Master mode.

### Master Mode

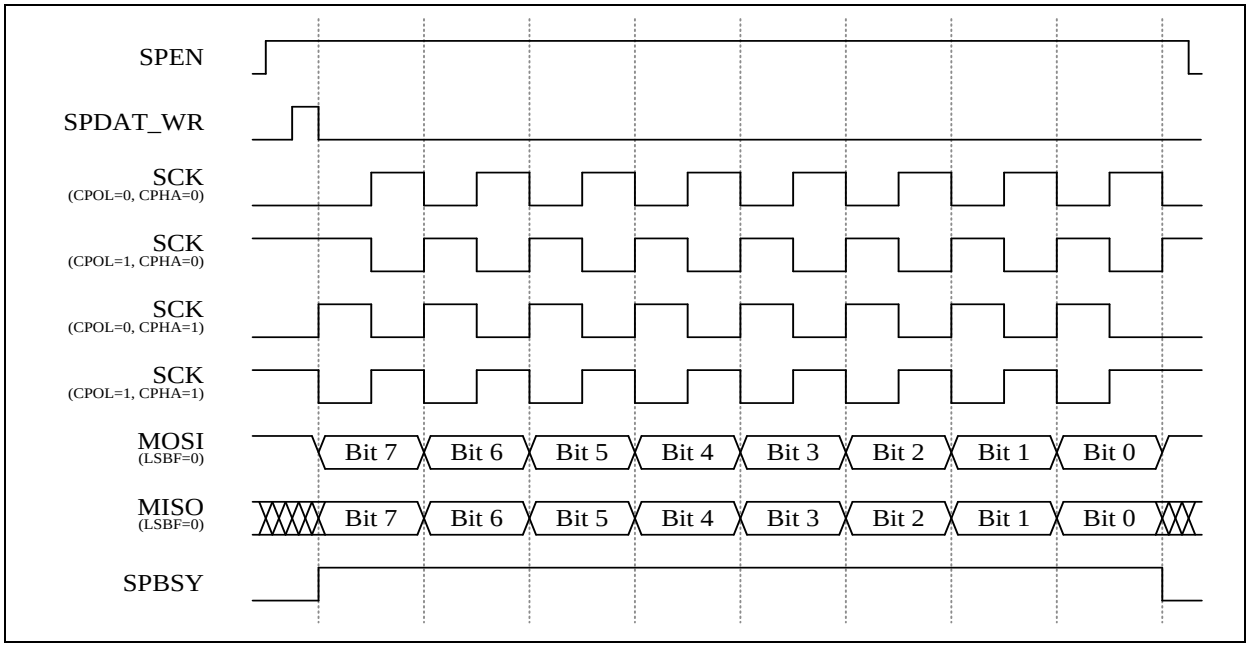
The SPI operates in Master mode by setting the MSTR bit in the SPCON. To start transmit, writing a data to the SPDAT. If SPBSY=0, the data will be transferred to the shift register and starts shift out on the MOSI line. The data of the Slave shift in from the MISO line at the same time. When the SPIF bit becomes set at the end of transfer, the receive data is written to receiver buffer and the RCVBF bit in the SPSTA is set. To prevent an overrun condition, software must read the SPDAT before next byte enters the shift register. The SPBSY bit will be set when writing a data to SPDAT to start transmit, and be cleared at the end of the eighth SCK period in Master mode.

### Slave Mode

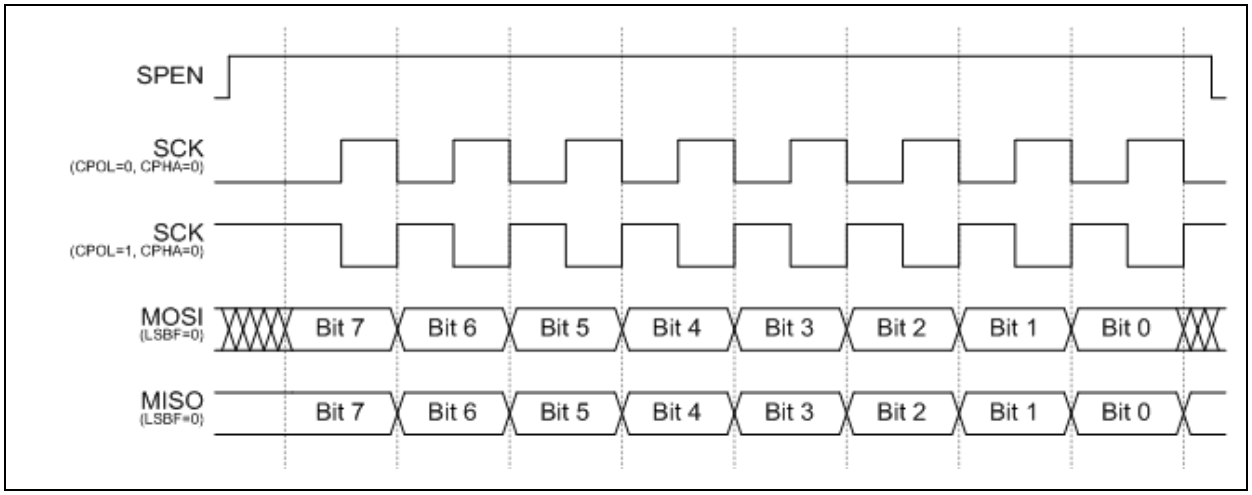
The SPI operates in Slave mode by clearing the MSTR bit in the SPCON. The transmission will start when the SPEN bit in the SPCON is set. The data from a Master will shift into the shift register through the MOSI line, and shift out from the shift register on the MISO line. When a byte enters the shift register, the data will be transferred to receiver buffer if RCVBF=0. If RCVBF=1, the newer received data will not be transferred to receiver buffer and the RCVOVF bit is set. After a byte enters the shift register, the SPIF and RCVBF bits are set. To prevent an overrun condition, software must read the SPDAT or write 0 to RCVBF before next byte enters the shift register. The maximum SCK frequency allowed in Slave mode is $F_{SYSCLK}/4$.

### Serial Clock

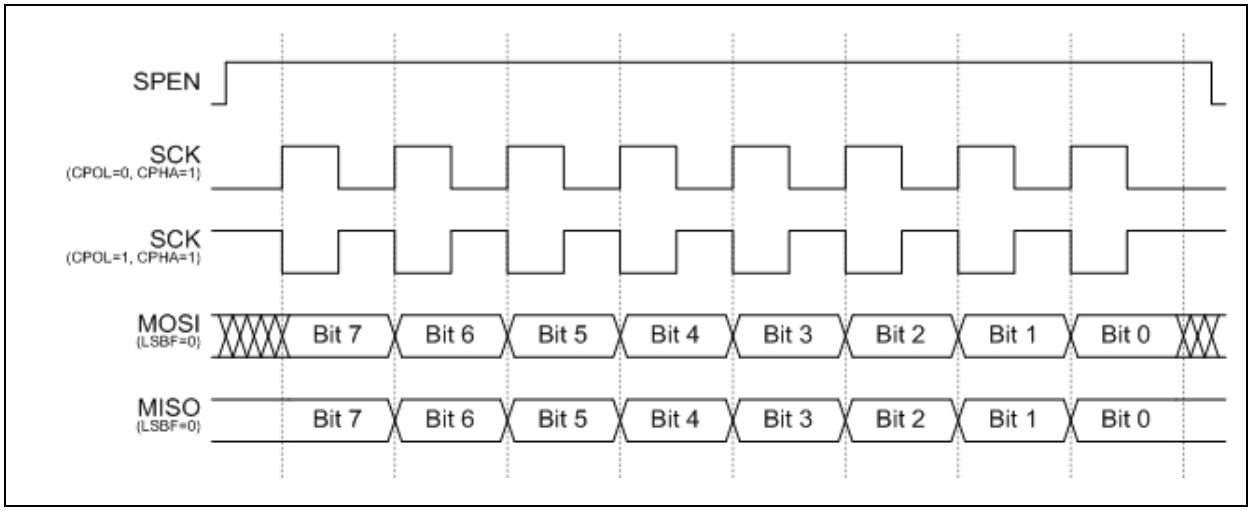
The SPI has four clock types by setting the CPOL and CPHA bits in the SPCON register. The CPOL bit defines the level of the SCK in SPI idle state. The level of the SCK in idle state is low when CPOL=0, and is high when CPOL=1. The CPHA bit defines the edges used to sample and shift data. The SPI sample data on the first edge of SCK period and shift data on the second edge of SCK period when CPHA=0. The SPI sample data on the second edge of SCK period and shift data on first edge of SCK period when CPHA=1. Figures below show the detail timing in Master and Slave modes. Both Master and Slave devices must be configured to use the same clock type before the SPEN bit is set. The SPCR controls the Master mode serial clock frequency. This register is ignored when operating in Slave mode. The SPI clock can select System clock divided by 2, 4, 8, or 16 in Master mode.




Master Mode Timing




Slave Mode Timing (CPHA=0)




Slave Mode Timing (CPHA=1)

In both Master and Slave modes, the SPIF interrupt flag is set by H/W at the end of a data transfer. If write data to SPDAT when SPBSY=1, the WCOL interrupt flag will be set by H/W. When this occurs, the data write to SPDAT will be ignored, and shift register will not be written.

| SFR **BCh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SPCON** | SPEN | MSTR | CPOL | CPHA | – | LSBF | SPCR | |
| R/W | R/W | R/W | R/W | R/W | – | R/W | R/W | |
| Reset | 0 | 0 | 0 | 0 | – | 0 | 0 | 0 |

BCh.7 **SPEN:** SPI Enable.
0: SPI Disable
1: SPI Enable, P2.4~P2.6 are SPI functional pins.

BCh.6 **MSTR:** Master Mode Enable.
0: Slave Mode
1: Master Mode

BCh.5 **CPOL:** SPI Clock Polarity
0: SCK is low in idle state
1: SCK is high in idle state

BCh.4 **CPHA:** SPI Clock Phase
0: Data sampled on first edge of SCK period
1: Data sampled on second edge of SCK period

BCh.2 **LSBF:** LSB First.
0: MSB first
1: LSB first

BCh.1~0 **SPCR:** SPI Clock Rate.
00: $F_{SYSCLK}/2$
01: $F_{SYSCLK}/4$
10: $F_{SYSCLK}/8$
11: $F_{SYSCLK}/16$

| SFR **BDh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SPSTA** | SPIF | WCOL | – | RCVOVF | RCVBF | SPBSY | – | – |
| R/W | R/W | R/W | – | R/W | R/W | R | – | – |
| Reset | 0 | 0 | – | 0 | 0 | – | – | – |

BDh.7 **SPIF:** SPI Interrupt Flag
Set by H/W at the end of a data transfer. Cleared by H/W when interrupt is vectored into. Write 0 to this bit will clear this flag.

BDh.6 **WCOL:** Write Collision Interrupt Flag
Set by H/W if write data to SPDAT when SPBSY=1. Write 0 to this bit or rewrite data to SPDAT when SPBSY=0 will clear this flag.

BDh.4 **RCVOVF:** Receive Buffer Overrun Flag
Set by H/W at the end of a data transfer and RCVBF=1. Write 0 to this bit or read SPDAT register will clear this flag.

BDh.3 **RCVBF:** Receive Buffer Full Flag
Set by H/W at the end of a data transfer. Write 0 to this bit or read SPDAT register will clear this flag.

BDh.2 **SPBSY:** SPI Busy Flag (Read Only)
Set by H/W when a SPI transfer is in progress.

| SFR **BEh** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **SPDAT** | SPDAT | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

BEh.7~0 **SPDAT:** SPI Transmit and Receive Data
The SPDAT register is used to transmit and receive data. Writing data to SPDAT place the data into shift register and start a transfer when in Master mode. Reading SPDAT returns the contents of the receive buffer.

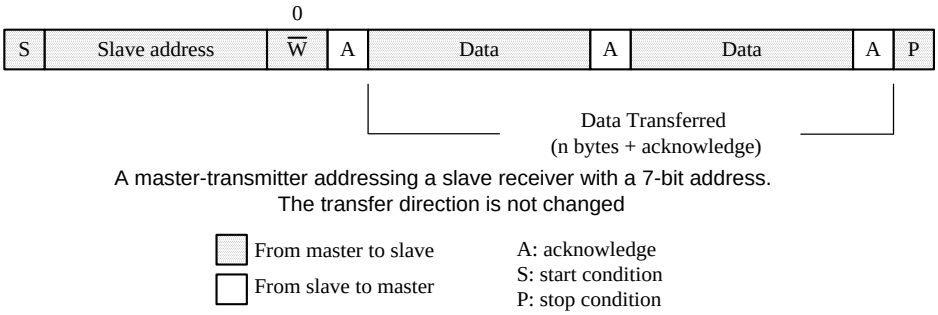
***Note6:*** also refer to Section 6 for more information about SPI Interrupt enable and priority.
***Note7:*** also refer to Section 7 for more information about SPI pins share with I/O pins

## 16. Master I²C Interface

**Master I²C interface transmit mode:**

At the beginning write slave address and direction bit to MIDAT and set MISTART. After the START condition (MISTART), the 7 bits slave address and one bit direction bit are sent. When MIIF convert to 1, address and direction bit transmission was complete. After sending the address and direction bit, user should clear MIIF and write MIDAT to start first data transmission. When MIIF convert to 1, data transfer to slave was complete. User can write MIDAT again to transfer next data to slave. Set MISTOP to finish transmit mode.

MISTART must remain at 1 for the next transfer. After the final data transmit/receive, set MISTOP to finish transmit/receive protocol. MISTART should remain 0 longer than a MSCL clock before starting the next Master I²C protocol. MSCL clock can be adjusted via MICR.




**Master I²C Transmit flow:**

(1) Write slave address and direction bit to MIDAT

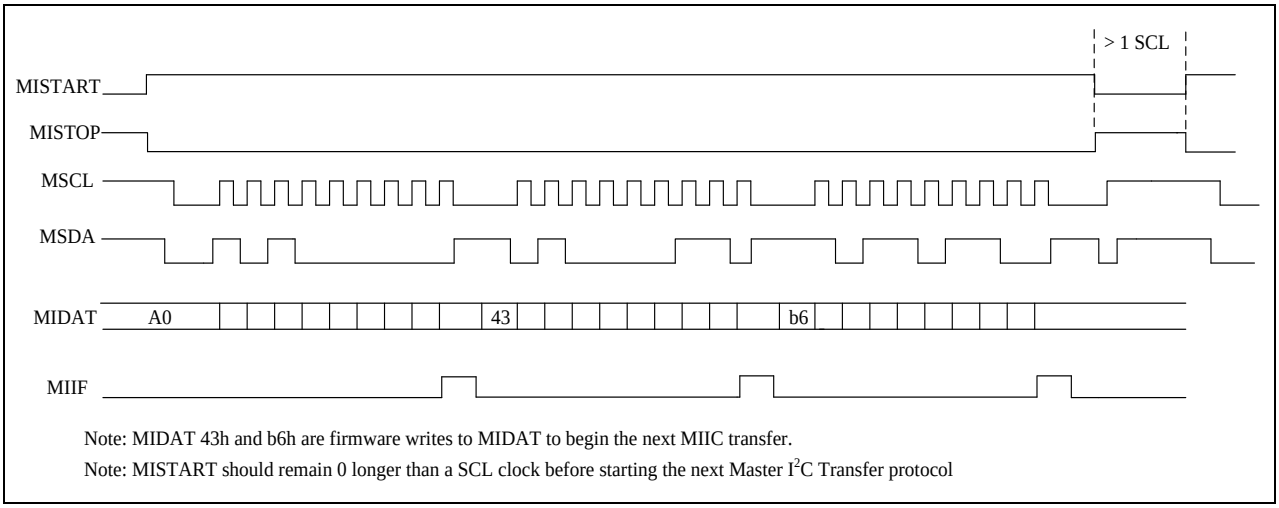
(2) Clear MISTOP and set MISTART to start I²C transmission

(3) Wait until MIIF convert to 1 (interrupt will be issued according to the user's request) and Clear MIIF

(4) Write data to MIDAT to start next transfer (MISTART must remain at 1)

(5) Wait until MIIF convert to 1 (interrupt will be issued according to the user's request) and Clear MIIF, Loop (4) ~ (5) for next transfer.

(6) Clear MISTART and set MISTOP to stop the I²C transfer



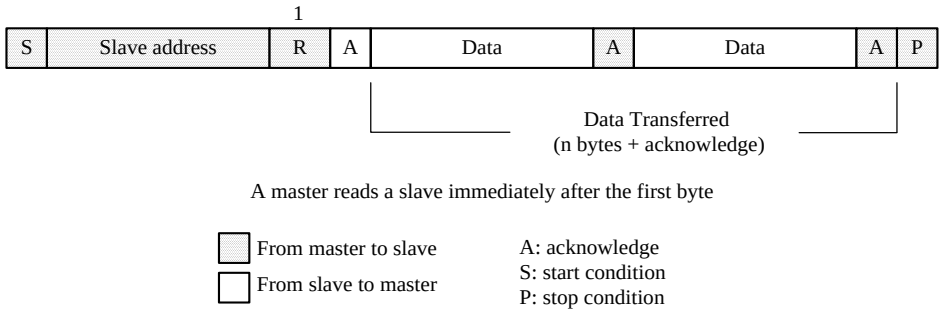

Master Transmit Timing

*Note: MISTART should remain 0 longer than a MSCL period before starting the next Master I²C protocol.*

**Master I²C interface Receive mode:**

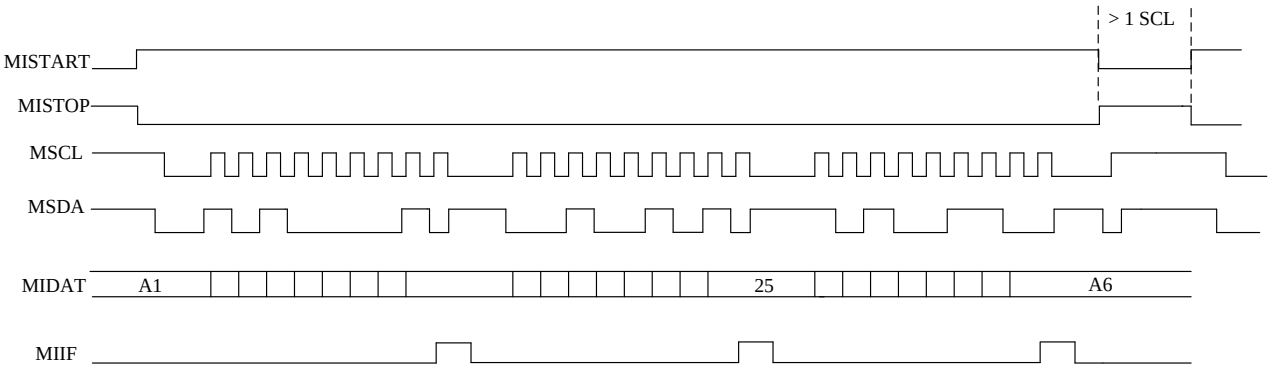
At the beginning write slave address and direction bit to MIDAT and set MISTART. After the START condition (MISTART), the 7 bits slave address and one bit direction bit are sent. When MIIF convert to 1, address and direction bit transmission was complete. After sending the address and direction bit, user should clear MIIF and read MIDAT to start first receive data (The first reading of MIDAT does not represent the data returned by the slave). When MIIF convert to 1, data receive from slave was complete. User can read MIDAT to get data from slave, and start next receive. Set MISTOP to finish receive mode.

MISTART must remain at 1 for the next transfer. After final data transmit/receive, set MISTOP to finish transmit/receive protocol. MISTART should remain 0 longer than a MSCL clock before starting the next Master I²C protocol. MSCL clock can be adjusted via MICR.




**Master I²C Receive flow:**

(1) Write slave address and direction bit to MIDAT
(2) Clear MISTOP and set MISTART to start I²C transmission
(3) Wait until MIIF convert to 1 (interrupt will be issued according to the user's request)
(4) Clear MIIF
(5) Read data from MIDAT to start first receive data
(The first reading of MIDAT does not represent the data returned by the slave)
(6) Wait until MIIF convert to 1
(7) Clear MIIF
(8) Read slave data from MIDAT and receive next data
(9) Loop (6) ~(8)
(10) Set MISTOP to stop the I²C transfer




Note: MIDAT 25h and A6h are data from slave

Note: MISTART should remain 0 longer than a SCL clock before starting the next Master I²C Transfer protocol

Master Receive Timing

| SFR **E1h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **MICON** | MIEN | MIACKO | MIIF | MIACKI | MISTART | MISTOP | MICR | |
| R/W | R/W | R/W | R/W | R | R/W | R/W | R/W | R/W |
| Reset | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |

E1h.7 **MIEN**:Master I²C enable
0: disable
1: enable

E1h.6 **MIACKO**: When Master I²C receive data, send acknowledge to I²C Bus
0: ACK to slave device
1: NACK to slave device

E1h.5 **MIIF**: Master I²C Interrupt flag
0: write 0 to clear it
1: Master I2C transfer one byte complete

E1h.4 **MIACKI**: When Master I²C transfer, acknowledgement form I²C bus (read only)
0: ACK received
1: NACK received

E1h.3 **MISTART**: Master I²C Start bit
1: start I²C bus transfer

E1h.2 **MISTOP**: Master I²C Stop bit
1: send STOP signal to stop I²C bus

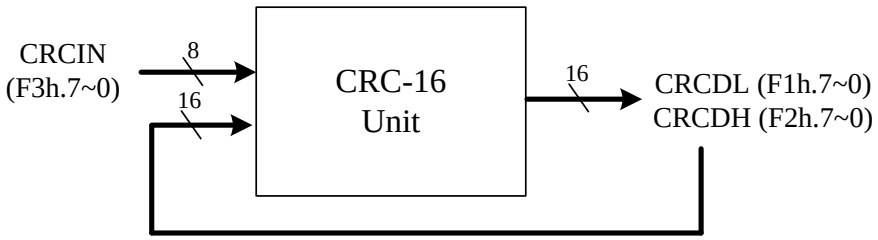
E1h.1~0 **MICR:** Master I²C (MSCL) clock frequency selection
00: Fsys/4 (ex. If Fsys=16MHz, I2C clock is 4 MHz)
01: Fsys/16 (ex. If Fsys=16MHz, I2C clock is 1 MHz)
10: Fsys/64 (ex. If Fsys=16MHz, I2C clock is 250 KHz)
11: Fsys/256 (ex. If Fsys=16MHz, I2C clock is 62.5 KHz)

| SFR **E2h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **MIDAT** | MIDAT | | | | | | | |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Reset | – | – | – | – | – | – | – | – |

E2h.7~0 **MIDAT**: Master I²C data shift register
(W):After Start and before Stop condition, write this register will resume transmission to I²C bus
(R): After Start and before Stop condition, read this register will resume receiving from I²C bus

## 17. Cyclic Redundancy Check (CRC)

The chip supports an integrated 16-bit Cyclic Redundancy Check function. The Cyclic Redundancy Check (CRC) calculation unit is an error detection technique test algorithm and uses to verify data transmission or storage data correctness. The CRC calculation takes a 8-bit data stream or a block of data as input and generates a 16-bit output remainder. The data stream is calculated by the same generator polynomial.




**CRC Block Diagram**

The CRC generator provides the 16-bit CRC result calculation based on the CRC-16-IBM polynomial. In this CRC generator, there are only one polynomial available for the numeric values calculation. It can't support the 16-bit CRC calculations based on any other polynomials. Each write operation to the CRCIN register creates a combination of the previous CRC value stored in the CRCDH and CRCDL registers. It will take one MCU instruction cycle to calculate.

**CRC-16-IBM (Modbus) Polynomial representation: $X^{16} + X^{15} + X^2 + 1$**

| SFR **F1h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CRCDL** | CRCDL | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

F1h.7~0 **CRCDL:** 16-bit CRC checksum data bit 7~0

| SFR **F2h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CRCDH** | CRCDH | | | | | | | |
| R/W | R/W | | | | | | | |
| Reset | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

F2h.7~0 **CRCDL:** 16-bit CRC checksum data bit 15~8

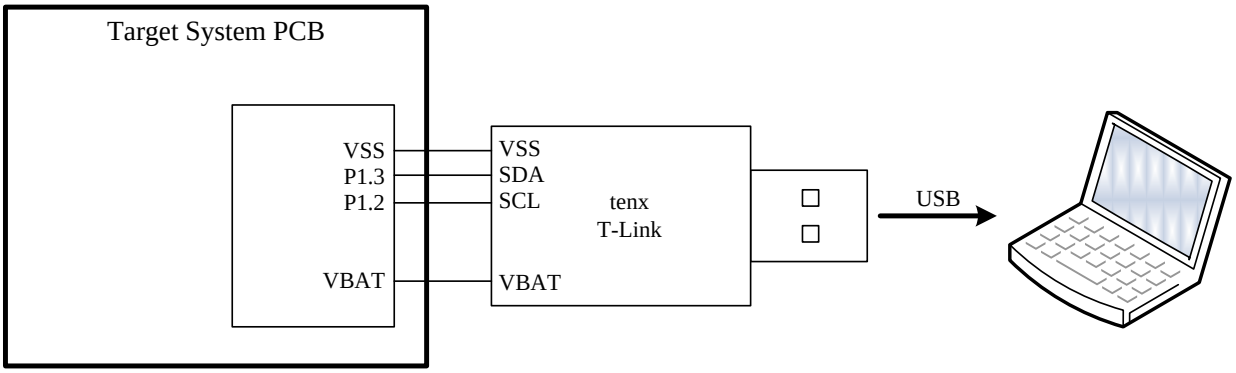
| SFR **F3h** | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| **CRCIN** | CRCIN | | | | | | | |
| W | W | | | | | | | |
| Reset | – | – | – | – | – | – | – | – |

F3h.7~0 **CRCIN:** CRC input data register

## 18. In Circuit Emulation (ICE) Mode

The chip can support the In Circuit Emulation mode. To use the ICE Mode, user just needs to connect P1.2 and P1.3 pin to the tenx proprietary EV module. The benefit is that user can emulate the whole system without changing the on board target device. But there are some limits for the ICE mode as below.

1. The chip must be un-protect.
2. The chip's P1.2 and P1.3 pins must work in input Mode (P1MOD2=0/1 and P1MOD3=0/1).
3. The Program ROM's addressing space 2D00h~2FFFh and 0033h~003Ah are occupied by tenx EV Module. So user Program cannot access these spaces.
4. The P1.2 and P1.3 pin's function cannot be emulated.




ICE Mode Connection

# SFR & CFGW MAP

| Adr | RST | NAME | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 80h | 1111-1111 | **P0** | P0.7 | P0.6 | P0.5 | P0.4 | P0.3 | P0.2 | P0.1 | P0.0 |
| 81h | 0000-0111 | **SP** | SP | | | | | | | |
| 82h | 0000-0000 | **DPL** | DPL | | | | | | | |
| 83h | 0000-0000 | **DPH** | DPH | | | | | | | |
| 87h | 0xxx-0000 | **PCON** | SMOD | – | – | – | GF1 | GF0 | PD | IDL |
| 88h | 0000-0000 | **TCON** | TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 |
| 89h | 0000-0000 | **TMOD** | GATE1 | CT1N | TMOD1 | | GATE0 | CT0N | TMOD0 | |
| 8Ah | 0000-0000 | **TL0** | TL0 | | | | | | | |
| 8Bh | 0000-0000 | **TL1** | TL1 | | | | | | | |
| 8Ch | 0000-0000 | **TH0** | TH0 | | | | | | | |
| 8Dh | 0000-0000 | **TH1** | TH1 | | | | | | | |
| 90h | 1111-1111 | **P1** | P1.7 | P1.6 | P1.5 | P1.4 | P1.3 | P1.2 | P1.1 | P1.0 |
| 91h | 0000-0000 | **P0OE** | P0OE | | | | | | | |
| 92h | x111-1111 | **PINMODE** | – | P2HSEG | | P2LSEG | | P0SEG | | |
| 93h | 0000-0000 | **P2OE** | P2OE | | | | | | | |
| 94h | 1100-0000 | **OPTION** | SXTGAIN | | STPPCK | SXTKICK | UART1W | UARTP1 | T2SEL | T1SEL |
| 95h | 0x00-0000 | **INTFLG** | LBDIF | – | TKIF | ADIF | PWMIF | IE2 | PNCIF | TF3 |
| 96h | 0000-0000 | **P1WKUP** | P1WKUP | | | | | | | |
| 97h | xxxx-xxx0 | **SWCMD** | MTPALL / SWRST | | | | | | | |
| 98h | 0000-0000 | **SCON** | SM0 | SM1 | SM2 | REN | TB8 | RB8 | TI | RI |
| 99h | xxxx-xxxx | **SBUF** | SBUF | | | | | | | |
| 9Ah | 1111-1111 | **PWM0PRD** | PWM0PRD | | | | | | | |
| 9Bh | 1000-0000 | **PWM0DTY** | PWM0DTY | | | | | | | |
| 9Ch | 1111-1111 | **PWM1PRD** | PWM0PRD | | | | | | | |
| 9Dh | 1000-0000 | **PWM1DTY** | PWM1DTY | | | | | | | |
| 9Eh | 1000-0000 | **PWM2DTY** | PWM2DTY | | | | | | | |
| 9Fh | 1000-0000 | **PWM3DTY** | PWM3DTY | | | | | | | |
| A0h | 1111-1111 | **P2** | P2.7 | P2.6 | P2.5 | P2.4 | P2.3 | P2.2 | P2.1 | P2.0 |
| A1h | 0000-0000 | **P3WKUP** | P37WK | P36WK | P35WK | P34WK | P25WK | P24WK | P31WK | P30WK |
| A2h | 0001-0101 | **P1MODL** | P1MOD3 | | P1MOD2 | | P1MOD1 | | P1MOD0 | |
| A3h | 0101-0101 | **P1MODH** | P1MOD7 | | P1MOD6 | | P1MOD5 | | P1MOD4 | |
| A4h | 1111-1111 | **P3MODL** | P3MOD3 | | P3MOD2 | | P3MOD1 | | P3MOD0 | |
| A5h | 0101-0101 | **P3MODH** | P3MOD7 | | P3MOD6 | | P3MOD5 | | P3MOD4 | |
| A6h | 0000-0000 | **P4OE** | P4OE | | | | | | | |
| A7h | 0001-1111 | **VCON** | IVCS | PWRSAV | PORPD | LBDPD | LVRPD | VDDSET | | |
| A8h | 0x00-0000 | **IE** | EA | – | ET2 | ES | ET1 | EX1 | ET0 | EX0 |
| A9h | 0000-0000 | **INTE1** | I2CIE | PWMIE | LBDIE | SPIE | ADTKIE | EX2 | PNCIE | ET3 |
| ABh | xxxx-xxxx | **TKDH** | TKEOC | – | TKDTH | | | | | |
| ACh | xxxx-xxxx | **TKDL** | TKDL | | | | | | | |
| ADh | 1100-1111 | **TKCON** | TKPD | TKTMR | | | TKCHS | | | |
| AEh | 1000-0000 | **TKCON2** | P4SEG | | | | TKREFC | | | |
| AFh | 0000-1100 | **RFCON** | P1RFC | | T0SEL | | RFCPSC | | RFCS | |
| B0h | 1111-1111 | **P3** | P3.7 | P3.6 | P3.5 | P3.4 | P3.3 | P3.2 | P3.1 | P3.0 |
| B1h | 0001-0010 | **LCON** | DSPON | LCDUTY | | | LCDCLK | | LCDFMR | |
| B2h | 0000-0001 | **LCON2** | LEDMOD | LCDPUMP | BIAS2 | LEDBLC | LCDBV | | | |
| B3h | xxxx-xxxx | **TM3SEC** | TM3SEC | | | | | | | |
| B4h | xxxx-xxxx | **TM3DL** | TM3DL | | | | | | | |
| B5h | xxxx-xxxx | **TM3DH** | – | TM3DH | | | | | | |
| B6h | 0000-0000 | **TM3RLD** | TM3RLD | | | | | | | |

| Adr | RST | NAME | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| B7h | 0000-0000 | **TM3ADJ** | TM3ADJS | TM3ADJ | | | | | | |
| B8h | xx00-0000 | **IP** | – | – | PT2 | PS | PT1 | PX1 | PT0 | PX0 |
| B9h | xx00-0000 | **IPH** | – | – | PT2H | PSH | PT1H | PX1H | PT0H | PX0H |
| BAh | xxx0-0000 | **IP1** | PI2C | PPWM | PLBD | PSPI | PADPK | PX2 | PPNC | PT3 |
| BBh | xxx0-0000 | **IP1H** | PI2CH | PPWMH | PLBDH | PSPIH | PADPKH | PX2H | PPNCH | PT3H |
| BCh | 0000-x000 | **SPCON** | SPEN | MSTR | CPOL | CPHA | – | LSBF | SPCR | |
| BDh | 00x0-0xxx | **SPSTA** | SPIF | WCOL | – | RCVOVF | RCVBF | SPBSY | – | – |
| BEh | 0000-0000 | **SPDAT** | SPDAT | | | | | | | |
| C2h | 0000-0000 | **LVSET** | LVRSEL | | | | LBDSEL | | | |
| C3h | xxxx-xxxx | **ADDTL** | LBDO | – | – | ADEOC | ADCDTL | | | |
| C4h | 0001-1111 | **ADCON** | ADCKS | | ADCSRV | ADCHS | | | | |
| C5h | xxxx-xxxx | **ADDTH** | ADDTH | | | | | | | |
| C6h | 0011-0000 | **XBAUD** | XBAUDS | BAUDRT | | | | | | |
| C7h | 0000-0000 | **EFTCON** | EFT2CS | EFT1CS | EFT1S | | EFTSLOW | FRCJMPE | FRCJMPS | CKHLDE |
| C8h | 0000-0000 | **T2CON** | TF2 | EXF2 | RCLK | TCLK | EXEN2 | TR2 | CT2N | CPRL2N |
| C9h | 000x-xxxx | **IAPWE** | MTPWE | IAPTO | EEPWE | IAPWE | | | | |
| CAh | 0000-0000 | **RCP2L** | RCP2L | | | | | | | |
| CBh | 0000-0000 | **RCP2H** | RCP2H | | | | | | | |
| CCh | 0000-0000 | **TL2** | TL2 | | | | | | | |
| CDh | 0000-0000 | **TH2** | TH2 | | | | | | | |
| CEh | 0000-0000 | **PWMCON** | PWM1CKS | PWM1PSC | | | PWM0CKS | PWM0PSC | | |
| CFh | x100-0000 | **PWMCON2** | PWRSAV2 | PWM5CLR | PWM0VX2 | PWM1SNK | PWM5CKS | PWM5PSC | | |
| D0h | 0000-0000 | **PSW** | CY | AC | F0 | RS1 | RS0 | OV | F1 | P |
| D2h | 0001-0000 | **WDTCON** | IAPHTW | TM3PSC | | | WDTMOD | | WDTPSC | |
| D3h | 0000-0011 | **AUX2** | P07ADC | P20ADC | P02TCO | LBDEDGE | VBGE | VBGOUT | IAPTE | |
| D4h | 1000-0000 | **PWM4DTY** | PWM4DTY | | | | | | | |
| D5h | 1000-0000 | **PWM5DTY** | PWM5DTY | | | | | | | |
| D6h | 1111-1111 | **PWM5PRD** | PWM5PRD | | | | | | | |
| D7h | 0000-0000 | **PWMOE** | PWM5OE | PWM4BOE | PWM4AOE | PWM3OE | PWM2OE | PWM1OE | PWM0POE | PWM0NOE |
| D8h | 0000-0111 | **CLKCON** | FCKTYPE | | SELFCK | SCKTYPE | STPFSUB | STPSCK | CLKPSC | |
| E0h | 0000-0000 | **ACC** | ACC.7 | ACC.6 | ACC.5 | ACC.4 | ACC.3 | ACC.2 | ACC.1 | ACC.0 |
| E1h | 000x-0100 | **MICON** | MIEN | MIACKO | MIIF | MIACKI | MISTART | MISTOP | MICR | |
| E2h | 0000-0000 | **MIDAT** | MIDAT | | | | | | | |
| E8h | 1111-1111 | **P4** | P4.7 | P4.6 | P4.5 | P4.4 | P4.3 | P4.2 | P4.1 | P4.0 |
| F0h | 0000-0000 | **B** | B.7 | B.6 | B.5 | B.4 | B.3 | B.2 | B.1 | B.0 |
| F1h | 1111-1111 | **CRCDL** | CRCDL | | | | | | | |
| F2h | 1111-1111 | **CRCDH** | CRCDH | | | | | | | |
| F3h | xxxx-xxxx | **CRCIN** | CRCIN | | | | | | | |
| F5h | xxxx-xxxx | **CFGVBG** | – | – | – | VBGTRIM | | | | |
| F7h | xxxx-xxxx | **CFGFRC** | – | FRCF | | | | | | |
| F8h | xx00-0000 | **AUX1** | – | – | TKSOC | ADSOC | CLRWDT | CLRTM3 | STPRFC | DPSEL |

| MTP Address | NAME | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| 3FFBh | **CFGVBG** | – | – | – | VBGTRIM | | | | |
| 3FFDh | **CFGFRC** | – | FRCF | | | | | | |
| 3FFFh | **CFGWH** | PROT | XRSTE | – | – | – | AGMOD | IAPHVS | – |

# SFR & CFGW DESCRIPTION

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| 80h | **P0** | 7~0 | P0 | R/W | FFh | Port0 data, also controls the P0.n pin's pull-up function. If the P0.n SFR data is "1" and the corresponding P0OE.n=0 (input mode), the pull-up is enabled. |
| 81h | **SP** | 7~0 | SP | R/W | 07h | Stack Point |
| 82h | **DPL** | 7~0 | DPL | R/W | 00h | Data Point low byte |
| 83h | **DPH** | 7~0 | DPH | R/W | 00h | Data Point high byte |
| 87h | **PCON** | 7 | SMOD | R/W | 0 | Set 1 to enable UART double baud rate |
| | | 3 | GF1 | R/W | 0 | General purpose flag bit |
| | | 2 | GF0 | R/W | 0 | General purpose flag bit |
| | | 1 | PD | R/W | 0 | Power down control bit, set 1 to enter Stop / Halt mode |
| | | 0 | IDL | R/W | 0 | Idle control bit, set 1 to enter Idle mode |
| 88h | **TCON** | 7 | TF1 | R/W | 0 | Timer1 overflow flag<br>Set by H/W when Timer/Counter 1 overflows. Cleared by H/W when CPU vectors into the interrupt service routine. |
| | | 6 | TR1 | R/W | 0 | Timer1 run control. 1: timer runs; 0: timer stops |
| | | 5 | TF0 | R/W | 0 | Timer0 overflow flag<br>Set by H/W when Timer/Counter 0 overflows. Cleared by H/W when CPU vectors into the interrupt service routine. |
| | | 4 | TR0 | R/W | 0 | Timer0 run control. 1:timer runs; 0:timer stops |
| | | 3 | IE1 | R/W | 0 | External Interrupt 1 (INT1 pin) edge flag<br>Set by H/W when an INT1 pin falling edge is detected. Cleared by H/W when CPU vectors into the interrupt service routine. |
| | | 2 | IT1 | R/W | 0 | External Interrupt 1 control bit<br>0: Low level active (level triggered) for INT1 pin<br>1: Falling edge active (edge triggered) for INT1 pin |
| | | 1 | IE0 | R/W | 0 | External Interrupt 0 (INT0 pin) edge flag<br>Set by H/W when an INT0 pin falling edge is detected. Cleared by H/W when CPU vectors into the interrupt service routine. |
| | | 0 | IT0 | R/W | 0 | External Interrupt 0 control bit<br>0: Low level active (level triggered) for INT0 pin<br>1: Falling edge active (edge triggered) for INT0 pin |
| 89h | **TMOD** | 7 | GATE1 | R/W | 0 | Timer1 gating control bit<br>0: Timer1 enable when TR1 bit is set<br>1: Timer1 enable only while the INT1 pin is high and TR1 bit is set |
| | | 6 | CT1N | R/W | 0 | Timer1 Counter/Timer select bit<br>0: Timer mode, Timer1 data increases at 2 System clock cycle rate<br>1: Counter mode, Timer1 increases by T1 pin or Slow clock event |
| | | 5~4 | TMOD1 | R/W | 00 | Timer1 mode select<br>00: 8-bit timer/counter (TH1) and 5-bit prescaler (TL1)<br>01: 16-bit timer/counter<br>10: 8-bit auto-reload timer/counter (TL1). Reloaded from TH1 at overflow.<br>11: Timer1 stops |
| | | 3 | GATE0 | R/W | 0 | Timer0 gating control bit<br>0: Timer0 enable when TR0 bit is set<br>1: Timer0 enable only while the INT0 pin is high and TR0 bit is set |
| | | 2 | CT0N | R/W | 0 | Timer0 Counter/Timer select bit<br>0: Timer mode, Timer0 data increases at 2 System clock cycle rate<br>1: Counter mode, Timer0 increases by T0 pin, Slow clock or RFC event |
| | | 1~0 | TMOD0 | R/W | 00 | Timer0 mode select<br>00: 8-bit timer/counter (TH0) and 5-bit prescaler (TL0)<br>01: 16-bit timer/counter<br>10: 8-bit auto-reload timer/counter (TL0). Reloaded from TH0 at overflow.<br>11: TL0 is an 8-bit timer/counter.<br>TH0 is an 8-bit timer/counter using Timer1's TR1 and TF1 bits. |
| 8Ah | **TL0** | 7~0 | TL0 | R/W | 00h | Timer0 data low byte |
| 8Bh | **TL1** | 7~0 | TL1 | R/W | 00h | Timer1 data low byte |
| 8Ch | **TH0** | 7~0 | TH0 | R/W | 00h | Timer0 data high byte |
| 8Dh | **TH1** | 7~0 | TH1 | R/W | 00h | Timer1 data high byte |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| 90h | **P1** | 7~0 | P1 | R/W | FFh | Port1 data |
| 91h | **P0OE** | 7~0 | P0OE | R/W | 00h | Port0 CMOS Push-Pull output enable control, 1=Enable. |
| 92h | **PINMODE** | 6~5 | P2HSEG | R/W | 11 | P2.4~P2.6 pin LCD/LED mode control.<br>00: P2.4~P2.6 are I/O pins<br>01: P2.4 and P2.5 are I/O pins, P2.6 is LCD/LED Segment pin<br>10: P2.4 is I/O pin, P2.5 and P2.6 are LCD/LED Segment pins<br>11: P2.4~P2.6 are LCD/LED Segment pins |
| | | 4~3 | P2LSEG | R/W | 11 | P2.1~P2.3 pin LCD/LED mode control.<br>00: P2.1~P2.3 are I/O pins<br>01: P2.1 and P2.2 are I/O pins, P2.3 is LCD/LED Segment pin<br>10: P2.1 is I/O pin, P2.2 and P2.3 are LCD/LED Segment pins<br>11: P2.1~P2.3 are LCD/LED Segment pins |
| | | 2~0 | P0SEG | R/W | 111 | Port0 LCD/LED mode control.<br>000: P0.0~P0.6 are I/O pins<br>001: P0.0~P0.5 are I/O pins, P0.6 is LCD/LED Segment pin<br>010: P0.0~P0.4 are I/O pins, P0.5~P0.6 are LCD/LED Segment pins<br>011: P0.0~P0.3 are I/O pins, P0.4~P0.6 are LCD/LED Segment pins<br>100: P0.0~P0.2 are I/O pins, P0.3~P0.6 are LCD/LED Segment pins<br>101: P0.0~P0.1 are I/O pins, P0.2~P0.6 are LCD/LED Segment pins<br>110: P0.0 is I/O pin, P0.1~P0.6 are LCD/LED Segment pins<br>111: P0.0~P0.6 are LCD/LED Segment pins |
| 93h | **P2OE** | 7~0 | P2OE | R/W | 00h | Port2 CMOS Push-Pull output enable control, 1=Enable. |
| 94h | **OPTION** | 7~6 | SXTGAIN | R/W | 11 | SXT oscillator gain<br>0=Lowest gain, 3=Highest Gain |
| | | 5 | STPPCK | R/W | 0 | Set 1 to stop UART/Timer0/1/2 clock in Idle mode for current reducing. |
| | | 4 | SXTKICK | R/W | 0 | Set 1 to kick SXT by LCD pump, for crystal start up @$V_{BAT}$ < 1.5V |
| | | 3 | UART1W | R/W | 0 | Set 1 to enable one wire UART mode, both TXD/RXD use P3.1 pin. |
| | | 2 | UARTP1 | R/W | 0 | UART pin select<br>0: P3.0 / P3.1 is UART RXD / TXD<br>1: P1.2 / P1.3 is UART RXD / TXD |
| | | 1 | T2SEL | R/W | 0 | Timer2 Counter mode (CT2N=1) input select<br>0: P1.0 pin (8051 standard)<br>1: Slow clock divided by 16 (SLOWCLK/16) |
| | | 0 | T1SEL | R/W | 0 | Timer1 Counter mode (CT1N=1) input select<br>0: P3.5 pin (8051 standard)<br>1: Slow clock divided by 16 (SLOWCLK/16) |
| 95h | **INTFLG** | 7 | LBDIF | R/W | 0 | LBD Interrupt Flag<br>Set by H/W at LBDO's rising or falling edge. Cleared by H/W when CPU vectors into the interrupt. S/W writes 7Fh to INTFLG to clear this flag. |
| | | 5 | TKIF | R/W | 0 | Touch Key Interrupt Flag<br>Set by H/W when TK end of conversion. S/W can write DFh to INTFLG to clear this bit. |
| | | 4 | ADIF | R/W | 0 | ADC Interrupt Flag<br>Set by H/W when ADC end of conversion. S/W can write EFh to INTFLG to clear this bit |
| | | 3 | PWMIF | R/W | 0 | PWM5 period counter full interrupt flag<br>Set by H/W when PWM5 period counter full. Cleared by H/W when CPU vectors into the interrupt. S/W can write F7h to INTFLG to clear this bit. |
| | | 2 | IE2 | R/W | 0 | External Interrupt 2 (INT2 pin) edge flag<br>Set by H/W when a falling edge is detected on the INT2 pin, no matter the EX2 is 0 or 1. Cleared by H/W when CPU vectors into the interrupt. S/W can write FBh to INTFLG to clear this bit. |
| | | 1 | PNCIF | R/W | 0 | Pin change Interrupt flag<br>Set by H/W when a Port1~3 pin state change is detected and its interrupt enable bit is set (P1WKUP/P3WKUP). PNCIE does not affect this flag's setting. Cleared by H/W when CPU vectors into the interrupt. S/W can write FDh to INTFLG to clear this bit. |
| | | 0 | TF3 | R/W | 0 | Timer3 Interrupt Flag<br>Set by H/W when Timer3 reaches TM3PSC setting cycles. Cleared by H/W when CPU vectors into the interrupt. S/W can write FEh to INTFLG to clear this bit. |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| 96h | **P1WKUP** | 7~0 | P1WKUP | R/W | 00h | P1.7~P1.0 pin individual Wake-up / Interrupt enable control<br>0: Disable; 1: Enable. |
| 97h | **SWCMD** | 7~0 | SWRST | W | – | Write 56h to generate S/W Reset |
| | | 7~0 | MTPALL | W | – | Write 65h to set MTPALL flag and enable MTP IAP; Write other value to clear MTPALL flag and disable IAP. It is recommended to clear it immediately after IAP access. |
| | | 0 | MTPALL | R | 0 | Flag indicates MTP area can be access by IAP or not |
| 98h | **SCON** | 7 | SM0 | R/W | 0 | Serial port mode select bit 0, 1 (SM0, SM1)=<br>00: Mode0: 8 bit shift register, Baud Rate = $F_{SYSCLK}$ / 2<br>01: Mode1: 8 bit UART, Baud Rate is variable<br>10: Mode2: 9 bit UART, Baud Rate = $F_{SYSCLK}$ / 32 or / 64<br>11: Mode3: 9 bit UART, Baud Rate is variable |
| | | 6 | SM1 | R/W | 0 | |
| | | 5 | SM2 | R/W | 0 | Serial port mode select bit 2<br>SM2 enables multiprocessor communication over a single serial line and modifies the above as follows. In Modes 2 & 3, if SM2 is set then the received interrupt will not be generated if the received ninth data bit is 0. In Mode 1, the received interrupt will not be generated unless a valid stop bit is received. In Mode 0, SM2 should be 0. |
| | | 4 | REN | R/W | 0 | Set 1 to enable UART Reception |
| | | 3 | TB8 | R/W | 0 | Transmitter bit 8, ninth bit to transmit in Modes 2 and 3 |
| | | 2 | RB8 | R/W | 0 | Receive Bit 8, contains the ninth bit that was received in Mode 2 and 3 or the stop bit in Mode 1 if SM2=0 |
| | | 1 | TI | R/W | 0 | Transmit Interrupt flag<br>Set by H/W at the end of the eighth bit in Mode 0, or at the beginning of the stop bit in other modes. Must be cleared by S/W |
| | | 0 | RI | R/W | 0 | Receive Interrupt flag<br>Set by H/W at the end of the eighth bit in Mode 0, or at the sampling point of the stop bit in other modes. Must be cleared by S/W. |
| 99h | **SBUF** | 7~0 | SBUF | R/W | – | UART transmit and receive data. Transmit data is written to this location and receive data is read from this location, but the paths are independent. |
| 9Ah | **PWM0PRD** | 7~0 | PWM0PRD | R/W | FFh | PWM0 Period, FFh=256 PWMCLK, 7Fh=128 PWMCLK |
| 9Bh | **PWM0DTY** | 7~0 | PWM0DTY | R/W | 80h | PWM0 Duty, 0=0 PWMCLK, 80h=128 PWMCLK, FFh=255 PWMCLK |
| 9Ch | **PWM1PRD** | 7~0 | PWM1PRD | R/W | FFh | PWM1~4 Period, FFh=256 PWMCLK, 7Fh=128 PWMCLK |
| 9Dh | **PWM1DTY** | 7~0 | PWM1DTY | R/W | 80h | PWM1 Duty, 0=0 PWMCLK, 80h=128 PWMCLK, FFh=255 PWMCLK |
| 9Eh | **PWM2DTY** | 7~0 | PWM2DTY | R/W | 80h | PWM2 Duty, 0=0 PWMCLK, 80h=128 PWMCLK, FFh=255 PWMCLK |
| 9Fh | **PWM3DTY** | 7~0 | PWM3DTY | R/W | 80h | PWM3 Duty, 0=0 PWMCLK, 80h=128 PWMCLK, FFh=255 PWMCLK |
| A0h | **P2** | 7~0 | P2 | R/W | FFh | Port2 data, also controls the P2.n pin's pull-up function. If the P2.n SFR data is "1" and the corresponding P2OE.n=0 (input mode), the pull-up is enabled. |
| A1h | **P3WKUP** | 7 | P37WK | R/W | 0 | P3.7~4, P2.5~4, P3.1~0 pin individual Wake-up / Interrupt enable control<br>0: Disable<br>1: Enable |
| | | 6 | P36WK | R/W | 0 | |
| | | 5 | P35WK | R/W | 0 | |
| | | 4 | P34WK | R/W | 0 | |
| | | 3 | P25WK | R/W | 0 | |
| | | 2 | P24WK | R/W | 0 | |
| | | 1 | P31WK | R/W | 0 | |
| | | 0 | P30WK | R/W | 0 | |
| A2h | **P1MODL** | 7~6 | P1MOD3 | R/W | 00 | P1.3 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3, P1.3 is ADC input |
| | | 5~4 | P1MOD2 | R/W | 01 | P1.2 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3, P1.2 is ADC input |
| | | 3~2 | P1MOD1 | R/W | 01 | P1.1 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3, P1.1 is ADC input |
| | | 1~0 | P1MOD0 | R/W | 01 | P1.0 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3, P1.0 is ADC input |
| A3h | **P1MODH** | 7~6 | P1MOD7 | R/W | 01 | P1.7 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3, P1.7 is ADC input |
| | | 5~4 | P1MOD6 | R/W | 01 | P1.6 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3, P1.6 is ADC input |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| A3h | **P1MODH** | 3~2 | P1MOD5 | R/W | 01 | P1.5 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3, P1.5 is ADC input |
| | | 1~0 | P1MOD4 | R/W | 01 | P1.4 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3, P1.4 is ADC input |
| A4h | **P3MODL** | 7~6 | P3MOD3 | R/W | 11 | P3.3 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3, P3.3 is LCD Segment |
| | | 5~4 | P3MOD2 | R/W | 11 | P3.2 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3, P3.2 is LCD Segment |
| | | 3~2 | P3MOD1 | R/W | 11 | P3.1 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3, P3.1 is LCD Segment |
| | | 1~0 | P3MOD0 | R/W | 11 | P3.0 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3, P3.0 is LCD Segment |
| A5h | **P3MODH** | 7~6 | P3MOD7 | R/W | 01 | P3.7 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Not defined |
| | | 5~4 | P3MOD6 | R/W | 01 | P3.6 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Not defined |
| | | 3~2 | P3MOD5 | R/W | 01 | P3.5 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3, P3.5 is ADC input |
| | | 1~0 | P3MOD4 | R/W | 01 | P3.4 Pin Control<br>00: Mode0; 01: Mode1; 10: Mode2; 11: Mode3, P3.4 is ADC input |
| A6h | **P4OE** | 7~0 | P4OE | R/W | 00h | Port4 CMOS Push-Pull output enable control, 1=Enable. |
| A7h | **VCON** | 7 | IVCS | R/W | 0 | Chip internal LDO Regulator enable control<br>0: LDO disable, $V_{DD}$ = $V_{BAT}$<br>1: LDO enable, $V_{DD}$ = LDO Regulator output |
| | | 6 | PWRSAV | R/W | 0 | Power saving mode control<br>0: No power saving<br>1: Power saving, disable POR in Halt mode, disable LVR/LBD in Idle/Halt/Stop mode, POR 1/16 duty. |
| | | 5 | PORPD | R/W | 0 | POR control, 1=force POR disable |
| | | 4 | LBDPD | R/W | 1 | LBD control, 1=force LBD disable |
| | | 3 | LVRPD | R/W | 1 | LVR control, 1=force LVR disable |
| | | 2~0 | VDDSET | R/W | 111 | $V_{DD}$ setting while IVCS=1.<br>000: $V_{DD}$ =$V_{BAT}$*0.375; 001: $V_{DD}$ =$V_{BAT}$*0.425;<br>010: $V_{DD}$ =$V_{BAT}$*0.475; 011: $V_{DD}$ =$V_{BAT}$*0.525;<br>100: $V_{DD}$ =$V_{BAT}$*0.575; 101: $V_{DD}$ =$V_{BAT}$*0.625;<br>110: $V_{DD}$ =$V_{BAT}$*0.675; 111: $V_{DD}$ =$V_{BAT}$*0.725; |
| A8h | **IE** | 7 | EA | R/W | 0 | Global interrupt enable control.<br>0: Disable all Interrupts.<br>1: Each interrupt is enabled or disabled by its own interrupt control bit. |
| | | 5 | ET2 | R/W | 0 | Set 1 to enable Timer2 interrupt |
| | | 4 | ES | R/W | 0 | Set 1 to enable Serial Port (UART) Interrupt |
| | | 3 | ET1 | R/W | 0 | Set 1 to enable Timer1 Interrupt |
| | | 2 | EX1 | R/W | 0 | Set 1 to enable external INT1 pin Interrupt & Stop/Halt mode wake up capability |
| | | 1 | ET0 | R/W | 0 | Set 1 to enable Timer0 Interrupt |
| | | 0 | EX0 | R/W | 0 | Set 1 to enable external INT0 pin Interrupt & Stop/Halt mode wake up capability |
| A9h | **INTE1** | 7 | I2CIE | R/W | 0 | Set 1 to enable Master I2C Interrupt |
| | | 6 | PWMIE | R/W | 0 | Set 1 to enable external PWM Interrupt & Halt mode wake up capability |
| | | 5 | LBDIE | R/W | 0 | Set 1 to enable LBD Interrupt |
| | | 4 | SPIE | R/W | 0 | Set 1 to enable SPI Interrupt |
| | | 3 | ADTKIE | R/W | 0 | Set 1 to enable ADC / Touch Key Interrupt |
| | | 2 | EX2 | R/W | 0 | Set 1 to enable external INT2 pin Interrupt & Stop/Halt mode wake up capability |
| | | 1 | PNCIE | R/W | 0 | Set 1 to enable Port1~3 Pin Change Interrupt |
| | | 0 | ET3 | R/W | 0 | Set 1 to enable Timer3 Interrupt & Halt mode wake up capability |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| ABh | **TKDH** | 7 | TKEOC | R | – | Touch Key End of Conversion, 1=EOC. |
| | | 5~0 | TKDTH | R | – | Touch Key Counter Data 13~8 |
| ACh | **TKDL** | 7~0 | TKDL | R | – | Touch Key Counter Data 7~0 |
| ADh | **TKCON** | 7 | TKPD | R/W | 1 | Touch Key Power Down<br>0: Touch Key enable<br>1: Touch Key disable |
| | | 6~4 | TKTMR | R/W | 100 | Touch Key Conversion Time<br>000: Conversion time shortest …<br>111: Conversion time longest |
| | | 3~0 | TKCHS | R/W | 1111 | Touch Key Channel Select<br>0000: TK0 (P1.0) 0101: TK5 (P1.5)<br>0001: TK1 (P1.1) 0110: TK6 (P1.6)<br>0010: TK2 (P1.2) 0111: TK7 (P1.7)<br>0011: TK3 (P1.3) 1000: TK8 (P3.4)<br>0100: TK4 (P1.4) 1001: TK9 (P3.5)<br>1111: TK15 (Internal reference) |
| AEh | **TKCON2** | 7~4 | P4SEG | R/W | 1000 | Port4 LCD/LED mode control.<br>0000: P4.0~P4.7 are I/O pins<br>0001: P4.0~P4.6 are I/O pins, P4.7 is LCD/LED Segment pin<br>0010: P4.0~P4.5 are I/O pins, P4.6~P4.7 are LCD/LED Segment pins<br>0011: P4.0~P4.4 are I/O pins, P4.5~P4.7 are LCD/LED Segment pins<br>0100: P4.0~P4.3 are I/O pins, P4.4~P4.7 are LCD/LED Segment pins<br>0101: P4.0~P4.2 are I/O pins, P4.3~P4.7 are LCD/LED Segment pins<br>0110: P4.0~P4.1 are I/O pins, P4.2~P4.7 are LCD/LED Segment pins<br>0111: P4.0 is I/O pin, P4.1~P4.7 are LCD/LED Segment pins<br>1000: P4.0~P4.7 are LCD/LED Segment pins |
| | | 3~0 | TKREFC | R/W | 0000 | Touch Key reference clock capacitor select<br>0000: Smallest (conversion time shortest) …<br>1111: Biggest (conversion time longest) |
| AFh | **RFCON** | 7~6 | P1RFC | R/W | 00 | P1.7~P1.4 pin RFC mode control.<br>00: P1.7~P1.4 are not RFC pins<br>01: P1.7 and P1.6 are RFC pins, P1.5 and P1.4 are not RFC pins<br>10: P1.7~P1.5 are RFC pins, P1.4 is not RFC pin<br>11: P1.7~P1.4 are RFC pins |
| | | 5~4 | T0SEL | R/W | 00 | Timer0 Counter mode (CT0N=1) T0 input select<br>00: P3.4 pin (8051 standard)<br>01: RFC clock divided by 1/4/16/64<br>10: Slow clock divided by 16 (SLOWCLK/16)<br>11: RFC clock divided by 1/4/16/64 gated by Timer2 overflow |
| | | 3~2 | RFCPSC | R/W | 11 | RFC clock divider to Timer0<br>00: divided by 64<br>01: divided by 16<br>10: divided by 4<br>11: divided by 1 |
| | | 1~0 | RFCS | R/W | 00 | Select RFC convert channel.<br>00: RFC0R (P1.6)<br>01: RFC1R (P1.5)<br>10: RFC2R (P1.4) |
| B0h | **P3** | 7~0 | P3 | R/W | FFh | Port 3 data |
| B1h | **LCON** | 7 | DSPON | R/W | 0 | LCD / LED display enable control. 1=Enable |
| | | 6~4 | LCDUTY | R/W | 001 | LCD / LED duty control.<br>000: 1/3 duty 011: 1/6 duty<br>001: 1/4 duty 100: 1/7 duty<br>010: 1/5 duty 101: 1/8 duty<br>111: All LED Segment DC output, SEG0~2 replace the COM0~2 |
| | | 3~2 | LCDCLK | R/W | 00 | LCD / LED clock source<br>00: SLOWCLK 10: FASTCLK/128<br>01: SLOWCLK/2 11: FASTCLK/256 |
| | | 1~0 | LCDFMR | R/W | 10 | LCD /LED Frame Rate, 3=Highest; 0=Lowest |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| B2h | **LCON2** | 7 | LEDMOD | R/W | 0 | LCD / LED mode select for COM and SEG pins<br>0: LCD mode;<br>1: LED mode |
| | | 6 | LCDPUMP | R/W | 0 | LCD pump / LED DMX mode select<br>0: LCD no pump / LED Normal mode<br>1: LCD Pump / LED DMX mode |
| | | 5 | BIAS2 | R/W | 0 | LCD Bias select<br>0: 1/3 Bias;<br>1: 1/2 Bias |
| | | 4 | LEDBLC | R/W | 0 | LED brightness balance<br>0: LED Normal Brightness;<br>1: LED Balanced Brightness |
| | | 3~0 | LCDBV | R/W | 0001 | LCD Brightness, VLCD Voltage level control<br>0000: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/40<br>0001: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/38<br>0010: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/37<br>0011: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/36<br>0100: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/35<br>0101: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/34<br>0110: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/33<br>0111: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/32<br>1000: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/31<br>1001: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/30<br>1010: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/29<br>1011: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/28<br>1100: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/27<br>1101: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/26<br>1110: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/25<br>1111: VLCD = (LCDPUMP + 1) * $V_{BAT}$ * 24/24 |
| B3h | **TM3SEC** | 7~0 | TM3SEC | R | – | Timer3 count data bit 22~15 |
| B4h | **TM3DL** | 7~0 | TM3DL | R | – | Timer3 count data bit 7~0 |
| B5h | **TM3DH** | 6~0 | TM3DH | R | – | Timer3 count data bit 14~8 |
| B6h | **TM3RLD** | 7~0 | TM3RLD | R/W | 00h | Timer3 overflow reload data for Timer3 bit 22~15 (TM3SEC) |
| B7h | **TM3ADJ** | 7 | TM3ADJS | R/W | 0 | Timer3 adjustment sign<br>0: Timer3 positive adjust, to increase Timer3 counting rate<br>1: Timer3 negative adjust, to decrease Timer3 counting rate |
| | | 6~0 | TM3ADJ | R/W | 00h | Timer3 adjust magnitude, 0.477 ppm per LSB.<br>The adjustment is calculated as ±TM3ADJ*0.477ppm. The total adjustable range is ± 61ppm. |
| B8h | **IP** | 5 | PT2 | R/W | 0 | Timer2 Interrupt Priority Low bit |
| | | 4 | PS | R/W | 0 | Serial Port (UART) Interrupt Priority Low bit |
| | | 3 | PT1 | R/W | 0 | Timer1 Interrupt Priority Low bit |
| | | 2 | PX1 | R/W | 0 | External INT1 Pin Interrupt Priority Low bit |
| | | 1 | PT0 | R/W | 0 | Timer0 Interrupt Priority Low bit |
| | | 0 | PX0 | R/W | 0 | External INT0 Pin Interrupt Priority Low bit |
| B9h | **IPH** | 5 | PT2H | R/W | 0 | Timer2 Interrupt Priority High bit |
| | | 4 | PSH | R/W | 0 | Serial Port (UART) Interrupt Priority High bit |
| | | 3 | PT1H | R/W | 0 | Timer1 Interrupt Priority High bit |
| | | 2 | PX1H | R/W | 0 | External INT1 Pin Interrupt Priority High bit |
| | | 1 | PT0H | R/W | 0 | Timer0 Interrupt Priority High bit |
| | | 0 | PX0H | R/W | 0 | External INT0 Pin Interrupt Priority High bit |
| BAh | **IP1** | 7 | PI2C | R/W | 0 | I2C Interrupt Priority Low bit |
| | | 6 | PPWM | R/W | 0 | PWM Interrupt Priority Low bit |
| | | 5 | PLBD | R/W | 0 | LBD Interrupt Priority Low bit |
| | | 4 | PSPI | R/W | 0 | SPI Interrupt Priority Low bit |
| | | 3 | PADTK | R/W | 0 | ADC / Touch Key Interrupt Priority Low bit |
| | | 2 | PX2 | R/W | 0 | External INT2 Pin Interrupt Priority Low bit |
| | | 1 | PPNC | R/W | 0 | Pin change Interrupt Priority Low bit |
| | | 0 | PT3 | R/W | 0 | Timer3 Interrupt Priority Low bit |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| BBh | **IP1H** | 7 | PI2CH | R/W | 0 | I2C Interrupt Priority High bit |
| | | 6 | PPWMH | R/W | 0 | PWM Interrupt Priority High bit |
| | | 5 | PLBDH | R/W | 0 | LBD Interrupt Priority High bit |
| | | 4 | PSPIH | R/W | 0 | SPI Interrupt Priority High bit |
| | | 3 | PADTKH | R/W | 0 | ADC / Touch Key Interrupt Priority High bit |
| | | 2 | PX2H | R/W | 0 | External INT2 Pin Interrupt Priority High bit |
| | | 1 | PPNCH | R/W | 0 | Pin change Interrupt Priority High bit |
| | | 0 | PT3H | R/W | 0 | Timer3 Interrupt Priority High bit |
| BCh | **SPCON** | 7 | SPEN | R/W | 0 | Set 1 to enable SPI & P2.4~P2.6 SPI pin function |
| | | 6 | MSTR | R/W | 0 | SPI Master Mode Enable.<br>0: Slave Mode; 1: Master Mode |
| | | 5 | CPOL | R/W | 0 | SPI Clock Polarity<br>0: SCK is low in idle state;<br>1: SCK is high in idle state |
| | | 4 | CPHA | R/W | 0 | SPI Clock Phase<br>0: Data sampled on first edge of SCK period<br>1: Data sampled on second edge of SCK period |
| | | 2 | LSBF | R/W | 0 | SPI LSB First.<br>0: MSB first; 1: LSB first |
| | | 1~0 | SPCR | R/W | 00 | SPI Clock Rate.<br>00: $F_{SYSCLK}/2$; 01: $F_{SYSCLK}/4$; 10: $F_{SYSCLK}/8$; 11: $F_{SYSCLK}/16$ |
| BDh | **SPSTA** | 7 | SPIF | R/W | 0 | SPI Interrupt Flag<br>Set by H/W at the end of a data transfer. Cleared by H/W when interrupt is vectored into. Write 0 to this bit will clear this flag. |
| | | 6 | WCOL | R/W | 0 | Write Collision Interrupt Flag<br>Set by H/W if write data to SPDAT when SPBSY=1. Write 0 to this bit or rewrite data to SPDAT when SPBSY=0 will clear this flag. |
| | | 4 | RCVOVF | R/W | 0 | Receive Buffer Overrun Flag<br>Set by H/W at the end of a data transfer and RCVBF=1. Write 0 to this bit or read SPDAT register will clear this flag. |
| | | 3 | RCVBF | R/W | 0 | Receive Buffer Full Flag<br>Set by H/W at the end of a data transfer. Write 0 to this bit or read SPDAT register will clear this flag. |
| | | 2 | SPBSY | R | – | SPI Busy Flag (Read Only)<br>Set by H/W when a SPI transfer is in progress. |
| BEh | **SPDAT** | 7~0 | SPDAT | R/W | 00h | SPI Transmit and Receive Data<br>The SPDAT register is used to transmit and receive data. Writing data to SPDAT place the data into shift register and start a transfer when in Master mode. Reading SPDAT returns the contents of the receive buffer. |
| C2h | **LVSET** | 7~4 | LVRSEL | R/W | 0000 | Low Voltage Reset select<br>0000: LVR=1.73V 0001: LVR=1.85V<br>0010: LVR=1.98V 0011: LVR=2.10V<br>0100: LVR=2.22V 0101: LVR=2.34V<br>0110: LVR=2.46V 0111: LVR=2.59V<br>1000: LVR=2.71V 1001: LVR=2.83V<br>1010: LVR=2.96V 1011: LVR=3.09V<br>1100: LVR=3.21V 1101: LVR=3.33V<br>1110: LVR=3.46V 1111: LVR=3.58V |
| | | 3~0 | LBDSEL | R/W | 0000 | Low Battery Detector select<br>0000: LBD=1.73V 0001: LBD=1.85V<br>0010: LBD=1.98V 0011: LBD=2.10V<br>0100: LBD=2.22V 0101: LBD=2.34V<br>0110: LBD=2.46V 0111: LBD=2.59V<br>1000: LBD=2.71V 1001: LBD=2.83V<br>1010: LBD=2.96V 1011: LBD=3.09V<br>1100: LBD=3.21V 1101: LBD=3.33V<br>1110: LBD=3.46V 1111: LBD=3.58V |
| C3h | **ADDTL** | 7 | LBDO | R | – | Low Battery Detector flag, 1=$V_{BAT}$ < LBDSEL's setting voltage |
| | | 4 | ADEOC | R | – | ADC end of conversion. 1=end |
| | | 3~0 | ADCDTL | R | – | ADC data bit 3~0 |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| C4h | **ADCON** | 7~6 | ADCKS | R/W | 00 | ADC clock rate select<br>00: $F_{SYSCLK}$/32 10: $F_{SYSCLK}$/8<br>01: $F_{SYSCLK}$/16 11: $F_{SYSCLK}$/4 |
| | | 5 | ADCSRV | R/W | 0 | ADC reference voltage select<br>0: $V_{BAT}$; 1: 2.4V |
| | | 4~0 | ADCHS | R/W | 1Fh | ADC channel select<br>00000: AD0 (P1.0) 00111: AD7 (P1.7)<br>00001: AD1 (P1.1) 01000: AD8 (P3.4)<br>00010: AD2 (P1.2) 01001: AD9 (P3.5)<br>00011: AD3 (P1.3) 01010: AD10 (P2.0)<br>00100: AD4 (P1.4) 01011: AD11 (P0.7)<br>00101: AD5 (P1.5) 01100: VBG (ADCSRV=0)<br>00110: AD6 (P1.6) 10111: $V_{BAT}$ /4 |
| C5h | **ADDTH** | 7~0 | ADDTH | R | – | ADC data bit 11~4 |
| C6h | **XBAUD** | 7 | XBAUDS | R/W | 0 | select UART extra baud rate generator<br>0: Baud rate uses Timer1/Timer2 overflow<br>1: Baud rate uses BAUDRT |
| | | 6~0 | BAUDRT | R/W | 30h | Extra baud rate |
| C7h | **EFTCON** | 7 | EFT2CS | R/W | 0 | EFT Detector1 enable control. 1=Enable |
| | | 6 | EFT1CS | R/W | 0 | EFT Detector2 enable control. 1=Enable |
| | | 5~4 | EFT1S | R/W | 00 | EFT Detector1 sensitivity adjust. 3=Highest |
| | | 3 | EFTSLOW | R/W | 0 | Force SYSCLK to SLOWCLK while EFT detected. 1=Enable |
| | | 2 | FRCJMPE | R/W | 0 | FRC spread frequency control. 1=Enable |
| | | 1 | FRCJMPS | R/W | 0 | FRC spread frequency scale select: 0=1%, 1=2% |
| | | 0 | CKHLDE | R/W | 0 | SYSCLK clock hold while EFT detected. 1=Enable |
| C8h | **T2CON** | 7 | TF2 | R/W | 0 | Timer2 overflow flag<br>Set by H/W when Timer/Counter 2 overflows unless RCLK=1 or TCLK=1. This bit must be cleared by S/W. |
| | | 6 | EXF2 | R/W | 0 | T2EX interrupt pin falling edge flag<br>Set when a capture or a reload is caused by a negative transition on T2EX pin if EXEN2=1. This bit must be cleared by S/W. |
| | | 5 | RCLK | R/W | 0 | UART receive clock control bit<br>0: Use Timer1 overflow as receive clock for serial port in mode 1 or 3<br>1: Use Timer2 overflow as receive clock for serial port in mode 1 or 3 |
| | | 4 | TCLK | R/W | 0 | UART transmit clock control bit<br>0: Use Timer1 overflow as transmit clock for serial port in mode 1 or 3<br>1: Use Timer2 overflow as transmit clock for serial port in mode 1 or 3 |
| | | 3 | EXEN2 | R/W | 0 | T2EX pin enable<br>0: T2EX pin disable<br>1: T2EX pin enable, it cause a capture or reload when a negative transition on T2EX pin is detected if RCLK=TCLK=0 |
| | | 2 | TR2 | R/W | 0 | Timer2 run control. 1:timer runs; 0:timer stops |
| | | 1 | CT2N | R/W | 0 | Timer2 Counter/Timer select bit<br>0: Timer mode, Timer2 data increases at 2 System clock cycle rate<br>1: Counter mode, Timer2 increases by T2 pin or Slow clock event |
| | | 0 | CPRL2N | R/W | 0 | Timer2 Capture/Reload control bit<br>0: Reload mode, auto-reload on Timer2 overflows or negative transitions on T2EX pin if EXEN2=1.<br>1: Capture mode, capture on negative transitions on T2EX pin if EXEN2=1. If RCLK=1 or TCLK=1, CPRL2N is ignored and timer is forced to auto-reload on Timer2 overflow. |
| C9h | **IAPWE** | 7~0 | IAPWE | W | – | Write 47h to set MTPWE control flag; Write E2h to set EEPWE control flag; Write other value to clear MTPWE and EEPWE flag. |
| | | 7 | MTPWE | R | 0 | Flag indicates MTP memory can be written by IAP or not, 1=Enable. |
| | | 6 | IAPTO | R | 0 | MTP (or EEPROM) write Time-Out flag, Set by H/W when MTP (or EEPROM) write Time-out occurs. Cleared by H/W when MTPWE=0 (or EEPWE=0). |
| | | 5 | EEPWE | R | 0 | Flag indicates EEPROM memory can be written or not, 1=Enable. |
| CAh | **RCP2L** | 7~0 | RCP2L | R/W | 00h | Timer2 reload/capture data low byte |
| CBh | **RCP2H** | 7~0 | RCP2H | R/W | 00h | Timer2 reload/capture data high byte |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| CCh | **TL2** | 7~0 | TL2 | R/W | 00h | Timer2 data low byte |
| CDh | **TH2** | 7~0 | TH2 | R/W | 00h | Timer2 data high byte |
| CEh | **PWMCON** | 7 | PWM1CKS | R/W | 0 | PWM1~4 clock source select<br>0: Slow clock<br>1: Fast clock |
| | | 6~4 | PWM1PSC | R/W | 000 | PWM1~4 clock prescaler<br>000: PWM clock is Slow/Fast clock divided by 128<br>001: PWM clock is Slow/Fast clock divided by 64<br>010: PWM clock is Slow/Fast clock divided by 32<br>011: PWM clock is Slow/Fast clock divided by 16<br>100: PWM clock is Slow/Fast clock divided by 8<br>101: PWM clock is Slow/Fast clock divided by 4<br>110: PWM clock is Slow/Fast clock divided by 2<br>111: PWM clock is Slow/Fast clock divided by 1 |
| | | 3 | PWM0CKS | R/W | 0 | PWM0 clock source select<br>0: Slow clock<br>1: Fast clock |
| | | 2~0 | PWM0PSC | R/W | 000 | PWM0 clock prescaler<br>000: PWM clock is Slow/Fast clock divided by 128<br>001: PWM clock is Slow/Fast clock divided by 64<br>010: PWM clock is Slow/Fast clock divided by 32<br>011: PWM clock is Slow/Fast clock divided by 16<br>100: PWM clock is Slow/Fast clock divided by 8<br>101: PWM clock is Slow/Fast clock divided by 4<br>110: PWM clock is Slow/Fast clock divided by 2<br>111: PWM clock is Slow/Fast clock divided by 1 |
| CFh | **PWMCON2** | 7 | PWRSAV2 | R/W | 0 | Power saving mode control<br>0: No power saving<br>1: Reduce Slow mode current consumption |
| | | 6 | PWM5CLR | R/W | 1 | Clear PWM5 Period counter<br>0: PWM5 run<br>1: PWM5 clear and hold |
| | | 5 | PWM0VX2 | R/W | 0 | PWM0P / PWM0N pump drive select<br>0: PWM0P/N normal drive<br>1: PWM0P/N pump drive (high level = $V_{BAT}$ * 2, need LCD pump) |
| | | 4 | PWM1SNK | R/W | 0 | PWM1 high sink select<br>0: PWM1 normal sink<br>1: PWM1 high sink (300mA) |
| | | 3 | PWM5CKS | R/W | 0 | PWM5 clock source select<br>0: Slow clock<br>1: Fast clock |
| | | 2~0 | PWM5PSC | R/W | 000 | PWM5 clock prescaler<br>000: PWM clock is Slow/Fast clock divided by 128<br>001: PWM clock is Slow/Fast clock divided by 64<br>010: PWM clock is Slow/Fast clock divided by 32<br>011: PWM clock is Slow/Fast clock divided by 16<br>100: PWM clock is Slow/Fast clock divided by 8<br>101: PWM clock is Slow/Fast clock divided by 4<br>110: PWM clock is Slow/Fast clock divided by 2<br>111: PWM clock is Slow/Fast clock divided by 1 |
| D0h | **PSW** | 7 | CY | R/W | 0 | ALU carry flag |
| | | 6 | AC | R/W | 0 | ALU auxiliary carry flag |
| | | 5 | F0 | R/W | 0 | General purpose user-definable flag |
| | | 4 | RS1 | R/W | 0 | Register Bank Select bit 1 |
| | | 3 | RS0 | R/W | 0 | Register Bank Select bit 0 |
| | | 2 | OV | R/W | 0 | ALU overflow flag |
| | | 1 | F1 | R/W | 0 | General purpose user-definable flag |
| | | 0 | P | R/W | 0 | Parity flag |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| D2h | **WDTCON** | 7 | IAPHTW | R/W | 0 | IAP high temperature write enable. 1=Enable |
| | | 6~4 | TM3PSC | R/W | 001 | Timer3 Interrupt rate<br>000: Timer3 interrupt occurs when 23 bit count data overflow<br>001: Timer3 interrupt rate is 32768 Slow clock cycles (1.0S for SXT)<br>010: Timer3 interrupt rate is 16384 Slow clock cycles (0.5S for SXT)<br>011: Timer3 interrupt rate is 8192 Slow clock cycles (0.25S for SXT)<br>100: Timer3 interrupt rate is 4096 Slow clock cycles (0.125S for SXT)<br>101: Timer3 interrupt rate is 2048 Slow clock cycles (62.5ms for SXT)<br>110: Timer3 interrupt rate is 1024 Slow clock cycles (31.2ms for SXT)<br>111: Timer3 interrupt rate is 512 Slow clock cycles (15.6ms for SXT) |
| | | 3~2 | WDTMOD | R/W | 00 | WDT control<br>00: WDT disable<br>01: WDT disable in Halt / Stop mode, enable in Idle / Slow / Fast mode<br>10: WDT disable in Idle / Halt / Stop mode, enable in Slow / Fast mode<br>11: WDT disable in Stop mode, enable in Halt / Idle / Slow / Fast mode |
| | | 1~0 | WDTPSC | R/W | 00 | WDT pre-scalar time select<br>00: WDT overflow is 2048 Slow clock cycle (64ms @SXT=32K)<br>01: WDT overflow is 4096 Slow clock cycle (128ms @SXT=32K)<br>10: WDT overflow is 8192 Slow clock cycle (256ms @SXT=32K)<br>11: WDT overflow is 16384 Slow clock cycle (512ms @SXT=32K) |
| D3h | **AUX2** | 7 | P07ADC | R/W | 0 | P0.7 ADC pin select. 1=Select P0.7 as ADC input |
| | | 6 | P20ADC | R/W | 0 | P2.0 ADC pin select. 1=Select P2.0 as ADC input |
| | | 5 | P02TCO | R/W | 0 | P0.2 TCO pin select. 1=Select P0.2 as $F_{SYSCLK}/2$ output |
| | | 4 | LBDEDGE | R/W | 0 | LBDIF trigger condition<br>0: LBDIF trigger by LBDO's rising edge. (when $V_{BAT}$ falling)<br>1: LBDIF trigger by LBDO's falling edge. (when $V_{BAT}$ rising) |
| | | 3 | VBGE | R/W | 0 | Force VBG enable. 1=Enable |
| | | 2 | VBGOUT | R/W | 0 | P1.1 VBG pin select. 1=Select P1.1 as VBG output |
| | | 1~0 | IAPTE | R/W | 11 | MTP (or EEPROM) write time-out enable.<br>00: Disable<br>01: wait 1ms to trigger time-out flag, and escape the write fail state<br>10: wait 4ms to trigger time-out flag, and escape the write fail state<br>11: wait 8ms to trigger time-out flag, and escape the write fail state |
| D4h | **PWM4DTY** | 7~0 | PWM4DTY | R/W | 80h | PWM4 Duty, 0=0 PWMCLK, 80h=128 PWMCLK, FFh=255 PWMCLK |
| D5h | **PWM5DTY** | 7~0 | PWM5DTY | R/W | 80h | PWM5 Duty, 0=0 PWMCLK, 80h=128 PWMCLK, FFh=255 PWMCLK |
| D6h | **PWM5PRD** | 7~0 | PWM5PRD | R/W | FFh | PWM5 Period, FFh=256 PWMCLK, 7Fh=128 PWMCLK |
| D7h | **PWMOE** | 7 | PWM5OE | R/W | 0 | PWM5 output to P1.3 |
| | | 6 | PWM4BOE | R/W | 0 | PWM4 output to P3.6 |
| | | 5 | PWM4AOE | R/W | 0 | PWM4 output to P1.1 |
| | | 4 | PWM3OE | R/W | 0 | PWM3 output to P1.0 |
| | | 3 | PWM2OE | R/W | 0 | PWM2 output to P3.4 |
| | | 2 | PWM1OE | R/W | 0 | PWM1 output to P3.5 |
| | | 1 | PWM0POE | R/W | 0 | PWM0P output to P3.7 |
| | | 0 | PWM0NOE | R/W | 0 | PWM0N output to P3.6 |
| D8h | **CLKCON** | 7~6 | FCKTYPE | R/W | 00 | Fast clock type select, can be changed only in Slow mode (SELFCK=0)<br>00: Fast clock is FRC<br>10: Fast clock is MRC<br>11: Fast clock is RFC, S/W must setup RFC circuitry before this setting. |
| | | 5 | SELFCK | R/W | 0 | System clock select. This bit can be changed only when STPFSUB=0 or FCKTYPE=3.<br>0: Slow clock (SRC/SXT)<br>1: Fast clock (FRC/MRC/RFC) |
| | | 4 | SCKTYPE | R/W | 0 | Slow clock Type. This bit can be changed only in Fast mode (SELFCK=1).<br>0: SRC<br>1: SXT, P0.7 and P2.0 are crystal oscillator pins |
| | | 3 | STPFSUB | R/W | 0 | FRC/MRC clock stop control. This bit can be changed only when SELFCK=0 or FCKTYPE=3.<br>0: FRC/MRC clock running<br>1: Stop FRC/MRC clock for power saving in Slow/Idle mode |
| | | 2 | STPSCK | R/W | 1 | Set 1 to stop Slow clock after PD=1 (Halt / Stop mode entry control) |

| Adr | SFR | Bit# | Bit Name | R/W | Rst | Description |
|---|---|---|---|---|---|---|
| | | 1~0 | CLKPSC | R/W | 11 | System clock prescaler, max effective delay is 16 cycles.<br>00: System clock is Fast/Slow clock divided by 16<br>01: System clock is Fast/Slow clock divided by 4<br>10: System clock is Fast/Slow clock divided by 2<br>11: System clock is Fast/Slow clock divided by 1 |
| E0h | **ACC** | 7~0 | ACC | R/W | 00h | Accumulator |
| E1h | **MICON** | 7 | MIEN | R/W | 0 | Master $I^2C$ enable control. 1=Enable |
| | | 6 | MIACKO | R/W | 0 | When Master $I^2C$ receive data, send acknowledge to $I^2C$ Bus<br>0: ACK to slave device<br>1: NACK to slave device |
| | | 5 | MIIF | R/W | 0 | Master $I^2C$ Interrupt flag<br>0: write 0 to clear it<br>1: Master I2C transfer one byte complete |
| | | 4 | MIACKI | R | 0 | When Master $I^2C$ transfer, acknowledgement form $I^2C$ bus (read only)<br>0: ACK received<br>1: NACK received |
| | | 3 | MISTART | R/W | 0 | Master $I^2C$ Start bit. 1=Start $I^2C$ bus transfer |
| | | 2 | MISTOP | R/W | 1 | Master $I^2C$ Stop bit. 1=Send STOP signal to stop $I^2C$ bus |
| | | 1~0 | MICR | R/W | 00 | Master $I^2C$ (MSCL) clock frequency selection<br>00: $F_{SYSCLK}/4$<br>01: $F_{SYSCLK}/16$<br>10: $F_{SYSCLK}/64$<br>11: $F_{SYSCLK}/256$ |
| E2h | **MIDAT** | 7~0 | MIDAT | W | – | Master $I^2C$ data shift register. After Start and before Stop condition, write this register will resume transmission to $I^2C$ bus |
| | | | | R | – | Master $I^2C$ data shift register. After Start and before Stop condition, read this register will resume receiving from $I^2C$ bus |
| E8h | **P4** | 7~0 | P4 | R/W | FFh | Port 4 data |
| F0h | **B** | 7~0 | B | R/W | 00h | B register |
| F1h | **CRCDL** | 7~0 | CRCDL | R/W | FFh | 16-bit CRC checksum data bit 7~0 |
| F2h | **CRCDH** | 7~0 | CRCDH | R/W | FFh | 16-bit CRC checksum data bit 15~8 |
| F3h | **CRCIN** | 7~0 | CRCIN | W | – | CRC input data register |
| F5h | **CFGVBG** | 4~0 | VBGTRIM | R/W | – | VBG adjustment. |
| F7h | **CFGFRC** | 6~0 | FRCF | R/W | – | FRC frequency adjustment. |
| F8h | **AUX1** | 5 | TKSOC | R/W | 0 | Rising edge of this bit will trigger a Touch Key conversion. |
| | | 4 | ADSOC | R/W | 0 | Rising edge of this bit will trigger an ADC conversion. |
| | | 3 | CLRWDT | R/W | 0 | Set to 1 to clear Watch Dog Timer |
| | | 2 | CLRTM3 | R/W | 0 | Set 1 to Clear Timer3 and force TM3SEC reload |
| | | 1 | STPRFC | R/W | 0 | Set 1 to stop RFC clock oscillating |
| | | 0 | DPSEL | R/W | 0 | Active DPTR Select |

| Adr | MTP | Bit# | Bit Name | Description |
|---|---|---|---|---|
| 3FFBh | **CFGVBG** | 4~0 | VBGTRIM | VBG adjustment. |
| 3FFDh | **CFGFRC** | 6~0 | FRCF | FRC frequency adjustment. |
| 3FFFh | **CFGWH** | 7 | PROT | MTP Code Protect, 1=Protect |
| | | 6 | XRSTE | Pin Reset enable, 1=enable. |
| | | 2 | AGMOD | Power on reset level select.<br>0: POR is 1.7V<br>1: POR is 1.1V |
| | | 1 | IAPHVS | MTP write time control. 1=Shorten write time |

# INSTRUCTION SET

Instructions are 1, 2 or 3 bytes long as listed in the 'byte' column below. Each instruction takes 1~8 System clock cycles to execute as listed in the 'cycle' column below.

| ARITHMETIC | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **byte** | **cycle** | **opcode** |
| ADD A,Rn | Add register to A | 1 | 2 | 28-2F |
| ADD A,dir | Add direct byte to A | 2 | 2 | 25 |
| ADD A,@Ri | Add indirect memory to A | 1 | 2 | 26-27 |
| ADD A,#data | Add immediate to A | 2 | 2 | 24 |
| ADDC A,Rn | Add register to A with carry | 1 | 2 | 38-3F |
| ADDC A,dir | Add direct byte to A with carry | 2 | 2 | 35 |
| ADDC A,@Ri | Add indirect memory to A with carry | 1 | 2 | 36-37 |
| ADDC A,#data | Add immediate to A with carry | 2 | 2 | 34 |
| SUBB A,Rn | Subtract register from A with borrow | 1 | 2 | 98-9F |
| SUBB A,dir | Subtract direct byte from A with borrow | 2 | 2 | 95 |
| SUBB A,@Ri | Subtract indirect memory from A with borrow | 1 | 2 | 96-97 |
| SUBB A,#data | Subtract immediate from A with borrow | 2 | 2 | 94 |
| INC A | Increment A | 1 | 2 | 04 |
| INC Rn | Increment register | 1 | 2 | 08-0F |
| INC dir | Increment direct byte | 2 | 2 | 05 |
| INC @Ri | Increment indirect memory | 1 | 2 | 06-07 |
| DEC A | Decrement A | 1 | 2 | 14 |
| DEC Rn | Decrement register | 1 | 2 | 18-1F |
| DEC dir | Decrement direct byte | 2 | 2 | 15 |
| DEC @Ri | Decrement indirect memory | 1 | 2 | 16-17 |
| INC DPTR | Increment data pointer | 1 | 4 | A3 |
| MUL AB | Multiply A by B | 1 | 8 | A4 |
| DIV AB | Divide A by B | 1 | 8 | 84 |
| DA A | Decimal Adjust A | 1 | 2 | D4 |

| LOGICAL | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **byte** | **cycle** | **opcode** |
| ANL A,Rn | AND register to A | 1 | 2 | 58-5F |
| ANL A,dir | AND direct byte to A | 2 | 2 | 55 |
| ANL A,@Ri | AND indirect memory to A | 1 | 2 | 56-57 |
| ANL A,#data | AND immediate to A | 2 | 2 | 54 |
| ANL dir,A | AND A to direct byte | 2 | 2 | 52 |
| ANL dir,#data | AND immediate to direct byte | 3 | 4 | 53 |
| ORL A,Rn | OR register to A | 1 | 2 | 48-4F |
| ORL A,dir | OR direct byte to A | 2 | 2 | 45 |
| ORL A,@Ri | OR indirect memory to A | 1 | 2 | 46-47 |
| ORL A,#data | OR immediate to A | 2 | 2 | 44 |
| ORL dir,A | OR A to direct byte | 2 | 2 | 42 |
| ORL dir,#data | OR immediate to direct byte | 3 | 4 | 43 |
| XRL A,Rn | Exclusive-OR register to A | 1 | 2 | 68-6F |
| XRL A,dir | Exclusive-OR direct byte to A | 2 | 2 | 65 |
| XRL A, @Ri | Exclusive-OR indirect memory to A | 1 | 2 | 66-67 |
| XRL A,#data | Exclusive-OR immediate to A | 2 | 2 | 64 |
| XRL dir,A | Exclusive-OR A to direct byte | 2 | 2 | 62 |
| XRL dir,#data | Exclusive-OR immediate to direct byte | 3 | 4 | 63 |
| CLR A | Clear A | 1 | 2 | E4 |
| CPL A | Complement A | 1 | 2 | F4 |
| SWAP A | Swap Nibbles of A | 1 | 2 | C4 |

| LOGICAL | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **byte** | **cycle** | **opcode** |
| RL A | Rotate A left | 1 | 2 | 23 |
| RLC A | Rotate A left through carry | 1 | 2 | 33 |
| RR A | Rotate A right | 1 | 2 | 03 |
| RRC A | Rotate A right through carry | 1 | 2 | 13 |

| DATA TRANSFER | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **byte** | **cycle** | **opcode** |
| MOV A,Rn | Move register to A | 1 | 2 | E8-EF |
| MOV A,dir | Move direct byte to A | 2 | 2 | E5 |
| MOV A,@Ri | Move indirect memory to A | 1 | 2 | E6-E7 |
| MOV A,#data | Move immediate to A | 2 | 2 | 74 |
| MOV Rn,A | Move A to register | 1 | 2 | F8-FF |
| MOV Rn,dir | Move direct byte to register | 2 | 4 | A8-AF |
| MOV Rn,#data | Move immediate to register | 2 | 2 | 78-7F |
| MOV dir,A | Move A to direct byte | 2 | 2 | F5 |
| MOV dir,Rn | Move register to direct byte | 2 | 4 | 88-8F |
| MOV dir,dir | Move direct byte to direct byte | 3 | 4 | 85 |
| MOV dir,@Ri | Move indirect memory to direct byte | 2 | 4 | 86-87 |
| MOV dir,#data | Move immediate to direct byte | 3 | 4 | 75 |
| MOV @Ri,A | Move A to indirect memory | 1 | 2 | F6-F7 |
| MOV @Ri,dir | Move direct byte to indirect memory | 2 | 4 | A6-A7 |
| MOV @Ri,#data | Move immediate to indirect memory | 2 | 2 | 76-77 |
| MOV DPTR,#data | Move immediate to data pointer | 3 | 4 | 90 |
| MOVC A,@A+DPTR | Move code byte relative DPTR to A | 1 | 8 | 93 |
| MOVC A,@A+PC | Move code byte relative PC to A | 1 | 8 | 83 |
| MOVX A,@Ri | Move external data(A8) to A | 1 | 8 | E2-E3 |
| MOVX A,@DPTR | Move external data(A16) to A | 1 | 8 | E0 |
| MOVX @Ri,A | Move A to external data(A8) | 1 | 8 | F2-F3 |
| MOVX @DPTR,A | Move A to external data(A16) | 1 | 8 | F0 |
| PUSH dir | Push direct byte onto stack | 2 | 4 | C0 |
| POP dir | Pop direct byte from stack | 2 | 4 | D0 |
| XCH A,Rn | Exchange A and register | 1 | 2 | C8-CF |
| XCH A,dir | Exchange A and direct byte | 2 | 2 | C5 |
| XCH A,@Ri | Exchange A and indirect memory | 1 | 2 | C6-C7 |
| XCHD A,@Ri | Exchange A and indirect memory nibble | 1 | 2 | D6-D7 |

| BOOLEAN | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **byte** | **cycle** | **opcode** |
| CLR C | Clear carry | 1 | 2 | C3 |
| CLR bit | Clear direct bit | 2 | 2 | C2 |
| SETB C | Set carry | 1 | 2 | D3 |
| SETB bit | Set direct bit | 2 | 2 | D2 |
| CPL C | Complement carry | 1 | 2 | B3 |
| CPL bit | Complement direct bit | 2 | 2 | B2 |
| ANL C,bit | AND direct bit to carry | 2 | 4 | 82 |
| ANL C,/bit | AND direct bit inverse to carry | 2 | 4 | B0 |
| ORL C,bit | OR direct bit to carry | 2 | 4 | 72 |
| ORL C,/bit | OR direct bit inverse to carry | 2 | 4 | A0 |
| MOV C,bit | Move direct bit to carry | 2 | 2 | A2 |
| MOV bit,C | Move carry to direct bit | 2 | 4 | 92 |

| BRANCHING | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **byte** | **cycle** | **opcode** |
| ACALL addr 11 | Absolute jump to subroutine | 2 | 6 | 11-F1 |
| LCALL addr 16 | Long jump to subroutine | 3 | 6 | 12 |
| RET | Return from subroutine | 1 | 6 | 22 |
| RETI | Return from interrupt | 1 | 6 | 32 |
| AJMP addr 11 | Absolute jump unconditional | 2 | 6 | 01-E1 |
| LJMP addr 16 | Long jump unconditional | 3 | 6 | 02 |
| SJMP rel | Short jump (relative address) | 2 | 6 | 80 |
| JC rel | Jump on carry=1 | 2 | 4 or 6 | 40 |
| JNC rel | Jump on carry=0 | 2 | 4 or 6 | 50 |
| JB bit,rel | Jump on direct bit=1 | 3 | 4 or 6 | 20 |
| JNB bit,rel | Jump on direct bit=0 | 3 | 4 or 6 | 30 |
| JBC bit,rel | Jump on direct bit=1 and clear | 3 | 4 or 6 | 10 |
| JMP @A+DPTR | Jump indirect relative DPTR | 1 | 6 | 73 |
| JZ rel | Jump on accumulator=0 | 2 | 4 or 6 | 60 |
| JNZ rel | Jump on accumulator≠0 | 2 | 4 or 6 | 70 |
| CJNE A,dir,rel | Compare A,direct, jump not equal relative | 3 | 4 or 6 | B5 |
| CJNE A,#data,rel | Compare A,immediate, jump not equal relative | 3 | 4 or 6 | B4 |
| CJNE Rn,#data,rel | Compare register,immediate, jump not equal relative | 3 | 4 or 6 | B8-BF |
| CJNE @Ri,#data,rel | Compare indirect,immediate, jump not equal relative | 3 | 4 or 6 | B6-B7 |
| DJNZ Rn,rel | Decrement register, jump not zero relative | 2 | 4 or 6 | D8-DF |
| DJNZ dir,rel | Decrement direct byte, jump not zero relative | 3 | 4 or 6 | D5 |

| MISCELLANEOUS | | | | |
|---|---|---|---|---|
| **Mnemonic** | **Description** | **byte** | **cycle** | **opcode** |
| NOP | No operation | 1 | 2 | 00 |

In the above table, an entry such as E8-EF indicates a continuous block of hex opcodes used for 8 different registers, the register numbers of which are defined by the lowest three bits of the corresponding code. Non-continuous blocks of codes, shown as 11-F1 (for example), are used for absolute jumps and calls with the top 3 bits of the code being used to store the top three bits of the destination address.

# ELECTRICAL CHARACTERISTICS

## Absolute Maximum Ratings

| Parameter | Rating | Unit |
|---|---|---|
| Supply voltage | $V_{SS}$ – 0.3 ~ $V_{SS}$ + 5.5 | V |
| Input voltage | $V_{SS}$ – 0.3 ~ $V_{BAT}$ + 0.3 | |
| Output voltage | $V_{SS}$ – 0.3 ~ $V_{BAT}$ + 0.3 | |
| Maximum Operating Voltage | 5.5 | |
| Output current high all pins | –80 | mA |
| Output current low all pins | +320 | |
| Operating temperature | –40 ~ 105 | ℃ |
| Storage temperature | –65 ~ +150 | |

## DC Characteristics ($T_A$=25°C)

| Parameter | Sym | Conditions | | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|---|
| Input High Voltage | $V_{IH}$ | all Input | $V_{BAT}$=3~5V | 0.6$V_{BAT}$ | – | – | V |
| Input Low Voltage | $V_{IL}$ | | | – | – | 0.2$V_{BAT}$ | |
| I/O Port, all LED pins Source Current | $I_{OH}$ | $V_{OH}$=0.9$V_{BAT}$ | $V_{BAT}$=1.5V | – | 1.5 | – | mA |
| | | | $V_{BAT}$=3V | – | 7 | – | |
| | | | $V_{BAT}$=5V | – | 18 | – | |
| I/O Port, LED SEG Sink Current | $I_{OL}$ | $V_{OL}$=0.1$V_{BAT}$ | $V_{BAT}$=1.5V | – | 5 | – | |
| | | | $V_{BAT}$=3V | – | 23 | – | |
| | | | $V_{BAT}$=5V | – | 52 | – | |
| LED COM Sink Current | | | $V_{BAT}$=3V | – | 40 | – | |
| | | | $V_{BAT}$=5V | – | 80 | – | |
| PWM1 Hi-Sink Current | | | $V_{BAT}$=3V | – | 230 | – | |
| | | | $V_{BAT}$=5V | – | 450 | – | |
| Power Supply Current (PWRSAV=1, PWRSAV2=1) | $I_{BAT}$ | Fast, FRC, 14.7MHz | $V_{BAT}$=5V $V_{DD}$=5V | – | 7.2 | – | mA |
| | | Fast, MRC, 8MHz | | – | 4.5 | – | |
| | | Fast, FRC, 14.7MHz | $V_{BAT}$=5V $V_{DD}$=3.1V | – | 4.0 | – | |
| | | Fast, MRC, 6MHz | | – | 2.5 | – | |
| | | Fast, MRC, 2.3MHz | $V_{BAT}$=3V $V_{DD}$=1.5V | – | 400 | – | uA |
| | | Slow, SXT, 32KHz | | – | 38 | – | |
| | | Slow, SRC, 35KHz | | – | 39 | – | |
| | | Halt 32KHz, LCD On | | – | 3.2 | – | |
| | | Halt 32KHz, LCD Off | | – | 1.3 | – | |
| | | Halt 32KHz, LCD On | $V_{BAT}$=1.5V $V_{DD}$=1.5V | – | 4.8 | – | |
| | | Halt 32KHz, LCD Off | | – | 0.8 | – | |
| | | Stop, $V_{BAT}$=3V | $V_{DD}$=3V | – | 0.1 | – | |
| | | | $V_{DD}$=1.5V | – | 0.5 | – | |
| Pull-Up Resistor | $R_{PU}$ | all I/O | $V_{BAT}$=5V | – | 38 | – | KΩ |
| | | | $V_{BAT}$=3V | – | 65 | – | |

## Operation Voltage ($V_{DD}$) ($T_A$=25℃)

| System Clock | Sym | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| FRC | $V_{DD}$ | FRC=14.7456MHz | 2.2 | – | 5.5 | V |
| MRC, SRC, SXT | $V_{DD}$ | – | 1.4 | – | 5.5 | |

## ADC, BandGap & POR Characteristics

| Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| Total Accuracy | $V_{BAT}$=3V, $V_{SS}$=0V | – | ±2.5 | ±4 | LSB |
| Integral Non-Linearity | | – | ±3.2 | ±5 | |
| Max Input Clock ($f_{ADC}$)<br>Total Accuracy | Source impedance（Rs < 10K ohm） | – | – | 2 | MHz |
| | Source impedance（Rs < 20K ohm） | – | – | 1 | |
| | Source impedance（Rs < 50K ohm） | – | – | 0.5 | |
| | Source is $V_{BG}$（ADCHS=01100b） | – | – | 2 | |
| ADC Conversion time | $F_{ADC}$ = 1MHz | – | 42 | – | μs |
| ADC Conversion current | $V_{BAT}$=5V, ADCSRV=0 | | 0.45 | | mA |
| | $V_{BAT}$=4V, ADCSRV=1 | | 0.6 | | |
| BandGap Voltage Reference ($V_{BG}$) | -40°C ~105°C, $V_{BAT}$=2V~5.5V | -2.0% | 1.18 | +2.0% | V |
| ADC Vref (ADCSRV=1) | -40°C ~105°C, $V_{BAT}$=3V~5.5V | -2.0% | 2.395 | +2.0% | |
| ADC Input Voltage | – | $V_{SS}$ | – | $V_{BAT}$ | |
| POR Voltage ($V_{POR}$) | AGMOD=0, 25°C | 1.6 | 1.75 | 2.0 | V |
| | AGMOD=1, 25°C | 1.0 | 1.1 | 1.35 | |

## Clock Timing ($T_A$=25℃)

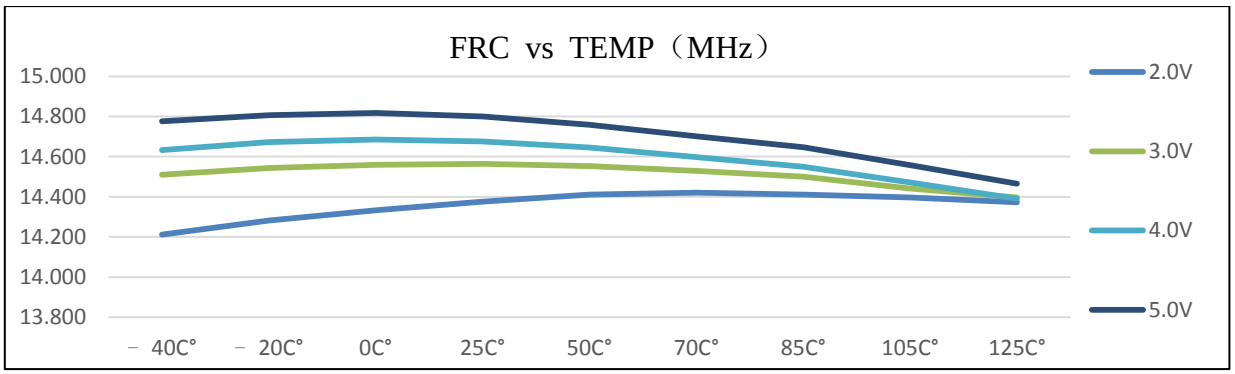
| Parameter | Sym | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| FRC Clock Frequency | $F_{FRC}$ | $V_{BAT}$=5V, 0℃~50℃ | –1% | 14.7456 | –1% | MHz |
| | | $V_{BAT}$=3V~5.5V, –40℃~105℃ | –3% | 14.7456 | +1.5% | |
| MRC Clock Frequency | $F_{MRC}$ | $V_{DD}$=3V | – | 6 | – | MHz |
| | | $V_{DD}$=1.5V | – | 2.3 | – | |
| SRC Clock Frequency | $F_{SRC}$ | $V_{DD}$=3V | – | 76 | – | KHz |
| | | $V_{DD}$=1.5V | – | 34 | – | |

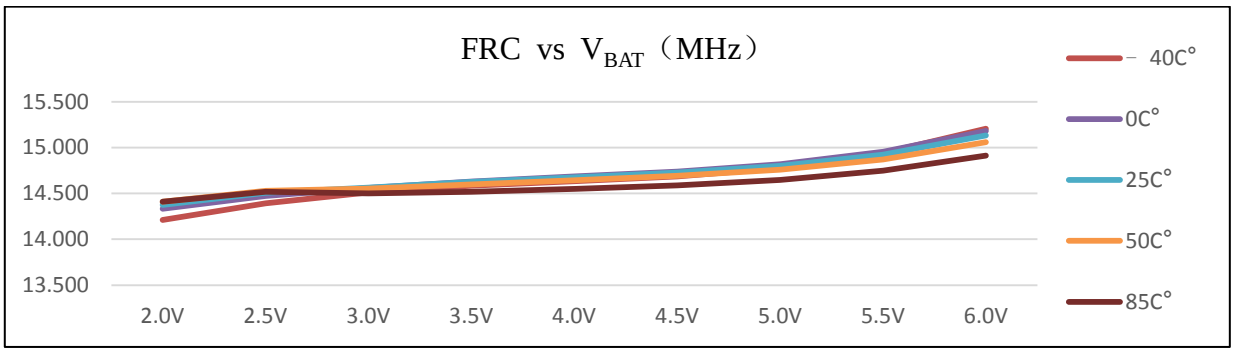
## EEPROM Characteristics

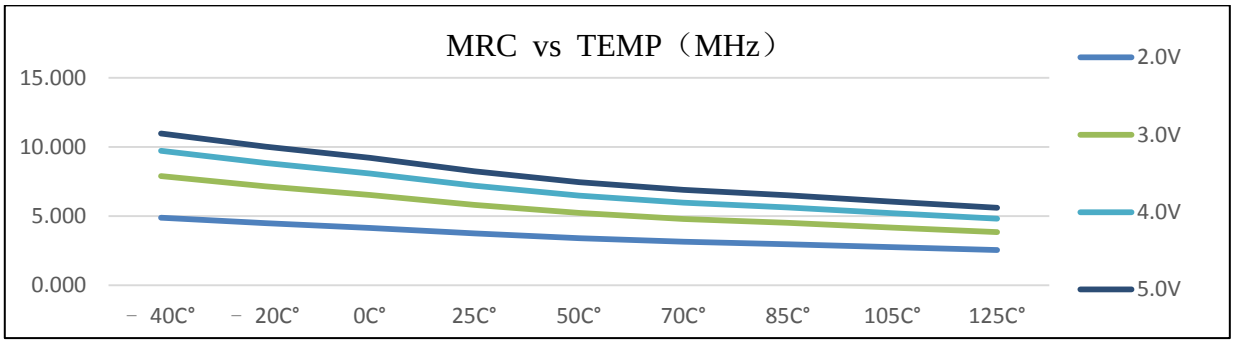
| Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| Write Voltage | –20°C ~ 105°C | 3.0 | 5 | 5.5 | V |
| | –40°C ~ 105°C | 3.5 | 5 | 5.5 | |
| Write Endurance | $V_{DD}$ = 4.0V~5.5V, –40°C~ 105°C | 20K | – | – | cycles |
| | $V_{DD}$ = 4.0V~5.5V, 0°C~ 105°C | 50K | – | – | |

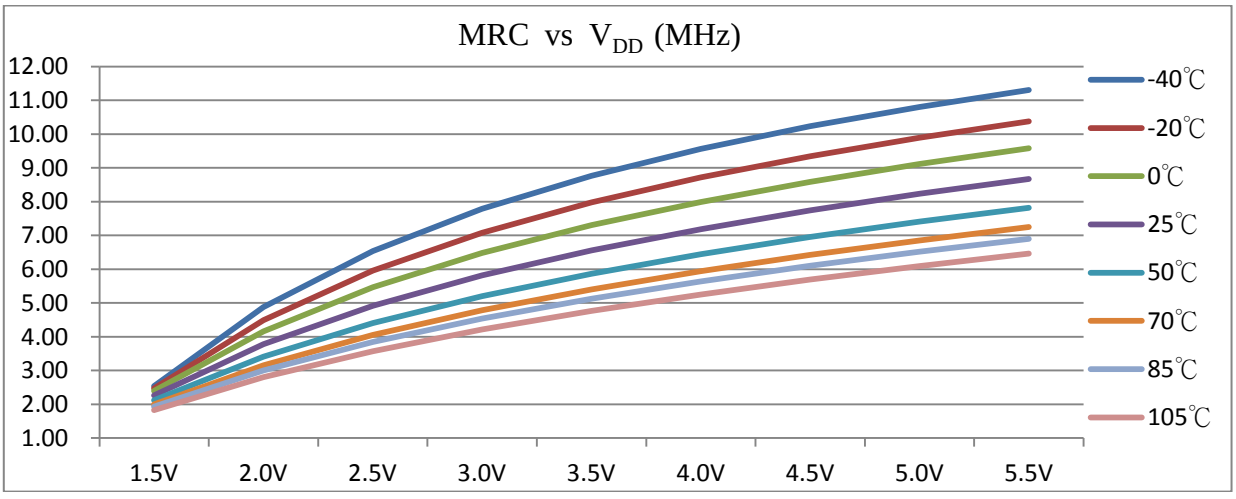
***Note:*** *The value of above parameter is based on the characteristics of tested samples.*

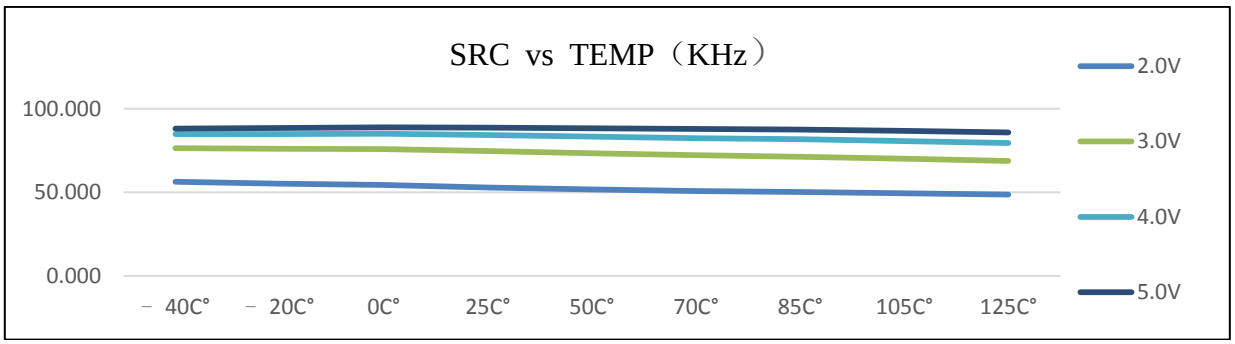
## Characteristic Graphs

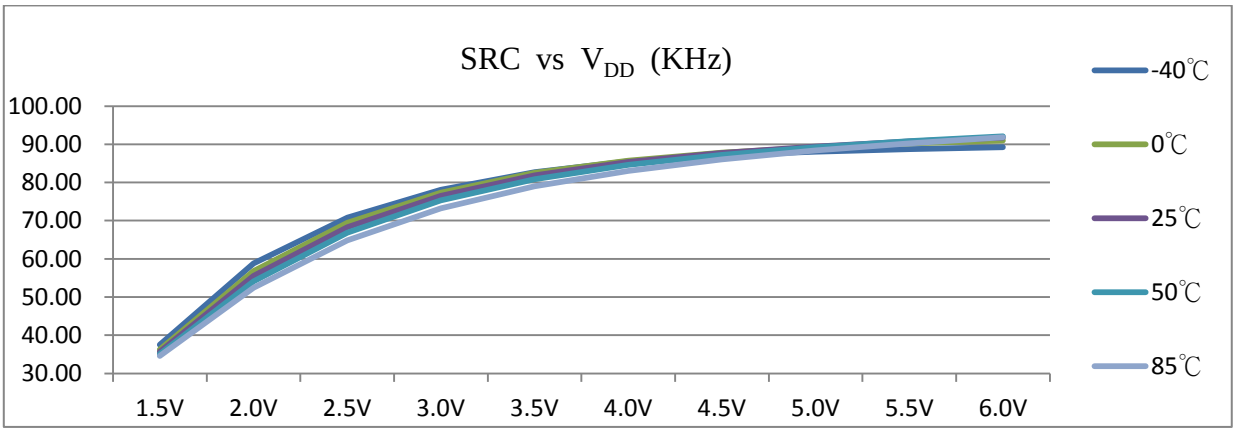


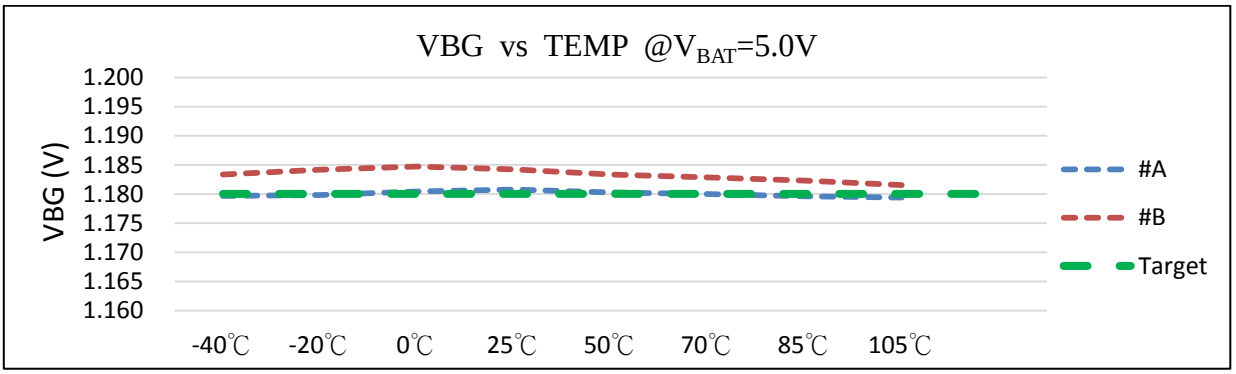



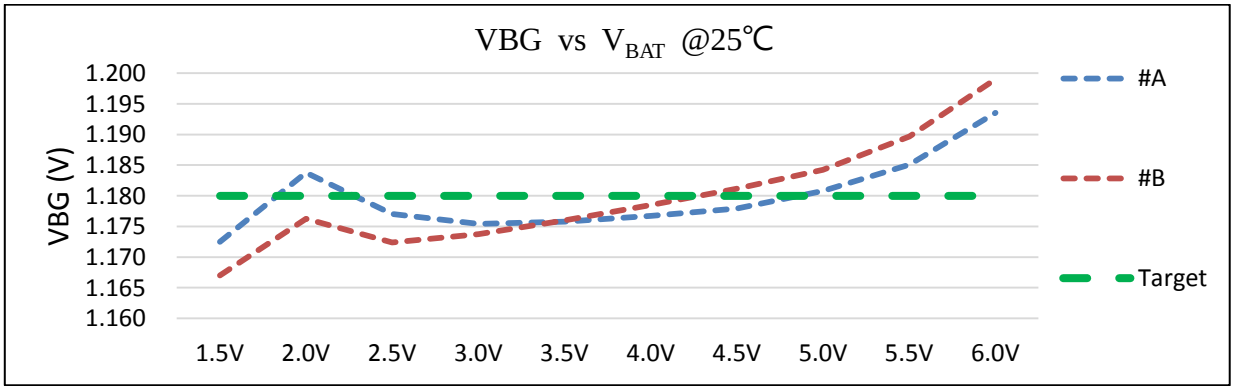



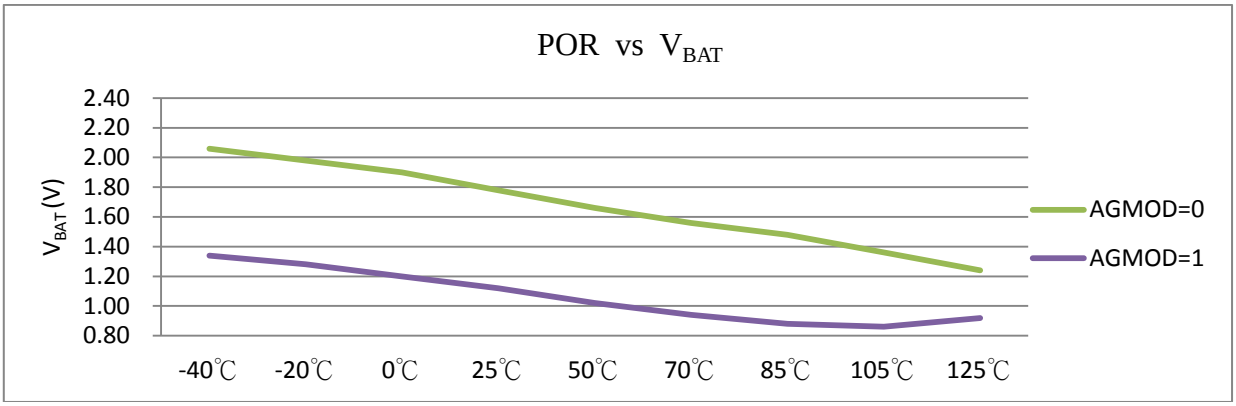


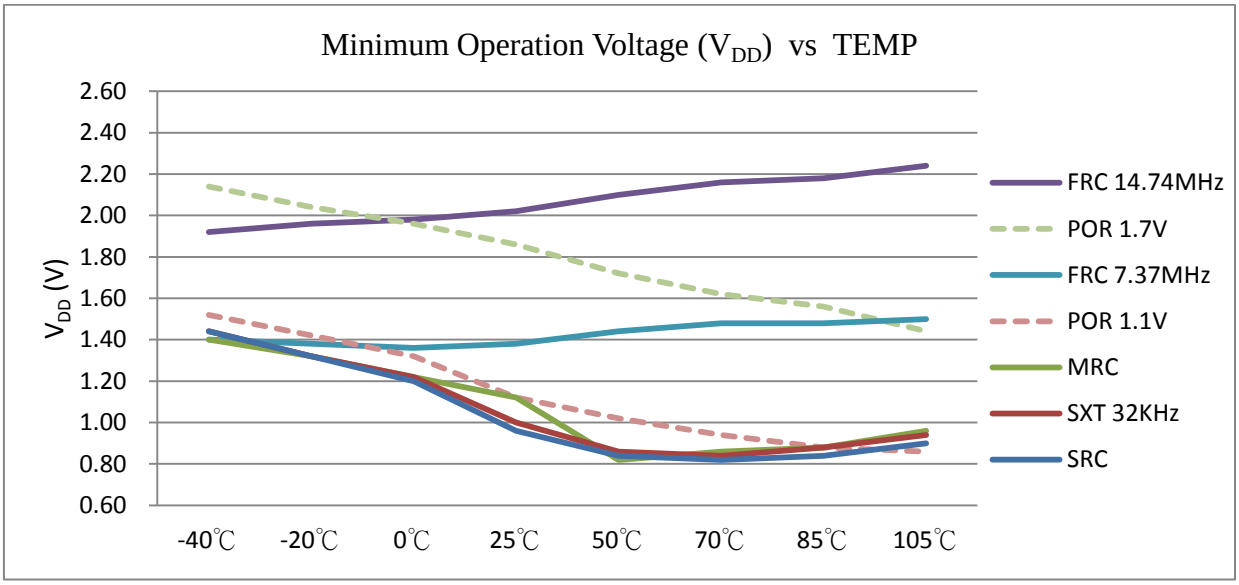




***Note**: Since LVR and LVD are derived from VBG, they have the same characteristic as VBG.*




***Note:*** *The value of above curve is based on the characteristics of tested samples. It does not mean all chips have the same characteristic.*

# PACKAGE INFORMATION

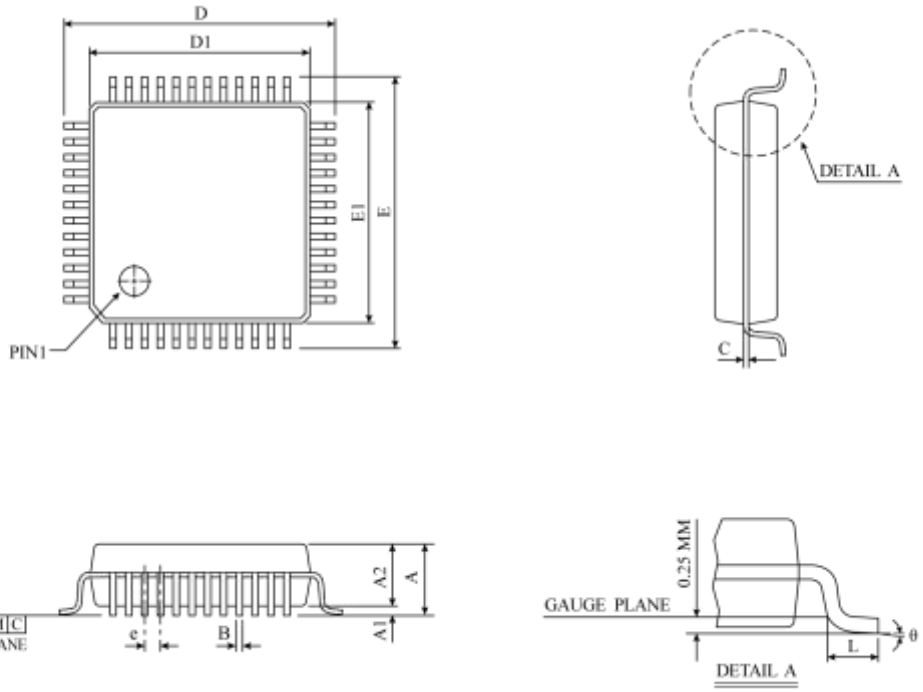
Please note that the package information provided is for reference only. Since this information is frequently updated, users can contact Sales to consult the latest package information and stocks.

## Ordering Information

| Ordering number | Package |
|---|---|
| TM52F2384-MTP | Wafer / Dice blank chip |
| TM52F2384-COD | Wafer / Dice with code |
| TM52F2384-MTP-72 | LQFP 48-pin ( 7x7mm ) |
| TM52F2384-MTP-73 | LQFP 64-pin (7x7 mm) |

## Package Information

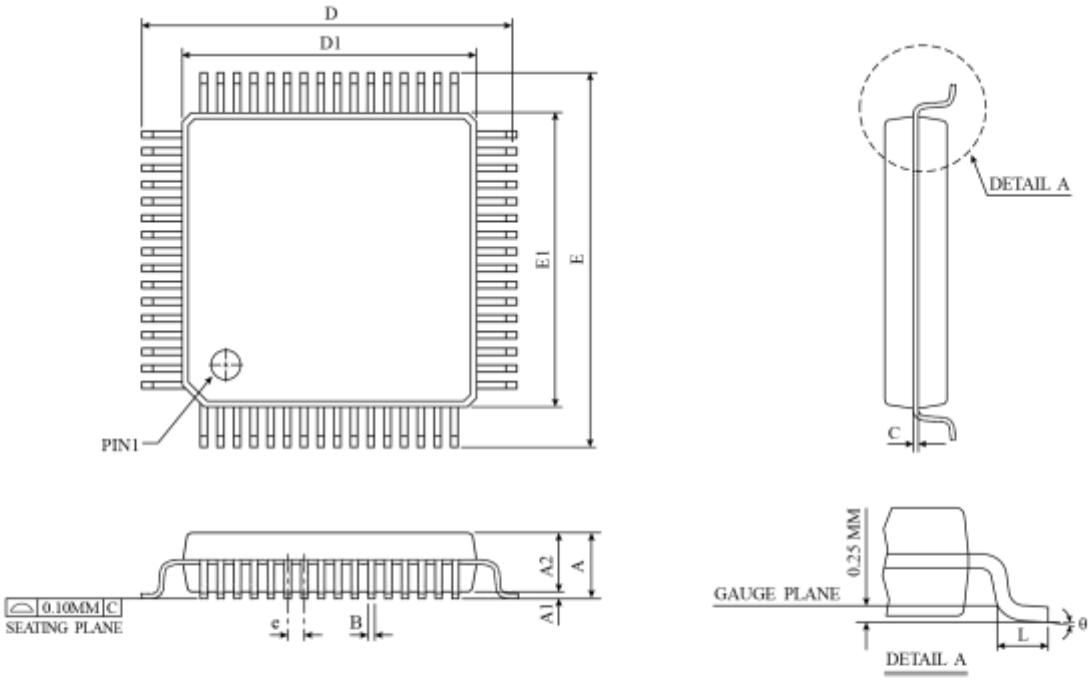
### LQFP-48 ( 7×7mm ) Package Dimension




| SYMBOL | DIMENSION IN MM | | | DIMENSION IN INCH | | |
|---|---|---|---|---|---|---|
| | MIN | NOM | MAX | MIN | NOM | MAX |
| A | - | - | 1.60 | - | - | 0.063 |
| A1 | 0.05 | 0.10 | 0.15 | 0.001 | 0.004 | 0.006 |
| A2 | 1.35 | 1.40 | 1.45 | 0.053 | 0.055 | 0.057 |
| B | 0.17 | 0.22 | 0.27 | 0.007 | 0.009 | 0.011 |
| C | 0.09 | 0.15 | 0.20 | 0.004 | 0.006 | 0.008 |
| D | 9.00 BSC | | | 0.354 BSC | | |
| D1 | 7.00 BSC | | | 0.276 BSC | | |
| E | 9.00 BSC | | | 0.354 BSC | | |
| E1 | 7.00 BSC | | | 0.276 BSC | | |
| e | 0.50 BSC | | | 0.020 BSC | | |
| L | 0.45 | 0.60 | 0.75 | 0.018 | 0.024 | 0.030 |
| θ | 0° | 3.5° | 7° | 0° | 3.5° | 7° |
| JEDEC | MS-026 (BBC) | | | | | |

△ * NOTES : DIMENSION " D1 " AND " E1 " DO NOT INCLUDE MOLD PROTRUSIONS. ALLOWABLE PROTRUSIONS IS 0.25 mm PER SIDE. " D1 " AND " E1 " ARE MAXIMUM PLASTIC BODY SIZE DIMENSIONS INCLUDING MOLD MISMACH.

## LQFP-64 ( 7×7mm ) Package Dimension

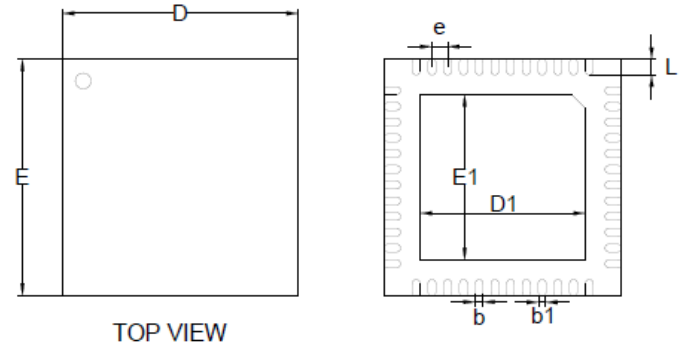



| SYMBOL | DIMENSION IN MM | | | DIMENSION IN INCH | | |
|---|---|---|---|---|---|---|
| | MIN | NOM | MAX | MIN | NOM | MAX |
| A | - | - | 1.60 | - | - | 0.063 |
| A1 | 0.05 | - | 0.15 | 0.002 | - | 0.006 |
| A2 | 1.35 | 1.40 | 1.45 | 0.053 | 0.055 | 0.057 |
| B | 0.13 | 0.18 | 0.23 | 0.005 | 0.007 | 0.009 |
| C | 0.09 | - | 0.20 | 0.004 | - | 0.008 |
| D | 9.00 BASIC | | | 0.354 BASIC | | |
| D1 | 7.00 BASIC | | | 0.276 BASIC | | |
| E | 9.00 BASIC | | | 0.354 BASIC | | |
| E1 | 7.00 BASIC | | | 0.276 BASIC | | |
| e | 0.40 BASIC | | | 0.016 BASIC | | |
| L | 0.45 | 0.60 | 0.75 | 0.018 | 0.024 | 0.030 |
| θ | 0° | 3.5° | 7° | 0° | 3.5° | 7° |
| JEDEC | MS-026 (BBD) | | | | | |

△ * NOTES：DIMENSION "D1" AND "E1" DO NOT INCLUDE MOLD PROTRUSIONS. ALLOWABLE PROTRUSIONS IS 0.25mm PER SIDE. "D1" AND "E1" ARE MAXIMUM PLASTIC BODY SIZE DIMENSIONS INCLUDING MOLD MISMACH.

**QFN-48 ( 6×6mm ) Package Dimension**

QFN6.0X6.0X0.55-48L产品外形图




| COMMON DIMENSIONS (UNITS OF MEASURE=MILLIMETER) | | | |
|---|---|---|---|
| SYMBOL | MILLMETER | | |
| | MIN | NOM | MAX |
| D | 5.95 | 6.00 | 6.05 |
| E | 5.95 | 6.00 | 6.05 |
| D1 | 4.15 | 4.20 | 4.25 |
| E1 | 4.15 | 4.20 | 4.25 |
| L | 0.35 | 0.40 | 0.45 |
| b | 0.15 | 0.20 | 0.25 |
| b1 | 0.13 | 0.18 | 0.23 |
| e | 0.40BSC | | |
| R | 0.05BSC | | |
| A1 | 0.152REF | | |
| A2 | 0.00 | 0.02 | 0.05 |
| A | 0.50 | 0.55 | 0.60 |
| PAD SIZE | 4.74x4.74 | | |

Jojon 志忠

深圳市志忠微电子有限公司

| | |
|---|---|
| 制作 | 周康 |
| 审核 | 李秋辉 |
| 客户确认 | |