Super High-speed 1T 8051 Core Flash MCU, 2 Kbytes SRAM, 32 Kbytes Flash ROM, 1 Kbytes Independent EEPROM, 31-channel TouchKey circuit,12-bit ADC, 1 Analog Comparator, LCD/LED Driver, 12-bit PWM, 3 Timers, MDU, UART, SSI, Check Sum Module

1 General Description

The SC92F8597/8596/8595/8593 (hereinafter referred to as the SC92F859X) is a series of enhanced 1T 8051 core industry-standard Flash Microcontroller unit (MCU) with integrated TouchKey function, the instruction set is compatible with the standard 8051 series.

The SC92F859X is integrated with 32 Kbytes Flash ROM, 256 bytes SRAM, external 1792 bytes RAM, 1 Kbytes independent EEPROM, 31-channel TouchKey circuit, up to 46 General-purpose I/Os (GPIO), 16 external interrupters, three 16-bit timers, 17-channel 12-bit high-precision ADC, 1 analog comparator, 4-channel 12-bit PWM, one 16 * 16-bit hardware Multiplier-Divide Unit (MDU), internal ±2% high-precision high-frequency 32MHz oscillator and ±4% precision low-frequency 32kHz oscillator, can be connected to external 32.768kHz crystal oscillators and 1 UART, 1 SSI (UART/TWI/SPI). To improve the reliability and simplify the circuit design, the SC92F859X is also built in with 4-level optional LVR voltage, 1.024V/ 2.4V/ 2.048V ADC reference voltage, low-power WDT and other high-reliability circuits.

The SC92F859X features excellent anti-interference performance, which make it possible to be widely applied to industrial control system, such as Internet of Things, intelligent home appliances, Charger, power supply, model airplane, interphone, wireless communication, gaming peripherals and consumer applications.

2 Features

Operating Conditions

- Voltage Range： 2.0V~5.5V
- Temperature Range： -40℃ ~ +105℃

EMS

- ESD
  - HBM：MIL-STD-883J Class 3A
  - MM：JEDEC EIA/JESD22-A115 Class C
- EFT
  - EN61000-4-4 Level 4

CPU

- 8-bit 1T 8051 core CMOS microcontroller, instruction set fully compatible with MCS-51

Flash ROM

- 32 Kbytes Flash ROM
  - The APROM area can be set to the range of IAP operation to 0K / 1K / 2K/all APROM through the Code Option setting item
- LDROM:
  - Used to store the user's BootLoader boot code
  - The LDROM can be set to 0K/1K/2K/4K through the Code Option setting item
- EEPROM:
  - Independent 1K bytes EEPROM
  - 100,000 times of writing, and more than 100 years of storage life at room temperature

SRAM:

- Internal 256 bytes
- 1792 Bytes Indirect Access RAM (XRAM)
- Additional 44 bytes PWM&LCD RAM

Unique ID:

- 96 bits Unique ID, which stores the unique identifier of an IC

System Clock（f_{SYS}）

- Built-in high frequency 32MHz oscillator (f_{HRC})
- The system clock frequency (fsys) of the IC can be selected and set by the programmer as: 32/16/8/2.66 MHz @2.0~5.5V
- Frequency Error: Across (2.0~5.5V) and (-20 ~ 85℃) application environment, the frequency error is not more than ±2%
- The system clock can be automatically calibrated by 32.768kHz external crystal oscillator, after calibration HRC accuracy can be infinitely close to the accuracy of external 32.768kHz crystal oscillator.

Built-in low-frequency crystal oscillator circuit:

- 32.768k oscillator can be connected externally as a Base Timer clock source, and wake up STOP

Built-in low-frequency 32kHz oscillator (LRC):

- Used as the clock source for Base Timer and WDT, and wake up STOP
- Frequency Error: Across (4.0~5.5V) and (-20 ~ 85℃) application environment, after the register correction frequency error is not more than ±4%

Low-voltage Reset (LVR)

- 4 options of reset voltage: 4.3/3.7/ 2.9/1.9V
- The default value can be selected by the Code Option

Flash Programming and Emulation

- 2-wire JTAG programming and emulation interface
- Jtag-specific mode and regular mode can be set through Code Option

Interrupts (INT)

- Up to 13 interrupt sources including Timer0~Timer2, INT0~2, ADC, PWM, UART, SSI, Base Timer, TK, CMP
- External interrupt contains 3 interrupt vectors, 16 interrupt ports. All can set up rising edge, falling edge, dual edge interrupt.
- Two-level interrupt priority capability

Digital Peripheral:

- Up to 46 bidirectional independently controllable I/O ports
  - The pull-up resistors can be set independently
  - All I/Os have large sink current drive capability (50mA)
  - P0~P3L (P3.0/1/2/3) port source drive capability is divided into four levels
- Built-in WDT, optional clock frequency division ratio
- 3 standard 80C51 timers： Timer0, Timer1, and Timer2. Timer2 provides the Capture function
- 4-channel common cycle and duty cycle independent adjustable 12-bit PWM
- 1 independent UART communication port
- 1 UART/SPI/TWI communication interfaces (SSI)
- 1 integrated with 16 * 16-bit hardware Multiplier-Divide Unit (MDU)

LCD/LED driver:

- Choose one of two LCD/LED drive functions, share registers and I/O ports
- LED driver: 8 X 24, 6 X 26, 5 X 27, or 4X 28 segments
- LED segment port source drive capability is selectable in four levels
- LCD driver: 8 X 24, 6 X 26, 5 X 27, or 4X 28 segments
- SC92F8595 without LCD/LED Driver

Analog Peripheral

- 17-channel 12-bit ±2 LSB ADC

  - Built-in 1.024V, 2.4V and 2.048V reference voltages
  - The ADC reference voltages is optional: V_{DD}, 1.024V, 2.4V, 2.048V
  - One internal channel can measure the voltage of the power supply
  - The sampling clock of the ADC circuit follows the f_{SYS}
  - ADC conversion can be set to complete the interrupt
- 1 Analog Comparator
  - The positive input of the analog comparator can be selected as:
    - CMP0-3 One of four input ports
    - Internal 1.5V reference voltage
  - 4-channel input and 1-channel reference voltage input
  - Can awaken the STOP mode
  - 16-level optional comparison voltage
- 31 channels low-power high sensitivity mode TouchKey circuit:
  - Support low-power mode
  - It can be used for touch control applications with high sensitivity, such as spacer touch key control and proximity sensing
  - Able to pass 10V dynamic CS test
  - It can realize 31 high sensitivity spacer touch key and derivative functions
  - High flexibility development software library support, low development difficulty
  - Automatic debugging software support, intelligent development

Power Saving Mode

- IDLE Mode: can be woken up by any interrupt
- STOP Mode: can be woken up by INT0~2, Base Timer, CMP and TK.

Naming Rules for 92 Series Products

Name	SC	92	F	8	5	9	7	X	P	48	R
S/R	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪

S/R	**Meaning**
①	SinOne Chip abbreviation
②	Name of product series
③	Product Type (F: Flash MCU)
④	Serial Number: 7: GP Series, 8: TK series
⑤	ROM Size: 1 for 2K, 2 for 4K, 3 for 8K, 4 for 16K, 5 for 32K
⑥	Subseries Number.: 0 ~ 9, A ~ Z
⑦	Number of Pins: 0: 8pin, 1: 16pin,2: 20pin,3: 28pin,5: 32pin,6: 44pin,7: 48pin,8: 64pin,9: 100pin
⑧	Package Type: (D: DIP; M: SOP; X: TSSOP; F: QFP; P: LQFP; Q: QFN; K: SKDIP)
⑨	Number of Pins.
⑩	Number of Pins.
⑪	Packaging Mode: (U: Tube; R: Tray; T: Reel)

Ordering Information

PRODUCT ID	PACKAGE	PACK
SC92F8593M28U	SOP28	TUBE
SC92F8593X28U	TSSOP28	TUBE
SC92F8595Q32R	QFN32	TRAY
SC92F8595P32R	LQFP32	TRAY
SC92F8596P44R	LQFP44	TRAY
SC92F8597Q48R	QFN48	TRAY
SC92F8597P48R	LQFP48	TRAY

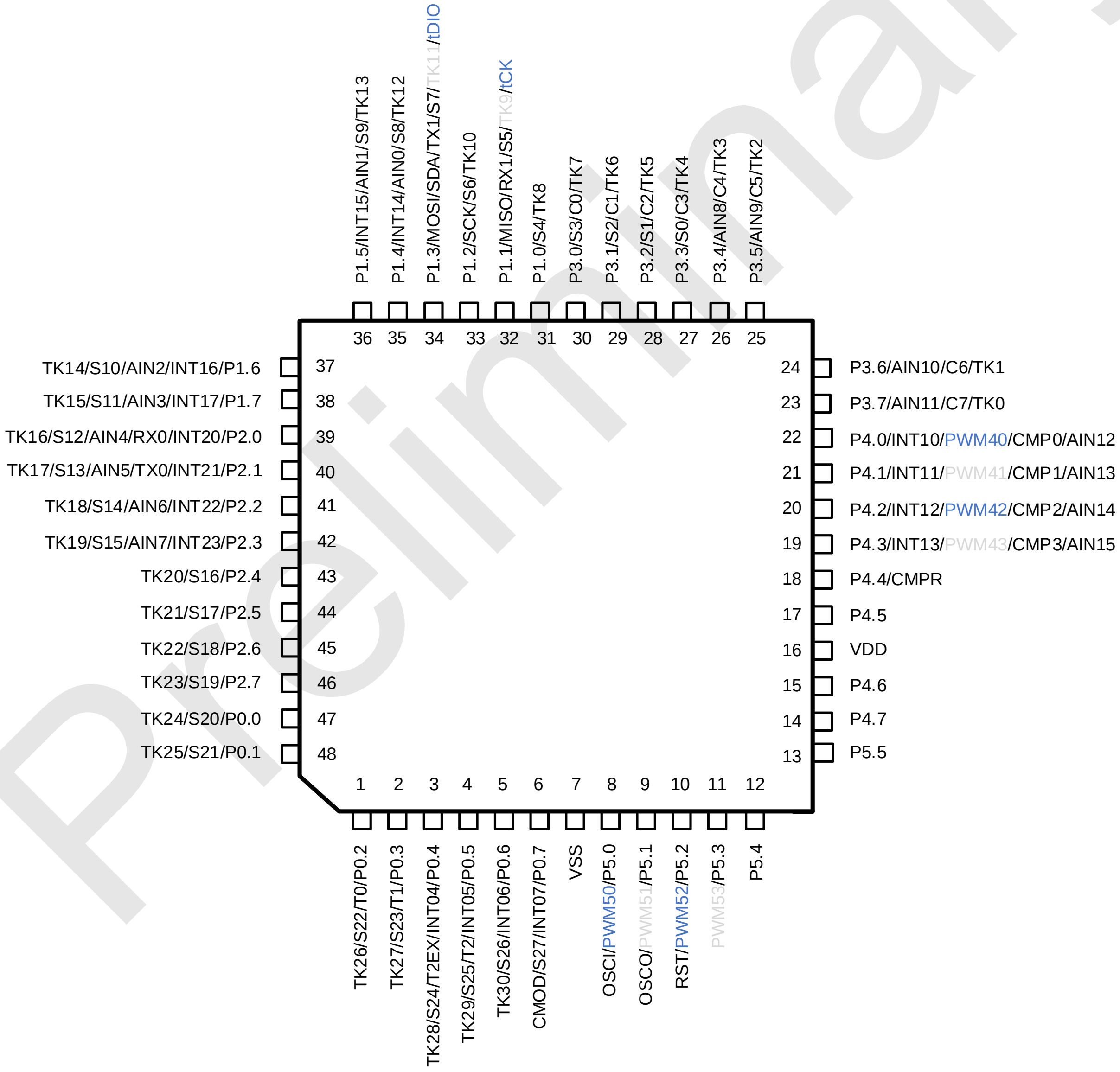
Contents

3 Pin Description

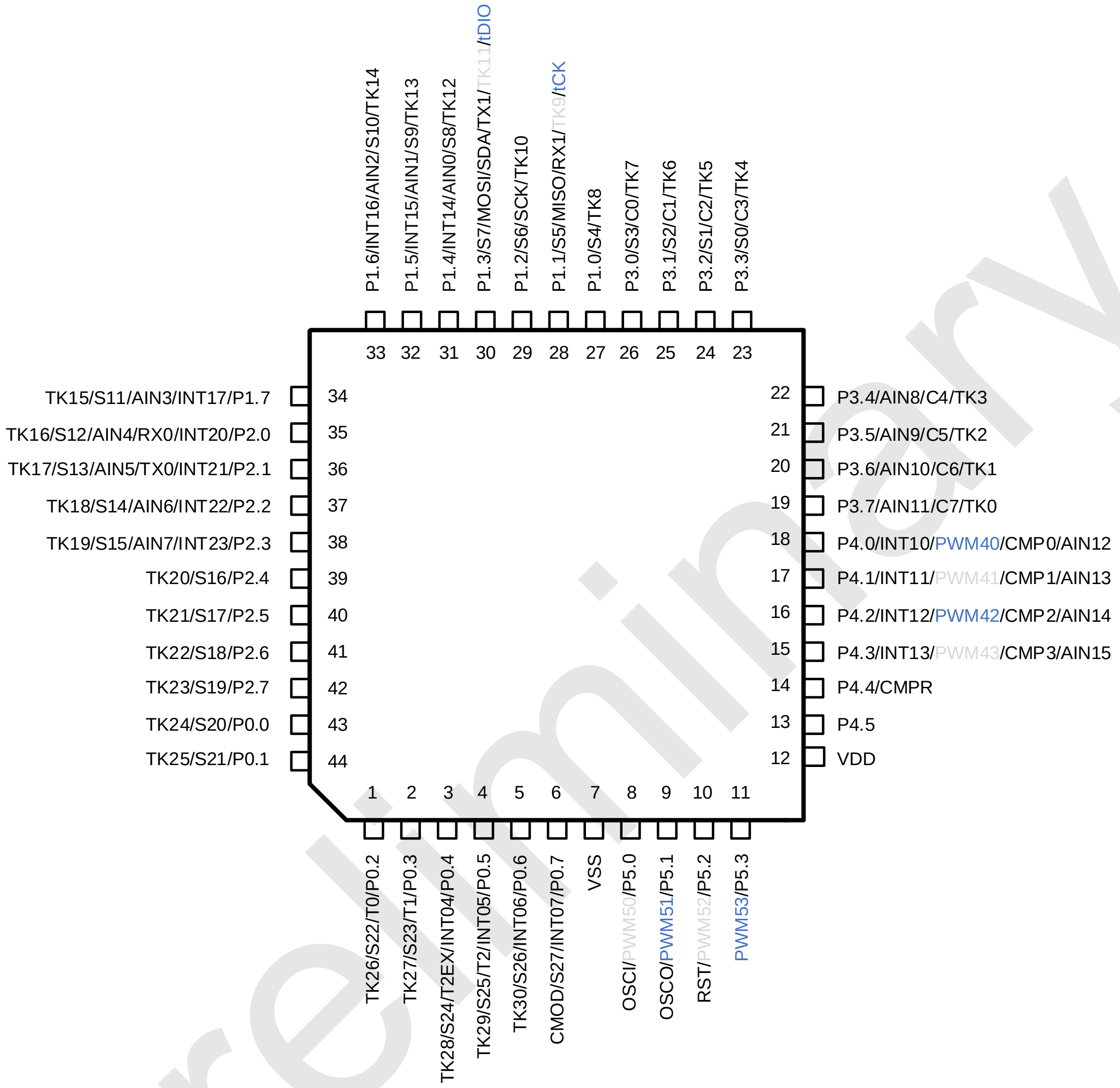
3.1 48/44 Pin Configuration

Note:

1. **In consideration of multiplexing of TK9/TK11 and TK debugging communication ports of the SC92F859X, if it is required to use the TK debugging function, please avoid using TK9/TK11!**
2. **PWM Group Ⅰ: PWM40/PWM42/PWM50/PWM52；PWM Group Ⅱ: PWM41/PWM43/PWM51/PWM53 Two groups of PWM output ports are not allowed to open at the same time! As shown in the 13 PWM.**



48 Pin Diagram



44 Pin Diagram

3.1.1 48/44 Pin Definition

Pin number		Pin Name	Type	Description
48	44			
1	1	**P0.2/T0/S22/TK26**	I/O	P0.2: GPIO P0.2 T0: Timer/Counter 0 External Input S22: LCD/LED SEG22 TK26: TK Channel 26
2	2	**P0.3/T1/S23/TK27**	I/O	P0.3: GPIO P0.3 T1: Timer/Counter 1 External Input S23: LCD/LED SEG23 TK27: TK Channel 27
3	3	**P0.4/INT04/T2EX/S24/TK28**	I/O	P0.4: GPIO P0.4 INT04: Input 4 of external interrupt 0 T2EX: External Capture for Timer2 S24: LCD/LED SEG24 TK28: TK Channel 28
4	4	**P0.5/INT05/T2/S25/TK29**	I/O	P0.5: GPIO P0.5 INT05: Input 5 of external interrupt 0 T2: Timer/Counter 2 External Input S25: LCD/LED SEG25 TK29: TK Channel 29
5	5	**P0.6/INT06/S26/TK30**	I/O	P0.6: GPIO P0.6 INT06: Input 6 of external interrupt 0 S26: LCD/LED SEG26 TK30: TK Channel 30

6	**6**	**P0.7/INT07/S27/CMOD**	I/O	P0.7: GPIO P0.7 INT07: Input 7 of external interrupt 0 S27: LCD/LED SEG27 CMOD: Touch Key internal capacitance
7	**7**	**VSS**	Power	Ground
8	**8**	**P5.0/PWM50/OSCI**	I/O	P5.0: GPIO P5.0 PWM50: PWM Group Ⅰ output port PWM50 OSCI: 32K oscillator output
9	**9**	**P5.1/PWM51/OSCO**	I/O	P5.1: GPIO P5.1 PWM51: PWM Group Ⅱ output port PWM51 OSCO: 32K oscillator output
10	**10**	**P5.2/PWM52/RST**	I/O	P5.2: GPIO P5.2 PWM52: PWM Group Ⅰ output port PWM52 RST: Reset pin
11	**11**	**P5.3/PWM53**	I/O	P5.3: GPIO P5.3 PWM53: PWM Group Ⅱ output port PWM53
12	**-**	**P5.4**	I/O	P5.4: GPIO P5.4
13	**-**	**P5.5**	I/O	P5.5: GPIO P5.5
14	**-**	**P4.7**	I/O	P4.7: GPIO P4.7

15	**-**	**P4.6**	I/O	P4.6: GPIO P4.6
16	**12**	**VDD**	Power	Power
17	**13**	**P4.5**	I/O	P4.5: GPIO P4.5
18	**14**	**P4.4/CMPR**	I/O	P4.4: GPIO P4.4 CMPR: The reference voltage of comparator input.
19	**15**	**P4.3/INT13/PWM43/CMP3/AIN15**	I/O	P4.3: GPIO P4.3 INT13: Input 3 of external interrupt 1 PWM43: PWM Group Ⅱ output port PWM43 CMP3: Analog comparator input channel 3 AIN15: ADC input channel 15
20	**16**	**P4.2/INT12/PWM42/CMP2/AIN14**	I/O	P4.2: GPIO P4.2 INT12: Input 2 of external interrupt 1 PWM42: PWM Group Ⅰ output port PWM42 CMP2: Analog comparator input channel 2 AIN14: ADC input channel 14
21	**17**	**P4.1/INT11/PWM41/CMP1/AIN13**	I/O	P4.1: GPIO P4.1 INT11: Input 1 of external interrupt 1 PWM41: PWM Group Ⅱ output port PWM41 CMP1: Analog comparator input channel 1 AIN13: ADC input channel 13

22	**18**	**P4.0/INT10/PWM40/CMP0/AIN12**	I/O	P4.0: GPIO P4.0 INT10: Input 0 of external interrupt 1 PWM40: PWM Group Ⅰ output port PWM40 CMP0: Analog comparator input channel 0 AIN12: ADC input channel 12
23	**19**	**P3.7/AIN11/C7/TK0**	I/O	P3.7: GPIO P3.7 AIN11: ADC Input Channel 11 C7: LCD/LED common output 7 TK0: TK Channel 0
24	**20**	**P3.6/AIN10/C6/TK1**	I/O	P3.6: GPIO P3.6 AIN10: ADC input channel 10 C6: LCD/LED common output 6 TK1: TK Channel 1
25	**21**	**P3.5/AIN9/C5/TK2**	I/O	P3.5: GPIO P3.5 AIN9: ADC input channel 9 C5: LCD/LED common output 5 TK2: TK Channel 2
26	**22**	**P3.4/AIN8/C4/TK3**	I/O	P3.4: GPIO P3.4 AIN8: ADC input channel 8 C4: LCD/LED common output 4 TK3: TK Channel 3
27	**23**	**P3.3/S0/C3/TK4**	I/O	P3.3: GPIO P3.3 S0: LCD/LED SEG 0 C3: LCD/LED common output 3

				TK4: TK Channel 4
28	**24**	**P3.2/S1/C2/TK5**	I/O	P3.2: GPIO P3.2 S1: LCD/LED SEG 1 C2: LCD/LED common output 2 TK5: TK Channel 5
29	**25**	**P3.1/S2/C1/TK6**	I/O	P3.1: GPIO P3.1 S2: LCD/LED SEG 2 C1: LCD/LED common output 1 TK6: TK Channel 6
30	**26**	**P3.0/S3/C0/TK7**	I/O	P3.0: GPIO P3.0 S3: LCD/LED SEG 3 C0: LCD/LED common output 0 TK7: TK Channel 7
31	**27**	**P1.0/S4/TK8**	I/O	P1.0: GPIO P1.0 S4: LCD/LED SEG 4 TK8: TK Channel 8
32	**28**	**P1.1/MISO/RX1/S5/TK9/tCK**	I/O	P1.1: GPIO P1.1 MISO: Master input / Slave output of SPI RX1: UART1 Receiver S5: LCD/LED SEG 5 TK9: TK Channel 9, **if it is required to use the TK debugging function, please avoid using TK9!** tCK: Programming and Emulation Clock Pin
33	**29**	**P1.2/SCK/S6/TK10**	I/O	P1.2: GPIO P1.2 SCK: SCK of SPI and TWI

				S6: LCD/LED SEG 6 TK10: TK Channel 10
34	**30**	**P1.3/MOSI/SDA/TX1/S7/TK11/tDIO**	I/O	P1.3: GPIO P1.3 MOSI: Master output / Slave input of SPI SDA: SDA of TWI TX1: UART1 Transmitter S7: LCD/LED SEG 7 TK11: TK channel 11, **if it is required to use the TK debugging function, please avoid using TK11** tDIO: Programming and Emulation Data Pin
35	**31**	**P1.4/INT14/AIN0/S8/TK12**	I/O	P1.4: GPIO P1.4 INT14: Input 4 of external interrupt 1 AIN0: ADC input channel 0 S8: LCD/LED SEG 8 TK12: TK Channel 12
36	**32**	**P1.5/INT15/AIN1/S9/TK13**	I/O	P1.5: GPIO P1.5 INT15: Input 5 of external interrupt 1 AIN1: ADC input channel 1 S9: LCD/LED SEG 9 TK13: TK Channel 13
37	**33**	**P1.6/INT16/AIN2/S10/TK14**	I/O	P1.6: GPIO P1.6 INT16: Input 6 of external interrupt 1 AIN2: ADC Input Channel 2 S10: LCD/LED SEG10 TK14: TK Channel 14
38	**34**	**P1.7/INT17/AIN3/S11/TK15**	I/O	P1.7: GPIO P1.7 INT17: Input 7 of external interrupt 1

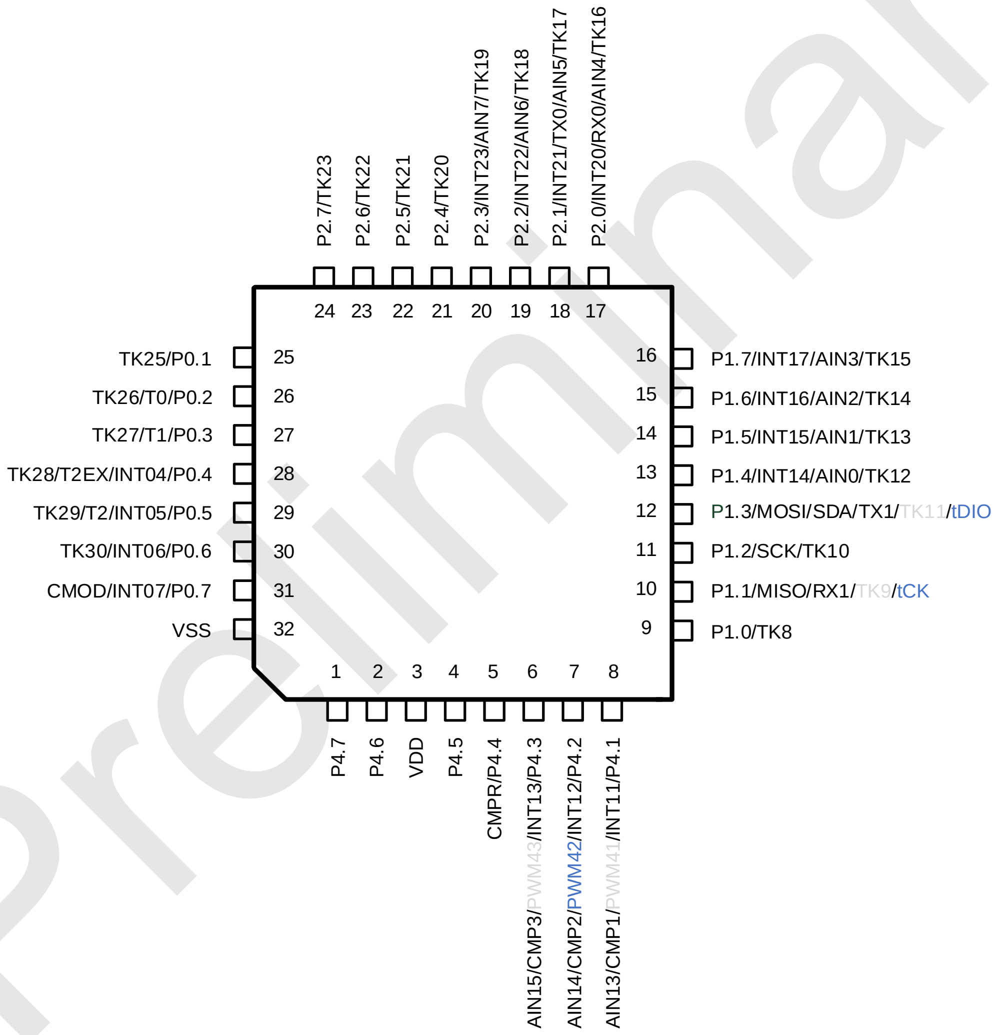
				AIN3: ADC Input Channel 3 S11: LCD/LED SEG11 TK15: TK Channel 15
39	**35**	**P2.0/INT20/RX0/AIN4/S12/TK16**	I/O	P2.0: GPIO P2.0 INT20: Input 0 of external interrupt 2 RX0: UART0 Receiver AIN4: ADC Input Channel 4 S12: LCD/LED SEG12 TK16: TK Channel 16
40	**36**	**P2.1/INT21/TX0/AIN5/S13/TK17**	I/O	P2.1: GPIO P2.1 INT21: Input 1 of external interrupt 2 TX0: UART0 Transmitter AIN5: ADC Input Channel 5 S13: LCD/LED SEG13 TK17: TK Channel 17
41	**37**	**P2.2/INT22/AIN6/S14/TK18**	I/O	P2.2: GPIO P2.2 INT22: Input 2 of external interrupt 2 AIN6: ADC Input Channel 6 S14: LCD/LED SEG14 TK18: TK Channel 18
42	**38**	**P2.3/INT23/AIN7/S15/TK19**	I/O	P2.3: GPIO P2.3 INT23: Input 3 of external interrupt 2 AIN7: ADC Input Channel 7 S15: LCD/LED SEG15 TK19: TK Channel 19
43	**39**	**P2.4/S16/TK20**	I/O	P2.4: GPIO P2.4 S16: LCD/LED SEG 16 TK20: TK Channel 20

44	**40**	**P2.5/S17/TK21**	I/O	P2.5: GPIO P2.5 S17: LCD/LED SEG 17 TK21: TK Channel 21
45	**41**	**P2.6/S18/TK22**	I/O	P2.6: GPIO P2.6 S18: LCD/LED SEG 18 TK22: TK Channel 22
46	**42**	**P2.7/S19/TK23**	I/O	P2.7: GPIO P2.7 S19: LCD/LED SEG 19 TK23: TK Channel 23
47	**43**	**P0.0/S20/TK24**	I/O	P0.0: GPIO P0.0 S20: LCD/LED SEG 20 TK24: TK Channel 24
48	**44**	**P0.1/S21/TK25**	I/O	P0.1: GPIO P0.1 S21: LCD/LED SEG 21 TK25: TK Channel 25

3.2 32 Pin Configuration

1. **In consideration of multiplexing of TK9/TK11 and TK debugging communication ports of the SC92F859X, if it is required to use the TK debugging function, please avoid using TK9/TK11!**

2. **PWM Group Ⅰ: PWM40/PWM42/PWM50/PWM52；PWM Group Ⅱ: PWM41/PWM43/PWM51/PWM53 Two groups of PWM output ports are not allowed to open at the same time! As shown in the 13 PWM.**



32 Pin Diagram

3.2.1 32 Pin Definition

Pin number	Pin Name	Type	Description
1	**P4.7**	I/O	P4.7: GPIO P4.7
2	**P4.6**	I/O	P4.6: GPIO P4.6
3	**VDD**	Power	Power
4	**P4.5**	I/O	P4.5: GPIO P4.5
5	**P4.4/CMPR**	I/O	P4.4: GPIO P4.4 CMPR: The reference voltage of comparator input.
6	**P4.3/INT13/PWM43/CMP3/AIN15**	I/O	P4.3: GPIO P4.3 INT13: Input 3 of external interrupt 1 PWM43: PWM Group Ⅱ output port PWM43 CMP3: Analog comparator input channel 3 AIN15: ADC input channel 15
7	**P4.2/INT12/PWM42/CMP2/AIN14**	I/O	P4.2: GPIO P4.2 INT12: Input 2 of external interrupt 1 PWM42: PWM Group Ⅰ output port PWM42 CMP2: Analog comparator input channel 2 AIN14: ADC input channel 14
8	**P4.1/INT11/PWM41/CMP1/AIN13**	I/O	P4.1: GPIO P4.1 INT11: Input 1 of external interrupt 1 PWM41: PWM Group Ⅱ output port PWM41 CMP1: Analog comparator input channel 1 AIN13: ADC input channel 13
9	**P1.0/TK8**	I/O	P1.0: GPIO P1.0 TK8: TK Channel 8
10	**P1.1/MISO/RX1/TK9/tCK**	I/O	P1.1: GPIO P1.1

			MISO: Master input/ Slave output of SPI RX1: UART1 Receiver TK9: TK Channel 9 tCK: Programming and Emulation Clock Pin
11	**P1.2/SCK/TK10**	I/O	P1.2: GPIO P1.2 SCK: SCK of SPI and TWI TK10: TK Channel 10
12	**P1.3/MOSI/SDA/TX1/TK11/tDIO**	I/O	P1.3: GPIO P1.3 MOSI: Master output / Slave input of SPI SDA: SDA of TWI TX1: UART1 Transmitter TK11: TK Channel 11 tDIO: Programming and Emulation Data Pin
13	**P1.4/INT14/AIN0/TK12**	I/O	P1.4: GPIO P1.4 INT14: Input 4 of external interrupt 1 AIN0: ADC input channel 0 TK12: TK Channel 12
14	**P1.5/INT15/AIN1/TK13**	I/O	P1.5: GPIO P1.5 INT15: Input 5 of external interrupt 1 AIN1: ADC input channel 1 TK13: TK Channel 13
15	**P1.6/INT16/AIN2/TK14**	I/O	P1.6: GPIO P1.6 INT16: Input 6 of external interrupt 1 AIN2: ADC input channel 2 TK14: TK Channel 14
16	**P1.7/INT17/AIN3/TK15**	I/O	P1.7: GPIO P1.7 INT17: Input 7 of external interrupt 1 AIN3: ADC input channel 3 TK15: TK Channel 15
17	**P2.0/INT20/RX0/AIN4/TK16**	I/O	P2.0: GPIO P2.0 INT20: Input 0 of external interrupt 2 RX0: UART0 Receiver

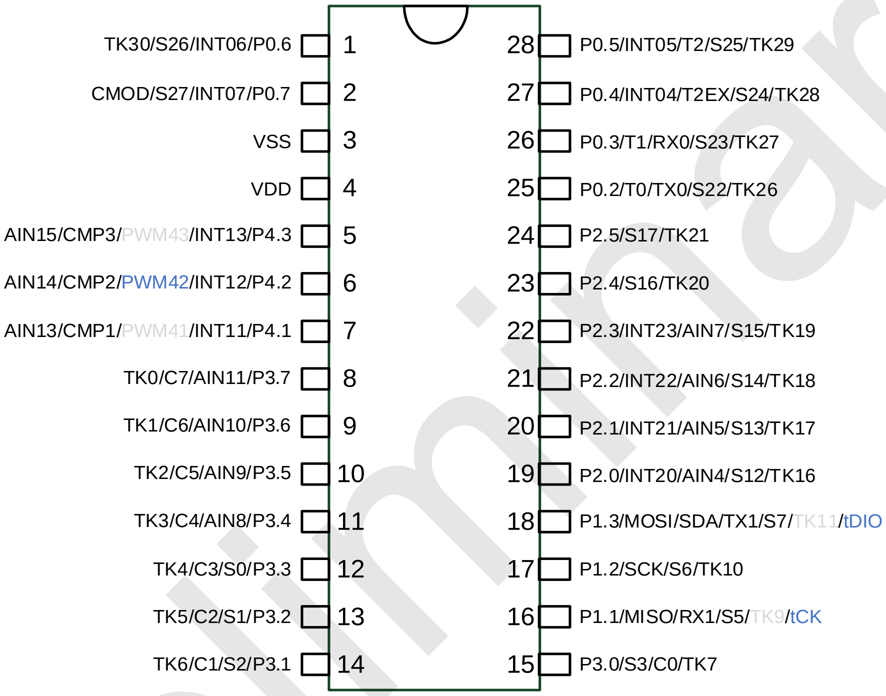
			AIN4: ADC input channel 4 TK16: TK Channel 16
18	**P2.1/INT21/TX0/AIN5/TK17**	I/O	P2.1: GPIO P2.1 INT21: Input 1 of external interrupt 2 TX0: UART0 Transmitter AIN5: ADC input channel 5 TK17: TK Channel 17
19	**P2.2/INT22/AIN6/TK18**	I/O	P2.2: GPIO P2.2 INT22: Input 2 of external interrupt 2 AIN6: ADC input channel 6 TK18: TK Channel 18
20	**P2.3/INT23/AIN7/TK19**	I/O	P2.3: GPIO P2.3 INT23: Input 3 of external interrupt 2 AIN7: ADC input channel 7 TK19: TK Channel 19
21	**P2.4/TK20**	I/O	P2.4: GPIO P2.4 TK20: TK Channel 20
22	**P2.5/TK21**	I/O	P2.5: GPIO P2.5 TK21: TK Channel 21
23	**P2.6/TK22**	I/O	P2.6: GPIO P2.6 TK22: TK Channel 22
24	**P2.7/TK23**	I/O	P2.7: GPIO P2.7 TK23: TK Channel 23
25	**P0.1/TK25**	I/O	P0.1: GPIO P0.1 TK25: TK Channel 25
26	**P0.2/T0/TK26**	I/O	P0.2: GPIO P0.2 T0: Timer/Counter 0 external input TK26: TK Channel 26
27	**P0.3/T1/TK27**	I/O	P0.3: GPIO P0.3 T1: Timer/Counter 1 external input TK27: TK Channel 27

28	**P0.4/INT04/T2EX/TK28**	I/O	P0.4: GPIO P0.4 INT04: Input 4 of external interrupt 0 T2EX: External capture for Timer2 TK28: TK Channel 28
29	**P0.5/INT05/T2/TK29**	I/O	P0.5: GPIO P0.5 INT05: Input 5 of external interrupt 0 T2: Timer/Counter 2 external input TK29: TK Channel 29
30	**P0.6/INT06/TK30**	I/O	P0.6: GPIO P0.6 INT06: Input 6 of external interrupt 0 TK30: TK Channel 30
31	**P0.7/INT07/CMOD**	I/O	P0.7: GPIO P0.7 INT07: Input 7 of external interrupt 0 CMOD: Touch Key internal capacitance
32	**VSS**	Power	Ground

3.3 28 Pin Configuration

1. **In consideration of multiplexing of TK9/TK11 and TK debugging communication ports of the SC92F859X, if it is required to use the TK debugging function, please avoid using TK9/TK11!**

2. **PWM Group Ⅰ: PWM40/PWM42/PWM50/PWM52；PWM Group Ⅱ: PWM41/PWM43/PWM51/PWM53 Two groups of PWM output ports are not allowed to open at the same time! As shown in the 13 PWM.**


TK30/S26/INT06/P0.6 1
CMOD/S27/INT07/P0.7 2
VSS 3
VDD 4
AIN15/CMP3/PWM43/INT13/P4.3 5
AIN14/CMP2/PWM42/INT12/P4.2 6
AIN13/CMP1/PWM41/INT11/P4.1 7
TK0/C7/AIN11/P3.7 8
TK1/C6/AIN10/P3.6 9
TK2/C5/AIN9/P3.5 10
TK3/C4/AIN8/P3.4 11
TK4/C3/S0/P3.3 12
TK5/C2/S1/P3.2 13
TK6/C1/S2/P3.1 14
28 P0.5/INT05/T2/S25/TK29
27 P0.4/INT04/T2EX/S24/TK28
26 P0.3/T1/RX0/S23/TK27
25 P0.2/T0/TX0/S22/TK26
24 P2.5/S17/TK21
23 P2.4/S16/TK20
22 P2.3/INT23/AIN7/S15/TK19
21 P2.2/INT22/AIN6/S14/TK18
20 P2.1/INT21/AIN5/S13/TK17
19 P2.0/INT20/AIN4/S12/TK16
18 P1.3/MOSI/SDA/TX1/S7/TK11/tDIO
17 P1.2/SCK/S6/TK10
16 P1.1/MISO/RX1/S5/TK9/tCK
15 P3.0/S3/C0/TK7

28 Pin Diagram

3.3.1 28 Pin Definition

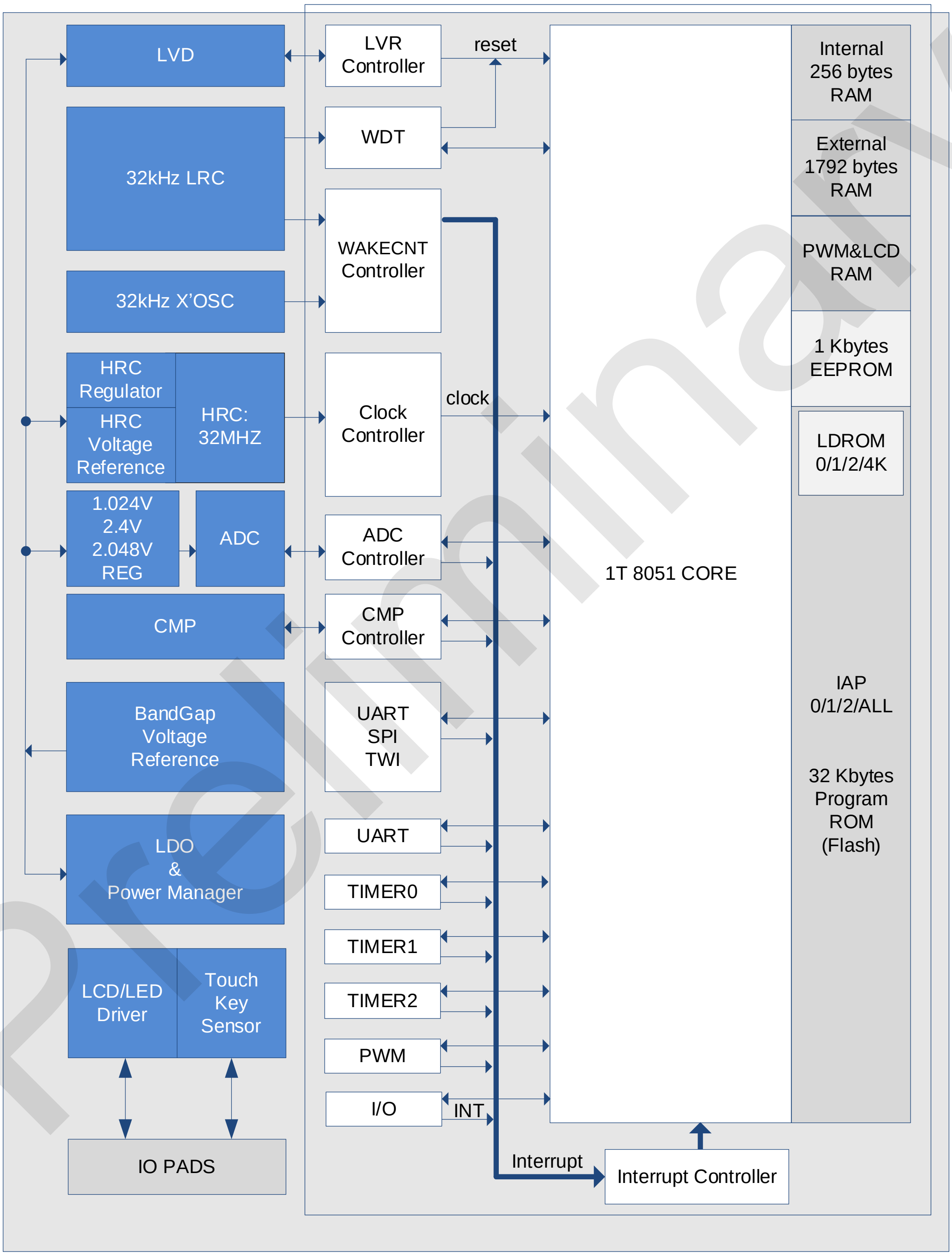
Pin number	Pin Name	Type	Description
1	**P0.6/INT06/S26/TK30**	I/O	P0.6: GPIO P0.6 INT06: Input 6 of external interrupt 0 S26: LCD/LED SEG26 TK30: TK Channel 30
2	**P0.7/INT07/S27**	I/O	P0.7: GPIO P0.7 INT07: Input 7 of external interrupt 0 S27: LCD/LED SEG27
3	**VSS**	Power	Ground
4	**VDD**	Power	Power
5	**P4.3/INT13/PWM43/CMP3/AIN15**	I/O	P4.3: GPIO P4.3 INT13: Input 3 of external interrupt 1 PWM43: PWM Group Ⅱ output port PWM43 CMP3: Analog comparator input channel 3 AIN15: ADC input channel 15
6	**P4.2/INT12/PWM42/CMP2/AIN14**	I/O	P4.2: GPIO P4.2 INT12: Input 2 of external interrupt 1 PWM42: PWM Group Ⅰ output port PWM42 CMP2: Analog comparator input channel 2 AIN14: ADC input channel 14
7	**P4.1/INT11/PWM41/CMP1/AIN13**	I/O	P4.1: GPIO P4.1 INT11: Input 1 of external interrupt 1 PWM41: PWM Group Ⅱ output port PWM41 CMP1: Analog comparator input channel 1 AIN13: ADC input channel 13
8	**P3.7/AIN11/C7/TK0**	I/O	P3.7: GPIO P3.7 AIN11: ADC input channel 11

			C7: LCD/LED common output 7 TK0: TK Channel 0
9	**P3.6/AIN10/C6/TK1**	I/O	P3.6: GPIO P3.6 AIN10: ADC input channel 10 C6: LCD/LED common output 6 TK1: TK Channel 1
10	**P3.5/AIN9/C5/TK2**	I/O	P3.5: GPIO P3.5 AIN9: ADC input channel 9 C5: LCD/LED common output 5 TK2: TK Channel 2
11	**P3.4/AIN8/C4/TK3**	I/O	P3.4: GPIO P3.4 AIN8: ADC input channel 8 C4: LCD/LED common output 4 TK3: TK Channel 3
12	**P3.3/S0/C3/TK4**	I/O	P3.3: GPIO P3.3 S0: LCD/LED SEG0 C3: LCD/LED common output 3 TK4: TK Channel 4
13	**P3.2/S1/C2/TK5**	I/O	P3.2: GPIO P3.2 S1: LCD/LED SEG1 C2: LCD/LED common output 2 TK5: TK Channel 5
14	**P3.1/S2/C1/TK6**	I/O	P3.1: GPIO P3.1 S2: LCD/LED SEG2 C1: LCD/LED common output 1 TK6: TK Channel 6
15	**P3.0/S3/C0/TK7**	I/O	P3.0: GPIO P3.0 S3: LCD/LED SEG3 C0: LCD/LED common output 0 TK7: TK Channel 7
16	**P1.1/MISO/RX1/S5/TK9/tCK**	I/O	P1.1: GPIO P1.1 MISO: Master input/ Slave output of SPI RX1: UART1 Receiver S5: LCD/LED SEG5 TK9: TK Channel 9

			tCK: Programming and Emulation Clock Pin
17	**P1.2/SCK/S6/TK10**	I/O	P1.2: GPIO P1.2 SCK: SCK of SPI and TWI S6: LCD/LED SEG6 TK10: TK Channel 10
18	**P1.3/MOSI/SDA/TX1/S7/TK11/tDIO**	I/O	P1.3: GPIO P1.3 MOSI: Master output / Slave input of SPI SDA: SDA of TWI TX1: UART1 Transmitter S7: LCD/LED SEG7 TK11: TK Channel 11 tDIO: Programming and Emulation Data Pin
19	**P2.0/INT20/AIN4/S12/TK16**	I/O	P2.0: GPIO P2.0 INT20: Input 0 of external interrupt 2 AIN4: ADC input channel 4 S12: LCD/LED SEG12 TK16: TK Channel 16
20	**P2.1/INT21/AIN5/S13/TK17**	I/O	P2.1: GPIO P2.1 INT21: Input 1 of external interrupt 2 AIN5: ADC input channel 5 S13: LCD/LED SEG13 TK17: TK Channel 17
21	**P2.2/INT22/AIN6/S14/TK18**	I/O	P2.2: GPIO P2.2 INT22: Input 2 of external interrupt 2 AIN6: ADC input channel 6 S14: LCD/LED SEG14 TK18: TK Channel 18
22	**P2.3/INT23/AIN7/S15/TK19**	I/O	P2.3: GPIO P2.3 INT23: Input 3 of external interrupt 2 AIN7: ADC input channel 7 S15: LCD/LED SEG15 TK19: TK Channel 19
23	**P2.4/S16/TK20**	I/O	P2.4: GPIO P2.4 S16: LCD/LED SEG16 TK20: TK Channel 20

24	**P2.5/S17/TK21**	I/O	P2.5: GPIO P2.5 S17: LCD/LED SEG17 TK21: TK Channel 21
25	**P0.2/T0/TX0/S22/TK26**	I/O	P0.2: GPIO P0.2 T0: Timer/Counter 0 external input TX0: UART0 Transmitter S22: LCD/LED SEG22 TK26: TK Channel 26
26	**P0.3/T1/RX0/S23/TK27**	I/O	P0.3: GPIO P0.3 T1: Timer/Counter 1 external input RX0: UART0 Receiver S23: LCD/LED SEG23 TK27: TK Channel 27
27	**P0.4/INT04/T2EX/S24/TK28**	I/O	P0.4: GPIO P0.4 INT04: Input 4 of external interrupt 0 T2EX: External capture for Timer 2 S24: LCD/LED SEG24 TK28: TK Channel 28
28	**P0.5/INT05/T2/S25/TK29**	I/O	P0.5: GPIO P0.5 INT05: Input 5 of external interrupt 0 T2: Timer/Counter 2 external input S25: LCD/LED SEG25 TK29: TK Channel 29

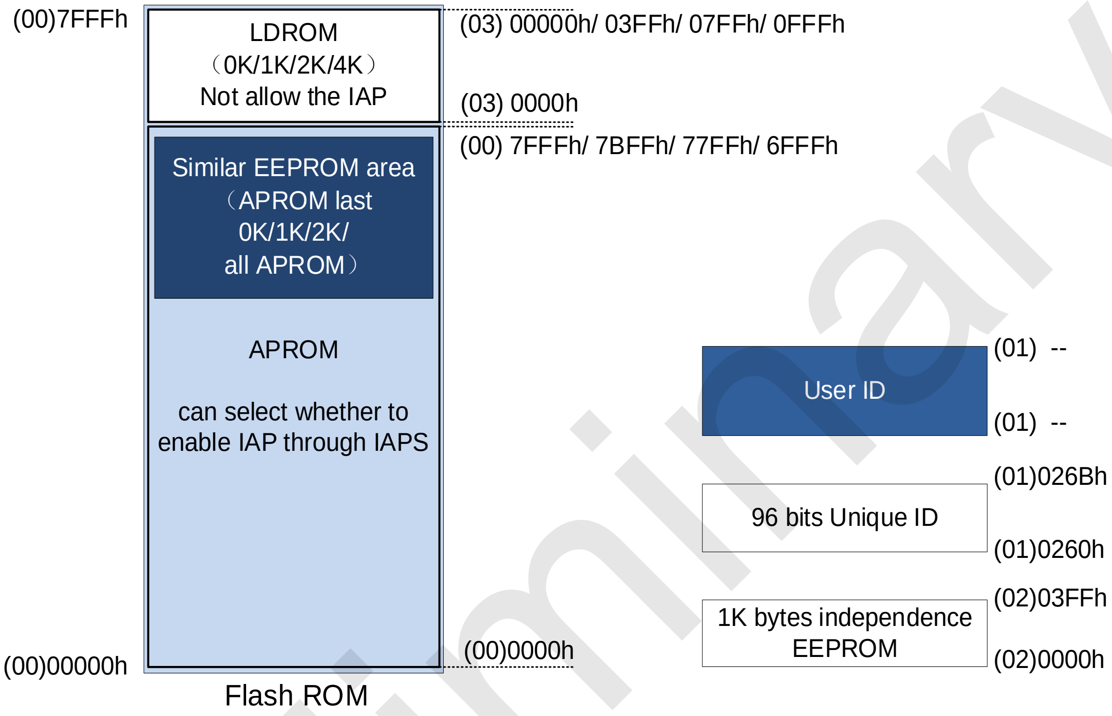
4 Inner Block Diagram



SC92F859X BLOCK DIAGRAM

5 Flash ROM and SRAM

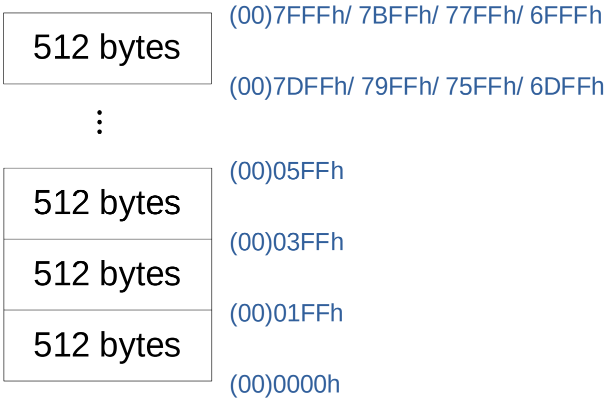
The Flash ROM of SC92F859X is divided into five regions: APROM/LDROM/EEPROM/User ID/Unique ID, as shown in the following figure:



5.1 APROM and LDROM

The SC92F859X's APROM and LDROM are two independent pieces of hardware that divide ROM by LDSIZE [1:0]. They are distinguished by the extended address "00" and "03" set by IAPADE register. They can be programmed and erased by SCLINK PRO provided by SinOne.

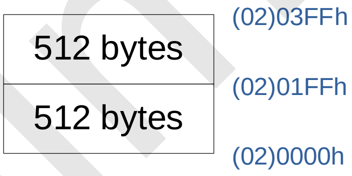
- The extended APROM address is "00". The area size is 28 to 32K bytes. This parameter is optional. The Customer Option item can set the range of IAP operations allowed to be 0K / 1K / 2K /all APROM;
- The extended LDROM address is "03". This parameter is optional. The area size ranges from 0 to 4K bytes.
- APROM and LDROM have 32 sectors, each 512 bytes, and can be repeatedly written for 100,000 times. Data can be stored for more than 100 years at 25 ° C.



SC92F859X APROM Sector Partition diagram

5.2 1K BYTES Independent EEPROM

1K bytes Independent EEPROM which the area address ranges from (02) 000H to 03FFH, where “(02)” is the extended address set by the IAPADE register. Independent EEPROM can be repeatedly written 100,000 times, data storage time of more than 100 years. Independent EEPROM supports blank checking, programming, verification, erasing and reading functions.



SC92F859X EEPROM Sector Partition diagram

Note: EEPROM erasure and write times is 100,000 times, user erasure and write times cannot exceed EEPROM rated programming times, otherwise there will be exceptions!

5.3 96 BITS UNIQUE ID AREA

96 bits Unique ID area. The address range is (01) 0260H to 026CH, where “(01)” is the extended address set by the IAPADE register. Stores the IC Unique ID. Users can read the IC Unique ID, but cannot write the Unique ID.

SC92F859X provides an independent Unique ID area, which can be pre-programmed with a 96-bit Unique code before delivery to ensure the uniqueness of the chip. The only way the user can get the serial number is by reading the relative address (01)0260H~(01)026BH via the IAP directive. Address (01)0260H~(01)026BH the 01 in the brackets indicates the extended address, which is set by the IAPADE register. Specific operation methods are as follows:

IAPADE (F4H) IAP Write to Extended Address Register (Read/Write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	IAPADER[7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7~0	**IAPADER[7: 0]**	IAP extended address: 0x00: Both MOVC and IAP are for Flash ROM 0x01: The Unique ID area is read but cannot be written 0x02: Both MOVC and IAP are for independent EEPROM 0x03: MOVC is performed in the LDROM region (**Note: only MOVC can be used, not IAP, this item is only valid for LDROM operation, APROM operation this item is not valid!)** Other: reserved

5.3.1 Unique ID Read Operating Demo Program In C Language

```
#include ”intrins.h”

unsigned char UniqueID [12];//store UniqueID

unsigned char code * POINT =0x0260;

unsigned char i;


EA = 0;                 // Disable the global interrupt

IAPADE = 0X01;                    // Expand address 0x01, select Unique ID area

for(i=0;i<12;i++)

{
    UniqueID [i]= *( POINT+i);         // Read the value of UniqueID
}
IAPADE = 0X00;                         // Expand address 0x00, return to Code area
EA = 1;              // Enable global interrupt
```

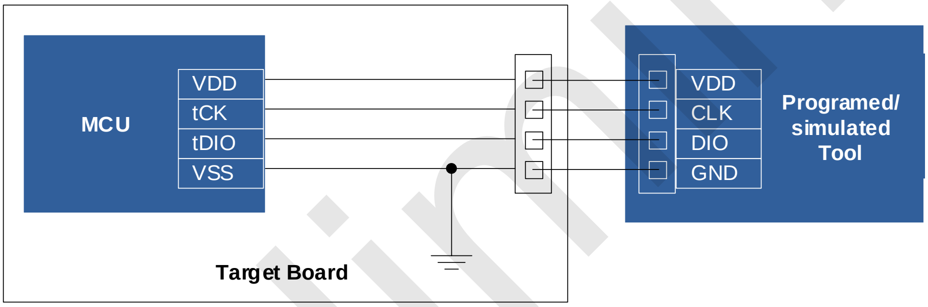
5.4 User ID area

The User ID area, which extended address is (01), is written into the factory. Users can read the User ID area, but forbit write the User ID area.

5.5 programing

SC92F859X's APROM, LDROM and EEPROM can be programmed through tDIO, tCK, VDD, VSS.

The specific connection is as follows:



ICP mode programing connection diagram

tDIO, tCK are 2-line JTAG signal lines, the user can configure the mode of these two ports through Customer Option item during programing:

- JTAG Mode

tDIO and tCK are dedicated ports for programing and simulation, and other functions of reuse are not available. This mode is generally used in online debugging phase to facilitate user simulation debugging; After JTAG mode takes effect, the chip can directly enter programing or simulation mode without powering on or off again.

- Normal Mode(JTAG ports are invalid)

The JTAG function is not available. Other functions reused with it can be used normally. This mode can prevent the programing ports from occupying MCU pin, convenient for users to maximize the use of MCU resources.

Note: When the JTAG port is invalid, the chip must be powered off completely and then powered on again before entering the programing or simulation mode, which will affect the burning and simulation in the live mode. SinOne suggests that users choose the configuration with invalid JTAG special port in mass production and choose JTAG mode in r&d and debugging phase.

The relevant Customer Option registers are as follows:

OP_CTM1 (C2H@FFH)

Code Option register1 (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	VREFS[1: 0]		OP_BL	DISJTG	IAPS[1: 0]		LDSIZE[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	read only	
POR	n	n	n	n	n	n	n	n

Bit number	Bit Mnemonic	Description
4	**DISJTG**	IO/JTAG port switching control 0 : JTAG mode is enabled, P1.1 and P1.3 can only be used as tCK/tDIO. Recommended settings during R&D and commissioning 1 : Normal mode (Normal), JTAG function is invalid. The recommended setting for the mass production burning stage.

5.6 In Application Programming (IAP)

The 64 Kbytes Flash ROM can be set to 1K, 2K, 4K, or 64K through the Code Option Settings. Flash ROM is divided into 128 sectors from (00)0000H to (00)FFFFH. The "00" in brackets is the expanded address set by the IAPADE register:

Application Programming (IAP) operations can be carried out in the APROM of SC92F859X (0K, 1K, 2K, or all APROM ranges are optional) and 1K bytes EEPROM. Users can implement remote program updates through IAP operations. You can also obtain Unique ID field or User ID field information via IAP reads. Before IAPS write data, you must erase the Sector to which the target address belongs. The length of a Sector is 512 bytes.

NOTE:

1. **During the IAP erase/write process, the CPU holds the program counter, and after the IAP erase/write is complete, the program counter continues to execute subsequent instructions.**
2. **IAP operation in APROM area has certain risks, users need to take corresponding security measures in the software, if improper operation may cause user program rewriting! This feature is not recommended unless it is required by the user (for example, for remote application updates).**

3. **The EEPROM erasure count is 100,000. Do not exceed the rated EEPROM erasure count; otherwise, an exception may occur.**

OP_CTM1 (C2H@FFH) Customer Option Register 1(Read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	VREFS[1: 0]		OP_BL	DISJTG	IAPS [1: 0]		LDSIZE [1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	Only read	
POR	n	n	n	n	n	n	n	n

Bit Number	Bit Mnemonic	Description
3~2	**IAPS[1: 0]**	IAP spatial range selection 00: Full Flash ROM **not allows** IAP operation 01: Last 1K Flash ROM **allows** IAP operation 10: Last 2K Flash ROM **allows** IAP operation 11: Full Flash ROM **allows** IAP operation **Note: The above setting items are invalid in BootLoader mode. The BootLoader program can perform IAP operation on the entire Flash ROM area.**

5.6.1 IAP Operation Related Registers

IAP Operation Related Register Description:

Symbol	Address	Description	7	6	5	4	3	2	1	0	POR
IAPKEY	F1H	Data protection register	IAPKEY[7: 0]								00000000b

IAPADL	F2H	IAP write address low register	IAPADR[7: 0]								00000000b
IAPADH	F3H	IAP write address high register	IAPADR[14: 8]								x0000000b
IAPADE	F4H	IAP write to extended address register	IAPADER[7: 0]								00000000b
IAPDAT	F5H	IAP data register	IAPDAT[7: 0]								00000000b
IAPCTL	F6H	IAP control register	BTLD	-	SERASE	PRG	-	-	CMD[1:0]		0x00xx00b

IAPKEY (F1H) IAP Protection Register (Read/Write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	IAPKEY[7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7~0	**IAPKEY[7: 0]**	Open IAP function and operation time limit setting Write a non-zero value n, representative: ① Enable the IAP function; ② If no IAP write command is received after n system clocks, the IAP function is turned off again.

IAPADL (F2H) IAP Write Address Low Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	IAPADR[7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7~0	**IAPADR[7: 0]**	IAP writes the low 8 bits of the address

IAPADH (F3H) IAP Write Address High Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	IAPADR[14: 8]							
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7~0	**IAPADR[14: 8]**	IAP writes the high 8 bits of the address

IAPADE (F4H) IAP Write to Extended Address Register (Read/Write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	IAPADER[7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7~0	**IAPADER[7: 0]**	IAP extended address: 0x00: Both MOVC and IAP are for Flash ROM 0x01: The Unique ID area is read but cannot be written 0x02: Both MOVC and IAP are for independent EEPROM 0x03: MOVC is performed in the LDROM region **(Note: only MOVC can be used, not IAP, this item is only valid for LDROM operation, APROM operation this item is not valid!)** Other: reserved

IAPDAT (F5H) IAP Data Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	IAPDAT[7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7~0	**IAPDAT**	Data written by IAP

IAPCTL (F6H) IAP Control register (Read/Write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	BTLD	-	SERASE	PRG	-	-	CMD[1: 0]	
R/W	R/W	-	R/W	R/W	-	-	R/W	R/W
POR	0	x	0	0	x	x	0	0

Bit number	Bit Mnemonic	Description
7	**BTLD**	BootLoader control bit 0: The program starts to run from the main program area (main program) after Reset; 1: The program starts to run from the BootLoader area after Reset
5	**SERASE**	Sector Erase(Sector Erase)control bit 0: No operation 1: Set CMD[1: 0]=10 after setting "1", then enter the Flash ROM sector erase operation, the specified sector of Flash ROM will be erased
4	**PRG**	Program control bits 0: No operation 1: Set CMD[1: 0]=10 after setting "1", then enter the Flash ROM write operation, and the data in the IADPDA register will be written to the specified Flash ROM address

1~0	**CMD[1: 0]**	IAP Command enable control bit 10: Perform write or sector erase operation commands Others: reserved **Note:** **1. After SERASE / PRG set to "1", CMD[1: 0]=10 must be configured, the corresponding operation will start to execute** **2. Only one IAP operation can be performed at a time, so ERASE / SERASE / PRG can only have one bit set 1 at the same time** **3. Be sure to add at least 8 NOP instructions after the IAP operation statement to ensure that the subsequent instructions can be executed normally after the IAP operation is completed**

5.6.2 IAP Operating Process

The IAP writing process of SC92F859X is as follows:

1. Write IAPDER[7:0]:
   - =0x00, Both MOVC and IAP are for Flash ROM
   - =0x01, The Unique ID area is read but cannot be written
   - =0x02, Both MOVC and IAP are for independent EEPROM
   - =0x03, MOVC is performed in the LDROM region **(Note: only MOVC can be used, not IAP, this item is only valid for LDROM operation, APROM operation this item is not valid!)**
2. Write IAPDAT [7:0], ready for IAP to write data
3. Write IAPADR [14:0] and prepare the target address for the IAP action
4. Write IAPKEY [7:0] to a non-zero value n, turn on IAP protection, and the IAP operation will be turned off if no write command is received within n system clocks
5. After the IAP writes, the CPU continues

5.6.3 IAP Operating Notes

1. When programming IC, if "APROM zone prevents IAP operation" is selected through Code Option, then IAPADE [7:0]=0x00 (APROM zone is selected), IAP cannot be operated, that is, data cannot be written, data can only be read by MOVC instruction;
2. When IAPADE is not 0x00, the MOVC and write target is non-APROM region. At this time, if there is an interrupt and MOVC operation occurs in the interrupt, the result of MOVC will be wrong and the program will run abnormally. To avoid this situation, if IAPADE is not 0x00 during IAP operation, it is important to turn off total interrupt (EA=0) before operation, and set IAPADE = 0x00 after operation before turning on total interrupt (EA=1);

3. During the IAP wipe/write process, the CPU holds the program counter, and the program counter only continues to execute subsequent instructions after the IAP wipe/write is complete;
4. IAP operation in APROM area has certain risks, users need to take corresponding security measures in the software, if improper operation may cause user program rewriting! This feature is not recommended unless required (for remote application updates, for example);
5. The EEPROM erasure count is 100,000. Do not exceed the rated EEPROM erasure count; otherwise, an exception may occur!

5.6.4 IAP Operating Demo Program In C Language

The header files shared by the following routines are as follows:

```
#include ”intrins.h”

unsigned int IAP_Add;

unsigned char IAP_Data;

unsigned char code * POINT =0x0000;
```

IAP Operation: Sector erase:

```
EA = 0;                  //Close global interrupt

IAPADE = 0x00;      //Expand address is 0x00, select Flash ROM

IAPADH = (unsigned char)((IAP_Add >> 8)); // Write the high-bit value of the IAP target address

IAPADL = (unsigned char)IAP_Add;              // Write the low-bit value of the IAP target address

IAPKEY = 0xF0;

IAPCTL = 0x20;        // Set sector erase bit

IAPCTL |= 0x02;       // Block erase

_nop_(); // Wait (at least 8 _nop_() required)

_nop_();

_nop_();

_nop_();

_nop_();

_nop_();

_nop_();

_nop_();

_nop_();
```

```
EA = 1;                 // Open global interrupt
```

IAP operation: Write data:

```
EA = 0;                 // Close global interrupt
IAPADE = 0X00;                              // Extended address is 0x00, choose Flash ROM
IAPDAT = IAP_Data;                          // Send data to IAP data register
IAPADH = (unsigned char)((IAP_Add >> 8)); // Write the high-bit value of the IAP target address
IAPADL = (unsigned char)IAP_Add;            // Write the low-bit value of the IAP target address
IAPKEY = 0xF0; // This value can be adjusted according to the actual situation; it is
                    necessary to ensure that after this instruction is executed and before
                    the IAPCTL is assigned,

                   // The time interval needs to be less than 240 (0xF0) system clocks, otherwise
                   the IAP is disabled;

                  // Pay special attention when opening interrupt
IAPCTL = 0X10;      // Set the IAP write operation bit.
IAPCTL |= 0X02;     // Execute write instruction
_nop_();            // Wait (at least 8 _nop_() required)
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
_nop_();
EA = 1;                 // Open global interrupt
```

IAP operation: read data:

```
EA = 0;                 // Close global interrupt
IAPADE = 0X00;                        //The extended address is 0x00, selectFlash ROM
IAP_Data = *( POINT+IAP_Add);         //Read the value of IAP_Add toIAP_Data
EA = 1;                 // Open global interrupt
```

5.7 BootLoader

LDROM used to store IC BootLoader boot code. LDROM supports blank checking (BLANK), programming (PROGRAM), verifying (VERIFY), erasing (ERASE) and reading (READ) functions in ICP mode. The user must erase the target sector before writing LDROM.

Users can realize ISP (In System Programing) function through LDROM: when ISP is executed, IC runs the boot code In LDROM area. When the boot code is executed, IC receives new program code through serial port, and then programs the received code into user code area through IAP command.

The LDROM has four address ranges:

- (03)0000H~(03)0000H（without LDROM）
- (03)0000H~(03)03FFH（1K）
- (03)0000H~(03)07FFH（2K）
- (03)0000H~(03)0FFFH（4K）

Where: "03" in the brackets above indicates the extended address, which is set by LDSIZE [1:0].

5.7.1 BootLoader Mode operation related registers

OP_CTM1 (C2H@FFH) Code Option Register1 (Read/Write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	VREFS[1: 0]		OP_BL	DISJTG	IAPS[1: 0]		LDSIZE [1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	Only read	
POR	n	n	n	n	n	n	n	n

Bit Number	Bit Mnemonic	Description
5	**OP_BL**	Program run area control bit

<table>
<tr><td></td><td></td><td>0: After the chip is reset, it enters APROM
1: After the chip is reset, it enters LDROM
1. The MOVC and IAP restrictions for APROM are as follows:
<table>
<tr><td>Operation</td><td>Is it operable?</td></tr>
<tr><td>LDROM MOVC</td><td>x</td></tr>
<tr><td>APROM MOVC</td><td>√</td></tr>
<tr><td>LDROM IAP</td><td>×</td></tr>
<tr><td>APROM IAP</td><td>√</td></tr>
</table>
2. The MOVC and IAP restrictions for LDROM are as follows:
<table>
<tr><td>Operation</td><td>Is it operable?</td></tr>
<tr><td>LDROM MOVC</td><td>√</td></tr>
<tr><td>APROM MOVC</td><td>√</td></tr>
<tr><td>LDROM IAP</td><td>×</td></tr>
<tr><td>ALL APROM IAP, not restricted by IAPRANGE</td><td>√</td></tr>
</table>
</td></tr>
<tr><td><b>1~0</b></td><td><b>LDSIZE [1:0]</b></td><td>LDROM space range selection
00: None LDROM, the APROM address is 0000H to 7FFFH
01: The last 1K APROM area of the Flash ROM is LDROM, and the APROM address is 0000H-7BFFh
10: The last 2K APROM area of the Flash ROM is LDROM, and the APROM address is 0000H-77FFh
11: The last 4K APROM area of the Flash ROM is LDROM, and the APROM address is 0000H-6FFFh
<b>NOTE: LDROM not allow IAP operation in anyways</b></td></tr>
</table>

IAPKEY (F1H) Data Protection Register (Read/Write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	IAPKEY[7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7~0	**IAPKEY[7: 0]**	Open IAP and operation time limit setting Write a value n greater than or equal to 0x40, which represents: ① Enable the IAP; ② If no IAP write command is received after n system clocks, the IAP is turned off again.

IAPADL (F2H) IAP Write Low Address Register (Read/Write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	IAPADR[7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description

7~0	**IAPADR[7: 0]**	IAP writes the low 8 bits of the address

IAPADH (F3H) IAP Write High Address Register (Read/Write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	IAPADR[14: 8]							
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7~0	**IAPADR[14: 8]**	IAP writes the high 8 bits of the address

IAPADE (F4H) IAP Write to Extended Address Register (Read/Write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	IAPADER[7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7~0	**IAPADER[7: 0]**	IAP extended address:

		0x00: Both MOVC and IAP are for Flash ROM 0x01: The Unique ID area is read but cannot be written 0x02: Both MOVC and IAP are for independent EEPROM 0x03: MOVC is performed in the LDROM region (**Note: only MOVC can be used, not IAP, this item is only valid for LDROM operation, APROM operation this item is not valid!)** Other: reserved

IAPDAT (F5H) IAP Data Register (Read/Write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	IAPDAT[7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7~0	**IAPDAT**	Data written by IAP

IAPCTL (F6H) IAP Control register (Read/Write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	BTLD	-	SERASE	PRG	-	-	CMD[1: 0]	
R/W	R/W	-	R/W	R/W	-	-	R/W	R/W
POR	0	x	0	0	x	x	0	0

Bit number	Bit Mnemonic	Description
7	**BTLD**	BootLoader control bit 0: The program starts to run from the main program area (main program) after Reset; 1: The program starts to run from the BootLoader area after Reset
5	**SERASE**	Sector Erase(Sector Erase)control bit 0: No operation 1: Set CMD [1: 0] =10 after setting "1", then enter the Flash ROM sector erase operation, the specified sector of Flash ROM will be erased
4	**PRG**	Program control bits 0: No operation 1: Set CMD[1: 0]=10 after setting "1", then enter the Flash ROM write operation, and the data in the IADPDA register will be written to the specified Flash ROM address
1~0	**CMD[1: 0]**	IAP Command enable control bit 10: Perform write or sector erase operation commands Others: reserved **Note:** **1. After SERASE / PRG set to "1", CMD[1: 0]=10 must be configured, the corresponding operation will start to execute** **2. Only one IAP operation can be performed at a time, so ERASE / SERASE / PRG can only have one bit set 1 at the same time** **3. Be sure to add at least 8 NOP instructions after the IAP operation statement to ensure that the subsequent instructions can be executed normally after the IAP operation is completed**

PCON (87h) Power Management Control Register (write only, *not readable*)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	SMOD	-	-	-	RST	-	STOP	IDL
R/W	write only	-	-	-	write only	-	write only	write only
POR	0	x	x	x	n	x	0	0

Bit number	Bit Mnemonic	Description
7	**SMOD**	When SM0~1 = 01 (UART0 mode 1) or SM0~1 = 11 (UART0 mode 3), the baud rate is set to bit: 0: the serial port runs at 1 division of the system clock 1: the serial port runs at 16 division of the system clock When SM0~1 = 00 (UART0 mode 0) Baud rate setting bit: 0: the serial port runs at 12 division of the system clock 1: the serial port runs at 4 division of the system clock
3	**RST**	Software reset control bit: Write status: 0: The program runs normally; 1: The CPU resets immediately after this bit is written to "1"

Bootloader Notes:

1. The user must erase the target sector before writing LDROM;
2. For the specific operation method, please refer to the description document "SinOne hardware BootLoader Function Implementation Application Guide" provided by SinOne.

5.8 Encryption

Users can choose whether to encrypt the SC92F859X's ROM through the settings on the computer program:

1. If the encryption function is disabled, users can read the last data written in APROM and LDROM by SC LINK;
2. If the encryption function is enabled, the data written in APROM (64 Kbytes Flash ROM) and LDROM will never be read from outsid. It is recommended to enable the encryption function during mass production;
3. The only way to release security encryption is to re-programming
4. The encryption has no effect on iap read and write operation
5. For the specific operation method, please refer to the chapter of Secure Encryption and Reading in the " SOC LINK Series Programmer Simulator User Manual ".

5.9 Customer Option Area (User Programming Settings)

There is a separate Flash area inside the SC92F859X to save the customer's initial settings. This area is called Customer Option area. The user writes this part of the code inside the IC when programming the IC. When the IC is reset and initialized, it will transfer this setting to SFR as the initial setting.

Option related SFR operation instructions:

The read and write operations of Option-related SFRs are controlled by OPINX and OPREG registers. The specific location of each Option SFR is determined by OPINX, as shown in the following table:

Symbol	OPINX Address	Instructions	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
OP_CTM0	0xC1	Code Option register0	ENWDT	ENXTL	SCLK[1:0]		DISRST	DISLVR	LVRS[1:0]	
OP_CTM1	0xC2	Code Option register1	VREFS[1:0]		OP_BL	DISJTG	IAPS[1:0]		LDSIZE[1:0]	
OP_HRCR	0x83	System Clock Change register	OP_HRCR[7:0]							

IFB Address	Symbol	Write read	Instructions

OP_CTM0[7]	**ENWDT**	**R/W**	**WDT Switch** 0： WDT invalid 1： WDT valid
OP_CTM0[6]	**ENXTL**	**R/W**	**External 32K crystal selector switch** 0: External 32K crystal Interface disable, P5.0 and P5.1 valid 1: External 32K crystal Interface enable, P5.0 and P5.1 invalid
OP_CTM0[5~4]	**SCLKS [1:0]**	**R/W**	**System clock frequency selection bits** 00: System clock frequency is HRC frequency divided by 1; 01: System clock frequency is HRC frequency divided by 2; 10: System clock frequency is HRC frequency divided by 4; 11: System clock frequency is HRC frequency divided by12;
OP_CTM0[3]	**DISRST**	**Read only**	**IO/RST Selection bit** 0: configure P5.2 as External Reset input pin 1: configure P5.2 as GPIO
OP_CTM0[2]	**DISLVR**	**R/W**	**LVR control bit** 0: LVR valid 1: LVR invalid
OP_CTM0[1~0]	**LVRS [1:0]**	**R/W**	**LVR voltage selection control** 11: 4.3V reset 10: 3.7V reset 01: 2.9V reset 00: 1.9V reset
OP_CTM1[7~6]	**VREFS [1:0]**	**R/W**	**Reference voltage selection** 00: Configure ADC VREF as VDD;

<table>
<tr><td></td><td></td><td></td><td>01: Configure ADC VREF as internal 1.024V
10: Configure ADC VREF as internal 2.4V
11: Configure ADC VREF as internal 2.048V</td></tr>
<tr><td><b>OP_CTM1[5]</b></td><td><b>OP_BL</b></td><td><b>R/W</b></td><td><b>Program run area control bit</b>
0: After the chip is reset, it enters APROM
1: After the chip is reset, it enters LDROM

1. The MOVC and IAP restrictions for APROM are as follows:
<table>
<tr><td>Operation</td><td>Is it operable?</td></tr>
<tr><td>LDROM MOVC</td><td>x</td></tr>
<tr><td>APROM MOVC</td><td>√</td></tr>
<tr><td>LDROM IAP</td><td>×</td></tr>
<tr><td>APROM IAP</td><td>√</td></tr>
</table>
2. The MOVC and IAP restrictions for LDROM are as follows:
<table>
<tr><td>Operation</td><td>Is it operable?</td></tr>
<tr><td>LDROM MOVC</td><td>√</td></tr>
<tr><td>APROM MOVC</td><td>√</td></tr>
<tr><td>LDROM IAP</td><td>×</td></tr>
<tr><td>ALL APROM IAP, not restricted by IAPRANGE</td><td>√</td></tr>
</table>
</td></tr>
<tr><td><b>OP_CTM1[4]</b></td><td><b>DISJTG</b></td><td><b>R/W</b></td><td><b>IO/JTAG Port switching control</b>
0: JTAG mode is enabled, P1.1 and P1.3 can only be used as tCK/tDIO.</td></tr>
</table>

			1: Normal mode (Normal), JTAG function is invalid.
OP_CTM1[3~2]	**IAPS[1:0]**	**R/W**	**IAP spatial range selection** 00: All Flash ROM not allows IAP operation 01: Last 1K Flash ROM allows IAP operation 10: Last 2K Flash ROM allows IAP operation 11: All Flash ROM allows IAP operation Note： **1. The preceding Settings are invalid in BootLoader mode, and the BootLoader program can perform IAP operations on the entire APROM region** **2. LDROM does not allows IAP operation**
OP_CTM1[1:0]	**LDSIZE[1:0]**	**Read only**	**LDROM space range selection** 00: None LDROM, the APROM address is 0000H to 7FFFH 01: The last 1K APROM area of the Flash ROM is LDROM, and the APROM address is 0000H-7BFFh 10: The last 2K APROM area of the Flash ROM is LDROM, and the APROM address is 0000H-77FFh 11: The last 4K APROM area of the Flash ROM is LDROM, and the APROM address is 0000H-6FFFh **NOTE: LDROM not allow IAP operation in anyways**
OP_HRCR	**OP_HRCR[7:0]**	**R/W**	**HRC frequency change register** User can change the high-frequency oscillator frequency f_{HRC} by modifying the value of this register, and then change the system clock frequency fsys: 1. The initial value of OP_HRCR[7: 0] after power-on OP_HRCR[s] is a fixed value to ensure that f_{HRC} is 32MHz, OP_HRCR[s] of each IC may be different 2. When the initial value is OP_HRCR[s], the system clock frequency fsys of the IC can be set to an accurate 32/16/8/2.66MHz through the Option item. When OP_HRCR [7: 0] changes by 1, the fsys frequency changes by about 0.18% The relationship between OP_HRCR [7: 0] and fsys output frequency is as follows:

OP_HRCR [7: 0] Value	fsys actual output frequency (32M as an example)
OP_HRCR [s]-n	32000*(1-0.18%*n)kHz
...	....
OP_HRCR [s]-2	32000*(1-0.18%*2) = 31 884.8kHz
OP_HRCR [s]-1	32000*(1-0.18%*1) = 31 942.4kHz
OP_HRCR [s]	32000kHz
OP_HRCR [s]+1	32000*(1+0.18%*1) = 32 057.6kHz
OP_HRCR [s]+2	32000*(1+0.18%*2) = 32 115.2kHz
...	...
OP_HRCR [s]+n	32000*(1+0.18%*n)kHz

5.9.1 Customer-Option-related Registers Operation Instructions

Option-related SFRs reading and writing operations are controlled by both OPINX and OPREG registers, with their respective position of Option SFR depending on OPINX and its value written to option-related SFR depending on register OPREG:

Symbol	Address	Instructions		POR
OPINX	FEH	Option pointer	OPINX[7: 0]	00000000b
OPREG	FFH	Option register	OPREG[7: 0]	nnnnnnnnb

The OPINX register stores the address of the related OPTION register when operating the Option related SFR, and the OPREG register stores the corresponding value.

For example: To set ENWDT (OP_CTM0.7) to 1, the specific operation method is as follows:

C language example:

```
OPINX = 0xC1;          // Write the address of OP_CTM0 to the OPINX register
OPREG |= 0x80;         // Set 1 for OP_CTM0.7
```

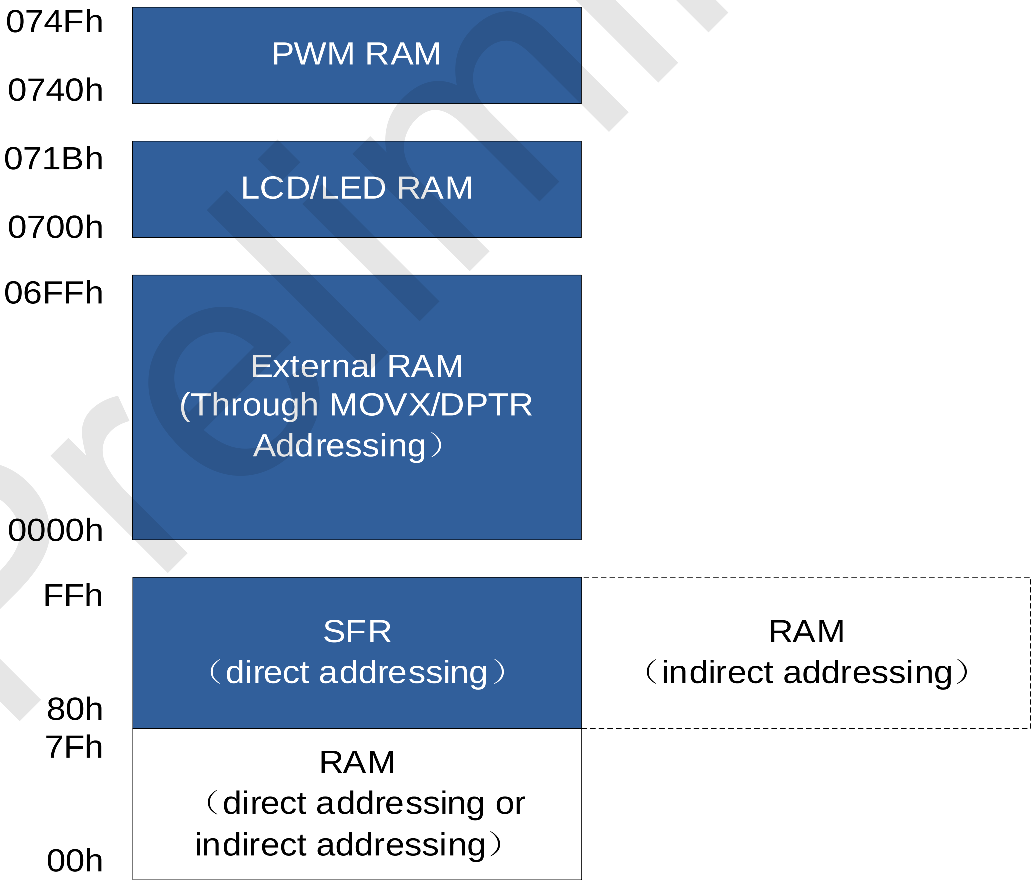
Assembly language example:

```
MOV OPINX,#C1H         ; Write the address of OP_CTM0 to the OPINX register
ORL OPREG,#80H         ; Set 1 for OP_CTM0.7
```

Note: It is forbidden to write any value beyond SFR address of Customer Option region into OPINX register! Or else, it may cause abnormal system operation.

5.10 SRAM

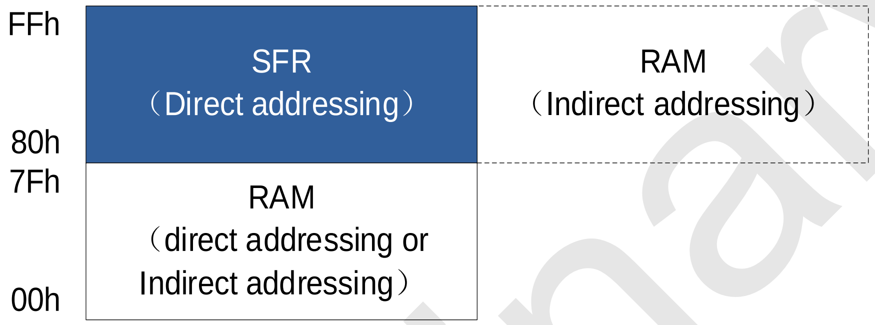
The SRAM of SC92F859X is divided into internal 256 bytes RAM and external 1792 bytes RAM.
The address range of the internal RAM is 00H to FFH, where high 128 bytes (address 80H to FFH) can only be addressed indirectly, and low 128 bytes (address 00H to 7FH) can be addressed directly or indirectly.
The address of special function register SFR is also 80H~FFH. However, the SFR differs from the internal high 128 bytes SRAM in that the SFR registers are addressed directly, while the internal high 128 bytes SRAM can only be addressed indirectly.
The address of the external RAM is 0000H~06FFH, but is addressed by MOVX instruction.



SRAM Structure Diagram

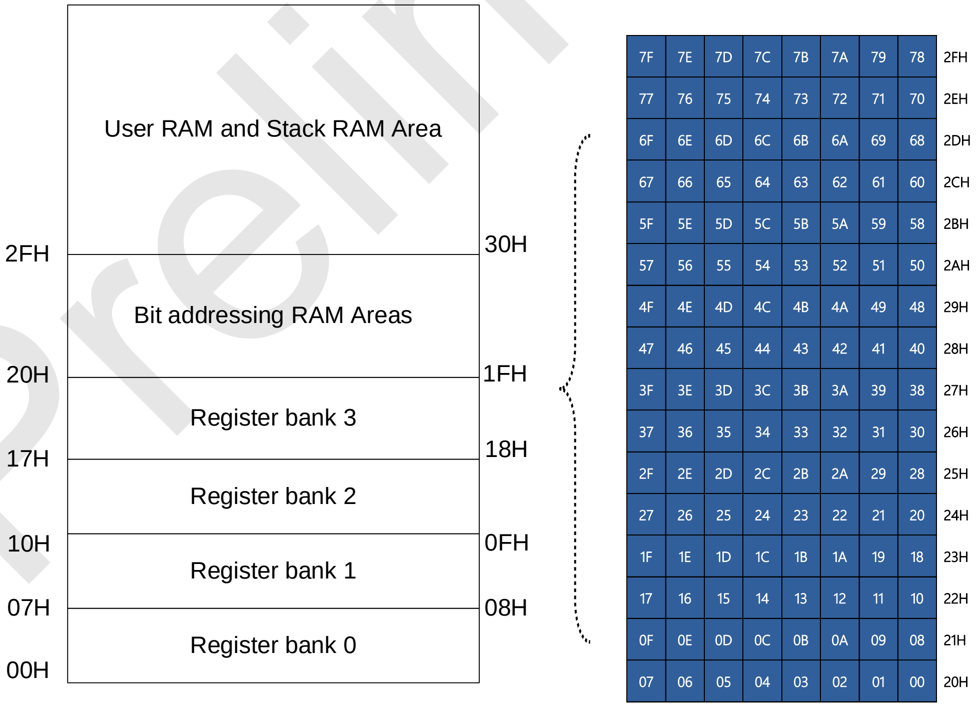
5.10.1 Internal 256 bytes SRAM

The internal low 128 bytes SRAM area can be divided into three parts: ① Operating register groups 0~3, address 00H~1FH, the combination of RS0 and RS1 in the program state word register PSW determines the current operating register, using operating register groups 0~3 can speed up the operation speed; ② Bit addressable area 20H~2FH, this area can be used as normal RAM or as bit-addressable RAM; When addressing by bit, the bit address is 00H~7FH, (this address is encoded by bit, different from the general SRAM encoded by byte address), the program can be distinguished by instruction; ③ User RAM and stack area, SC92F859X after reset, the 8-bit stack pointer points to the stack area, the user will generally set the initial value in the initialization program, it is recommended to set in E0H~FFH unit interval.



Internal 256 bytes RAM Structure Diagram

Internal low 128 bytes RAM structure as the following：



7F	7E	7D	7C	7B	7A	79	78	2FH
77	76	75	74	73	72	71	70	2EH
6F	6E	6D	6C	6B	6A	69	68	2DH
67	66	65	64	63	62	61	60	2CH
5F	5E	5D	5C	5B	5A	59	58	2BH
57	56	55	54	53	52	51	50	2AH
4F	4E	4D	4C	4B	4A	49	48	29H
47	46	45	44	43	42	41	40	28H
3F	3E	3D	3C	3B	3A	39	38	27H
37	36	35	34	33	32	31	30	26H
2F	2E	2D	2C	2B	2A	29	28	25H
27	26	25	24	23	22	21	20	24H
1F	1E	1D	1C	1B	1A	19	18	23H
17	16	15	14	13	12	11	10	22H
0F	0E	0D	0C	0B	0A	09	08	21H
07	06	05	04	03	02	01	00	20H

SRAM Structure Diagram

5.10.2 External 1792 bytes SRAM

External 1792 bytes RAM can be accessed via MOVX @DPTR, A; It is also possible to use MOVX A, @Ri, or MOVX @Ri, A in conjunction with the EXADH register to access the external 1792 bytes RAM: the EXADH register holds the high address of the external SRAM, and the Ri register holds the low 8-bit address of the external SRAM.

EXADH (F7H) External SRAM Operation Address High Bit (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	-	-	EXADH [3: 0]			
POR	x	x	x	x	0	0	0	0

Bit number	Bit Mnemonic	Description
3~0	**EXADH [3: 0]**	High-bit of external SRAM operation address
7~4	**-**	reserved

5.10.3 PWM&LCD 44 BYTES SRAM

The 0740H to 074FH of the RAM address is used as the 16-bytes PWM SRAM. For specific operation methods, refer to 13.2 PWM General Configuration Register.

The 0700H to 071BH of the RAM address is used as the 28-bytes PWM SRAM. For specific operation methods, refer to 15.2 LCD/LED Display RAM Configuration.

6 Special Function Register (SFR)

6.1 SFR Mapping

The SC92F859X provides some registers equipped with special functions, called SFR. The addresses of these registers are located at 80H~FFH, some are bit-addressable, and others are not. It is very convenient for these bit addressable registers to change the value of single bit, of which the address is end up with figure “0” or “8”. All SFR shall use direct addressing for addressing.

The SC92F859X SFR Map is as follows:

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
F8h	-	-	-	-	CHKSUML	CHKSUMH	OPINX	OPREG
F0h	B	IAPKEY	IAPADL	IAPADH	IAPADE	IAPDAT	IAPCTL	EXADH
E8h	-	EXA0	EXA1	EXA2	EXA3	EXBL	EXBH	OPERCON
E0h	ACC	-	-	-	-	-	-	-
D8h	P5	P5CON	P5PH	-	-	-	-	-
D0h	PSW	-	-	PWMCON	PWMCFG	-	-	-
C8h	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2	BTMCON	WDTCON
C0h	P4	P4CON	P4PH	-	-	-	INT2F	INT2R
B8h	IP	IP1	INT0F	INT0R	INT1F	INT1R	-	-
B0h	P3	P3CON	P3PH	P3VO	-	-	CMPCFG	CMPCON
A8h	IE	IE1	ADCCFG2	ADCCFG0	ADCCFG1	ADCCON	ADCVL	ADCVH
A0h	P2	P2CON	P2PH	P2VO	-	-	-	-
98h	SCON	SBUF	P0CON	P0PH	P0VO	SSCON0	SSCON1	SSDAT
90h	P1	P1CON	P1PH	DDRCON	P1VO	SSCON2	IOHCON0	IOHCON1

88h	TCON	TMOD	TL0	TL1	TH0	TH1	TMCON	OTCON
80h	P0	SP	DPL	DPH	-	-	-	PCON
	Bit addressable	Non-bit addressable						

Note: The empty part of the SFR register are not recommended for users.

6.2 SFR Instructions

Mnemonic	Add	Description	7	6	5	4	3	2	1	0	POR
P0	80H	P0 port data register	P07	P06	P05	P04	P03	P02	P01	P00	00000000b
SP	81H	Stack pointer	SP[7:0]								00000111b
DPL	82H	DPTR data pointer low	DPL[7:0]								00000000b
DPH	83H	DPTR data pointer high	DPH[7:0]								00000000b
PCON	87H	Power management control register	SMOD	-	-	-	RST	-	STOP	IDL	0xxx0x00b
TCON	88H	Timer control register	TF1	TR1	TF0	TR0	IE1	-	IE0	-	00000x0xb
TMOD	89H	Timer operating mode register	-	C/T1	M11	M01	-	C/T0	M10	M00	x000x000b
TL0	8AH	Timer 0 low 8 bits	TL0[7:0]								00000000b
TL1	8BH	Timer 1 low 8 bits	TL1[7:0]								00000000b
TH0	8CH	Timer 0 high 8 bits	TH0[7:0]								00000000b
TH1	8DH	Timer 1 high 8 bits	TH1[7:0]								00000000b
TMCON	8EH	Timer frequency control register	-	-	-	-	-	T2FD	T1FD	T0FD	xxxxx000b
OTCON	8FH	Output control register	SSMOD[1:0]		-	-	VOIRS[1:0]		SCS	BIAS	00xx0000b
P1	90H	P1 port data register	P17	P16	P15	P14	P13	P12	P11	P10	00000000b
P1CON	91H	P1 port input/output control register	P1C7	P1C6	P1C5	P1C4	P1C3	P1C2	P1C1	P1C0	00000000b
P1PH	92H	P1 port pull-up resistor control register	P1H7	P1H6	P1H5	P1H4	P1H3	P1H2	P1H1	P1H0	00000000b
DDRCON	93H	Display drive control register	DDRON	DMOD	DUTY[1:0]		VLCD[3:0]				00000000b
P1VO	94H	P1 port display driver output register	P17VO	P16VO	P15VO	P14VO	P13VO	P12VO	P11VO	P10VO	00000000b
SSCON2	95H	SSI control register 2	SSCON2[7:0]								00000000b
IOHCON0	96H	IOH setting register 0	P1H[1:0]		P1L[1:0]		P0H[1:0]		P0L[1:0]		00000000b
IOHCON1	97H	IOH setting register 1	-	-	P3L[1:0]		P2H[1:0]		P2L[1:0]		xx000000b
SCON	98H	Serial control register	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00000000b
SBUF	99H	Serial data buffer register	SBUF[7:0]								00000000b
P0CON	9AH	P0 port input/output control register	P0C7	P0C6	P0C5	P0C4	P0C3	P0C2	P0C1	P0C0	00000000b
P0PH	9BH	P0 port pull-up resistor control register	P0H7	P0H6	P0H5	P0H4	P0H3	P0H2	P0H1	P0H0	00000000b
P0VO	9CH	P0 port display driver output register	P07VO	P06VO	P05VO	P04VO	P03VO	P02VO	P01VO	P00VO	00000000b
SSCON0	9DH	SSI control register 0	SSCON0[7:0]								00000000b
SSCON1	9EH	SSI control register 1	SSCON1[7:0]								00000000b
SSDAT	9FH	SSI data register	SSDAT[7:0]								00000000b
P2	A0H	P2 port data register	P27	P26	P25	P24	P23	P22	P21	P20	00000000b
P2CON	A1H	P2 port input/output control register	P2C7	P2C6	P2C5	P2C4	P2C3	P2C2	P2C1	P2C0	00000000b
P2PH	A2H	P2 port pull-up resistor control register	P2H7	P2H6	P2H5	P2H4	P2H3	P2H2	P2H1	P2H0	00000000b
P2VO	A3H	P2 port display driver output register	P27VO	P26VO	P25VO	P24VO	P23VO	P22VO	P21VO	P20VO	00000000b
IE	A8H	Interrupt enable register	EA	EADC	ET2	EUART	ET1	EINT1	ET0	EINT0	00000000b
IE1	A9H	Interrupt enable register 1	-	-	ECMP	ETK	EINT2	EBTM	EPWM	ESSI	xx000000b

Mnemonic	Add	Description	7	6	5	4	3	2	1	0	POR
ADCCFG2	AAH	ADC setting register 2	-	-	-	-	-	LOWSP	ADCCK[1:0]		xxxxx000b
ADCCFG0	ABH	ADC setting register 0	EAIN7	EAIN6	EAIN5	EAIN4	EAIN3	EAIN2	EAIN1	EAIN0	00000000b
ADCCFG1	ACH	ADC setting register 1	EAIN15	EAIN14	EAIN13	EAIN12	EAIN11	EAIN10	EAIN9	EAIN8	00000000b
ADCCON	ADH	ADC control register	ADCEN	ADCS	EOC/ ADCIF	ADCIS[4:0]					00000000b
ADCVL	AEH	ADC result register	ADCV[3:0]				-	-	-	-	1111xxxxb
ADCVH	AFH	ADC result register	ADCV[11:4]								11111111b
P3	B0H	P3 port data register	P37	P36	P35	P34	P33	P32	P31	P30	00000000b
P3CON	B1H	P3 port input/output control register	P3C7	P3C6	P3C5	P3C4	P3C3	P3C2	P3C1	P3C0	00000000b
P3PH	B2H	P3 port pull-up resistor control register	P3H7	P3H6	P3H5	P3H4	P3H3	P3H2	P3H1	P3H0	00000000b
P3VO	B3H	P3 port display driver output register	P37VO	P36VO	P35VO	P34VO	P33VO	P32VO	P31VO	P30VO	00000000b
CMPCFG	B6H	Analog comparator setting register	-	-	-	CMPP	CMPIM[1:0]		CMPIS[1:0]		xxx00000b
CMPCON	B7H	Analog comparator control register	CMPEN	CMPIF	CMPSTA	-	CMPRF[3:0]				000x0000b
IP	B8H	Interrupt priority control register	-	IPADC	IPT2	IPUART	IPT1	IPINT1	IPT0	IPINT0	x0000000b
IP1	B9H	Interrupt priority control register 1	-	-	IPCMP	IPTK	IPINT2	IPBTM	IPPWM	IPSSI	xx000000b
INT0F	BAH	INT0 falling edge interrupt control register	INT0F7	INT0F6	INT0F5	INT0F4	-	-	-	-	0000xxxxb
INT0R	BBH	INT0 rising edge interrupt control register	INT0R7	INT0R6	INT0R5	INT0R4	-	-	-	-	0000xxxxb
INT1F	BCH	INT1 falling edge interrupt control register	INT1F7	INT1F6	INT1F5	INT1F4	INT1F3	INT1F2	INT1F1	INT1F0	00000000b
INT1R	BDH	INT1 rising edge interrupt control register	INT1R7	INT1R6	INT1R5	INT1R4	INT1R3	INT1R2	INT1R1	INT1R0	00000000b
P4	C0H	P4 port data register	P47	P46	P45	P44	P43	P42	P41	P40	00000000b
P4CON	C1H	P4 port input/output control register	P4C7	P4C6	P4C5	P4C4	P4C3	P4C2	P4C1	P4C0	00000000b
P4PH	C2H	P4 port pull-up resistor control register	P4H7	P4H6	P4H5	P4H4	P4H3	P4H2	P4H1	P4H0	00000000b
INT2F	C6H	INT2 falling edge interrupt control register	-	-	-	-	INT2F3	INT2F2	INT2F1	INT2F0	xxxx0000b
INT2R	C7H	INT2 rising edge interrupt control register	-	-	-	-	INT2R3	INT2R2	INT2R1	INT2R0	xxxx0000b
T2CON	C8H	Timer 2 control register	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	00000000b
T2MOD	C9H	Timer 2 operating mode register	-	-	-	-	-	-	T2OE	DCEN	xxxxxx00b
RCAP2L	CAH	Timer 2 reload low 8 bits	RCAP2L[7:0]								00000000b
RCAP2H	CBH	Timer 2 reload high 8 bits	RCAP2H[7:0]								00000000b
TL2	CCH	Timer 2 low 8 bits	TL2[7:0]								00000000b
TH2	CDH	Timer 2 high 8 bits	TH2[7:0]								00000000b
BTMCON	CEH	Low frequency timer control register	ENBTM	BTMIF	-	-	BTMFS[3:0]				00x0000b
WDTCON	CFH	WDT control register	-	-	-	CLRWDT	-	WDTCKS[2:0]			xxx0x000b
PSW	D0H	Program status word register	CY	AC	F0	RS1	RS0	OV	F1	P	00000000b
PWMCON	D3H	PWM control register	PWMPD[7:0]								00000000b
PWMCFG	D4H	PWM setting register	ENPWM	PWMIF	PWMCK[1:0]		PWMPD[11:8]				00000000b
P5	D8H	P5 port data register	-	-	P55	P54	P53	P52	P51	P50	xx000000b
P5CON	D9H	P5 port input/output control register	-	-	P5C5	P5C4	P5C3	P5C2	P5C1	P5C0	xx000000b
P5PH	DAH	P5 port pull-up resistor control register	-	-	P5H5	P5H4	P5H3	P5H2	P5H1	P5H0	xx000000b
ACC	E0H	accumulator	ACC[7:0]								00000000b
EXA0	E9H	Extended Accumulator 0	EXA[7:0]								00000000b
EXA1	EAH	Extended Accumulator 1	EXA[15:8]								00000000b
EXA2	EBH	Extended Accumulator 2	EXA[23:16]								00000000b
EXA3	ECH	Extended Accumulator 3	EXA[31:24]								00000000b
EXBL	EDH	Extended B register L	EXB[7:0]								00000000b
EXBH	EEH	Extended B register H	EXB[15:8]								00000000b
OPERCON	EFH	Arithmetic control register	OPERS	MD	-	-	-	-	-	CHKSUMS	00xxxxx0b
B	F0H	B register	B[7:0]								00000000b

Mnemonic	Add	Description	7	6	5	4	3	2	1	0	POR
IAPKEY	F1H	Data protection register	IAPKEY[7:0]								00000000b
IAPADL	F2H	IAP write address low register	IAPADR[7:0]								00000000b
IAPADH	F3H	IAP write address high register	-	IAPADR[14:8]							x0000000b
IAPADE	F4H	IAP write to extended address register	IAPADER[7:0]								00000000b
IAPDAT	F5H	IAP data register	IAPDAT[7:0]								00000000b
IAPCTL	F6H	IAP control register	BTLD	-	SERASE	PRG	-	-	CMD[1:0]		0x00xx00b
EXADH	F7H	High-bit address of external SRAM operation address	-	-	-	-	-	EXADH [2:0]			xxxxx000b
CHKSUML	FCH	Check Sum Result register low level	CHKSUML[7:0]								00000000b
CHKSUMH	FDH	Check Sum Result register high level	CHKSUMH[7:0]								00000000b
OPINX	FEH	Customer Option pointer	OPINX[7:0]								00000000b
OPREG	FFH	Customer Option register	OPREG[7:0]								nnnnnnnnb

6.2.1 PWM Duty Cycle Adjustment Register

Add	7	6	5	4	3	2	1	0	POR
740H	ENP40	INV40	-	-	PDT40[11:8]				00000000b
741H	PDT40[7:0]								00000000b
742H	ENP41	INV41	-	-	PDT41[11:8]				00000000b
743H	PDT41[7:0]								00000000b
744H	ENP42	INV42	-	-	PDT42[11:8]				00000000b
745H	PDT42[7:0]								00000000b
746H	ENP43	INV43	-	-	PDT43[11:8]				00000000b
747H	PDT43[7:0]								00000000b
748H	ENP50	INV50	-	-	PDT50[11:8]				00000000b
749H	PDT50[7:0]								00000000b
74AH	ENP51	INV51	-	-	PDT51[11:8]				00000000b
74BH	PDT51[7:0]								00000000b
74CH	ENP52	INV52	-	-	PDT52[11:8]				00000000b
74DH	PDT52[7:0]								00000000b
74EH	ENP53	INV53	-	-	PDT53[11:8]				00000000b
74FH	PDT53[7:0]								00000000b

6.2.2 LCD/LED Display RAM Configuration

Add	7	6	5	4	3	2	1	0
	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0
700H	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0
701H	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1
702H	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2
703H	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3
704H	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4
705H	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5
706H	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6
707H	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7
708H	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8
709H	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9
70AH	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10
70BH	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11

70CH	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12
70DH	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13
70EH	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14
70FH	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15
710H	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16
711H	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17
712H	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18
713H	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19
714H	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20
715H	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21
716H	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22
717H	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23
718H	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24
719H	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25
71AH	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26
71BH	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27

7 Power, Reset And System Clock

7.1 Power Circuit

The SC92F859X power supply system includes BG, LDO, POR, LVR and other circuits, which can achieve reliable operation in the range of 2.0~5.5V. In addition, the IC has a built-in, accurate 2.048/1.024V voltage that can be used as an internal reference voltage for the ADC. Users can find the specific settings in the 18 high-speed analog-to-digital converter (ADC).

7.2 Power-on Reset

After the SC92F859X power-on, the processes carried out before execution of client software are as follows:

- Reset stage
- Loading information stage
- Normal operation stage

7.2.1 Reset Stage

The SC92F859X will always be in the reset mode, There will not be a valid clock until the voltage supplied to the SC92F859X is higher than certain voltage. The duration of reset stage is related to rising speed of external power. Once the external supply voltage is up to built-in POR voltage, the reset stage would be completed.

7.2.2 Loading Information Stage

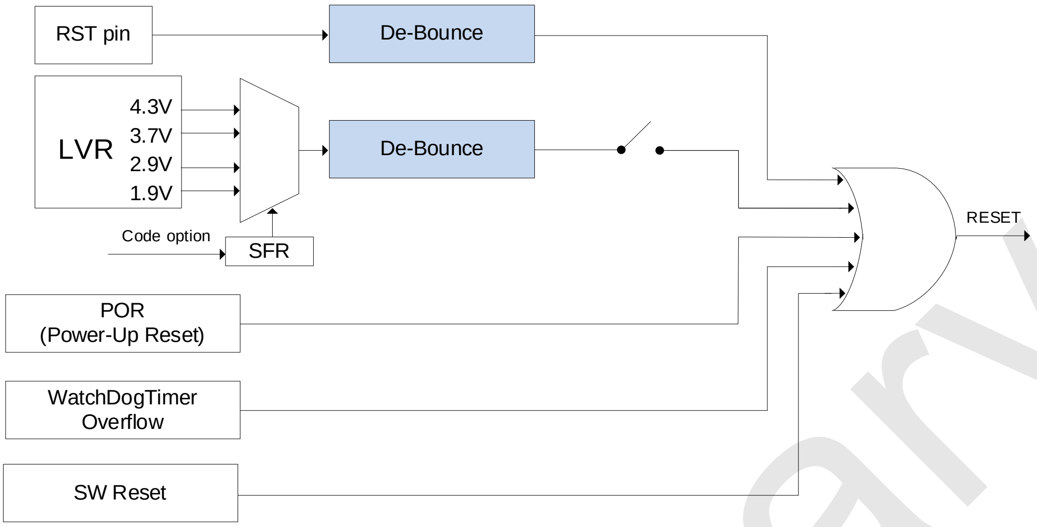
There is a warm-up counter inside The SC92F859X. During the reset stage, the warm-up counter is cleared to 0 until the voltage exceeds the POR voltage, the internal RC oscillator starts to oscillate, and the warm-up counter starts counting. When the internal warm-up counter counts to a certain number, every certain number of HRC clocks will read a byte of data from the IFB (including Customer Option) in the Flash ROM and store it in the internal system register. This reset signal will not end until the warm-up is completed.

7.2.3 Normal Operation Stage

After finishing the Loading Information stage, The SC92F859X starts to read the instruction code from Flash and enters the normal operation stage. The LVR voltage is the set value of Customer Option written by user.

7.3 Reset Modes

The SC92F859X has 5 reset methods: ① External reset ② Low-voltage reset LVR ③ Power-on reset POR ④ Watchdog WDT reset ⑤ Software reset. The circuit diagram of the reset part of the SC92F859X is as follows:

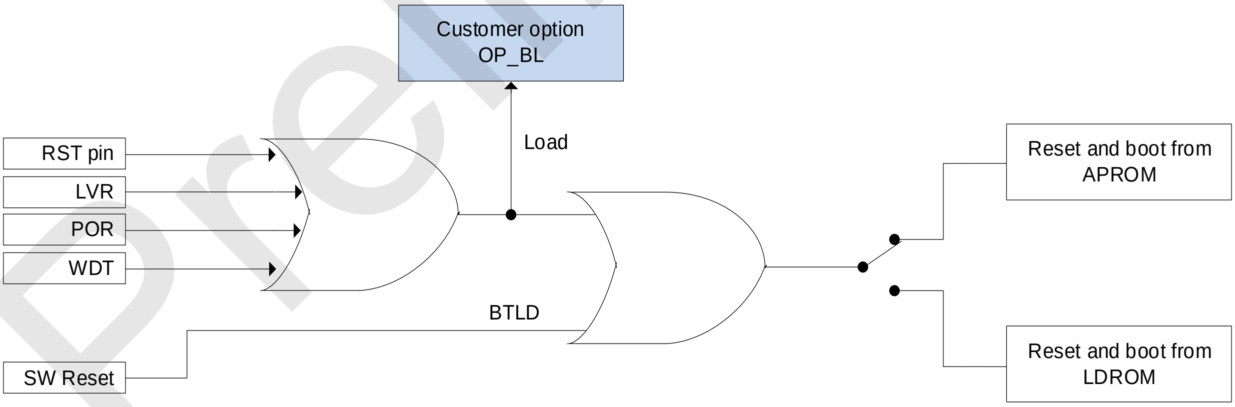


SC92F859X Reset circuit diagram

After reset the boot area：

After the external RST reset, low voltage reset LVR, power-on reset POR, watchdog WDT, the chip starts from the boot region (APROM/LDROM) set by the user OP_BL.

After the software is reset, the chip is started according to the boot region (APROM/LDROM) set by BTLD (IAPCTL.7).



SC92F859X’s boot area switch after reset

7.3.1 External Reset

External reset is a reset pulse signal of a certain width given to SC92F859X from external RST pin to realize the reset of SC92F859X. The user can configure the P5.2/RST pin as RST (reset pin) by Customer Option.

7.3.2 Low-voltage Reset LVR

The SC92F859X provides a low-voltage reset circuit. There are 4-level LVR voltage options: 4.3V, 3.7V, 2.9V, 1.9V. The default value is the Option value written by the user. When the VDD voltage is lower than the low voltage reset threshold and the duration is longer than the T_{LVR}, a reset occurs. Where, T_{LVR} is the buffeting time of LVR, which is about 30μs.

OP_CTM0(C1H@FFH) Customer Option Register 0 (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	ENWDT	ENXTL	SCLKS[1: 0]		DISRST	DISLVR	LVRS[1: 0]	
R/W	R/W	R/W	R/W		R/W	R/W	R/W	
POR	n	n	n		n	n	n	

Bit number	Bit Mnemonic	Description
2	**DISLVR**	LVR enable setting 0: LVR valid 1: LVR invalid
1~0	**LVRS [1: 0]**	LVR voltage threshold selection control 11: 4.3V 10: 3.7V 01: 2.9V 00: 1.9V

7.3.3 Power-on Reset (POR)

The SC92F859X has a power-on reset circuit inside. When the power supply voltage VDD reaches the POR reset voltage, the system automatically resets.

7.3.4 Watchdog Reset (WDT)

The SC92F859X has a WDT, the clock source of which is the internal 32k Hz LRC. The user can choose whether to enable the watchdog reset function by Customer Option.

OP_CTM0 (C1H@FFH) Customer Option Register 0 (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	ENWDT	ENXTL	SCLKS[1: 0]		DISRST	DISLVR	LVRS[1: 0]	
R/W	R/W	R/W	R/W		R/W	R/W	R/W	
POR	n	n	n		n	n	n	

Bit number	Bit Mnemonic	Description
7	**ENWDT**	WDT control bit (This bit is transferred by the system to the value set by the user Code Option) 1: WDT valid 0: WDT invalid

WDTCON (CFH) WDT Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	-	CLRWDT	-	WDTCKS[2: 0]		
R/W	-	-	-	R/W	-	R/W		
POR	x	x	x	0	x	0	0	0

Bit number	Bit Mnemonic	Description
4	**CLRWDT**	Clear WDT (Only valid when set to 1)

<table>
<tr><td></td><td></td><td>1: WDT counter restart, cleared by system hardware</td></tr>
<tr><td>2~0</td><td><b>WDTCKS [2: 0]</b></td><td>Watchdog clock selection
<table>
<tr><td>WDTCKS[2: 0]</td><td>WDT overflow time</td></tr>
<tr><td>000</td><td>500ms</td></tr>
<tr><td>001</td><td>250ms</td></tr>
<tr><td>010</td><td>125ms</td></tr>
<tr><td>011</td><td>62.5ms</td></tr>
<tr><td>100</td><td>31.5ms</td></tr>
<tr><td>101</td><td>15.75ms</td></tr>
<tr><td>110</td><td>7.88ms</td></tr>
<tr><td>111</td><td>3.94ms</td></tr>
</table></td></tr>
<tr><td>7~5,3</td><td><b>-</b></td><td>Reserved</td></tr>
</table>

7.3.5 Software Reset

PCON (87h) Power Management Control Register (write only, *unreadable*)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	SMOD	-	-	-	RST	-	STOP	IDL
R/W	Write only	-	-	-	Write only	-	Write only	Write only

POR	0	x	x	x	n	x	0	0

Bit number	Bit Mnemonic	Description
3	**RST**	Software reset control bit: Write status: 0: The program runs normally; 1: The CPU resets immediately after this bit is written to "1"

7.3.6 Register Reset Value

When The SC92F859X is in reset state, most registers will return to their initial state. The watchdog（WDT) is turned off. The initial value of the program counter PC is 0000h, and the initial value of the stack pointer SP is 07h. The "hot restart" Reset (such as WDT, LVR, software reset, etc.) will not affect the SRAM, and the SRAM value is always the value before the reset. The loss of SRAM content will occur when the power supply voltage is so low that the RAM cannot be saved.

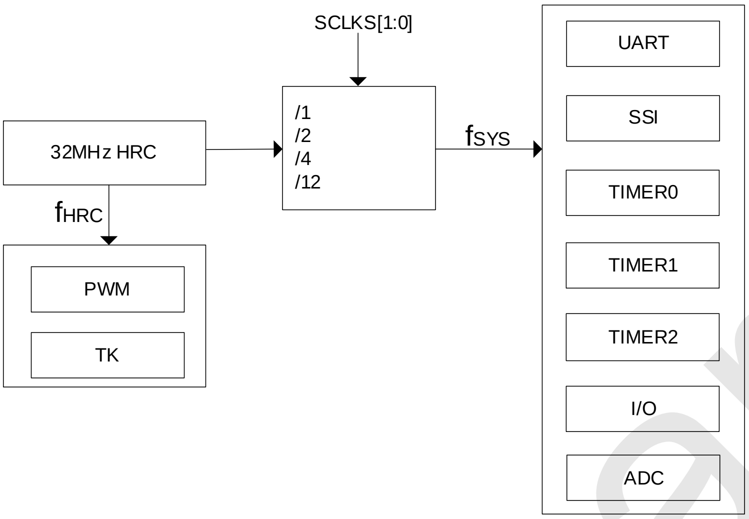
The SFR register POR，please see the 6.2 SFR Instructions for the detailed information.

7.4 High-speed RC Oscillator

The SC92F859X has a built-in high-precision high-frequency oscillator (HRC) with adjustable oscillation frequency. The HRC is accurately adjusted to 32MHz@5V/25℃ at the factory. Users can set the system clock to 32/16/ 8/2.66MHz through the Customer Option when programming. The adjustment process is to filter out the effect of the deviation on the accuracy. This HRC will have a certain drift under the influence of the ambient temperature and operating voltage. For pressure drift (2.0~5.5V) and (-20~85℃) temperature drift generally within ±2%.

The HRC can be automatically calibrated by connecting an external 32.768kHz crystal oscillator. Users only need to connect an external 32.768kHz crystal oscillator, Users can set the external 32.768kHz crystal oscillator function through the Customer Opiton when programming. The HRC self-calibration function is automatically turned on after the IC is powered on.

Note: The clock source of the touch circuit is fixed as fHRC = 32MHz, which will not change with the switch of internal and external system clocks



SC92F859X Internal clock relationship

OP_CTM0 (C1H@FFH) Customer Option Register 0 (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	ENWDT	ENXTL	SCLKS[1: 0]		DISRST	DISLVR	LVRS[1: 0]	
R/W	R/W	R/W	R/W		R/W	R/W	R/W	
POR	n	n	n		n	n	n	

Bit number	Bit Mnemonic	Description
5~4	**SCLKS[1: 0]**	System clock frequency selection bits 00: System clock frequency is HRC frequency divided by 1; 01: System clock frequency is HRC frequency divided by 2;

		10: System clock frequency is HRC frequency divided by 4; 11: System clock frequency is HRC frequency divided by 12;

The SC92F859X has a special function: the user can modify the value of SFR to adjust the HRC frequency within a certain range. The user can achieve this by configuring the OP_HRCR register. Note: HRC can be automatically calibrated by connecting a 32.768kHz crystal oscillator. Therefore, if the user uses the 32.768kHz external crystal oscillator function, the HRC frequency will always be corrected to 32MHz. At this time, adjusting OP_HRCR cannot change the HRC frequency.

OP_HRCR (83h@FFH) System Clock Change Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	OP_HRCR[7: 0]							
R/W	R/W							
POR	n	n	n	n	n	n	n	n

Bit number	Bit Mnemonic	Description
7~0	**OP_HRCR[7: 0]**	**HRC frequency change register** The user can change the high-frequency oscillator frequency f_{HRC} by modifying the value of this register, and then change the system clock frequency fsys of the IC: 1. The initial value of OP_HRCR[7: 0] after power-on OP_HRCR[s] is a fixed value to ensure that f_{HRC} is 32MHz, OP_HRCR[s] of each IC may be different 2. When the initial value is OP_HRCR[s], the system clock frequency fsys of the IC can be set to an accurate 32/16/8/4MHz through the Option item. When OP_HRCR [7: 0] changes by 1, the fsys frequency changes by about 0.18% The relationship between OP_HRCR [7: 0] and fsys output frequency is as follows:

		OP_HRCR [7: 0] value	fsys actual output frequency (32M as an example)
		OP_HRCR [s]-n	32000*(1-0.18%*n)kHz
		...	....
		OP_HRCR [s]-2	32000*(1-0.18%*2) = 31 884.8kHz
		OP_HRCR [s]-1	32000*(1-0.18%*1) = 31 942.4kHz
		OP_HRCR [s]	32000kHz
		OP_HRCR [s]+1	32000*(1+0.18%*1) = 32 057.6kHz
		OP_HRCR [s]+2	32000*(1+0.18%*2) = 32 115.2kHz
		...	...
		OP_HRCR [s]+n	32000*(1+0.18%*n)kHz

Note:

1. **After each power-on of the IC, the value of OP_HRCR[7: 0] is the value of the high-frequency oscillator frequency f_{HRC} closest to 32MHz; the user can correct the value of HRC after each power-on to allow the system clock frequency fsys of the IC to work at Frequency required by users;**

2. **In order to ensure the reliable operation of the IC, the maximum operating frequency of the IC should not exceed 10% of 32MHz, that is 35.2MHz;**

3. **Please confirm that the change of HRC frequency will not affect other functions.**

7.5 Low-speed RC Oscillator and Low-speed Clock Timer

The SC92F859X has a built-in RC and 32.768kHz crystal oscillator circuit with a frequency of 32kHz, which can be used as the clock source of the Base Timer. The oscillator is directly connected to a Base Timer, which can wake the CPU from STOP mode and generate an interrupt.

BTMCON (FBH) Low-frequency Timer Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	ENBTM	BTMIF	-	-	BTMFS[3: 0]			
R/W	R/W	R/W	-	-	R/W			
POR	0	0	x	x	0	0	0	0

Bit number	Bit Mnemonic	Description
7	**ENBTM**	Low frequency Base Timer start control 0: Base Timer and its clock source do not start 1: Base Timer and its clock source start
6	**BTMIF**	Base Timer interrupt application flag When the CPU accepts the Base Timer interrupt, this flag will be automatically cleared by hardware.
3~0	**BTMFS [3: 0]**	Low frequency clock interrupt frequency selection 0000: An interrupt is generated every 15.625ms 0001: An interrupt is generated every 31.25ms 0010: An interrupt is generated every 62.5ms 0011: An interrupt is generated every 125ms 0100: An interrupt is generated every 0.25 seconds 0101: An interrupt is generated every 0.5 seconds 0110: An interrupt is generated every 1.0 seconds 0111: An interrupt is generated every 2.0 seconds 1000: An interrupt is generated every 4.0ms 1001: An interrupt is generated every 8.0 seconds

		1010: An interrupt is generated every 16.0 seconds 1011: An interrupt is generated every 32.0 seconds 1100~1111: reserved
5~4	-	reserved

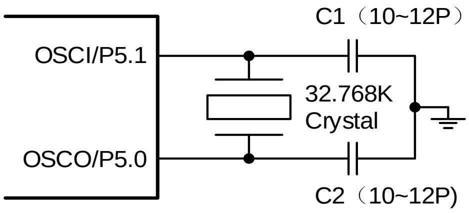
OP_CTM0 (C1H@FFH) Customer Option Register 0 (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	ENWDT	ENXTL	SCLKS[1: 0]		DISRST	DISLVR	LVRS[1: 0]	
R/W	R/W	R/W	R/W		R/W	R/W	R/W	
POR	n	n	n		n	n	n	

Bit number	Bit Mnemonic	Description
6	**ENXTL**	External 32k crystal selector switch 0: The external 32k crystal is off, P5.0 and P5.1 are valid, and the internal LRC is valid; 1: The external 32k crystal is turned on, P5.0 and P5.1 are invalid, and the internal LRC is invalid.

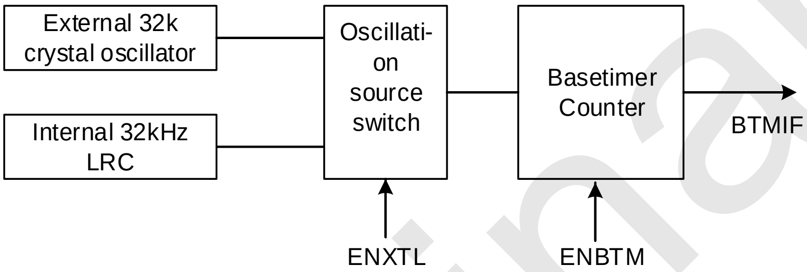
Note: HRC can be automatically calibrated by connecting a 32.768kHz crystal oscillator. Therefore, if the user uses the 32.768kHz external crystal oscillator function, the HRC frequency will always be corrected to 32MHz. At this time, adjusting OP_HRCR cannot change the HRC frequency.

The connection circuit used by P5.0/P5.1 external 32k oscillator as BaseTimer is as follows:



32k external crystal connection diagram

The internal and external oscillation selection relationship of Base Timer is as follows:



Base Timer Structure diagram

7.6 Power Saving Modes

The SC92F859X supports two different software selectable power-reducing modes: IDLE and STOP. These modes are accessed through the PCON register.

Setting the PCON.1 bit enters STOP mode. STOP mode stops the internal high-frequency oscillator in order to minimize power consumption. In STOP mode, users can wake up the SC92F859X through external interrupts INT0~INT2, low-frequency clock interrupt and WDT, or STOP through external reset.

Setting the PCON.0 bit enters IDLE mode. In IDLE mode the program stops running and all CPU states are saved before entering IDLE mode. IDLE mode can be woken up by any interrupt.

PCON (87H) Power Management Control Register (read/write) (write only, *not readable*)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	SMOD	-	-	-	RST	-	STOP	IDL
R/W	Write only	-	-	-	Write only	-	Write only	Write only
POR	0	x	x	x	n	x	0	0

Bit number	Bit Mnemonic	Description
1	**STOP**	STOP mode bit. 0：Normal operation mode 1: Energy saving mode, high frequency oscillator stops working, low frequency oscillator and WDT can choose to work or not according to the setting.
0	**IDL**	IDLE mode bit. 0：Normal operation mode 1: Energy saving mode, the program stops running, but external devices continue running in a timely manner. All CPU states are saved before entering IDLE mode.

Notes:

When Configuring MCU to enter STOP or IDLE mode, the instruction of configuring PCON register should be followed by 8 “NOP” instructions rather than other instructions. Or else, it will be unable to execute following instructions normally after wake-up!

For example: set MCU to enter STOP mode:

Example in C Language

```
#include”intrins.h”

PCON |= 0x02;  // PCON bit1 STOP bit write 1, configure the MCU to enter STOP mode

_nop_();                    // At least 8 _nop_() are required

_nop_();

_nop_();

_nop_();

_nop_();

_nop_();

_nop_();

_nop_();
```

……

Assembly Language:

```
ORL PCON,#02H                  ; PCON bit1 STOP bit write 1, configure the MCU to enter STOP mode
NOP                     ; At least 8 NOPs are requiredNOP
NOP
NOP
NOP
NOP
NOP
NOP
```

……

8 CPU and Instruction Set

8.1 CPU

The SC92F859X is built around an enhanced super-high-speed 1T 8051 core, and its instructions are fully compatible with classic 8051 core.

8.2 Addressing Mode

The addressing modes of 1T 8051 CPU instructions of the SC92F859X are: ①Immediate Addressing ② Direct Addressing ③ Indirect Addressing ④ Register Addressing ⑤ Relative Addressing ⑥ Indexed Addressing ⑦ Bit Addressing.

8.2.1 Immediate Addressing

Immediate addressing is also called immediate data. It directly gives the operands participating in the operation in the instruction operand. Examples of instructions are as follows:

MOV A, #50H (This instruction moves the immediate value 50H to accumulator A)

8.2.2 Direct Addressing

In direct addressing mode, the instruction operand field gives the address of the operand to participate in the operation. The direct addressing mode can only be used to represent special function registers, internal data registers, and bit address spaces. The special function registers and bit address spaces can only be accessed by direct addressing.

Examples are as follows:

ANL 50H, #91H

(indicating that the number in the 50H unit is ANDed with the immediate 91H, and the result is stored in the 50H unit. 50H is direct address, representing a unit in the internal data register RAM.)

8.2.3 Indirect Addressing

Indirect addressing is indicated by adding the "@" symbol before R0 or R1. Assuming that the data in R1 is 40H, and the data in the internal data memory 40H unit is 55H, the instruction is

MOV A, @R1 (Move data 55H to accumulator A).

8.2.4 Register Addressing

When register addressing, operate on the selected operating registers R7~R0, accumulator A, general register B, address register and carry C. Registers R7~R0 are represented by the low three bits of the instruction code, and ACC, B, DPTR and carry bit C are implicitly contained in the instruction code. Therefore, register addressing also includes an implicit addressing method. The selection of the register operating area is determined by RS1 and RS0 in the program status word register PSW. The register specified by the instruction operand refers to the register in the current operating area.

INC R0 Refers to(R0)+1→R0

8.2.5 Relative Addressing

Relative addressing is to add the current value in the program counter PC to the number given by the second byte of the instruction, and the result is used as the branch address of the branch instruction. The branch address also becomes the branch destination address, the current value in the PC becomes the base address, and the number given by the second byte of the instruction becomes the offset. Since the destination address is relative to the base address in the PC, this addressing method becomes relative addressing. The offset is a signed number, and the range that can be expressed is -128~+127. This addressing method is mainly used for branch instructions.

```
JC $+50H
```

It means that if the carry bit C is 0, the content in the program counter PC does not change, that is, it does not transfer. If the carry bit C is 1, the current value as base address in the PC plus the offset 50H will be used as the destination address of the branch instruction.

8.2.6 Indexed Addressing

In the indexed addressing mode, the instruction operand specifies an index register that stores the index base address. In indexed addressing, the offset is added to the index base value, and the result is used as the address of the operand. The index registers are the program counter PC and the address register DPTR.

```
MOVC A, @A+DPTR
```

It indicates that the accumulator A is an offset register, and its content is added to the content of the address register DPTR. The result is used as the address of the operand, and the number in this unit is taken out and sent to the accumulator A.

8.2.7 Bits Addressing

Bit addressing refers to the addressing mode when performing bit operations on some internal data memory RAMs and special function registers that can perform bit operations. When performing bit operations, with the help of carry bit C as a bit operation accumulator, the instruction operand directly gives the address of the bit, and then performs bit operation on the bit according to the nature of the opcode. The bit address is exactly the same as the byte address encoding method in direct byte addressing, which is mainly distinguished by the nature of the operation instruction, and special attention should be paid when using it.

MOV C, 20H (The value of the bit manipulation register with address 20H is sent to carry bit C)

9 Interrupts

SC92F859X provides 13 interrupt sources: TIMER 0~2, INT0~2, ADC, PWM, UART, SSI, BASE TIMER, TK, CMP. The 13 interrupt sources are divided into two interrupt priorities and can be set to either high or low priority separately. Three external interrupts can be set as up, down or both trigger conditions for each interrupt source respectively. Each interrupt has its own priority setting bit, interrupt flag, interrupt vector and enable bit respectively. The total enable bit EA can open or close all interrupts.

9.1 Interrupt Source and Vector

The list of the SC92F859X interrupt sources, interrupt vectors, and related control bits are as follows:

Interrupt Source	Interrupt condition	Interrupt Flag	Interrupt Enable Control	Interrupt Priority Control	Interrupt Vector	Query Priority	Interrupt Number (C51)	Flag Clear Mode	Capability of Waking up STOP
INT0	External interrupt 0 conditions are met	IE0	EINT0	IPINT0	0003H	1 (high)	0	H/W Auto	YES
Timer 0	Timer 0 overflow	TF0	ET0	IPT0	000BH	2	1	H/W Auto	NO
INT1	External interrupt 1 conditions are met	IE1	EINT1	IPINT1	0013H	3	2	H/W Auto	YES
Timer 1	Timer 1 overflow	TF1	ET1	IPT1	001BH	4	3	H/W Auto	NO
UART	Receive or send completed	RI/TI	EUART	IPUART	0023H	5	4	Must user Clear	NO
Timer 2	Timer 2 overflow	TF2	ET2	IPT2	002BH	6	5	Must user Clear	NO
ADC	ADC conversio n completed	ADCIF	EADC	IPADC	0033H	7	6	Must user Clear	NO
SSI	Receive or send completed	SPIF/TWIF	ESSI	IPSPI	003BH	8	7	Must user Clear	NO

PWM	PWM overflow	PWMIF	EPWM	IPPWM	0043H	9	8	Must user Clear	NO
BTM	Base timer overflow	BTMIF	EBTM	IPBTM	004BH	10	9	H/W Auto	YES
INT2	External interrupt 2 conditions are met	-	EINT2	IPINT2	0053H	11	10	-	YES
TK	Touch Key counter overflowed	TKIF	ETK	IPTK	005BH	12	11	H/W Auto	YES
CMP	Comparato-r interrupt condition met	CMPIF	ECMP	IPCMP	0063H	13	12	Must user Clear	YES

Under the circumstance where the master interrupt control bit EA and the respective interrupt control bit have been enable, the interrupt occurrence is shown below:

Timer Interrupt: Interrupt generates when Timer 0 or Timer 1 overflows and the interrupt flag TF0 or TF1 is set to “1”. When the microcontroller unit responds to the timer interrupt, the interrupt flag TF0 or TF1 is reset automatically by hardware. Interrupt generates when Timer 2 overflows and the interrupt flag TF2 is set to “1”. Once Timer 2 interrupt generates, the hardware would not automatically clear TF2 bit, which must be cleared by the user’s software.

UART Interrupt: When UART0 completes receiving or transmitting a frame of data, bit RI or TI will be set to “1” automatically by hardware, and UART interrupt occurs. Once UART interrupt occurs, the hardware would not automatically clear up RI/TI bit, which shall be cleared by user’s software.

ADC Interrupt: After ADC conversion is completed, ADC interrupt generates, whose interrupt flag is the ADC conversion completion flag EOC/ADCIF (ADCCON.5). When user starts ADCS conversion, EOC will be reset automatically by hardware. Once conversion completes, EOC would be set to “1” automatically by hardware. User should clear the ADC interrupt flag by software when the interrupt service routine is executed after ADC interrupt generates.

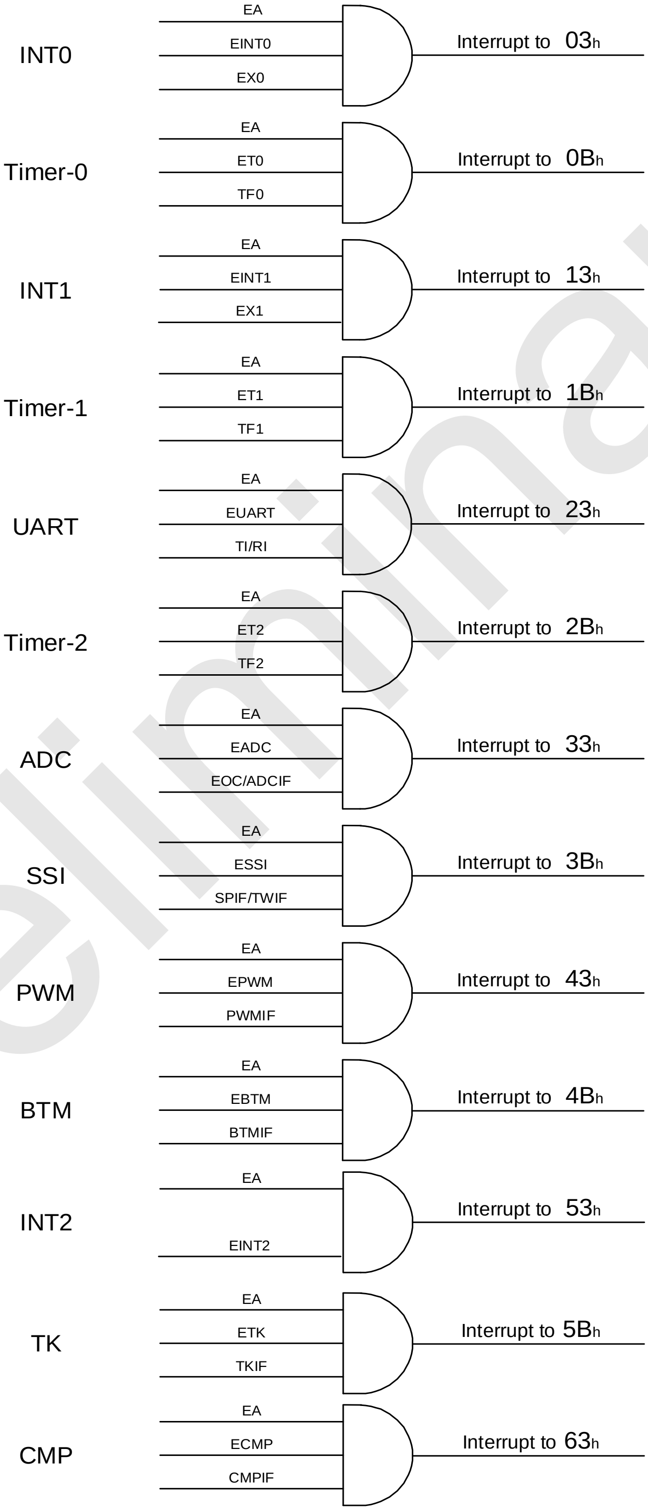
SSI Interrupt: When SSI completes receiving or transmitting a frame of data, SPIF/TWIF bit will be set to “1” automatically by hardware, and SSI interrupt generates. When the microcontroller unit serves SSI interrupt, the interrupt flag SPIF/TWIF must be cleared by software.

PWM Interrupt: When PWM counter overflows (beyond PWMPD), the flag will be set as 1 automatically by hardware. Meanwhile, if the PWM interrupt control bit IE1[1] (EPWM) is set as 1, PWM interrupt will occurs. Once PWM interrupt occurs, the hardware would not clear the interrupt flag automatically, which shall be cleared by user’s software.

External Interrupt INT0 ~ 2: When any external interrupt pin meets the interrupt conditions, external interrupt generates. The external interrupt INT0 and INT1 would set up interrupt flag IE0 and IE1 respectively, which will be automatically cleared by hardware rather than user. User can set the priority level of each interrupt through IP register. Besides, external interrupt INT0 ~ 2 can also wake up STOP mode of microcontroller unit.

9.2 Interrupt Structure Diagram

The interrupt structure of SC92F859X is shown below:



SC92F859X Interrupt structure and vector

9.3 Interrupt Priority

The interrupt of SC92F859X microcontroller has two interrupt priorities. The request of these interrupt sources can be programmed as interrupt of high priority or interrupt of low priority, so that the nested interrupt service programs of two levels can be realized. An ongoing low-priority interrupt can be interrupted by a higher-priority interrupt request, but not by another interrupt request of the same priority. The interrupt is executed until the end. When the return instruction RETI is encountered, the new interrupt request can be responded to by executing another instruction after the main program is returned.

Sach as：

- Low-priority interrupts can be interrupted by higher-priority interrupt requests, and vice versa.
- Any interrupt must not be interrupted by interrupt requests of the same priority during the response.

Interrupt query order: If there are several interrupts of the same priority in SC92F859X microcontroller, the priority order of interrupt response is the same as the interrupt query number in C51, that is, the small query number will give priority to response, and the large query number will respond slowly.

9.4 Interrupt Processing Flow

When an interrupt is generated and responded by the CPU, the main program execution is interrupted and the following operations will be performed:

- The currently executing instruction is finished;
- The PC value is pushed into the stack to protect the scene;
- The interrupt vector address is loaded into the program counter PC;
- Execute the corresponding interrupt service program;
- The interrupt service routine ends and RETI;
- Unstack the PC value and return to the program before the interruption.

In this process, the system will not immediately execute other interrupts of the same priority, but will retain the interrupt request that has occurred, and after the current interrupt processing is completed, go to execute a new interrupt request.

9.5 Interrupt-related SFR Registers

IE (A8H) Interrupt Enable Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	EA	EADC	ET2	EUART	ET1	EINT1	ET0	EINT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7	**EA**	Interrupt enable total control 0: Close all interrupts 1: Enable all interrupts
6	**EADC**	ADC interrupt enable control 0: Disable ADC interrupt 1: Allow the ADC to generate an interrupt when the conversion is complete
5	**ET2**	Timer 2 interrupt enable control 0: Disable Timer 2 interrupt 1: Enable Timer 2 interrupt
4	**EUART**	UART interrupt enable control 0: Disable UART interrupt 1: Allow UART interrupt
3	**ET1**	Timer 1 interrupt enable control 0: Disable Timer 1: Enable Timer 1 interrupt
2	**EINT1**	External interrupt 1 enable control 0: close INT1 interrupt 1: Enable INT1 interrupt
1	**ET0**	Timer 0 interrupt enable control

		0: Disable TIMER0 interrupt 1: Enable TIMER0 interrupt
0	**EINT0**	External interrupt 0 enable control 0: close INT0 interrupt 1: Enable INT0 interrupt

IP (B8H) Interrupt Priority Control Register (Read/Write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	IPADC	IPT2	IPUART	IPT1	IPINT1	IPT0	IPINT0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
6	**IPADC**	ADC interrupt priority selection 0: ADC interrupt priority is low 1: ADC interrupt priority is high
5	**IPT2**	Timer 2 interrupt priority selection 0: Timer 2 interrupt priority is low 1: Timer 2 interrupt priority is high
4	**IPUART**	UART interrupt priority selection 0: UART interrupt priority is low 1: UART interrupt priority is high

3	**IPT1**	Timer 1 interrupt priority selection 0: Timer 1 interrupt priority is low 1: Timer 1 interrupt priority is high
2	**IPINT1**	INT1 counter interrupt priority selection 0: INT1 interrupt priority is low 1: INT1 interrupt priority is high
1	**IPT0**	Timer 0 interrupt priority selection 0: Timer 0 interrupt priority is low 1: Timer 0 interrupt priority is high
0	**IPINT0**	INT0 counter interrupt priority selection 0: INT0 interrupt priority is low 1: INT0 interrupt priority is high
7	**-**	Reserved

IE1 (A9H) Interrupt Enable Register 1 (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	ECMP	ETK	EINT2	EBTM	EPWM	ESSI
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description

5	**ECMP**	Analog comparator interrupt enable control 0: Disable the analog comparator interrupt 1: Open the analog comparator interrupt
4	**TK**	Touch Key interrupts enable control 0: Turn off Touch Key interrupt 1: Open Touch Key interrupt
3	**EINT2**	External interrupt 2 enable control 0: close INT2 interrupt 1: Open INT2 interrupt
2	**EBTM**	Base Timer interrupt enable control 0: Disable Base Timer interrupt 1: Enable Base Timer interrupt
1	**EPWM**	PWM interrupt enable control 0: Disable PWM interrupt 1: Enable interrupt when PWM count overflows
0	**ESSI**	SSI interrupt enable control 0: Disable serial port interrupt 1: Allow serial port interrupt

IP1 (B9H) Interrupt Priority Control Register 1 (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	IPCMP	IPTK	IPINT2	IPBTM	IPPWM	IPSSI
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W

POR	x	x	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
5	**IPCMP**	Analog comparator interrupt priority selection 0: Analog comparator interrupt priority is low 1: Analog comparator interrupt priority is high
4	**IPTK**	Touch Key interrupts priority selection 0: Touch Key interrupt priority is low 1: Touch Key interrupt priority is high
3	**IPINT2**	INT2 counter interrupt priority selection 0: INT2 interrupt priority is low 1: INT2 interrupt priority is high
2	**IPBTM**	Base Timer interrupt priority selection 0: Base Timer interrupt priority is low 1: Base Timer interrupt priority is high
1	**IPPWM**	PWM interrupt enable selection 0: PWM interrupt priority is low 1: PWM interrupt priority is high
0	**IPSSI**	Three-in-one serial port SSI0 interrupt priority selection 0: SSI interrupt priority is low 1: SSI interrupt priority is high

TCON (88H) Timer Control Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	TF1	TR1	TF0	TR0	IE1	-	IE0	-
R/W	R/W	R/W	R/W	R/W	R/W	-	R/W	-
POR	0	0	0	0	0	x	0	x

Bit number	Bit Mnemonic	Description
3	**IE1**	INT1 overflow interrupt request flag. INT1 generates an overflow. When an interrupt occurs, the hardware sets IE1 to "1" and applies for an interrupt. When the CPU responds, the hardware clears "0".
1	**IE0**	INT0 overflow interrupt request flag. INT0 generates an overflow. When an interrupt occurs, the hardware sets IE0 to "1" and applies for an interrupt. When the CPU responds, the hardware clears "0".
2,0	**-**	Reserved

INT0F (B4H) INT0 Falling Edge Interrupt Control Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	INT0F7	INT0F6	INT0F5	INT0F4	-	-	-	-
R/W	R/W	R/W	R/W	R/W	-	-	-	-
POR	0	0	0	0	x	x	x	x

Bit number	Bit Mnemonic	Description

7~4	**INT0Fn** **(n=7~4)**	INT0 falling edge interrupt control 0: INT0n falling edge interrupt close 1: INT0n falling edge interrupt enable
3~0	**-**	Reserved

INT0R (BBH) INT0 Rising Edge Interrupt Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	INT0R7	INT0R6	INT0R5	INT0R4	-	-	-	-
R/W	R/W	R/W	R/W	R/W	-	-	-	-
POR	0	0	0	0	x	x	x	x

Bit number	Bit Mnemonic	Description
7~4	**INT0Rn** **(n=7~4)**	INT0 rising edge interrupt control 0: INT0n rising edge interrupt close 1: INT0n rising edge interrupt enable
3~0	**-**	Reserved

INT1F (BCH) INT1 Falling Edge Interrupt Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	INT1F7	INT1F6	INT1F5	INT1F4	INT1F3	INT1F2	INT1F1	INT1F0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7~0	**INT1Fn** **(n=7~0)**	INT1 falling edge interrupt control 0: INT1n falling edge interrupt close 1: INT1n falling edge interrupt enable

INT1R (BDH) INT1 Rising Edge Interrupt Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	INT1R7	INT1R6	INT1R5	INT1R4	INT1R3	INT1R2	INT1R1	INT1R0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7~0	**INT1Rn** **(n=7~0)**	INT1 rising edge interrupt control 0: INT1n rising edge interrupt off 1: INT1n rising edge interrupt enable

INT2F (BEH) INT2 Falling Edge Interrupt Control Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	-	-	-	INT2F3	INT2F2	INT2F1	INT2F0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
POR	x	x	x	x	0	0	0	0

Bit number	Bit Mnemonic	Description
3~0	**INT2Fn** **(n=3~0)**	INT2 falling edge interrupt control 0: INT2n falling edge interrupt close 1: INT2n falling edge interrupt enable
7~4	**-**	Reserved

INT2R (BFH) INT2 Rising Edge Interrupt Control Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	-	-	-	INT2R3	INT2R2	INT2R1	INT2R0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
POR	x	x	x	x	0	0	0	0

Bit number	Bit Mnemonic	Description
3~0	**INT2Rn**	INT2 rising edge interrupt control

	(n=3~0)	0: INT2n rising edge interrupt close 1: INT2n rising edge interrupt enable
7~4	**-**	Reserved

10 Timer/Counter T0 and T1

Timer 0 and Timer 1 inside the SC92F859X MCU are two 16-bit timers/counters. They have two operating modes: counting mode and timing mode. There is a control bit C/Tx in the special function register TMOD to select whether T0 and T1 are timers or counters. They are essentially an addition counter, but the source of the count is different. The source of the timer is the system clock or its divided clock, but the source of the counter is the input pulse of the external pin. Only when TRx=1, T0 and T1 will be opened to count.

In counter mode, for each pulse on the P0.2/T0 and P0.3/T1 pins, the count value of T0 and T1 increases by 1, respectively.

In the timer mode, the count source of T0 and T1 can be selected as fsys/12 or fsys through the special function register TMCON (fsys is the divided system clock).

There are 4 operating modes for timer/counter T0, and 3 operating modes for timer/counter T1 (mode 3 does not exist):

- Mode 0: 13-bit timer/counter mode
- Mode 1: 16-bit timer/counter mode
- Mode 2: 8-bit auto-reload mode
- Mode 3: Two 8-bit timer/counter modes

In the above modes, modes 0, 1, and 2 of T0 and T1 are the same, and mode 3 is different.

10.1 T0 and T1-related Registers

Symbol	Address	Description	7	6	5	4	3	2	1	0	POR
TCON	88H	Timer control register	TF1	TR1	TF0	TR0	IE1	-	IE0	-	00000x0xb
TMOD	89H	Timer operating mode register	-	C/T1	M11	M01	-	C/T0	M10	M00	x000x000b
TL0	8AH	Timer 0 low 8 bits	TL0[7: 0]								00000000b
TL1	8BH	Timer 1 low 8 bits	TL1[7: 0]								00000000b
TH0	8CH	Timer 0 high 8 bits	TH0[7: 0]								00000000b
TH1	8DH	Timer 1 high 8 bits	TH1[7: 0]								00000000b
TMCON	8EH	Timer frequency control register	-		-	-	-	T2FD	T1FD	T0FD	xxxxx000b

The explanation of each register is as follows:

TCON (88H) Timer Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	TF1	TR1	TF0	TR0	IE1	-	IE0	-
R/W	R/W	R/W	R/W	R/W	R/W	-	R/W	-
POR	0	0	0	0	0	x	0	x

Bit number	Bit Mnemonic	Description
7	**TF1**	T1 overflow interrupt request flag. T1 generates an overflow. When an interrupt occurs, the hardware sets TF1 to "1" and applies for an interrupt. When the CPU responds, the hardware clears "0".
6	**TR1**	Operation control bit of timer T1. This bit is set and cleared by software. When TR1=1, T1 is allowed to start counting. When TR1=0, T1 counting is prohibited.
5	**TF0**	T0 overflow interrupt request flag. T0 overflows. When an interrupt occurs, the hardware sets TF0 to "1" and applies for an interrupt. When the CPU responds, the hardware clears "0".
4	**TR0**	Operation control bit of timer T0. This bit is set and cleared by software. When TR0=1, T0 is allowed to start counting. When TR0=0, T0 counting is prohibited.
2,0	**-**	Reserved

TMOD (89H) Timer Operating Mode Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bit Mnemonic	-	C/T1	M11	M01	-	C/T0	M10	M00
R/W	-	R/W	R/W	R/W	-	R/W	R/W	R/W
POR	x	0	0	0	x	0	0	0
	T1				T0			

Bit number	Bit Mnemonic	Description
6	**C/T1**	TMOD[6] control timer 1 0: Timer, T1 count comes from fsys frequency division 1: Counter, T1 count comes from external pin T1/P0.3
5~4	**M11,M01**	Timer/Counter 1 mode selection 00: 13-bit timer/counter, the high 3-bits of TL1 are invalid 01: 16-bit timer/counter, TL1 and TH1 all are valid 10: 8-bit auto-reload timer, automatically reload the value stored in TH1 into TL1 when overflow 11: Timer/Counter 1 is invalid (stop counting)
2	**C/T0**	TMOD[2] control timer 0 0: Timer, T0 count comes from fsys frequency division 1: Counter, T0 count comes from external pin T0/P0.2
1~0	**M10,M00**	Timer/Counter 0 mode selection 00: 13-bit timer/counter, the high 3-bits of TL0 are invalid 01: 16-bit timer/counter, TL0 and TH0 all are valid 10: 8-bit auto-reload timer, automatically reload the value stored in TH0 into TL0 when overflow

		11: Timer 0 is now a dual 8-bit timer/counter. TL0 is an 8-bit timer/counter controlled by the control bits of standard timer 0; TH0 is only an 8-bit timer controlled by the control bits of timer 1.
7,3	-	Reserved

TMOD[0]~TMOD[2] in TMOD register is to set the operating mode of T0; TMOD[4]~TMOD[6] is to set the operating mode of T1.

The timer and counter Tx functions are selected by the control bits C/Tx of the special function register TMOD. M0x and M1x are used to select the Tx operating mode. TRx acts as the switch control of T0 and T1. Only when TRx=1, T0 and T1 are turned on.

TMCON (8EH) Timer Frequency Control Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	USMD2[1: 0]		-	-	-	-	T1FD	T0FD
R/W	R/W	R/W	-	-	-	-	R/W	R/W
POR	0	0	x	x	x	x	0	0

Bit number	Bit Mnemonic	Description
1	**T1FD**	T1 input frequency selection control 0: T1 frequency is derived from fsys/12 1: T1 frequency is derived from fsys
0	**T0FD**	T0 input frequency selection control 0: T0 frequency is derived from fsys/12 1: T0 frequency is derived from fsys

IE (A8H) Interrupt Enable Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	EA	EADC	ET2	EUART	ET1	EINT1	ET0	EINT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
3	**ET1**	Timer 1 interrupt enable control 0: Disable Timer 1 interrupt 1: Enable Timer 1 interrupt
1	**ET0**	Timer 0 interrupt enable control 0: Disable Timer 0 interrupt 1: Enable Timer 0 interrupt

IP (B8H) Interrupt Priority Control Register (Read/Write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	IPADC	IPT2	IPUART	IPT1	IPINT1	IPT0	IPINT0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description

3	**IPT1**	Timer 1 interrupt priority 0: Set the interrupt priority of Timer 1 to "Low" 1: Set the interrupt priority of Timer 1 to "High"
1	**IPT0**	Timer 0 interrupt priority 0: Set the interrupt priority of Timer 0 to "Low" 1: Set the interrupt priority of Timer 0 to "High"

10.2 T0 Operating Modes

By setting M10 and M00 (TMOD [1], TMOD [0]) in the register TMOD, timer/counter 0 can realize 4 different operating modes.

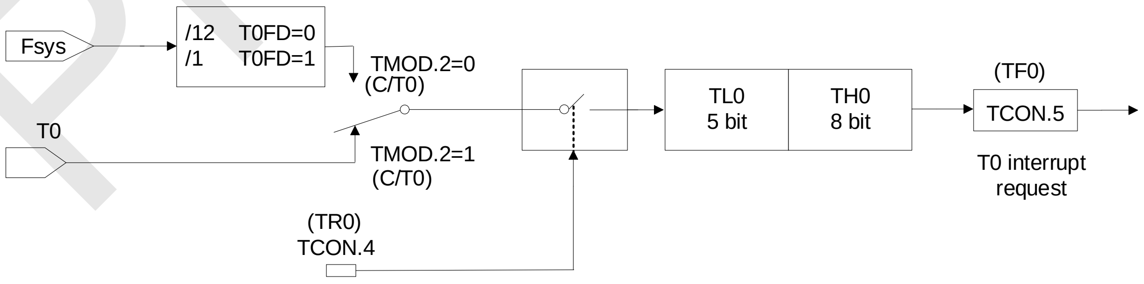
Operating Mode 0: 13-bit Counter/Timer

TH0 register stores the high 8-bits (TH0.7~TH0.0) of the 13-bit counter/timer, and the TL0 stores the low 5-bits (TL0.4~TL0.0). The high 3-bits of TL0 (TL0.7~TL0.5) are uncertain values and should be ignored when reading. When the 13-bit timer/counter overflows, the system will set the timer overflow flag TF0 to 1. If the timer 0 interrupt is enabled, an interrupt will be generated.

C/T0 bit selects the clock input source of the counter/timer. If C/T0=1, the level change of the timer 0 input pin T0 (P0.2) from high to low will increase the timer 0 data register by 1. If C/T0=0, select the frequency division of the system clock as the clock source of timer 0.

When TR0 is set to 1, the timer T0 is started. Setting TR0 does not forcibly reset the timer, meaning that if TR0 is set, the timer register will start counting from the value when TR0 was cleared last time. Therefore, before enabling the timer, the initial value of the timer register should be set.

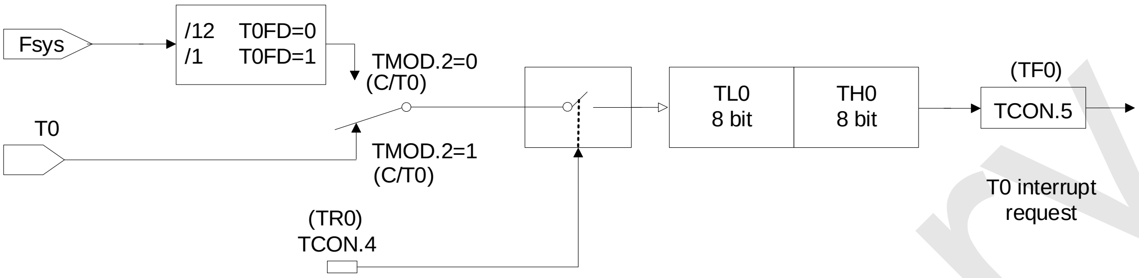
When applied as a timer, TOFD can be configured to select the frequency division ratio of the clock source.



Timer/counter operating mode 0: 13-bit timer/counter

Operating Mode 1: 16-bit Counter/Timer

Except for using a 16-bit (all 8-bit data of TL0 is valid) counters/timers, Mode 1 and Mode 0 operate in the same way. The way to open and configure the counter/timer is the same.



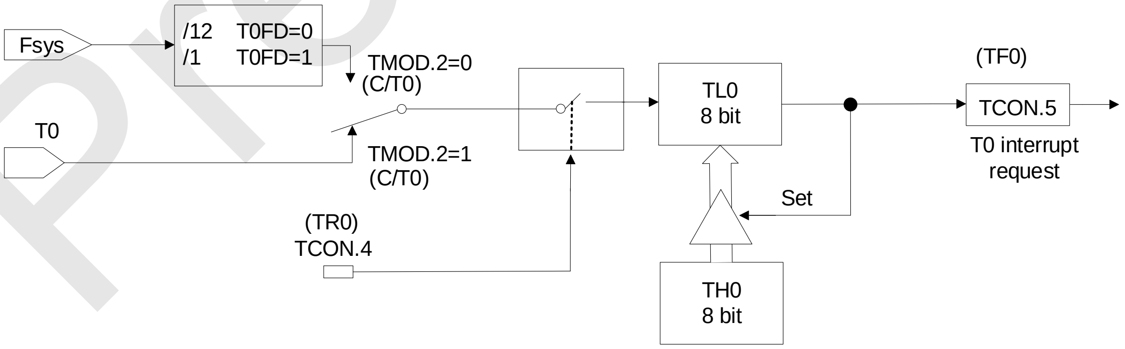
Timer/Counter Operating Mode 1: 16-bit Timer/Counter

Operating Mode 2: 8-bit Automatic Reload Counter/Timer

In operating mode 2, Timer 0 is an 8-bit auto-reload counter/timer. TL0 stores the count value, and TH0 stores the reload value. When the counter in TL0 overflows to 0x00, the timer overflow flag TF0 is set to 1, and the value of register TH0 is reloaded into register TL0. If the timer interrupt is enabled, an interrupt will be generated when TF0 is set to 1, but the reload value in TH0 will not change. Before allowing the timer to count correctly, TL0 must be initialized to the required value.

Except for the auto-reload function, the counter/timer in operating mode 2 is enabled and configured in the same way as in modes 0 and 1.

When used as a timer, the register TMCON.0 (T0FD) can be configured to select the ratio of the timer clock source divided by the system clock fsys.



Timer/counter operating mode 2: 8-bit timer/counter with automatic reload

Operating Mode 3: Two 8-bit Counters/Timers (Timer 0 Only)

In operating mode 3, Timer 0 is used as two independent 8-bit counters/timers, which are controlled by TL0 and TH0, respectively. TL0 is controlled by timer 0 control bits (in TCON) and status bits (in TMOD): TR0, C/T0, TF0. Timer 0 can select the timer mode or counter mode through T0 TMOD.2 (C/T0).

TH0 sets related control by timer 1 control TCON, but TH0 is only limited to timer mode and cannot be set to counter mode by TMOD.2 (C/T0). TH0 is enabled by the control of the timer control bit TR1, and TR1=1 needs to be set. When an overflow occurs and an interrupt is generated, TF1 will be set to 1, and the interrupt will be processed according to T1.

When T0 is set to operating mode 3, the TH0 timer occupies the interrupt resources of T1 and the registers in TCON, and the 16-bit counter of T1 will stop counting, which is equivalent to "TR1=0". When using the TH0 timer to work, set TR1=1.

10.3 T1 Operating Mode

By setting M11 and M01 (TMOD[5], TMOD[4]) in the register TMOD, timer/counter 1 can realize three different operating modes.

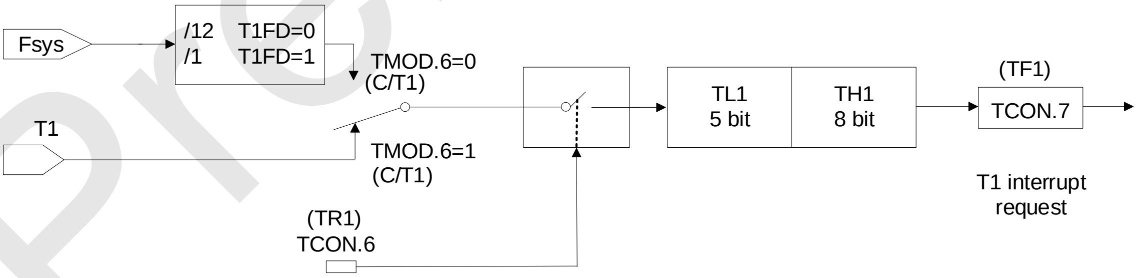
Operating mode 0: 13-bit Timer/Counter

The TH1 register stores the high 8-bits (TH1.7~TH1.0) of the 13-bit counter/timer; the TL1 stores the low 5 bits (TL1.4~TL1.0). The high 3-bits of TL1 (TL1.7~TL1.5) are uncertain values and should be ignored when reading. When the 13-bit timer counter increments and overflows, the system sets the timer overflow flag TF1 to 1. If Timer 1 interrupt is enabled, an interrupt will be generated. The C/T1 bit selects the clock source of the counter/timer.

If C/T1=1, the level of timer 1 input pin T1 (P0.3) changes from high to low, which will increase the timer 1 data register by 1. If C/T1=0, select the frequency division of the system clock as the clock source of timer 1.

Set TR1 to enable the timer. Setting TR1 does not forcibly reset the timer, meaning that if TR1 is set to 1, the timer register will start counting from the value when TR1 was cleared to 0 last time. Therefore, before enabling the timer, the initial value of the timer register should be set.

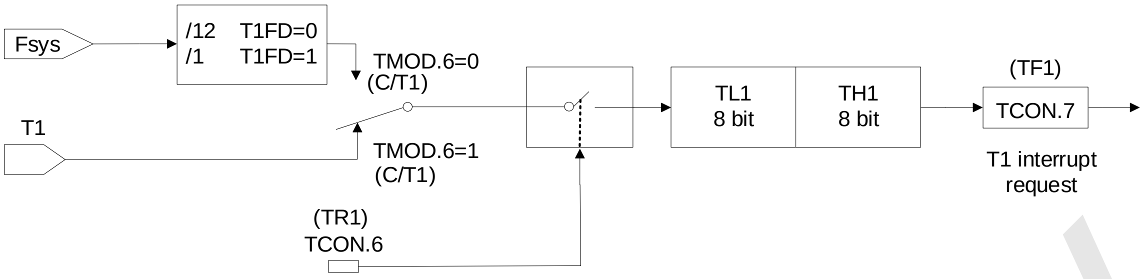
When applied as a timer, T1FD can be configured to select the frequency division ratio of the clock source.



Timer/counter operating mode 0: 13-bit timer/counter

Operating mode 1: 16-bit Counter/Timer

Except for using a 16-bit (all 8-bit data of TL1 is valid) counter/timer, Mode 1 and Mode 0 operate in the same way. The way to open and configure the counter/timer is the same.



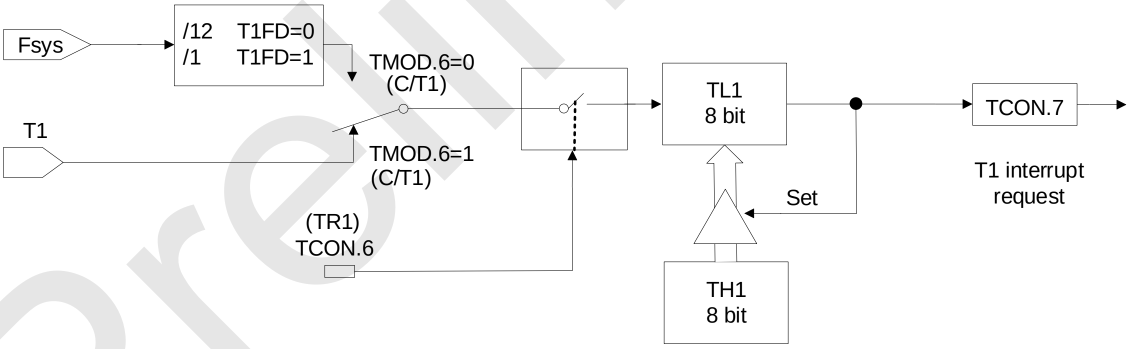
Timer/counter operating mode 1: 16-bit timer/counter

Operating mode 2: 8-bit Automatic Reload Counter/Timer

In operating mode 2, Timer 1 is an 8-bit auto-reload counter/timer. TL1 stores the count value, and TH1 stores the reload value. When the counter in TL1 overflows to 0x00, the timer overflow flag TF1 is set to 1, and the value of register TH1 is reloaded into register TL1. If the timer interrupt is enabled, an interrupt will be generated when TF1 is set to 1, but the reload value in TH1 will not change. Before allowing the timer to count correctly, TL1 must be initialized to the required value.

Except for the auto-reload function, the counter/timer in operating mode 2 is enabled and configured in the same way as modes 0 and 1.

When used as a timer, the register TMCON.1 (T1FD) can be configured to select the ratio of the timer clock source divided by the system clock fsys.



Timer/counter operating mode 2: 8-bit timer/counter with automatic reload

11 Timer/Counter T2

SC92F859X internal Timer2 has two operating modes: counting mode and timing mode. The special function register T2CON has a control bit C/T2 to select whether T2 is a timer or counter. They are essentially an addition counter, just from different sources. The source of the timer is the system clock or its divider clock, but the source of the counter is the input pulse of the external pin. TR2 is the on/off control of T2 counting in timer/counter mode. T2 is only turned on when TR2=1.

In counter mode, for each pulse on the T2 pin, the T2 count increases by 1.

In timer mode, the special function register TMCON can be used to select whether the source of T2 count is Fsys/12 or Fsys.

Timer/counter T2 has 4 operating modes:

- mode 0: 16-bit capture mode
- mode 1: automatically reloads a 16-bit timer mode
- mode 2: The Baudrate generator mode
- mode 3: the programmable clock output mode.

11.1 T2-related Registers

Symbol	Address	Description	7	6	5	4	3	2	1	0	POR
T2CON	C8H	Timer 2 control register	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	00000000b
T2MOD	C9H	Timer 2 operating mode register	-	-	-	-	-	-	T2OE	DCEN	xxxxxx00b
RCAP2L	CAH	Timer 2 reload low 8 bits	RCAP2L [7:0]								00000000b
RCAP2H	CBH	Timer 2 reload high 8 bits	RCAP2H [7:0]								00000000b
TL2	CCH	Timer 2 low 8 bits	TL2[7:0]								00000000b
TH2	CDH	Timer 2 high 8 bits	TH2[7:0]								00000000b
TMCON	8EH	Timer frequency control register	-	-	-	-	-	T2FD	T1FD	T0FD	xxxxx000b

T2CON (C8H) Timer 2 Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7	**TF2**	Timer 2 overflow flag 0: No overflow (must be cleared by software) 1: Overflow (if RCLK = 0 and TCLK = 0, set by hardware 1)
6	**EXF2**	Flag bit detected by external event input (falling edge) of T2EX pin 0: No external event input (must be cleared by software) 1: External input detected (if EXEN2 = 1, set by hardware)
5	**RCLK**	UART0 receive clock control bit 0: Timer 1 generates the receive baud rate 1: Timer 2 generates the receive baud rate
4	**TCLK**	UART0 transmit clock control bit 0: Timer 1 generates transmission baud rate 1: Timer 2 generates transmission baud rate
3	**EXEN2**	T2EX pin is used as a reload/capture trigger enable/disable control: 0: Ignore events on T2EX pin

		1: When Timer 2 is not used as the UART0 clock, a falling edge on the T2EX pin is detected, and a capture or reload is generated
2	**TR2**	Timer 2 start/stop control bit 0: stop timer 2 1: Start timer 2
1	**C/T2**	Timer 2 Timer/counter mode selection positioning 2 0: Timer mode, T2 pin is used as I/O port 1: Counter mode
0	**CP/RL2**	Capture/reload mode selection positioning 0: 16-bit timer/counter with reload function 1: 16-bit timer/counter with capture function, T2EX is timer 2 external capture signal input port

T2MOD (C9H) Timer 2 Operating Mode Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	-	-	-	-	T2OE	DCEN
R/W	-	-	-	-	-	-	R/W	R/W
POR	x	x	x	x	x	x	0	0

Bit number	Bit Mnemonic	Description
1	**T2OE**	Timer 2 output enable bit 0: Set T2 as clock input or I/O port 1: Set T2 as the clock output

0	**DCEN**	Count down enable bit 0: Timer 2 is prohibited as an up/down counter, Timer 2 is only used as an up counter 1: Allow Timer 2 as an up/down counter
7~2	**-**	Reserved

TMCON (8EH) Timer Frequency Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	-	-	-	T2FD	T1FD	T0FD
R/W	-	-	-	-	-	R/W	R/W	R/W
POR	x	x	x	x	x	0	0	0

Bit number	Bit Mnemonic	Description
1	**T2FD**	T2 input frequency selection control 0: T2 frequency is derived from fsys/12 1: T2 frequency is derived from fsys

IE (A8H) Interrupt Enable Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	EA	EADC	ET2	EUART	ET1	EINT1	ET0	EINT0

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
5	**ET2**	Timer 2 interrupt enable control 0: Disable Timer 2 interrupt 1: Enable Timer 2 interrupt

IP (B8H) Interrupt Priority Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	IPADC	IPT2	IPUART	IPT1	IPINT1	IPT0	IPINT0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
5	**IPT2**	Timer 2 interrupt priority 0: Set the interrupt priority of Timer 2 to "Low" 1: Set the interrupt priority of Timer 2 to "High"

11.2 Timer 2 Operating Mode

The operating mode and configuration mode of timer 2 are as follows:

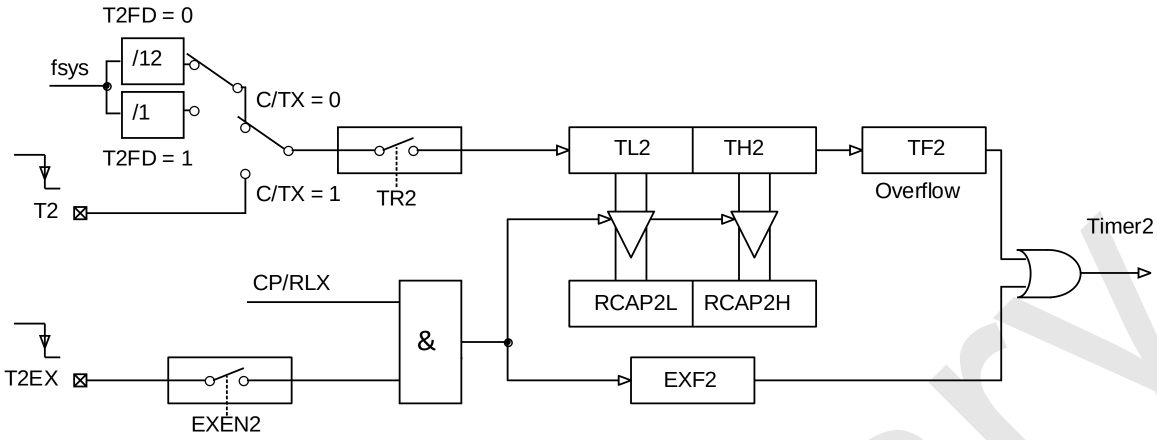
$C/\overline{T2}$	T2OE	DCEN	TR2	$CP/\overline{RL2}$	RCLK	TCLK	Mode	
X	0	X	1	1	0	0	0	16-bit capture
X	0	0	1	0	0	0	1	16-bit auto-reload timer
X	0	1	1	0	0	0		
X	0	X	1	X	1	X	2	Baud rate generator
					X	1		
0	1	X	1	X	0	0	3	Only for programmable clock
					1	X	3	Programmable clock output with baud rate generator
					X	1		
X	X	X	0	X	X	X	X	Timer 2 stops, T2EX channel is still allowed
1	1	X	1	X	X	X		Not recommended

Operating Mode 0: 16-bit Capture

In the capture mode, the EXEN2 bit of T2CON has two options.

If EXEN2 = 0, Timer 2 acts as a 16-bit timer or counter. If ET2 is enabled, Timer 2 can set TF2 overflow to generate an interrupt.

If EXEN2 = 1, Timer 2 performs the same operation, but the falling edge on external input T2EX can also cause the current values in TH2 and TL2 to be captured in RCAP2H and RCAP2L, respectively. In addition, the falling edge on T2EX also Can cause EXF2 in T2CON to be set. If ET2 is enabled, the EXF2 bit also generates an interrupt like TF2.



Mode 0: 16-bit capture

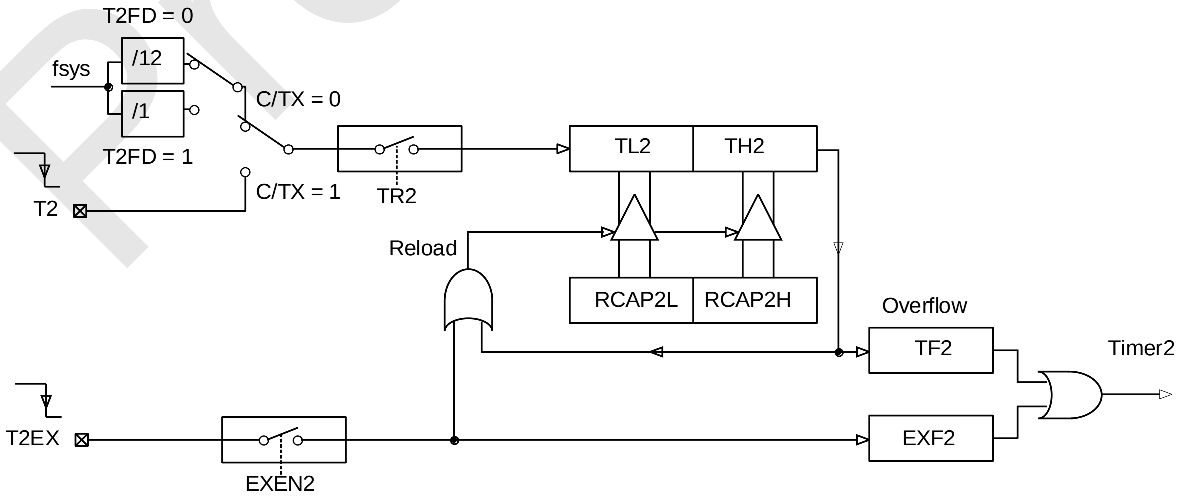
Operating Mode 1: 16-bit Auto-Reload Timer

In 16-bit auto-reload mode, Timer 2 can be selected to count up or count down. This function is selected by the DCEN bit (down counting allowed) in T2MOD. After the system is reset, the reset value of the DCEN bit is 0, and the timer 2 counts up by default. When DCEN is set to 1, Timer 2 counts up or down depending on the level on the T2EX pin.

When DCEN = 0, two options are selected through the EXEN2 bit in T2CON.

If EXEN2 = 0, Timer 2 increments to 0xFFFFH, sets the TF2 bit after overflow, and the timer automatically loads the 16-bit values of registers RCAP2H and RCAP2L written in user software into the TH2 and TL2 registers.

If EXEN2 = 1, either an overflow or a falling edge on the external input T2EX can trigger a 16-bit overload. When there is a falling edge on T2EX, the EXF2 position set. If ET2 is enabled, both TF2 and EXF2 bits can generate an interrupt.



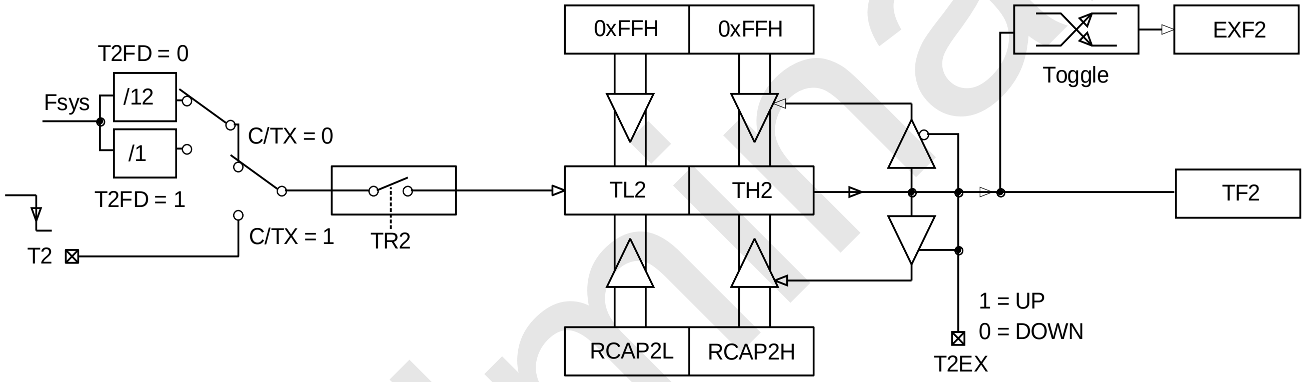
Mode 1: 16-bit auto-reload DCEN = 0

Setting the DCEN bit allows Timer 2 to count up or down. When DCEN = 1, the T2EX pin controls the direction of the count, and EXEN2 control is invalid.

Setting T2EX causes Timer 2 to count up. The timer overflows to 0xFFFFH, and then sets the TF2 bit. Overflow can also cause the 16-bit values on RCAP2H and RCAP2L to be reloaded into the timer register, respectively.

Setting T2EX to 0 causes Timer 2 to count down. When the values of TH2 and TL2 are equal to the values of RCAP2H and RCAP2L, the timer overflows. The TF2 bit is set and 0xFFFFH is reloaded into the timer register.

Regardless of whether Timer 2 overflows or not, the EXF2 bit is used as the 17th bit of the result. In this operating mode, EXF2 is not used as an interrupt flag.



Mode 1: 16-bit auto-reload DCEN = 1

Operating Mode 2: Baud Rate Generator

Set TCLK and/or RCLK in the T2CON register to select Timer 2 as the baud rate generator. The baud rate of the receiver and transmitter can be different. If Timer 2 acts as a receiver or transmitter, then Timer 1 acts as another baud rate generator.

Set TCLK and/or RCLK in the T2CON register to make Timer 2 enter the baud rate generator mode, which is similar to the automatic reload mode.

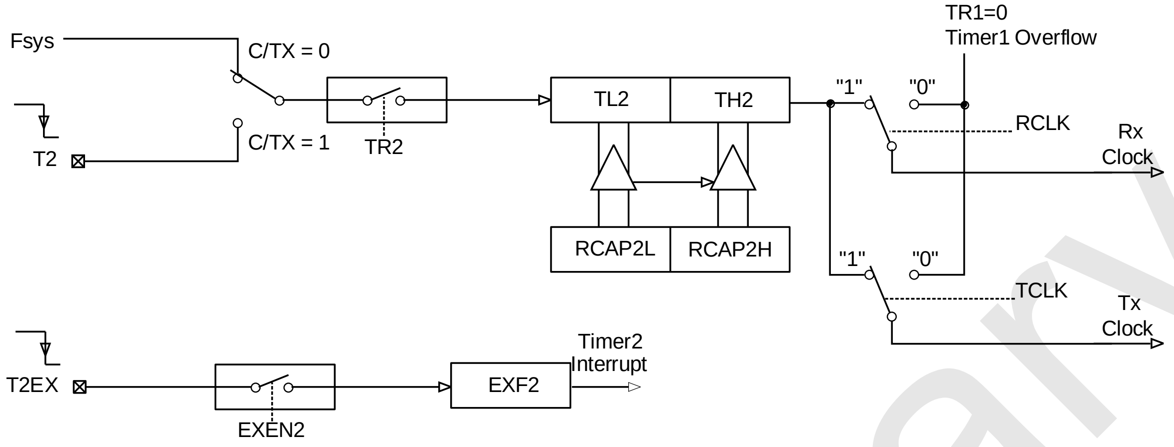
The overflow of Timer 2 will reload the values in the RCAP2H and RCAP2L registers to the Timer 2 count, but no interrupt will be generated.

If EXEN2 is set to 1, the falling edge on the T2EX pin will set up EXF2, but it will not cause a heavy load. So when Timer 2 is used as a baud rate transmitter, T2EX can be used as an additional external interrupt.

The baud rate in UART0 mode 1 and 3 is determined by the overflow rate of timer 2 according to the following equation:

$\text{BaudRate} = \frac{\text{fsys}}{[\text{RCAP2H,RCAP2L}]}$; (Note: [RCAP2H, RCAP2L] must be bigger than 0x0010**)**

The schematic diagram of Timer 2 as a baud rate generator is as follows:



Mode 2: Baud rate generator

Operating Mode 3: Programmable Clock Output

In this way, T2(P0.5) can be programmed to output a 50% duty cycle clock cycle: when $C/\overline{T2}$ = 0; T2OE = 1, timer 2 is enabled as a clock generator.

In this way, T2 outputs a clock with a 50% duty cycle.

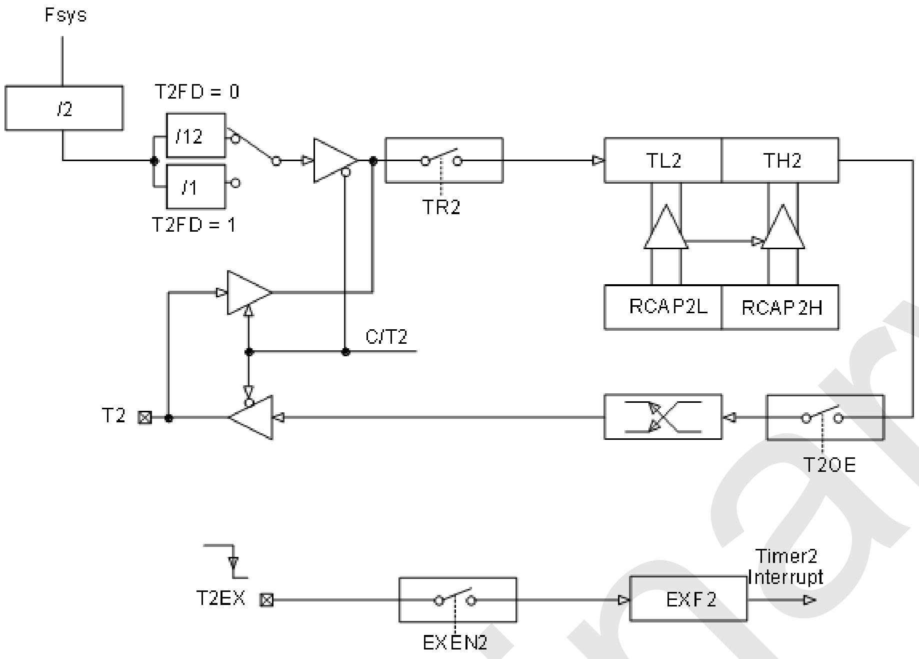
$$\text{Colck Out Frequency} = \frac{\text{fn2}}{(65536-[\text{RCAP2H,RCAP2L}])\times 4};$$

Among them, fn2 is the timer 2 clock frequency:

$$\text{fn2} = \frac{\text{fsys}}{12}; \quad \text{T2FD} = 0$$

$$\text{fn2} = \text{fsys}; \quad \text{T2FD} = 1$$

Timer 2 overflow does not generate an interrupt, and the T2 port is used as a clock output.



Mode 3: Programmable clock output

Note:

1. **Both TF2 and EXF2 can cause the interrupt request of Timer 2, both have the same vector address;**

2. **When the event occurs or at any other time, TF2 and EXF2 can be set to 1 by software, and only software and hardware reset can clear it to 0;**

3. **When EA = 1 and ET2 = 1, setting TF2 or EXF2 to 1 can cause Timer 2 to interrupt;**

4. **When Timer 2 is used as a baud rate generator, writing TH2/TL2 or RCAP2H/RCAP2L during UART0 communication will affect the accuracy of the baud rate and cause communication errors.**

12 Multiplier-Divider Unit (MDU)

The SC92F859X provides a 16-bit hardware multiplier and divider, which consists of extended accumulators EXA0~EXA3, extended B register EXB and operation control register OPERCON. It can replace software for 16-bit × 16-bit multiplication and 32-bit/16-bit division and increase program running efficiency.

Symbol	Address	Description	7	6	5	4	3	2	1	0	POR
EXA0	E9H	Extended Accumulator 0	EXA [7: 0]								00000000b
EXA1	EAH	Extended Accumulator 1	EXA [15: 8]								00000000b
EXA2	EBH	Extended Accumulator 2	EXA [23: 16]								00000000b
EXA3	ECH	Extended Accumulator 3	EXA [31: 24]								00000000b
EXBL	EDH	Extended B register L	EXB [7: 0]								00000000b
EXBH	EEH	Extended B register H	EXB [15: 8]								00000000b

OPERCON (EFH) Operation control register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	OPERS	MD	-	-	-	-	-	CRCSTA
R/W	R/W	R/W	-	-	-	-	-	R/W
POR	0	0	x	x	x	x	x	0

<table>
<tr><th>Bit number</th><th>Bit Mnemonic</th><th>Description</th></tr>
<tr><td>7</td><td><b>OPERS</b></td><td>Multiplier-divider operation start trigger control (Operater Start)
Write "1" to this bit to start a multiplication and division calculation, that is, this bit is just the trigger signal for the multiplication and division to start calculation. When the bit is zero, it means that the calculation has been completed. This bit can only be written to 1 valid.</td></tr>
<tr><td>6</td><td><b>MD</b></td><td>Multiplication and division
0: Multiplication operation. The multiplicand and multiplier are written and the product is read as follows:
<table>
<tr><th>Byte
Operand</th><th>Byte 3</th><th>Byte 2</th><th>Byte 1</th><th>Byte 0</th></tr>
<tr><td>multiplicand 16bit</td><td>-</td><td>-</td><td>EXA1</td><td>EXA0</td></tr>
<tr><td>multiplier 16bit</td><td>-</td><td>-</td><td>EXBH</td><td>EXBL</td></tr>
<tr><td>multiplier 32bit</td><td>EXA3</td><td>EXA2</td><td>EXA1</td><td>EXA0</td></tr>
</table>
1: Divide operation, write the dividend and divisor, read the quotient and remainder as follows:
<table>
<tr><th>Byte
Operand</th><th>Byte 3</th><th>Byte 2</th><th>Byte 1</th><th>Byte 0</th></tr>
<tr><td>dividend 32bit</td><td>EXA3</td><td>EXA2</td><td>EXA1</td><td>EXA0</td></tr>
<tr><td>divisor 16bit</td><td>-</td><td>-</td><td>EXBH</td><td>EXBL</td></tr>
<tr><td>quotient 32bit</td><td>EXA3</td><td>EXA2</td><td>EXA1</td><td>EXA0</td></tr>
<tr><td>remainder 16bit</td><td>-</td><td>-</td><td>EXBH</td><td>EXBL</td></tr>
</table>
</td></tr>
</table>

Note:

1. **It is forbidden to perform read or write operations on the EXA and EXB data registers during the calculation operation.**

2. **The time required for the operation conversion of the multiplier-divider is 16/fsys.**

13 PWM

The SC92F859X supports a maximum of 12-bit PWM output with 4 common cycles and a separate adjustable duty cycle. The functions of the PWM of the SC92F859X are as follows:

① 12-bit PWM accuracy;

② All PWM common one cycle;

③ The duty cycle of each PWM can be set separately;

④ The forward and reverse direction of each PWM output can be independently set;

⑤ 4-channel PWM can be switched to two sets of output ports：

a) PWM Group Ⅰ: PWM40/PWM42/PWM50/PWM52

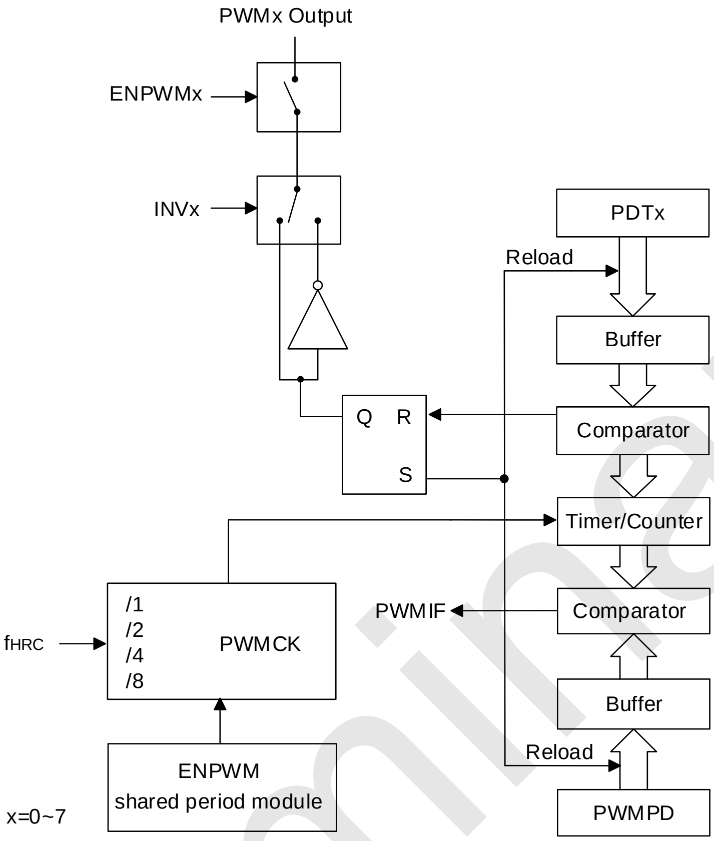
b) PWM Group Ⅱ: PWM41/PWM43/PWM51/PWM53

Note：

Two groups of PWM output ports are not allowed to open at the same time!

For example, if PWM50 is enabled, the output port of PWM group Ⅰ is enabled. In this case, the output port of PWM group Ⅱ is disabled. In other words, PWM41 / PWM43 / PWM51 / PWM53 is disabled.

13.1 PWM Structure Diagram



SC92F859X PWM Structure Diagram

13.2 PWM General Configuration Register

PWMCFG (D4H) PWM Set Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	ENPWM	PWMIF	PWMCK[1: 0]		PWMPD[11: 8]			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7	**ENPWM**	PWM module switch control (Enable PWM) 1: Allow Clock to enter the PWM unit, the PWM is in the operating state, and the state of the PWM output port is controlled by the register ENPxy (x=4~5, y=0~3) 0: The PWM unit stops operating, the PWM counter is cleared, and all PWM output ports are set to the GPIO state
6	**PWMIF**	PWM interrupt request flag When the PWM counter overflows (that is, when the count exceeds PWMPD), this bit is automatically set to 1 by the hardware. If IE1[1] (EPWM) is also set to 1, the PWM interrupt is generated at this time. After the PWM interrupt occurs, the hardware will not automatically clear this bit. This bit must be cleared by the user's software.
5~4	**PWMCK[1: 0]**	PWM Clock Source Selector (PWM Clock Source Selector) 00: f_{HRC} 01: f_{HRC} /2 10: f_{HRC} /4 11: f_{HRC} /8

3~0	**PWMPD[11: 8]**	The period of the PWM is set to the upper 4 bits; This value represents the (period – 1) of the PWM output waveform; that is, the period value of the PWM output is(PWMPD[11: 0] + 1)* PWM clock；

PWMCON (D3H) PWM Control Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	**PWMPD[7: 0]**							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7~0	**PWMPD[7: 0]**	The period shared by PWM sets the low 8 bits; This value represents the (period – 1) of the PWM output waveform; that is, the period value of the PWM output is(PWMPD[11: 0] + 1)* PWM clock;

IE1 (A9H) Interrupt Enable Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	-	ECMP	ETK	EINT2	EBTM	EPWM	ESSI0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
1	**EPWM**	PWM interrupt enable control 0: Disable PWM interrupt 1: Enable interrupt when PWM counter overflows

IP1 (B9H) Interrupt Priority Register 1 (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	IPCMP	-	IPINT2	IPBTM	IPPWM	IPSSI0
R/W	-	-	R/W	-	R/W	R/W	R/W	R/W
POR	x	x	0	x	0	0	0	0

Bit number	Bit Mnemonic	Description
1	**IPPWM**	PWM interrupt priority selection 0: Set the PWM interrupt priority to "low" 1: Set the PWM interrupt priority to "High"

PWM Duty Cycle Adjustment Register PDTxy (Read/Write)

740H	ENP40	INV40	-	-	PDT40[11:8]
741H	PDT40[7:0]				
742H	ENP41	INV41	-	-	PDT41[11:8]
743H	PDT41[7:0]				
744H	ENP42	INV42	-	-	PDT42[11:8]
745H	PDT42[7:0]				
746H	ENP43	INV43	-	-	PDT43[11:8]
747H	PDT43[7:0]				
748H	ENP50	INV50	-	-	PDT50[11:8]
749H	PDT50[7:0]				
74AH	ENP51	INV51	-	-	PDT51[11:8]
74BH	PDT51[7:0]				
74CH	ENP52	INV52	-	-	PDT52[11:8]
74DH	PDT52[7:0]				
74EH	ENP53	INV53	-	-	PDT53[11:8]
74FH	PDT53[7:0]				

Bit number	Bit Mnemonic	Description
7	**ENPxy** **(x=4~5, y=0~3)**	Pxy port PWM waveform output selection 0: PWM output of Pxy port is turned off and used as GPIO port 1 1: When ENPWM=1, Pxy is used as PWM waveform output port Note： 1. The 8-channel PWM channel is divided into two groups.PWM Group Ⅰ: PWM40/PWM42/PWM50/PWM52; PWM Group Ⅱ: PWM41/PWM43/PWM51/PWM53. 2. Two groups of PWM waveform output selector switches cannot be turned on at the same time! For example, if ENP50 is enabled, the

		output port of PWM group Ⅰ is enabled. In this case, the waveform output selection switch of PWM group Ⅱ is disabled, in other word, PWM41 / PWM43 / PWM51 / PWM53 is disabled.
6	**INVxy** **(x=4~5, y=0~3)**	Pxy port PWM waveform output reverse control 1: PWM waveform output of Pxy port is reversed 0: PWM waveform output of Pxy port is not reversed
3~0	**PDTxy [11: 8]** **(x=4~5, y=0~3)**	Pxy port PWM waveform duty cycle length setting The high-level width of the PWM waveform on the Pxy pin is (PDTxy [11: 0]) PWM clocks

Bit number	Bit Mnemonic	Description
7~0	**PDTxy [7: 0]** **(x=4~5, y=0~3)**	Pxy port PWM waveform duty cycle length setting; The high-level width of the PWM waveform on the Pxy pin is (PDTxy [11: 0]) PWM clocks

Note:

1. If ENPWM is set to 1, the PWM module is turned on, but ENPxy=0, the PWM output is turned off and used as a GPIO port. At this time, the PWM module can be used as a 12-bit Timer. At this time, EPWM (IE1.1) is set to 1, and the PWM will still generate an interrupt.

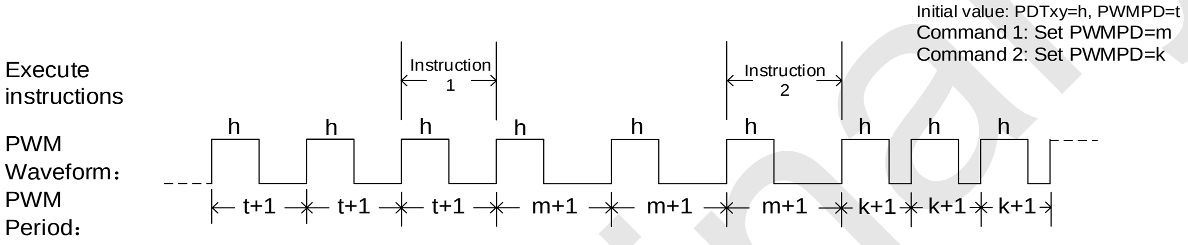
13.3 PWM Waveforms and Directions

The effect of changing SFR parameters on the PWM waveform is as follows:

① Duty cycle change characteristics

When the PWMn outputs a waveform, if the duty cycle needs to be changed, it can be achieved by changing the value of the high-level setting register (PDTxy). But need to pay attention: change the value of PDTxy, the duty ratio will not change immediately, but wait for the end of this cycle and change it in the next cycle.

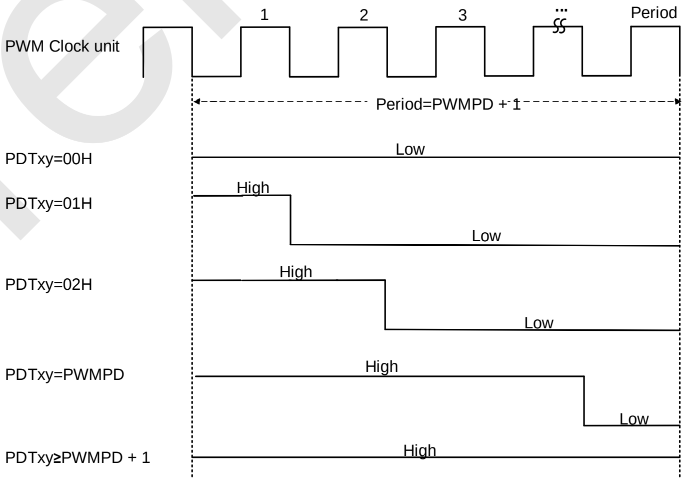
② Periodic change characteristics



Periodic change characteristic diagram

When the PWM outputs a waveform, if the period needs to be changed, it can be achieved by changing the value of the period setting register PWMPD. Change the value of PWMPD, the cycle will not change immediately, but wait for the end of the cycle, and change in the next cycle, refer to the figure above.

③ Relationship between period and duty cycle



Relationship between cycle and duty cycle

The relationship between period and duty cycle is shown in the figure above. The premise of this result is that the PWM output inverse control (INVxy) is initially 0. If you want to get the opposite result, you can set INVxy to 1.

14 General-purpose I/O (GPIO)

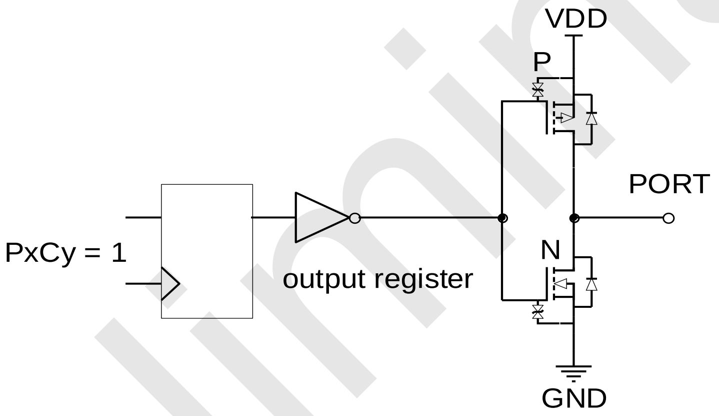
The SC92F859X provides up to 46 bidirectional GPIO ports that can be controlled. The input and output control registers are used to control the input and output status of each port. When the port is used as an input, each I/O port has an internal pull-up resistor controlled by PxPHy. The 46 IOs are multiplexed with other functions. Among them, P3 can be set to output 1/4VDD or 1/3VDD voltage, which can be used as a COM driver for LCD display. When the I/O port is in the input or output state, the actual state value of the port is read from the port data register.

Note: The unused and unleaded IO ports should be set to strong push-pull output mode.

14.1 GPIO Structure Diagram

Strong Push-pull Output Mode

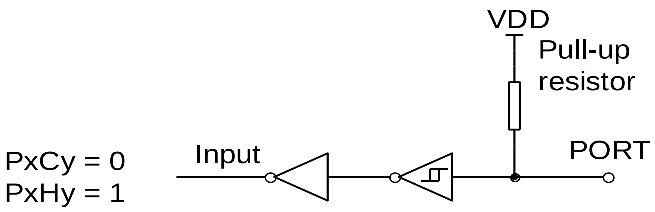
In the strong push-pull output mode, it can provide continuous high-current drive: an output greater than 18mA is high, and an output greater than 65mA is low. The schematic diagram of the port structure of the strong push-pull output mode is as follows:



Strong push-pull output mode

Pull-up Input Mode

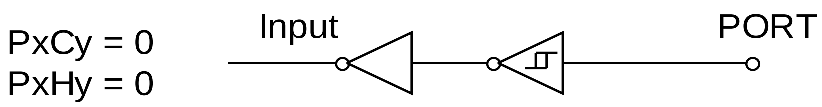
In the pull-up input mode, a pull-up resistor is constantly connected to the input port. Only when the input port is pulled low, the low-level signal is detected. The schematic diagram of the port structure with pull-up input mode is as follows:



Input mode with pull-up resistor

High Impedance Input Mode (Input only)

The schematic diagram of the port structure of the high impedance input mode is as follows:



High impedance input mode

14.2 I/O Port-related Registers

P0CON (9AH) P0 Port Input/Output Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	P0C7	P0C6	P0C5	P0C4	P0C3	P0C2	P0C1	P0C0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P0PH (9BH) P0 Port pull-up Resistor Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	P0H7	P0H6	P0H5	P0H4	P0H3	P0H2	P0H1	P0H0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P1CON (91H) P1 Port Input/Output Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bit Mnemonic	P1C7	P1C6	P1C5	P1C4	P1C3	P1C2	P1C1	P1C0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P1PH (92H) P1 Port Pull-up Resistor Control Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	P1H7	P1H6	P1H5	P1H4	P1H3	P1H2	P1H1	P1H0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P2CON (A1H) P2 Port Input/output Control Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	P2C7	P2C6	P2C5	P2C4	P2C3	P2C2	P2C1	P2C0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P2PH (A2H) P2 Port Pull-up Resistor Control Register (read/write)

Bit number	7	6	5	4	3	2	1	0

Bit Mnemonic	P2H7	P2H6	P2H5	P2H4	P2H3	P2H2	P2H1	P2H0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P3CON (B1H) P3 Port Input/output Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	P3C7	P3C6	P3C5	P3C4	P3C3	P3C2	P3C1	P3C0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P3PH (B2H) P3 Port Pull-up Resistor Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	P3H7	P3H6	P3H5	P3H4	P3H3	P3H2	P3H1	P3H0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P4CON (C1H) P4 Port Input/output Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bit Mnemonic	P4C7	P4C6	P4C5	P4C4	P4C3	P4C2	P4C1	P4C0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P4PH (C2H) P4 Port Pull-up Resistor Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	P4H7	P4H6	P4H5	P4H4	P4H3	P4H2	P4H1	P4H0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

P5CON (D9H) P5 Port Input/output Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	P5C5	P5C4	P5C3	P5C2	P5C1	P5C0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	0	0	0	0	0	0

P5PH (DAH) P5 Port Pull-up Resistor Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bit Mnemonic	-	-	P5H5	P5H4	P5H3	P5H2	P5H1	P5H0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7~0	**PxCy** **(x=0~4, y=0~7** **x=5, y=0~5)**	Px port input and output control: 0: Pxy is the input mode (initial value at power-on) 1: Pxy is a strong push-pull output mode
7~0	**PxHy** **(x=0~4, y=0~7** **x=5, y=0~5)**	The Px port pull-up resistor setting is only valid when PxCy=0: 0: Pxy is the high-impedance input mode (initial value at power-up), and the pull-up resistor is turned off; 1: Pxy pull-up resistor is on

P0 (80H) P0 Port Data Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	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
POR	0	0	0	0	0	0	0	0

P1 (90H) P1 Port Data Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	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
POR	0	0	0	0	0	0	0	0

P2 (A0H) P2 Port Data Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	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
POR	0	0	0	0	0	0	0	0

P3 (B0H) P3 Port Data Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	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
POR	0	0	0	0	0	0	0	0

P4 (C0H) P4 Port Data Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	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
POR	0	0	0	0	0	0	0	0

P5 (D8H) P5 Port Data Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	-	P5.5	P5.4	P5.3	P5.2	P5.1	P5.0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7~0	**P0.x** **(x=0~7)**	P0 port latch register data
7~0	**P1.x** **(x=0~7)**	P1 port latch register data
7~0	**P2.x** **(x=0~7)**	P2 port latch register data
7~0	**P3.x**	P3 port latch register data

	(x=0~7)	
7~0	**P4.x** **(x=0~7)**	P4 port latch register data
5~0	**P5.x** **(x=0~5)**	P5 port latch register data

IOHCON0 (96H) IOH Setting Register 0 (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	P1H[1: 0]		P1L[1: 0]		P0H[1: 0]		P0L[1: 0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description

7~6	**P1H[1: 0]**	P1 high four IOH settings
		00: Set P1 high four IOH level 0 (maximum);
		01: Set P1 high four IOH level 1;
		10: Set P1 high four IOH level 2;
		11: Set P1 high four IOH level 3 (minimum);

5~4	**P1L[1: 0]**	P1 low four IOH settings
		00: Set P1 low four IOH level 0 (maximum);
		01: Set P1 low four IOH level 1;
		10: Set P1 low four IOH level 2;

		11: Set P1 low four IOH level 3 (minimum);

3~2	**P0H[1: 0]**	P0 high four IOH settings
		00: Set P0 high four IOH level 0 (maximum);
		01: Set P0 high four IOH level 1;
		10: Set P0 high four IOH level 2;
		11: Set P0 high four IOH level 3 (minimum)

1~0	**P0L[1: 0]**	P0 low four IOH settings
		00: Set P0 low four IOH level 0 (maximum);
		01: Set P0 low four IOH level 1;
		10: Set P0 low four IOH level 2;
		11: Set P0 low four IOH level 3 (minimum);

IOHCON1 (97H) IOH Setting Register 1 (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	P3L[1: 0]		P2H[1: 0]		P2L[1: 0]	
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	x	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description

5~4	**P3L[1: 0]**	P3 low four IOH settings
		00: set P3 low four IOH level 0 (maximum);
		01: Set P3 low four IOH level 1;

		10: Set P3 low four IOH level 2;
		11: Set P3 low four IOH level 3 (minimum);

3~2	**P2H[1: 0]**	P2 high four IOH settings
		00: Set P2 high four IOH level 0 (maximum);
		01: Set P2 high four IOH level 1;
		10: Set P2 high four IOH level 2;
		11: Set P2 high four IOH level 3 (minimum);

1~0	**P2L[1: 0]**	P2 low four IOH settings
		00: Set P2 low four IOH level 0 (maximum);
		01: Set P2 low four IOH level 1;
		10: Set P2 low four IOH level 2;
		11: Set P2 low four IOH level 3 (minimum);

7~6	**-**	Reserved

15 LCD/LED Display Driver

The SC92F8597/8596/8593 integrates hardware LCD/LED display drive circuit inside, which can facilitate users to realize LCD and LED display drive. Its main features are as follows:

1. Choose one of LCD and LED display driver;
2. LCD and LED display drivers share related IO ports and registers.

The LCD display driver **functions are as follows:**

1. 4 display drive modes are available: 8 X 24, 6 X 26, 5 X 27, or 4X 28 segments;
2. 2 kinds of offset methods are available: 1/4 Bias and 1/3 Bias;
3. 4 levels of com port drive capability are optional;
4. The display drive circuit can choose the built-in 32 kHz LRC or external 32k oscillator as the clock source, the frame frequency is about 64Hz.

The LED **display driver functions are as follows:**

1. 4 display drive modes are available: 8 X 24, 6 X 26, 5 X 27, or 4X 28 segments;
2. Seg port drive capability is optional in 4 levels;
3. The display drive circuit can select the built-in 32 kHz LRC or external 32k oscillator as the clock source, and the frame frequency is about 64Hz.

Note: SC92F8595 does not support LCD/LED functionality!

15.1 LCD/LED Display Drive-related Registers

DDRCON (93H) Display Drive Control Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	DDRON	DMOD	DUTY[1: 0]		VLCD[3: 0]			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description

7	**DDRON**	**LCD/LED display drive enable control** 0: The display driver scan is turned off 1: The display driver scan is turned on
6	**DMOD**	**LCD/LED display drive mode** 0: LCD mode; 1: LED mode

5~4	**DUTY[1: 0]**	**LCD/LED display duty control**
		00: 1/8 duty cycle, S4~S27 are segments, C0~C7 are common;
		01: 1/6 duty cycle, S2~S27 are segments, C2~C7 are common;
		10: 1/5 duty cycle, S1~S27 are segments, C3~C7 are common;
		11: 1/4 duty cycle, S0~S27 is segment, C4~C7 is common, or S4~S27 is segment, C0~C3 is common

3~0	**VLCD[3: 0]**	**LCD voltage regulation** VLCD=V_{DD}*(17+VLCD[3: 0])/32

P0VO (9CH) P0 Port Display Driver Output Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	P07VO	P06VO	P05VO	P04VO	P03VO	P02VO	P01VO	P00VO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description

7~0	**P0nVO**	**Open P0n port display driver output** 0: Disable the display driver output function of P0n port 1: Enable the display driver output function of P0n port

P1VO (94H) P1 Port Display Driver Output Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	P17VO	P16VO	P15VO	P14VO	P13VO	P12VO	P11VO	P10VO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7~0	**P1nVO**	**Open P1n port display driver output** 0: Disable the display driver output function of P1n port 1: Enable the display driver output function of P1n port

P2VO (A3H) P2 Port Display Driver Output Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	P27VO	P26VO	P25VO	P24VO	P23VO	P22VO	P21VO	P20VO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7~0	**P2nVO**	**Open P2n port display driver output** 0: Disable the display driver output function of P2n port 1: Enable the display driver output function of P2n port

P3VO (B3H) P3 Port Display Driver Output Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	P37VO	P36VO	P35VO	P34VO	P33VO	P32VO	P31VO	P30VO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7~0	**P3nVO**	**Open P3n port display driver output** 0: Disable the display driver output function of P3n port 1: Enable the display driver output function of the P3n port

OTCON (8FH) Output Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	SSMOD[1:0]		-	-	VOIRS[1: 0]		SCS	BIAS

R/W	R/W	R/W	-	-	R/W	R/W	R/W	R/W
POR	0	0	x	x	0	0	0	0

Bit number	Bit Mnemonic	Description
3~2	**VOIRS[1: 0]**	**LCD voltage output port voltage divider resistance selection** 00: Set the total resistance of the internal voltage divider to be 100kΩ 01: Set the total resistance of the internal voltage divider resistor to 200kΩ 10: Set the total resistance of the internal voltage divider to 400kΩ 11: Set the total resistance of the internal voltage divider to 800kΩ Each time Common is switched, the first 1/16 time is fixed to select a 100k resistor, and the last 15/16 time is switched to the resistance value selected by VORIS
1	**SCS**	**LCD/LED Segment/Common multiplex pin selection** 0: When set to 1/4 duty cycle, S0~S27 are segments and C4~C7 are common 1: When set to 1/4 duty cycle, S4~S27 are segments and C0~C3 are common
0	**BIAS**	**LCD display drive bias voltage setting:** 0: 1/4 bias voltage; 1: 1/3 bias voltage

15.2 LCD/LED Display RAM Configuration

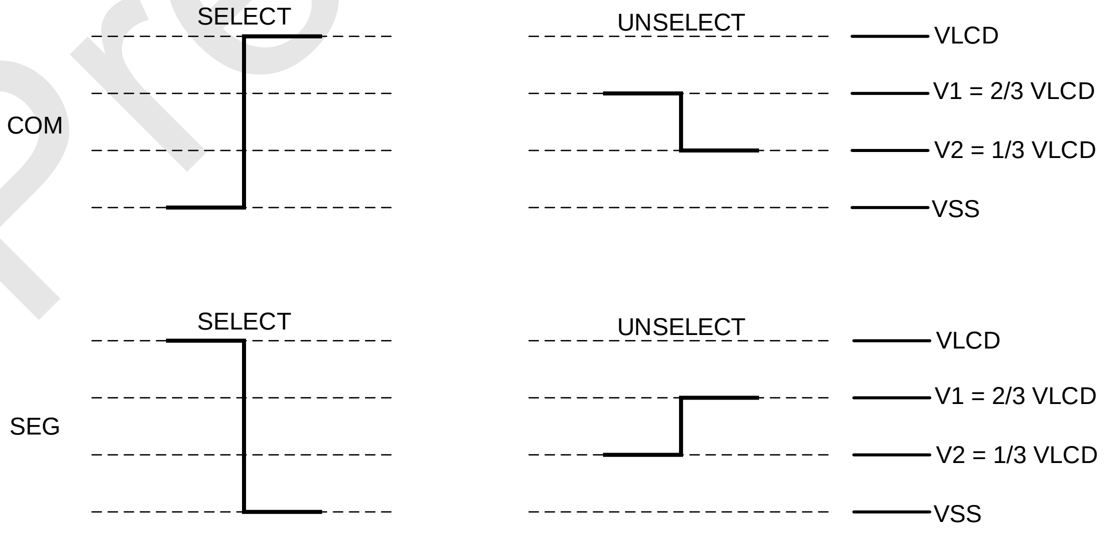
Address	7	6	5	4	3	2	1	0
	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0

700H	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0	SEG0
701H	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1	SEG1
702H	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2	SEG2
703H	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3	SEG3
704H	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4	SEG4
705H	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5	SEG5
706H	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6	SEG6
707H	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7	SEG7
708H	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8	SEG8
709H	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9	SEG9
70AH	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10
70BH	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11
70CH	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12
70DH	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13
70EH	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14
70FH	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15
710H	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16
711H	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17

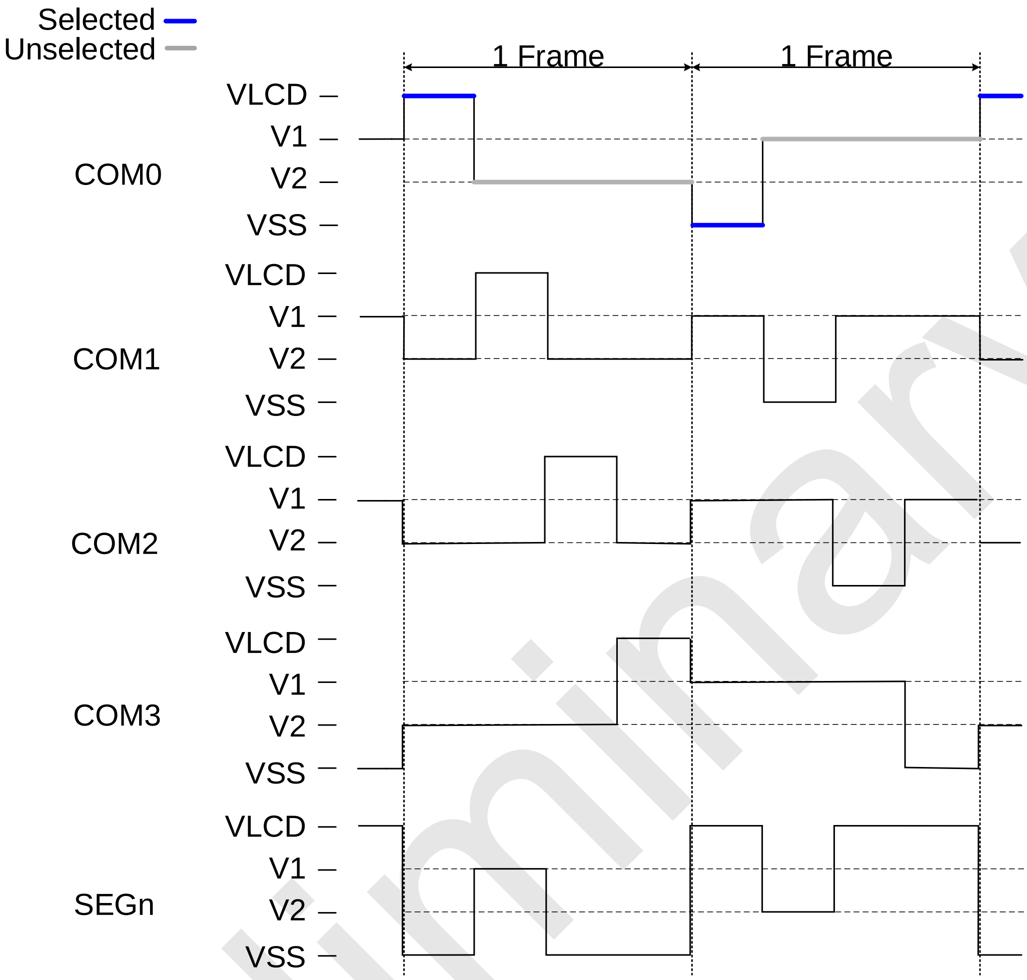
712H	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18
713H	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19
714H	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20
715H	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21
716H	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22
717H	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23
718H	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24
719H	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25
71AH	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26
71BH	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27

15.3 LCD Waveform

15.3.1 1/3Bias LCD Waveform

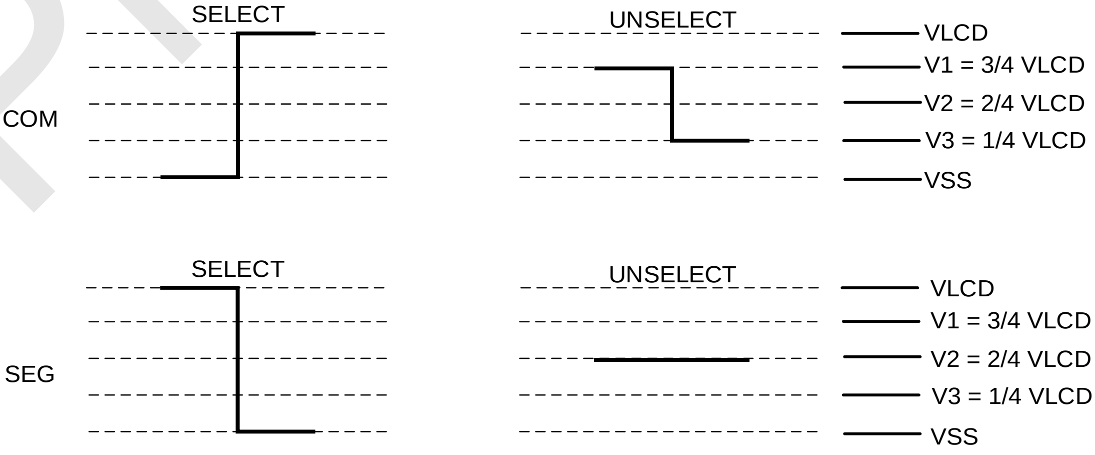


1/3 Bias LCD Gated and non-gated voltages

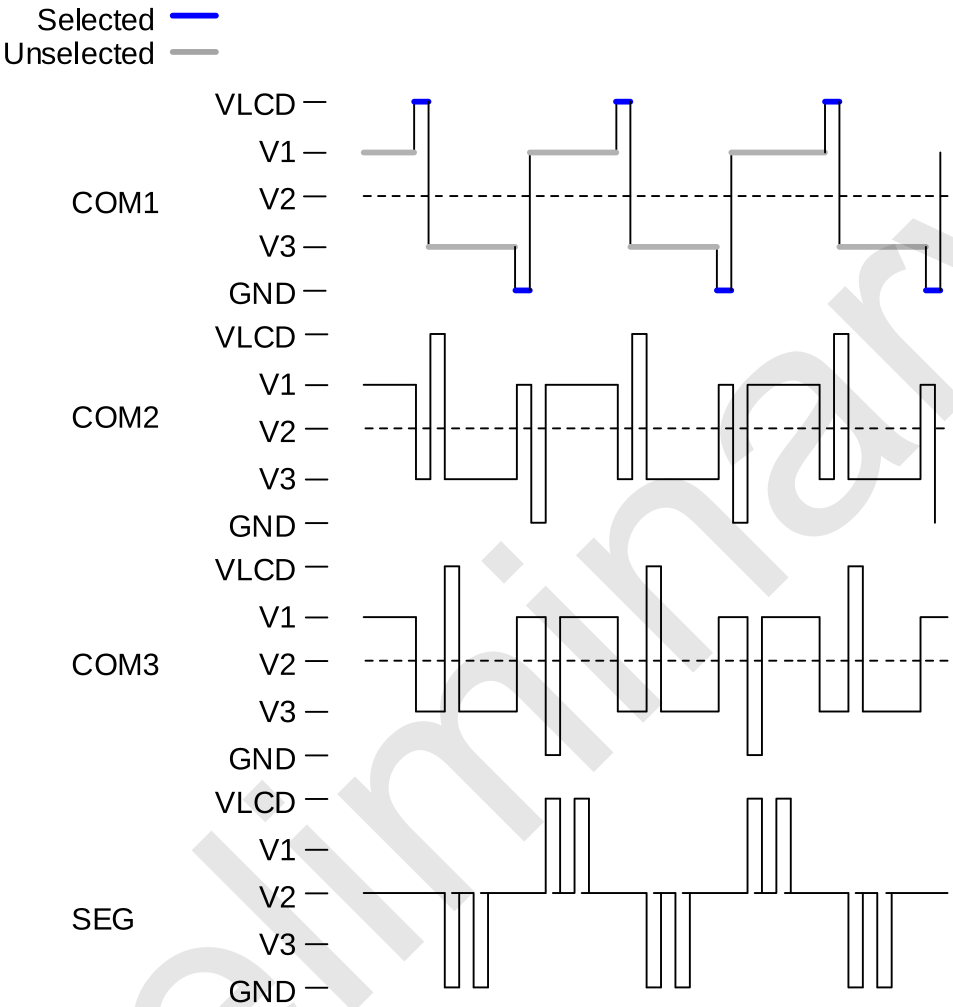


1/3 Bias Waveforms of COM and SEG in LCD applications

15.3.2 1/4Bias LCD Waveform

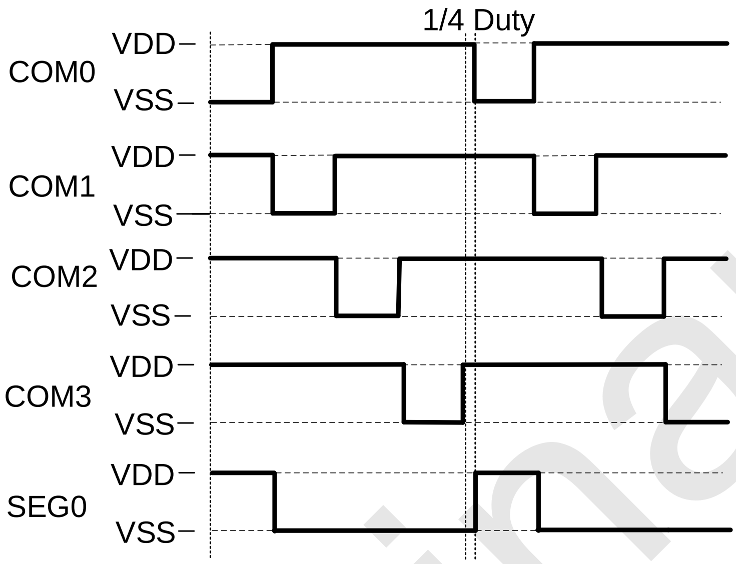


1/4 Bias LCD Gated and non-gated voltages



1/4 Bias Waveforms of COM and SEG in LCD applications

15.4 LED Waveform



Waveforms of COM and SEG in LED applications

15.5 LCD/LED Example

15.5.1 LCD Configuration Demo Program

```
unsigned char xdata LCDRAM[30] _at_ 0x1000;

unsigned char lcd_addr;

unsigned char lcd_data;


DDRCON |= 0x00;         //0: LCD mode 1: LED mode

DDRCON |= 0x30;         //1/4 Duty cycle

DDRCON |= 0x07;         // VLCD=VDD*3/4

DDRCON |= 0x80;         // Display driver scan on


P0VO = 0xFF;            // Open the display driver output function of P0 port

P1VO = 0xFF;            // Open the display driver output function of P1 port

P2VO = 0xFF;            // Open the display driver output function of P2 port
```

```
P3VO = 0xFF;              // Open the display driver output function of P3 port
OTCON = 0x06; // Set the total resistance of the internal voltage divider resistor to 200KΩ
                          //1/4 Bias voltage；S4~S27 is segment,C0~C3 is common
LCDRAM[lcd_addr] = lcd_data;   /Write the value to be displayed to the LCD RAM
```

15.5.2 LED Configuration Demo Program

```
unsigned char xdata LEDRAM[30] _at_ 0x1000;
unsigned char led_addr;
unsigned char led_data;

DDRCON |= 0x4F;           //0: LCD mode1 : LED mode
                          //LED mode; 1/8 Duty cycle
                          //S4~S27 is segment,C0~C7 is common；
DDRCON |= 0x80;           // Display driver scan on
IOHCON0 = 0xC0;           // Set P1 high four IOH level 3 (minimum), other pins IOH level 0
                          (maximum)
IOHCON1 = 0x00;
P0VO = 0xFF;              // Open the display driver output function of P0 port
P1VO = 0xFF;              // Open the display driver output function of P1 port
P2VO = 0xFF;              // Open the display driver output function of P2 port
P3VO = 0xFF;              // Open the display driver output function of P3 port
OTCON = 0x00;
LCDRAM[led_addr] = led_data;   //Write the value to be displayed to the LED RAM
```

16 SERIAL INTERFACE (UART0)

The SC92F859X supports a full-duplex serial port, which can be conveniently used for connection with other devices or equipment, such as Wifi module circuit or other UART communication interface driver chip. The functions and features of UART0 are as follows:

1. Three communication modes are available: Mode 0, Mode 1 and Mode 3;
2. Can choose Timer 1 or Timer 2 as the baud rate generator;
3. Interrupt RI/TI can be generated after transmission and reception are completed, and the interrupt flag needs to be cleared by software.

16.1 UART0 Related Register

SCON (98H) Serial Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	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
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7~6	**SM0~1**	Serial communication mode control bit 00: Mode 0, 8-bit half-duplex synchronous communication mode, serial data is sent and received on the RX pin. The TX pin is used as the transmit shift clock. 8 bits are sent and received per frame, the low bit is received or sent first; 01: Mode 1, 10-bit full-duplex asynchronous communication, consisting of 1 start bit, 8 data bits and 1 stop bit, the communication baud rate is variable; 10: reserved; 11: Mode 3, 11-bit full-duplex asynchronous communication, consisting of 1 start bit, 8 data bits, a programmable 9th bit, and 1 stop bit. The communication baud rate is variable.

5	**SM2**	Serial communication mode control bit 2, this control bit is only valid for mode 3 0: set RI to generate an interrupt request every time a complete data frame is received; 1: When a complete data frame is received, RI will be set to generate an interrupt request only when RB8=1.
4	**REN**	Receive enable control bit 0: data reception is not allowed; 1: Allow receiving data.
3	**TB8**	Only valid for mode 3, which is the 9th bit of the transmitted data
2	**RB8**	Only valid for mode 3, the 9th bit of the received data
1	**TI**	Transmit interrupt flag
0	**RI**	Receive interrupt flag

SBUF (99H) Serial Data Buffer Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	SBUF[7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7~0	**SBUF[7: 0]**	**Serial data buffer register**

		SBUF contains two registers: a transmit shift register and a receive latch. The data written to SBUF will be sent to the transmit shift register and start the transmission process. Reading SBUF will return the contents of the receive latch.

PCON (87h) Power Management Control Register (write only, *not readable*)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	SMOD	-	-	-	RST	-	STOP	IDL
R/W	write only	-	-	-	write only	-	write only	write only
POR	0	x	x	x	n	x	0	0

Bit number	Bit Mnemonic	Description
7	**SMOD**	When SM0~1 = 01 (UART0 mode 1) or SM0~1 = 11 (UART0 mode 3), the baud rate is set to bit: 0: the serial port runs at 1 division of the system clock 1: the serial port runs at 16 division of the system clock When SM0~1 = 00 (UART0 mode 0) Baud rate setting bit: 0: the serial port runs at 12 division of the system clock 1: the serial port runs at 4 division of the system clock

16.2 Baud Rate of Serial Communication

In mode 0, the baud rate can be programmed as 1/12 or 1/4 of the system clock, determined by SMOD(PCON.7) bits. When SMOD is 0, the serial port operating at the 4-divisor of system clock. When SMOD is 1, the serial port operating at the 4-divisor of the system clock.

In modes 1 and 3, the serial port clock source can be programmed as either 1 or 16 sub-frequencies of the system clock, determined by SMOD(PCON.7) bits. When SMOD is 0, the serial port runs at 1 division of the system clock. When SMOD is 1, the serial port operating at 16 dividers of the system clock. After the serial port clock source is determined, set the baud rate overflow rate by timer 1 or Timer 2:

When TCLK (T2CON. 4) and RCLK (T2CON. 5) bits are both 0, then timer 1 is in baud rate generator mode, and the baud rate overflow rate of UART0 is set by [TH1, TL1]. The formula is as follows, note: When timer 1 acts as a baud rate generator, timer 1 must stop counting, i.e. TR1=0:

■ SMOD = 0： $\mathrm{BaudRate} = \frac{\mathrm{Fsys}}{[\mathrm{TH1,TL1}]}$; (Note: [TH1, TL1] must be bigger than 0x0010**)**

■ SMOD = 1： $\mathrm{BaudRate} = \frac{1}{16} * \frac{\mathrm{Fsys}}{[\mathrm{TH1,TL1}]}$;

When either TCLK(T2CON. 4) or RCLK(T2CON. 5) is 1, then timer 2 is in baud rate generator mode, and the baud rate overflow rate of UART0 is set by [RCAP2H, RCAP2L], the formula is as follows:

■ SMOD = 0： $\mathrm{BaudRate} = \frac{\mathrm{Fsys}}{[\mathrm{RCAP2H,RCAP2L}]}$; (Note： [RCAP2H,RCAP2L] must be bigger than 0x0010**)**

■ SMOD = 1： $\mathrm{BaudRate} = \frac{1}{16} * \frac{\mathrm{Fsys}}{[\mathrm{RCAP2H,RCAP2L}]}$;

17 SPI/TWI/UART Serial Interface (SSI)

The SC92F859X internally integrates three-select one universal serial circuits interface (referred to as SSI), which can facilitate the connection between MCU and devices or equipment with different interfaces. The user can configure the SSI interface to any one of SPI, TWI and UART through the USMD1[1: 0], USMD0[1: 0] bits of the configuration register OTCON, or the USMD2[1: 0] bits of TMCON. Its characteristics are as follows:

The SC92F859X internally integrates three-select one serial interface circuit (referred to as SSI), which facilitates the connection between MCU and devices or devices with different interfaces. The user can configure the SSI interface to any of the SPI, TWI and UART communication modes by configuring the SSMOD[1:0] bit of register OTCON. Its characteristics are as follows:

1. SPI mode can be configured as one of master mode or slave mode
2. TWI mode communication only can be configured as slave mode
3. There are two UART modes:

Mode 1: 10-bit full-duplex asynchronous communication

Mode 3: 11-bit full-duplex asynchronous communication

The specific configuration method is as follows:

OTCON (8FH) Output Control Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	SSMOD[1:0]		-	-	VOIRS[1: 0]		SCS	BIAS
R/W	R/W	R/W	-	-	R/W	R/W	R/W	R/W
POR	0	0	x	x	0	0	0	0

Bit number	Bit Mnemonic	Description

7~6	**SSMOD[1:0]**	**SSI Communication mode control bit**
		00: SSI close
		01: SSI Set to SPI communication mode;
		10: SSI Set to TWI communication mode;
		11: SSI Set to UART communication mode;

17.1 SPI

SSMOD[1:0] = 01, the SSI is configured as SPI interface. Serial Peripheral Device Interface (SPI) is a high-speed serial communication interface that allows the MCU to communicate with peripheral devices (including other MCUS) in full duplex, synchronous serial communication.

17.1.1 SPI Operation Related Registers

SSCON0 (9DH) SPI control register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	SPEN	-	MSTR	CPOL	CPHA	SPR2	SPR1	SPR0
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	x	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7	**SPEN**	**SPI Enable control** 0: close SPI 1: open SPI
5	**MSTR**	**SPI master-slave selection** 0: SPI is slave device 1: SPI is master device
4	**CPOL**	**Clock polarity control bit** 0: SCK is low in idle state 1: SCK is high in idle state
3	**CPHA**	**Clock phase control bit**

		0: Collect data on the first edge of the SCK cycle 1: Collect data on the second edge of the SCK cycle
2~0	**SPR[2: 0]**	**SPI Clock rate selection bit** 000：f_{SYS} /4 001：f_{SYS} /8 010：f_{SYS} /16 011：f_{SYS} /32 100：f_{SYS} /64 101：f_{SYS} /128 110：f_{SYS} /256 111：f_{SYS} /512
6	**-**	Reserved

SSCON1 (9EH) SPI Status Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	SPIF	WCOL	-	-	TXE	DORD	-	TBIE
R/W	R/W	R/W	-	-	R/W	R/W	-	R/W
POR	0	0	x	x	0	0	x	0

Bit number	Bit Mnemonic	Description
7	**SPIF**	**SPI data transmission flag** 0: Cleared by software 1: Indicates that data transmission has been completed, set by hardware
6	**WCOL**	**Write conflict flag** 0: Cleared by software, indicating that the write conflict has been processed 1: Set by hardware to indicate that a conflict is detected Transmission direction selection bit
2	**DORD**	**Transmission direction selection bit** 0: MSB first sent 1: LSB first sent
0	**TBIE**	**Send cache interrupt to allow control bit** 0：When TXE=1, intecrrupts are not allowed 1：When TXE=1, the SPI interrupt is generated
5~4,1	**-**	Reserved

SSDAT (9FH) SPI Data register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	SSDAT [7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7~0	**SSDAT [7: 0]**	**SPI data buffer register** Data written to SSDAT is placed in the send shift register SSDAT gets data to receive the shift register

17.1.2 Signal Description

Master-Out/Slave-In (MOSI):

This signal connects the master device and a slave device. Data is serially transmitted from the master device to the slave device through MOSI, the master device outputs, and the slave device inputs.

Master-In and Slave-Out (MISO):

This signal connects the slave device and the master device. Data is serially transmitted from the slave device to the master device through MISO, the slave device is output, and the master device is input. When the SPI is configured as a slave device and not selected, the MISO pin of the slave device is in a high impedance state.

SPI Serial Clock (SCK):

The SCK signal is used to control the synchronous movement of input and output data on the MOSI and MISO lines. A byte is transmitted on the wire every 8 clock cycles. If the slave is not selected, the SCK signal is ignored by the slave.

17.1.3 Operating Modes

SPI can be configured as one of master mode or slave mode. The configuration and initialization of the SPI module are completed by setting the SPI control register SSCON0 and the SPI status register SSCON1. After the configuration is complete, set SSCON0, SSCON1, SSDAT (SPI data registers) to complete data transmission.

During SPI communication, data is shifted in and out serially synchronously. The serial clock line (SCK) keeps the movement and sampling of data on the two serial data lines (MOSI and MISO) synchronized. If the slave is not selected, it cannot participate in activities on the SPI bus.

When the SPI master transmits data to the slave over the MOSI line, the slave sends data to the master over the MISO line in response, which enables synchronous full-duplex transmission of data sending and receiving under the same clock. The send shift register and the receive shift register use the same special function address. A write to the SPI data register SSDAT will write to the send shift register, and a read to the SSDAT register will get data from the receive shift register.

The SPI interface of some devices will lead to SS pins (select pins from the device, low efficiency). When communicating with THE SPI of SC92F859X, the connection mode of SS pins of other devices on the SPI bus should be connected according to different communication modes. The following table lists the connection modes of SS pins of other devices on the SPI bus under different SPI communication modes of SC92F859X:

SC92F859X SPI	Other devices on the SPI bus	Mode	Slave SS (Slave selection pin)
Master mode	Slave mode	One master and one slave	Pull down
		One master and multiple slaves	The SC92F859X leads to multiple I/Os, which are connected to the SS pin of the slave. Before data transmission, the SS pin of the slave device must be set low
Slave mode	Master mode	One master and one slave	Pull up

Master Mode

- **Mode Startup:**

  The SPI master device controls the start of all data transfers on the SPI bus. When the MSTR bit in the SPI control register SSCON0 is set to 1, the SPI runs in the master mode and only one master device can start the transfer.

- **Transmitting:**

  In SPI main mode, write a byte of data to THE SPI data register SSDAT, and the data will be written to the send shift buffer. If the send shift register already has a data, the main SPI generates a WCOL signal indicating that the write is too fast. But the data in the send shift register is not affected and the send is not interrupted. In addition, if the send shift register is not empty, the master device immediately sends the data in the shift register to the MOSI line serially according to the SPI clock frequency on the SCK. When the transfer is complete, the SPIF bit in the SSCON1 register is set to 1. If SPI interrupts are allowed, an interrupt is also generated at SPIF position 1.

- **Receiving:**

  When the master device transmits data to the slave device through the MOSI line, the corresponding slave device also transmits the contents of its transfer register to the receive shift register of the master device through the MISO line, realizing full duplex operation. Therefore, SPIF flag position 1 indicates both the completion of transmission and the completion of receiving data. The data received from the device is stored in the receive shift register of the main device in the MSB or LSB priority transfer direction. When a byte of data is fully moved into the receive register, the processor can get the data by reading the SSDAT register.

Slave mode

- **Mode Startup:**

  When the MSTR bit in the SPI control register SSCON0 register is cleared to 0, SPI runs in slave mode.

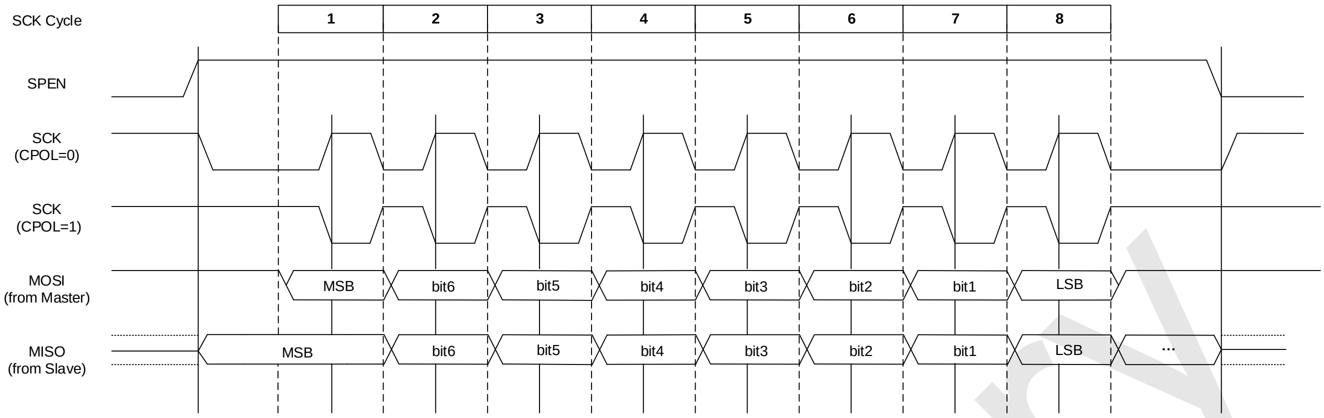
- **Transmitting and Receiving :**

  In slave mode, according to the SCK signal controlled by the master device, data is shifted in through the MOSI pin, and the MISO pin is shifted out. A bit counter records the number of edges of SCK. When the receiving shift register shifts in 8-bit data (one byte) and the sending shift register shifts out 8-bit data (one byte), the SPIF flag bit is set to 1. The data can be obtained by reading the SSDAT register. If the SPI interrupt is enabled, an interrupt will also be generated when SPIF is set. At this time, the receiving shift register keeps the original data and the SPIF bit is 1, so that the SPI slave device will not receive any data until SPIF is cleared. The SPI slave device must write the data to be transmitted into the transmit shift register before the master device starts a new data transmission. If no data is written before starting to send, the slave device will transmit the "0x00" byte to the master device. If the SSDAT write operation occurs during the transfer, the WCOL flag of the SPI slave device is set to 1, that is, if the transfer shift register already contains data, the WCOL bit of the SPI slave device is set to 1, indicating that the write SSDAT conflicts. But the data of the shift register is not affected, and the transmission will not be interrupted.

17.1.4 Transfer Form

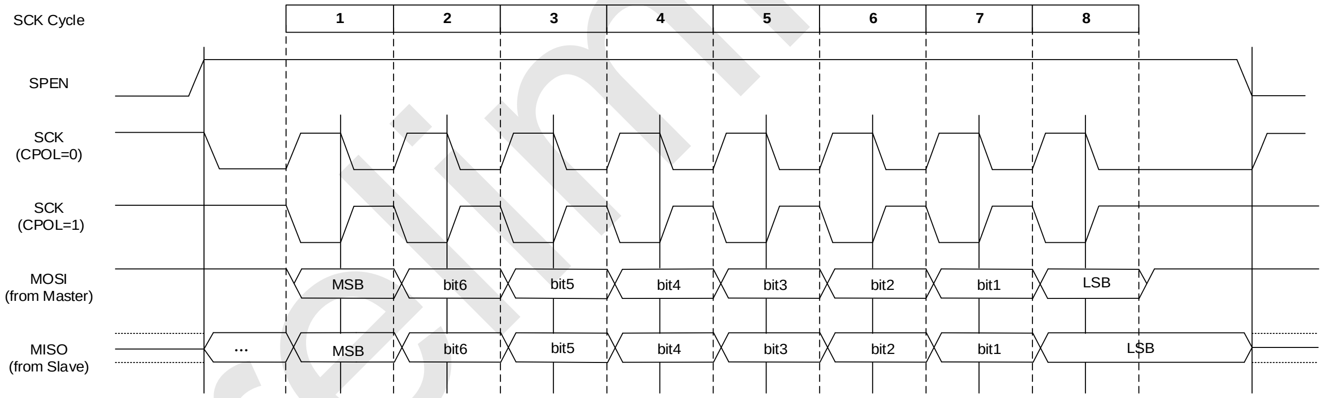
By software setting the CPOL bit and CPHA bit of the SPI control register SSCON0, the user can select four combinations of SPI clock polarity and phase. The CPOL bit defines the polarity of the clock, that is, the level state when idle, and it has little effect on the SPI transmission format. The CPHA bit defines the phase of the clock, that is, defines the clock edge that allows data sampling and shifting. In the two devices of master-slave communication, the setting of the clock polarity phase should be the same.

When CPHA = 0, the first edge of SCK captures data, and the slave must prepare the data before the first edge of SCK.



CPHA = 0 Data transfer diagram

When CPHA = 1, the master device outputs data to the MOSI line on the first edge of SCK, the slave device uses the first edge of SCK as the sending signal, and the second edge of SCK starts to capture data, So the user must write SSDAT inside two edges of the first SCK. This form of data transmission is the preferred form of communication between a master device and a slave device.



CPHA = 1 Data transfer diagram

17.1.5 Error Detection

Writing to SSDAT during the data transmission sequence will cause a write conflict, and the WCOL bit in the SPI status register SSCON1 is set to 1. WCOL bit 1 will not cause interruption, and transmission will not be aborted. The WCOL bit needs to be cleared by software.

17.2 TWI

SSMOD [1:0] = 10, SSI is configured as TWI interface. SC92F859X can only be used as a slave in TWI communication.

SSCON0 (9DH) TWI Control Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	TWEN	TWIF	-	GCA	AA	STATE[2: 0]		
R/W	R/W	R/W	-	Read	R/W	Read	Read	Read
POR	0	0	X	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7	**TWEN**	**TWI enable control** 0: Disable TWI 1: Enable TWI
6	**TWIF**	**TWI interrupt flag** 0：Cleared by software 1：Under the following conditions, the interrupt flag bit is set by hardware: ① The first frame address matches successfully ② Successfully receive or send 8-bit data ③ Receive repeated start condition ④ The slave receives a stop signal
4	**GCA**	**General address response flag** 0: Non-response general address 1: When GC is set to 1 and the general address matches at the same time, this bit is set to 1 by hardware and automatically cleare
3	**AA**	**Answer enable bit**

		0: not allow receive information sent by Master 1: allow receive information sent by Master
2~0	**STATE[2: 0]**	**State machine status flag** 000: The slave is in the idle state, waiting for TWEN to be set to 1, and detecting the TWI start signal. When the slave receives the stop condition, the jump will go to this state 001: The slave is receiving the first frame address and read/write bit (the 8th bit is the read/write bit, 1 is read, and 0 is write). The slave will jump to this state after receiving the start condition 010: slave receiving data status 011: slave sending data status 100: In the state of sending data from the slave, when the master returns to UACK, it jumps to this state and waits for a restart signal or a stop signal. 101: When the slave is in the sending state, writing 0 to AA will enter this state, waiting for a restart signal or a stop signal.

SSCON1 (9EH) TWI Address Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	TWA[6: 0]							GC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description

7~1	**TWA[6: 0]**	TWI address register TWA[6: 0] cannot be written as all 0s, 00H is dedicated to general address addressing. Invalid setting in Master mode
0	**GC**	TWI general address enable 0: Forbid to respond to general address 00H 1: Allow response to general address 00H

SSDAT (9FH) TWI Data Buffer Register (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	TWDAT[7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
7~0	**TWDAT [7: 0]**	TWI Data buffer register

17.2.1 Signal Description

TWI Clock Signal Line (SCL)

The clock signal is sent by the master and connected to all slaves. One byte of data is transmitted every 9 clock cycles. The first 8 cycles are used for data transmission, and the last clock is used as the receiver's response clock. It should be high when it is idle, pulled up by the pull-up resistor on the SCL line.

TWI Data Signal Line (SDA)

SDA is a bidirectional signal line, which should be high when it is idle, and is pulled high by the pull-up resistor on the SDA line.

17.2.2 Slave Operating Mode

- **Mode Start:**

  When the TWI enable flag is turned on (TWEN = 1) and the start signal sent by the Master is received at the same time, the mode is started.

  The slave enters the state of receiving the first frame address (STATE[2: 0] = 001) from the idle mode (STATE[2: 0] = 000), and waits for the first frame of data from the master. The first frame of data is sent by the Master, including 7-bit address bits and 1 bit for reading and writing. All slaves on the TWI bus will receive the first frame of data from the Master. The Master releases the SDA signal line after sending the first frame of data. If the address sent by the Master is the same as the value in a slave's own address register, it means that the slave is selected. The selected slave will judge the 8th bit on the bus, that is, the data read and write bit (=1, read command; =0, write command), then occupy the SDA signal line, give the Master a low-level response signal in the 9th clock cycle of SCL, and then release the bus. After the slave is selected, it will enter different states according to the different read and write bits:

- **Non-general Address Response, Slave Device Receiving Mode:**

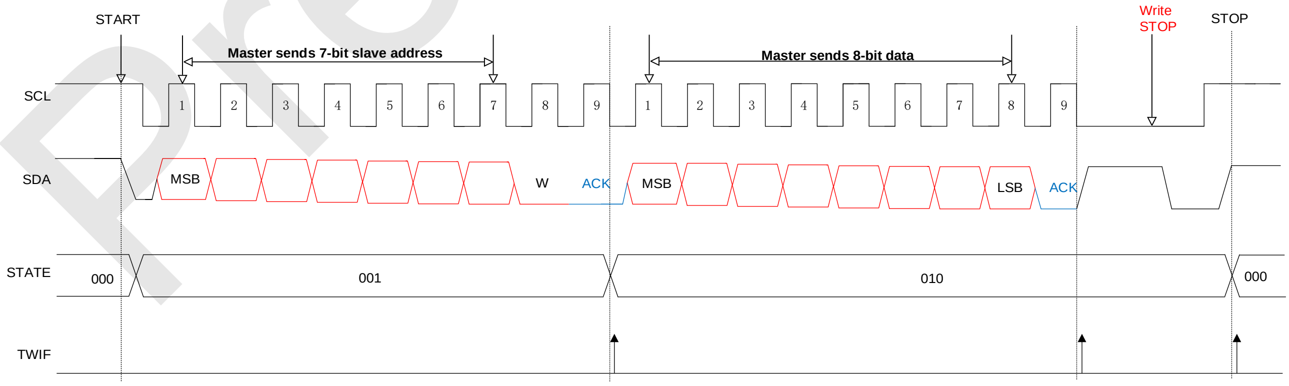
  If the read/write bit received in the first frame is write (0), the slave enters the slave receiving state (STATE [2: 0] = 010) and waits for the data sent by the Master. The master must release the bus every time it sends 8 bits and wait for the response signal from the slave in the 9th cycle.

  1. If the response signal of the slave is low, the communication of the master can be in the following three ways:

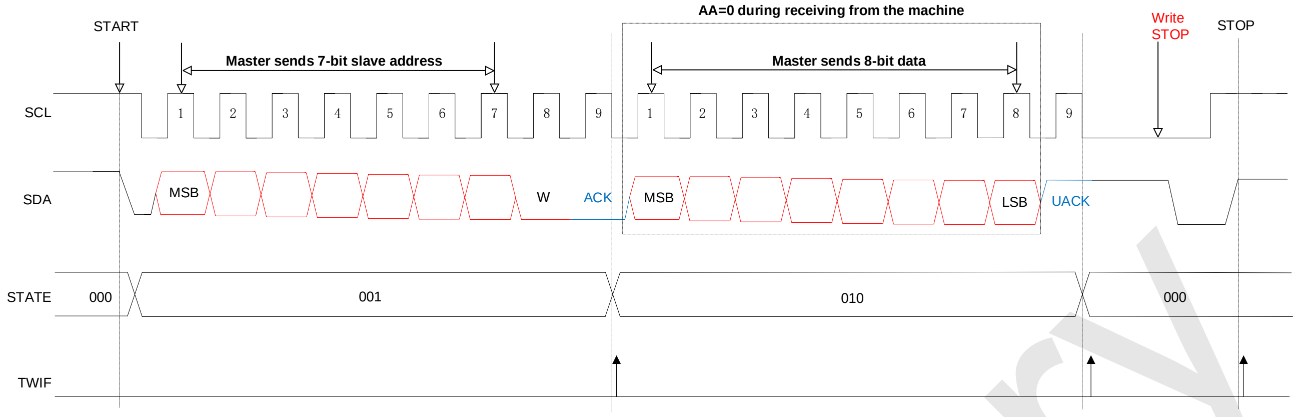
     1) Continue to send data;

     2) Resend the start signal (start), at this time the slave re-enters the state of receiving the first frame address (STATE[2: 0] = 001);

     3) Send a stop signal to indicate the end of this transmission, and the slave returns to the idle state, waiting for the next start signal from the Master.



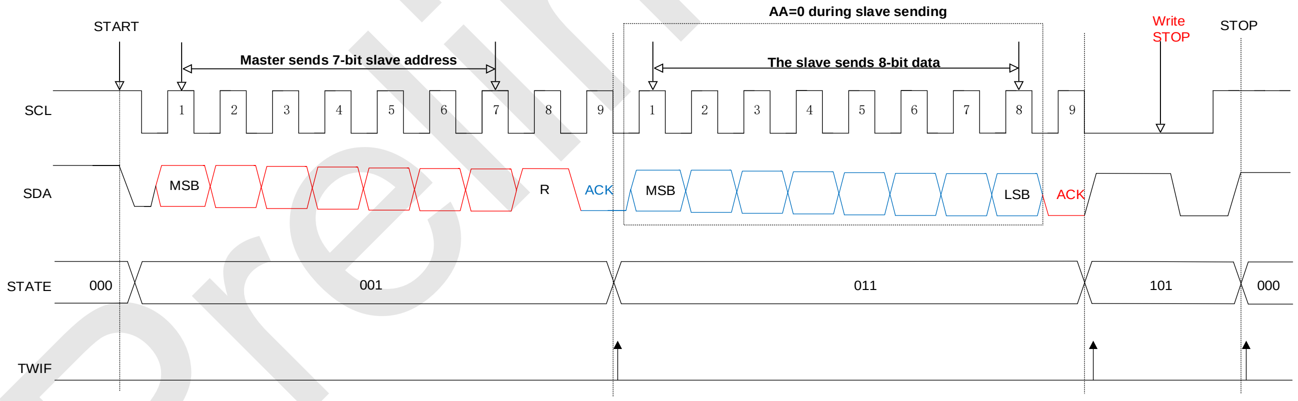
  2. If the slave responds to a high level (during the receiving process, the AA value in the slave register is rewritten to 0), it means that after the current byte is transmitted, the slave will actively end the transmission and return to the idle state (STATE [2: 0] = 000), no longer receive data from the Master.



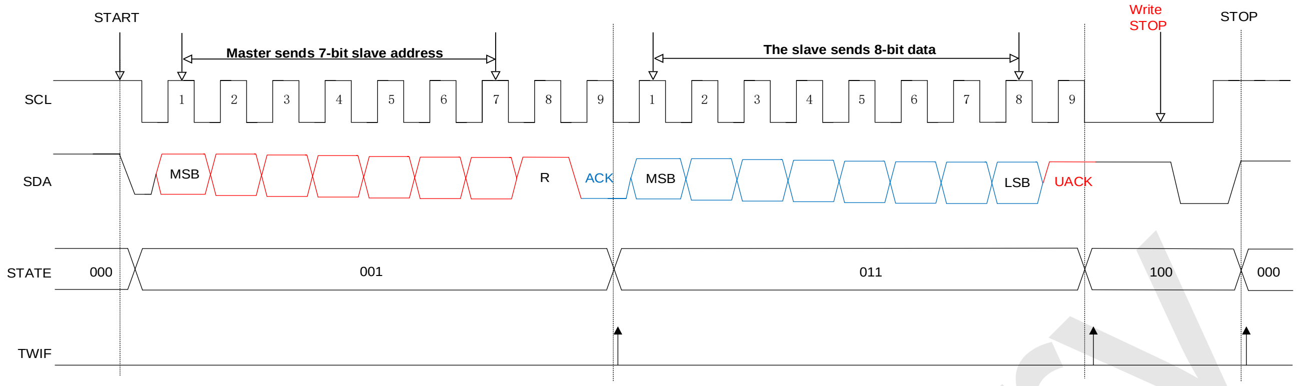
- **Non-general Address Response, Slave Device Transmitting Mode:**

If the read/write bit received in the first frame is read (1), the slave will occupy the bus and send data to the Master. Every time 8 bits of data are sent, the slave releases the bus and waits for the response from the master:

1. If the master responds with a low level, the slave continues to send data. In the process of sending, if the AA value in the slave register is rewritten to 0, the slave will actively end the transmission and release the bus after the current byte is transmitted, and wait for the stop signal or restart signal of the master (STATE[2: 0] = 101).



2. If the master responds to a high level, the slave STATE[2: 0] = 100, waiting for the master's stop signal or restart signal.



- **General Address Response:**

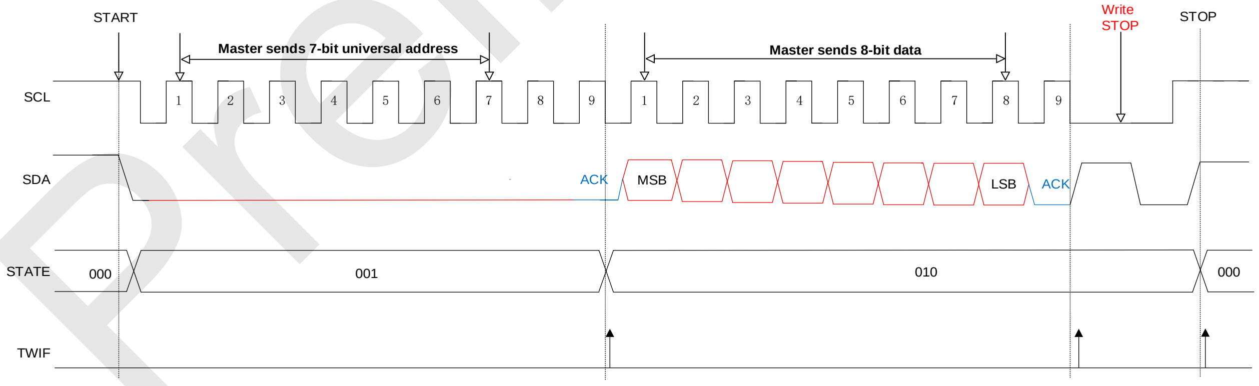
When GC=1, the general address is allowed to be used at this time. The slave enters the state of receiving the first frame address (STATE[2: 0] = 001), the address bit data in the first frame of data received is 0x00, and all slaves respond to the master at this time. The read and write bits sent by the master must be write (0), and all slaves enter the state of receiving data (STATE[2: 0] = 010) after receiving. The Master releases the SDA line every time 8 data is sent, and reads the status on the SDA line:

1. If there is a response from the slave, the communication of the master can be in the following three ways:

   1) Continue to send data;

   2) Restart;

   3) Send a stop signal to end this communication.



2. If no slave responds, SDA is idle.

Note: When using the universal address in the one-master multiple-slave mode, the read and write bits sent by the Master cannot be in the read (1) state, otherwise, all devices on the bus will respond except for the device sending the data.

17.2.3 Slave Mode Operation Steps

1. Configure SSMOD[1: 0] and select TWI mode;

2. Configure the TWIn control registers SSCON0;

3. Configure the TWI address register SSCON1;

4. If the slave receives data, it waits for the interrupt flag bit TWIF in SSCON0 to be set. Every time the slave receives 8 bits of data, TWIF will be set to 1. The interrupt flag bit TWIF needs to be manually cleared;

5. If the slave sends data, write the data to be sent into TWDAT, and TWI will automatically send the data. Every 8 bits are sent, the interrupt flag bit TWIF will be set.

17.3 Serial Interface (UART)

SSMOD[1: 0] = 11, SSI is configured as UART interface.

SSCON0 (9DH) Serial Port 1 Control Register (Read/Write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	SM0	-	SM2	REN	TB8	RB8	TI	RI
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	x	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**SM0**	Serial Communication Mode Control Bit 0: Mode 1, 10-bit full-duplex asynchronous communication composing of 1 starting bit, 8 data bits and 1 stopping bit, with communication baud rate changeable; 1: Mode 3, 11-bit full-duplex asynchronous communication, composing of 1 starting bit, 8 data bits and 1 programmable 9th bit and 1 stopping bit, with communication baud rate changeable.
5	**SM2**	Serial Communication Mode Control Bit 2, this control bit is only valid for mode 3 0: Configure RI for receiving each complete data frame to generate interrupt request; 1: When receiving a complete data frame and only when RB8=1, will RI be configured to generate interrupt request.
4	**REN**	Receive Allowing Control Bit 0: Receiving data not allowed; 1: Receiving data allowed.
3	**TB8**	Only valid for mode 3, 9th bit of receiving data

2	**RB8**	Only valid for mode 3, 9th bit of receiving data
1	**TI**	Transmit Interrupt Flag Bit
0	**RI**	Receive Interrupt Flag Bit
6	-	Reserved

SSCON1 (9EH) Serial Port 1 Baud Rate Control Register Low (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	BAUDL [7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

SSCON2 (95H) Serial Port 1 Baud Rate Control Register Low (Read/Write)

Bit Number	7	6	5	4	3	2	1	0
Bit Mnemonic	BAUDH [7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description

7 ~ 0	**BAUD [15: 0]**	Serial Port Baud Rate Control Bit $\mathrm{BaudRate} = \frac{\mathrm{fsys}}{\mathrm{BAUD1H, BAUD1L}}$ **Note: [BAUD1H, BAUD1L] must be larger than 0x0010**

SSDAT (9FH) Serial Port Data Cache Register (Read/Write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	SBUF[7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7 ~ 0	**SBUF[7: 0]**	Serial Data Buffer SBUF contains two registers: one for transmit shift register and one for receiving latch, data writing to SBUF will be sent to shift register and initiate transmitting process, reading SBUF1 will return the contents of receiving latch.

18 High-speed Analog-to-Digital Converter (ADC)

The SC92F859X integrates 17 channels of 12-bit high precision successive approximation type ADC, and the external 16 channels of ADC and other functions of the IO port are multiplexed. The internal channel can be connected to 1/4 VDD, and the internal 1.024V, 2.4V and 2.048V reference voltage is used for Measure the VDD voltage. 1 MHz super-high-speed sampling clock, the total time from sampling to completion of conversion is as low as 2μs

There are 4 choices for the ADC reference voltage of SC92F859X:

① VDD pin (that is directly the internal VDD);

② The reference voltage output by the internal Regulator is exactly 1.024V.

③ The reference voltage output by the internal Regulator is exactly 2.4V (at this time MCU supply voltage VDD can not be less than 2.9V)

④ The reference voltage output by the internal Regulator is accurately 2.048V.

Note: The sampling clock of the ADC circuit follows the Fsys.

18.1 ADC-related Registers

ADCCON (ADH) ADC Control Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	ADCEN	ADCS	EOC/ADCIF	ADCIS[4: 0]				
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	n

Bit number	Bit Mnemonic	Description
7	**ADCEN**	Power on ADC 0: Disable the ADC module power 1: Enable the ADC module power
6	**ADCS**	ADC start trigger control (ADC Start)

		Write “1” to this bit to start ADC conversion, that is, this bit is only the trigger signal of ADC conversion. This bit can only be written with 1 to be valid.
5	**EOC /ADCIF**	Conversion complete/ADC Interrupt Flag (End Of Conversion / ADC Interrupt Flag) 0: Conversion has not been completed 1: ADC conversion is complete. Need user software to clear ADC conversion complete flag EOC: when the user sets ADCS to start conversion, this bit will be automatically cleared to 0 by the hardware; when the conversion is completed, this bit will be automatically set to 1 by the hardware; ADC interrupt request flag ADCIF: This bit is also used as an interrupt request flag for ADC interrupt. If the user enables the ADC interrupt, the user must clear this bit by software after the ADC interrupt occurs.
4~0	**ADCIS[4: 0]**	ADC Input Selector (ADC Input Selector) 00000: select AIN0 as ADC input 00001: select AIN1 as ADC input 00010: select AIN2 as ADC input 00011: select AIN3 as ADC input 00100: select AIN4 as ADC input 00101: select AIN5 as ADC input 00110: select AIN6 as ADC input 00111: select AIN7 as ADC input 01000: select AIN8 as ADC input 01001: select AIN9 as ADC input 01010: select AIN10 as ADC input 01011: select AIN11 as ADC input 01100: select AIN12 as ADC input 01101: select AIN13 as ADC input 01110: select AIN14 as ADC input

		01111: select AIN15 as ADC input 10000~11110: reserved 11111: ADC input is 1/4 V_{DD}, which can be used to measure power supply voltage

ADCCFG2 (AAH) ADC Set Register 2 (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	-	-	-	LOWSP	ADCCK[1:0]	
R/W	-	-	-	-	-	R/W	R/W	R/W
POR	x	x	x	x	x	0	0	0

Bit number	Bit Mnemonic	Description
2	**LOWSP**	ADC sampling clocks selector: 0: Set ADC sampling time to 6 ADC sampling clock cycles 1: Set ADC sampling time to 36 ADC sampling clock cycles LOWSP controls the sampling clock frequency of ADC. The conversion clock frequency of ADC is controlled by ADCCK[1:0] and is not affected by LOWSP bit ADC needs 6 or 36 ADC sampling clocks plus 14 ADC conversion clocks to complete the whole process from sampling to conversion. Therefore, in actual use, the total time of ADC from sampling to completion of conversion is calculated as follows: LOWSP=0: Tadc1=(6+14)/Fadc LOWSP=1: Tadc2=(36+14)/ Fadc
1~0	**ADCCK[1:0]**	ADC Sampling Clocks Selector 00: Set the ADC clock frequency Fadc = Fsys/16 01: Set ADC sampling clock frequency Fadc = Fsys/12

		10: Set ADC sampling clock frequency Fadc = Fsys/6 11: Set the ADC clock frequency Fadc = Fsys/4 **Note: The sampling clock of the ADC circuit follows the Fsys**
7~4	**-**	Reserved

ADCCFG0 (ABH) ADC Set Register 0 (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	EAIN7	EAIN6	EAIN5	EAIN4	EAIN3	EAIN2	EAIN1	EAIN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

ADCCFG1 (ACH) ADC Set Register 1 (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	EAIN15	EAIN14	EAIN13	EAIN12	EAIN11	EAIN10	EAIN9	EAIN8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
0	**EAINx** **(x=0~15)**	**ADC port setting register** 0: Set AINx as IO port

		1: Set AINx as ADC input and automatically remove the pull-up resistor.

OP_CTM1 (C2H@FFH) Customer Option Register 1(Read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	VREFS[1: 0]		OP_BL	DISJTG	IAPS [1: 0]		LDSIZE [1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	Only read	
POR	n	n	n	n	n	n	n	n

Bit Number	Bit Mnemonic	Description
7~6	**VREFS[1: 0]**	IAP spatial range selection 00: Set the VREF of ADC to VDD 01: Set the VREF of the ADC to an internally accurate 1.024V 10: Set the VREF of the ADC to an internally accurate 2.4V 11: Set the VREF of the ADC to an internally accurate 2.048V

ADCVL (AEH) ADC Conversion Value Register (low bit) (read/write)

Bit number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	ADCV[3: 0]				-	-	-	-
R/W	R/W	R/W	R/W	R/W	-	-	-	-
POR	1	1	1	1	x	x	x	x

ADCVH (AFH) ADC Conversion Value Register (high bit) (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	ADCV[11: 4]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

Bit number	Bit Mnemonic	Description
11~4	**ADCV[11: 4]**	High 8-bits of ADC conversion value
3~0	**ADCV[3: 0]**	Low 4-bits of ADC conversion value

IE (A8H) Interrupt Enable Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	EA	EADC	ET2	EUART	ET1	EINT1	ET0	EINT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
6	**EADC**	ADC interrupt enable control 0: Do not allow EOC/ADCIF to generate interrupts

		1: Enable EOC/ADCIF to generate interrupt

IP (B8H) Interrupt Priority Control Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	IPADC	IPT2	IPUART	IPT1	IPINT1	IPT0	IPINT0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	x	0	0	0	0	0	0	0

Bit number	Bit Mnemonic	Description
6	**IPADC**	ADC interrupt priority selection 0: Set the interrupt priority of ADC to "low" 1: Set the interrupt priority of ADC to "High"

18.2 ADC Conversion Steps

The actual operation steps required for the user to perform ADC conversion are as follows:

① Set the ADC input pin; (set the bit corresponding to AINx as ADC input, usually the ADC pin will be fixed in advance);

② Set ADC reference voltage Vref, set the frequency used for ADC conversion;

③ Enable the ADC module power supply;

④ Select ADC input channel; (set ADCIS bit, select ADC input channel);

⑤ Start ADCS and start conversion;

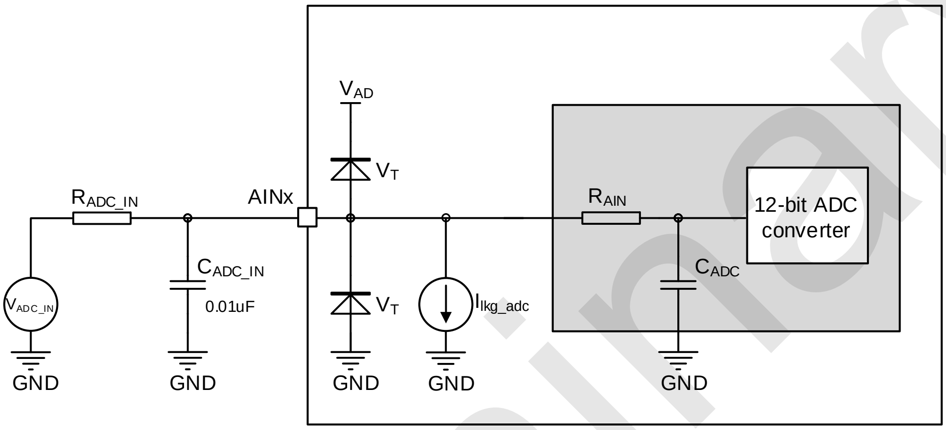
⑥ Wait for EOC/ADCIF=1. If the ADC interrupt is enabled, the ADC interrupt will be generated. The user needs to clear the EOC/ADCIF flag by software;

⑦ Get 12-bit data from ADCVH and ADCVL, first high bit and then low bit, one conversion is completed;

⑧ If you do not change the input channel, repeat steps 5~7 for the next conversion.

Note: Before setting IE [6] (EADC), the user is better to clear EOC/ADCIF with software, and also clear the EOC/ADCIF when the ADC interrupt service routine is executed to avoid continuous ADC interrupts.

18.3 ADC



Note: The ADC Characteristics, please see the 22.6 ADC Characteristics for the detailed information.

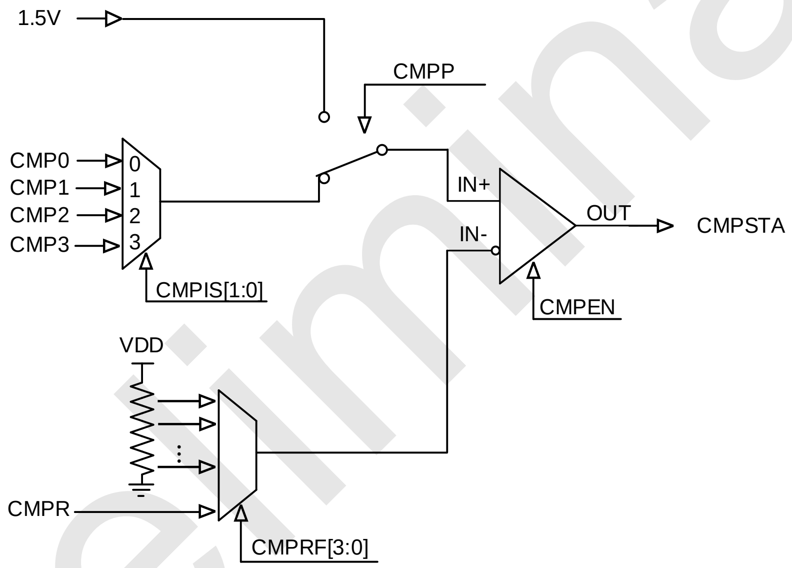
19 Analog Comparator

The SC92F859X has a built-in analog comparator, which can be used for alarm circuit, power supply voltage monitoring circuit, zero-crossing detection circuit, etc.

This comparator has four analog signal positive input terminals: CMP0~3, which can be switched through CMPIS [1: 0]. The negative input voltage can be switched to one of the external voltage on the CMPR pin or the internal 16-level comparison voltage through CMPRF[3: 0].

The CMPIM[1: 0] can conveniently set the interrupt mode of the comparator. When the interrupt condition set by CMPIM[1: 0] occurs, the comparator interrupt flag CMPIF will be set to 1, and the interrupt flag needs to be cleared by software.

19.1 Block Diagram of Analog Comparator



Block Diagram of Analog Comparator

CMPCON (B7H) Analog Comparator Control Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	CMPEN	CMPIF	CMPSTA	-	CMPRF[3: 0]			
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W

POR	0	0	0	x	0	0	0	0

Bit number	Bit Mnemonic	Description
7	**CMPEN**	**Analog comparator enable control bit** 0: Disable the analog comparator; 1: Enable analog comparator
6	**CMPIF**	**Analog comparator interrupt flag** 0: Comparator interrupt is not triggered; 1: When the comparator meets the interrupt trigger condition, this bit will be automatically set to 1 by the hardware. If IE1[5] (ECMP) is also set to 1 at this time, the comparator interrupt is generated. After the comparator interrupt occurs, the hardware will not automatically clear this bit, the user's software must be responsible for clearing this bit.
5	**CMPSTA**	**Analog comparator output status** 0: The voltage at the positive terminal of the comparator is less than the voltage at the negative terminal 1: The voltage at the positive terminal of the comparator is greater than the voltage at the negative terminal
3~0	**CMPRF[3: 0]**	**Analog comparator negative terminal comparison voltage selection:** 0000: Select CMPR as the comparison voltage of the analog comparator; 0001: Select 1/16V_{DD} as the comparison voltage of the analog comparator; 0010: Select 2/16V_{DD} as the comparison voltage of the analog comparator; 0011: Select 3/16V_{DD} as the comparison voltage of the analog comparator; 0100: Select 4/16V_{DD} as the comparison voltage of the analog comparator;

		0101: Select 5/16V_{DD} as the comparison voltage of the analog comparator; 0110: Select 6/16V_{DD} as the comparison voltage of the analog comparator; 0111: Select 7/16V_{DD} as the comparison voltage of the analog comparator; 1000: Select 8/16V_{DD} as the comparison voltage of the analog comparator; 1001: Select 9/16V_{DD} as the comparison voltage of the analog comparator; 1010: Select 10/16V_{DD} as the comparison voltage of the analog comparator; 1011: Select 11/16V_{DD} as the comparison voltage of the analog comparator; 1100: Select 12/16V_{DD} as the comparison voltage of the analog comparator; 1101: Select 13/16V_{DD} as the comparison voltage of the analog comparator; 1110: Select 14/16V_{DD} as the comparison voltage of the analog comparator; 1111: Select 15/16V_{DD} as the comparison voltage of the analog comparator;
4	-	Reserved

CMPCFG (B6H) Analog Comparator Setting Register (read/write)

Bit number	7	6	5	4	3	2	1	0
Bit Mnemonic	-	-	-	CMPP	CMPIM[1: 0]		CMPIS[1: 0]	
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
POR	x	x	x	0	0	0	0	0

Bit number	Bit Mnemonic	Description
4	**CMPP**	**Analog comparator positive input options:** 0: The positive input of the analog comparator is CMP0~3, which is set by CMPIS[1:0] 1: The positive input of the analog comparator is the internal 1.5V reference voltage
3~2	**CMPIM[1: 0]**	**Analog comparator interrupt mode selection:** 00: no interrupt 01: Rising-edge interrupt: An interrupt will be generated after IN+ is less than IN- to greater than IN-; 10: Falling-edge interrupt: An interrupt will be generated after IN+ is greater than IN- to less than IN-; 11: Double-edge interrupt: IN+ from less than IN- to greater than IN-, or IN+ from greater than IN- to less than IN- will generate an interrupt;
1~0	**CMPIS[1: 0]**	**Analog comparator positive terminal input channel selection:** 00: Select CMP0 as the input of the positive terminal of the analog comparator; 01: Select CMP1 as the input of the positive terminal of the analog comparator; 10: Select CMP2 as the input of the positive terminal of the analog comparator; 11: Select CMP3 as the input of the positive terminal of the analog comparator;
7~4	**-**	Reserved

20 High Sensitivity Mode Space Capacitive Touch Circuit

The SC92F859X has a 31 - channel high sensitivity mode capacitive touch circuit. Its features are as follows:

1. It can be used for high sensitivity touch applications such as space button touch and proximity sensing
2. It can pass 10V dynamic CS test
3. It can realize 31 touch control keys and derivative functions
4. High flexibility to develop software library support, low development difficulty
5. Automated debugging software support, intelligent development
6. The touch module can work in the low-power mode under the MCU STOP mode

20.1 Power Consumption of TouchKey Circuits

The SC92F859X allows touch scanning to be enabled in STOP Mode: this approach can reduce the overall power consumption of the MCU for touch applications with low-power requirements.

Users can understand that the touch circuit of SC92F859X has two power consumption modes:

1. Normal operation mode

2. Low-power operation mode

The two power consumption modes are defined as follows:

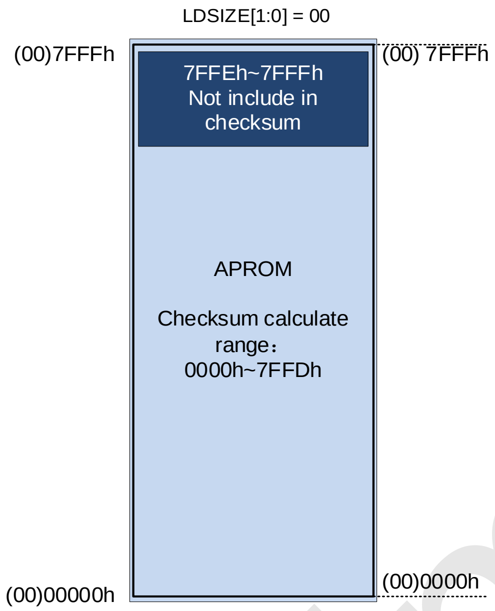
Instructions	Normal operation mode	Low-power operation mode
CPU	RUN（Normal mode）	Stop（STOP Mode）
Touch the circuit	RUN	RUN

Note: Users can quickly and easily achieve the required touch functions by using the touch key library file provided by Sin One (available for download from the official website of Sin One).

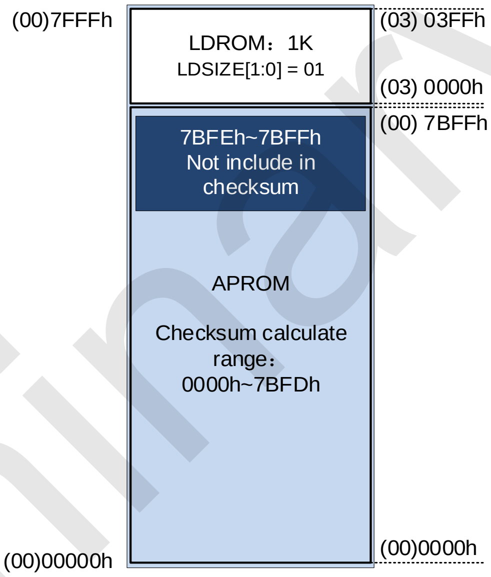
21 CheckSum Module

SC92F859X has built in a checksum module to generate 16-bit checksum value of protocol. User can confirm whether the protocol in code area is right or wrong by comparing checksum value and theoretical value.

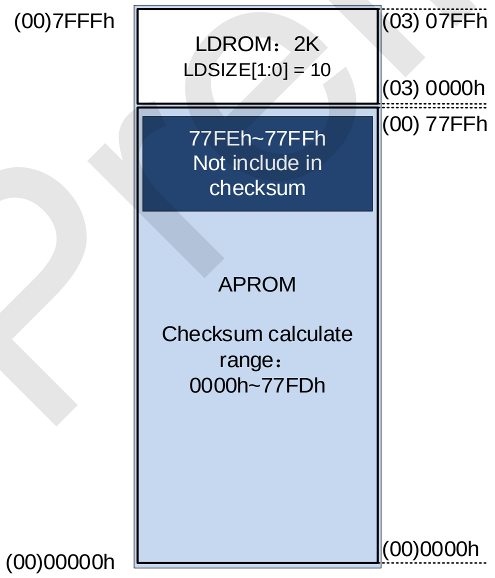
Checksum value is the sum of data in the whole code memory, but the last two bytes won't be included. The range of APROM will change as the LDSIZE[1:0] changes, therefore, the range of Checksum module will be different, as shown below:



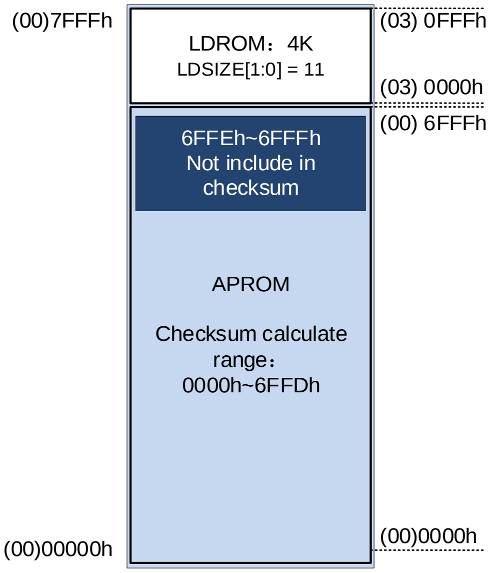
APROM:32KB



APROM:31KB



APROM:30KB



APROM:28KB

Note:

1. **The Checksum value is the sum of the whole APROM minus the last two bytes.**
2. **If there is remaining data in address unit, it will lead to the difference between Checksum value and theoretical value. Thus, we suggest user to clear the entire APROM before downloading code in order to guarantee that the Checksum value and theoretical value are the same.**

21.1 CheckSum-Related Registers

CHKSUML (FCH) Check Sum Result Register Low Bit (Read/Write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	CHKSUML[7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7 ~ 0	**CHKSUML [7: 0]**	CheckSum Result Register Low Bit

CHKSUMH (FDH) Check Sum Result Register High Bit (Read/Write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	CHKSUMH[7: 0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7 ~ 0	**CHKSUMH [7: 0]**	CheckSum Result Register High Bit

OPERCON (EFH) Arithmetic Control Register (Read/Write)

Bit Number	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Bit Mnemonic	-	-	-	-	-	-	-	CHKSUMS
R/W	-	-	-	-	-	-	-	R/W
POR	x	x	x	x	x	x	x	0

Bit Number	Bit Mnemonic	Description
0	**CHKSUMS**	CheckSum Operation Starts Trigger **C**ontrol Bit (Start) Write “1” for this bit, start to conduct Check sum calculation. This bit is valid for only writing 1.

22 Electrical Characteristics

22.1 Absolute Maximum Ratings

Symbol	Parameter	Min Value	Max Value	Unit
VDD/VSS	DC supply voltage	-0.3	5.5	V
Voltage ON any Pin	Input/output voltage of any pin	-0.3	V_{DD}+0.3	V
T_A	Ambient temperature	-40	105	℃
T_{STG}	Storage temperature	-55	125	℃
I_{VDD}	Current through VDD	-	200	mA
I_{VSS}	Current through VSS	-	200	mA

22.2 Recommended Operating Conditions

Symbol	Parameter	Min Value	Max Value	Unit	System Clock Frequency
V_{DD}	Operating voltage	2.0	5.5	V	32Mhz
T_A	Ambient temperature	-40	105	℃	-

22.3 FLASH ROM Parameter

Symbol	Parameter	Min Value	Typical Value	Max Value	Unit	Testing Conditions
N_{END}	Erase times	100,000			Cycles	
T_{IDR}	Data retention time	100			years	TA=+25°C

$T_{S\text{-}Erase}$	Single sector erase time		5		ms	TA=+25°C
$T_{all\text{-}Erase}$	32K Flash ROM erase time		20		ms	TA=+25°C
T_{Write}	Single byte write time		30		µs	TA=+25°C

22.4 DC Characteristics

VDD = 5V, TA = +25℃,unless otherwise specified

Symbol	Parameter	Min Value	Typical Value	Max Value	Unit	Testing Conditions
Current						
I_{op1}	Operating current	-	3.5	-	mA	f_{SYS} =32MHz
I_{op2}	Operating current	-	2.5	-	mA	f_{SYS} =16MHz
I_{op3}	Operating current	-	1.8	-	mA	f_{SYS} =8MHz
I_{op4}	Operating current	-	1.4	-	mA	f_{SYS} =2.66MHz
I_{pd1}	Standby Current (Power Down Mode)	-	2.5	6.0	µA	
I_{IDL1}	Standby Current (IDLE Mode)	-	1.7	-	mA	f_{SYS} =32MHz
I_{BTM}	Base Timer Operating Current	-	1.3	3	µA	BTMFS[3: 0]= 1000 One interrupt occurs for every 4.0 seconds
I_{WDT}	WDT Current	-	1.3	3	µA	WDTCKS[2: 0]=

						000 WDT overflows every 500ms
I_{TK1}	TouchKey operating current (High Sensitivity Mode)	-	0.8	1.2	mA	
IO Port Features						
V_{IH1}	Input high voltage	$0.7V_{DD}$	-	$V_{DD}+0.3$	V	
V_{IL1}	Input low voltage	-0.3	-	$0.3V_{DD}$	V	
V_{IH2}	Input high voltage	$0.8V_{DD}$	-	V_{DD}	V	Schmidt trigger input: RST/tCK/SCK
V_{IL2}	Input low voltage	-0.2	-	$0.2V_{DD}$	V	
I_{OL1}	Output low current	-	27	-	mA	V_{Pin}=0.4V
I_{OL2}	Output low current	-	50	-	mA	V_{Pin}=0.8V
I_{OH1}	Output high current P3H-P5	-	10	-	mA	V_{Pin}=4.3V
I_{OH2}	Output high current P3H-P5	-	4	-	mA	V_{Pin}=4.7V
I_{OH3}	Output high current P0-P3L	-	10	-	mA	V_{Pin}=4.3V Pxyz=0, I_{OH} level 0
	Output high current P0-P3L	-	7	-	mA	V_{Pin}=4.3V Pxyz=1, I_{OH} level 1
	Output high curren P0-P3L	-	5	-	mA	V_{Pin}=4.3V Pxyz=2, I_{OH} level 2

	Output high current P0-P3L	-	3	-	mA	V_{Pin}=4.3V Pxyz=3, I_{OH} level 3
I_{OH4}	Output high current P0-P3L	-	4	-	mA	V_{Pin}=4.7V Pxyz=0, I_{OH} level 0
	Output high current P0-P3L	-	3	-	mA	V_{Pin}=4.7V Pxyz=1, I_{OH} level 1
	Output high current P0-P3L	-	2	-	mA	V_{Pin}=4.7V Pxyz=2, I_{OH} level 2
	Output high current P0-P3L	-	1	-	mA	V_{Pin}=4.7V Pxyz=3, I_{OH} level 3
I_{lkg1}	Input leakage current	-1	-	1	μA	Input only mode V_{IN}=V_{DD} or V_{SS}
R_{PH1}	Pull-up resistance	-	30	-	kΩ	

VDD = 3.3V, TA = +25℃, unless otherwise specified

Symbol	Parameter	Min Value	Typical Value	Max Value	Unit	Testing Conditions
Current						
I_{op5}	Operating current	-	3.3	-	mA	f_{SYS} =32MHz
I_{op6}	Operating current	-	2.3	-	mA	f_{SYS} =16MHz
I_{op7}	Operating current	-	1.8	-	mA	f_{SYS} =8MHz
I_{op8}	Operating current	-	1.4	-	mA	f_{SYS} =2.66MHz

I_{pd2}	Standby Current (Power Down Mode)	-	2.5	6.0	μA	
I_{IDL2}	Standby Current (IDLE Mode)	-	1.7	-	mA	
I_{TK2}	TouchKey operating current (High Sensitivity Mode)	-	0.4	0.6	mA	
I/O Port Features						
V_{IH3}	Input high voltage	$0.7V_{DD}$	-	$V_{DD}+0.3$	V	
V_{IL3}	Input low voltage	-0.3	-	$0.3V_{DD}$	V	
V_{IH4}	Input high voltage	$0.8V_{DD}$	-	V_{DD}	V	Schmidt trigger input: RST/tCK/SCK
V_{IL4}	Input low voltage	-0.2	-	$0.2V_{DD}$	V	
I_{OL3}	Output low current	-	20	-	mA	V_{Pin}=0.4V
I_{OL4}	Output low current	-	35	-	mA	V_{Pin}=0.8V
I_{OH5}	Output high current	-	3	-	mA	V_{Pin}=3.0V
I_{lkg2}	Input leakage current	-1	-	1	μA	Input only mode $V_{IN}=V_{DD}$ or V_{SS}
R_{PH2}	Pull-up resistance	-	55	-	kΩ	

22.5 AC Characteristics

(V_{DD} = 2.0V ~ 5.5V, T_A = +25℃, unless otherwise specified)

Symbol	Parameter	Min	Typical	Max	Unit	Conditions

T_{POR}	Power On Reset time	-	15	-	ms	
T_{PDW}	Power Down Mode waking-up time	-	65	130	µs	
T_{Reset}	Reset Pulse Width	18	-	-	µs	Valid for Low level
T_{LVR}	LVR De-Bounce time	-	30	-	µs	
f_{HRC}	RC oscillation stability	31.36	32	32.64	MHz	V_{DD}=2.0~5.5V T_A=-20~85 ℃

22.6 ADC Characteristics

TA = +25℃,unless otherwise specified

Symbol	Parameter	Min	Typical	Max	Unit	Conditions
V_{AD1}	ADC supply voltage 1	2.7	5.0	5.5	V	Vref = 2.4V
V_{AD2}	ADC supply voltage 2	2.4	5.0	5.5	V	Vref = 1.024V Or Vref = V_{DD}
V_{REF1}	Internal reference 1.024V voltage output	1.004	1.024	1.044	V	V_{DD}=5V or 3.3V
V_{REF2}	Internal reference 2.4V voltage output	2.38	2.40	2.42	V	
V_{REF3}	Internal reference 2.048V voltage output	2.028	2.048	2.068	V	
N_R	precision	-	12	-	bit	GND≤V_{AIN}≤V_{DD}
V_{AIN}	ADC input voltage	GND	-	VDD	V	
R_{AIN}	ADC input resistance	1	-	-	MΩ	V_{IN}=5V

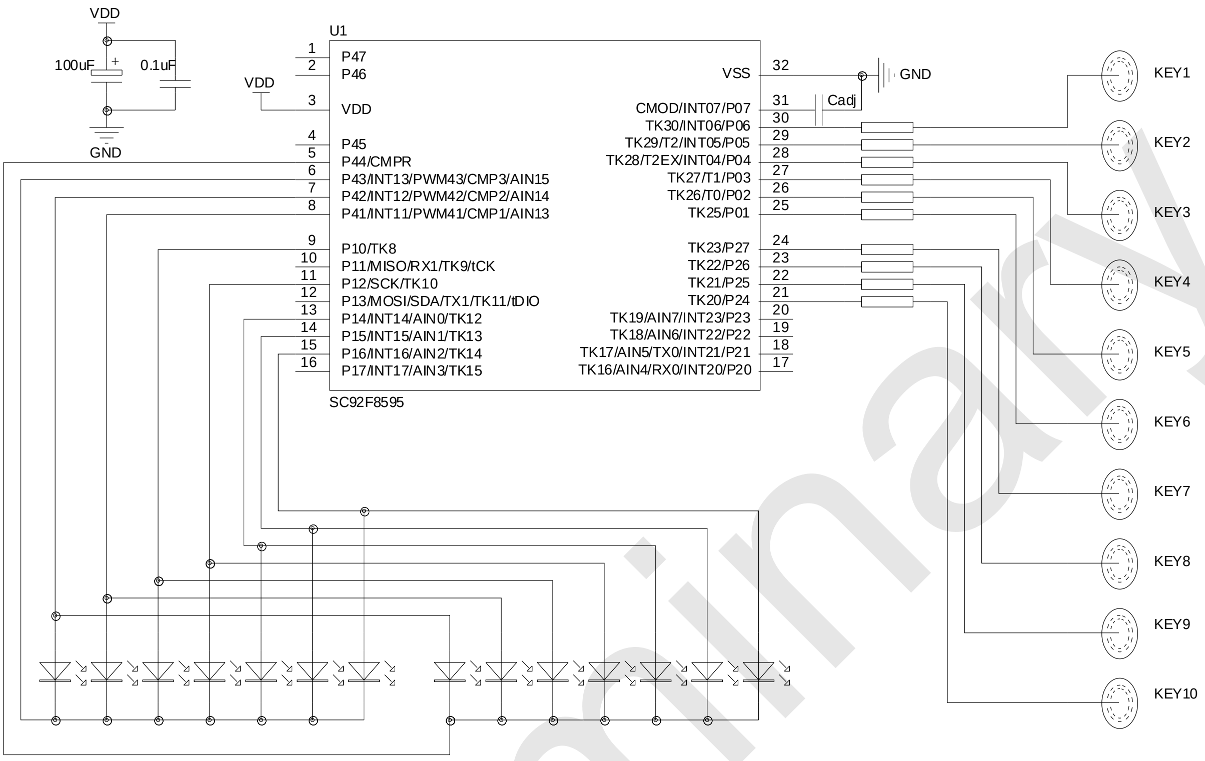
I_{lkg_ADC}	ADC input leakage current	-1	-	1	μA	V_{IN}=V_{AINx}
I_{ADC1}	ADC conversion current 1	-	-	2	mA	ADC Open Module VDD=5V
I_{ADC2}	ADC conversion current 2	-	-	1.8	mA	ADC Open Module VDD=3.3V
DNL	Differential nonlinear error	-	-	±3	LSB	VDD=5V VREF=5V
INL	Integrated nonlinear error	-	-	±4	LSB	
E_Z	Error of offset	-	-	±7	LSB	
E_F	Full scale error	-	-	±8	LSB	
E_{AD}	Total absolute error	-	-	±8	LSB	
T_{ADC1}	ADC conversion time 1	-	7.5	-	μs	ADC Clock = 2.67MHz ADC sampling period = 6
T_{ADC2}	ADC conversion time 2	-	15	-	μs	ADC Clock = 1.33MHz ADC sampling period= 6

22.7 Simulate comparator electrical characteristics

VDD = 5V, TA = +25℃,unless otherwise specified

Symbol	Parameter	Min	Typical	Max	Unit	Conditions
V_{CM}	Input voltage range	0	-	VDD	V	
V_{OS}	Offset voltage	-	10	30	mV	
V_{HYS}	Compare voltage back difference	-	25	-	mV	
I_{CMP}	The comparator converts the current	-	-	100	μA	V_{DD}=5V
T_{CMP}	The response time	-	-	2	μs	

23 Application Circuit



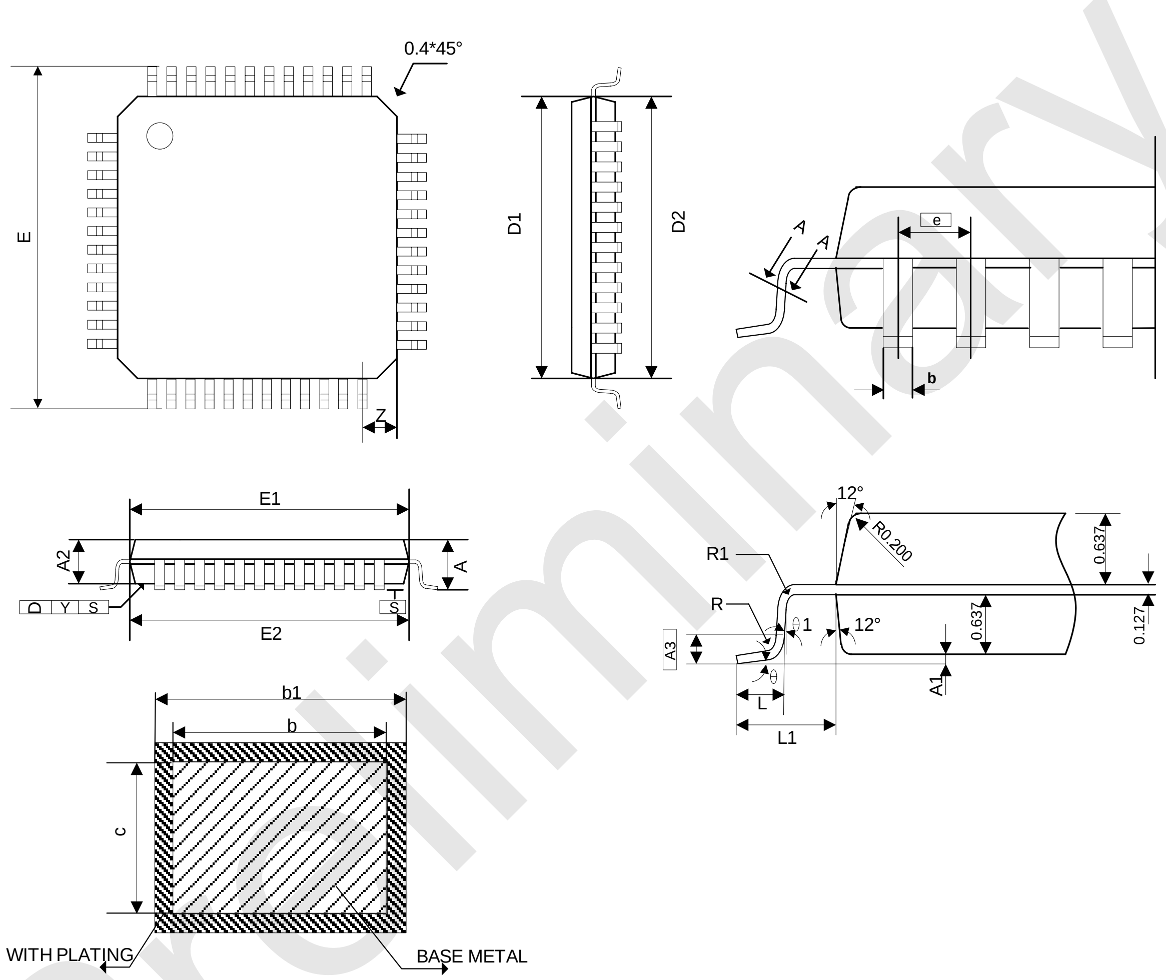
24 Ordering Information

PRODUCT NO	PKG	PACKING
SC92F8593M28U	SOP28	TUBE
SC92F8593X28U	TSSOP28	TUBE
SC92F8595Q32R	QFN32	PLATE
SC92F8595P32R	LQFP32	PLATE
SC92F8596P44R	LQFP44	PLATE
SC92F8597Q48R	QFN48	PLATE
SC92F8597P48R	LQFP48	PLATE

25 Packageing Information

SC92F8597P48R

LQFP48(7X7) Dimension Unit: mm

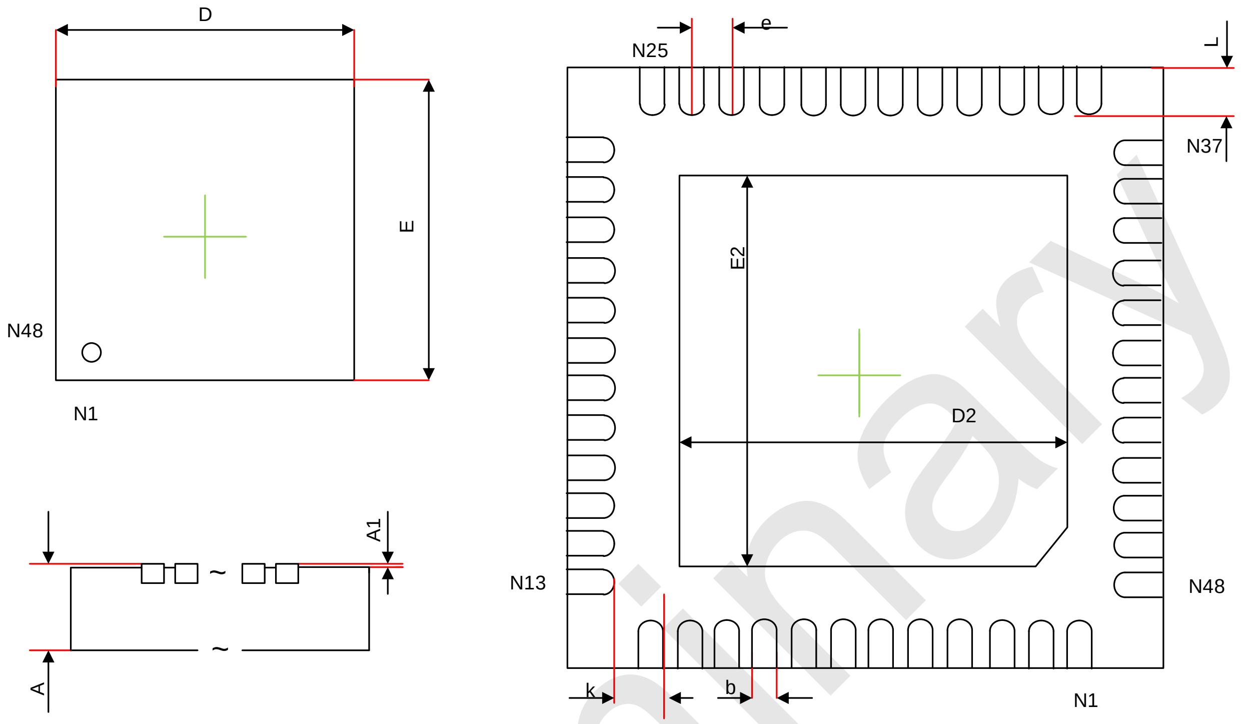


Symbol	mm (milimetre)		
	Min	Normal	Min
A	1.45	1.55	1.65
A1	0.01	--	0.21
A2	1.3	1.4	1.5
A3	--	0.254	--
b	0.15	0.20	0.25

b1	0.16	0.22	0.28
c	0.12	--	0.17
D1	6.85	6.95	7.05
D2	6.90	7.00	7.10
E	8.8	9.00	9.20
E1	6.85	6.95	7.05
E2	6.9	7.00	7.10
e	--	0.5	--
L	0.43	--	0.75
L1	0.90	1.0	1.10
R	0.1	--	0.25
R1	0.1	--	--
θ	0°	--	10°
θ1	0°	--	--
y	--	--	0.1
Z	--	0.75	--

SC92F8597Q48R

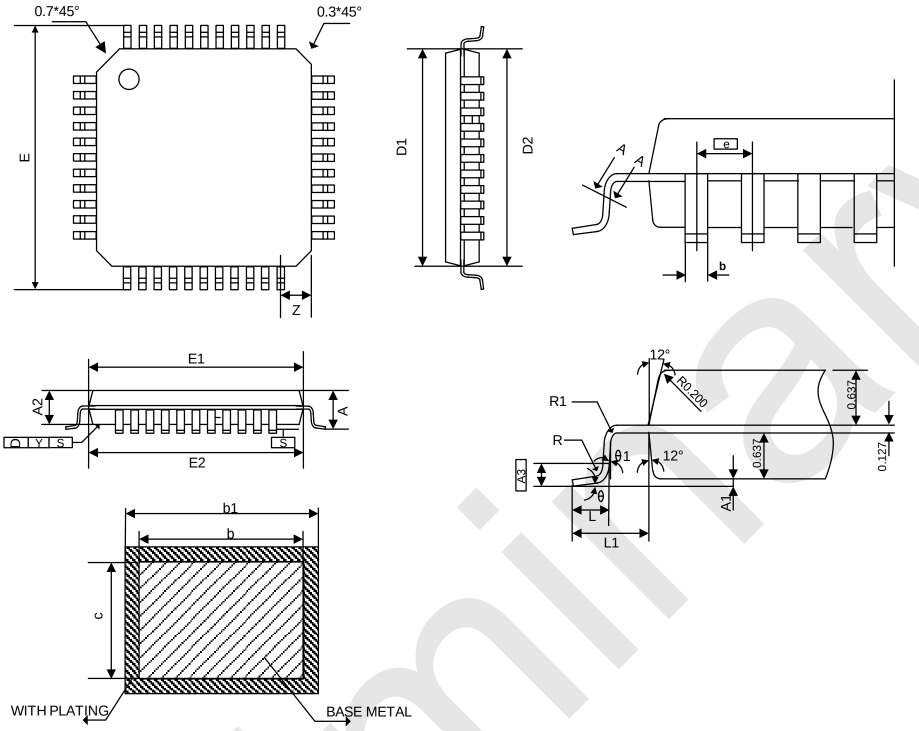
QFN48 (7X7) Dimension Unit: mm



Symbol	mm (milimetre)		
	Min	Normal	Min
A	0.70	0.75	0.80
A1	0	0.02	0.05
b	0.18	0.25	0.30
D	6.90	7.00	7.10
D2	5.10	5.30	5.50
e	0.50 BSC.		
k	0.50	--	--
E	6.90	7.00	7.10
E2	5.10	5.30	5.50
L	0.35	0.40	0.45

SC92F8596P44R

LQFP44(10X10) Dimension Unit: mm

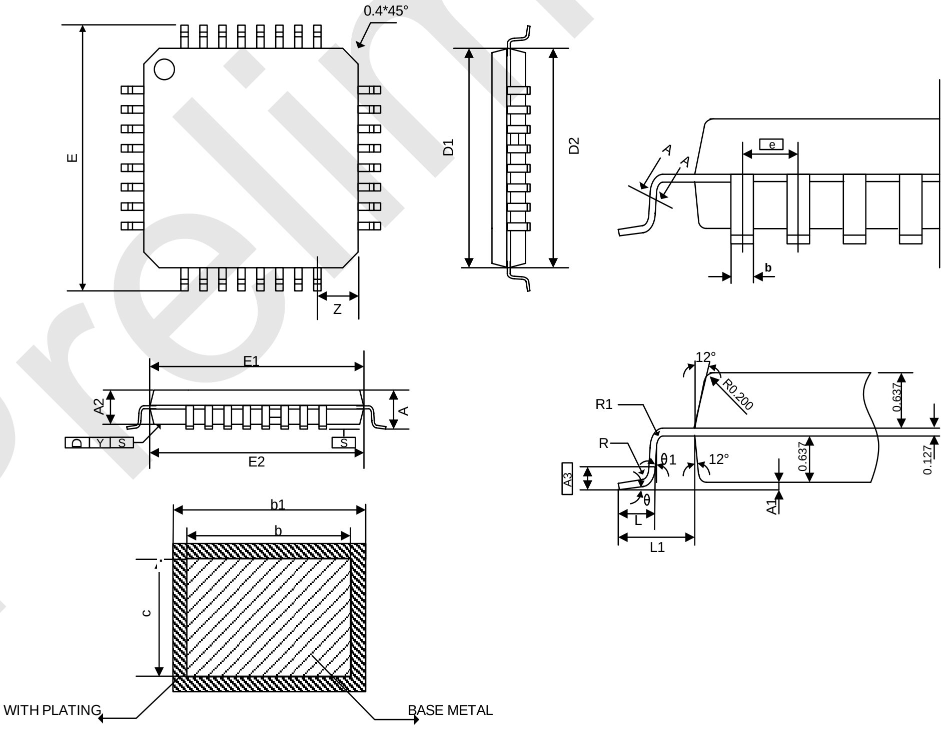


Symbol	mm (milimetre)		
	Min	Normal	Min
A	1.45	1.55	1.65
A1	0.015	--	0.21
A2	1.3	1.4	1.5
A3	--	0.254	--
b	0.25	0.30	0.36
b1	0.26	0.32	0.38
c	0.12	0.13	0.14
D1	9.85	9.95	10.05
D2	9.90	10.00	10.10
E	11.80	12.00	12.20
E1	9.85	9.95	10.05

E2	9.90	10.00	10.10
e	--	0.8	--
L	0.42	--	0.75
L1	0.95	1.0	1.15
R	0.08	--	0.25
R1	0.08	--	--
θ	0°	--	10°
θ1	0°	--	--
y	--	--	0.1
Z	--	1.0	--

SC92F8595P32R

LQFP32(7X7) Dimension Unit: mm

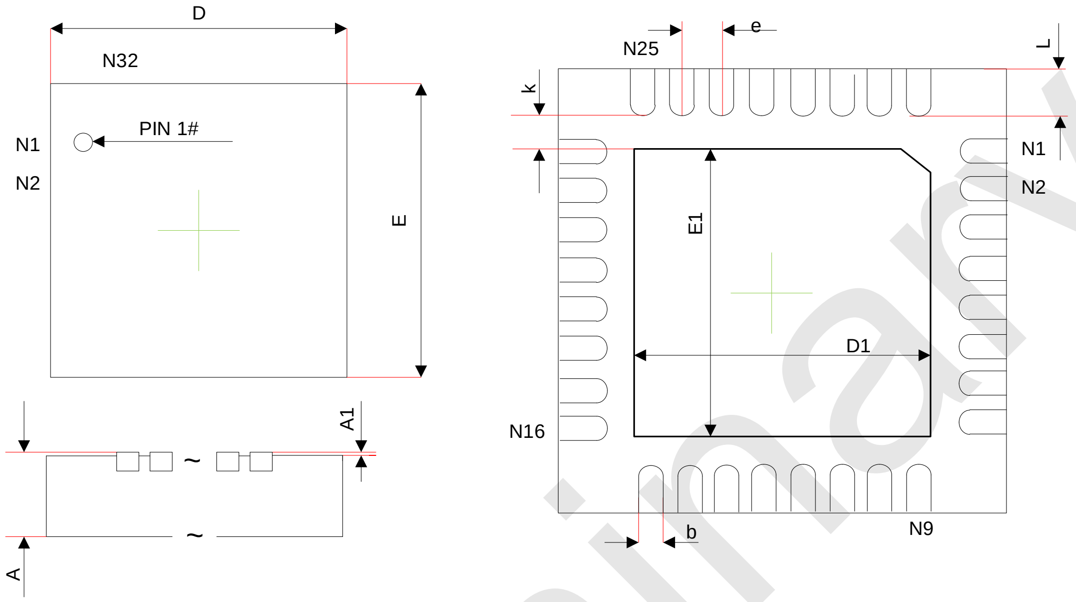


Symbol	mm (milimetre)		
	Min	Normal	Min

A	1.45	1.55	1.65
A1	0.01	--	0.21
A2	1.30	1.4	1.5
A3	--	0.254	--
b	0.30	0.35	0.41
b1	0.31	0.37	0.43
c	0.12	0.13	0.14
D1	6.85	6.95	7.05
D2	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.85	6.95	7.05
E2	6.90	7.00	7.10
e	--	0.8	--
L	0.43	--	0.75
L1	0.90	1.0	1.10
R	0.1	--	0.25
R1	0.1	--	--
θ	0°	--	10°
θ1	0°	--	--
y	--	--	0.1
Z	--	0.70	--

SC92F8595Q32R

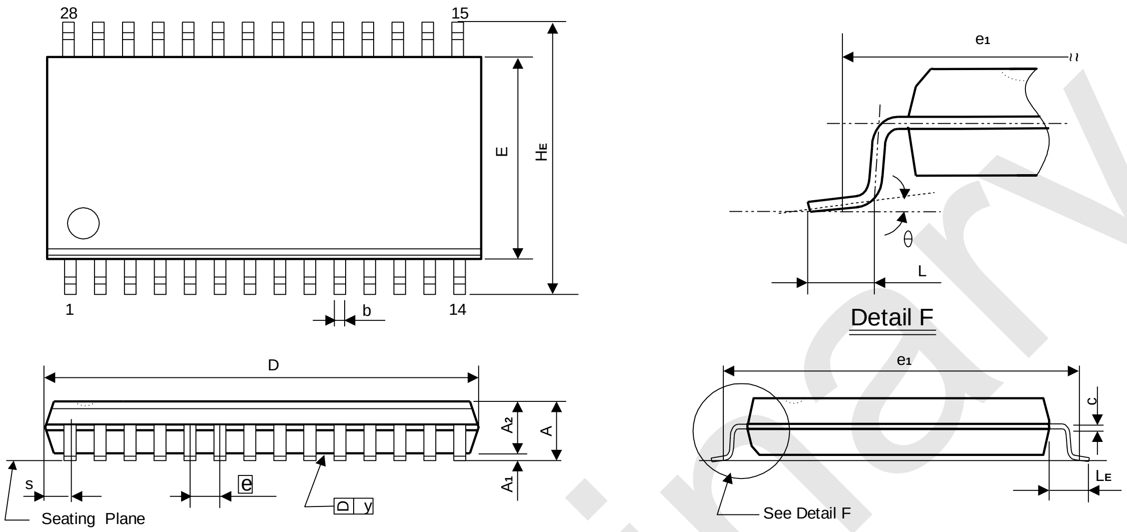
QFN32 (5X5) Dimension Unit: mm



Symbol	mm (milimetre)		
	Min	Normal	Min
A	0.70	0.75	0.80
A1	--	0.02	0.05
b	0.18	0.25	0.30
D	4.90	5.00	5.10
E	4.90	5.00	5.10
e	0.5 BSC		
k	0.4 REF		
D1	3.30	3.45	3.60
E1	3.30	3.45	3.60
L	0.30	0.40	0.50

SC92F8593M28U

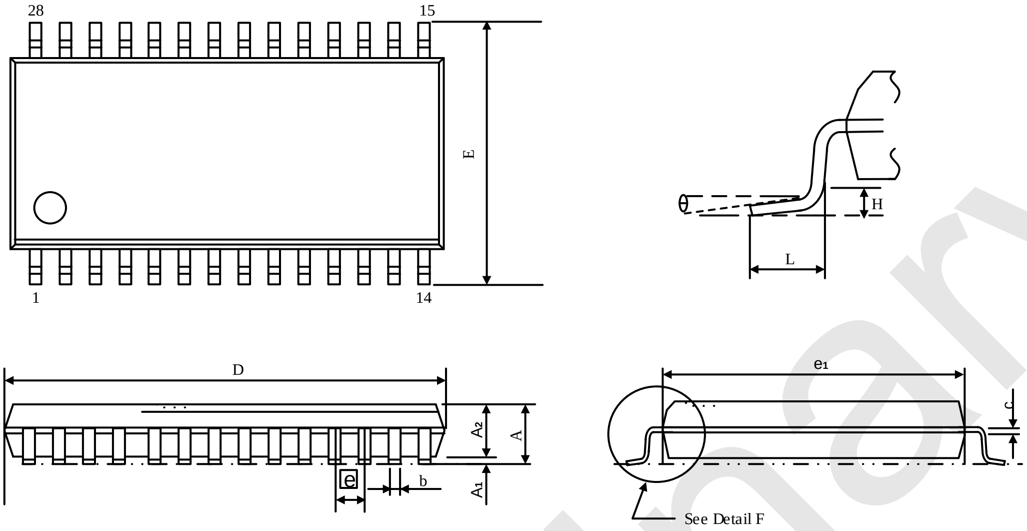
SOP28 Dimension Unit: mm



Symbol	mm (milimetre)		
	Min	Normal	Min
A	2.40	2.56	2.65
A1	0.100	0.200	0.300
A2	2.240	2.340	2.440
b	0.39	---	0.48
C	0.254(BSC)		
D	17.80	18.00	18.20
E	7.30	7.50	7.70
HE	10.100	10.300	10.500
e	1.270(BSC)		
L	0.7	0.85	1.0
LE	1.3	1.4	1.5
θ	0°	-	8°

SC92F8593X28U

TSSOP28 Dimension Unit: mm



Symbol	mm (milimetre)		
	Min	Normal	Min
A	-	-	1.200
A1	0.050	-	0.150
A2	0.800	-	1.050
b	0.190	-	0.300
c	0.090	-	0.200
D	9.600	-	9.800
E	6.250	-	6.550
e1	4.300	-	4.500
e	0.65(BSC)		
L	-	-	1.0
θ	0°	-	8°
H	0.05	-	0.25

26 Revision History

Revision	Changes	Date
V0.3	1. ADC section: Added ADC connection circuit diagram 2. Electrical Characteristics section: Added input leakage current parameters and ADC input leakage current parameters	October 2023
V0.2	1. Modified description of PWM channel: Modified from "8 channel PWM channel" to "8 select 4 PWM channel" 2. Modified strong push-pull output mode circuit diagram in GPIO section 3. Deleted the description about disabling MOVC in the first 256bytes 4. Added the description of LVR Debounce time 5. Optimized the description of ESD and EFT levels	October 2022
V0.1	Initial Release.	August 2022

Important Notice

Shenzhen SinOne Microelectronics Co., Ltd. (hereinafter referred to as SinOne) reserves the right to change, correct, enhance, modify and improve SinOne products, documents or services at any time without prior notice. Cytron considers the information provided to be accurate and reliable. The information in this document will be used in August 2022. In the actual production design, please refer to the latest data manual of each product and other relevant materials.