

---

Shenzhen Guoxin Artificial  
Intelligence Co.

# STC8A8K64D4 Series Microcontrollers Technical Reference Manual

STC MCU

**2022/11/14**

# table of contents

<b>1</b>	<b>Overview .....</b>	<b>1</b>
<b>2</b>	<b>Characteristics, price and pinout .....</b>	<b>2</b>
2.1	STC8A8K64D4-LQFP64/48/44, PDIP40 Series .....	2
2.1.1	Features and price (with 16-bit hardware multiplier-divider MDU16, quasi-16-bit microcontroller) .....	2
2.1.2	Pinout Diagram, Minimum System.....	5
2.1.3	Pin Description.....	9
<b>3</b>	<b>Function foot switching .....</b>	<b>16</b>
3.1	Function pin switching related registers.....	16
3.1.1	Bus Speed Control Register (BUS_SPEED) .....	17
3.1.2	Peripheral Port Switching Control Register 1 (P_SW1), Serial Port 1, CCP, SPI Switching 17	
3.1.3	Peripheral port switching control register 2 (P_SW2), serial port 2/3/4, I2C, comparator output switching.....	17
3.1.4	Clock Select Register (MCLKOCR) .....	18
3.1.5	Enhanced PWM Control Register (PWMnCR) .....	18
3.1.6	LCM Interface Configuration Register (LCMIFCFG) .....	21
3.2	Example procedure .....	22
3.2.1	Serial port 1 Switching.....	22
3.2.2	Serial 2 Switching .....	23
3.2.3	Serial 3 Switching .....	25
3.2.4	Serial 4 Switching .....	26
3.2.5	SPI Switching.....	28
3.2.6	PWM Switching.....	29
3.2.7	PCA/CCP/PWM Switching.....	32
3.2.8	I2C Switching.....	34
3.2.9	Comparator output switching.....	35
3.2.10	Master clock output switching.....	37
<b>4</b>	<b>Package Size Diagram.....</b>	<b>40</b>
4.1	LQFP44 package size diagram (12mm*12mm) .....	40
4.2	LQFP48 package size diagram (9mm*9mm) .....	41
4.3	QFN48 Package Size Diagram (6mm*6mm) .....	42
4.4	LQFP64 package size diagram (12mm*12mm) .....	43
4.5	QFN64 Package Size Diagram (8mm*8mm) .....	44
4.6	STC8A8K64D4 Series Microcontroller Naming Rules .....	45
<b>5</b>	<b>ISP Download and Typical Application Wiring Diagram.....</b>	<b>46</b>
5.1	STC8A8K64D4 Series ISP Download Application Wiring Diagram.....	46
5.1.1	Download using the STC-USB LinkID tool, supporting both online and offline downloads 46	
5.1.2	Software emulation USB direct ISP download, recommended to try, emulation not supported	

STC8A8K64D4 Series		Official website:	Technical Support:	Selection Consultant:
Technical Manual		(5V systems) <a href="http://www.STCAI.com">www.STCAI.com</a>	19864585985	13922805190 48
5.1.3	Software emulation USB direct ISP download, recommended to try, emulation not supported (3.3V systems) .....			50
5.1.4	Download using RS-232 converter, also supports emulation (using high precision ADC) 52			
5.1.5	Download using RS-232 converter, also supports emulation (using general-precision ADC) 53			
5.1.6	Download with PL2303-GL, which also supports emulation.....			54
5.1.7	Download with Universal USB to Serial Tool, supports ISP online download, also supports emulation .....			55
5.1.8	Download with U8-Mini tool, supports ISP online and offline download, also supports emulation .....			56
5.1.9	Download with U8W tool, supports ISP online and offline download, also supports emulation .....			57
5.1.10	Microcontroller Power Control Reference Circuit.....			59
5.2	STC-ISP Download Software Advanced Applications .....			60
5.2.1	Publication of project procedures.....			60
5.2.2	Program encryption and then transfer (to prevent serial port analysis of the program during burn-in).....			64
5.2.3	Publish project program + program encrypted post-transfer used in combination			68
5.2.4	User-defined downloads (enabling non-stop downloads).....			69
5.3	Notes related to driver digital signatures .....			73
5.3.1	About mandatory digital signatures for drivers .....			73
5.3.2	Windows 10 steps to turn off mandatory digital signing for drivers .....			74
<b>6</b>	<b>Clock, Reset, Power Saving Mode and System Power Management .....</b>			<b>79</b>
6.1	System Clock Control .....			79
6.1.1	System Clock Select Register (CLKSEL) .....			79
6.1.2	Clock division register (CLKDIV) .....			80
6.1.3	High-speed internal high-precision IRC control register (HIRCCR) .....			80
6.1.4	External Oscillator Control Register (XOSCCR).....			80
6.1.5	Internal 32KHz Low Speed IRC Control Register (IRC32KCR) .....			81
6.1.6	Master Clock Output Control Register (MCLKOCR) .....			81
6.2	STC8A8K64D4 Series Internal IRC Frequency Adjustment.....			82
6.2.1	IRC Band Select Register (IRCBAND) .....			82
6.2.2	Internal IRC Frequency Adjustment Register (IRTRIM) .....			82
6.2.3	Internal IRC Frequency Trim Register (LIRTRIM) .....			83
6.2.4	Clock division register (CLKDIV) .....			83
6.2.5	Example of dividing out 3MHz user operating frequency and user dynamically changing frequency chasing.....			84
6.3	System Reset.....			87
6.3.1	Watchdog reset (WDT_CONTR) .....			88
6.3.2	Software reset (IAP_CONTR) .....			90
6.3.3	Low Pressure Reset (RSTCFG) .....			91
6.3.4	Low-level power-on reset reference circuit (generally not required).....			92
6.3.5	Low Level Key Manual Reset Reference Circuit.....			92

Technical Manual	Legacy 8051 High Level Power-Up Reset Reference Circuit.....	93
6.4	External Crystal and External Clock Circuit.....	94
6.4.1	External crystal input circuit.....	94
6.4.2	External clock input circuit (P1.6 cannot be used as normal I/O) .....	94
6.5	Clock stop/power saving mode and system power management .....	94
6.5.1	Power Control Register (PCON) .....	94
6.6	Power-down wake-up timer.....	96
6.6.1	Power-down wake-up timer count register (WKTCL, WKTCH) .....	96
6.7	Model procedures.....	97
6.7.1	Selecting the system clock source.....	97
6.7.2	Master clock divider output.....	99
6.7.3	Watchdog Timer Application.....	101
6.7.4	Soft reset for custom downloads.....	103
6.7.5	Low voltage detection.....	105
6.7.6	Power saving mode .....	107
6.7.7	Wake-up power saving mode using INT0/INT1/INT2/INT3/INT4 pin interrupts.....	109
6.7.8	Use T0/T1/T2/T3/T4 pin interrupts to wake up MCU power save mode.....	112
6.7.9	Wake up MCU power saving mode using RxD/RxD2/RxD3/RxD4 pin interrupts .....	117
6.7.10	Wake up MCU power saving mode using SDA pin of I2C .....	120
6.7.11	Wake up power saving mode using the power down wake up timer .....	123
6.7.12	LVD interrupt wake-up power saving mode, recommended to use with power-down wake-up timer .....	125
6.7.13	Comparator interrupt wake-up power saving mode, recommended to use with power-down wake-up timer.....	127
6.7.14	Operating voltage (battery voltage) detected using LVD function.....	130
7	<b>Memory</b> .....	<b>135</b>
7.1	Program memory.....	135
7.2	Data memory.....	136
7.2.1	Internal RAM.....	136
7.2.2	Program Status Register (PSW) .....	136
7.2.3	Internal Expansion RAM, XRAM, XDATA.....	137
7.2.4	Auxiliary Register (AUXR).....	138
7.2.5	External Expansion RAM, XRAM, XDATA.....	139
7.2.6	Bus Speed Control Register (BUS_SPEED) .....	139
7.2.7	Bit-Addressable Data Memory in 8051 .....	141
7.3	Special parameters in memory that can be burned into the program during ISP download FLASH.....	143
7.4	Read-only unique ID numbers and important parameters (CHIPID) stored in special function registers.....	145
7.4.1	CHIP's Global Unique ID Number Explained.....	146
7.4.2	CHIP's Internal Reference Signal Source Interpretation.....	146
7.4.3	CHIP's internal 32K IRC oscillation frequency interpretation.....	147
7.4.4	CHIP's High Precision IRC Parameter Interpretation.....	148
7.4.5	Interpretation of CHIP's test time parameters.....	149

STC8A8K64D4 Series		Official website:	Technical Support:	Selection Consultant:
Technical Manual		<a href="http://www.STC8A8K64D4.com">www.STC8A8K64D4.com</a>	19864585985	13922805190
7.4.0	CHIP's chip package form number interpretation.....			149
7.5	Example procedure.....			150
7.5.1	Read internal 1.19V reference source value (from CHIPID).....			150
7.5.2	Read internal 1.19V reference signal source value (from Flash program memory (ROM))			153
7.5.3	<del>Read internal 1.19V reference source value (from RAM).....</del>			156
7.5.4	Read globally unique ID number (from CHIPID).....			159
7.5.5	Read globally unique ID number (from Flash program memory (ROM)).....			163
7.5.6	<del>Read globally unique ID number (from RAM).....</del>			166
7.5.7	Read 32K power-down wake-up timer frequency (from CHIPID).....			169
7.5.8	Read 32K power-down wake-up timer frequency (from Flash program memory (ROM))			173
7.5.9	<del>Read 32K power-down wake-up timer frequency (from RAM).....</del>			176
7.5.10	User-defined internal IRC frequency (read from CHIPID).....			179
7.5.11	User-defined internal IRC frequency (read from Flash program memory (ROM)).			186
7.5.12	<del>User-defined internal IRC frequency (read from RAM).....</del>			191
8	<b>Special Function Register.....</b>			<b>194</b>
8.1	STC8A8K64D4-64Pin/48Pin Series.....			194
8.2	Special Function Register List.....			196
9	<b>I/O Port.....</b>			<b>206</b>
9.1	I/O port-related registers.....			206
9.1.1	Port Data Register (Px).....			207
9.1.2	Port mode configuration registers (PxM0, PxM1).....			208
9.1.3	Port pull-up resistor control register (PxPU).....			208
9.1.4	Port Schmidt Trigger Control Register (PxNCS).....			209
9.1.5	Port Level Shift Speed Control Register (PxSR).....			209
9.1.6	Port Drive Current Control Register (PxDR).....			209
9.1.7	Port Digital Signal Input Enable Control Register (PxIE).....			210
9.2	Configuring I/O Ports.....			211
9.3	Structure diagram of I/O.....			213
9.3.1	Quasi two-way port (weak pull-up).....			213
9.3.2	Push-Pull Output.....			213
9.3.3	High resistance input.....			214
9.3.4	Open Drain Output.....			214
9.3.5	New 4.1K pull-up resistor.....			215
9.3.6	How to set the I/O port external output speed.....			215
9.3.7	How to set the I/O port current drive capability.....			216
9.3.8	How to Reduce I/O Port External Radiation.....			216
9.4	Model program.....			217
9.4.1	Port Mode Setting.....			217
9.4.2	Bidirectional port read/write operation.....			218
9.5	A Typical Triode Control Circuit.....			221
9.6	Typical Light Emitting Diode Control Circuit.....			221
9.7	Hybrid Voltage Supply System 3V/5V Device I/O Port Interconnect.....			222
9.8	How to make the I/O port go low on power-up reset.....			223
9.9	Wiring diagram for driving 8 digital tubes (serial extension, 3 lines) using 74HC595			224

9.10	I/O port direct drive LED digital tube application wiring diagram.....	225
9.11	Directly drive segment LCD with I/O port of STC series MCUs.....	226
<b>10</b>	<b>Command system.....</b>	<b>245</b>
<b>11</b>	<b>Interrupt systems.....</b>	<b>249</b>
11.1	STC8A8K64D4 Series Interrupt Sources .....	249
11.2	STC8A8K64D4 Interrupt Structure Diagram.....	252
11.3	STC8A8K64D4 Series Interrupt List.....	253
11.4	Interrupt-related registers.....	256
11.4.1	Interrupt enable register (interrupt allow bit) .....	258
11.4.2	Interrupt request register (interrupt flag bit) .....	263
11.4.3	Interrupt Priority Register .....	267
11.5	Model procedures.....	273
11.5.1	INT0 interrupt (rising and falling edge), can support both rising and falling edge	273
11.5.2	INT0 Interrupt (falling edge) .....	275
11.5.3	INT1 interrupt (rising and falling edge), can support both rising and falling edge	276
11.5.4	INT1 Interrupt (falling edge) .....	278
11.5.5	INT2 interrupt (falling edge), falling edge interrupt only supported.....	280
11.5.6	INT3 interrupt (falling edge), falling edge interrupt only supported.....	282
11.5.7	INT4 interrupt (falling edge), falling edge interrupt only supported.....	284
11.5.8	Timer 0 Interrupt .....	286
11.5.9	Timer 1 Interrupt .....	287
11.5.10	Timer 2 Interrupt .....	289
11.5.11	Timer 3 Interrupt .....	291
11.5.12	Timer 4 Interrupt .....	294
11.5.13	UART1 Interrupt.....	296
11.5.14	UART2 Interrupt.....	298
11.5.15	UART3 Interrupt.....	301
11.5.16	UART4 Interrupt.....	303
11.5.17	LVD Interrupt .....	306
11.5.18	SPI Interrupt.....	308
11.5.19	Comparator Interrupt .....	310
11.5.20	I2C Interrupt .....	312
<b>12</b>	<b>Normal I/O ports are interruptible, not traditional external interrupts .....</b>	<b>316</b>
12.1	I/O port interrupt-related registers .....	316
12.1.1	Port Interrupt Enable Register (PxINTE) .....	317
12.1.2	Port Interrupt Flag Register (PxINTF) .....	317
12.1.3	Port Interrupt Mode Configuration Registers (PxIM0, PxIM1) .....	318
12.1.4	Port Interrupt Priority Control Registers (PINIPL, PINIPH) .....	318
12.1.5	Port Interrupt Power-down Wakeup Enable Register (PxWKUE) .....	318
12.2	Example program.....	320
12.2.1	P0 port falling edge interrupt.....	320
12.2.2	P1 port rising edge interrupt .....	324
12.2.3	P2 port low interrupt.....	327
12.2.4	P3 port high interrupt .....	331

<b>13</b>	<b>Timer/Counter</b>	<b>336</b>
13.1	Timer Related Registers	336
13.2	Timer 0/1	338
13.2.1	Timer 0/1 Control Register (TCON)	338
13.2.2	Timer 0/1 Mode Register (TMOD)	338
13.2.3	Timer 0 Mode 0 (16-bit Auto Reload Mode)	339
13.2.4	Timer 0 Mode 1 (16-bit non-reloadable mode)	340
13.2.5	Timer 0 Mode 2 (8-bit Auto Reload Mode)	341
13.2.6	Timer 0 Mode 3 (Non-Maskable Interrupt 16-bit Auto Reload, Real-Time OS Metronome)	341
13.2.7	Timer 1 Mode 0 (16-bit Auto Reload Mode)	342
13.2.8	Timer 1 Mode 1 (16-bit non-reloadable mode)	343
13.2.9	Timer 1 Mode 2 (8-bit Auto Reload Mode)	344
13.2.10	Timer 0 Counting register (TL0, TH0)	344
13.2.11	Timer 1 Counting Register (TL1, TH1)	344
13.2.12	Auxiliary Register 1 (AUXR)	344
13.2.13	Interrupt and Clock Output Control Register (INTCLKO)	345
13.2.14	Timer 0 Timing formula	345
13.2.15	Timer 1 Timing formula	346
13.3	Timer 2 (24-bit timer, 8-bit prescaler + 16-bit timing)	347
13.3.1	Auxiliary Register 1 (AUXR)	347
13.3.2	Interrupt and Clock Output Control Register (INTCLKO)	347
13.3.3	Timer 2 Counting Register (T2L, T2H)	347
13.3.4	Timer 2's 8-bit Prescaler Register (TM2PS)	347
13.3.5	Timer 2 operating mode	348
13.3.6	Timer 2 Calculation formula	348
13.4	Timer 3/4 (24-bit Timer, 8-bit Prescaler + 16-bit Timing)	349
13.4.1	Timer 4/3 Control Register (T4T3M)	349
13.4.2	Timer 3 Counting Register (T3L, T3H)	349
13.4.3	Timer 4 Count Register (T4L, T4H)	349
13.4.4	Timer 3's 8-bit Prescaler Register (TM3PS)	350
13.4.5	Timer 4's 8-bit Prescaler Register (TM4PS)	350
13.4.6	Timer 3 operating mode	350
13.4.7	Timer 4 Operating Modes	352
13.4.8	Timer 3 Calculation formula	352
13.4.9	Timer 4 Calculation formula	353
13.5	Model procedures	354
13.5.1	Timer 0 (mode 0-16 bit auto reload), used as timing	354
13.5.2	Timer 0 (mode 1- 16 bits not automatically reloaded), used as timing	355
13.5.3	Timer 0 (mode 2-8 bit auto reload), used as timing	357
13.5.4	Timer 0 (mode 3-16 bit auto reload non-maskable interrupt), used as timing	359
13.5.5	Timer 0 (external count - extended T0 for external falling edge interrupt)	361
13.5.6	Timer 0 (measuring pulse width - INTO high level width)	363
13.5.7	Timer 0 (Mode 0), Clock Divider Output	365
13.5.8	Timer 1 (mode 0-16 bit auto reload), used as timing	367



STC8A8K64D4 Series	Official website:	Technical Support:	Selection Consultant:
Technical Manual	<a href="http://www.STC8.com">www.STC8.com</a>	19864585085	13922805190
13.5.9	Timer 1 (mode 1 16-bit auto reload), used as timing.....	369	
13.5.10	Timer 1 (mode 2-8 bit auto reload), used as timing.....	371	
13.5.11	Timer 1 (external count - extended T1 for external falling edge interrupt) .	372	
13.5.12	Timer 1 (measuring pulse width - INT1 high level width).....	374	
13.5.13	Timer 1 (Mode 0), Clock Divider Output.....	376	
13.5.14	Timer 1 (mode 0) does serial port 1 baud rate generator.....	378	
13.5.15	Timer 1 (mode 2) as serial port 1 baud rate generator.....	382	
13.5.16	Timer 2 (16-bit auto reload), used for timing.....	386	
13.5.17	Timer 2 (external count - extended T2 for external falling edge interrupt) .	388	
13.5.18	Timer 2, Clock Divider Output.....	390	
13.5.19	Timer 2 do serial port 1 baud rate generator.....	392	
13.5.20	Timer 2 do serial port 2 baud rate generator.....	396	
13.5.21	Timer 2 Make serial port 3 Baud rate generator.....	400	
13.5.22	Timer 2 do serial port 4 baud rate generator.....	404	
13.5.23	Timer 3 (16-bit auto reload), used for timing.....	408	
13.5.24	Timer 3 (external count - extended T3 for external falling edge interrupt) .	410	
13.5.25	Timer 3, Clock Divider Output.....	412	
13.5.26	Timer 3 do serial port 3 baud rate generator.....	414	
13.5.27	Timer 4 (16-bit auto reload), used as timing.....	418	
13.5.28	Timer 4 (external count - extended T4 for external falling edge interrupt) .	421	
13.5.29	Timer 4, Clock Divider Output.....	423	
13.5.30	Timer 4 do serial 4 baud rate generator.....	425	
<b>14</b>	<b>Serial communication .....</b>	<b>430</b>	
14.1	Serial port function pin switching.....	430	
14.2	Serial Port Related Registers.....	430	
14.3	Serial port 1.....	432	
14.3.1	Serial 1 Control Register (SCON) .....	432	
14.3.2	Serial 1 Data Register (SBUF) .....	432	
14.3.3	Power Management Register (PCON) .....	433	
14.3.4	Auxiliary Register 1 (AUXR) .....	433	
14.3.5	Serial port 1 Mode 0, Mode 0 Baud rate calculation formula.....	433	
14.3.6	Serial 1 Mode 1, Mode 1 Baud rate calculation formula.....	434	
14.3.7	Serial 1 Mode 2, Mode 2 Baud Rate Calculation Formula.....	437	
14.3.8	Serial 1 Mode 3, Mode 3 Baud Rate Calculation Formula.....	437	
14.3.9	Automatic address recognition.....	438	
14.3.10	Serial 1 Slave Address Control Register (SADDR, SADEN) .....	438	
14.4	Serial port 2.....	440	
14.4.1	Serial port 2 control register (S2CON).....	440	
14.4.2	Serial 2 Data Register (S2BUF) .....	440	
14.4.3	Serial 2 Mode 0, Mode 0 Baud rate calculation formula.....	440	
14.4.4	Serial Port 2 Mode 1, Mode 1 Baud Rate Calculation Formula.....	441	
14.5	Serial port 3.....	443	
14.5.1	Serial port 3 control register (S3CON).....	443	
14.5.2	Serial 3 Data Register (S3BUF) .....	443	

STC8A8K64D4 Series		Official website:	Technical Support:	Selection Consultant:
Technical Manual		<a href="http://www.STC8A.com">www.STC8A.com</a>	19864585985	13922805190
14.5.3	Serial 3 Mode 0, Mode 0 Baud Rate Calculation Formula.....			443
14.5.4	Serial Port 3 Mode 1, Mode 1 Baud Rate Calculation Formula.....			444
14.6	Serial port 4.....			446
14.6.1	Serial Port 4 Control Register (S4CON).....			446
14.6.2	Serial 4 Data Register (S4BUF) .....			446
14.6.3	Serial Port 4 Mode 0, Mode 0 Baud Rate Calculation Formula.....			446
14.6.4	Serial Port 4 Mode 1, Mode 1 Baud Rate Calculation Formula.....			447
14.7	Serial Port Notes.....			449
14.8	Model program.....			450
14.8.1	Serial port 1 Use timer 2 as baud rate generator.....			450
14.8.2	Serial Port 1 Use Timer 1 (Mode 0) as a baud rate generator .....			453
14.8.3	Serial Port 1 Use Timer 1 (Mode 2) as a baud rate generator .....			457
14.8.4	Serial Port 2 Using Timer 2 as a Baud Rate Generator.....			461
14.8.5	Serial Port 3 Using Timer 2 as a Baud Rate Generator.....			465
14.8.6	Serial Port 3 Using Timer 3 as a Baud Rate Generator.....			469
14.8.7	Serial Port 4 Using Timer 2 as a Baud Rate Generator.....			473
14.8.8	Serial Port 4 Using Timer 4 as a Baud Rate Generator.....			477
14.8.9	Serial Multi-Machine Communication.....			482
14.8.10	Serial to LIN Bus.....			483
15	<b>Comparator, Power Failure Detection, Internal 1.19V Reference Signal Source..</b>			<b>492</b>
15.1	Comparator output function pin switching.....			492
15.2	Comparator Internal Structure Diagram.....			492
15.3	Comparator-related registers.....			493
15.3.1	Comparator Control Register 1 (CMPCR1).....			493
15.3.2	Comparator Control Register 2 (CMPCR2).....			494
15.3.3	Comparator Extended Configuration Register (CMPEXCFG) .....			494
15.4	Model program.....			496
15.4.1	Use of Comparators (Interrupt Method) .....			496
15.4.2	Use of Comparators (Query Approach) .....			499
15.4.3	Comparator for external power-down detection (user data should be saved to EEPROM in time during power-down).....			502
15.4.4	Comparator detects operating voltage (battery voltage).....			503
16	<b>IAP/EEPROM/DATA-FLASH .....</b>			<b>508</b>
16.1	EEPROM operation time.....			508
16.2	EEPROM-related registers.....			508
16.2.1	EEPROM Data Register (IAP_DATA).....			508
16.2.2	EEPROM Address Register (IAP_ADDR) .....			509
16.2.3	EEPROM Command Register (IAP_CMD).....			509
16.2.4	EEPROM Trigger Register (IAP_TRIG) .....			509
16.2.5	EEPROM control register (IAP_CONTR) .....			509
16.2.6	EEPROM Wait Time Control Register (IAP_TPS) .....			510
16.3	EEPROM size and address.....			511
16.4	Model program.....			513
16.4.1	EEPROM Basic Operation.....			513

Technical Manual	Reading the EEPROM using MOVX.....	516
16.4.2	Reading the EEPROM using MOVX.....	516
16.4.3	Sending EEPROM data using the serial port.....	520
16.4.4	Serial Port 1 Read/Write EEPROM-with MOVX Read.....	524
16.4.5	Password Erase Write - Multi-Sector Backup - Serial 1 Operation.....	531
<b>17</b>	<b>ADC Analog to Digital, Internal 1.19V Reference Signal Source.....</b>	<b>540</b>
17.1	ADC-related registers.....	540
17.1.1	ADC Control Register (ADC_CONTR), PWM Triggered ADC Control.....	540
17.1.2	ADC Configuration Register (ADCCFG).....	541
17.1.3	ADC conversion result registers (ADC_RES, ADC_RES1).....	542
17.1.4	ADC Timing Control Register (ADCTIM).....	543
17.1.5	ADC Extended Configuration Register (ADCEXCFG).....	544
17.2	ADC Related Calculation Formula.....	545
17.2.1	ADC Speed Formula.....	545
17.2.2	ADC conversion result calculation formula.....	545
17.2.3	Inverse ADC Input Voltage Formula.....	545
17.2.4	Backpropagation of the operating voltage calculation formula.....	546
17.3	12-bit ADC Static Characteristics.....	546
17.4	ADC Application Reference Wiring Diagram.....	547
17.4.1	High Accuracy ADC Applications.....	547
17.4.2	ADC General Applications (applications that do not require high ADC accuracy).....	548
17.5	Paradigm procedures.....	549
17.5.1	ADC Basic Operations (Query Method).....	549
17.5.2	ADC Basic Operation (Interrupt Method).....	551
17.5.3	Formatting ADC Conversion Results.....	553
17.5.4	ADC auto-conversion multiple averaging.....	556
17.5.5	Measuring External or Battery Voltages with ADC Channel 15.....	558
17.5.6	ADC for capacitive-sensitive touch keys.....	562
17.5.7	ADC for Key Scan Application Wiring Diagram.....	574
17.5.8	Detecting Negative Voltage Reference Line Diagram.....	576
17.5.9	Application of Common Addition Circuits in ADCs.....	577
<b>18</b>	<b>PCA/CCP/PWM Applications.....</b>	<b>578</b>
18.1	PCA function pin switching.....	578
18.2	PCA-related registers.....	578
18.2.1	PCA Control Register (CCON).....	579
18.2.2	PCA Mode Register (CMOD).....	579
18.2.3	PCA counter register (CL, CH).....	580
18.2.4	PCA Module Mode Control Register (CCAPMn).....	580
18.2.5	PCA Module Mode Capture Value/Compare Value Registers (CCAPnL, CCAPnH).....	580
18.2.6	PCA Module PWM Mode Control Register (PCA_PWMn).....	581
18.3	PCA operating mode.....	582
18.3.1	Capture Mode.....	582
18.3.2	Software Timer Mode.....	582
18.3.3	High-speed pulse output mode.....	583
18.3.4	PWM Pulse Width Modulation Mode and Frequency Calculation Formula.....	583

STC8A8K64D4 Series	Official website:	Technical Support:	Selection Consultant:
Technical Manual	<a href="http://www.STC8A8K64D4.com">www.STC8A8K64D4.com</a>	19864585985	13912805190
18.4	Reference Circuit Diagram for 8 to 16-Bit DACs Using CCP/PCA/PWM Modules ...	587	
18.5	Model program.....	588	
18.5.1	PCA Output PWM (6/7/8/10 bits) .....	588	
18.5.2	PCA capture measurement pulse width.....	591	
18.5.3	PCA Implementation of 16-bit Software Timing.....	595	
18.5.4	PCA Output High Speed Pulse.....	598	
18.5.5	PCA Extended External Interrupt.....	601	
<b>19</b>	<b>Enhanced PWM with up to 15-bit accuracy .....</b>	<b>604</b>	
19.1	Enhanced PWM Output Function Pin Switching .....	604	
19.2	PWM-related registers.....	606	
19.2.1	Enhanced PWM Global Configuration Register (PWMSET) .....	607	
19.2.2	Enhanced PWM Configuration Register (PWMCFG) .....	607	
19.2.3	PWM Interrupt Flag Register (PWMIF) .....	608	
19.2.4	PWM Exception Detection Control Register (PWMnFDCR) .....	608	
19.2.5	PWM Counter Register (PWMCH, PWMCL) .....	609	
19.2.6	PWM Clock Select Register (PWMCKS), Output Frequency Calculation Formula .....	609	
19.2.7	PWM Triggered ADC Counter Register (PWMTADC) .....	610	
19.2.8	PWM level output setting count register (PWMnT1, PWMnT2) .....	610	
19.2.9	PWM Channel Control Register (PWMnCR) .....	611	
19.2.10	PWM Channel Level Hold Control Register (PWMnHLD) .....	612	
19.3	Example procedure .....	613	
19.3.1	Output arbitrary period and arbitrary duty cycle waveforms .....	613	
19.3.2	Waveforms for Complementary Symmetric Deadband Control with Two PWMs.....	615	
19.3.3	PWM implementation of fading light (breathing light) .....	619	
19.3.4	Using PWM to Trigger ADC Conversions .....	623	
<b>20</b>	<b>Synchronous Serial Peripheral Interface SPI .....</b>	<b>628</b>	
20.1	SPI Function Pin Switching.....	628	
20.2	SPI-related registers.....	628	
20.2.1	SPI Status Register (SPSTAT) .....	628	
20.2.2	SPI Control Register (SPCTL), SPI Speed Control .....	628	
20.2.3	SPI Data Register (SPDAT) .....	629	
20.3	SPI communication method.....	630	
20.3.1	Single Master Single Slave.....	630	
20.3.2	Mutually dominated from.....	630	
20.3.3	Single Master Multi-Slave.....	631	
20.4	Configuring SPI.....	632	
20.5	Data Mode .....	634	
20.6	Example procedure .....	635	
20.6.1	SPI Single Master Single Slave System Host Program (Interrupt Method) .....	635	
20.6.2	SPI Single Master Single Slave System Slave Program (Interrupt Method) .....	637	
20.6.3	SPI Single Master Single Slave System Host Program (Query Method) .....	639	
20.6.4	SPI Single Master Single Slave System Slave Program (Query Method) .....	641	
20.6.5	SPI Mutual Master-Slave System Program (Interrupt Method) .....	644	
20.6.6	SPI Mutual Master-Slave System Program (Query Method) .....	646	

STC8A8K64D4 Series		Official website:	Technical Support:	Selection Consultant:
Technical Manual		<a href="http://www.STCAI.com">www.STCAI.com</a>	19864585985	13922805190
<b>21</b>	<b>I<sup>2</sup>C bus</b>			<b>650</b>
21.1	I2C function pin switching			650
21.2	I <sup>2</sup> C Related registers			650
21.3	I <sup>2</sup> C Host Mode			651
21.3.1	I2C Configuration Register (I2CCFG), Bus Speed Control			651
21.3.2	I2C Host Control Register (I2CMSCR)			652
21.3.3	I2C Host Auxiliary Control Register (I2CMSAUX)			653
21.3.4	I2C Host Status Register (I2CMSST)			653
21.4	I <sup>2</sup> C Slave Mode			655
21.4.1	I2C Slave Control Register (I2CSLCR)			655
21.4.2	I2C Slave Status Register (I2CSLST)			655
21.4.3	I2C Slave Address Register (I2CSLADR)			657
21.4.4	I2C data registers (I2CTXD, I2CRXD)			658
21.5	Example procedure			659
21.5.1	I <sup>2</sup> C Host Mode Access AT24C256 (Interrupt Mode)			659
21.5.2	I <sup>2</sup> C Host Mode Access AT24C256 (Query Mode)			665
21.5.3	I <sup>2</sup> C Host Mode Access PCF8563			671
21.5.4	I <sup>2</sup> C Slave mode (interrupt mode)			676
21.5.5	I <sup>2</sup> C Slave mode (query mode)			681
21.5.6	Test I <sup>2</sup> C Slave Mode Code for Master Code			685
<b>22</b>	<b>LCM Interface</b>			<b>692</b>
22.1	LCM Interface Function Pin Switching			692
22.2	LCM Related Registers			692
22.2.1	LCM Interface Configuration Register (LCMIFCFG)			693
22.2.2	LCM Interface Configuration Register 2 (LCMIFCFG2)			693
22.2.3	LCM Interface Control Register (LCMIFCR)			694
22.2.4	LCM Interface Status Register (LCMIFSTA)			694
22.2.5	LCM Interface Data Register (LCMIFDATL, LCMIFDATH)			694
22.3	LCD Interface Timing Diagram			695
22.3.1	I8080 mode			695
22.3.2	M6800 Mode			696
<b>23</b>	<b>DMA</b>			<b>697</b>
23.1	DMA-related registers			697
23.2	Memory-to-memory data read/write (M2M_DMA)			699
23.2.1	M2M_DMA Configuration Register (DMA_M2M_CFG)			699
23.2.2	M2M_DMA Control Register (DMA_M2M_CR)			700
23.2.3	M2M_DMA Status Register (DMA_M2M_STA)			700
23.2.4	M2M_DMA Transfer Total Bytes Register (DMA_M2M_AMT)			700
23.2.5	M2M_DMA Transfer Completion Byte Register (DMA_M2M_DONE)			701
23.2.6	M2M_DMA Send Address Register (DMA_M2M_TXAx)			701
23.2.7	M2M_DMA Receive Address Register (DMA_M2M_RXAx)			701
23.3	ADC Data Automatic Storage (ADC_DMA)			702
23.3.1	ADC_DMA Configuration Register (DMA_ADC_CFG)			702
23.3.2	ADC_DMA Control Register (DMA_ADC_CR)			702

23.3.3	ADC_DMA Status Register (DMA_ADC_STA) .....	702
23.3.4	ADC_DMA Receive Address Register (DMA_ADC_RXAx) .....	702
23.3.5	ADC_DMA Configuration Register 2 (DMA_ADC_CFG2) .....	703
23.3.6	ADC_DMA Channel Enable Register (DMA_ADC_CHSWx) .....	703
23.3.7	Data storage format of ADC_DMA.....	704
23.4	Data exchange between SPI and memory (SPI_DMA) .....	706
23.4.1	SPI_DMA Configuration Register (DMA_SPI_CFG) .....	706
23.4.2	SPI_DMA Control Register (DMA_SPI_CR) .....	706
23.4.3	SPI_DMA Status Register (DMA_SPI_STA) .....	707
23.4.4	SPI_DMA Transfer Total Byte Register (DMA_SPI_AMT) .....	707
23.4.5	SPI_DMA Transfer Completion Byte Register (DMA_SPI_DONE) .....	707
23.4.6	SPI_DMA Send Address Register (DMA_SPI_TXAx) .....	707
23.4.7	SPI_DMA Receive Address Register (DMA_SPI_RXAx) .....	707
23.4.8	SPI_DMA Configuration Host 2 Device (DMA_SPI_CFG2) .....	708
23.5	Data exchange between serial port 1 and memory (UR1T_DMA, UR1R_DMA) .....	709
23.5.1	UR1T_DMA Configuration Register (DMA_UR1T_CFG) .....	709
23.5.2	UR1T_DMA control register (DMA_UR1T_CR) .....	709
23.5.3	UR1T_DMA Status Register (DMA_UR1T_STA) .....	709
23.5.4	UR1T_DMA Transfer Total Bytes Register (DMA_UR1T_AMT) .....	710
23.5.5	UR1T_DMA Transfer Completion Byte Register (DMA_UR1T_DONE) .....	710
23.5.6	UR1T_DMA Send Address Register (DMA_UR1T_TXAx) .....	710
23.5.7	UR1R_DMA Configuration Register (DMA_UR1R_CFG) .....	710
23.5.8	UR1R_DMA Control Register (DMA_UR1R_CR) .....	710
23.5.9	UR1R_DMA Status Register (DMA_UR1R_STA) .....	711
23.5.10	UR1R_DMA Transfer Total Bytes Register (DMA_UR1R_AMT) .....	711
23.5.11	UR1R_DMA Transfer Completion Byte Register (DMA_UR1R_DONE) .....	711
23.5.12	UR1R_DMA Receive Address Register (DMA_UR1T_RXAx) .....	711
23.6	Data exchange between serial port 2 and memory (UR2T_DMA, UR2R_DMA) .....	712
23.6.1	UR2T_DMA Configuration Register (DMA_UR2T_CFG) .....	712
23.6.2	UR2T_DMA control register (DMA_UR2T_CR) .....	712
23.6.3	UR2T_DMA Status Register (DMA_UR2T_STA) .....	712
23.6.4	UR2T_DMA Transfer Total Bytes Register (DMA_UR2T_AMT) .....	713
23.6.5	UR2T_DMA Transfer Completion Byte Register (DMA_UR2T_DONE) .....	713
23.6.6	UR2T_DMA Send Address Register (DMA_UR2T_TXAx) .....	713
23.6.7	UR2R_DMA Configuration Register (DMA_UR2R_CFG) .....	713
23.6.8	UR2R_DMA Control Register (DMA_UR2R_CR) .....	713
23.6.9	UR2R_DMA Status Register (DMA_UR2R_STA) .....	714
23.6.10	UR2R_DMA Transfer Total Bytes Register (DMA_UR2R_AMT) .....	714
23.6.11	UR2R_DMA Transfer Completion Byte Register (DMA_UR2R_DONE) .....	714
23.6.12	UR2R_DMA Receive Address Register (DMA_UR2T_RXAx) .....	714
23.7	Data exchange between serial port 3 and memory (UR3T_DMA, UR3R_DMA) .....	715
23.7.1	UR3T_DMA Configuration Register (DMA_UR3T_CFG) .....	715
23.7.2	UR3T_DMA control register (DMA_UR3T_CR) .....	715
23.7.3	UR3T_DMA Status Register (DMA_UR3T_STA) .....	715



Technical Manual	UR3T_DMA Transfer Total Bytes Register (DMA_UR3T_AMT) .....	716
23.7.4	UR3T_DMA Transfer Completion Byte Register (DMA_UR3T_DONE) .....	716
23.7.5	UR3T_DMA Send Address Register (DMA_UR3T_TXAx) .....	716
23.7.6	UR3R_DMA Configuration Register (DMA_UR3R_CFG) .....	716
23.7.7	UR3R_DMA Control Register (DMA_UR3R_CR) .....	716
23.7.8	UR3R_DMA Status Register (DMA_UR3R_STA) .....	717
23.7.9	UR3R_DMA Transfer Total Byte Register (DMA_UR3R_AMT) .....	717
23.7.10	UR3R_DMA Transfer Completion Byte Register (DMA_UR3R_DONE) .....	717
23.7.11	UR3R_DMA Receive Address Register (DMA_UR3T_RXAx) .....	717
23.7.12	UR3R_DMA Receive Address Register (DMA_UR3T_RXAx) .....	717
23.8	Data exchange between serial port 4 and memory (UR4T_DMA, UR4R_DMA) .....	718
23.8.1	UR4T_DMA Configuration Register (DMA_UR4T_CFG) .....	718
23.8.2	UR4T_DMA Control Register (DMA_UR4T_CR) .....	718
23.8.3	UR4T_DMA Status Register (DMA_UR4T_STA) .....	718
23.8.4	UR4T_DMA Transfer Total Bytes Register (DMA_UR4T_AMT) .....	719
23.8.5	UR4T_DMA Transfer Completion Byte Register (DMA_UR4T_DONE) .....	719
23.8.6	UR4T_DMA Send Address Register (DMA_UR4T_TXAx) .....	719
23.8.7	UR4R_DMA Configuration Register (DMA_UR4R_CFG) .....	719
23.8.8	UR4R_DMA Control Register (DMA_UR4R_CR) .....	719
23.8.9	UR4R_DMA Status Register (DMA_UR4R_STA) .....	720
23.8.10	UR4R_DMA Transfer Total Bytes Register (DMA_UR4R_AMT) .....	720
23.8.11	UR4R_DMA Transfer Completion Byte Register (DMA_UR4R_DONE) .....	720
23.8.12	UR4R_DMA Receive Address Register (DMA_UR4T_RXAx) .....	720
23.9	Data read and write between LCM and memory (LCM_DMA) .....	721
23.9.1	LCM_DMA Configuration Register (DMA_LCM_CFG) .....	721
23.9.2	LCM_DMA Control Register (DMA_LCM_CR) .....	721
23.9.3	LCM_DMA Status Register (DMA_LCM_STA) .....	722
23.9.4	LCM_DMA Transfer Total Bytes Register (DMA_LCM_AMT) .....	722
23.9.5	LCM_DMA Transfer Completion Byte Register (DMA_LCM_DONE) .....	722
23.9.6	LCM_DMA Send Address Register (DMA_LCM_TXAx) .....	722
23.9.7	LCM_DMA Receive Address Register (DMA_LCM_RXAx) .....	722
23.10	Example procedure .....	723
23.10.1	Serial 1 Interrupt Mode with PC Send and Receive Test - DMA Receive Timeout Interrupt .....	723
23.10.2	Serial 1 Interrupt Mode and PC Send and Receive Test - DMA Data Checksum .....	728
<b>24</b>	<b>Enhanced Double Data Pointer .....</b>	<b>735</b>
24.1	Related special function registers .....	735
24.1.1	Group 1 16-bit Data Pointer Register (DPTR0) .....	735
24.1.2	Group 2 16-bit Data Pointer Register (DPTR1) .....	735
24.1.3	Data Pointer Control Register (DPS) .....	735
24.1.4	Data Pointer Control Register (TA) .....	736
24.2	Example procedure .....	738
24.2.1	Example code 1 .....	738
24.2.2	Example code 2 .....	739
<b>25</b>	<b>MDU16 Hardware 16-Bit Multiplier-Divider .....</b>	<b>741</b>

STC8A8K64D4 Series	Official website:	Technical Support:	Selection Consultant:
Technical Manual	<a href="http://www.STCAL.com">www.STCAL.com</a>	19864585985	13922805190
25.1	Related special function registers.....		741
25.1.1	Operand 1 Data register (MD0 to MD3) .....		741
25.1.2	Operand 2 Data register (MD4 to MD5) .....		741
25.1.3	MDU mode control register (ARCON), number of clocks required for operation .....		742
25.1.4	MDU Operation Control Register (OPCON) .....		742
25.2	About MDU16's web application miscellany (provide ideas, for reference only) .....		744
25.3	Example procedure .....		746
<b>Appendix A</b>	<b>Compiler (assembler)/emulator/header file usage guidelines .....</b>		<b>748</b>
<b>Appendix</b>	<b>BSTC-ISP Download Software Advanced Applications .....</b>		<b>757</b>
B.1	Post project procedures.....		757
B.2	Program encryption and then transfer (to prevent serial port analysis of the program during burn-in).....		761
B.3	Post project program + program encrypted post-transfer used in combination..		765
B.4	User-defined downloads (enables non-stop downloads) .....		766
<b>Appendix C</b>	<b>How to Test I/O Ports.....</b>		<b>770</b>
<b>Appendix D</b>	<b>How to Make a Legacy 8051 Microcontroller Learning Board Emulatable.....</b>		<b>771</b>
<b>Appendix</b>	<b>ESTC-USB Driver Installation Instructions.....</b>		<b>773</b>
<b>Appendix</b>	<b>FUSB Download Step-by-Step Demonstration.....</b>		<b>836</b>
<b>Appendix</b>	<b>GRS485 Automatic Control or I/O Port Control Circuit Diagram.....</b>		<b>840</b>
<b>Appendix</b>	<b>HSTC Tool User Manual.....</b>		<b>841</b>
H.1	Overview.....		841
H.2	System Programmable (ISP) Process Description.....		841
H.3	USB type online/offline download tool U8W/U8W-Mini.....		842
H.3.1	Installing the U8W/U8W-Mini Driver .....		844
H.3.2	Features of U8W.....		847
H.3.3	Online download instructions for U8W.....		848
H.3.4	Instructions for offline download of U8W .....		851
H.3.5	Features of the U8W-Mini.....		859
H.3.6	Online download instructions for the U8W-Mini .....		860
H.3.7	Offline Download Instructions for the U8W-Mini.....		861
H.3.8	Production/Update U8W/U8W-Mini .....		867
H.3.9	U8W/U8W-Mini Setting Pass-Through Mode (available for emulation) .....		869
H.3.10	Reference Circuit for the U8W/U8W-Mini.....		869
H.4	STC Universal USB to Serial Port Tool.....		871
H.4.1	STC Universal USB to Serial Tool Appearance Diagram.....		871
H.4.2	STC Universal USB to Serial Tool Layout Diagram.....		872
H.4.3	STC Universal USB to Serial Tool Driver Installation.....		873
H.4.4	Downloading Programs to MCUs Using the STC Universal USB to Serial Tool ...		874
H.4.5	Emulating User Code with the STC Universal USB to Serial Tool .....		876
H.5	Application wiring diagram.....		883
H.5.1	U8W Tool Application Reference Wiring Diagram.....		883
H.5.2	STC Universal USB to Serial Tool Application Reference Wiring Diagram.....		883
<b>Appendix</b>	<b>ISTC Simulation User Manual.....</b>		<b>885</b>
I.1	Overview.....		885



STC8A8K64D4 Series		Official website:	Technical Support:	Selection Consultant:
Technical Manual		<a href="http://www.stc8a.com">www.stc8a.com</a>	19864585985	13922805190
I.2	Installing Keil Software.....			886
I.3	Installing the Emulation Driver.....			887
I.4	Serial Direct Emulation.....			890
I.4.1	Making a Serial Emulation Chip.....			890
I.4.2	Serial Emulation Setup in Keil Software .....			893
I.4.3	Simulation using the serial port in Keil software.....			895
I.5	USB direct emulation (currently only supported by the STC8H8K64U-B version of the chip).....			897
I.5.1	Making USB Emulation Chips.....			897
I.5.2	USB Emulation Setup in Keil Software.....			901
I.5.3	Emulating with USB in Keil Software .....			903
Appendix RS485 Partial Wiring Diagram in .....		JU8W Download Tool		905
Appendix K Automatic start of ISP download upon receipt of user command when running a user program (without power).....				906
Appendix L Developing Your Own ISP Programs Using STC's IAP Series Microcontrollers..				908
Appendix M Method for resetting user programs to the system area for ISP download (without power)				920
Appendix N Example ISP Download Program for STC8A8K64D4 Series Microcontrollers Using a Third-Party MCU				926
Appendix O Using a Third-Party Application to Call the STC Release Project Program for an ISP Download to a Microcontroller .....				934
Appendix P Methods for Creating Multi-File Projects in Keil.....				938
Appendix Q Handling of compilation errors in Keil for interrupt numbers greater than 31 ....				942
Appendix R Electrical Characteristics .....				952
R.1	Absolute Maximum Rating.....			952
R.2	DC Characteristics (3.3V) .....			953
R.3	DC Characteristics (5.0V) .....			955
R.4	Internal IRC temperature drift characteristics (reference temperature 25° C)			956
R.5	Low voltage reset threshold voltage (test temperature 25° C) .....			956
Appendix S Application Notes.....				957
S.1	STC8A8K64D4-64Pin/48Pin Series.....			957
Appendix T PCB Design Guidance for Touch Buttons .....				958
Appendix Solder Method for .....		UQFN/DFN Package Components		960
Appendix Notes on the VSTC8A8K64D4 Series Microcontroller Replacing the STC8A8K64S4A12 Series				963
Appendix W Updated records.....				965

STC MCU

# 1 Overview

The STC8A8K64D4 series microcontroller is a microcontroller that does not require an external crystal and an external reset, and is an 8051 microcontroller that targets super anti-interference/ultra-low cost/high speed/low power consumption. At the same operating frequency, the STC8A8K64D4 series microcontroller is approximately 12 times faster than the conventional 8051 (11.2~13.2 times faster), and it takes only 147 clocks to execute all 111 instructions sequentially, while the conventional 8051 requires 1944 clocks. The STC8A8K64D4 series microcontroller is a single clock/machine cycle (1T) microcontroller produced by STC, which is a new generation 8051 microcontroller with wide voltage/high speed/high reliability/low power consumption/strong anti-static/strong anti-interference and super encryption. Instruction code is fully compatible with traditional 8051.

MCU with internal high precision R/C clock ( $\pm 0.3\%$ ,  $+25^{\circ}\text{C}$  at room temperature),  $-1.38\%$  to  $+1.42\%$  temperature drift ( $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$ )  
 $-0.88\%$  to  $+1.05\%$  temperature drift ( $-20^{\circ}\text{C}$  to  $+65^{\circ}\text{C}$ ). 4MHz to 45MHz wide range configurable for ISP programming (Note: temperature range is  $-40^{\circ}\text{C}$ )

The maximum frequency must be kept below 45MHz at  $\sim +85^{\circ}\text{C}$ , eliminating the need for an expensive external crystal and external reset circuitry (highly reliable reset circuitry has been integrated internally, and a 4-stage reset threshold voltage is optional for ISP programming).

There are three optional clock sources inside the MCU: internal high precision IRC clock (adjustable during ISP download), internal 32KHz low speed IRC, external 4M~33M crystal or external clock signal. Clock source can be freely selected in the user code and the clock source can be divided by an 8-bit divider before providing the clock signal to the CPU and various peripherals (such as timer, serial port, SPI, etc.).

The MCU provides two low-power modes: IDLE mode and STOP mode. In IDLE mode, the MCU stops providing clock to the CPU, the CPU has no clock, and the CPU stops executing instructions, but all peripherals are still working, and the power consumption is about 1.0mA (6MHz operating frequency). The power consumption can be reduced to 0.6uA@Vcc=5.0V and 0.4uA@Vcc=3.3V.

掉电模式可以使用 INT0(P3.2)、INT1(P3.3)、INT2(P3.6)、INT3(P3.7)、INT4(P3.0)、T0(P3.4)、T1(P3.5)、T2(P1.2)、T3(P0.4)、T4(P0.6)、RXD(P3.0/P3.6/P1.6/P4.3)、RXD2(P1.0/P4.0)、RXD3(P0.0/P5.0)、RXD4(P0.2/P5.0)、CCP0(P1.7/P2.3/P7.0/P3.3)、CCP1(P1.6/P2.4/P7.1/P3.2)、CCP2(P1.5/P2.5/P7.2/P3.1)、CCP3(P1.4/P2.6/P7.3/P3.0)、I2C\_SDA(P1.4/P2.4/P3.3)、SPI\_SS(P1.2/P2.2/P3.5), and all ports I/O interrupts, the Comparator interrupt, low voltage detection interrupt, and power-down wake-up timer wake-up.

The MCU provides a rich set of digital peripherals (serial, timer, PCA, enhanced PWM, and I<sup>2</sup>C, SPI) to interface with the analog peripherals (12-bit\*15-way ultra-high-speed ADC and comparator with a speed of up to 800K samples per second) can meet the design needs of a wide range of users.

The STC8A8K64D4 series microcontroller has an enhanced dual data pointer integrated inside. Through program control, the data pointer can be automatically incremented or

document and the two data pins can be automatically switched.

product lines	I/O	UART	timers	ADC	enhanced PWM	PCA	CMP	SPI	I2C	MDU16	I/O disruptions	LCM	DMA
STC8A8K64D4 Series-64Pin/48Pin	59	4	5	15CH*12B	●	●	●	●	●	●	●	●	●





- ✓ 8192 bytes of internal extended RAM (internal XDATA)

#### ➤ Clock control

- ✓ Internal high-precision IRC (4MHz to 45MHz, adjustable up or down during ISP programming, and user software dividable to lower frequencies, e. g. 100KHz)  
Error  $\pm 0.3\%$  (25° C at room temperature)  
-1.38% to +1.42% temperature drift (full temperature range, -40° C to 85° C)  
-0.88% to +1.05% temperature drift (temperature range, -20° C to 65° C)
- ✓ Internal 32KHz low-speed IRC (large error)
- ✓ External crystal (4MHz to 45MHz) and external clock

#### ➤ reset (a dislocated joint)

- ✓ Hardware Reset  
Power-on reset, measured voltage 1.69V to 1.82V. (Valid when the chip is not enabled for low-voltage reset)

The power-on reset voltage consists of a voltage range of an upper limit voltage and a lower limit voltage. When the operating voltage drops down from 5V/3.3V to the lower threshold voltage of the power-on reset, the chip is in the reset state; when the voltage rises from 0V to the upper threshold voltage of the power-on reset, the chip is released from the reset state.

Reset pin reset, factory default P5.4 is I/O port, ISP download can set P5.4 pin as reset pin (Note: when set P5.4

(When the pin is a reset pin, the reset level is low)

Watchdog overflow reset

Low voltage detection reset, providing 4 levels of low voltage detection voltage: 2.0V (measured 1.90V to 2.04V), 2.4V (measured 2.30V to 2.50V), and

2.7V (measured 2.61V to 2.82V), 3.0V (measured 2.90V to 3.13V).

Each low voltage detection voltage stage is a voltage range consisting of an upper voltage and a lower voltage, with low voltage detection taking effect when the operating voltage drops down from 5V/3.3V to the lower threshold voltage for low voltage detection, and when the voltage rises from 0V to the upper threshold voltage for low voltage detection.

- ✓ Software Reset  
Software method to write reset trigger register

#### ➤ disruptions

- ✓ Provides 43 interrupt sources: INT0 (supports rising and falling edge interrupts), INT1 (supports rising and falling edge interrupts), INT2 (supports falling edge interrupts only), INT3 (supports falling edge interrupts only), INT4 (supports falling edge interrupts only), Timer 0, Timer 1, Timer 2, Timer 3, Timer 4, Serial 1, Serial 2, Serial 3, Serial 4 ADC Analog to Digital Conversion, LVD Low Voltage Detection, SPI, I2C, Comparator, PCA/CCP/PWM, Enhanced PWM, Enhanced PWM Exception Detection, all I/O interrupts (8 groups), LCD driver interrupts, DMA receive and transmit interrupts for serial port 1, DMA receive and transmit interrupts for serial port 2, DMA receive and transmit interrupts for serial port 3, DMA receive and transmit interrupts for serial port 4, DMA interrupts for SPI, DMA interrupts for ADC DMA interrupt for SPI, DMA interrupt for ADC, DMA interrupt for LCD driver, and memory-to-memory DMA interrupt.
- ✓ Provides 4 levels of interrupt priority

INT4 (P3.0), T0 (P3.4), T1 (P3.5), T2 (P1.2), T3 (P0.4), T4 (P0.6), RXD (P3.0/P3.6/P1.6/P4.3), RXD2 (P1.0/P4.0), RXD3 (P0.0/P5.0), RXD4 (P0.2/P5.0), CCP0 (P1.7/P2.3/P7.0/P3.3), CCP1 (P1.6/P2.4/P7.1/P3.2), CCP2 (P1.5/P2.5/P7.2/P3.1), CCP3 (P1.4/P2.6/P7.3/P3.0), I2C\_SDA (P1.4/P2.4/P3.3), SPI\_SS (P1.2/P2.2/P3.5), and I/O interrupts for all ports, and Comparator interrupt, low voltage detection interrupt, power-down wake-up timer wake-up

➤ digital peripheral

- ✓ 5 16-bit timers: Timer 0, Timer 1, Timer 2, Timer 3, Timer 4, with Mode 3 of Timer 0 having NMI (non-maskable interrupt) function, mode 0 of timer 0 and timer 1 is 16-bit auto-reload mode
- ✓ 4 high-speed serial ports: serial port 1, serial port 2, serial port 3, serial port 4, baud rate clock source up to FOSC/4
- ✓ Four 16-bit PCA modules: CCP0, CCP1, CCP2, CCP3, for capture, high-speed pulse output, and 6/7/8/10-bit PWM



exports

- ✓ 8 groups of 15-bit enhanced PWM for control signals with deadband and support for external anomaly detection, plus 4 groups of conventional  
PCA/CCP/PWM can be used as PWM
- ✓ SPI: Supports host and slave modes and automatic host/slave switching
- ✓ I<sup>2</sup>C: Master and slave modes supported
- ✓ **MDU16:** Hardware 16-bit multiplier and divider (supports 32-bit divide by 16-bit, 16-bit divide by 16-bit, 16-bit by 16-bit, data shift, and data normalization)
- ✓ **I/O port interrupts:** all I/O support interrupts, each I/O interrupt has a separate interrupt entry address, all I/O interrupts can support 4 interrupt modes: high interrupt, low interrupt, rising edge interrupt, falling edge interrupt  
(I/O port interrupts in this series are power-down wake-up capable and have 4 levels of interrupt priority)
- ✓ **LCD driver module:** supports both 8080 and 6800 interfaces and 8 and 16 bit data widths
- ✓ **DMA:** Support SPI shift to receive data to memory, SPI shift to send data to memory, serial port 1/2/3/4 to receive data to memory, and  
Serial port 1/2/3/4 sends data from memory, ADC automatically samples data to memory (while calculating average values), LCD driver sends data from memory, and memory-to-memory data copy
- ✓ **Hardware numeric ID:** 32 bytes supported

➤ Analog Peripherals

- ✓ Ultra-high-speed ADC supporting **12-bit high-precision** 15-channel (channel 0 to channel 14) analog-to-digital conversion **at speeds up to 800K (800,000 ADC conversions per second)**
- ✓ Channel 15 of the ADC is used to test the internal 1.19V reference source (the chip is shipped with the internal reference source adjusted to 1.19V)
- ✓ Comparator, a set of comparators (the positive side of the comparator selects the CMP+ port, CMP+\_2, CMP+\_3 and all ADC input ports, and the negative side of the comparator selects the CMP- port and the internal 1.19V reference source, so the comparator can be used as a multiplexer for time division)
- ✓ DACs: 8 enhanced PWM timers can be used as 8 DACs, 4 PCAs can be used as 4 DACs

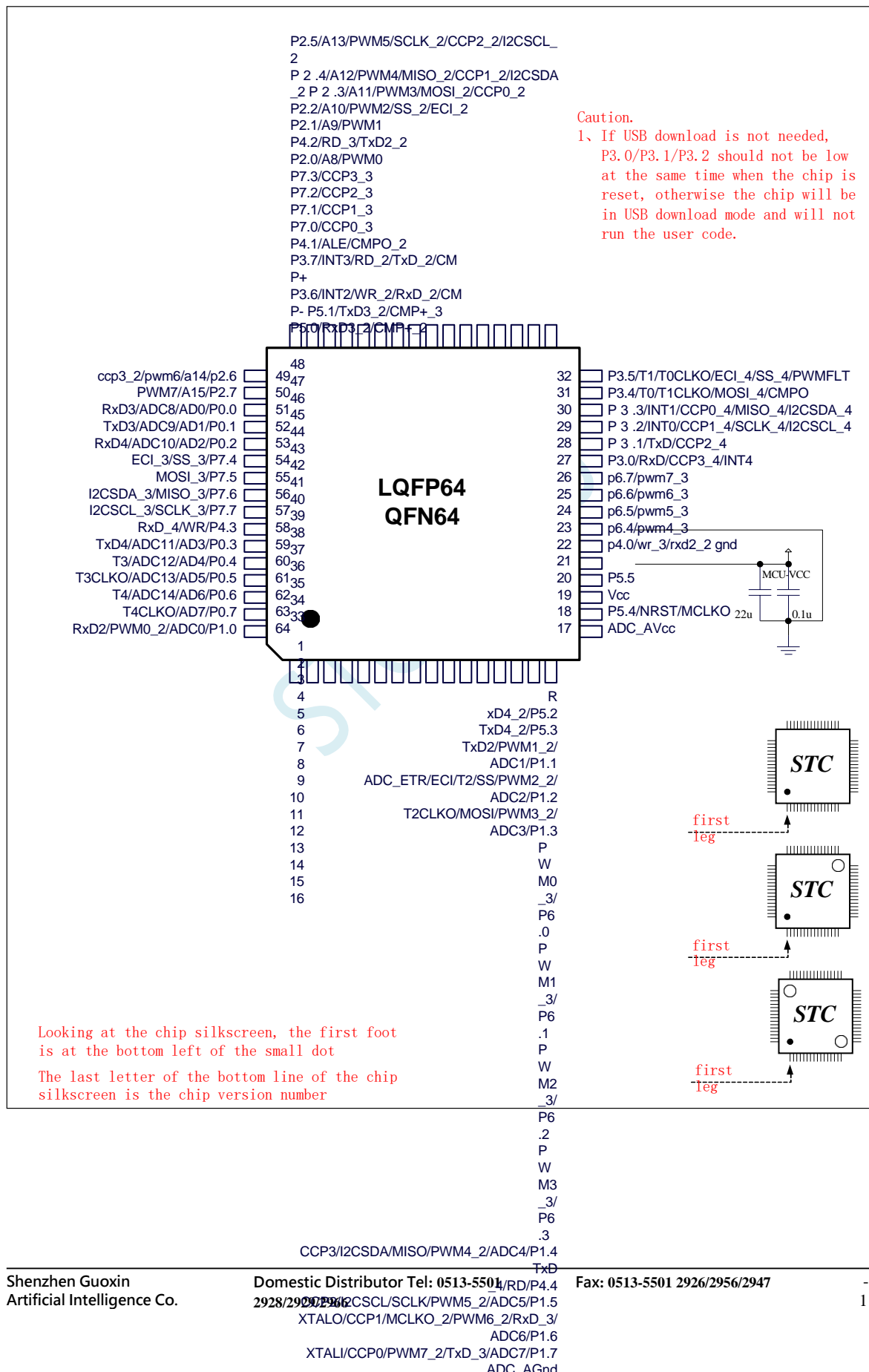
➤ GPIO

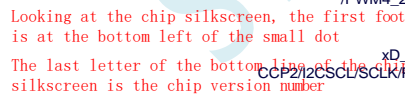
- ✓ Up to 59 GPIOs: P0.0~P0.7, P1.0~P1.7, P2.0~P2.7, P3.0~P3.7, P4.0~P4.4, P5.0~P5.5, P6.0~P6.7, P7.0~P7.7
- ✓ All GPIOs support the following 4 modes: quasi-bidirectional port mode, strong push-pull output mode, open-drain output mode, and high-resistance input mode
- ✓ **Except for P3.0 and P3.1, the status of all I/O ports after power up is high resistance input state, user must set I/O first when using I/O port**  
**In addition, each I/O can be independently enabled with an internal 4K pull-up resistor**

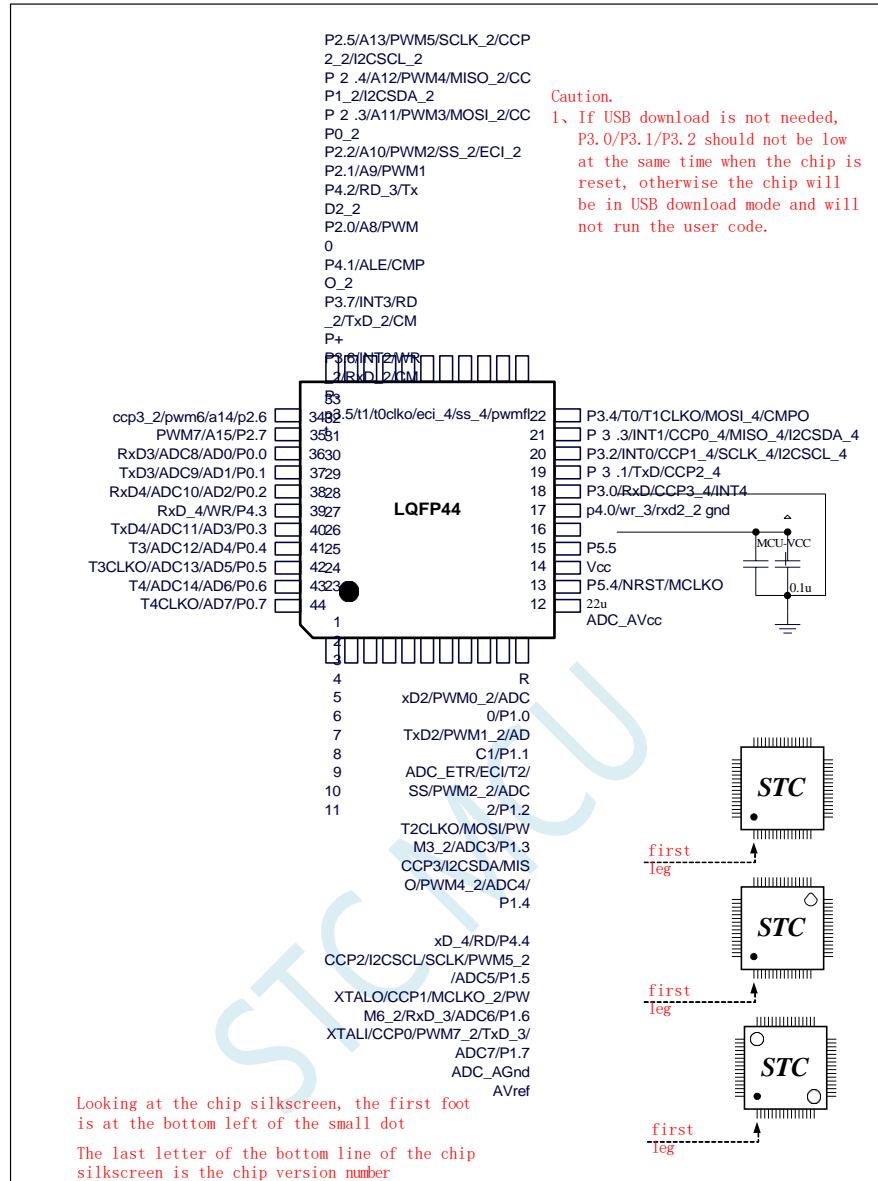
➤ package

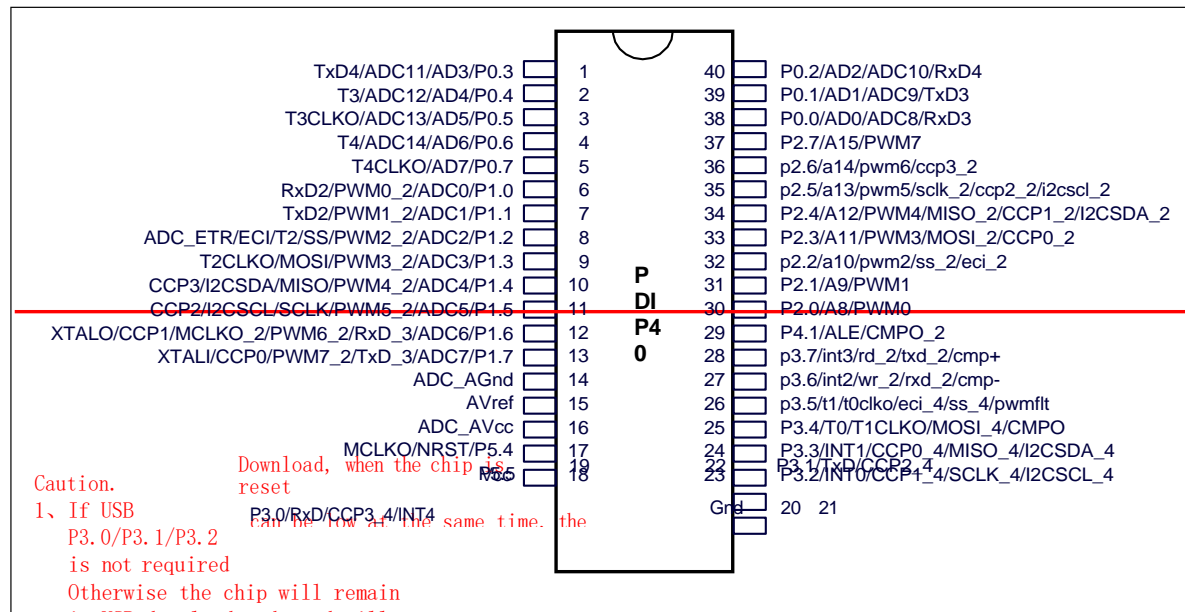
- ✓ LQFP64, LQFP48, LQFP44

### 2.1.2 Pinout Diagram, Minimum System









## 2.1.3 Pin Description

num ber				name (of a thing)	types	instr ucti ons
LQFP64	LQFP48	LQFP44	PDIP40			
1	1			P5.2	I/O	Standard IO port
				RxD4_2	I	Receive pin of serial port 4
2	2			P5.3	I/O	Standard IO port
				TxD4_2	O	Transmit pin of serial port 4
3	3	2	7	P1.1	I/O	Standard IO port
				ADC1	I	ADC Analog Input Channel 1
				PWM1_2	O	Enhanced PWM channel 1 output pin
				TxD2	O	Transmit pin of serial port 2
4	4	3	8	P1.2	I/O	Standard IO port
				ADC2	I	ADC analog input channel 2
				PWM2_2	O	Enhanced PWM channel 2 output pin
				SS	I/O	SPI Slave Selection
				T2	I	Timer 2 External clock input
				ECI	I	External pulse input for PCA
				ADC_ETR	I	ADC external trigger input
5	5	4	9	P1.3	I/O	Standard IO port
				ADC3	I	ADC analog input channel 3
				PWM3_2	O	Enhanced PWM channel 3 output pin
				MOSI	I/O	SPI host output slave input
				T2CLKO	O	Timer 2 clock division output
6				P6.0	I/O	Standard IO port
				PWM0_3	O	Enhanced PWM Channel 0 Output Pin
7				P6.1	I/O	Standard IO port
				PWM1_3	O	Enhanced PWM channel 1 output pin
8				P6.2	I/O	Standard IO port
				PWM2_3	O	Enhanced PWM channel 2 output pin
9				P6.3	I/O	Standard IO port
				PWM3_3	O	Enhanced PWM channel 3 output pin

num ber				name (of a thing)	types	instr ucti ons
LQFP64	LQFP48	LQFP44	PDIP40			
10	6	5	10	P1.4	I/O	Standard IO port
				ADC4	I	ADC analog input channel 4
				PWM4_2	O	Enhanced PWM channel 4 output pin
				MISO	I/O	SPI host input slave output
				SDA	I/O	Data line for I2C interface
				CCP3	I/O	Capture input and pulse output of PCA
11	7	6		P4.4	I/O	Standard IO port
				RD	O	Read signal line for external bus
				TxD_4	O	Transmit pin of serial port 1
12	8	7	11	P1.5	I/O	Standard IO port
				ADC5	I	ADC analog input channel 5
				PWM5_2	O	Enhanced PWM channel 5 output pin
				SCLK	I/O	Clock pin of SPI
				SCL	I/O	Clock line for I2C
				CCP2	I/O	Capture input and pulse output of PCA
13	9	8	12	P1.6	I/O	Standard IO port
				ADC6	I	ADC Analog Input Channel 6
				RxD_3	I	Receive pin of serial port 1
				PWM6_2	O	Enhanced PWM channel 6 output pin
				MCLKO_2	O	Master clock divider output
				CCP1	I/O	Capture input and pulse output of PCA
				XTALO	O	Output pin of the external crystal
14	10	9	13	P1.7	I/O	Standard IO port
				ADC7	I	ADC Analog Input Channel 7
				TxD_3	O	Transmit pin of serial port 1
				PWM7_2	O	Enhanced PWM channel 7 output pin
				CCP0	I/O	Capture input and pulse output of PCA
				XTALI	I	Input pin for external crystal/external clock
15	11	10	14	ADC_AGnd	GND	ADC Ground
16	12	11	15	AVref	I	Reference voltage pin of ADC
17	13	12	16	ADC_AVcc	VCC	ADC Power Pin
18	14	13	17	P5.4	I/O	Standard IO port
				NRST	I	Reset pin (low reset)

STC8A8K64D4 Series		Official website:		Technical Support:		Selection Consultant:
Technical Manual		<a href="http://www.STCAL.com">www.STCAL.com</a>		MCLKO	O <sup>19864585985</sup>	Master clock divider output
19	15	14	18	Vcc	VCC	Power foot
20	16	15	19	P5.5	I/O	Standard IO port
21	17	16	20	Gnd	GND	earth (wire)



num ber				name (of a thing)	types	instr ucti ons
LQFP64	LQFP48	LQFP44	PDIP40			
22	18	17		P4.0	I/O	Standard IO port
				WR_3	O	Write signal line for external bus
				RxD2_2	I	Receive pin of serial port 2
23				P6.4	I/O	Standard IO port
				PWM4_3	O	Enhanced PWM channel 4 output pin
24				P6.5	I/O	Standard IO port
				PWM5_3	O	Enhanced PWM channel 5 output pin
25				P6.6	I/O	Standard IO port
				PWM6_3	O	Enhanced PWM channel 6 output pin
26				P6.7	I/O	Standard IO port
				PWM7_3	O	Enhanced PWM channel 7 output pin
27	19	18	21	P3.0	I/O	Standard IO port
				RxD	I	Receive pin of serial port 1
				CCP3_4	I/O	Capture input and pulse output of PCA
				INT4	I	External interrupts 4
28	20	19	22	P3.1	I/O	Standard IO port
				TxD	O	Transmit pin of serial port 1
				CCP2_4	I/O	Capture input and pulse output of PCA
29	21	20	23	P3.2	I/O	Standard IO port
				INT0	I	External interrupt 0
				CCP1_4	I/O	Capture input and pulse output of PCA
				SCLK_4	I/O	Clock pin of SPI
				SCL_4	I/O	Clock line for I2C
30	22	21	24	P3.3	I/O	Standard IO port
				INT1	I	External interrupt 1
				CCP0_4	I/O	Capture input and pulse output of PCA
				MISO_4	I/O	SPI host input slave output
				SDA_4	I/O	Data line for I2C interface
31	23	22	25	P3.4	I/O	Standard IO port
				T0	I	Timer 0 External clock input
				T1CLKO	O	Timer 1 Clock division output
				MOSI_4	I/O	SPI host output slave input
				CMPO	O	Comparator Output

num ber				name (of a thing)	types	instr ucti ons
LQFP64	LQFP48	LQFP44	PDIP40			
32	24	23	26	P3.5	I/O	Standard IO port
				T1	I	Timer 1 External clock input
				T0CLKO	O	Timer 0 Clock division output
				ECL_4	I	External pulse input for PCA
				SS_4	I	Slave select pin for SPI (host is output)
				PWMFLT	I	Enhanced external exception detection pin for PWM
33	25			P5.0	I/O	Standard IO port
				RxD3_2	I	Receive pin of serial port 3
				CMP+_2	I	Comparator positive input
34	26			P5.1	I/O	Standard IO port
				TxD3_2	O	Transmit pin of serial port 3
				CMP+_3	I	Comparator positive input
35	27	24	27	P3.6	I/O	Standard IO port
				INT2	I	External interrupts 2
				WR_2	O	Write signal line for external bus
				RxD_2	I	Receive pin of serial port 1
				CMP-	I	Comparator negative input
36	28	25	28	P3.7	I/O	Standard IO port
				INT3	I	External interrupts 3
				RD_2	O	Read signal line for external bus
				TxD_2	O	Transmit pin of serial port 1
				CMP+	I	Comparator positive input
37	29	26	29	P4.1	I/O	Standard IO port
				ALE	O	address latch signal
				CMPO_2	O	Comparator Output
38				P7.0	I/O	Standard IO port
				CCP0_3	I/O	Capture input and pulse output of PCA
39				P7.1	I/O	Standard IO port
				CCP1_3	I/O	Capture input and pulse output of PCA
40				P7.2	I/O	Standard IO port
				CCP2_3	I/O	Capture input and pulse output of PCA
41				P7.3	I/O	Standard IO port
				CCP3_3	I/O	Capture input and pulse output of PCA
42	30	27	30	P2.0	I/O	Standard IO port
				A8	I	address bus (computer)
				PWM0	O	Enhanced PWM Channel 0 Output Pin
43	31	28		P4.2	I/O	Standard IO port
				RD_3	O	Read signal line for external bus
				TxD2_2	O	Transmit pin of serial port 2

num ber				name (of a thing)	types	instr ucti ons
LQFP64	LQFP48	LQFP44	PDIP40			
44	32	29	31	P2.1	I/O	Standard IO port
				A9	I	address bus (computer)
				PWM1	O	Enhanced PWM channel 1 output pin
45	33	30	32	P2.2	I/O	Standard IO port
				A10	I	address bus (computer)
				PWM2	O	Enhanced PWM channel 2 output pin
				SS_2	I	Slave select pin for SPI (host is output)
46	34	31	33	ECI_2	I	External pulse input for PCA
				P2.3	I/O	Standard IO port
				A11	I	address bus (computer)
				PWM3	O	Enhanced PWM channel 3 output pin
				MOSI_2	I/O	SPI host output slave input
47	35	32	34	CCP0_2	I/O	Capture input and pulse output of PCA
				P2.4	I/O	Standard IO port
				A12	I	address bus (computer)
				PWM4	O	Enhanced PWM channel 4 output pin
				MISO_2	I/O	SPI host input slave output
				SDA_2	I/O	Data line for I2C interface
48	36	33	35	CCP1_2	I/O	Capture input and pulse output of PCA
				P2.5	I/O	Standard IO port
				A13	I	address bus (computer)
				PWM5	O	Enhanced PWM channel 5 output pin
				SCLK_2	I/O	Clock pin of SPI
49	37	34	36	SCL_2	I/O	Clock line for I2C
				CCP2_2	I/O	Capture input and pulse output of PCA
				P2.6	I/O	Standard IO port
				A14	I	address bus (computer)
50	38	35	37	PWM6	O	Enhanced PWM channel 6 output pin
				CCP3_2	I/O	Capture input and pulse output of PCA
				P2.7	I/O	Standard IO port
51	39	36	38	A15	I	address bus (computer)
				PWM7	O	Enhanced PWM channel 7 output pin
				P0.0	I/O	Standard IO port
				AD0	I	address bus (computer)
				ADC8	I	ADC Analog Input Channel 8
				RxD3	I	Receive pin of serial port 3

num ber				name (of a thing)	type s	instr ucti ons
LQFP64	LQFP48	LQFP44	PDIP40			
52	40	37	39	P0.1	I/O	Standard IO port
				AD1	I	address bus (computer)
				ADC9	I	ADC Analog Input Channel 9
				TxD3	O	Transmit pin of serial port 3
53	41	38	40	P0.2	I/O	Standard IO port
				AD2	I	address bus (computer)
				ADC10	I	ADC Analog Input Channel 10
				RxD4	I	Receive pin of serial port 4
54				P7.4	I/O	Standard IO port
				SS_3	I	Slave select pin for SPI (host is output)
				ECI_3	I	External pulse input for PCA
55				P7.5	I/O	Standard IO port
				MOSI_3	I/O	SPI host output slave input
56				P7.6	I/O	Standard IO port
				MISO_3	I/O	SPI host input slave output
				SDA_3	I/O	Data line for I2C interface
57				P7.7	I/O	Standard IO port
				SCLK_3	I/O	Clock pin of SPI
				SCL_3	I/O	Clock line for I2C
58	42	39		P4.3	I/O	Standard IO port
				WR	O	Write signal line for external bus
				RxD_4	I	Receive pin of serial port 1
59	43	40	1	P0.3	I/O	Standard IO port
				AD3	I	address bus (computer)
				ADC11	I	ADC Analog Input Channel 11
				TxD4	O	Transmit pin of serial port 4
60	44	41	2	P0.4	I/O	Standard IO port
				AD4	I	address bus (computer)
				ADC12	I	ADC Analog Input Channel 12
				T3	I	Timer 3 External clock input

num ber				name (of a thing)	type s	instr ucti ons
LQFP64	LQFP48	LQFP44	PDIP40			
61	45	42	3	P0.5	I/O	Standard IO port
				AD5	I	address bus (computer)
				ADC13	I	ADC Analog Input Channel 13
				T3CLKO	O	Timer 3 clock division output
62	46	43	4	P0.6	I/O	Standard IO port
				AD6	I	address bus (computer)
				ADC14	I	ADC Analog Input Channel 14
				T4	I	Timer 4 External clock input
63	47	44	5	P0.7	I/O	Standard IO port
				AD7	I	address bus (computer)
				T4CLKO	O	Timer 4 clock division output
64	48	1	6	P1.0	I/O	Standard IO port
				ADC0	I	ADC Analog Input Channel 0
				PWM0_2	O	Enhanced PWM Channel 0 Output Pin
				RxD2	I	Receive pin of serial port 2

## 3 Function foot cut change

The STC8A8K64D4 series microcontroller's special peripheral serial port, SPI, PCA, I<sup>2</sup>C, and bus control pins can be switched directly across multiple I/Os to enable time-sharing multiplexing of one peripheral as multiple devices.

### 3.1 Function pin switching related registers

symbolic	description	address	Bit Addresses and Symbols								reset value
			B7	B6	B5	B4	B3	B2	B1	B0	
BUS_SPEED	Bus Speed Control Register	A1H	RW_S[1:0]						SPEED[1:0]		00xx,xx00
P_SW1	Peripheral port switching register 1	A2H	S1_S[1:0]		CCP_S[1:0]		SPI_S[1:0]		0	-	nn00,000x
P_SW2	Peripheral port switching register 2	BAH	EAXFR		I2C_S[1:0]		CMPO_S	S4_S	S3_S	S2_S	0x00,0000

symbolic	description	address	Bit Addresses and Symbols								reset value
			B7	B6	B5	B4	B3	B2	B1	B0	
PWM0CR	PWM0 control register	FF14H	ENC0O	C0INI	-	C0_S[1:0]		EC0I	EC0T2SI	EC0T1SI	00x0,0000
PWM1CR	PWM1 control register	FF1CH	ENC1O	C1INI	-	C1_S[1:0]		EC1I	EC1T2SI	EC1T1SI	00x0,0000
PWM2CR	PWM2 control register	FF24H	ENC2O	C2INI	-	C2_S[1:0]		EC2I	EC2T2SI	EC2T1SI	00x0,0000
PWM3CR	PWM3 Control Register	FF2CH	ENC3O	C3INI	-	C3_S[1:0]		EC3I	EC3T2SI	EC3T1SI	00x0,0000
PWM4CR	PWM4 Control Register	FF34H	ENC4O	C4INI	-	C4_S[1:0]		EC4I	EC4T2SI	EC4T1SI	00x0,0000
PWM5CR	PWM5 control register	FF3CH	ENC5O	C5INI	-	C5_S[1:0]		EC5I	EC5T2SI	EC5T1SI	00x0,0000
PWM6CR	PWM6 Control Register	FF44H	ENC6O	C6INI	-	C6_S[1:0]		EC6I	EC6T2SI	EC6T1SI	00x0,0000
PWM7CR	PWM7 Control Register	FF4CH	ENC7O	C7INI	-	C7_S[1:0]		EC7I	EC7T2SI	EC7T1SI	00x0,0000
MCLKOCR	Master Clock Output Control Register	FE05H	MCLKO_S	MCLKODIV [6:0]							0000,0000
LCMIFCFG	LCM Interface Configuration Register	FE50H	LCMIFIE	-	LCMIFIP[1:0]		LCMIFPS[1:0]		D16_D8	M68_80	0x00,0000
LCMIFCFG2	LCM Interface Configuration Register 2	FE51H		LCMIFEXPS[1:0]		SETUPT[2:0]			HOLDT[1:0]		x000,0000

### 3.1.1 Bus Speed Control Register (BUS\_SPEED)

symbolic	address	B7	B6	B5	B4	B3	B2	B1	B0
BUS_SPEED	A1H	RW_S[1:0]						SPEED[1:0]	

RW\_S[1:0]: external bus RD/WR control line select bits

RW_S[1:0]	RD	WR
00	P4.4	P4.3
01	P3.7	P3.6
10	P4.2	P4.0
11	retain	

### 3.1.2 Peripheral port switching control register 1 (P\_SW1) serial port 1, CCP, SPI

#### switch over

symbolic	address	B7	B6	B5	B4	B3	B2	B1	B0
P_SW1	A2H	S1_S[1:0]		CCP_S[1:0]		SPI_S[1:0]		0	-

S1\_S[1:0]: Serial port 1 function pin select bits

S1_S[1:0]	RxD	TxD
00	P3.0	P3.1
01	P3.6	P3.7
10	P1.6	P1.7
11	P4.3	P4.4

CCP\_S[1:0]: PCA function pin select bits

CCP_S[1:0]	ECI	CCP0	CCP1	CCP2	CCP3
00	P1.2	P1.7	P1.6	P1.5	P1.4
01	P2.2	P2.3	P2.4	P2.5	P2.6
10	P7.4	P7.0	P7.1	P7.2	P7.3
11	P3.5	P3.3	P3.2	P3.1	P3.0

SPI\_S[1:0]: SPI function pin select bits

SPI_S[1:0]	SS	MOSI	MISO	SCLK
00	P1.2	P1.3	P1.4	P1.5
01	P2.2	P2.3	P2.4	P2.5
10	P7.4	P7.5	P7.6	P7.7
11	P3.5	P3.4	P3.3	P3.2

### 3.1.3 Peripheral port switching control register 2 (P\_SW2)

#### serial port 2/3/4, I2C, comparator output switching

symbolic	address	B7	B6	B5	B4	B3	B2	B1	B0
P_SW2	BAH	EAXFR	-	I2C_S[1:0]		CMPO_S	S4_S	S3_S	S2_S





0: Access to XFR is prohibited

1: Enables access to XFR.

**When access to the XFR is required, the EAXFR must be set to 1 before the XFR can be read or written to properly**

I2C\_S[1:0]: I<sup>2</sup>C function pin select bits

I2C_S[1:0]	SCL	SDA
00	P1.5	P1.4
01	P2.5	P2.4
10	P7.7	P7.6
11	P3.2	P3.3

CMPO\_S: Comparator output pin select bit

CMPO_S	CMPO
0	P3.4
1	P4.1

S4\_S: Serial port 4 function pin selection bit

S4_S	RxD4	TxD4
0	P0.2	P0.3
1	P5.2	P5.3

S3\_S: Serial port 3 function pin selection bit

S3_S	RxD3	TxD3
0	P0.0	P0.1
1	P5.0	P5.1

S2\_S: Serial port 2 function pin select bit

S2_S	RxD2	TxD2
0	P1.0	P1.1
1	P4.0	P4.2

### 3.1.4 Clock Select Register (MCLKOCR)

symbolic	address	B7	B6	B5	B4	B3	B2	B1	B0
MCLKOCR	FE05H	MCLKO_S	MCLKODIV [6:0]						

MCLKO\_S: Master clock output pin select bit

MCLKO_S	MCLKO
0	P5.4
1	P1.6

### 3.1.5 Enhanced PWM Control Register (PWMnCR)

symbolic	address	B7	B6	B5	B4	B3	B2	B1	B0
PWM0CR	FF14H	ENC0O	C0INI	-	C0_S[1:0]		EC0I	EC0T2SI	EC0T1SI
PWM1CR	FF1CH	ENC1O	C1INI	-	C1_S[1:0]		EC1I	EC1T2SI	EC1T1SI
PWM2CR	FF24H	ENC2O	C2INI	-	C2_S[1:0]		EC2I	EC2T2SI	EC2T1SI
PWM3CR	FF2CH	ENC3O	C3INI	-	C3_S[1:0]		EC3I	EC3T2SI	EC3T1SI

STC8A8K64D4 Series			Official website:		Technical Support:		Selection Consultant:	
Technical Manual	FF34H	ENC4C	<a href="http://www.STCAL.com">www.STCAL.com</a>	-	C4_S[1..9]	EC4I	EC4T2SI	EC4T1SI

PWM5CR	FF3CH	ENC5O	C5INI	-	C5_S[1:0]	EC5I	EC5T2SI	EC5T1SI
PWM6CR	FF44H	ENC6O	C6INI	-	C6_S[1:0]	EC6I	EC6T2SI	EC6T1SI
PWM7CR	FF4CH	ENC7O	C7INI	-	C7_S[1:0]	EC7I	EC7T2SI	EC7T1SI

C0\_S[1:0]: Enhanced PWM channel 0 output pin select bits

C0_S[1:0]	PWM0
00	P2.0
01	P1.0
10	P6.0
11	retain

C1\_S[1:0]: Enhanced PWM channel 1 output pin select bits

C1_S[1:0]	PWM1
00	P2.1
01	P1.1
10	P6.1
11	retain

C2\_S[1:0]: Enhanced PWM channel 2 output pin select bits

C2_S[1:0]	PWM2
00	P2.2
01	P1.2
10	P6.2
11	retain

C3\_S[1:0]: Enhanced PWM channel 3 output pin select bits

C3_S[1:0]	PWM3
00	P2.3
01	P1.3
10	P6.3
11	retain

C4\_S[1:0]: Enhanced PWM channel 4 output pin select bits

C4_S[1:0]	PWM4
00	P2.4
01	P1.4
10	P6.4
11	retain

C5\_S[1:0]: Enhanced PWM channel 5 output pin select bits

C5_S[1:0]	PWM5
00	P2.5
01	P1.5
10	P6.5
11	retain

C6\_S[1:0]: Enhanced PWM channel 6 output pin select bits

C6_S[1:0]	PWM6
00	P2.6
01	P1.6
10	P6.6

11	retain
----	--------

C7\_S[1:0]: Enhanced PWM channel 7 output pin select bits

C7_S[1:0]	PWM7
00	P2.7
01	P1.7
10	P6.7
11	retain

STC MCU

### 3.1.6 LCM Interface Configuration Register (LCMIFCFG )

symbolic	addresses	B7	B6	B5	B4	B3	B2	B1	B0
LCMIFCFG	FE50H	LCMIFIE	-	LCMIFIP[1:0]		LCMIFDPS[1:0]		D16_D8	M68_I80
LCMIFCFG2	FE51H		LCMIFCPS[1:0]		SETUPT[2:0]			HOLDT[1:0]	

LCMIFCPS[1:0]: LCM interface control pin select bits

LCMIFCPS [1:0]	RS	Read signal RD of I8080 Enable signal E of the M6800	Write signal WR of I8080 M6800 read/write signal RW
00	P4.1	P4.4	P4.3
01	P4.1	P3.7	P3.6
10	P4.1	P4.2	P4.0
11	P4.0	P3.7	P3.6

LCMIFDPS[1:0]: LCM interface data pin select bits

LCMIFDPS [1:0]	D16_D8	Data High Byte DB[15:8]	Data low byte DB[7:0]
00	0	N/A	P2[7:0]
01	0	N/A	P6[7:0]
10	0	N/A	P2[7:0]
11	0	N/A	P6[7:0]
00	1	P2[7:0]	P0[7:0]
01	1	P6[7:0]	P2[7:0]
10	1	P2[7:0]	P7 [7:0]
11	1	P6[7:0]	P7 [7:0]

## 3.2 paradigm procedure

### 3.2.1 Serial port 1 Switching

#### C code

// Tested operating frequency of 11.0592MHz

```
#include "reg51.h"
```

```
sfr      P_SW1      = 0xa2;

sfr      P0M1      = 0x93;
sfr      P0M0      = 0x94;
sfr      P1M1      = 0x91;
sfr      P1M0      = 0x92;
sfr      P2M1      = 0x95;
sfr      P2M0      = 0x96;
sfr      P3M1      = 0xb1;
sfr      P3M0      = 0xb2;
sfr      P4M1      = 0xb3;
sfr      P4M0      = 0xb4;
sfr      P5M1      = 0xc9;
sfr      P5M0      = 0xca;
```

```
void main()
```

```
{
    P0M0      =
    0x00; P0M1
    = 0x00;
    P1M0      =
    0x00; P1M1
    = 0x00;
    P2M0      =
    0x00; P2M1
    = 0x00;
    P3M0      =
    0x00; P3M1
    = 0x00;
    P4M0      =
    0x00; P4M1
    = 0x00;
    P5M0      =
    0x00; P5M1
    = 0x00;

    P_SW1 = 0x00; //RXD/P3.0, TXD/P3.1
    // P_SW1 = 0x40; //RXD_2/P3.6, TXD_2/P3.7
    // P_SW1 = 0x80; //RXD_3/P1.6, TXD_3/P1.7
    // P_SW1 = 0xc0; //RXD_4/P4.3, TXD_4/P4.4

    while (1);
}
```

#### assembly code

Operating frequency tested at 11.0592MHz

STC8A8K64D4 Series		Official website:	Technical Support:	Selection Consultant:
Technical Manual		<a href="http://www.STCAL.com">www.STCAL.com</a>	19864585985	13922805190
<i>P_SWI</i>	<i>DATA</i>	<i>0A2H</i>		
<i>P0M1</i>	<i>DATA</i>	<i>093H</i>		
<i>P0M0</i>	<i>DATA</i>	<i>094H</i>		

STC8A8K64D4 Series		Official website:	Technical Support:	Selection Consultant:
Technical Manual		<a href="http://www.STCAL.com">www.STCAL.com</a>	19864585985	13922805190
<i>P1M1</i>	<i>DATA</i>	<i>091H</i>		
<i>P1M0</i>	<i>DATA</i>	<i>092H</i>		
<i>P2M1</i>	<i>DATA</i>	<i>095H</i>		
<i>P2M0</i>	<i>DATA</i>	<i>096H</i>		
<i>P3M1</i>	<i>DATA</i>	<i>0B1H</i>		
<i>P3M0</i>	<i>DATA</i>	<i>0B2H</i>		
<i>P4M1</i>	<i>DATA</i>	<i>0B3H</i>		
<i>P4M0</i>	<i>DATA</i>	<i>0B4H</i>		
<i>P5M1</i>	<i>DATA</i>	<i>0C9H</i>		
<i>P5M0</i>	<i>DATA</i>	<i>0CAH</i>		
	<i>ORG</i>	<i>0000H</i>		
	<i>LJMP</i>	<i>MAIN</i>		
	<i>ORG</i>	<i>0100H</i>		
<i>MAIN:</i>	<i>MOV</i>	<i>SP, #5FH</i>		
	<i>MOV</i>	<i>P0M0, #00H</i>		
	<i>MOV</i>	<i>P0M1, #00H</i>		
	<i>MOV</i>	<i>P1M0, #00H</i>		
	<i>MOV</i>	<i>P1M1, #00H</i>		
	<i>MOV</i>	<i>P2M0, #00H</i>		
	<i>MOV</i>	<i>P2M1, #00H</i>		
	<i>MOV</i>	<i>P3M0, #00H</i>		
	<i>MOV</i>	<i>P3M1, #00H</i>		
	<i>MOV</i>	<i>P4M0, #00H</i>		
	<i>MOV</i>	<i>P4M1, #00H</i>		
	<i>MOV</i>	<i>P5M0, #00H</i>		
	<i>MOV</i>	<i>P5M1, #00H</i>		
	<i>MOV</i>	<i>P_SW1, #00H</i>	<i>;RXD/P3.0, TXD/P3.1</i>	
<i>;</i>	<i>MOV</i>	<i>P_SW1, #40H</i>	<i>;RXD_2/P3.6, TXD_2/P3.7</i>	
<i>;</i>	<i>MOV</i>	<i>P_SW1, #80H</i>	<i>;RXD_3/P1.6, TXD_3/P1.7</i>	
<i>;</i>	<i>MOV</i>	<i>P_SW1, #0C0H</i>	<i>;RXD_4/P4.3, TXD_4/P4.4</i>	
	<i>SJMP</i>	<i>\$</i>		
	<i>END</i>			

## 3.2.2 Serial port 2 Switching

### C code

```
//test operating frequency of
```

```
11.0592MHz #include "reg51.h"
```

```
sfr      P_SW2      = 0xba;

sfr      P0M1       = 0x93;
sfr      P0M0       = 0x94;
sfr      P1M1       = 0x91;
sfr      P1M0       = 0x92;
sfr      P2M1       = 0x95;
sfr      P2M0       = 0x96;
sfr      P3M1       = 0xb1;
sfr      P3M0       = 0xb2;
```



```

sfr      P4M1      = 0xb3;
sfr      P4M0      = 0xb4;
sfr      P5M1      = 0xc9;
sfr      P5M0      = 0xca;

```

```

void main()
{
    P0M0      =
    0x00; P0M1
    = 0x00;
    P1M0      =
    0x00; P1M1
    = 0x00;
    P2M0      =
    0x00; P2M1
    = 0x00;
    P3M0      =
    0x00; P3M1
    = 0x00;
    P4M0      =
    0x00; P4M1
    = 0x00;
    P5M0      =
    0x00; P5M1
    = 0x00;

    P_SW2 = 0x00; //RXD2/P1.0, TXD2/P1.1
// P_SW2 = 0x01; //RXD2_2/P4.0, TXD2_2/P4.2

    while (1);
}

```

## assembly code

Operating frequency tested at 11.0592MHz

```

P_SW2      DATA      0BAH

P0M1      DATA      093H
P0M0      DATA      094H
P1M1      DATA      091H
P1M0      DATA      092H
P2M1      DATA      095H
P2M0      DATA      096H
P3M1      DATA      0B1H
P3M0      DATA      0B2H
P4M1      DATA      0B3H
P4M0      DATA      0B4H
P5M1      DATA      0C9H
P5M0      DATA      0CAH

            ORG        0000H
            LJMP      MAIN

            ORG        0100H
MAIN:
            MOV        SP, #5FH
            MOV        P0M0, #00H
            MOV        P0M1, #00H
            MOV        P1M0, #00H

```

STC8A8K64D4 Series  
Technical Manual

*MOV*  
*MOV*  
*MOV*  
*MOV*  
*MOV*

Official website:  
[www.stc8a.com](http://www.stc8a.com)  
*P1M1, #00H*  
*P2M0, #00H*  
*P2M1, #00H*  
*P3M0, #00H*  
*P3M1, #00H*

Technical Support:  
19864585985

Selection Consultant:  
13922805190

STC8A8K64D4 Series	Official website:	Technical Support:	Selection Consultant:
Technical Manual	<a href="http://www.STCAL.com">www.STCAL.com</a>	19864585985	13922805190
<i>MOV</i>	<i>p4m0, #00h</i>		
<i>MOV</i>	<i>p4m1, #00h</i>		
<i>MOV</i>	<i>p5m0, #00h</i>		
<i>MOV</i>	<i>p5m1, #00h</i>		
<i>MOV</i>	<i>P_SW2,#00H</i>	<i>;RXD2/P1.0, TXD2/P1.1</i>	
<i>MOV</i>	<i>P_SW2,#01H</i>	<i>;RXD2_2/P4.0, TXD2_2/P4.2</i>	
<i>SJMP</i>	<i>\$</i>		
<i>END</i>			

### 3.2.3 Serial port 3 Switching

#### C code

//test operating frequency of

11.0592MHz #include "reg51.h"

```
sfr      P_SW2      = 0xba;

sfr      P0M1      = 0x93;
sfr      P0M0      = 0x94;
sfr      P1M1      = 0x91;
sfr      P1M0      = 0x92;
sfr      P2M1      = 0x95;
sfr      P2M0      = 0x96;
sfr      P3M1      = 0xb1;
sfr      P3M0      = 0xb2;
sfr      P4M1      = 0xb3;
sfr      P4M0      = 0xb4;
sfr      P5M1      = 0xc9;
sfr      P5M0      = 0xca;
```

void main()

```
{
    P0M0      =
    0x00; P0M1
    = 0x00;
    P1M0      =
    0x00; P1M1
    = 0x00;
    P2M0      =
    0x00; P2M1
    = 0x00;
    P3M0      =
    0x00; P3M1
    = 0x00;
    P4M0      =
    0x00; P4M1
    = 0x00;
    P5M0      =
    0x00; P5M1
    = 0x00;

    P_SW2 = 0x00; //RXD3/P0.0, TXD3/P0.1
// P_SW2 = 0x02; //RXD3_2/P5.0, TXD3_2/P5.1
}
```

STC8A8K64D4 Series	Official website:	Technical Support:	Selection Consultant:
Technical Manual <i>while (1);</i> }	<a href="http://www.STCAL.com">www.STCAL.com</a>	19864585985	13922805190
<hr/>			
assembly code			
<hr/>			

*P\_SW2 DATA 0BAH*

*P0M1 DATA 093H*

*P0M0 DATA 094H*

*P1M1 DATA 091H*

*P1M0 DATA 092H*

*P2M1 DATA 095H*

*P2M0 DATA 096H*

*P3M1 DATA 0B1H*

*P3M0 DATA 0B2H*

*P4M1 DATA 0B3H*

*P4M0 DATA 0B4H*

*P5M1 DATA 0C9H*

*P5M0 DATA 0CAH*

*ORG 0000H*

*LJMP MAIN*

*ORG 0100H*

*MAIN:*

*MOV SP, #5FH*

*MOV P0M0, #00H*

*MOV P0M1, #00H*

*MOV P1M0, #00H*

*MOV P1M1, #00H*

*MOV P2M0, #00H*

*MOV P2M1, #00H*

*MOV P3M0, #00H*

*MOV P3M1, #00H*

*MOV P4M0, #00H*

*MOV P4M1, #00H*

*MOV P5M0, #00H*

*MOV P5M1, #00H*

*MOV P\_SW2, #00H ;RXD3/P0.0, TXD3/P0.1*

*; RXD3\_2/P5.0, TXD3\_2/P5.1*

*SJMP \$*

*END*

### 3.2.4 Serial port 4 Switching

#### C code

//test operating frequency of

11.0592MHz #include "reg51.h"

*sfr P\_SW2 = 0xba;*

*sfr P0M1 = 0x93;*

*sfr P0M0 = 0x94;*

*sfr P1M1 = 0x91;*

*sfr P1M0 = 0x92;*

```

sfr    P2M1    =    0x95;
sfr    P2M0    =    0x96;
sfr    P3M1    =    0xb1;
sfr    P3M0    =    0xb2;
sfr    P4M1    =    0xb3;
sfr    P4M0    =    0xb4;
sfr    P5M1    =    0xc9;
sfr    P5M0    =    0xca;

```

```

void main()
{
    P0M0    =
    0x00; P0M1
    =    0x00;
    P1M0    =
    0x00; P1M1
    =    0x00;
    P2M0    =
    0x00; P2M1
    =    0x00;
    P3M0    =
    0x00; P3M1
    =    0x00;
    P4M0    =
    0x00; P4M1
    =    0x00;
    P5M0    =
    0x00; P5M1
    = 0x00;

    P_SW2 = 0x00;                //RXD4/P0.2, TXD4/P0.3
// P_SW2 = 0x04;                //RXD4_2/P5.2, TXD4_2/P5.3

    while (1);
}

```

## assembly code

Operating frequency tested at 11.0592MHz

```

P_SW2      DATA      0BAH

P0M1        DATA      093H
P0M0        DATA      094H
P1M1        DATA      091H
P1M0        DATA      092H
P2M1        DATA      095H
P2M0        DATA      096H
P3M1        DATA      0B1H
P3M0        DATA      0B2H
P4M1        DATA      0B3H
P4M0        DATA      0B4H
P5M1        DATA      0C9H
P5M0        DATA      0CAH

            ORG         0000H
            LJMP        MAIN

```

*MAIN:*

*ORG*

*MOV*

*MOV*

*MOV*

*MOV*

*MOV*

*SP, #5FH*

*P0M0, #00H*

*P0M1, #00H*

*P1M0, #00H*

*P1M1, #00H*

STC8A8K64D4 Series	Official website:	Technical Support:	Selection Consultant:
Technical Manual	<a href="http://www.STCAI.com">www.STCAI.com</a>	19864585985	13922805190
<i>MOV</i>	<i>P2M0, #00H</i>		
<i>MOV</i>	<i>P2M1, #00H</i>		
<i>MOV</i>	<i>P3M0, #00H</i>		
<i>MOV</i>	<i>P3M1, #00H</i>		
<i>MOV</i>	<i>P4M0, #00H</i>		
<i>MOV</i>	<i>P4M1, #00H</i>		
<i>MOV</i>	<i>P5M0, #00H</i>		
<i>MOV</i>	<i>P5M1, #00H</i>		
<i>MOV</i>	<i>P_SW2, #00H</i>	<i>;RXD4/P0.2, TXD4/P0.3</i>	
<i>MOV</i>	<i>P_SW2, #04H</i>	<i>;RXD4_2/P5.2, TXD4_2/P5.3</i>	
<i>SJMP</i>	<i>\$</i>		
<i>END</i>			

## 3.2.5 SPI toggle

### C code

```
//test operating frequency of
```

```
11.0592MHz #include "reg51.h"
```

```
sfr      P_SW1      = 0xa2;

sfr      P0M1       = 0x93;
sfr      P0M0       = 0x94;
sfr      P1M1       = 0x91;
sfr      P1M0       = 0x92;
sfr      P2M1       = 0x95;
sfr      P2M0       = 0x96;
sfr      P3M1       = 0xb1;
sfr      P3M0       = 0xb2;
sfr      P4M1       = 0xb3;
sfr      P4M0       = 0xb4;
sfr      P5M1       = 0xc9;
sfr      P5M0       = 0xca;
```

```
void main()
```

```
{
```

```
    P0M0 = 0x00;
    P0M1 = 0x00;
    P1M0 = 0x00;
    P1M1 = 0x00;
    P2M0 = 0x00;
    P2M1 = 0x00;
    P3M0 = 0x00;
    P3M1 = 0x00;
    P4M0 = 0x00;
    P4M1 = 0x00;
    P5M0 = 0x00;
    P5M1 = 0x00;
```

```
    P_SW1 = 0x00;
```

```
//    P_SW1 = 0x04;
```

```
//    P_SW1 = 0x08;
```

```
//SS/P1.2, MOSI/P1.3, MISO/P1.4, SCLK/P1.5
```

```
//SS_2/P2.2, MOSI_2/P2.3, MISO_2/P2.4, SCLK_2/P2.5
```

```
//SS_3/P7.4 MOSI_3/P7.5, MISO_3/P7.6, SCLK_3/P7.7
```



```
    while (1);  
}
```

## assembly code

Operating frequency tested at 11.0592MHz

```
P_SW1      DATA      0A2H  
  
P0M1      DATA      093H  
P0M0      DATA      094H  
P1M1      DATA      091H  
P1M0      DATA      092H  
P2M1      DATA      095H  
P2M0      DATA      096H  
P3M1      DATA      0B1H  
P3M0      DATA      0B2H  
P4M1      DATA      0B3H  
P4M0      DATA      0B4H  
P5M1      DATA      0C9H  
P5M0      DATA      0CAH  
  
          ORG          0000H  
          LJMP         MAIN  
  
MAIN:     ORG          0100H  
  
          MOV          SP, #5FH  
          MOV          P0M0, #00H  
          MOV          P0M1, #00H  
          MOV          P1M0, #00H  
          MOV          P1M1, #00H  
          MOV          P2M0, #00H  
          MOV          P2M1, #00H  
          MOV          P3M0, #00H  
          MOV          P3M1, #00H  
          MOV          P4M0, #00H  
          MOV          P4M1, #00H  
          MOV          P5M0, #00H  
          MOV          P5M1, #00H  
  
          MOV          P_SW1, #00H      ;SS/P1.2, MOSI/P1.3, MISO/P1.4, SCLK/P1.5  
;          MOV          P_SW1, #04H      ;SS_2/P2.2, MOSI_2/P2.3, MISO_2/P2.4, SCLK_2/P2.5  
;          MOV          P_SW1, #08H      ;SS_3/P7.4 MOSI_3/P7.5, MISO_3/P7.6, SCLK_3/P7.7  
;          MOV          P_SW1, #0CH      ;SS_4/P3.5, MOSI_4/P3.4, MISO_4/P3.3, SCLK_4/P3.2  
  
          SJMP         $  
  
          END
```

## 3.2.6 PWM toggle

### assembly code

```
P_SW2      DATA      0BAH  
PWM0CR     EQU         0FF14H
```

STC8A8K64D4 Series		Official website:	Technical Support:	Selection Consultant:
Technical Manual		<a href="http://www.STCAL.com">www.STCAL.com</a>	19864585985	13922805190
<i>PWM1CR</i>	<i>EQU</i>	<i>0FF1CH</i>		
<i>PWM2CR</i>	<i>EQU</i>	<i>0FF24H</i>		
<i>PWM3CR</i>	<i>EQU</i>	<i>0FF2CH</i>		
<i>PWM4CR</i>	<i>EQU</i>	<i>0FF34H</i>		
<i>PWM5CR</i>	<i>EQU</i>	<i>0FF3CH</i>		
<i>PWM6CR</i>	<i>EQU</i>	<i>0FF44H</i>		
<i>PWM7CR</i>	<i>EQU</i>	<i>0FF4CH</i>		
<i>P0M1</i>	<i>DATA</i>	<i>093H</i>		
<i>P0M0</i>	<i>DATA</i>	<i>094H</i>		
<i>P1M1</i>	<i>DATA</i>	<i>091H</i>		
<i>P1M0</i>	<i>DATA</i>	<i>092H</i>		
<i>P2M1</i>	<i>DATA</i>	<i>095H</i>		
<i>P2M0</i>	<i>DATA</i>	<i>096H</i>		
<i>P3M1</i>	<i>DATA</i>	<i>0B1H</i>		
<i>P3M0</i>	<i>DATA</i>	<i>0B2H</i>		
<i>P4M1</i>	<i>DATA</i>	<i>0B3H</i>		
<i>P4M0</i>	<i>DATA</i>	<i>0B4H</i>		
<i>P5M1</i>	<i>DATA</i>	<i>0C9H</i>		
<i>P5M0</i>	<i>DATA</i>	<i>0CAH</i>		
	<i>ORG</i>	<i>0000H</i>		
	<i>LJMP</i>	<i>MAIN</i>		
	<i>ORG</i>	<i>0100H</i>		
MAIN:				
	<i>MOV</i>	<i>SP, #3FH</i>		
	<i>MOV</i>	<i>P0M0, #00H</i>		
	<i>MOV</i>	<i>P0M1, #00H</i>		
	<i>MOV</i>	<i>P1M0, #00H</i>		
	<i>MOV</i>	<i>P1M1, #00H</i>		
	<i>MOV</i>	<i>P2M0, #00H</i>		
	<i>MOV</i>	<i>P2M1, #00H</i>		
	<i>MOV</i>	<i>P3M0, #00H</i>		
	<i>MOV</i>	<i>P3M1, #00H</i>		
	<i>MOV</i>	<i>P4M0, #00H</i>		
	<i>MOV</i>	<i>P4M1, #00H</i>		
	<i>MOV</i>	<i>P5M0, #00H</i>		
	<i>MOV</i>	<i>P5M1, #00H</i>		
	<i>MOV</i>	<i>P_SW2, #80H</i>		
	<i>MOV</i>	<i>A, #00H</i>		<i>;PWM0/P2.0</i>
<i>;</i>	<i>MOV</i>	<i>A, #08H</i>		<i>;PWM0_2/p1.0</i>
<i>;</i>	<i>MOV</i>	<i>A, #10H</i>		<i>;pwm0_3/p6.0</i>
	<i>MOV</i>	<i>DPTR, #PWM0CR</i>		
	<i>MOVX</i>	<i>@DPTR, A</i>		
	<i>MOV</i>	<i>A, #00H</i>		<i>;PWM1/P2.1</i>
<i>;</i>	<i>MOV</i>	<i>A, #08H</i>		<i>;pwm1_2/p1.1</i>
<i>;</i>	<i>MOV</i>	<i>A, #10H</i>		<i>;pwm1_3/p6.1</i>
	<i>MOV</i>	<i>DPTR, #PWM1CR</i>		
	<i>MOVX</i>	<i>@DPTR, A</i>		
	<i>MOV</i>	<i>A, #00H</i>		<i>;PWM2/P2.2</i>
<i>;</i>	<i>MOV</i>	<i>A, #08H</i>		<i>;pwm2_2/p1.2</i>
<i>;</i>	<i>MOV</i>	<i>A, #10H</i>		<i>;pwm2_3/p6.2</i>
	<i>MOV</i>	<i>DPTR, #PWM2CR</i>		
	<i>MOVX</i>	<i>@DPTR, A</i>		
	<i>MOV</i>	<i>A, #00H</i>		<i>;PWM3/P2.3</i>
<i>;</i>	<i>MOV</i>	<i>A, #08H</i>		<i>;pwm3_2/p1.3</i>
<i>;</i>	<i>MOV</i>	<i>A, #10H</i>		<i>;pwm3_3/p6.3</i>

STC8A8K64D4 Series Technical Manual	Official website: <a href="http://www.STC8A1.com">www.STC8A1.com</a>	Technical Support: 19864585985	Selection Consultant: 13922805190
<i>MOV</i>	<i>DPTR,#PWM3CR</i>		
<i>MOVX</i>	<i>@DPTR,A</i>		
<i>MOV</i>	<i>A,#00H</i>	<i>;PWM4/P2.4</i>	
<i>;</i>	<i>MOV</i>	<i>A,#08H</i>	<i>;pwm4_2/p1.4</i>
<i>;</i>	<i>MOV</i>	<i>A,#10H</i>	<i>;pwm4_3/p6.4</i>
<i>MOV</i>	<i>DPTR,#PWM4CR</i>		
<i>MOVX</i>	<i>@DPTR,A</i>		
<i>MOV</i>	<i>A,#00H</i>	<i>;PWM5/P2.5</i>	
<i>;</i>	<i>MOV</i>	<i>A,#08H</i>	<i>;pwm5_2/p1.5</i>
<i>;</i>	<i>MOV</i>	<i>A,#10H</i>	<i>;pwm5_3/p6.5</i>
<i>MOV</i>	<i>DPTR,#PWM5CR</i>		
<i>MOVX</i>	<i>@DPTR,A</i>		
<i>MOV</i>	<i>A,#00H</i>	<i>;PWM6/P2.6</i>	
<i>;</i>	<i>MOV</i>	<i>A,#08H</i>	<i>;pwm6_2/p1.6</i>
<i>;</i>	<i>MOV</i>	<i>A,#10H</i>	<i>;pwm6_3/p6.6</i>
<i>MOV</i>	<i>DPTR,#PWM6CR</i>		
<i>MOVX</i>	<i>@DPTR,A</i>		
<i>MOV</i>	<i>A,#00H</i>	<i>;PWM7/P2.7</i>	
<i>;</i>	<i>MOV</i>	<i>A,#08H</i>	<i>;pwm7_2/p1.7</i>
<i>;</i>	<i>MOV</i>	<i>A,#10H</i>	<i>;pwm7_3/p6.7</i>
<i>MOV</i>	<i>DPTR,#PWM7CR</i>		
<i>MOVX</i>	<i>@DPTR,A</i>		
<i>MOV</i>	<i>P_SW2,#00H</i>		
<i>SJMP</i>	<i>\$</i>		
<i>END</i>			

## C code

```
#include "reg51.h"
```

```
#define PWM0CR (*(unsigned char volatile xdata *)0xff14)
#define PWM1CR (*(unsigned char volatile xdata *)0xff1c)
#define PWM2CR (*(unsigned char volatile xdata *)0xff24)
#define PWM3CR (*(unsigned char volatile xdata *)0xff2c)
#define PWM4CR (*(unsigned char volatile xdata *)0xff34)
#define PWM5CR (*(unsigned char volatile xdata *)0xff3c)
#define PWM6CR (*(unsigned char volatile xdata *)0xff44)
#define PWM7CR (*(unsigned char volatile xdata *)0xff4c)
```

```
sfr P_SW2 = 0xba;
```

```
sfr P0M1 = 0x93;
```

```
sfr P0M0 = 0x94;
```

```
sfr P1M1 = 0x91;
```

```
sfr P1M0 = 0x92;
```

```
sfr P2M1 = 0x95;
```

```
sfr P2M0 = 0x96;
```

```
sfr P3M1 = 0xb1;
```

```
sfr P3M0 = 0xb2;
```

```
sfr P4M1 = 0xb3;
```

```
sfr P4M0 = 0xb4;
```

```
sfr P5M1 = 0xc9;
```

```
sfr P5M0 = 0xca;
```

```
void main()
```

```
{
```

```
    P0M0 = 0x00;
```

```

P0M1 = 0x00;
P1M0 = 0x00;
P1M1 = 0x00;
P2M0 = 0x00;
P2M1 = 0x00;
P3M0 = 0x00;
P3M1 = 0x00;
P4M0 = 0x00;
P4M1 = 0x00;
P5M0 = 0x00;
P5M1 = 0x00;

P_SW2 = 0x80;
PWM0CR = 0x00;           //PWM0/P2.0
//PWM0CR = 0x08;         //PWM0_2/P1.0
//PWM0CR = 0x10;         //PWM0_3/P6.0
PWM1CR = 0x00;           //PWM1/P2.1
//PWM1CR = 0x08;         //PWM1_2/P1.1
//PWM1CR = 0x10;         //PWM1_3/P6.1
PWM2CR = 0x00;           //PWM2/P2.2
//PWM2CR = 0x08;         //PWM2_2/P1.2
//PWM2CR = 0x10;         //PWM2_3/P6.2
PWM3CR = 0x00;           //PWM3/P2.3
//PWM3CR = 0x08;         //PWM3_2/P1.3
//PWM3CR = 0x10;         //PWM3_3/P6.3
PWM4CR = 0x00;           //PWM4/P2.4
//PWM4CR = 0x08;         //PWM4_2/P1.4
//PWM4CR = 0x10;         //PWM4_3/P6.4
PWM5CR = 0x00;           //PWM5/P2.5
//PWM5CR = 0x08;         //PWM5_2/P1.5
//PWM5CR = 0x10;         //PWM5_3/P6.5
PWM6CR = 0x00;           //PWM6/P2.6
//PWM6CR = 0x08;         //PWM6_2/P1.6
//PWM6CR = 0x10;         //PWM6_3/P6.6
PWM7CR = 0x00;           //PWM7/P2.7
//PWM7CR = 0x08;         //PWM7_2/P1.7
//PWM7CR = 0x10;         //PWM7_3/P6.7
P_SW2 = 0x00;

while (1);
}

```

## 3.2.7 PCA/CCP/PWM Switching

### assembly code

*P\_SW1 DATA 0A2H*

```

P0M1 DATA 093H P0M0
DATA 094H P1M1 DATA
091H P1M0 DATA 092H
P2M1 DATA 095H P2M0
DATA 096H P3M1 DATA
0B1H P3M0 DATA 0B2H
P4M1 DATA 0B3H P4M0
DATA 0B4H P5M1 DATA
0C9H

```

*P5M0 DATA 0CAH*

*ORG 0000H*

*LJMP MAIN*

*MAIN: ORG 0100H*

*MOV SP, #3FH*

*MOV P0M0, #00H*

*MOV P0M1, #00H*

*MOV P1M0, #00H*

*MOV P1M1, #00H*

*MOV P2M0, #00H*

*MOV P2M1, #00H*

*MOV P3M0, #00H*

*MOV P3M1, #00H*

*MOV P4M0, #00H*

*MOV P4M1, #00H*

*MOV P5M0, #00H*

*MOV P5M1, #00H*

*MOV P\_SW1, #00H ;ECI/P1.2, CCP0/P1.7, CCP1/P1.6, CCP2/P1.5, CCP3/P1.4*  
*; MOV P\_SW1, #10H ;eci\_2/p2.2, ccp0\_2/p2.3, ccp1\_2/p2.4, ccp2\_2/p2.5, ccp3\_2/p2.6*  
*; MOV P\_SW1, #20H ;eci\_3/p7.4, ccp0\_3/p7.0, ccp1\_3/p7.1, ccp2\_3/p7.2, ccp3\_3/p7.3*  
*; MOV P\_SW1, #30H ;eci\_4/p3.5, ccp0\_4/p3.3, ccp1\_4/p3.2, ccp2\_4/p3.1, ccp3\_4/p3.0*

*SJMP \$*

*END*

## C code

*#include "reg51.h"*

*sfr P\_SW1 = 0xa2;*  
  
*sfr P0M1 = 0x93;*  
*sfr P0M0 = 0x94;*  
*sfr P1M1 = 0x91;*  
*sfr P1M0 = 0x92;*  
*sfr P2M1 = 0x95;*  
*sfr P2M0 = 0x96;*  
*sfr P3M1 = 0xb1;*  
*sfr P3M0 = 0xb2;*  
*sfr P4M1 = 0xb3;*  
*sfr P4M0 = 0xb4;*  
*sfr P5M1 = 0xc9;*  
*sfr P5M0 = 0xca;*

*void main()*

*{*  
*P0M0 = 0x00;*  
*P0M1 = 0x00;*  
*P1M0 = 0x00;*  
*P1M1 = 0x00;*  
*P2M0 = 0x00;*  
*P2M1 = 0x00;*  
*P3M0 = 0x00;*  
*P3M1 = 0x00;*  
*P4M0 = 0x00;*  
*P4M1 = 0x00;*  
*P5M0 = 0x00;*  
*P5M1 = 0x00;*  
*}*

```

P4M0 = 0x00;
P4M1 = 0x00;
P5M0 = 0x00;
P5M1 = 0x00;

P_SW1 = 0x00; //ECI/P1.2, CCP0/P1.7, CCP1/P1.6, CCP2/P1.5, CCP3/P1.4
// P_SW1 = 0x10; //ECI_2/P2.2, CCP0_2/P2.3, CCP1_2/P2.4, CCP2_2/P2.5, CCP3_2/P2.6
// P_SW1 = 0x20; //ECI_3/P7.4, CCP0_3/P7.0, CCP1_3/P7.1, CCP2_3/P7.2, CCP3_3/P7.3
// P_SW1 = 0x30; //eci_4/p3.5, ccp0_4/p3.3, ccp1_4/p3.2, ccp2_4/p3.1, ccp3_4/p3.0

while (1);
}

```

## 3.2.8 I2C Switching

### C code

```
//test operating frequency of
```

```
11.0592MHz #include "reg51.h"
```

```

sfr      P_SW2      = 0xba;

sfr      P0M1      = 0x93;
sfr      P0M0      = 0x94;
sfr      P1M1      = 0x91;
sfr      P1M0      = 0x92;
sfr      P2M1      = 0x95;
sfr      P2M0      = 0x96;
sfr      P3M1      = 0xb1;
sfr      P3M0      = 0xb2;
sfr      P4M1      = 0xb3;
sfr      P4M0      = 0xb4;
sfr      P5M1      = 0xc9;
sfr      P5M0      = 0xca;

```

```

void main()
{
    P0M0 =
    0x00; P0M1
    = 0x00;
    P1M0 =
    0x00; P1M1
    = 0x00;
    P2M0 =
    0x00; P2M1
    = 0x00;
    P3M0 =
    0x00; P3M1
    = 0x00;
    P4M0 =
    0x00; P4M1
    = 0x00;
    P5M0 =
    0x00; P5M1
    = 0x00;
}

```

STC8A8K64D4 Series	Official website:	Technical Support:	Selection Consultant:
Technical Manual	<a href="http://www.STCAL.com">www.STCAL.com</a>	19864585985	13922805190
<i>P_SW2 = 0x00;</i>		<i>//SCL/P1.5, SDA/P1.4</i>	
<i>// P_SW2 = 0x10;</i>		<i>//SCL_2/P2.5, SDA_2/P2.4</i>	
<i>// P_SW2 = 0x20;</i>		<i>//SCL_3/P7.7, SDA_3/P7.6</i>	
<i>// P_SW2 = 0x30;</i>		<i>//SCL_4/P3.2, SDA_4/P3.3</i>	
<i>while (1);</i>			

**assembly code**

Operating frequency tested at 11.0592MHz

```

P_SW2      DATA      0BAH

P0M1        DATA      093H
P0M0        DATA      094H
P1M1        DATA      091H
P1M0        DATA      092H
P2M1        DATA      095H
P2M0        DATA      096H
P3M1        DATA      0B1H
P3M0        DATA      0B2H
P4M1        DATA      0B3H
P4M0        DATA      0B4H
P5M1        DATA      0C9H
P5M0        DATA      0CAH

            ORG          0000H
            LJMP         MAIN

MAIN:       ORG          0100H

            MOV          SP, #5FH
            MOV          P0M0, #00H
            MOV          P0M1, #00H
            MOV          P1M0, #00H
            MOV          P1M1, #00H
            MOV          P2M0, #00H
            MOV          P2M1, #00H
            MOV          P3M0, #00H
            MOV          P3M1, #00H
            MOV          P4M0, #00H
            MOV          P4M1, #00H
            MOV          P5M0, #00H
            MOV          P5M1, #00H

            MOV          P_SW2, #00H      ;SCL/P1.5, SDA/P1.4
;            MOV          P_SW2, #10H     ;SCL_2/P2.5, SDA_2/P2.4
;            MOV          P_SW2, #20H     ;SCL_3/P7.7, SDA_3/P7.6
;            MOV          P_SW2, #30H     ;SCL_4/P3.2, SDA_4/P3.3

            SJMP         $

            END

```

**3.2.9 Comparator output switching****C code**

// Tested operating frequency of 11.0592MHz

#include "reg51.h"



```

sfr      P_SW2      = 0xba;

sfr      P0M1      = 0x93;
sfr      P0M0      = 0x94;
sfr      P1M1      = 0x91;
sfr      P1M0      = 0x92;
sfr      P2M1      = 0x95;
sfr      P2M0      = 0x96;
sfr      P3M1      = 0xb1;
sfr      P3M0      = 0xb2;
sfr      P4M1      = 0xb3;
sfr      P4M0      = 0xb4;
sfr      P5M1      = 0xc9;
sfr      P5M0      = 0xca;

```

```

void main()
{
    P0M0      =
    0x00; P0M1
    = 0x00;
    P1M0      =
    0x00; P1M1
    = 0x00;
    P2M0      =
    0x00; P2M1
    = 0x00;
    P3M0      =
    0x00; P3M1
    = 0x00;
    P4M0      =
    0x00; P4M1
    = 0x00;
    P5M0      =
    0x00; P5M1
    = 0x00;

    P_SW2 = 0x00; //CMPO/P3.4
// P_SW2 = 0x08; //CMPO_2/P4.1

    while (1);
}

```

## assembly code

Operating frequency tested at 11.0592MHz

P_SW2	DATA	0BAH
P0M1	DATA	093H
P0M0	DATA	094H
P1M1	DATA	091H
P1M0	DATA	092H
P2M1	DATA	095H
P2M0	DATA	096H
P3M1	DATA	0B1H
P3M0	DATA	0B2H
P4M1	DATA	0B3H
P4M0	DATA	0B4H
P5M1	DATA	0C9H
P5M0	DATA	0CAH

STC8A8K64D4 Series  
Technical Manual

*ORG*  
*LJMP*  
*ORG*

Official website:  
[www.STCAL.com](http://www.STCAL.com)  
*MAIN*  
*0100H*

Technical Support:  
19864585985

Selection Consultant:  
13922805190

MAIN:

```

MOV    SP, #5FH
MOV    P0M0, #00H
MOV    P0M1, #00H
MOV    P1M0, #00H
MOV    P1M1, #00H
MOV    P2M0, #00H
MOV    P2M1, #00H
MOV    P3M0, #00H
MOV    P3M1, #00H
MOV    P4M0, #00H
MOV    P4M1, #00H
MOV    P5M0, #00H
MOV    P5M1, #00H

```

```

MOV    P_SW2, #00H      ;CMPO/P3.4
;      MOV    P_SW2, #08H      ;CMPO_2/P4.1

SJMP    $

```

```

END

```

### 3.2.10 Master clock output switching

#### C code

```

//test operating frequency of

```

```

11.0592MHz #include "reg51.h"

```

```

#define    MCLKOCR    (*(unsigned char volatile xdata *)0xfe00)

```

```

sfr    P_SW2    =    0xba;

```

```

sfr    P0M1    =    0x93;

```

```

sfr    P0M0    =    0x94;

```

```

sfr    P1M1    =    0x91;

```

```

sfr    P1M0    =    0x92;

```

```

sfr    P2M1    =    0x95;

```

```

sfr    P2M0    =    0x96;

```

```

sfr    P3M1    =    0xb1;

```

```

sfr    P3M0    =    0xb2;

```

```

sfr    P4M1    =    0xb3;

```

```

sfr    P4M0    =    0xb4;

```

```

sfr    P5M1    =    0xc9;

```

```

sfr    P5M0    =    0xca;

```

```

void main()

```

```

{

```

```

    P0M0 = 0x00;

```

```

    P0M1 = 0x00;

```

```

    P1M0 = 0x00;

```

```

    P1M1 = 0x00;

```

```

    P2M0 = 0x00;

```

```

    P2M1 = 0x00;

```

```

    P3M0 = 0x00;

```

```

    P3M1 = 0x00;

```

```

P4M0      =
0x00; P4M1
=      0x00;
P5M0      =
0x00; P5M1
= 0x00;

P_SW2 = 0x80;
MCLKOCR = 0x04;           //HIRC/4 output via MCLKO/P5.4
// MCLKOCR = 0x84;       //HIRC/4 output via MCLKO_2/P1.6
P_SW2 = 0x00;

while (1);
}

```

## assembly code

Operating frequency tested at 11.0592MHz

```

P_SW2      DATA      0BAH

MCLKOCR     EQU        0FE05H

P0M1        DATA      093H
P0M0        DATA      094H
P1M1        DATA      091H
P1M0        DATA      092H
P2M1        DATA      095H
P2M0        DATA      096H
P3M1        DATA      0B1H
P3M0        DATA      0B2H
P4M1        DATA      0B3H
P4M0        DATA      0B4H
P5M1        DATA      0C9H
P5M0        DATA      0CAH

                ORG      0000H
                LJMP     MAIN

                ORG      0100H
MAIN:
                MOV      SP, #5FH
                MOV      P0M0, #00H
                MOV      P0M1, #00H
                MOV      P1M0, #00H
                MOV      P1M1, #00H
                MOV      P2M0, #00H
                MOV      P2M1, #00H
                MOV      P3M0, #00H
                MOV      P3M1, #00H
                MOV      P4M0, #00H
                MOV      P4M1, #00H
                MOV      P5M0, #00H
                MOV      P5M1, #00H

                MOV      P_SW2, #80H
                MOV      A, #04H           ;HIRC/4 output via MCLKO/P5.4
;                MOV      A, #84H       ;HIRC/4 output via MCLKO_2/P1.6
                MOV      DPTR, #MCLKOCR
                MOVX     @DPTR, A

```



*SJMP*

*\$*

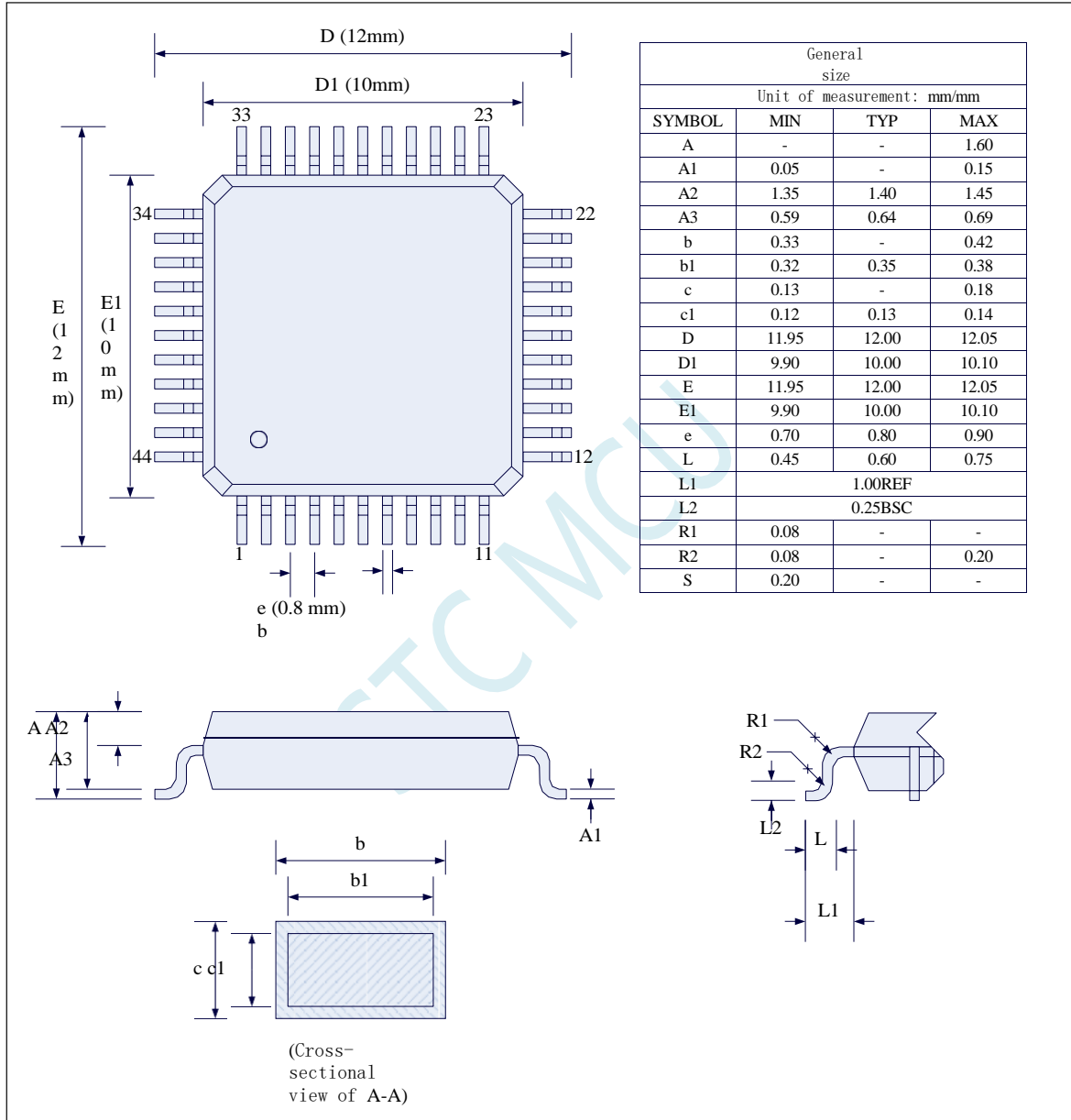
*END*

---

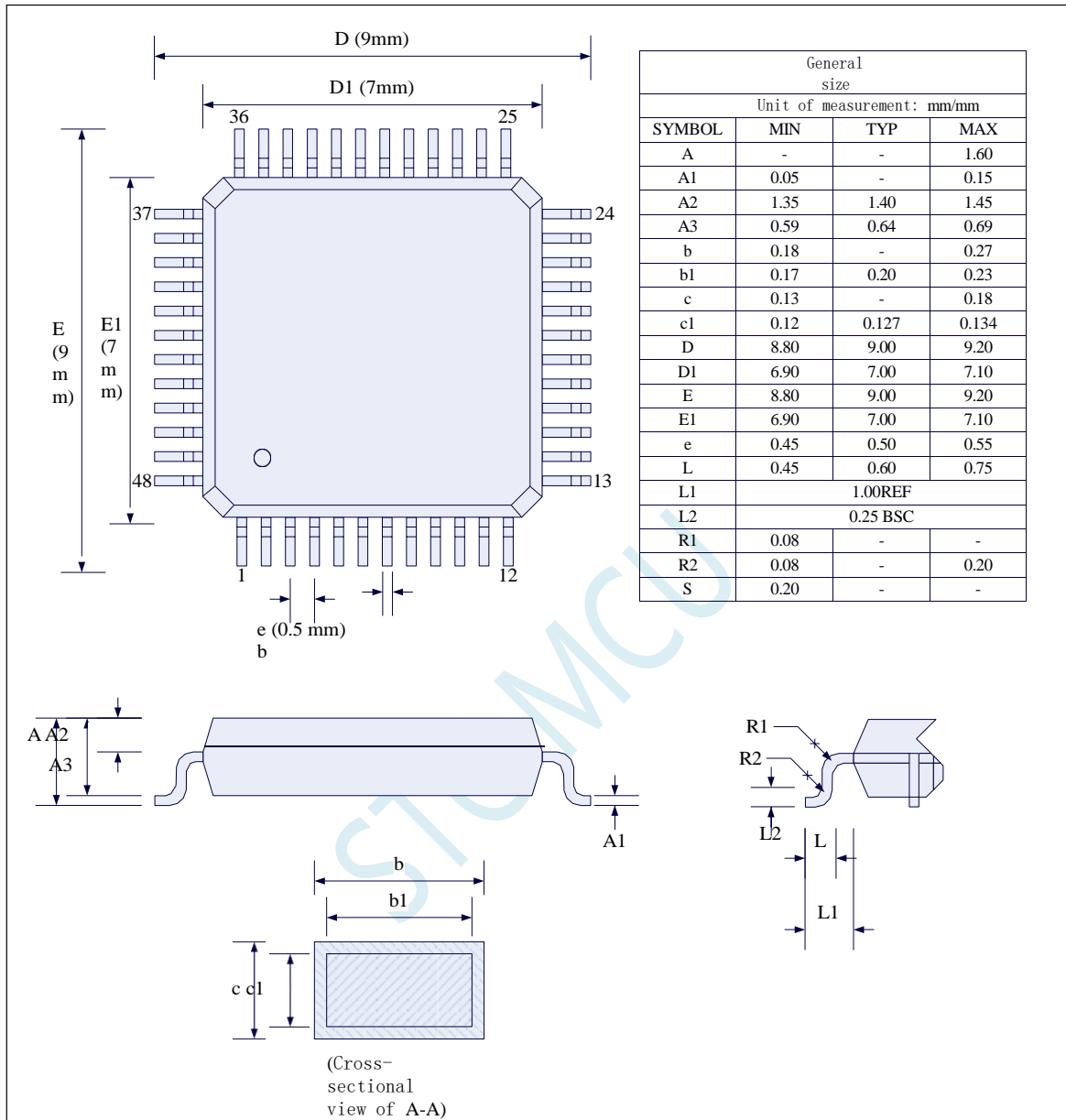
STC MCU

## 4 Package size Figure

### 4.1 LQFP44 package size diagram (12mm\*12mm )

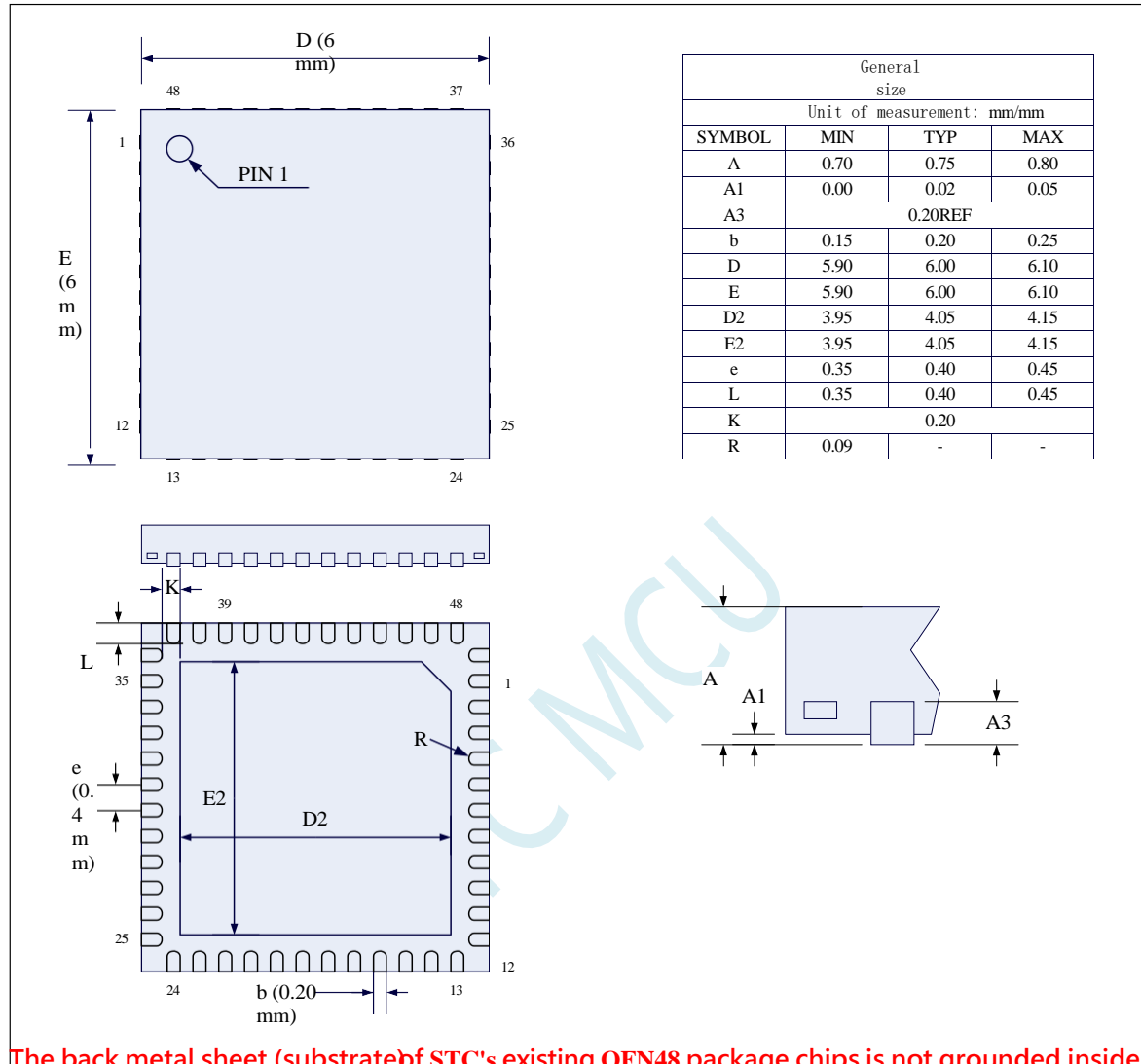


## 4.2 LQFP48 package size diagram (9mm\*9mm)



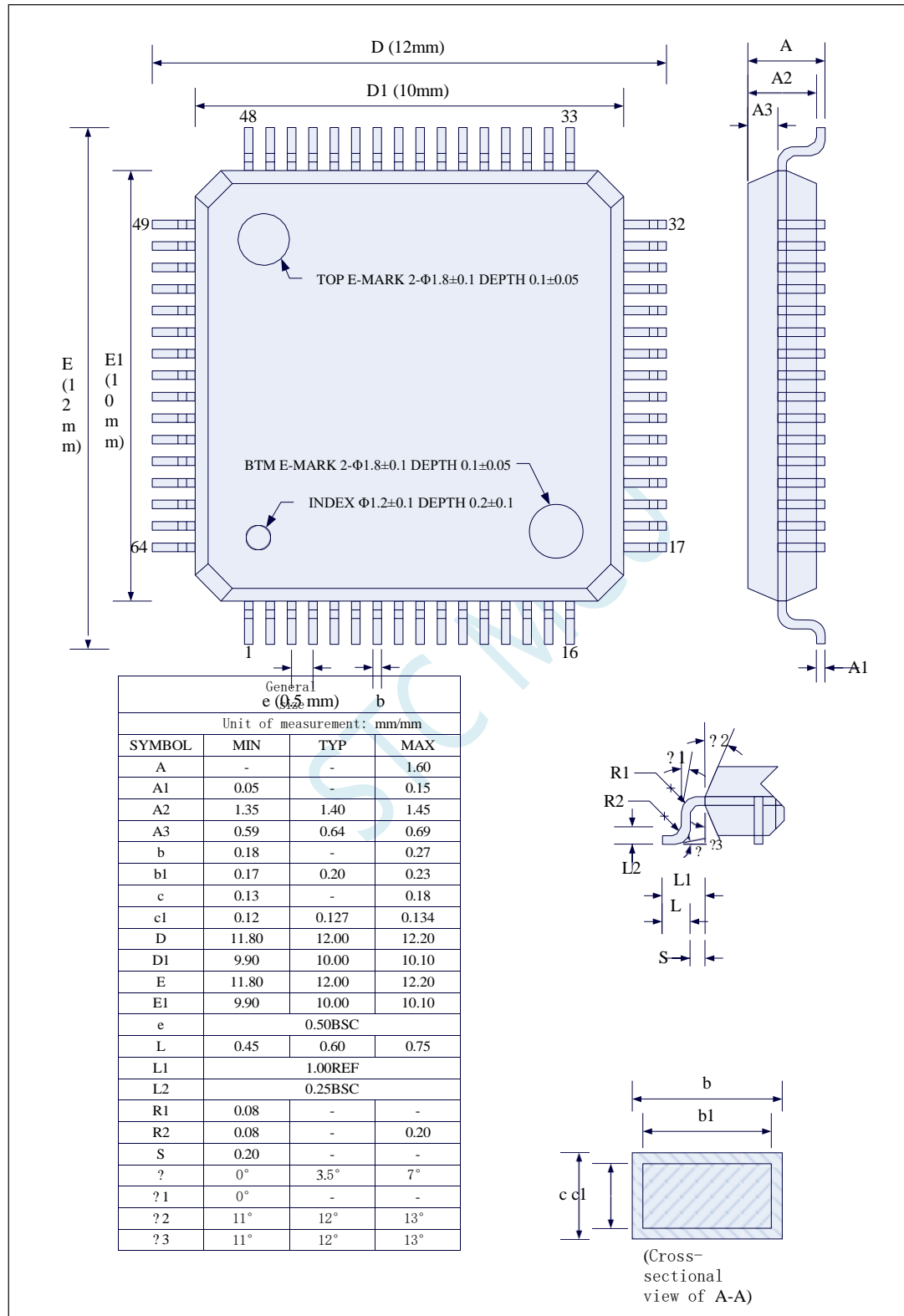


## 4.3 QFN48 package size diagram (6mm\*6mm)

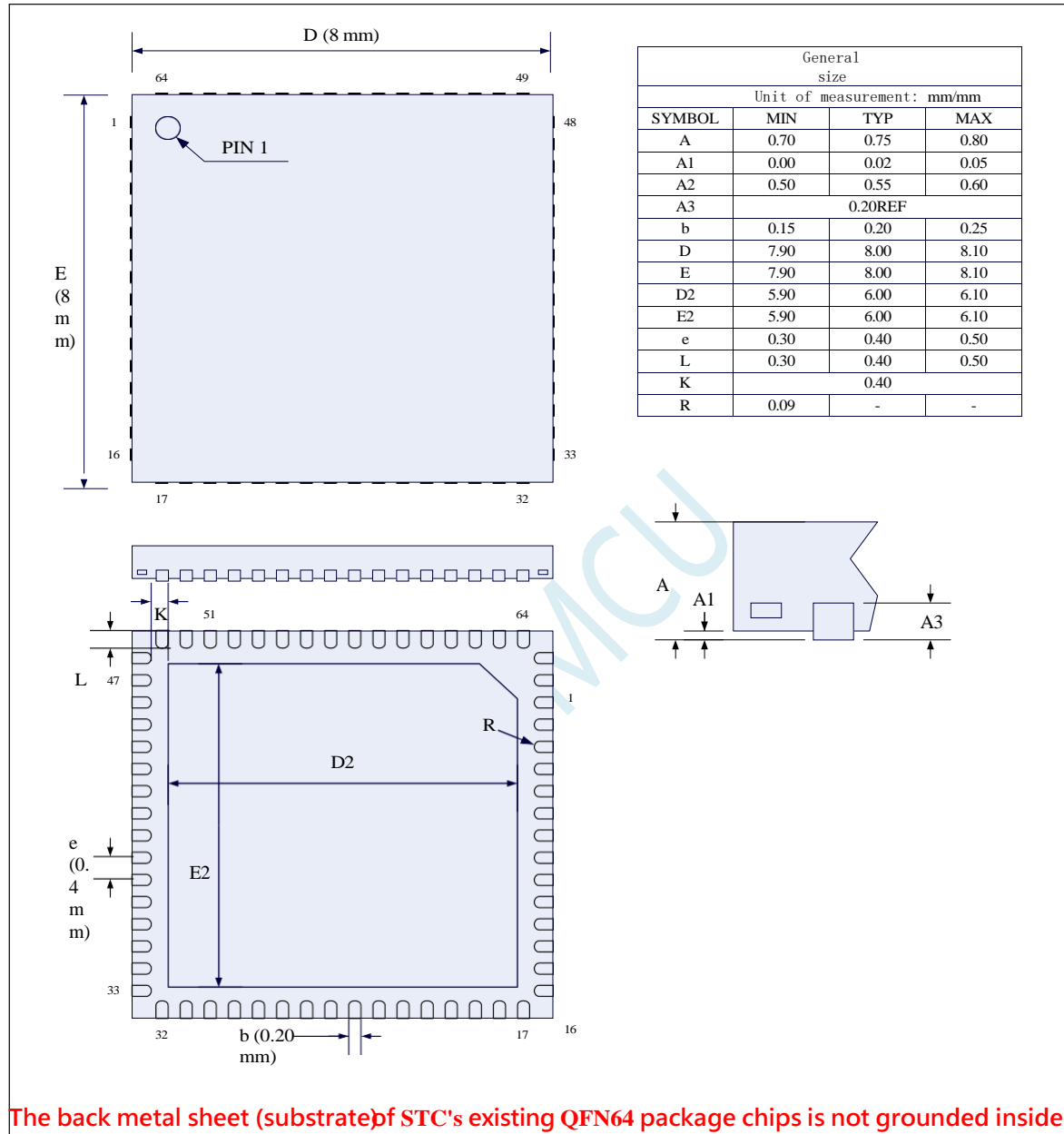


The back metal sheet (substrate) of STC's existing QFN48 package chips is not grounded inside the chip and may or may not be grounded on the user's PCB board without affecting chip performance

## 4.4 LQFP64 package size diagram (12mm\*12mm )

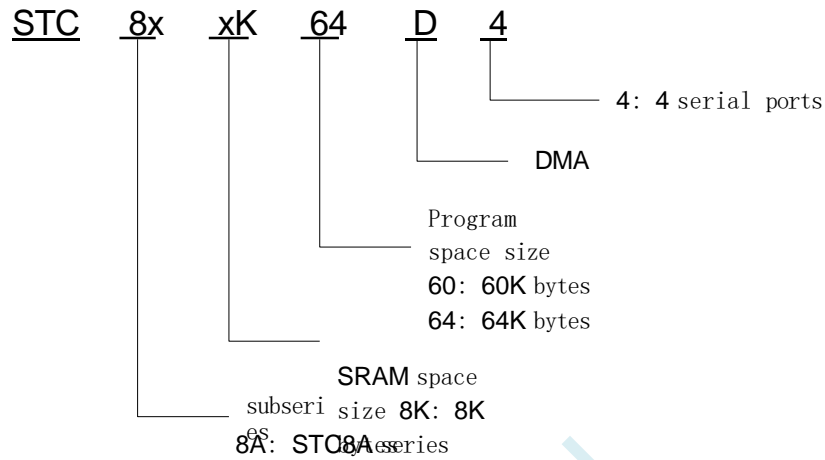


## 4.5 QFN64 package size diagram (8mm\*8mm)



The back metal sheet (substrate) of STC's existing QFN64 package chips is not grounded inside the chip and may or may not be grounded on the user's PCB board without affecting chip performance

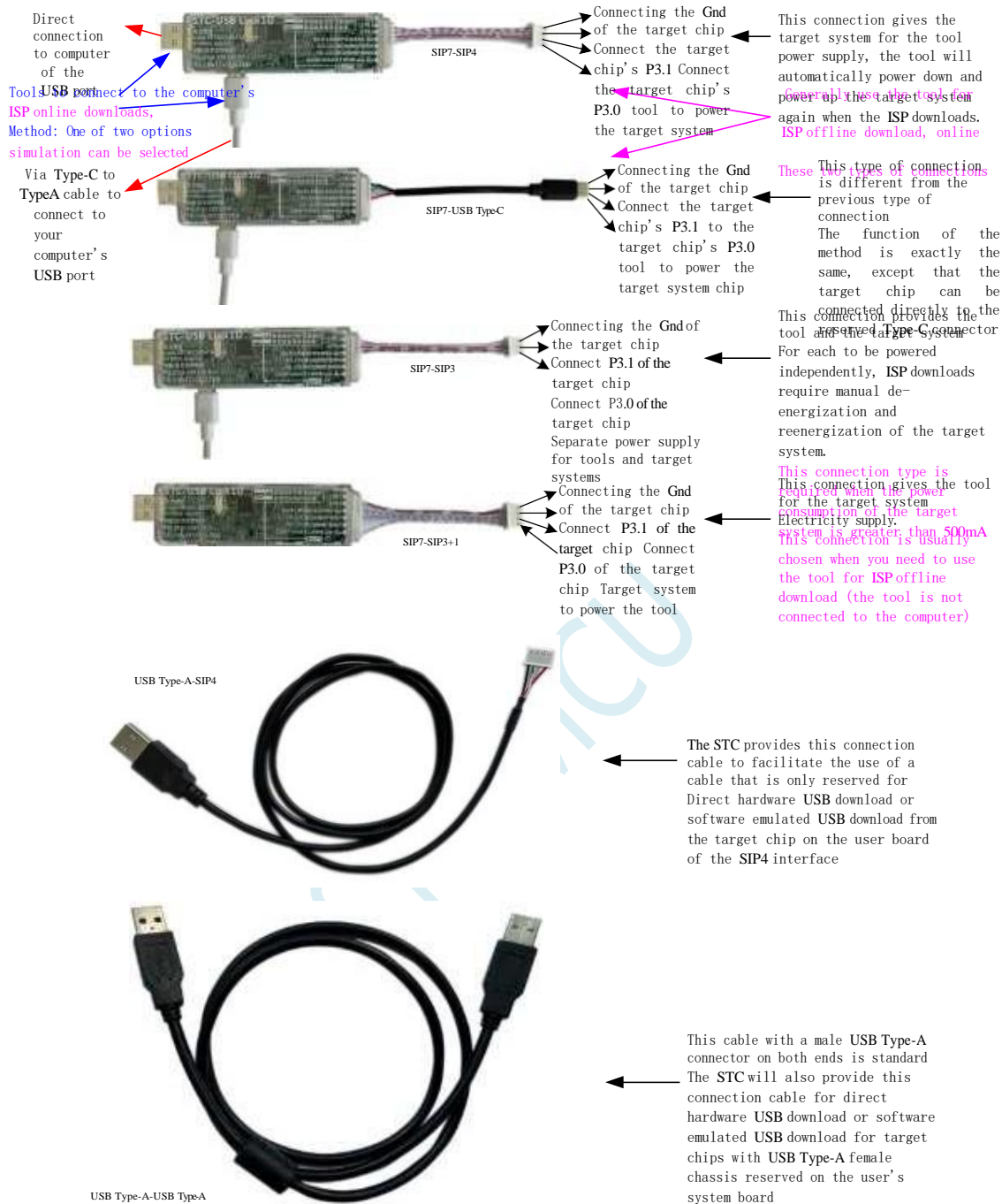
## 4.6 STC8A8K64D4 Series Microcontroller Naming Rules



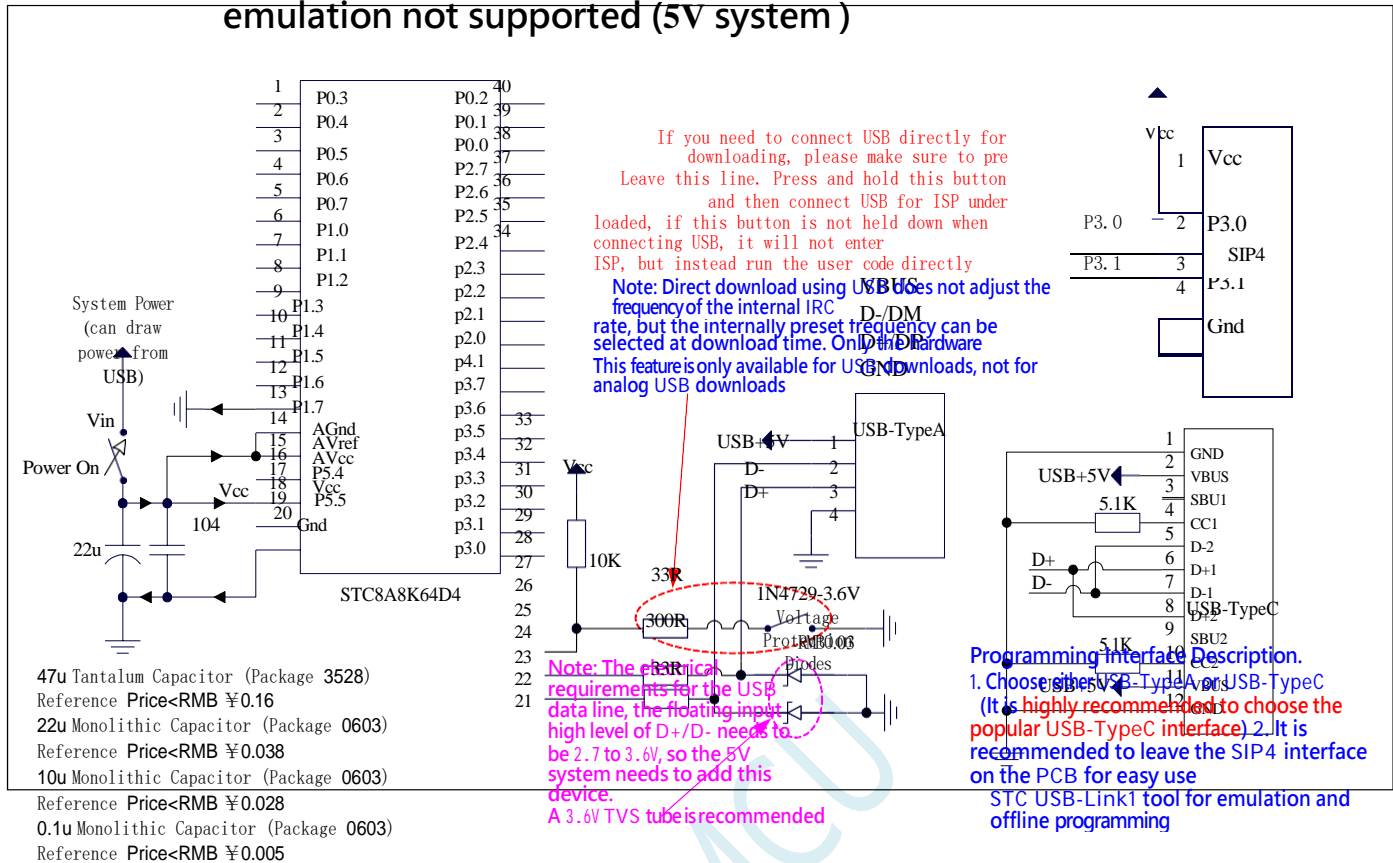


**Note:** It has been found that when using the USB cable for ISP downloading, the voltage drop on the USB cable is too thin, resulting in

The power supply is insufficient for ISP downloading, so be sure to use a USB-enhanced cable when using a USB cable for ISP downloading.

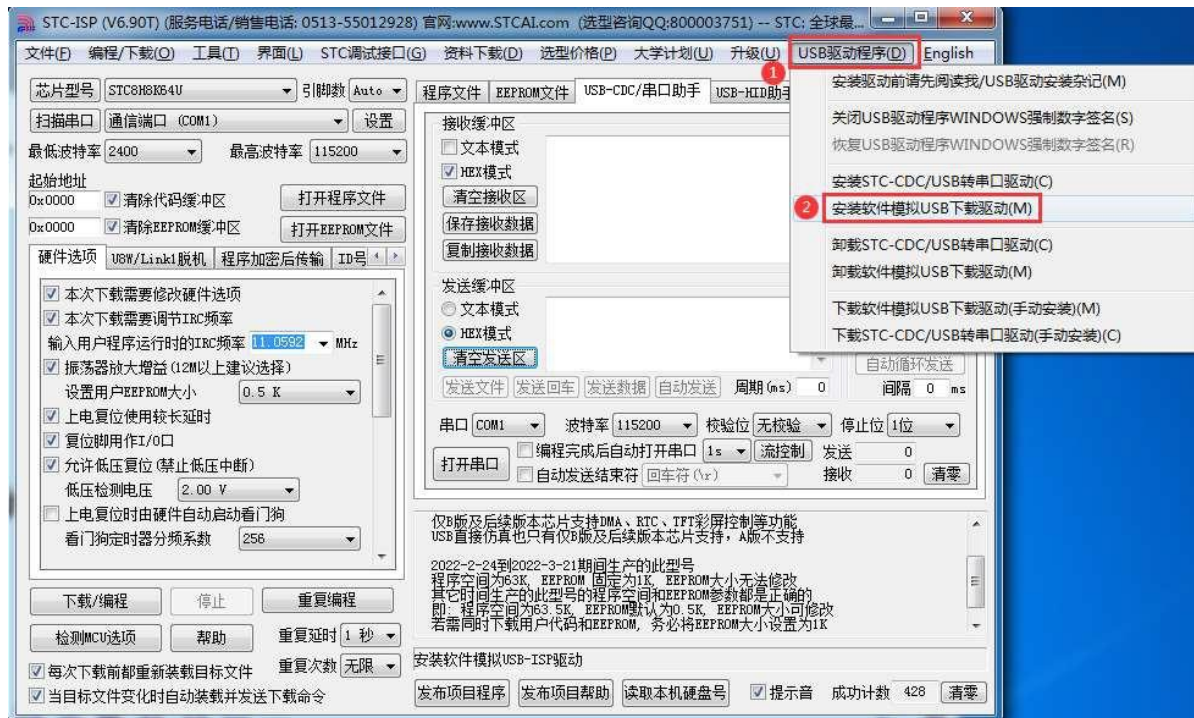


## 5.1.2 Software emulation USB direct ISP download, recommended to try, emulation not supported (5V system)



Nowadays, STC8G/STC8H MCUs without hardware USB basically support software emulation of USB downloading user programs, because they use the USB-SCAN communication protocol, regardless of any version of the operating system, you have to install the driver. Install the driver in the place shown below in the STC-ISP download software.

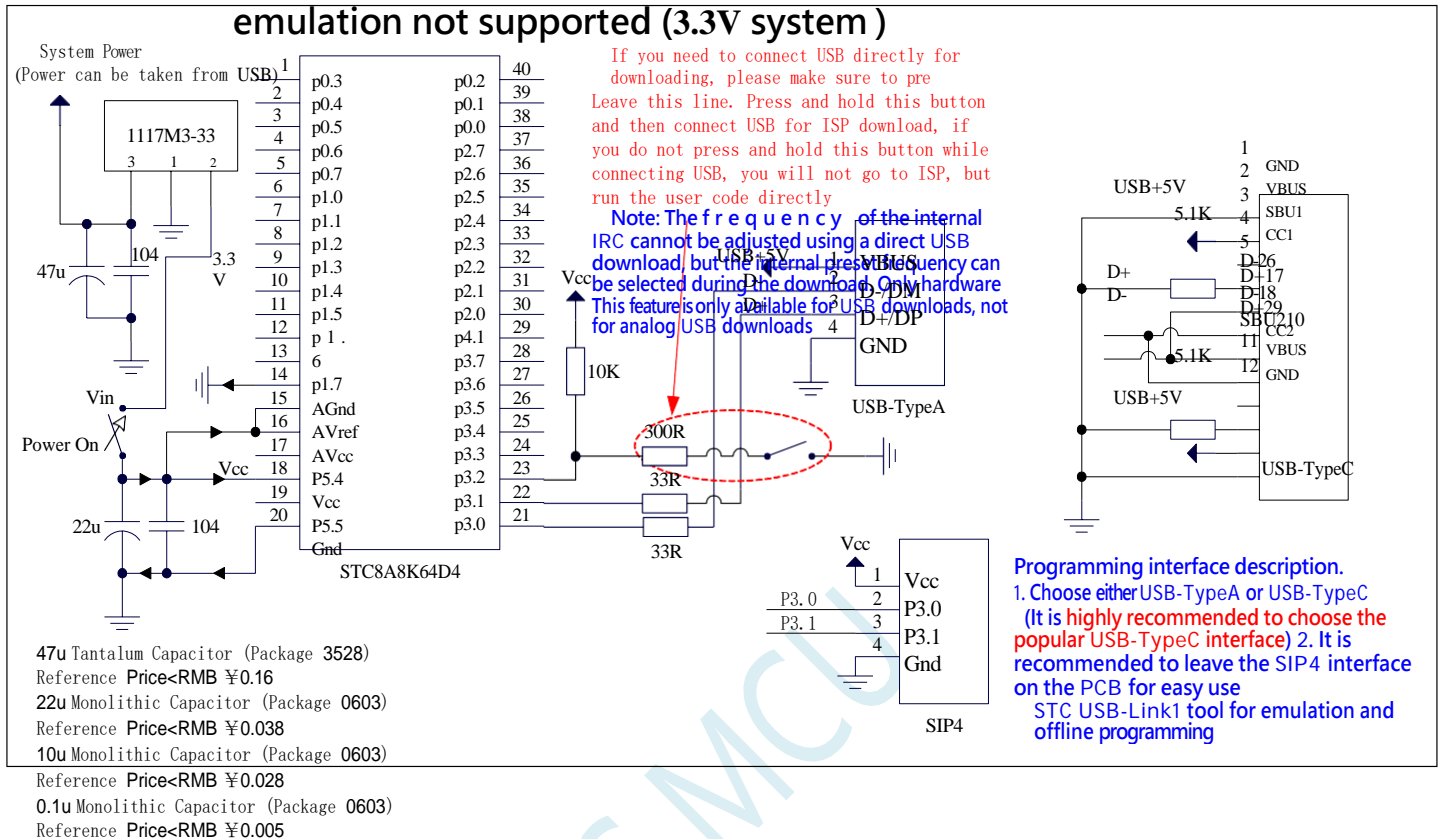




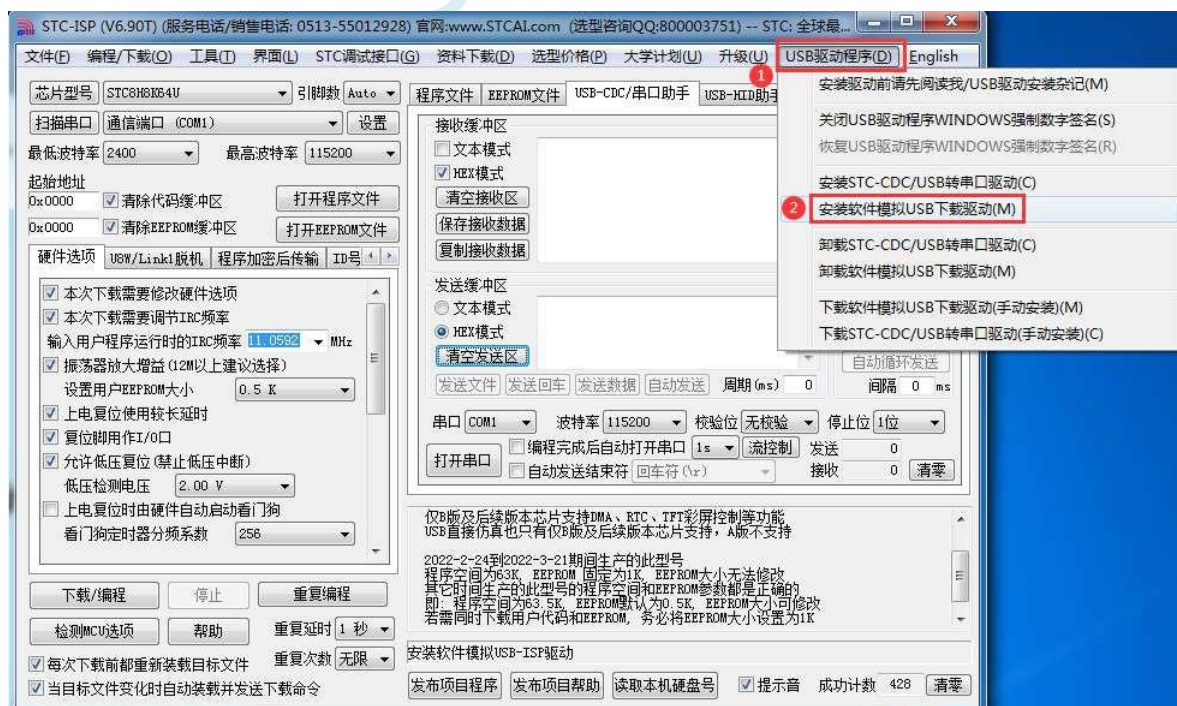
ISP Download steps.

1. D-/P3.0, D+/P3.1 connected to PC-USB port
2. the P3.2 and GND short, the experiment box board P3.2/INT0 button press
3. Re-energize the target chip. If the target chip has been powered down, it can be powered up directly; if the target chip is powered up, it needs to be powered down and powered up again (cold start) Wait for the "STC USB Writer (HID1)" to be recognized automatically in the STC-ISP download software, then it is not related to the P3.2 status (at this time, you don't need to keep pressing the P3.2 port, it doesn't matter if you are tired of pressing the handle all the time, it will be a problem if you press the button)
4. click the "Download/Program" button in the download software (Note: the order of operation of USB download and serial download is different, **do not** (Click the Download button first, and wait until the computer recognizes the "STC USB Writer (HID1)" device before clicking the Download button to start downloading)

### 5.1.3 Software emulation USB direct ISP download, recommended to try, emulation not supported (3.3V system)



Nowadays, STC8G/STC8H MCUs without hardware USB basically support software emulation of USB downloading user programs, because they use the USB-SCAN communication protocol, regardless of any version of the operating system, you have to install the driver. Install the driver in the place shown below in the STC-ISP download software.



## ISP Download steps.

1. D-/P3.0, D+/P3.1 connected to PC-USB port
2. the P3.2 and GND short, the experiment box board P3.2/INT0 button press
3. Re-energize the target chip. If the target chip has been powered down, it can be powered up directly; if the target chip is powered up, it needs to be powered down and powered up again (cold start). Wait for the "STC USB Writer (HID1)" to be recognized automatically in the STC-ISP download software, then it is not related to the P3.2 status (at this time, you don't need to keep pressing the P3.2 port, it doesn't matter if you are tired of pressing the handle all the time, it will be a problem if you press the button)
4. click the "Download/Program" button in the download software (Note: the order of operation of USB download and serial download is different, **do not** (Click the Download button first, and wait until the computer recognizes the "STC USB Writer (HID1)" device before clicking the Download button to start downloading)

STC MCU

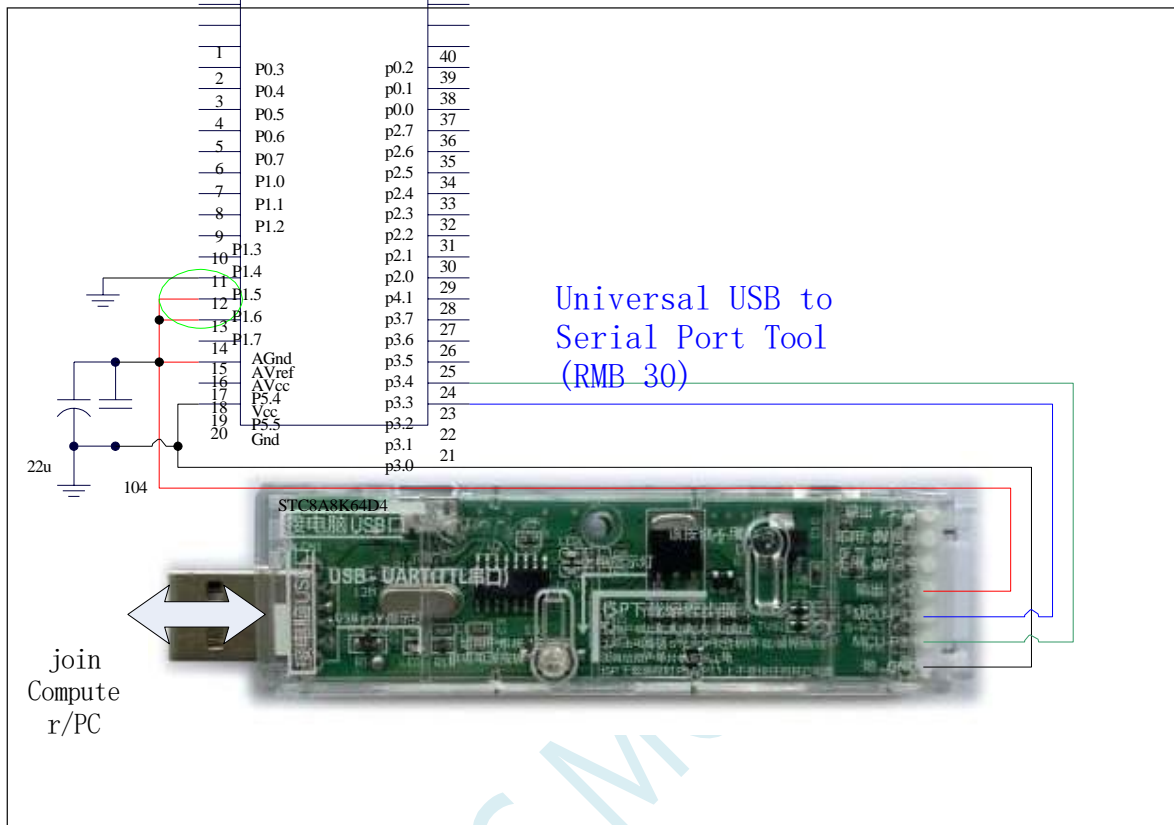








### 5.1.7 Download using the Universal USB to Serial Tool, which supports ISP online downloads and can also support emulation

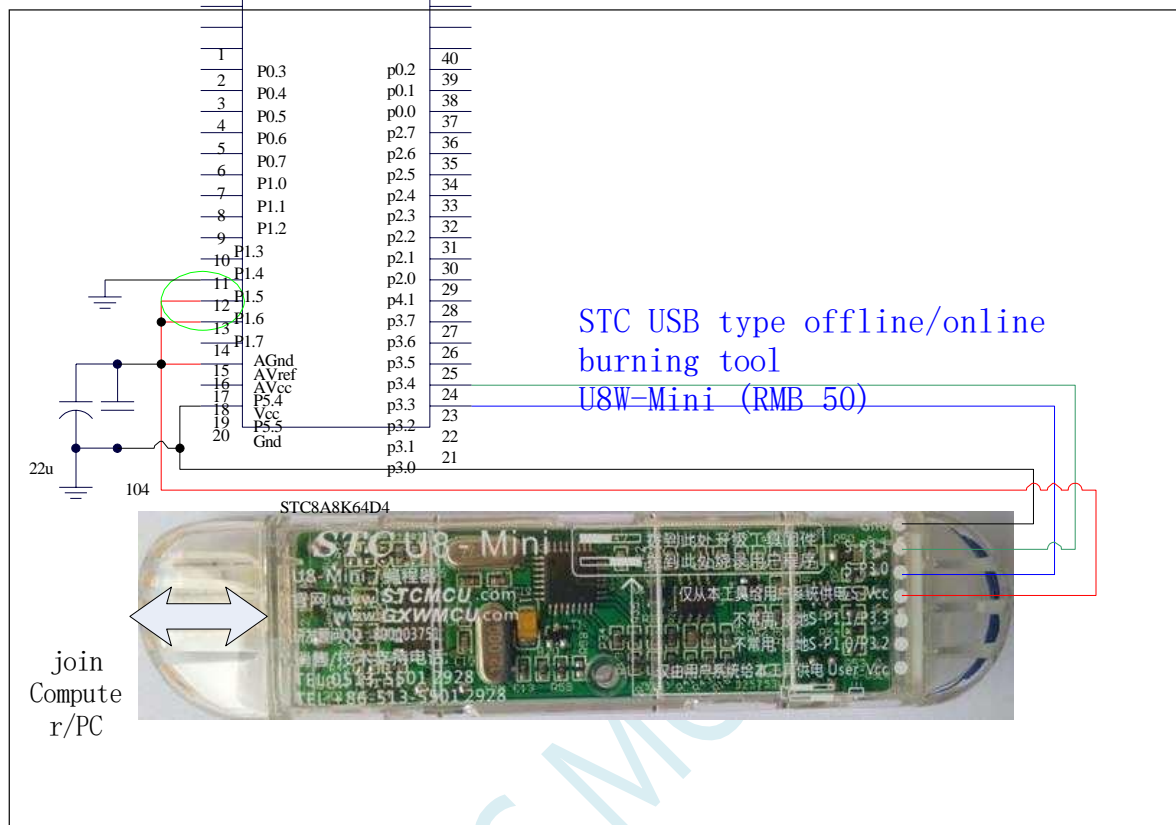


#### ISP Download steps.

1. Connect the Universal USB to Serial Tool to the target chip as shown in the diagram
2. Press the power button to make sure that the target chip is in a **powered-down state** (a state where the power-up indicator is off) **Note: The tool is not powered externally the first time it is powered up, so you can skip this step if you are using this tool for the first time.**
3. Click the "Download/Program" button in the STC-ISP download software
4. Press the power button again to power up the target chip (the power-on indicator is on)
5. Start ISP Download

**Note: It has been found that when using the USB cable for ISP downloading, the voltage drop on the USB cable is too thin, resulting in insufficient power supply during ISP downloading.**

## 5.1.8 Download using the U8-Mini tool, which supports ISP online and offline downloads, and also supports emulation



### ISP Download steps.

1. Connect the U8-Mini to the target chip as shown in the diagram
2. Click the "Download/Program" button in the STC-ISP download software
3. Start ISP Download

Note: If you use the U8-Mini to power the target system, the total current of the target system must not be greater than 200mA, otherwise the download will fail. **Note: Currently, it has been found that when using the USB cable to power the ISP download, the voltage drop on the USB cable is too thin, resulting in insufficient power supply during the ISP download, so please be sure to use a USB reinforced cable when using the USB cable to power the ISP download.**

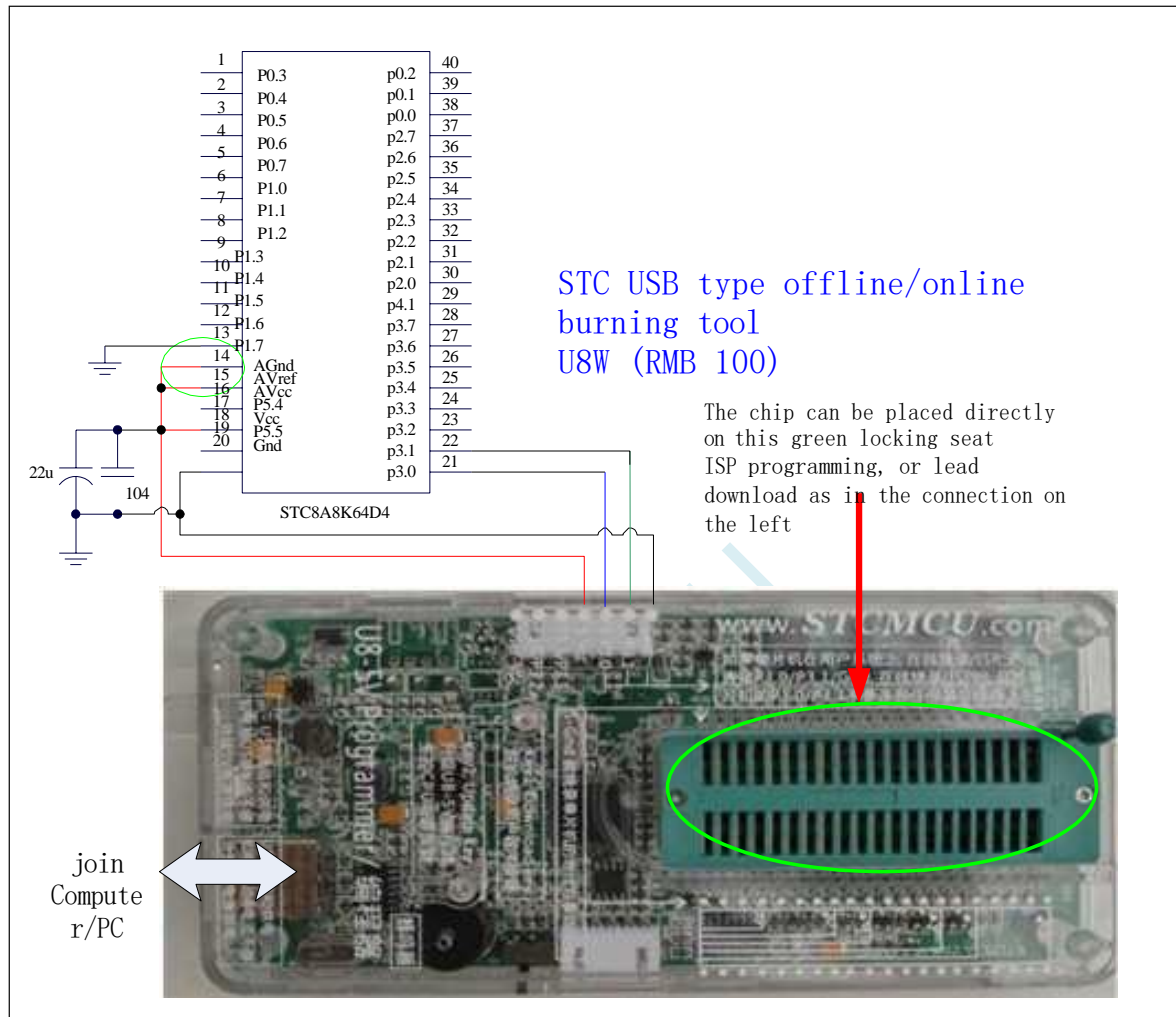
To use the U8-Mini for emulation, the U8-Mini must first be set to passthrough mode. The U8W/U8W-Mini implements USB to serial passthrough mode as follows.

1. First, the U8W/U8W-Mini firmware must be upgraded to v1.37 and above
2. After the U8W/U8W-Mini is powered on, it is in normal download mode. At this time, press and hold the Key1 (Download) button on the tool, then press the Key2 (Power) button, then release the Key2 (Power) button, then release the Key1 (Download) button, the U8W/U8W-Mini will enter USB to serial port pass-through mode. (**Press Key1 Press Key2 Release Key2 Release Key1**)
3. The U8W/U8W-Mini tool in pass-through mode is simply a USB to serial port that does not have offline download function, if you need to restore



For the original function of the U8W/U8W-Mini, simply press the Key2 (Power) button again separately.

### 5.1.9 Download using the U8W tool, which supports ISP online and offline downloads, and can also support imitation real



#### ISP Download Steps (Connected Method).

1. Connect the U8W to the target chip as shown in the diagram
2. Click the "Download/Program" button in the STC-ISP download software
3. Start ISP Download

Note: If the U8W is used to power the target system, the total current of the target system must not be greater than 200mA, otherwise the download will fail.

#### ISP download steps (on board approach)

1. Place the target chip in the direction of pin 1 near the locking wrench and pin down flush
2. Click the "Download/Program" button in the STC-ISP download software
3. Start ISP Download

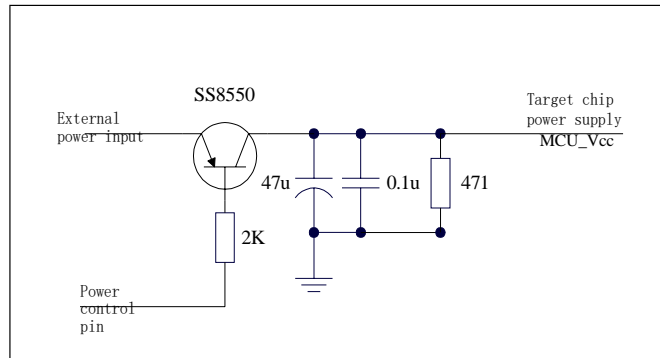
**Note: It has been found that when using the USB cable for ISP downloading, the voltage drop on the USB cable is too thin, resulting in insufficient power supply during ISP downloading.**

To use the U8W for emulation, the U8W must first be set to passthrough mode. The U8W/U8W-Mini implements USB to serial passthrough mode as follows.

1. First, the U8W/U8W-Mini firmware must be upgraded to v1.37 and above
2. After the U8W/U8W-Mini is powered on, it is in normal download mode. At this time, press and hold the Key1 (Download) button on the tool, then press the Key2 (Power) button, then release the Key2 (Power) button, then release the Key1 (Download) button, the U8W/U8W-Mini will enter USB to serial port pass-through mode. (**Press Key1 Press Key2 Release Key2 Release Key1**)
3. The U8W/U8W-Mini tool in pass-through mode is simply a USB to serial port that does not have offline download function, if you need to restore  
For the original function of the U8W/U8W-Mini, simply press the Key2 (power) button again separately.

STC MCU

### 5.1.10 Microcontroller Power Control Reference Electricity Road



## 5.2 STC-ISP Download Software Advanced Applications

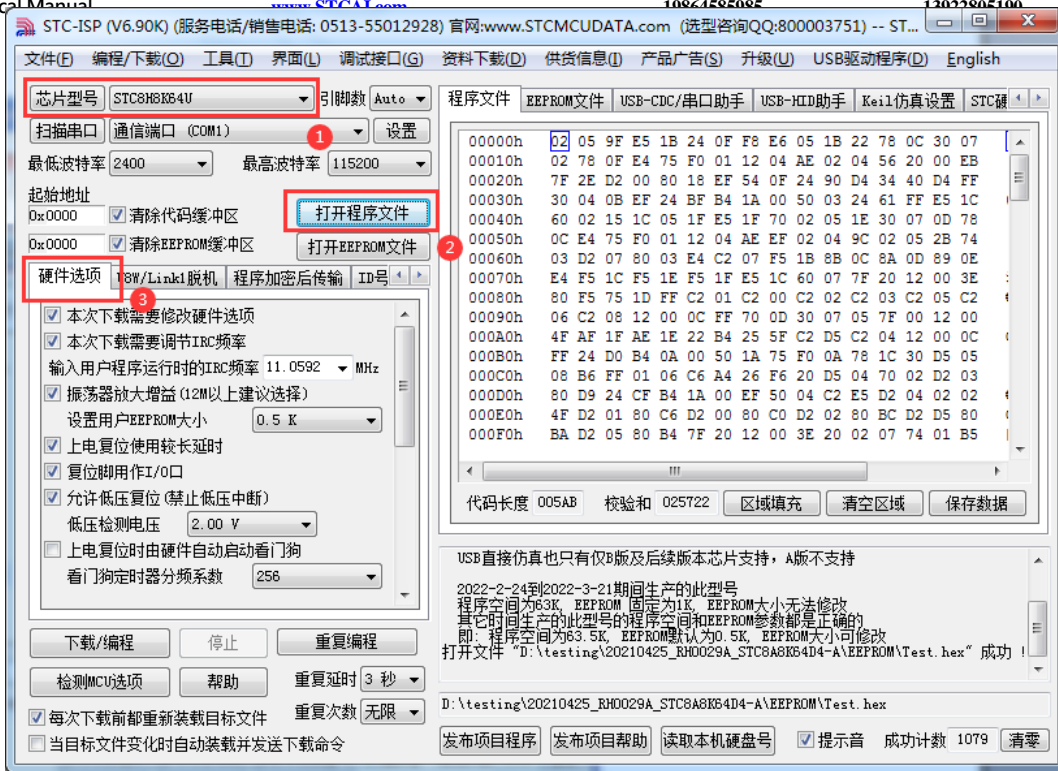
### 5.2.1 Posting project procedures

The Publish Project program function is essentially the packaging of the user's program code with associated option settings into a **super simple executable file with the user's own interface that** can be programmed directly to the target chip for download.

About the interface, the user can customize (the user can modify the title of the release project program, button names and help information) while the user can also specify the target computer's hard disk number and the target chip ID number, specify the target computer's hard disk number, then you can control the release application can only run on the specified computer (to **prevent burners will easily steal the program from the computer, such as through the network sent away**) (to prevent burners from easily stealing the program from the computer, such as through the network sent away, such as through the USB disk baked away, can not be prevented, of course, steal your computer that can not be done, so the STC offline download tool than computer burning security, can limit the number of burnable chips, so that the receptionist clerk lady burn, let the boss's wife can burn) copy to other computers, the application can not run. Similarly, when the ID number of the target chip is specified, then the user code can only be downloaded to the target chip with the corresponding ID number (especially useful for a device to sell tens of millions of products --- tanks, can be sent to customers to upgrade their own, without risking their lives to run to war-torn Iraq to upgrade the software) for other chips with inconsistent ID numbers, can not be downloaded for programming.

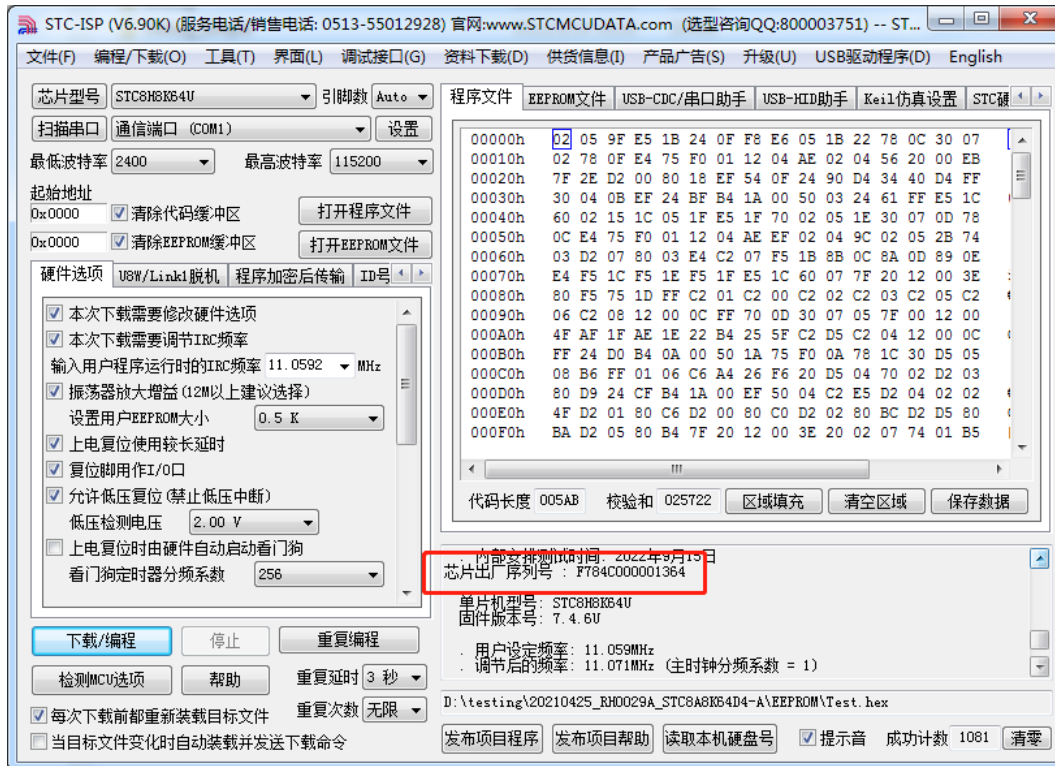
The detailed procedure for issuing a project is as follows.

- 1、First select the model of the target chip
- 2、Open the program code file
3. Set the appropriate hardware options

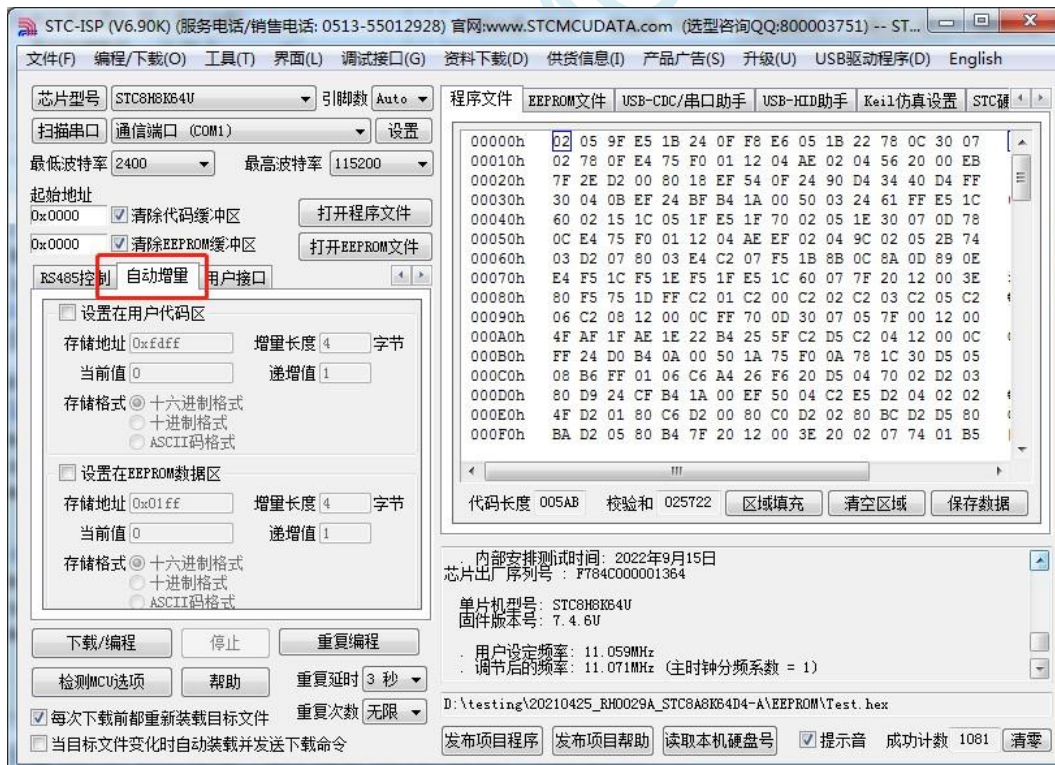


4, try to burn the chip and write down the ID number of the target chip, as shown below, the chip ID number is "F784C000001364"

(Skip this step if you do not need to verify the ID number of the target chip)

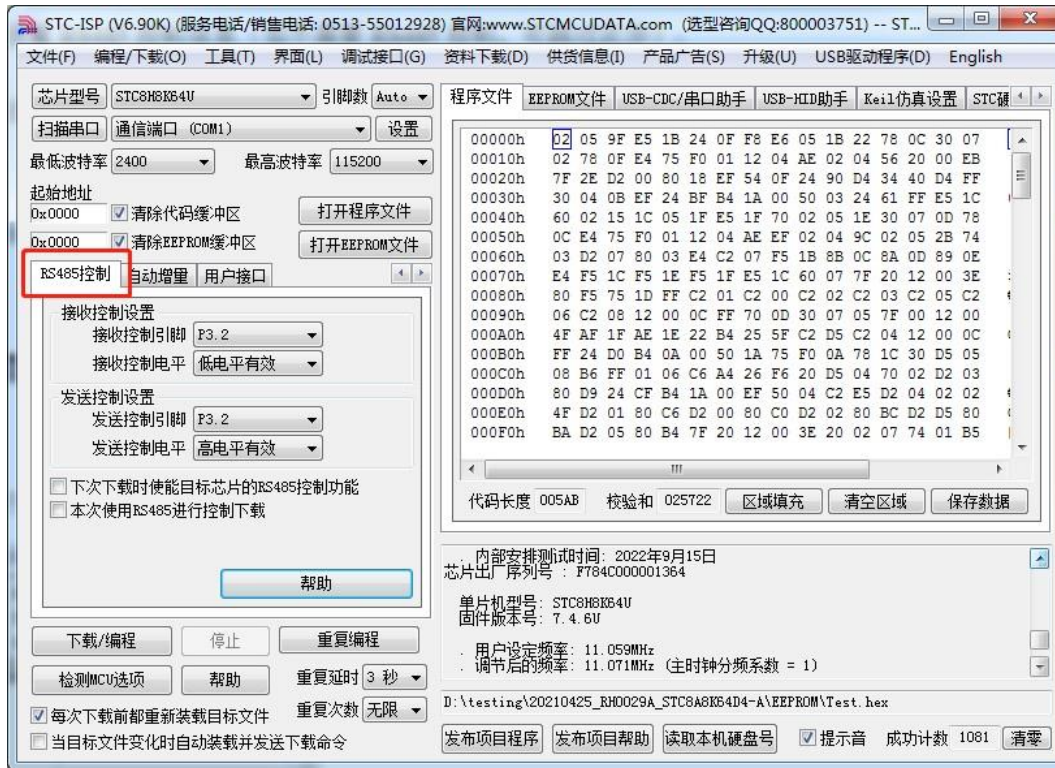


5, set the automatic increment (if you do not need automatic increment, you can skip this step)

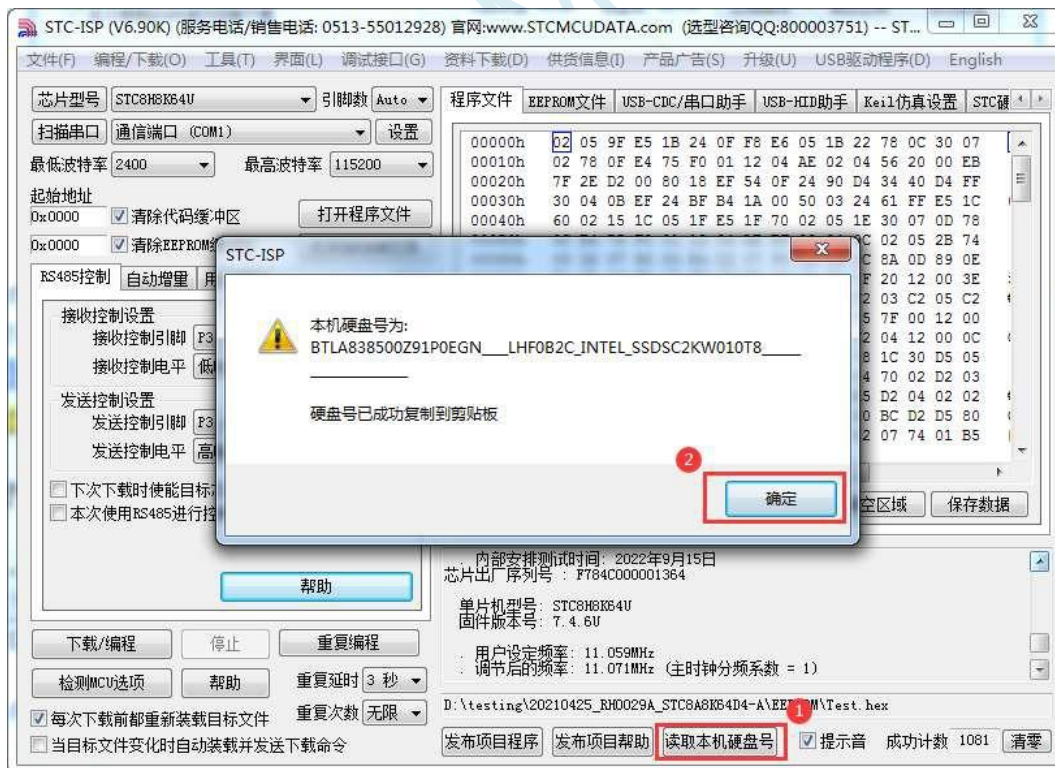


6, Setting RS485 control information (if RS485 control is not needed, you can skip this step)





- Click the "Read local hard disk number" button on the interface and write down the hard disk number of the target computer (if you do not need to verify the hard disk number



of the target computer, you can skip this step)

- Click the "Publish Project Program" button to enter the setup interface for publishing applications.
- Modify the title of the release software, the name of the download button, the name of

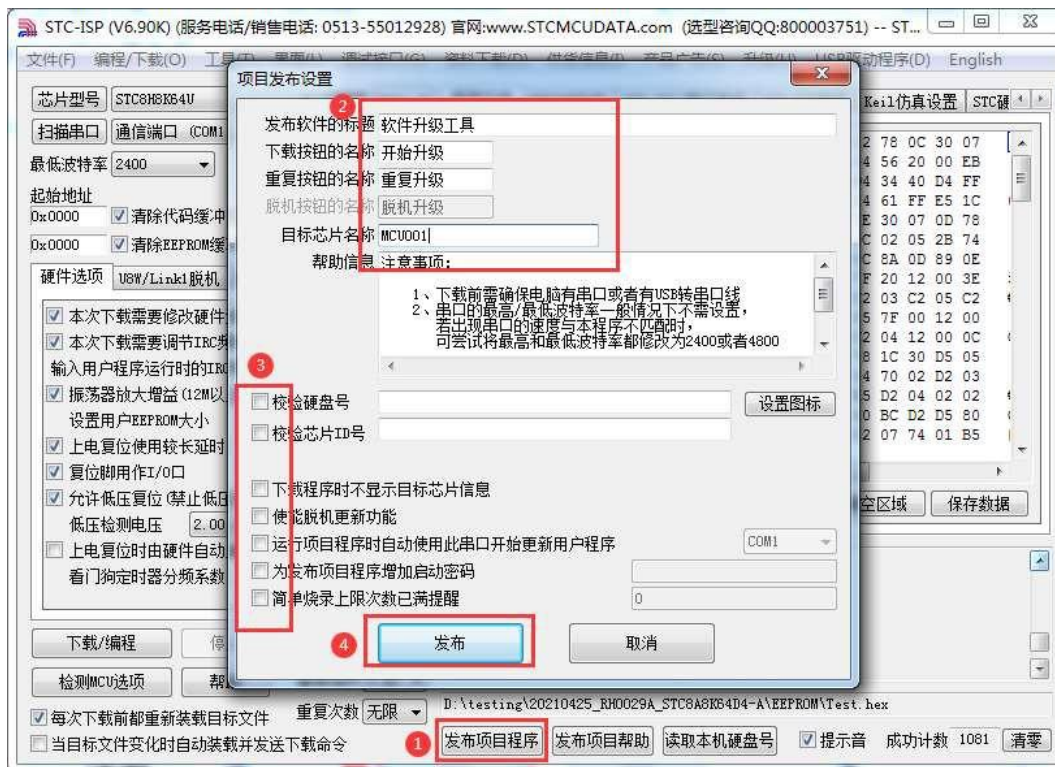


according to your respective needs

- 10、If you need to verify the hard disk number of the target computer, you need to check the "Verify hard disk number", and enter the text box behind the previous note

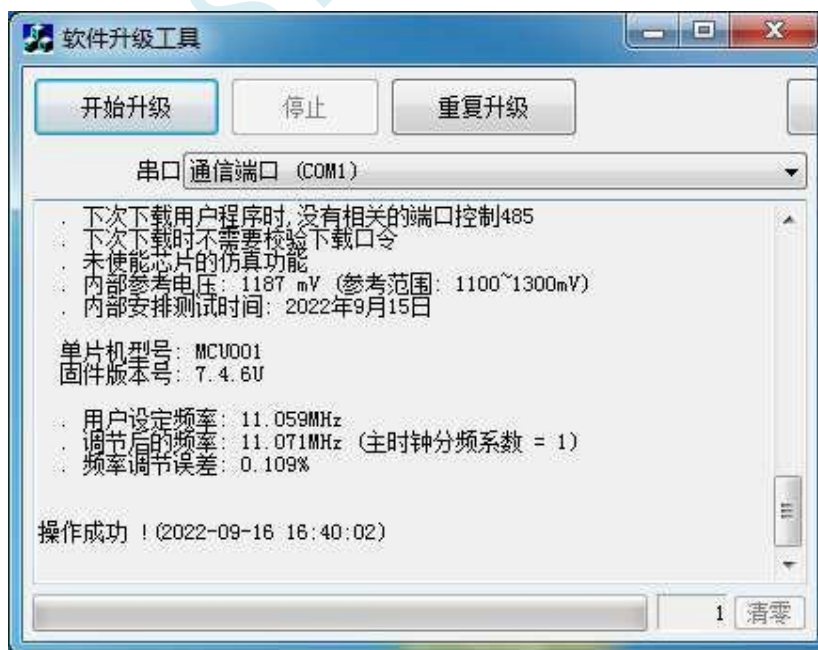
The hard drive number of the target computer under

- 11、If you need to verify the ID number of the target chip, you need to check "Verify chip ID number" and enter the ID number of the target chip noted in the text box at the



back.

- 12, and finally click the Publish button to save the project release program, you can get the corresponding executable file. The released project program hits the interface as follows



## 5.2.2 Program encrypted and transferred (to prevent serial port analysis of the program during burn-in )

At present, all ordinary serial port downloading and burning programming are using **explicit communication** (when the computer and the target chip communication, or offline downloading board and the target chip communication) the problem: If the burner analyzes the data of serial port communication when downloading and burning programming, the master can lead 2 lines out on the serial port when burning, and analyze the actual user program code by analyzing the data of serial port communication. **Of course, using STC's offline download board to burn the program is always better than using a computer to burn the program (to prevent the burner will easily steal the program from the computer, such as through the network sent away, such as through the U disk baked away, can not prevent, of course, steal your computer that can not be done, so STC's offline download tool is safer than the computer burn, let the front desk clerk lady burn, let the boss's wife burn can be)** Even the STC world's first offline download tool, for to prevent the genius of the unscrupulous in the offline download tool burning process by analyzing the serial communication data to analyze the actual user program code, there is no way to achieve the requirements, which requires the use of the latest STC microcontroller provided by the program encryption after the transfer function.

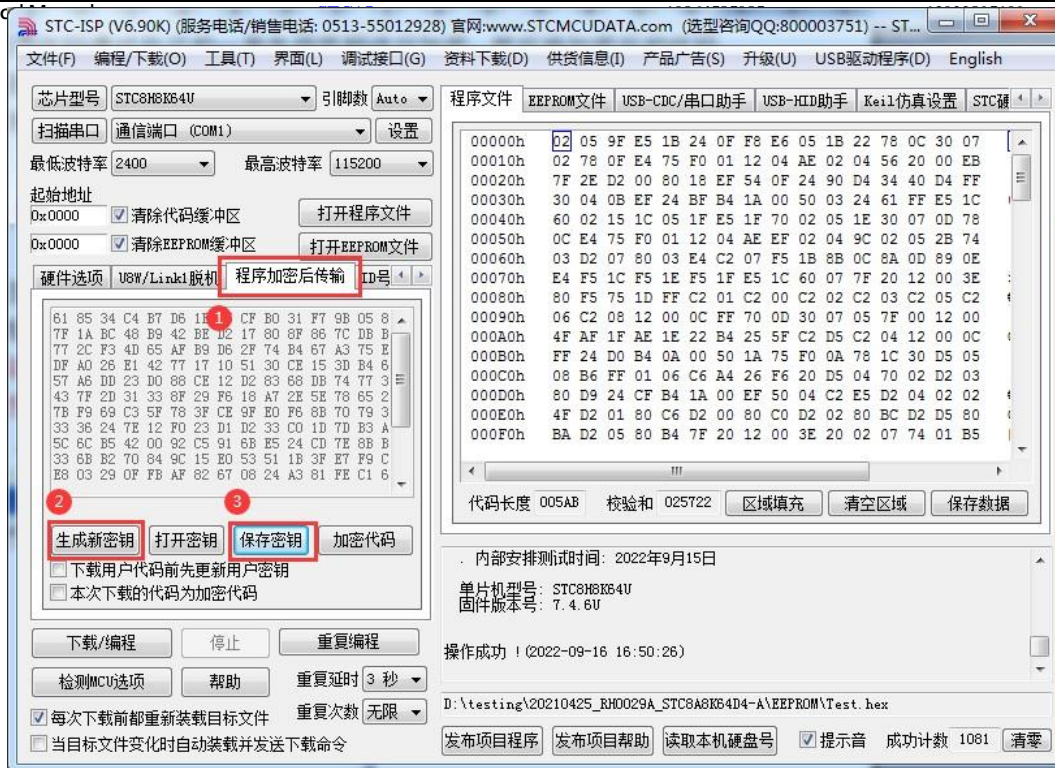
Program encryption after transmission download is the user will first encrypt the program code through their own set of special keys, and then the encrypted code and then download through the serial port, at this time the download transmission is encrypted file, through the serial port analysis out of the encrypted garbled code, such as not by sending someone to sneak into your company to steal your computer inside the encryption key, there is no value, it will play a role in preventing the burner program when the burner personnel The purpose of monitoring the serial port to analyze the code.

The use of the post-program encryption transfer function requires several steps as follows.

1. Generate and save the new key

As shown below, go to the "Transfer after encryption" page and click the "Generate new key" button to display the newly generated 256-byte key in the buffer. Then click the "Save Key" button to save the new key as a key file with ". K" extension (**note: this key file must be saved, as all future code files need to be encrypted with this key, and the key generation is non-repeating, i.e. it is impossible to generate two identical keys at any time, so once**

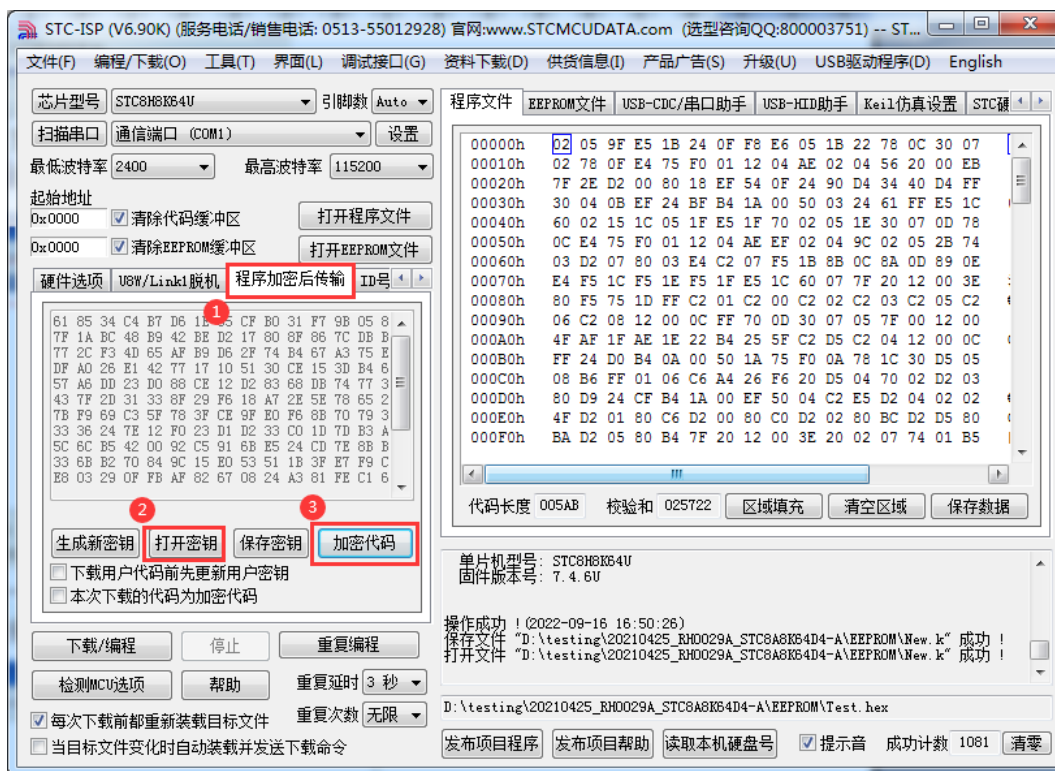
**Lost key file will not be regained**) For example, we save the key as "New.k".



## 2. Encryption of code files

Before encrypting the file, we need to open our own key. If our key is already stored in the buffer, do not open it again. As shown below, click the "Open key" button on the "Transfer after encryption" page to open the key file we saved earlier.

For example, ["New.k"](#), then go back to the "Transfer after encryption" page and click the "Encrypt code" button, as shown in the following figure, firstly, a pop-up will appear "Open

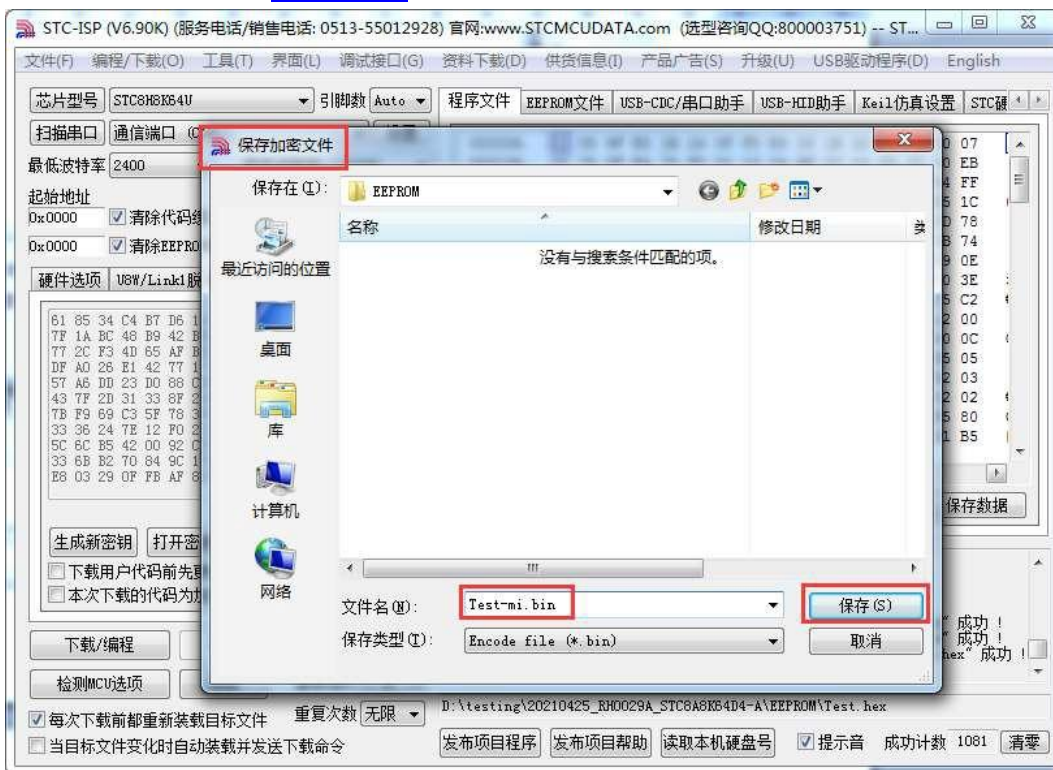


Source File (Unencrypted) dialog box will pop up, at this time, the original unencrypted code file is selected.

After clicking the Open button, a similar dialog box will pop up immediately, but this is a dialog box for saving the encrypted file. As shown in the figure below, click the Save button to save the encrypted file.

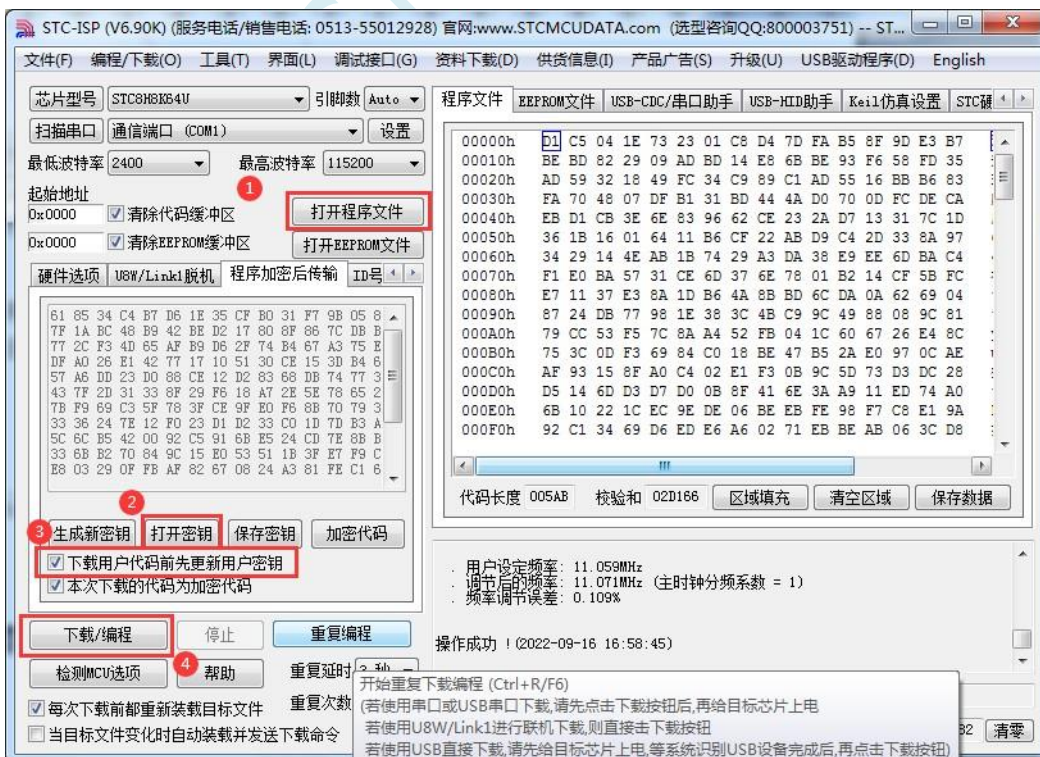






### 3. Update the user key to the target chip

Before updating the key, we need to open our own key. If our key is already stored in the buffer, do not open it again. As shown below, click the "Open Key" button on the "Custom Encryption Download" page to open our previously saved key file, e.g. "New.k". After the key is opened, as shown below, check the "Update user key before downloading user code" option and "The downloaded code is encrypted code" option, then open our previously encrypted file and click the

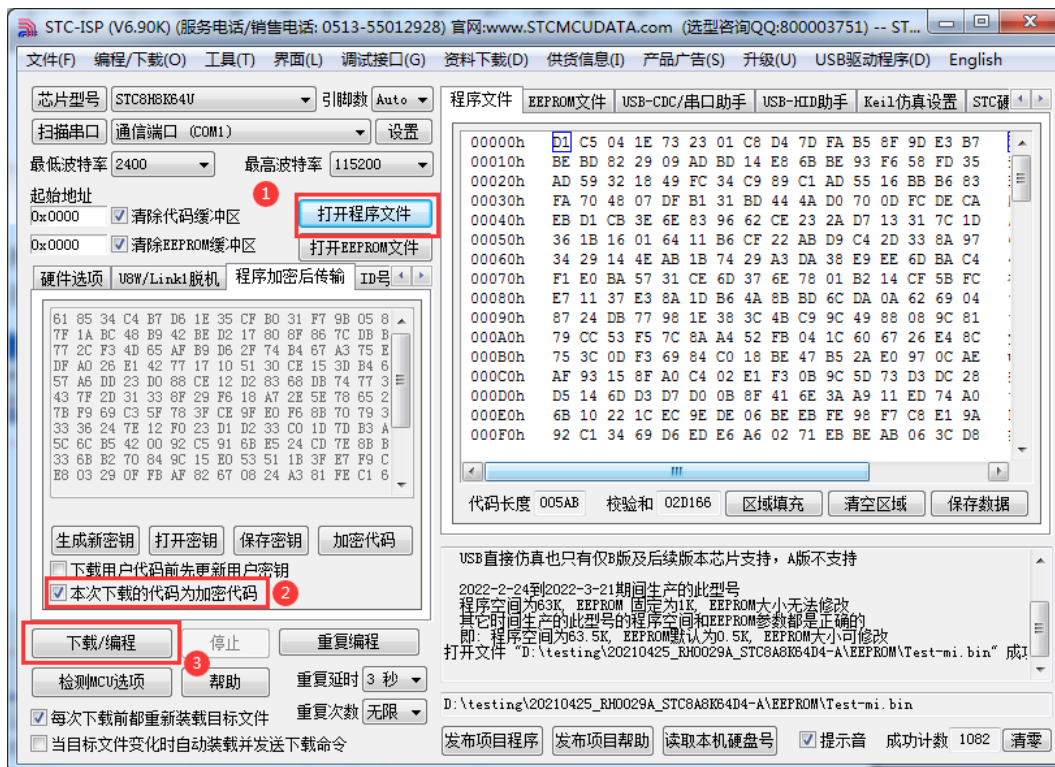


Technical Manual  
Download/Program" button on the left corner of the interface. Click the "Download/Program" button in the bottom left corner of the interface, and download the target chip in the normal way to update the user key.



## 4. Encrypted update of user code

After the key is updated successfully, the target chip will have the function to receive the encrypted code and restore it. At this time, if you need to update/update



the code again, you only need to refer to the second step to encrypt the target code, and then the following figure

For a new STC microcontroller, you can combine step 3 and step 4, i.e. update the key to the target microcontroller and download the encrypted code to the microcontroller at the same time, if you have already performed step 3 (i.e. you have already updated the key to the target chip), only need to follow step 4 for the subsequent code update, just select the option "Transfer the encrypted code after the program" on the "Transfer the encrypted code" page. In the "Transfer after encryption" page, select the option of "The downloaded code is encrypted code" (need to select the option of "Update user key before downloading user code") then open the file we encrypted before, open it and click Click the "Download/Program" button in the lower left corner of the interface, and download the target chip in the normal way to complete the purpose of updating the user code with the user's own encrypted file (to prevent the code from being analyzed by the burner through the monitoring serial port when burning the program)

### 5.2.3 Release Project Program + Program encrypted for post-transmission use in combination

Two new special features can be used in conjunction with the release of the project program and the post-encryption transfer of the program. First of all, the post-encryption transmission of the program ensures the confidentiality of the user's code during the serial communication transfer during the programming burn-in, while the release of the project program allows the end-user to update the program remotely (without the need for the solution company's personnel to be physically present) So the combination of the two functions is very suitable for the solution company/manufacturer when the software needs to be updated, so that the end user can update the software of the end product by themselves, but also to ensure that the field burner can not analyze the useful program through the serial port, and is highly recommended for the solution company.

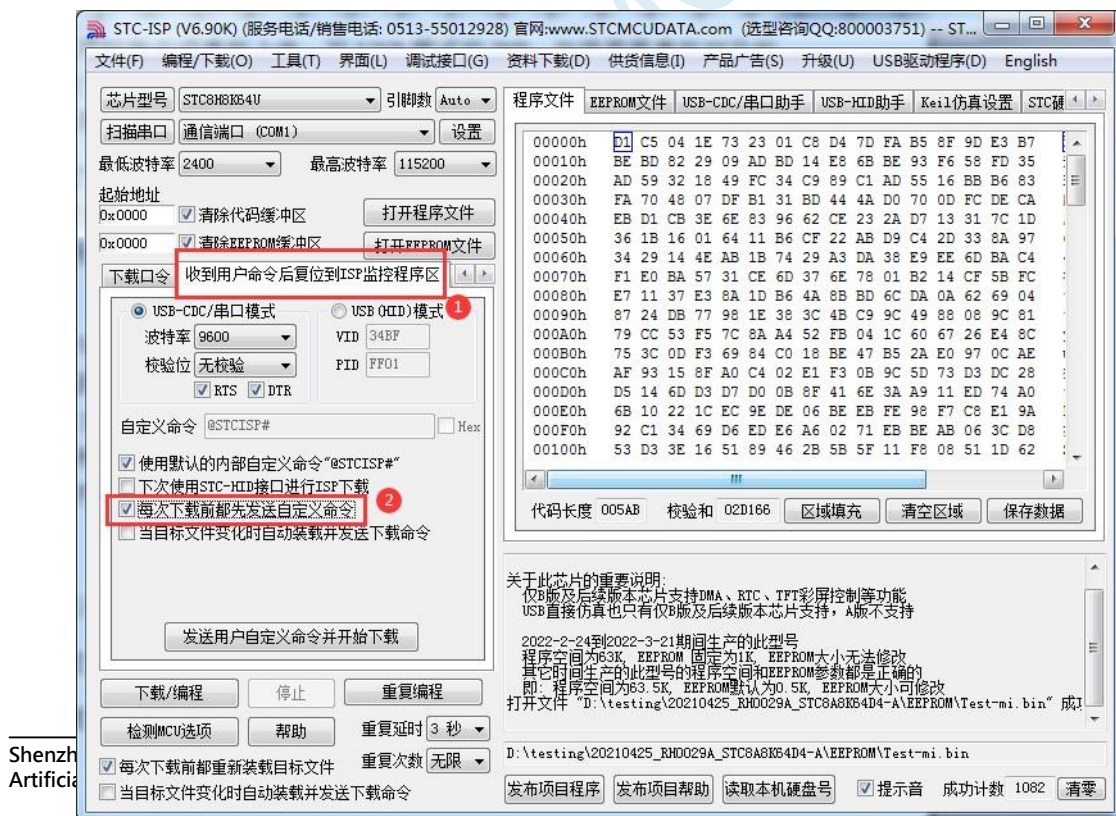
STC MCU

### 5.2.4 User-defined downloads (for non-stop downloads)

Downloading the user's target program to the **STC** microcontroller is accomplished by executing the microcontroller's internal **ISP** system code and the host computer for serial or **USB** communication. However, the internal **ISP** code of the **STC** microcontroller will only be executed every time the power is re-stopped and re-powered, which requires the user to re-power the target microcontroller every time he needs to update the program, while for the **USB mode ISP**, in addition to re-powering the target chip, the **P3.2** port needs to be pulled down to **GND** at power-up. For projects in the development stage, the code needs to be modified and updated frequently, and it is very troublesome to re-power the target chip every time it is downloaded.

The **STC** microcontroller adds a soft reset register (**IAP\_CONTR**) to the hardware design, allowing the user to set this register to determine whether the CPU resets and re-executes the user code or resets to the **ISP area** to execute the **ISP system code**. When **0x20** is written to the **IAP\_CONTR** register, the **CPU** resets and re-executes the user code; when **0x60** is written to the **IAP\_CONTR** register, the **CPU** resets and resets to the **ISP area** to execute the **ISP system code**.

To implement a non-stop **ISP** download, the user can design a piece of code in the program, such as detecting a special key, or monitoring the serial port waiting for a special serial command, and when the download condition is detected, the software triggers the soft reset register to reset to the **ISP** area to execute the **ISP** system code, thus achieving a non-stop **ISP** download. When the trigger condition is an external key, the key status can be monitored in real time in the user code. To fully synchronize the **STC-ISP** software with the user-triggered soft reset, use the "Reset to **ISP** monitor area after receiving user command" function provided in the **STC-ISP** software.



The steps to achieve a non-stop ISP download are as follows.

1. Write user code and add serial port command monitoring program to the user code  
(Reference code below, test microcontroller model is

STC8H8K64U) #include "stc8h.h"

```
#define FOSC 11059200UL
#define BAUD (65536 - FOSC/4/115200)

char code *STCISPCMD = "@STCISP#"; // custom
download command char index;

void uart_isr() interrupt 4
{
    char dat;

    if (TI)
    {
        TI = 0;
    }

    if (RI)
    {
        RI = 0;
        dat = SBUF; //Receive data from serial port

        if (dat == STCISPCMD[index]) //determine if the received data and the current
            command character match
        {
            index++; //index+1 if match
            if (STCISPCMD[index] == '\0') //judge if the command
                match is complete IAP_CONTR = 0x60; //soft
            }
            reset to ISP if matching is complete
        else
        {
            index = 0; //If it doesn't match,
            you need to start from the beginning if (dat ==
        }
        STCISPCMD[index])
        index++;
    }
}

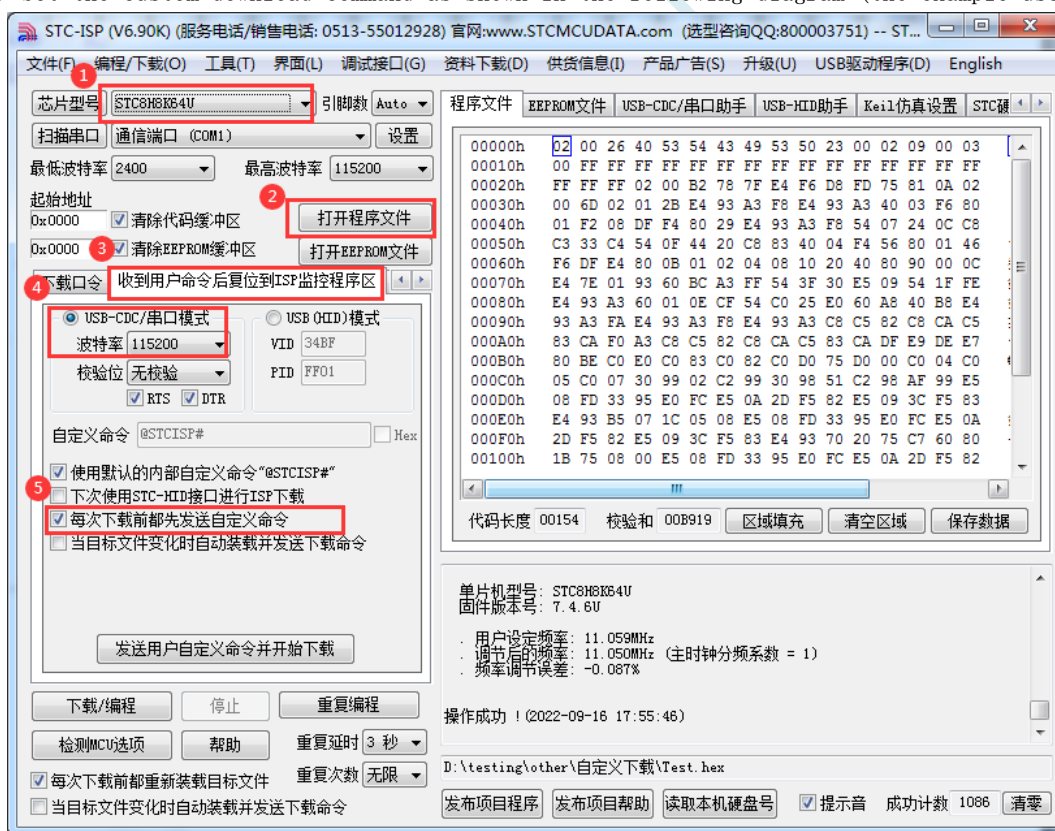
void main()
{
    P0M0 = 0x00; P0M1 = 0x00;
```

P1M0 = 0x00; P1M1 = 0x00;

```
P2M0 = 0x00; P2M1 = 0x00;  
P3M0 = 0x00; P3M1 = 0x00;
```

```
SCON = 0x50; //Serial port initialization  
AUXR = 0x40;  
TMOD = 0x00;  
TH1 = BAUD >> 8;  
TL1 = BAUD;  
TR1 = 1;  
ES = 1;  
EA = 1;  
  
index = 0; //initialize command  
  
while (1);  
}
```

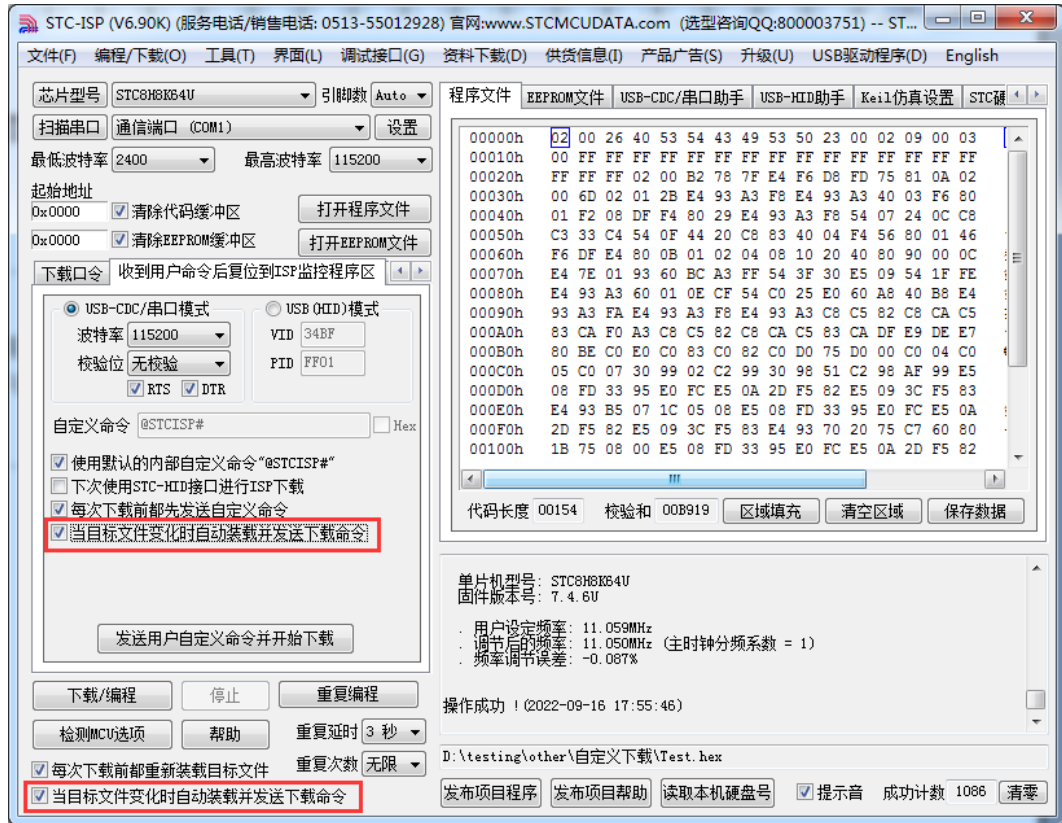
2. Set the custom download command as shown in the following diagram (the example uses the STC



default command "@STCISP#")

- 3, the first download needs to re-power the target microcontroller, after each update only need to click the "Download/Program" button in the download software, the download software will automatically send the download command to the target microcontroller, the target microcontroller received the command automatically reset to the system ISP area, you can achieve non-stop update of user code.

- 4, STC-ISP can also realize the project development stage, fully automatic download function, that is, when the download software detects that the target code has been



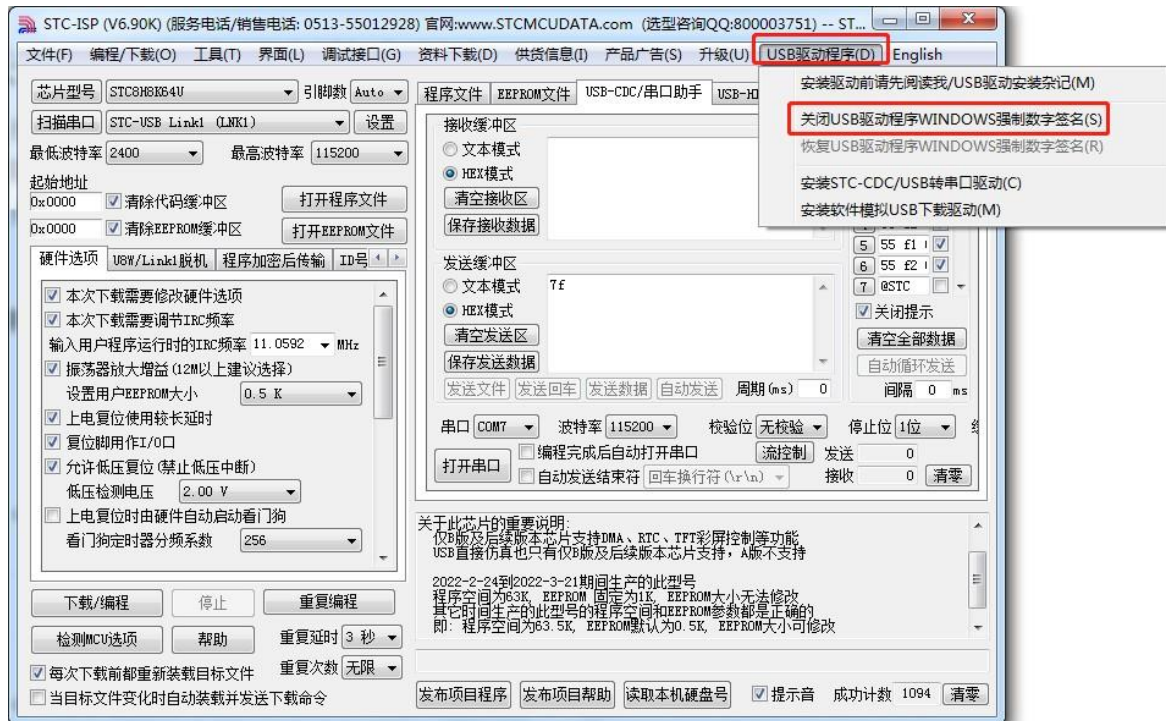


## 5.3 Instructions related to digital signature of drivers

### 5.3.1 About mandatory digital signatures for drivers

Windows systems starting with the Vista version of 64-bit Windows force the driver to be digitally signed for installation, otherwise the driver cannot be installed and the USB device cannot be used properly. Currently, the drivers provided by STC are not WHQL certified (Microsoft logo certified) at the moment, and may not be installed successfully on 64-bit Windows systems starting from Vista until the mandatory digital signature of the drivers is turned off.

The latest STC-ISP download software provides a simple disabling of the driver's forced digital signature as shown in the menu directory below.



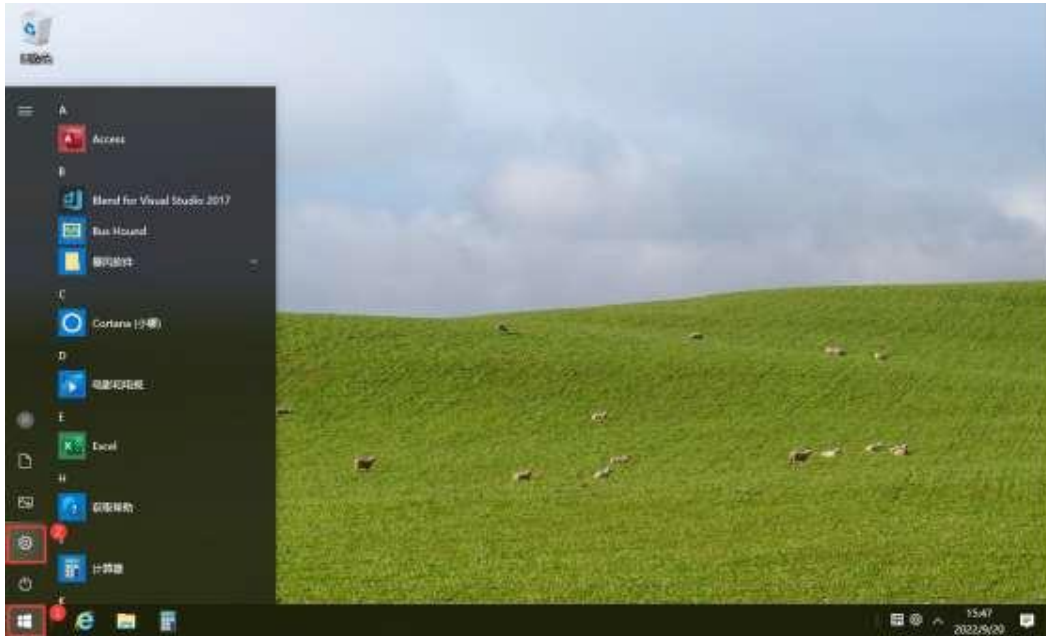
This menu item uses the TestSigning and NoIntegrityChecks parameters automatically set by the system's BCDEDIT utility to temporarily disable the driver's forced digital signature. For systems with Secure Boot enabled, the BCDEDIT utility will not work and Secure Boot needs to be disabled in the BIOS (set Secure Boot to Disable)

Currently tested WinXP/Win7-32/Win7-64 systems can disable digital signature normally, but Win10 and later systems do not work. To disable digital signatures on Win10 and later systems, you still need to disable forced digital signatures from the system boot menu. Please refer to the next section for detailed steps



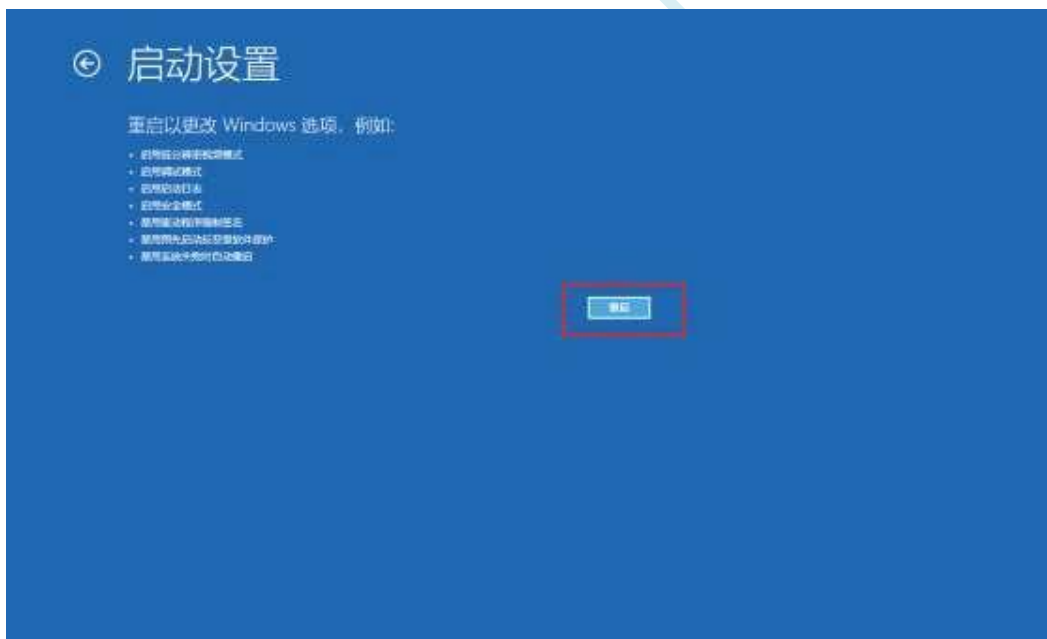
## 5.3.2 Windows 10 Steps to turn off mandatory digital signatures for drivers

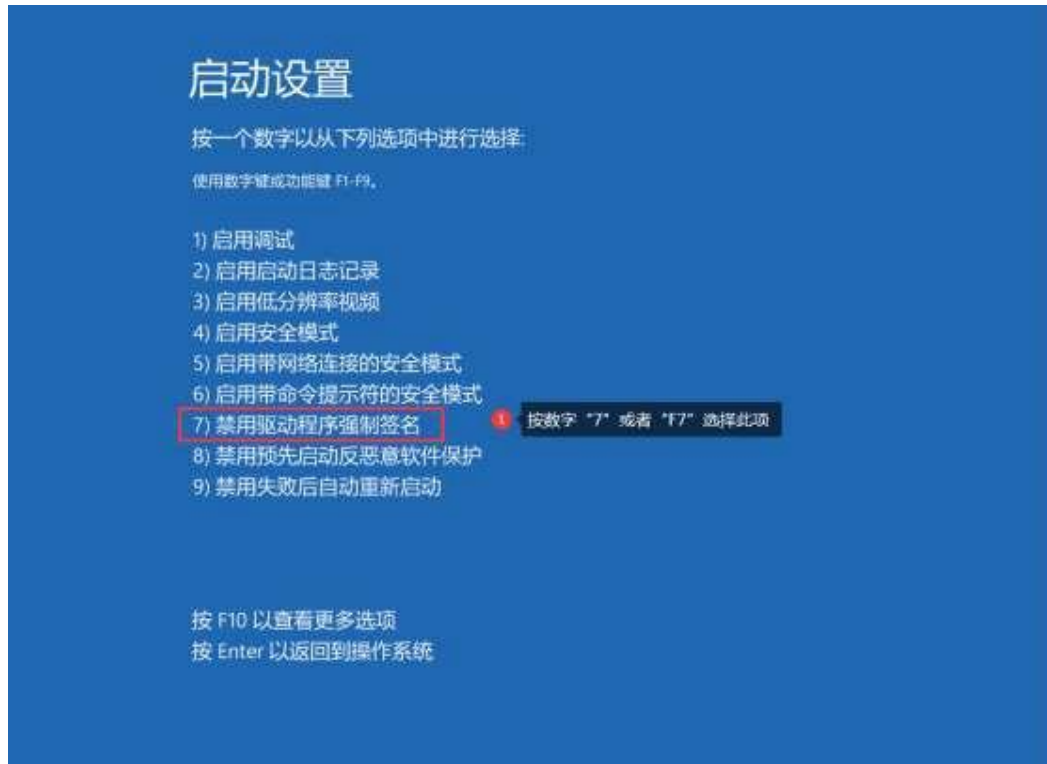
Select the "Settings" function in the "Start" menu











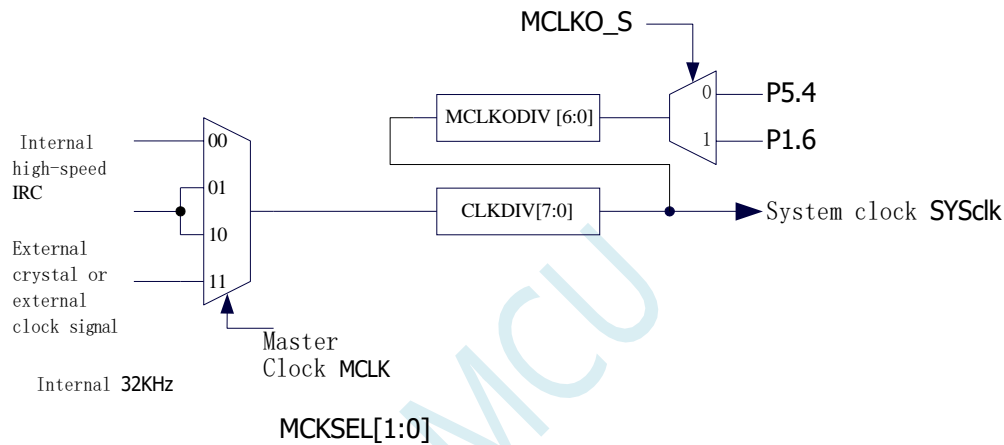
The driver's mandatory digital signature can be temporarily disabled after a system reboot

## 6 Clock, Reset, Power Save Mode and System Power Management Management

### 6.1 System clock control

The system clock controller provides the clock source for the microcontroller CPU and all peripheral systems. There are three clock sources available for the system clock: internal high precision IRC, internal 32KHz IRC (with large error) and external crystal. The user can programmatically enable and disable each clock source, as well as provide internal clock division to reduce power consumption.

After the microcontroller enters power-down mode, the clock controller will turn off all clock sources



System Clock Structure Diagram

#### Related Registers

sym bolic	description	addre ss	Bit Addresses and Symbols								reset value
			B7	B6	B5	B4	B3	B2	B1	B0	
CLKSEL	Clock Select Register	FE00H	-						MCKSEL[1:0]		xxxx,xx00
CLKDIV	Clock division register	FE01H									Nnnn, nnnn
HIRCCR	Internal high-speed oscillator control register	FE02H	ENHIRC	-	-	-	-	-	-	HIRCST	1xxx,xxx0
XOSCCR	External crystal control register	FE03H	ENXOSC	XITYPE	XCFILTER[1:0]		NMXCG	-	-	XOSCST	0000,0xx0
IRC32KCR	Internal 32K oscillator control register	FE04H	ENIRC32K	-	-	-	-	-	-	IRC32KST	0xxx,xxx0
MCLKOCR	Master Clock Output Control Register	FE05H	MCLKO_S	MCLKODIV [6:0]							0000,0000
IRCDDB	Internal IRC De-Shake Control	FE06H	IRCDDB[7:0]								1000,0000

## 6.1.1 System Clock Select Register (CLKSEL )

symbolic	addresses	B7	B6	B5	B4	B3	B2	B1	B0
CLKSEL	FE00H	-						MCKSEL[1:0]	

MCKSEL[1:0]: master clock source selection

MCKSEL[1:0]	master clock source
00	Internal high precision IRC
01	External crystal oscillator or external input clock signal
10	-
11	Internal 32KHz low speed IRC

## 6.1.2 Clock division register (CLKDIV )

symbolic	address	B7	B6	B5	B4	B3	B2	B1	B0
CLKDIV	FE01H								

CLKDIV: Master clock dividing factor. The system clock SYSCLK is the clock signal after dividing the master clock MCLK.

CLKDIV	System clock frequency
0	MCLK/1
1	MCLK/1
2	MCLK/2
3	MCLK/3
...	...
control	MCLK/x
...	...
255	MCLK/255

Note: After a user program reset, the system will automatically set the initial value of this register based on the crossover factor required to set the operating frequency during the last ISP download

## 6.1.3 Internal High Speed High Precision IRC Control Register (HIRCCR )

symbolic	address	B7	B6	B5	B4	B3	B2	B1	B0
HIRCCR	FE02H	ENHIRC	-	-	-	-	-	-	HIRCST

ENHIRC: Internal high speed, high accuracy IRC enable bit

0: Turn off internal high-precision IRC

1: Enables internal high-precision IRC

HIRCST: Internal high precision IRC frequency stabilization flag bit.(read-only bit)

When the internal IRC is enabled from the stop state, a period of time must pass before the oscillator frequency stabilizes, and when the oscillator frequency stabilizes, the clock controller automatically sets the HIRCST flag bit to 1. So when the user program needs to switch the clock to use the internal IRC, it must first set ENHIRC=1 to enable the oscillator, and then keep querying the oscillator stabilization flag bit HIRCST until the flag bit changes to 1, and then the clock source switch.

## 6.1.4 External Oscillator Control Register (XOSCCR )

symbolic	address	B7	B6	B5	B4	B3	B2	B1	B0
XOSCCR	FE03H	ENXOSC	XITYPE	XCFILTER[1:0]	NMXCG	-	-	-	XOSCST

ENXOSC: External crystal

oscillator enable bit 0:

Turn off external crystal



1: Enables external  
crystal oscillator

XITYPE: External clock

source type

0: The external clock source is an external clock signal (or active crystal) The source only needs to be connected to XTALI (P1.7) of the microcontroller **(at this point P1.6**

**(The port is fixed in high resistance input mode, which can be used to read external digital signals or as ADC input, but is generally not recommended because of the high frequency oscillation of the adjacent P1.7 port which can affect the signal of P1.6)**

1: The external clock source is a crystal oscillator. The signal source is connected to XTALI (P1.7) and XTALO (P1.6) of the microcontroller XOSCST: External crystal oscillator frequency stabilization flag bit.(read-only bit)

When an external crystal oscillator is enabled from a stopped state, a period of time must pass before the oscillator frequency stabilizes when the oscillator

Once the frequency is stable, the clock controller automatically sets the XOSCST flag to position 1. So when the user program needs to switch the clock to use an external crystal oscillator, it must first set ENXOSC=1 to enable the oscillator, and then keep querying the oscillator stability flag bit XOSCST until the flag bit changes to 1 before the clock source can be switched.

XCFILTER[1:0]: external crystal oscillator anti-interference control register

00: Selectable when the external crystal oscillator frequency is 48M or less

01: Selectable when the external crystal oscillator frequency is 24M or less

1x: Selectable for external crystal oscillator frequencies of 12M and below

NMXCG: External crystal

oscillator gain control 0:

Low gain

1: High gain

## 6.1.5 Internal 32KHz low-speed IRC control register (IRC32KCR )

symbolic	address	B7	B6	B5	B4	B3	B2	B1	B0
IRC32KCR	FE04H	ENIRC32K	-	-	-	-	-	-	IRC32KST

ENIRC32K: Internal 32K low-speed IRC enable bit

0: Turn off internal 32K low-speed IRC

1: Enables internal 32K low-speed IRC

IRC32KST: Internal 32K low-speed IRC frequency stabilization flag bit(read-only bit)

When the internal 32K low-speed IRC is enabled from the stop state, a period of time must elapse before the oscillator frequency stabilizes, and when the oscillator frequency stabilizes, the clock controller automatically sets the IRC32KST flag bit to 1. So when the user program needs to switch the clock to use the internal 32K low-speed IRC, it must first set ENIRC32K=1 to enable the oscillator, and then keep querying the oscillator stability flag bit IRC32KST until the flag bit changes to 1, then it can The clock source switch is performed.

## 6.1.6 Master Clock Output Control Register (MCLKOCR )

symbolic	address	B7	B6	B5	B4	B3	B2	B1	B0
MCLKOCR	FE05H	MCKO_S	MCLKODIV [6:0]						

MCLKODIV[6:0]: master clock output dividing factor

(Note: the clock source of the master clock divider output is the system clock after CLKDIV dividing)

MCLKODIV [6:0]	System clock divider output frequency
0000000	No clock output
0000001	SYSClk/1
0000010	SYSClk /2
0000011	SYSClk /3
...	...
1111110	SYSClk /126
1111111	SYSClk /127

MCU pin selection 0: System clock divider output to P5.4 port

1: System clock divider output to P1.6 port

## 6.2 STC8A8K64D4 Series Internal IRC Frequency Tuning

The STC8A8K64D4 series microcontrollers all have a high precision internal IRC oscillator integrated inside. When users download using the ISP download software, the ISP download software will automatically adjust the frequency according to the user's selection/setting, and the general frequency value can be adjusted to  $\pm 0.3\%$  or less, and the temperature drift of the adjusted frequency can reach -1.35% to 1.30% in the full temperature range (-40°C to 85°C).

The STC8A8K64D4 series has four internal IRC bands with center frequencies of 6MHz, 10MHz, 27MHz, and 44MHz, respectively. The adjustment range for the 44M band is 29MHz to 55MHz (Note: there may be a manufacturing error of about 5% for different chips and different generation batches) It is recommended that users do not set the IRC frequency higher than 45MHz during ISP download.

Note: For general users, the internal IRC frequency adjustment can be left alone, because the frequency adjustment is done automatically when the ISP download is performed. Therefore, if the user does not need to adjust the frequency by himself, then the following 4 registers should not be modified arbitrarily, otherwise it may cause the operating frequency to change.

If you need to dynamically select the chip's preset frequencies in your own code, please refer to the list of preset frequencies and the "User-defined Internal Example program for "IRC Frequency

The internal IRC frequency adjustment uses the following 4 registers to adjust the relevant registers

symbolic	description	address	Bit Addresses and Symbols								reset value
			B7	B6	B5	B4	B3	B2	B1	B0	
IRCBAND	IRC Band Selection	9DH	-	-	-	-	-	-	SEL[1:0]		0000,00nn
LIRTRIM	IRC Frequency Trim Register	9EH	-	-	-	-	-	-	-	LIRTRIM	0000,00nn
IRTRIM	IRC Frequency Adjustment Register	9FH	IRTRIM[7:0]								Nnnn, nnnn
CLKDIV	Clock division register	FE01H	B7	B6	B5	B4	B3	B2	B1	B0	Nnnn, nnnn
IRCBAND	9DH	-	-	-	-	-	-	-	SEL[1:0]		

SEL[1:0]: band

selection 00:

selects the 6MHz

band

01: Select 10MHz band

10: Select 27MHz band

11: Select 44MHz band

### 6.2.2 Internal IRC Frequency Adjustment Register (IRTRIM)

symbolic	address	B7	B6	B5	B4	B3	B2	B1	B0
----------	---------	----	----	----	----	----	----	----	----

IRTRIM[7:0]: Internal high precision IRC frequency adjustment register

IRTRIM adjusts the IRC frequency over 256 levels, and the frequency values adjusted at each level are linearly distributed overall, with local fluctuations. Macroscopically, the frequency adjusted at each level is about 0.24%, i.e., the frequency at IRTRIM (n+1) is about 0.24% faster than the frequency at IRTRIM (n). However, since the IRC frequency adjustment is not 0.24% at each level (the maximum value of the frequency adjusted at each level is about 0.55%, the minimum value is about 0.02%, and the overall average value is about 0.24%), it can cause local fluctuations.

## 6.2.3 Internal IRC Frequency Trim Register (LIRTRIM )

symbolic	address	B7	B6	B5	B4	B3	B2	B1	B0
LIRTRIM	9EH	-	-	-	-	-	-	-	LIRTRIM

LIRTRIM: Internal high precision IRC frequency trim register

## 6.2.4 Clock division register (CLKDIV )

symbolic	addresses	B7	B6	B5	B4	B3	B2	B1	B0
CLKDIV	FE01H								

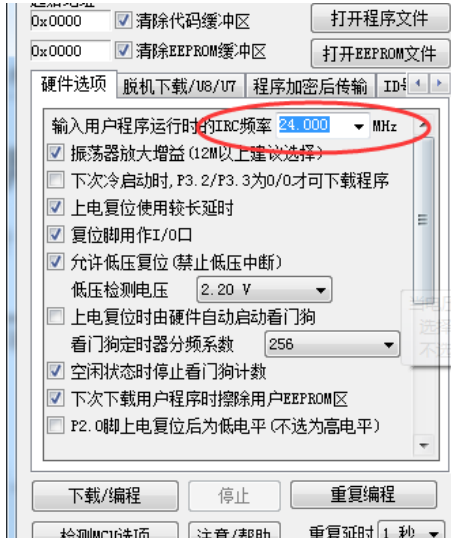
CLKDIV: Master clock dividing factor. The system clock SYSCLK is the clock signal after dividing the master clock MCLK.

CLKDIV	System clock frequency
0	MCLK/1
1	MCLK/1
2	MCLK/2
3	MCLK/3
...	...
control	MCLK/x
...	...
255	MCLK/255

## 6.2.5 Crossover out to 3MHz user operating frequency and user dynamically changing frequency chasing example

To obtain a frequency of 3 MHz, the method  $24 \text{ MHz} \div 8$  can be used.

First select the internal IRC operating frequency of 24MHz when performing the ISP download, as shown in the following figure.



The clock source is then selected in the code to be the internal IRC and the CLKDIV register is used for 8 division.

### C code

```
//Test operating frequency of 24MHz
```

```
#include "reg51.h"
#include "intrins.h"
```

```
#define CLKSEL      (*(unsigned char volatile xdata *)0xfe00)
#define CLKDIV      (*(unsigned char volatile xdata *)0xfe01)
#define HIRCCR      (*(unsigned char volatile xdata *)0xfe02)
#define XOSCCR      (*(unsigned char volatile xdata *)0xfe03)
#define IRC32KCR    (*(unsigned char volatile xdata *)0xfe04)
```

```
sfr P_SW2          = 0xba;
sfr IRTRIM         = 0x9f;
```

```
sfr P0M1           = 0x93;
sfr P0M0           = 0x94;
sfr P1M1           = 0x91;
sfr P1M0           = 0x92;
sfr P2M1           = 0x95;
sfr P2M0           = 0x96;
sfr P3M1           = 0xb1;
sfr P3M0           = 0xb2;
sfr P4M1           = 0xb3;
sfr P4M0           = 0xb4;
sfr P5M1           = 0xc9;
sfr P5M0           = 0xca;
```

```
void main()
```

```
{
```

```
    P0M0 = 0x00;
```



```

P0M1      =
0x00; P1M0
=      0x00;
P1M1      =
0x00; P2M0
=      0x00;
P2M1      =
0x00; P3M0
=      0x00;
P3M1      =
0x00; P4M0
=      0x00;
P4M1      =
0x00; P5M0
=      0x00;
P5M1 = 0x00;

P_SW2 = 0x80;
CLKSEL = // Select internal IRC ( default )
0x00;
CLKDIV = /clock 8 division
0x08; P_SW2 =
0x00;

IRTRIM++; //IRC Frequency up 3‰ for fine tuning (note judging
// IRTRIM--; boundaries)
// IRC frequency down 3‰ for fine tuning (note
judgment boundaries)

while (1);
}

```

## assembly code

; tested operating frequency of 24MHz

```

P_SW2      DATA      0BAH
IRTRIM      DATA      09FH

CLKSEL      EQU        0FE00H
CLKDIV      EQU        0FE01H
HIRCCR      EQU        0FE02H
XOSCCR      EQU        0FE03H
IRC32KCR    EQU        0FE04H

P0M1        DATA      093H
P0M0        DATA      094H
P1M1        DATA      091H
P1M0        DATA      092H
P2M1        DATA      095H
P2M0        DATA      096H
P3M1        DATA      0B1H
P3M0        DATA      0B2H
P4M1        DATA      0B3H
P4M0        DATA      0B4H
P5M1        DATA      0C9H
P5M0        DATA      0CAH

ORG         0000H
LJMP        MAIN

```

STC8A8K64D4 Series	Official website:	Technical Support:	Selection Consultant:
Technical Manual	<a href="http://www.STCAL.com">www.STCAL.com</a>	19864585985	13922805190
<i>ORG</i>	<i>0I00H</i>		
<i>MAIN:</i>			
<i>MOV</i>	<i>SP, #5FH</i>		
<i>MOV</i>	<i>P0M0, #00H</i>		
<i>MOV</i>	<i>P0M1, #00H</i>		
<i>MOV</i>	<i>P1M0, #00H</i>		
<i>MOV</i>	<i>P1M1, #00H</i>		