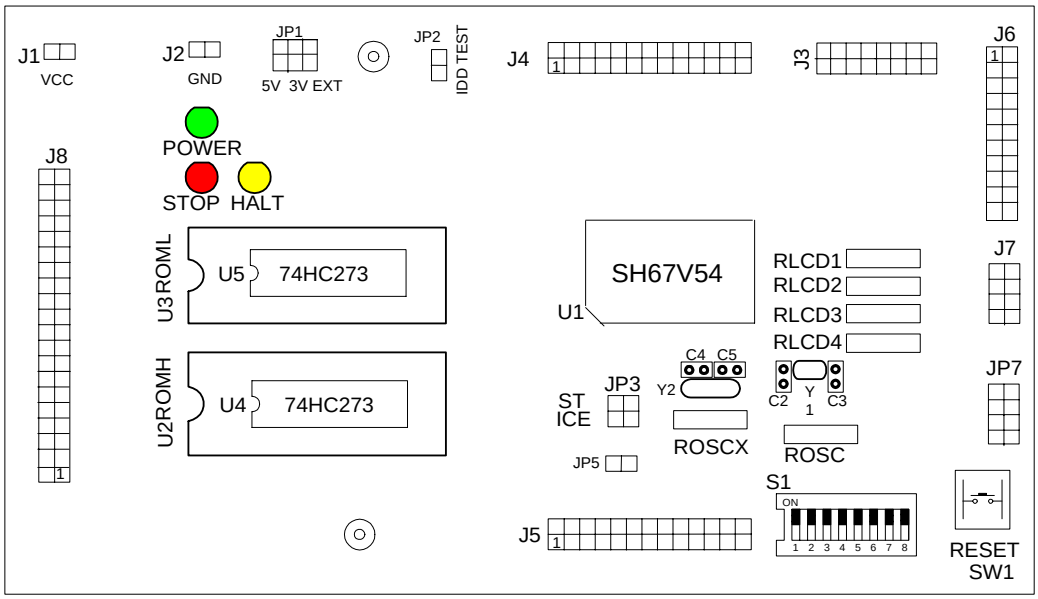
SH67P54/K54/P53/K53 EVB

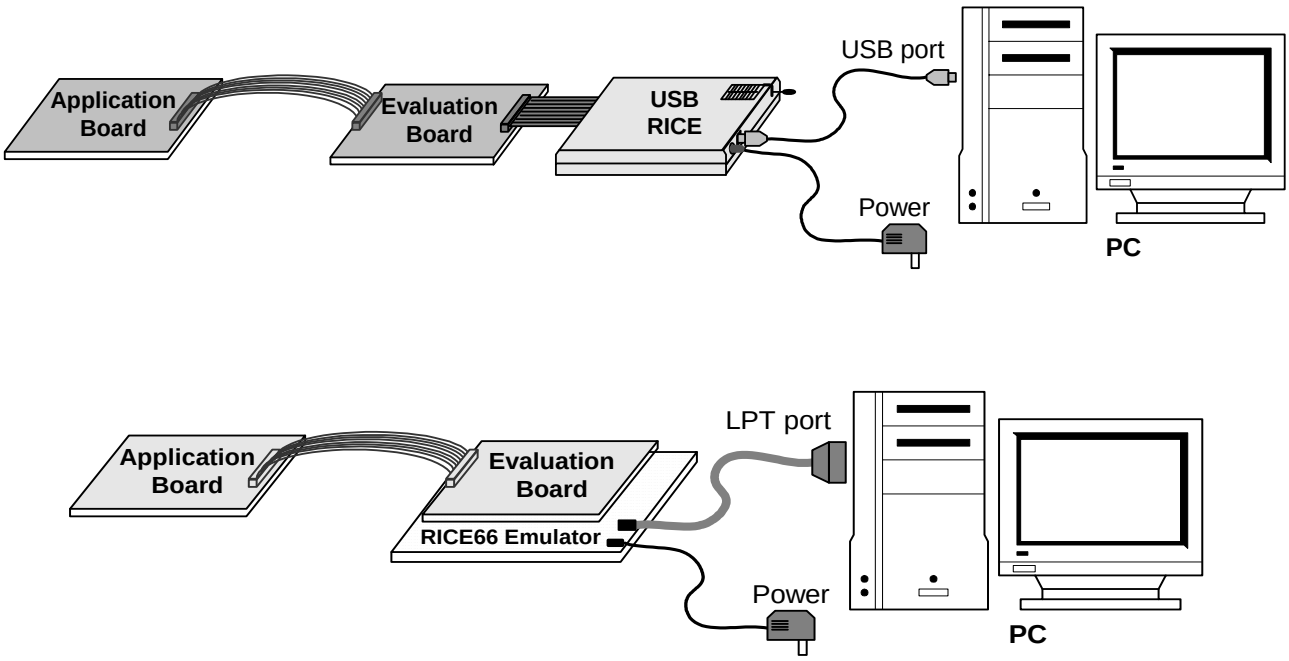
Application Note for SH67P54/K54/P53/K53 EVB

SH67P54/K54/P53/K53 Evaluation board (EVB)

The SH67P54/K54/P53/K53 EVB is used to evaluate the SH67P54/K54/P53/K53 chip's function for the development of application program. It contains of a SH67V54 chip to evaluate the functions of SH67P54/K54/P53/K53. The following figure shows the placement diagram of SH67P54/K54/P53/K53 EVB.

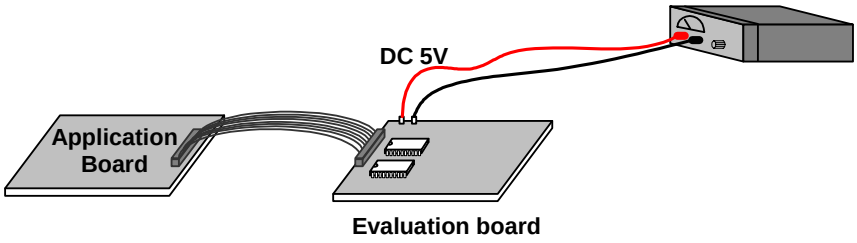


There are two configurations of SH67P54/K54/P53/K53 EVB in application development: ICE mode and stand-alone mode. In ICE mode, the ICE (motherboard) is connected to the EVB by ICE interface.



(a) ICE mode

In standalone mode, the EVB is no longer connected to the motherboard, but the Flash (or EPROM) must be inserted to the socket that stored the application program.



(b) Stand-alone mode

The process of program evaluation on SH67P54/K54/P53/K53 EVB

User can use **Sino Wealth Rice66 Integrated Development Environment (IDE)** to emulate the program and produce the obj file. Rice66 IDE is a real-time in-circuit emulator program. It provides real-time and transparent emulation support for the SH6X series 4-bit micro-controller. And integrate assembler can create binary (*.obj) file and the other files.

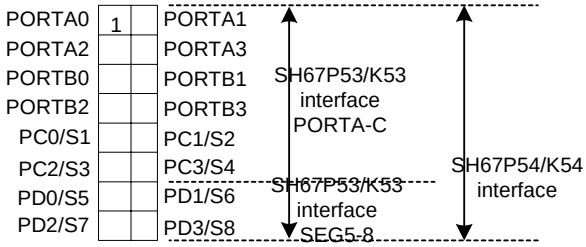
Use Flash (or EPROM) In standalone mode

Rice66 IDE is built-in with an object file depart function. The command ”Split object file” can separate the one 16 bits object file into two 8 bits files, which contain the high and low bytes respectively.

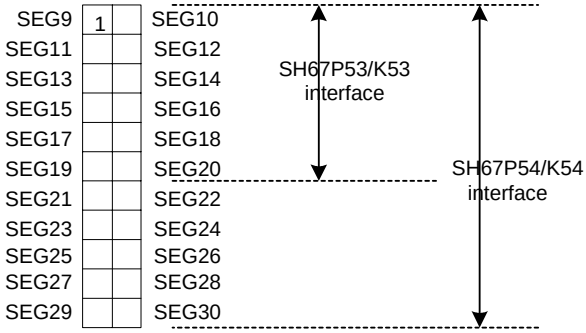
Write the high/low byte obj file to Flash (or EPROM) and insert them to EVB (ROMH and ROML). Then, user can evaluate the program in standalone mode.

SH67P54/K54/P53/K53 interface connector:

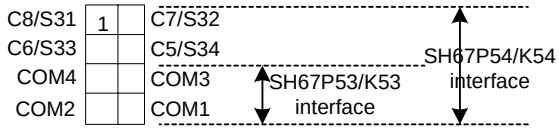
IO port interface: J3 (TOP View from EVB)



LCD interface:**J6 (TOP View from EVB)**



LCD interface:**J7 (TOP View from EVB)**



External VCC input for stand alone mode:

J1, J2 -The external power input when the EVB worked in stand-alone mode. The voltage of Vcc must be 5V±5%.

Interface to ICE:

J4, J5 -connect to RICE66 2.0, or connect to RICE66 3.0 with a transition board.
J8 -connect to RICE66 3.0 directly.

Interface to test the EV chip operating current

JP2 - User can test the EV chip current through JP2

Note: In ICE mode, the current value is correct only when the RICE66 runs in external clock from EVB mode. (Select the "external clock from EVB" in OSC Frequency Config manual.)

Switch setting:

S1:

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Remarks
BD1	BD0	LPD1	LPD0	WDT	LCD/LED	OSC1	OSC0	
X	X	X	X	X	X	X	ON	262K RC oscillator
X	X	X	X	X	X	X	OFF	32768Hz Crystal oscillator
X	X	X	X	X	X	ON	X	4MHz - 8MHz
X	X	X	X	X	X	OFF	X	400KHz – 4MHz
X	X	X	X	X	ON	X	X	LED Matrix
X	X	X	X	X	OFF	X	X	LCD Driving
X	X	X	X	ON	X	X	X	WDT Enable
X	X	X	X	OFF	X	X	X	WDT Disable
X	X	X	ON	X	X	X	X	LVR Enable
X	X	X	OFF	X	X	X	X	LVR Disable
X	X	ON	X	X	X	X	X	2.5V LVR
X	X	OFF	X	X	X	X	X	4.0V LVR
X	ON	X	X	X	X	X	X	BD0=0
X	OFF	X	X	X	X	X	X	BD0=1
ON	X	X	X	X	X	X	X	BD1=1
OFF	X	X	X	X	X	X	X	BD1=0

Jumper setting:

JP1	EV chip power supply select
Short at 3V position	The power of EV chip is set as internal 3V power source.
Short at 5V position	The power of EV chip is set as internal 5V power source.
Short at EXT position	The EV chip use external power supply that was input from EXT_VDD pin.

JP3	EVB ICE/Stand-alone mode select
Short at Stand-alone position	Select stand-alone mode. (The system clock is provided by the on board oscillator.)
Short at With-ICE position	Select with-ICE mode. (The system clock is provided by the ICE.)

JP5	Overflow
Short	Enable the stack overflow function
Open	Disable the stack overflow function

JP7: Test pins. No connect for user.

S2: Reset the whole system when push the button.

Diagnostic LED:

Power LED: The LED will be turned on when the EVB is powered.

HALT LED: The LED will be turned on when the system is in HALT mode.

STOP LED: The LED will be turned on when the system is in STOP mode.

Oscillator setting:

The SH67P54/K54/P53/K53 EVB supports 2 on board oscillators, one is low frequency oscillator (OSC: 32KHz Crystal or 262KHz RC) and the other is high frequency oscillator (OSCX: 400K –4MHz Ceramic or Internal 4MHz).

Capacitor selection for oscillator

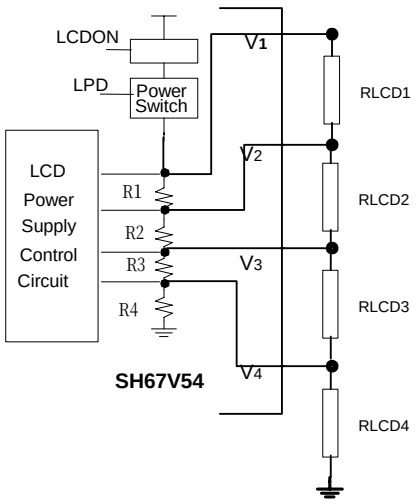
Ceramic Resonators		
Frequency	**C1**	**C2**
455kHz	47~100pF	47~100pF
3.58MHz	-	-
4MHz	-	-

*- The specified ceramic resonator has internal built-in load capacity

Crystal Oscillator		
Frequency	**C1**	**C2**
32.768kHz	5~12.5pF	5~12.5pF
4MHz	8~15pF	8~15pF
8MHz	8~15pF	8~15pF

LCD external bias resistor socket :

RV1, RV2, RV3, RV4 in PCB V1.0; RLCD1, RLCD2, RLCD3, RLCD4 in PCB V1.1.



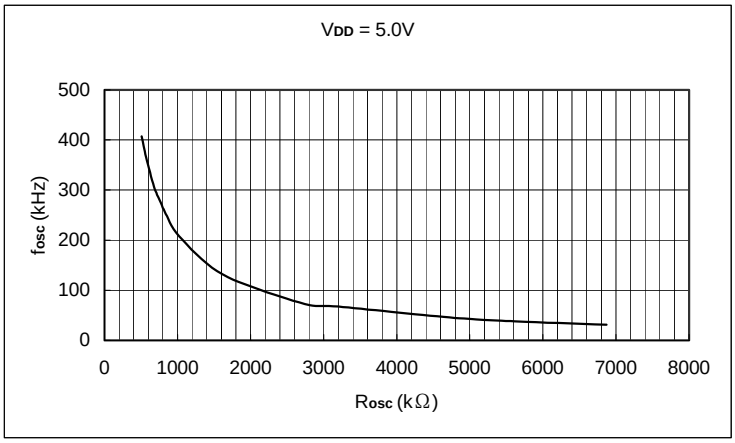
The internal resistors of LCD bias can be selected by software in SH67P54/K54/P53/K53.When large LCD panel is used, user can set the value of $07 to increase the bias current for better LCD performance. But it will cost more power, User can also use external parallel connection resistances for complex bias current.

Notes: To avoid somewhat leakage current, it is necessary to set LCD driver to the 1/4 duty before the system enters a STOP mode.

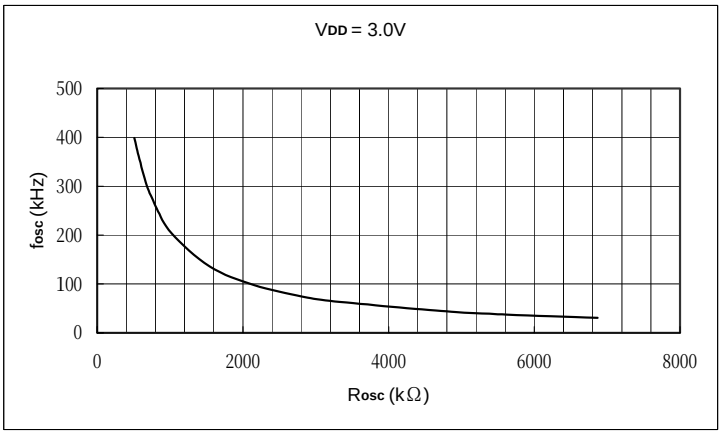
RC oscillator Characteristics Graphs (for reference only)

Typical RC Oscillator Resistor vs. Frequency:

(1) fosc vs. Rosc

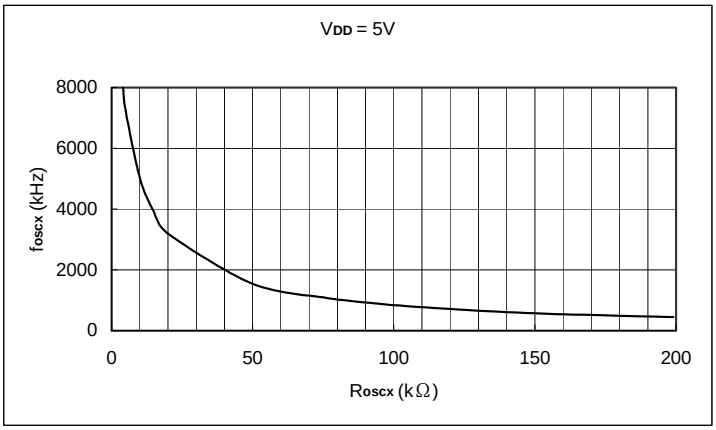


Resistor vs. fOSC, VDD = 5.0V

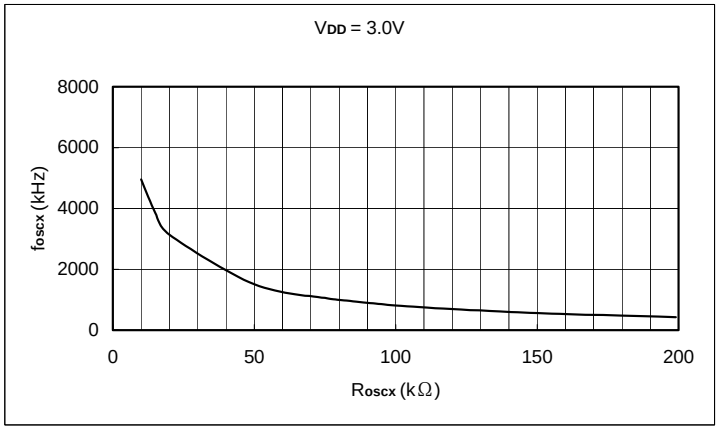


Resistor vs. fOSC, VDD = 3.0V

(2) foscx vs. Roscx



Resistor vs. foscx, VDD = 5.0V



Resistor vs. foscx, VDD = 3.0V

Notes:

1. Application notes:

1.1 After entering into the RICE66 and successfully downloaded the user program, use the F5 key on the PC keyboard to reset the EVB before running the program. If abnormal response occurs, the user must switch off the ICE power and quit RICE66, then wait for a few seconds before restarting.

1.2 When running the RICE66 for the first time, the user needs to select the correct MCU type, clock frequency ... then save the settings and restart RICE66 again.

1.3 Can't Step (F8) or Over (F9) a HALT and STOP instruction.

1.4 Can't emulate the interrupt function in Step (F8) operating mode.

1.5 When you want to escape from HALT or STOP (in ICE mode), please press F5 key on the PC keyboard twice.

1.6 The maximum current limit supplied from EVB to the target is 100mA. When the current in the target is over 100mA, please use external power supply.

2. Programming Notes:

2.1 Clear the data RAM and initialize all system registers during the initial programming.

2.2 Never use the reserved registers.

2.3 Do not execute arithmetic operation with those registers that only have 1, 2 or 3 bits. This kind of operation may not produce the result you expected.

2.4 To add "p=6xxx" and "romsize=xxxx" at the beginning of a program. If any problem occurs during the compilation of the program, check the device and set if it was set correctly.

2.5 Both index register DPH and DPM have three bits; so pay attention to the destination address when using them.

2.6 The clock of the Base timer and LCD driver is sourced by the low frequency oscillator(OSC). When evaluating the LCD and the Base-timer function, the "External from EVB" item in the Config manual should be selected.

2.7 It takes more than 0.3 second to wake up from STOP mode when using 32.768kHz crystal. So, if the system is awake when the key is pressed, the key may have been released when the program starts to read the Key value.

2.8 Notes for interrupt:

2.8.1 Please make sure that the IE flag is enabled before entering into a "HALT" or a "STOP mode. It means that the "HALT" or "STOP" instruction must follow the set "IE" instruction closely.

2.8.2 After the CPU had responded to an interrupt, IRQ should be cleared before resetting IE in order to avoid multi-responses.

2.8.3 Interrupt Enable instruction will be automatically cleared after entering into the interrupt-processing subroutine. If setting IE is too early, it is possible to reenter into the interrupt. So the Interrupt Enable instruction should be placed at the last 3 instructions of the subroutine.

2.8.4 CPU will not respond to any interrupt during the next two instructions after the Interrupt Enable flag be set from 0 to 1.

2.8.5 After CPU has responded to an interrupt, IE will be cleared by the hardware. It is recommended to clear the IRQ at the end of interrupt subroutine.

2.8.6 The stack has four (or eight) levels. If an interrupt is enabled, there will be only three levels that can be used.

2.8.7 It is recommended that the last line of program is "END".

2.8.8 **Interrupt function cannot be simulated in single step mode; it runs properly in full speed mode.**

Examples:

1> Description: CPU can not wakeup after executing the “HALT” or “STOP” instruction.

Program: Interrupt Enable instruction is set outside the interrupt subroutine

<Wrong example>	**<Correct example >**
……	……
LDI IE, 0FH ; enable interrupt	LDI IE, 0FH ; enable interrupt
NOP	~~NOP~~
NOP	~~NOP~~
HALT	HALT

Analysis: After two “NOP” instructions, if an interrupt request comes or IRQ is non-zero during the third instruction cycle, CPU will respond to the interrupt and IE will be cleared. Then when returning to main program, CPU starts to execute “HALT” or “STOP” and will not be activated, because IE is cleared to zero and all interrupts are disabled.
Solution: “HALT” or “STOP” are being followed closely by the “LDI IE, 0FH”

2> Description: CPU responds to one interrupt several times.

Program: Interrupt Enable instruction is placed outside the interrupt subroutine.

```
L1:
    ……
    LDI     IE, 0FH          ; enable interrupts
    NOP
    NOP
    JUMP   L1
```

Analysis: After executing this two “NOP” instructions, and IRQ is not cleared in time, CPU will respond to the interrupt again when it executes the two instructions followed by “LDI IE, 0FH”. This will happen again and again. So CPU responds to one interrupt several times.
Solution: The relative IRQ flag is cleared in time after responding to the interrupt.

3> Description: CPU is running dead in the interrupt-processing program.

Program: an interrupt subroutine.

```
ENTERINT:
    ……
    LDI    IE, 0FH
    NOP
    LDA STACK, 0
    RTNI
```

Analysis: After executing “LDI IE, 0FH” and the following two instructions, an interrupt request comes or the last relative IRQ flag is not cleared in time, then CPU will respond to the interrupt again, so the interrupt is nesting again. When the stack is over 4 levels, it will run into a dead loop.
Solution: Make sure that the CPU can quit from interrupt subroutine within two instruction cycles after interrupt is enabled; After the interrupt is responded, the relative IRQ flag should be cleared before enabling the interrupt.

2.9 Notes for TIMER

2.9.1 When setting the Timer Counter, write first T0L, then T0H.

2.9.2 After setting TM0, T0L, T0H, there is no need to rewrite after the Timer counts overflow, otherwise it will cause a time error every time. The timer is interrupted by that the reload registrar that was set in different time.

2.10 Notes for I/O

2.10.1 Never do the logical operation with the I/O ports. Especially when the I/O ports are connected with the other components externally.

2.10.2 When the internal pull-up resistor is turned on, "1" must be written to I/O Port before Reading.

2.10.3 When counting external pulse, please directly read the PORT status to make sure that the counted number is correct.

2.10.4 The Key De-bounce time is recommended to be 50ms. But in the Rubber Key application, it is best to test Rubber Key's De-bounce time.

2.11 Notes for LCD

2.11.1 To avoid somewhat leakage current, it is necessary to set LCD driver to the 1/4 duty before the system enters a STOP mode.

2.12 Notes for PSG (only for SH67P54/K54)

2.11.2 After PSG finishes playing sound, please reset CH1EN/CH2EN to "0", otherwise it may generate a 32KHz switch signal in PA.1 & PA.2 pin, its frequency is out of the hearing range and consume much more power than normal operation. This situation can be monitor by the green LED D9. If it emit light, it means the PA.1 & PA.2 has sound output or toggling or at a high stage, thus consumes power. So it is better to enable only one PSG channel when we use PSG function to produce one tone. **Don't enable two PSG channels together to produce one tone**, otherwise it will produce some unpredicted errors. For example, the volume sometime will become louder and sometimes will become smaller. (It's caused by the reason of synchronized phases for the two channels.)

2.11.3 It's may be louder if we get the same tones by using two PSG channels, but it will produce some unpredicted errors listed in (a). **You can get the louder tones by tuning the volume register**.

2.11.4 If it is necessary to use 2 channels together (Ex. To play two channel melody), don't let the score always be the same tones as we can do, then the unpredicted errors will not occur or it will be ignore through user hearing.

2.11.5 In ICE mode, there may be some errors when the 455Hz ceramic is used as PSG source clock. To avoid this problem, users should remove the 455Hz ceramic on EV Board, and connect the ICE clock to the input pin of 455Hz ceramic.

2.13 Notes for key scan

2.13.1 If there is no surplus of SEGMENT, it is necessary to use audion to separate the LCD displaying and key scan. If there are more SEGMENTs, which can be directly used as scanning output line, then each SEGMENT must have pull-up resistors. The value of the resistors should not be too small, and it should be about 2M.

2.13.2 To reduce the effect to the maximum extent that SEGMENT as I/O port be on LCD displaying, data stored in the corresponding memory of scanning port should be pre-processed first, then SEGMENT is switched to the I/O state and the state of input port is immediately read before the state of SEGMENT is switched back again. Last the data of keyboard is processed in the keyboard-scanning program. This will shorten the time of Segment in the I/O state, and minimize the effect on LCD. (Duty ratio of key scanning in the whole period of LCD displaying should be reduced as possible as we can).

2.13.3 If two or more keys are pressed down simultaneously, two or more short-circuits of Segment will appear. And if electrical levels of the Segments are not equal, conflicts among them will appear and will have great effect on LCD displaying. If they are equal, this problem can be avoided. So, the program should set the Segments displaying RAM to 0 or 1 simultaneously in order to equal the electrical level when the Segments used as keyboard scanning lines.

2.13.4 When SEGMENT signal is switched from LCD SEGMENT state to I/O state, the value of corresponding memory $358-$36D (the corresponding memory units of SEGMENT when it is in I/O state) is shown on LCD. If the value is 0, the corresponding place of LCD is lightened and if the value is 1, it isn't. Thus, at the beginning of the program, the value of memory $358-$36D should be set to 1, this can reduce LCD flicker when entering the keyboard scanning program.

Application notes Revision History

Revision No.	History	Date
2.1	Change the format based on the new model	Feb.2006
2.0	Add "Programming Notes 31" for LCD driver mode setting	Jan.2006
1.0	Original	Sep.2004