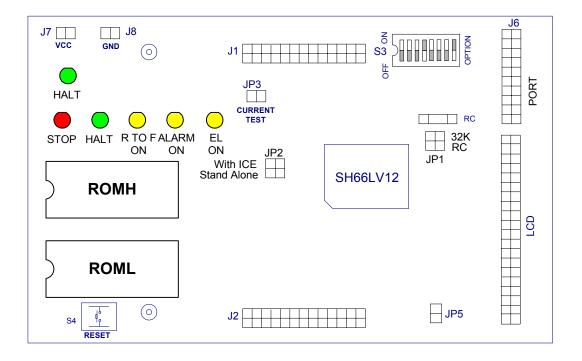**SINO WEALTH**

SH66L12 EVB

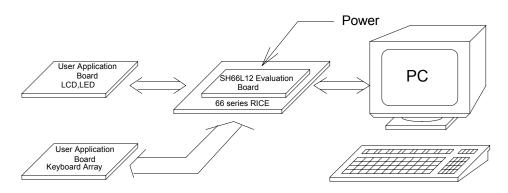## Application Notice for SH66L12 EVB

SH66L12 EVB

The SH66L12 EVB is used to evaluation the SH66L12 chip's function for the development application program. It contains of a SH66LV12 chip to evaluation the functions of SH66L12 including the R2F, EL driver and the LCD waveform. The following diagram shows the placement of SH66L12 EVB.
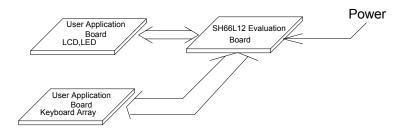

There are two configurations of SH66L12 EVB in application development: ICE mode and stand-alone mode.
In the ICE mode, the SH66xx ICE (motherboard) is connected to the SH66L12 EVB by the ICE interface.

(a) ICE mode

In the standalone mode, the SH66L12 EVB is no longer connected to the motherboard. But the EPROM chip (27512 or 27256) which store the application program must be placed in the sockets of EPROM.

(b) Stand-alone mode

The process of your program's evaluation on SH66L12 EVB

Uasm66.exe: assemble the program, and get binary (*.obj) file and the other files.
Depart the one 16 bits obj file to the two 8 bit files by convert.exe.

Usage example (for example: aaa.asm):
    1.Run the SH66 series assemble program:
     C: >uasm66 /p 66L12 aaa.asm ; to produce the obj file: aaa.obj
    2. Depart the aaa.obj to two 8 bits files aaah.obj and aaal.obj, for example:
     C: > convert
        Input the 16 bits (.obj) file aaa.obj
        Then aaah.obj and aaal.obj will be created.
    3. Write the aaah.obj to EPROM (ROMH)
     Write the aaal.obj to EPROM (ROML)
    4. Put the two EPROMs (ROML and ROMH) into the EPROM sockets.

SH66L12 EVB Programming Notices

1. Clear data RAM and initialize all system registers at the beginning.
2. Do not perform logical operation with I/O ports. Especially when the I/O ports have external connections.
3. Do not perform arithmetic operation with those registers only have 1, 2 or 3 bits. This kind of operation may not get the result you expected.
4. Never use reserved registers.
5. If "IE" instruction (interrupt enable) is set outside the interrupt processing program and there is "HALT" or "STOP" instruction, this two instructions should be followed "IE" instruction closely.
6. After CPU responding to an interrupt, IRQ should be cleared before resetting IE in order to avoid many responses to one interrupt.
7. Interrupt Enable instruction will be automatically cleared after entering interrupt-processing program. If setting IE too early, there is a possibility of re-entry the interrupt. So the Interrupt Enable instruction should be placed at the end and followed closely by two instructions include "RTNI".
8. During the two successive instruction cycles next to Interrupt Enable instruction, CPU will not respond to any interrupts.
9. After CPU responding to interrupts, each bit of IE will be cleared by hardware while IRQ should be cleared by software.
10. It is necessary to add NOP before and after the HALT instruction, else the CPU will execute error instruction when it wakes up from the HALT.

    .
    . .
    NOP
    HALT
    NOP
    .
    . .

11. It is wise to set the Interrupt Enable flag before you return from subroutine in two instructions.

    .
    . .
    LDI   IE, 04H;   Enable timer0 interrupt
    LDA   Temp,0
    RTNI

12. When you set the Interrupt enable flag as the following and your subroutine not be set Interrupt Enable flag, then your system will never wake up if an interrupt entered between the NOPs.

    .
    . .
    LDI   IE, 1111B; IE = Interrupt enable flag
    NOP
    NOP
    NOP
    HALT
    .
    . .

13. To add "p=66L12" or "romsize=2048" at the beginning of a program. If found problem in compiling the program. Check if the SH6566.dev is located at the program directory.
14. When setting Timer Counter, first fill TOL, then TOH.
15. After setting TM0, TOL, TOH, it is unnecessary to reset them after interrupt each time. If TM0, TOL, TOH are reset after each TIMER interrupt, interrupt interval time will not equal because the interrupt timing is not successive.
16. Any instruction containing writing to or reading form memory, it should not be used to operate with I/O Port. It is best to avoid using those instructions such as "SUB, ADD " which do not contain Write operation with I/O Port but have computation operation.
17. "1" must be written to I/O Port before Reading.
18. Writing "1" to I/O Open Drain and then entering "STOP" will cause current leakage ranging from tens to hundreds micro-ampere. So pull up or pull down resistors value from 1 to 2 MΩ must be used to prevent I/O Float when I/O in Open Drain mode.
19. Directly reading PORT states ensure the count is correct.
20. Interrupt activating from STOP at the first time can save power.
21. When the Compiler of old version compiles program, the last line will be read twice. So, if the last line is an instruction, two same operations will be occurred. If there is Label in the last line, compiler will give an error named 'repeated definition'. This will happen in main program or included files and it is recommended that the last line should be a blank line or END.
22. The stack has four layers, if an interrupt is enabled, there only have three layers can be used. Otherwise, if an interrupt comes, the stack will be overflowed that will cause CPU Reset or other errors.

23. Key De-bounce time is recommended to be 50ms. If a user use Rubber Key, it is best to test Rubber Key's De-bounce time.
24. Index register DPH and DPM both have three bits only, so pay attention to the referred address when using them.
25. It takes 0.3 second to wake up from STOP when using 32768Hz crystal. So, if the system is waked up by key pressing, the key may have been released when the program begins to read Key value. Please pay more attention to this problem.
26. Before the program enters "STOP" instruction, LCD should be turned off previously.
27. The "NOP" instruction should be added at the beginning of the program to ensure the IC is stability.

LCD interface connector : (Top View from EVB)

SEG38			SEG37
SEG36			SEG35
SEG34			SEG33
SEG32			SEG31
SEG30			SEG29
SEG28			SEG27
SEG26			SEG25
SEG24			SEG23
SEG22			SEG21
SEG20			SEG19
SEG18			SEG17
SEG16			SEG15
SEG14			SEG13
SEG12			SEG11
SEG10			SEG9
SEG8			SEG7
SEG6			SEG5
SEG4			SEG3
SEG2			SEG1
COM4			COM3
COM2		1	COM1

Port interface connector : J6(Top View from EVB)

PORTA[0]			PORTA[1]
PORTA[2]			PORTA[3]
PORTB[0]			PORTB[1]
PORTB[2]			PORTB[3]
PORTC[0]			PORTC[1]
PORTC[2]			PORTC[3]
PORTD[0]			PORTD[1]
PORTD[2]			PORTD[3]
CX			GND

External Vcc input: J7, J8 (The external power input when the EV. Board worked in stand alone mode. The voltage of Vcc_Ext must be 5V±5%)

Most important:

Incorrect power input (GND connected to VCC pin J7, and VCC connected to GND pin J8) will hurt or break down the EV board.

Jumper setting:

JP2 (SH66L12 EVB ICE/Stand-alone mode select):

- If you short the "With ICE" positions, the clock of SH66L12 EVB is fed from the ICE.
- If you short the "Stand alone" positions (only for stand-alone mode), the clock of SH65V13 is voltage-controlled oscillator and the resistor has been on the EVB.

JP3 (Current test):

- Supply 1.5V voltage to EV chip. (Through pin 1 to pin 2.)
- You can test the current of EV used through JP3.
- Also you can input external voltage (1.2~1.7V) through pin 2 to the EV chip.

Switch setting:

SW1(OPTION):

1.Bit 1 (Alarm frequency select)
   in off position: 4K.
   in on position: 2K.

2.Bit 2 (clock source select)
   in off position: 32K Crystal.
   in on position: 131K RC.

3.Bit 3 (PF0) and bit4 (PF1) are LCD pump frequency select.
   PF0, PF1in off position is 0, on is 1.
   PF0, 1 = 0,0   (4K)
   PF0, 1 = 0,1   (1K)
   PF0, 1 = 1,0   (8K)
   PF0, 1 = 1,1   (2K)

4.Bit 5 (B0) and bit6 (B1) are bonding option.
   B0 in off position is 0, on is 1. The note for OP0 on PCB (Ver1.1) is not correct.
   B1 in off position is 0, on is 1.
   Programmer can read the state of bonding option through the system register $0CH.

5.Bit 7: warmskp switch.
   If switch is on, the warm-up counter's length would be reduced..
6.Bit 8 STACK OVER ON/OFF switch.
   The stack overflow function will on when it turned on.

S4 (RESET):
   Reset the EVB when push the button.

LED declare:

STOP indicate:
   D2 is lighted when the CPU was in STOP mode.
HALT indicate:
   D5 is lighted when the CPU was in HALT mode.
R to F indicate:
   D2 is lighted when R to F is working.
Alarm indicate:
   D4 is lighted when alarm is working.
EL indicate:
   D3 is lighted when EL driver is working.

SH66L12 EVB

Notice:

Evaluate your program with ICE indicate:

1. After enter to RICE66 and successfully download the user program, push the F5 (Reset) on PC keyboard before run your program when you evaluate your program with ICE. If there were abnormal response, the user should push the F5 key again until the ICE work normally.
2. First time run RICE66, need to select an appropriate MCU type, clock frequency ... save the settings and restart RICE66 again.
3. Can't execute Step (F8) or Step over call (F9) while the ICE worked in HALT or STOP mode.
4. When you want to escape from HALT or STOP (in ICE mode), you should press the F5 key on PC keyboard twice.
5. The maximum current limit of the 3V power is 100mA, when the user uses internal 3V power to drive external device such as LED.
6. When EV. Board worked in "with ICE" mode, you can input the clock from EXOSC_IN as the system clock. (refer to the RICE66 User's Guide)