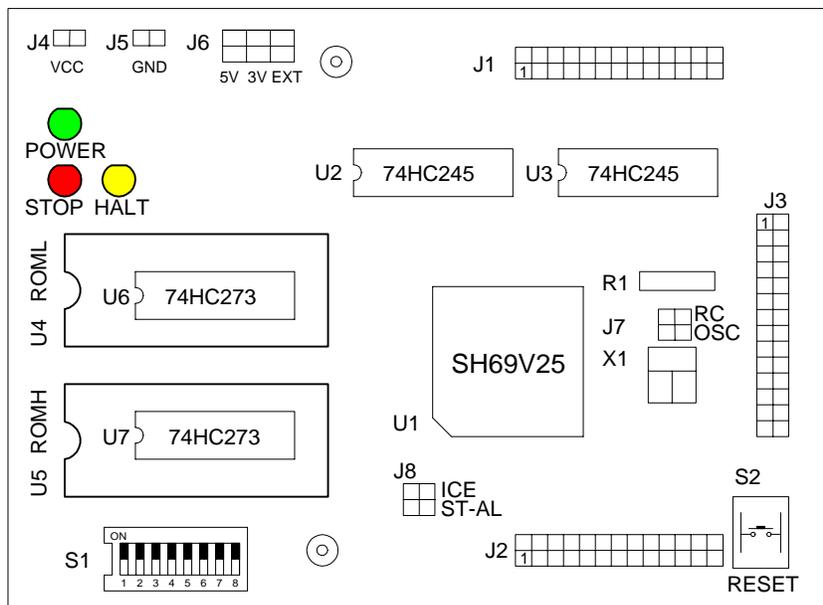# SH69P25/P23 EVB

## Application Notice for SH69P25/P23 EVB

SH69P25/P23 EVB

The SH69P25/P23 EVB is used to evaluate the SH69P25/P23 chip's function for the development of application program. It contains of a SH69V25 chip to evaluate the functions of SH69P25/P23. The following figure shows the placement diagram of SH69P25/P23 EVB.

There are two configurations of SH69P25/P23 EVB in application development: ICE mode and stand-alone mode.
In the ICE mode, the SH66xx ICE (motherboard) is connected to the SH69P25/P23 EVB by the ICE interface.
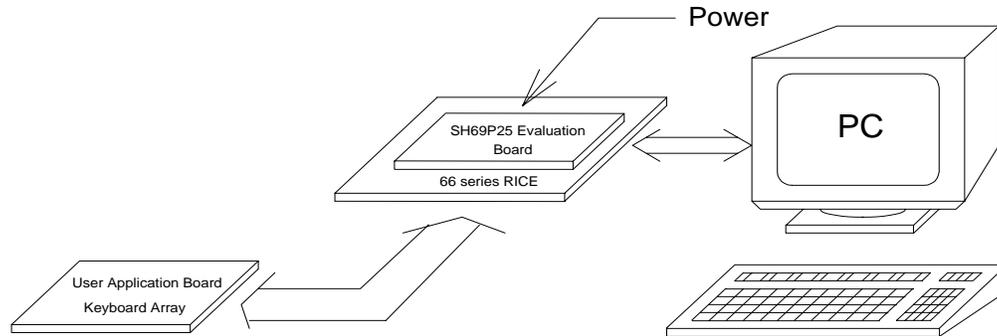
(a) ICE mode

In the standalone mode, the SH69P25/P23 EVB is no longer connected to the motherboard. But the EPROM board has must be connected to the SH69P25/P23 EVB by the EPROM interface. The EPROMs which may be the 27512 or 27256 store the application program;.
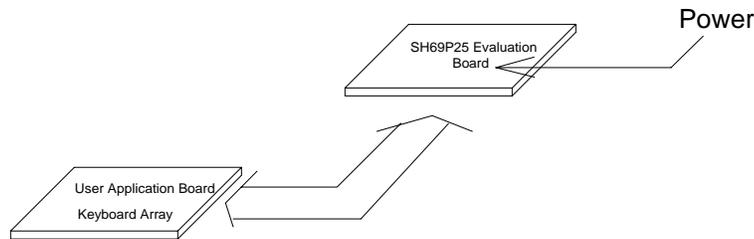
(b) Stand-alone mode

The process of your program's evaluation on SH69P25/P23 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 bits files by convert.exe.

Usage example (for example: aaa.asm):
    1.Run the SH66 series assemble program:
     C: >uasm66 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) on the EV board U4,U5.

SH69P25/P23 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 or 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 Interrupt Enable flag before you return from subroutine in two instruction.
    .
    . .
    LDI   IE, 04H;   Enable timer0 interrupt
    LDA   Temp,0
    RTNI
12. When you set Interrupt enable flag as the following and your subroutine do not set Interrupt Enable flag, then your system will never wake up if an interrupt entered between the NOPs.
    Sample: incorrect use!
    . .
    LDI   IE, 1111B; IE = Interrupt enable flag
    NOP
    NOP
    NOP
    HALT
    .
    . .
13. To add "p=66xx" and "romsize=xxxx" at the beginning of a program. If any problem is found 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 from 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. When the internal pull-up resistor is on, "1" must be written to I/O Port before Reading. And when the internal pull-low resistor is on, "0" 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 M 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.
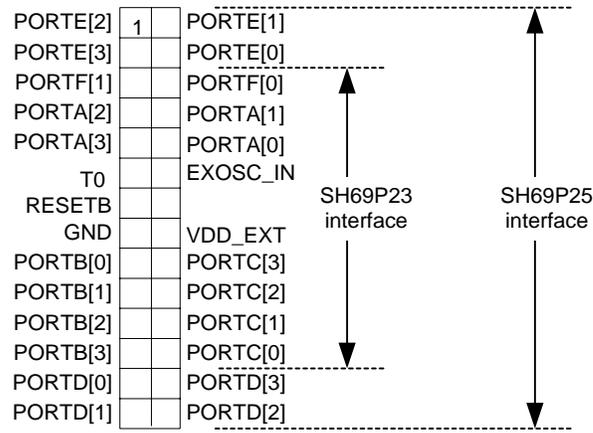
22. The stack has four layers, if an interrupt is enabled, there only have three layers can be used. Otherwise, if an interrupt comes, the stack will be overflowed that will cause CPU Reset or other errors.

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 about 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. The "NOP" instruction should be added at the beginning of the program to ensure the IC is stability.

27. When SH69P25/P23 EVB is used to evaluate the SH69P23, $0B-$0C must be always kept as 00h and $19-$1A must be always kept as 0Fh in user's program.

28. SH69P25/P23 EV board is designed for both SH69P25 and SH69P23.

SH69P25/P23 interface connector: J3 (TOP View from EVB)

PORTE[2]	1	PORTE[1]	
PORTE[3]		PORTE[0]	
PORTF[1]		PORTF[0]	
PORTA[2]		PORTA[1]	
PORTA[3]		PORTA[0]	
T0		EXOSC_IN	
RESETB			SH69P23 interface / SH69P25 interface
GND		VDD_EXT	
PORTB[0]		PORTC[3]	
PORTB[1]		PORTC[2]	
PORTB[2]		PORTC[1]	
PORTB[3]		PORTC[0]	
PORTD[0]		PORTD[3]	
PORTD[1]		PORTD[2]	

External Vcc input : J4,J5 (The external power input when the EV. Board worked in stand alone mode. The voltage of Vcc must be 5V±5%)

Jumper setting:

J6 (EVB power select):

- ■ Short the 5V positions, the voltage (5V) of SH69V25 is internal source.
- ■ Short the 3V positions, the voltage (3V) of SH69V25 is internal source.
- ■ Short the EXT positions, the external power (2.4~6.0V, refer to SH69P25 or SH69P23 spec) of SH69V25 is input from VDD PIN of J3 (J3/16).

J8 (SH69P25/P23 EVB ICE/Stand-alone mode select):

- ■ If you short the "ICE" position (only for ICE mode), the clock of SH69P25/P23 EVB is fed from the ICE.
- ■ If you short the "Stand alone" positions (only for stand-alone mode), the system clock is selected by S1(bit 6, 7, 8) and J7.

Switch setting:

S1:

1. If you turn on bit 1 (STKOF) of this switch, then **enable the stack overflow** function in ICE mode; else disable it.
2. If you turn on bit 2 (LPDV) of this switch, then select **the low level LPD(2.5V)**, else select **the high level LPD(4V)**.
3. If you turn on bit 3 (LPD) of this switch, then **enable the LPD** function, else **disable the LPD** function
4. If you turn on bit 4 (WDT) of this switch, then **enalbe the Watch Dog function**, else **disable the Watch Dog function**.
5. If you turn on bit 5 (OSC3) of this switch, then select the high frequency clock; else select the low frequency clock.
6. OSC2, OSC1, OSC0:

Off,	off,	off	select external clock from OSCI;
On,	off,	off	select RC oscillator;
On,	on,	off	select 400K~4MHz Crystal or ceramic resonator;
On,	on,	on	select 32.768K Crystal resonator;

S2:

Reset the whole system when push the button.

LED declare:

Power: Green LED is lighted when power of the EVB is on.
HALT: Yellow LED is lighted when the system has gone into the HALT mode
STOP: Red LED is lighted when the system has gone into the STOP mode

www 520101. con

SH69P25/P23 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 power off the ICE, quit RICE66 and wait for a few seconds before restart.
2. First time run RICE66, need to select an appropriate MCU type, clock frequency ... save the settings and restart RICE66 again.
3. Can't Step (F8) or Over (F9) a HALT and STOP instruction.
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 (JP3/8) as the system clock. (refer to the RICE66 User's Guide)

Ver 1.0   7/7