Microprocessor Lab-work #1 [II]

Lab. equipment and environments 104-09-01

[1] **Subject and goals**

(a) use of μ-Vision 51IDE residing on MegaWin82G516

(a.1) s/w development context

* assembler, linker, loader under 51μVision

(a.2) h/w trouble shooting

(b) use of program writer("burner")

(c) use of logic analyzer

(d) 51 machine codes observations

[get yourself acquainted with the lab ASAP, and … enjoy the time to come in the next 10 weeks]

[2] **Preparations**

(a) **Refer to the ckt schematic diagram**:

(a.1) for the circuit module(s), as well as constituent circuit components involved with the lab-work.

(a.2) for data path(s) from 51CPU to the target LED module(s)?

(a.3) for associated wiring and layout in the 51μP-PCB

(b) **Datasheets reading**:

(b.1) pin-out and functions of the circuit components

(c) **S/W development environment:**

(c.1) μ-Vision 51IDE manual

(c.2) MegaWin82G516 manual

(d) **Readiness-evaluation:**

Can you or can you not

(d.1) whenever possible, check the target circuit module to see if it's working properly or not by manual wiring the circuitry?

(d.2) carry out trouble shooting along the path way when the lab-work isn't going as expected? How will you do that?

[3] **Lab-work for all:**

(a) Environment check before a GO

(a.1) tools/equipment item check

Each lab team should find on the assigned lab-table the following items; report to TAs immediately for the event of any missing.

** **51μ*P*** experiment circuit board with an adaptor, as well as a bunch of wiring jumpers,

** PC-based logic analyzer with two probing sets, each being with 16 data acquisition channels [presently unavailable],

** USB interconnecting cable for μ-Vision 51IDE under MegaWin82G516.

(a.2) functionality check on

** **51μ*P*** experiment circuit board H/W, especially the target module for the lab-work [see example in [3](a.1)],

** data acquisition channels of the logic analyzer [see example in[3] ,

(b) Labworks to be done

TASK 1	Driving 89c51 experiment circuit board with Megawin MPC82G516 under μ-Vision51IDE ** creating a project for the labwork assignment by setting up μ-Vision51 context parameters properly ** developing the assembly code for the labwork under μ-Vision51 ** replacing 89c51 chip on the expt. circuit board with 51IDE emulator for driving the target board ** running the emulator for code debugging and circuit trouble shooting

[task1.1] refer to the schematic circuit diagram for interconnecting 51IDE and experiment circuit board as described below.

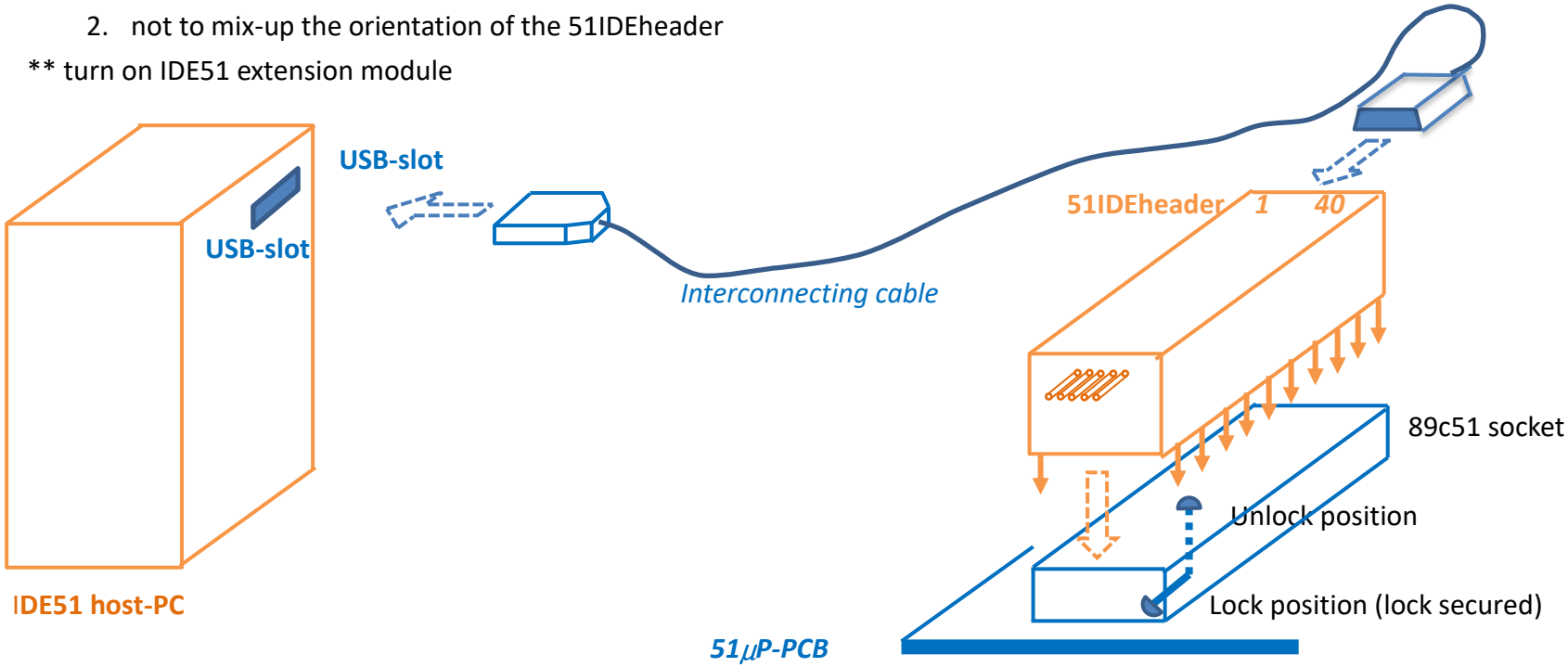
** **power off the circuit board**;

** remove 89c51 chip with care **not to break or bend IC pin-outs**;

** plug 51IDEheader into the 89c51 IC socket with care:

1. not to mess-up pin-out number;
2. not to mix-up the orientation of the 51IDEheader

** turn on IDE51 extension module

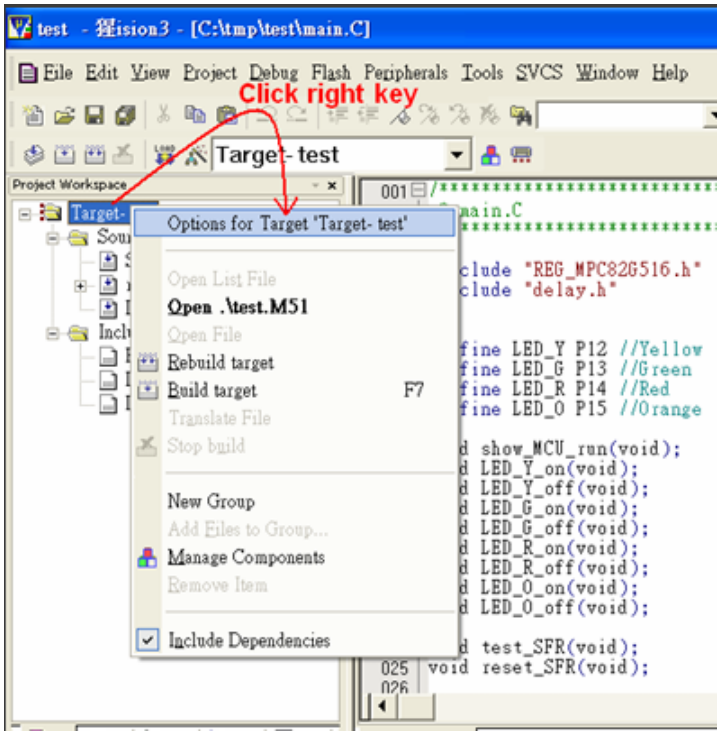


[task1.2] code preparation under 51IDE:

** invoke 51IDE under 51μVision;

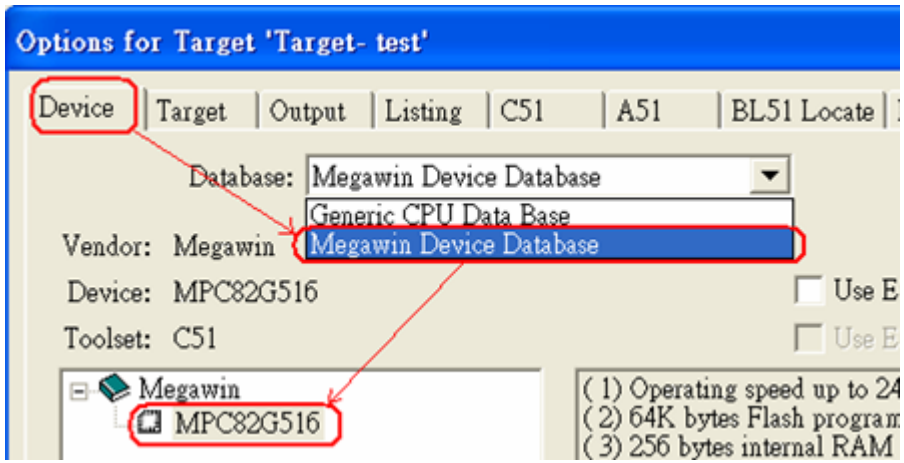
** carry out all environmental settings via 51IDE command list as shown below;

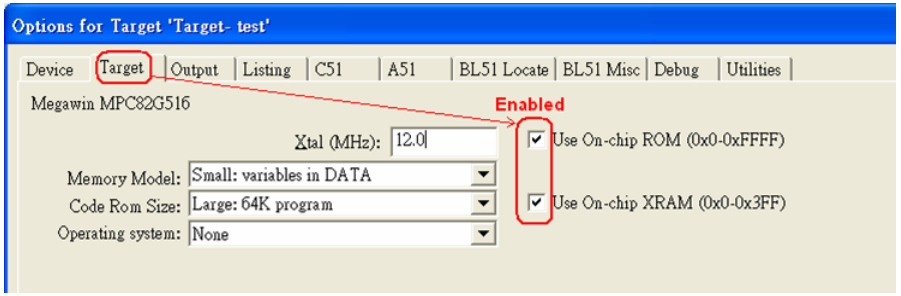
(task1.2.1) create the *target*



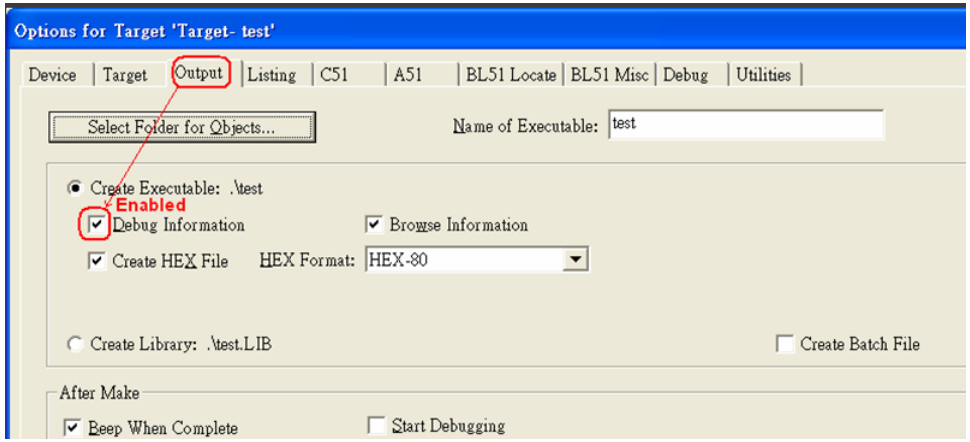
(task1.2.2) target environment setting

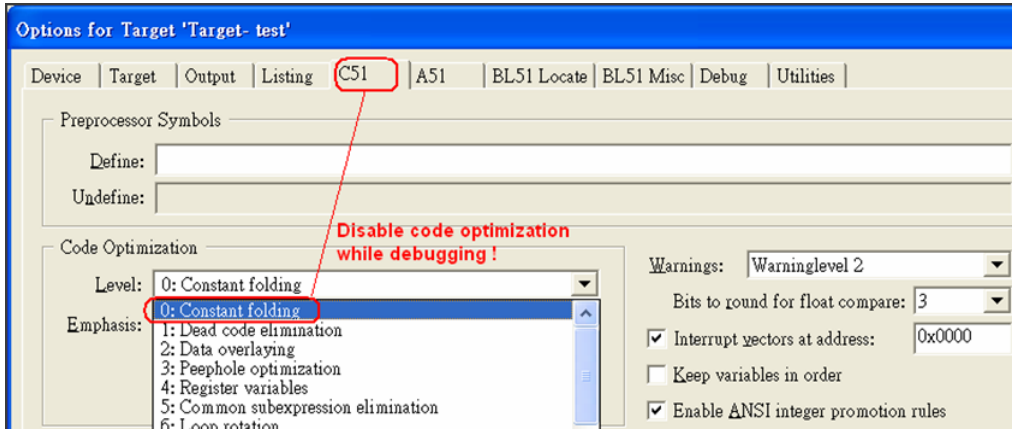
Device	target	output	listing	C51	A51	BL51locate	B51 misc	Debug	Utilities



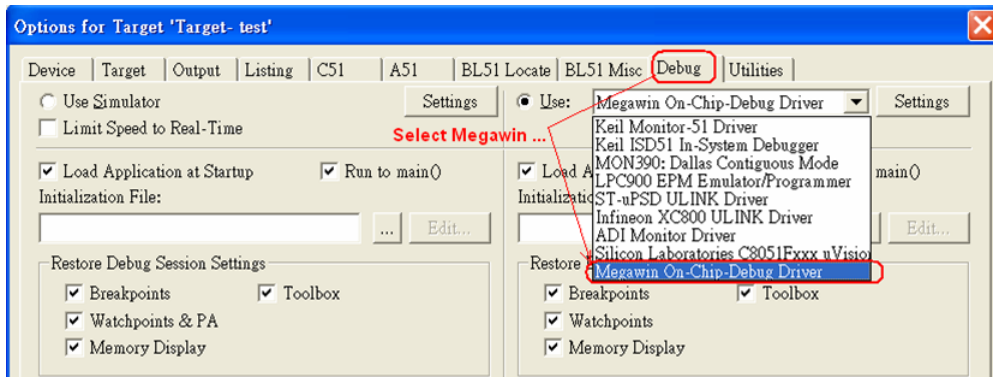


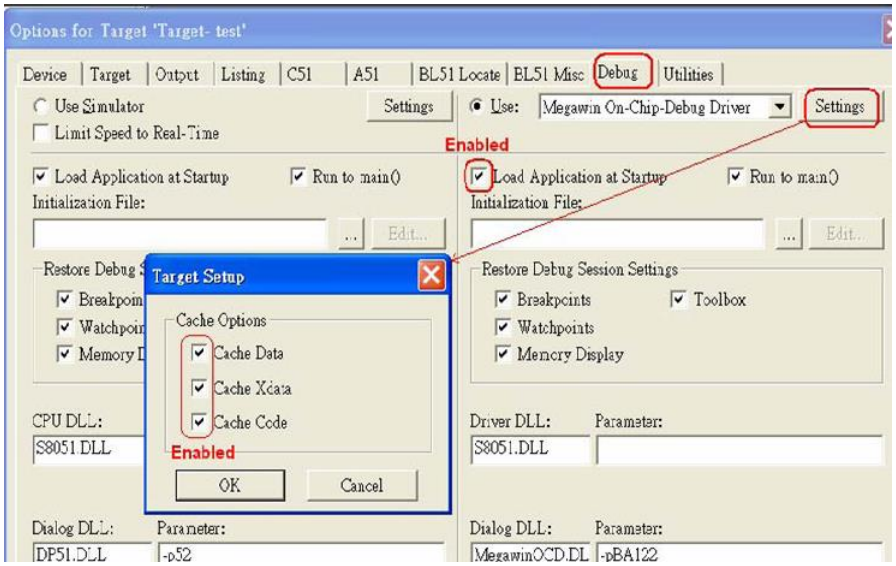
Device	target	output	listing	C51	A51	BL51locate	B51 misc	Debug	Utilities



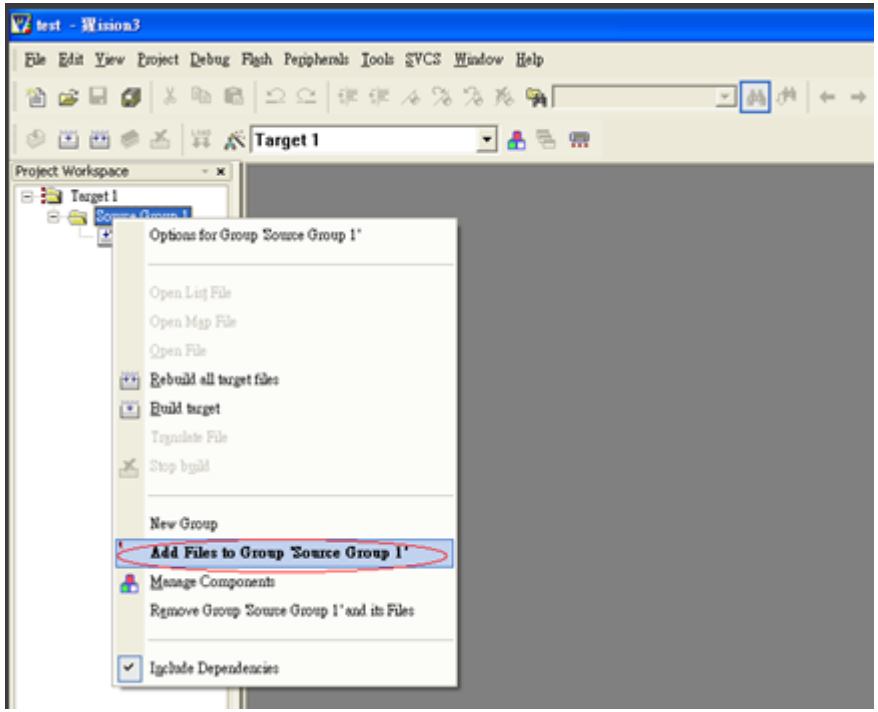


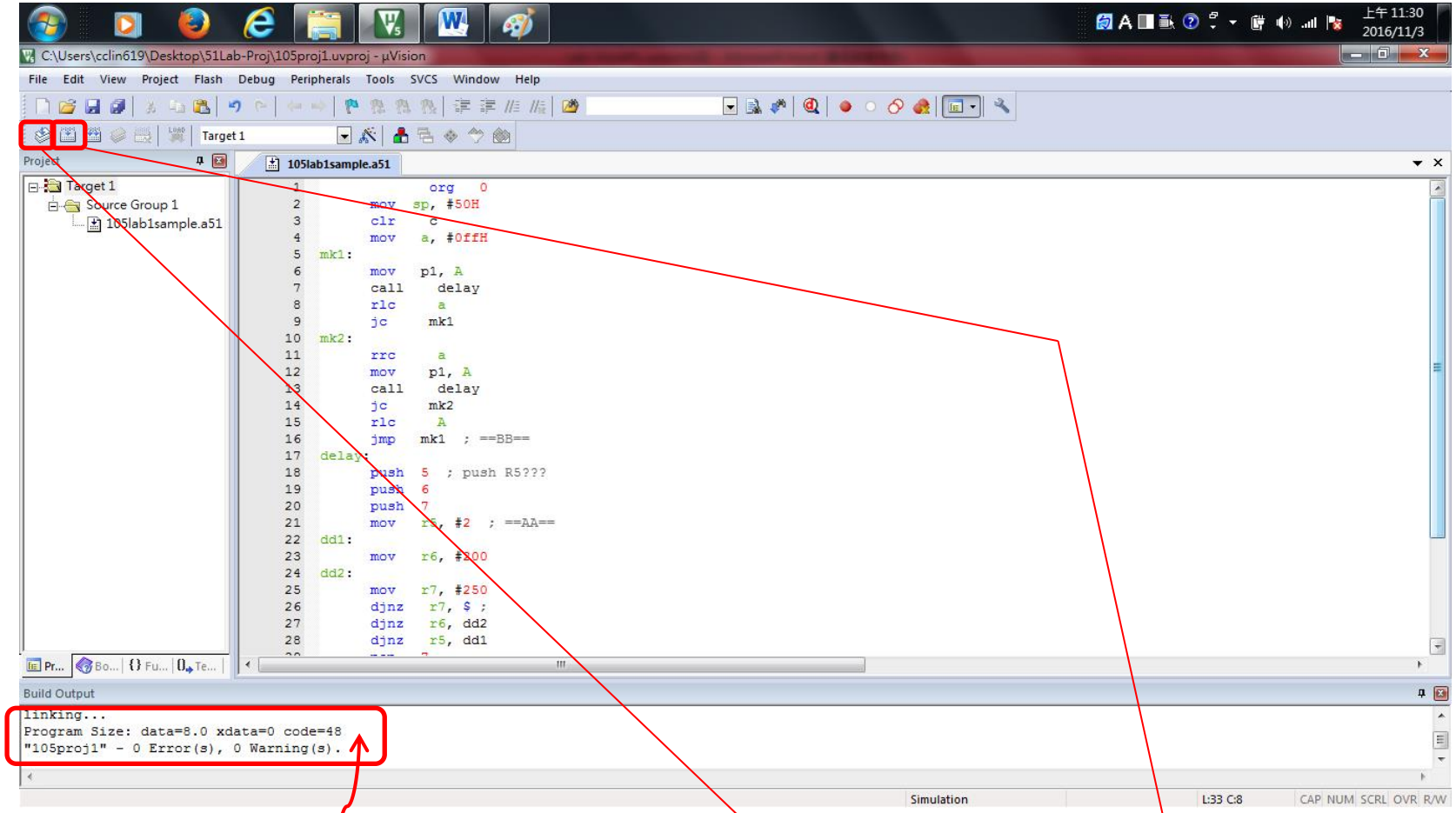
device	target	output	listing	C51	A51	BL51locate	B51 misc	Debug	Utilities





(task1.2.3) edit the following sample 51 assembly codes under 51IDE





syntax-error report

linking report

Translate [code assemblying]

Target Building [linking]

```
        org     0
    mov   sp, #50H
    clr     c
    mov     a, #0ffH
mk1:
    mov     p1, A
    call    delay
    rlc     a
    jc      mk1
mk2:
        rrc     a
        mov     p1, A
        call    delay
        jc      mk2
        rlc     A
        jmp     mk1 ==BB==
delay:
        push    5    ; push R5???
        push    6
        push    7
        mov     r5, #2    ; ==AA==
dd1:
        mov     r6, #200
dd2:
        mov     r7, #250
        djnz    r7, $ ;
        djnz    r6, dd2
        djnz    r5, dd1
        pop     7
        pop     6
        pop     5
        ret
        end
```

(task1.2.4) get the code ready for execution:

1)) invoke the μ-Vision51IDE code development/emulation tool

2)) create a new project for this labwork

** device database setup with megawin MPC82G516

** project target options setting

device	target	output	listing	C51	A51	BL51locate	B51 misc	Debug	Utilities

3)) prepare the source 51 codes of labwork1

4)) invoke 51 assembler

** according to the assembler report, revise the err-plagued code

** repeat with 4)) until bug-free (syntactically)

5)) invoke the emulator (dScope)

** emulation control

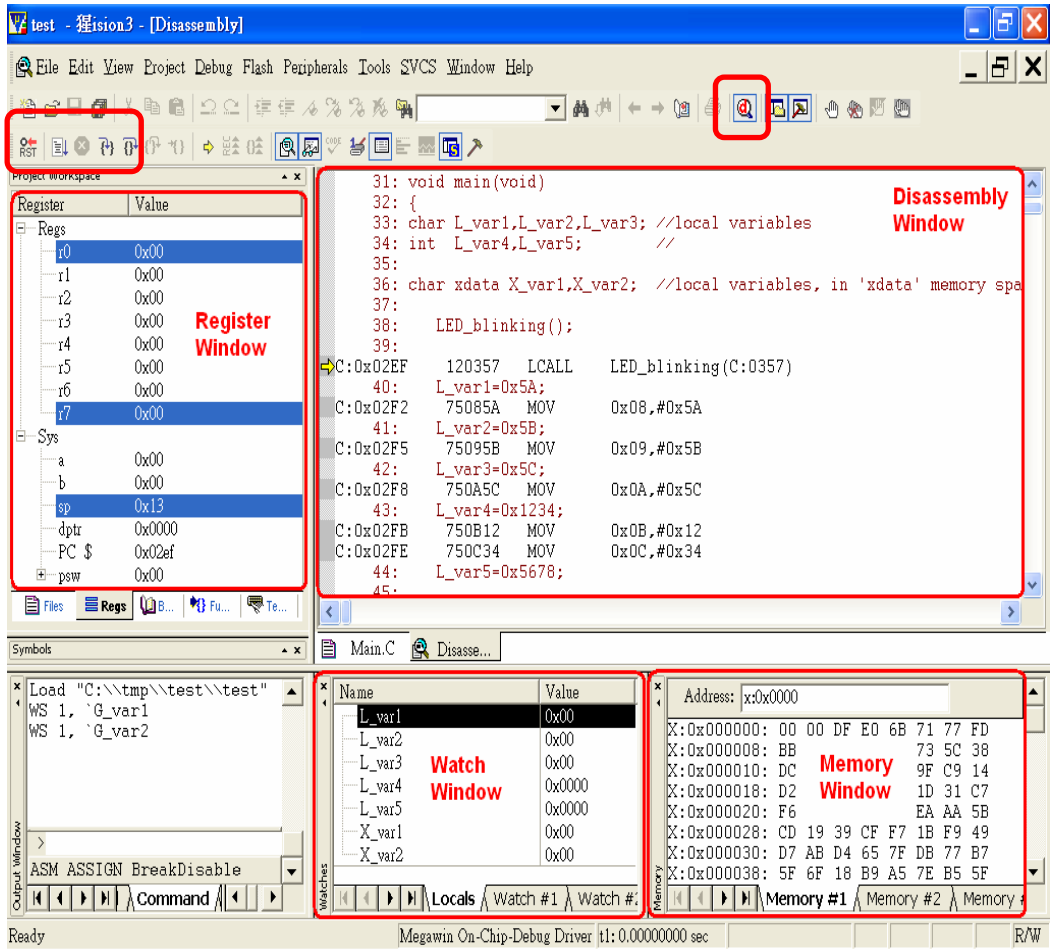
reset	free-run	halt	stepping1	stepping2	break point	. . .

** emulation monitoring

src. lines	machine codes	inst. address	reg. status	C-space contents	D-space contents	X-space contents	run-time update

[task1.3] set up 51IDE for code emulation (after an syntax-error free report from the assembler done with one's assembly source code)

** for code debugging or circuit trouble shooting, invoke ***dScope*** function under 51IDE system, after which the operating context would appear as shown below.

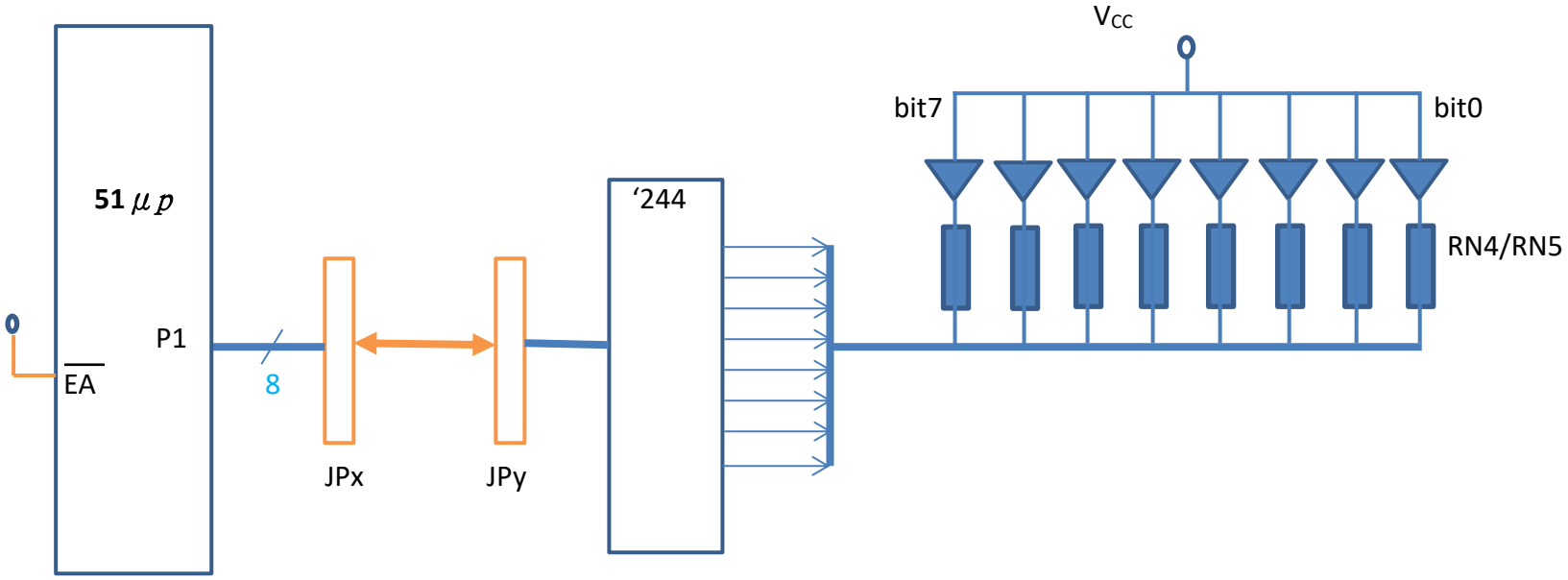


[task1.4] jumper-wiring for LED module setup on the expt. circuit board

** refer to the schematic circuit diagram, do all jumper-wiring necessary for setting up the circuitry as required below.

** whenever doing the jumper-wiring, 51PCB must be powered-OFF.

** whenever plugging an IC chip into or removing an IC chip from the circuit board, 51PCB must be powered-OFF.

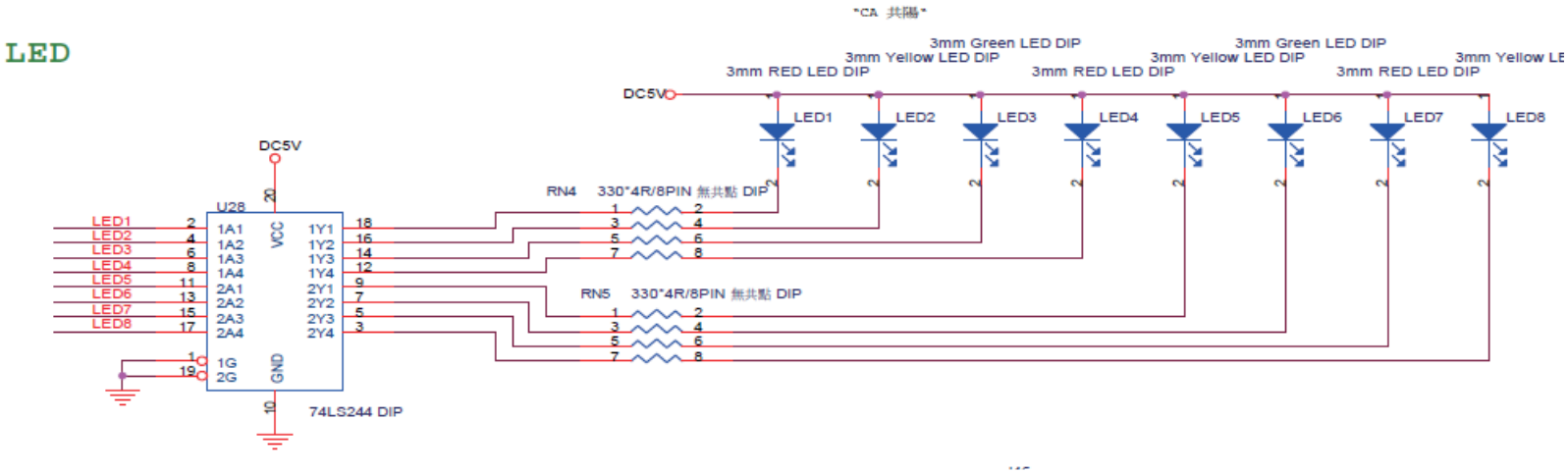


** the ckt block diagram above is a simplified version of the real circuitry; one must consult to the schematic circuit diagram offered by the manufacturer for further details concerning physical wiring in the 51PCB circuit board. For instance,

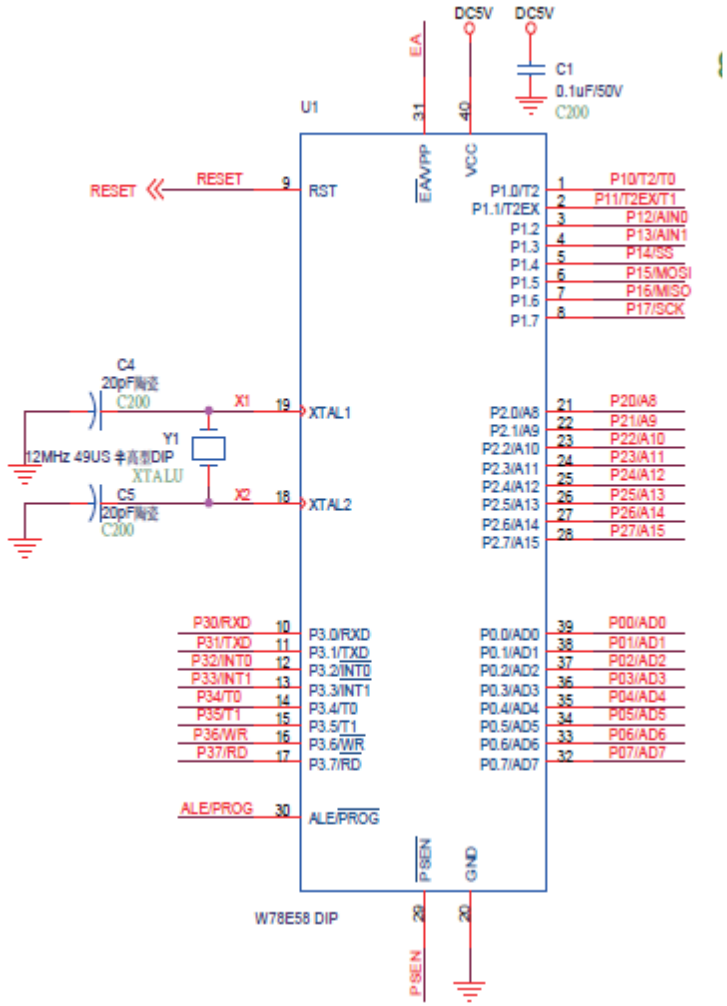
- how P1 and jmper JPx are interconnected?
- how 74244 IC module and jmper JPy are interconnected?
- how 74244 IC and the discrete LED module are interconnected?

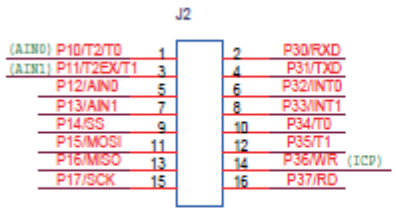
** the operational control on a discrete LED should be clarified (ON, OFF, intensity regulating).

** the functionality of the IC module TTL 74244 should be clarified.

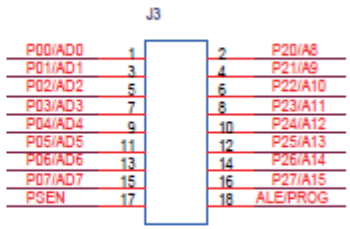


[circuitry in details shown in the datasheet offered by the manufacturer]





2x8Pin180° 2.54mm Male Header DIP



2x9Pin180° 2.54mm Male Header DIP

[task1.5] task execution:

** **power-on** the 51 experiment circuit board

** click appropriate icons in the operating panel for activating proper emulation mode: using 51IDE internal code and data memory for this task

** click appropriate icons for resetting 51IDE, and one should see on the operation panel

1)) code contents or data contents in the memory-window area

2)) 51CPU status in the register-window area

** 51IDE will execute the code in free-run, step-over, or step-into mode when appropriate icon is clicked; LED module, as well as 51CPU status, would react accordingly

** check the machine codes of the source line marked with **==BB==**,

1)) is it a 2-byte or 3-byte machine codes generated by **A51** assembler?

2)) if it's a 2-byte code instruction, is it an **AJUMP** or a **SJUMP** instruction? What's the target address? In absolute or relative format?

3)) if it's a 3-byte code instruction, what's the target address?

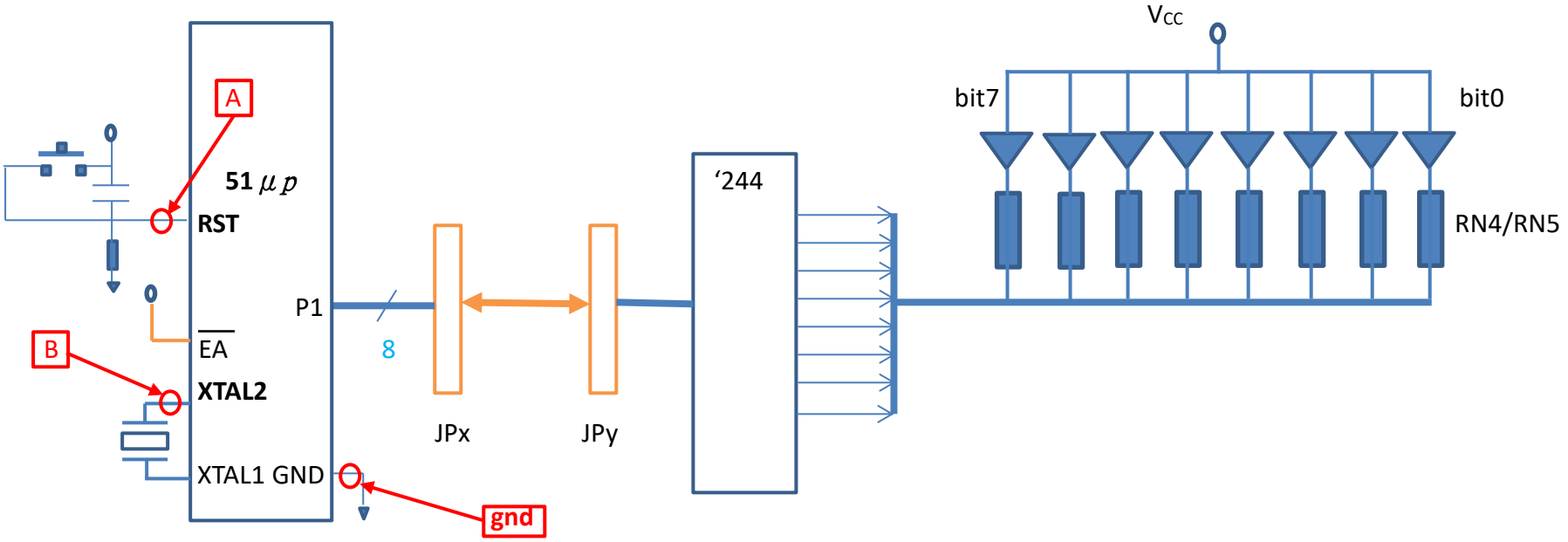
** 1)) click appropriate icons for modifying the code memory such that the source line marked with **==AA==** becomes

```
mov     r5, #10
```

2)) reset 51IDE and execute the modified codes, what's the response of the LED module?

TASK 2 probing signals on a running 89c51 system ** tag data acquisition channels of the logic analyzer to targeted signal lines in the circuit 1)) **RESET** to 89c51 3)) ground of the experiment circuit board 2)) timing clock for 89c51 ** set up the logic analyzer accordingly as required by the task observation at hand ** start running the target system, and then analyzer operation

[task2.1] use the same circuit and codes as set up in **TASK 1**, **power off the circuit board** while tagging the LA probing channels; Refer to the schematic circuit diagram, tag two probing channels respectively at **RST** and **XTAL2** pin-out of 89c51; do not forget to tag the ground of 89c51 circuit board, as depicted below.



[task2.2] logic analyzer setting up

** invoke the logic analyzer system to see the operating panel appear on the screen

** click appropriate icons in the operating panel for activating the two probing channels already tagged to RST and XTAL2 pin-out of 89c51 chip

(i) enable channel ID of the probing channel in use, name the channel (e.g. **RESET** for the line tagged to RST and **CLK** for the line to XTAL2)

(ii) set the triggering mode for each enabled channel, choosing low-to-high or high-to-low transition triggering (or so called edge-triggering) for **RST** acquisition

(iii) set data acquisition mode as one shot

(iv) set other parameters whenever necessary

[task2.3] task execution:

** **power on** the 51 experiment circuit board

** start executing the codes under 51IDE emulation, the LED module should start working as it does in TASK1

** start data acquisition by clicking RUN icon in the control panel of the logic analyzer

** press the reset switch of the circuit board, which would complete the data acquisition process, and the wave forms of **CLK** and **RST** would then appear on the display of the logic analyzer control panel

** save the waveforms on the display in a file for printout later on

TASK 3 programming 89c51 for driving the discrete LED module

** discrete LED module set up

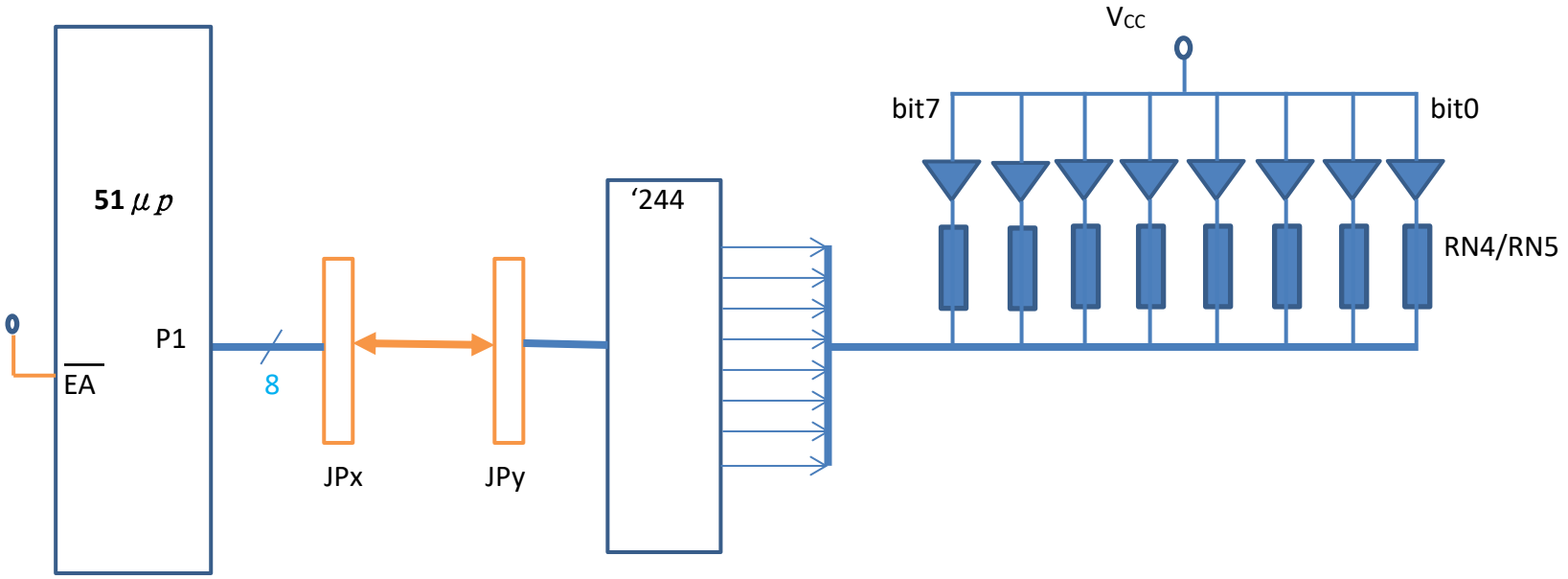
** code preparation

** programming (“burning”) 89c51

** ‘hot’ execution → the 8 discrete LEDs display in leftward and rightward scanning pattern alternately

[task3.1] jumper-wiring for LED module setup

Refer to the schematic circuit diagram, do all jumper-wiring necessary for setting up the circuitry as required below.



[task3.2] code preparation:

** prepare the HEX-file of the codes obtained in (task1.3.4)

[task3.3] programming 89c51:

** make a copy for **your_file.hex** on a flash disk

** use PC-based universal IC programmer (code burner) in the lab for writing **your_file.hex** into built-in code memory of 89c51 as follows.

1)) start the IC programmer system
1)) put 89c51 chip in the IC socket of the burner, secure the lock; care should be taken to avoid improper placement of the chip,
3)) specify the IC model to be programmed: AT89c51 by ATMEL
4)) clear the built-in code memory of 89c51
5)) download **your_file.hex** from the disk
6)) start the 'burning' process
7)) unlock the IC socket, remove 89c51 chip

[task3.4] task execution:

** **power off** the 51 experiment circuit board

** plug the code-programed 89c51 chip into the CPU socket in the expt. circuit board

** start execution by turning on the 51 experiment circuit board, reset the system

** start trouble-shooting if necessary

[**Key**: checking along the data path in a stage by stage manner, from the start: inside of 89c51 to the end: the target module.]

[4] **Observations**

Is the 1st labwork going smoothly? If so, then you've had a quiet and cool night seeing everything exactly as expected in the labwork's setting, yet nothing else more. ☺

If not, my compliments to you for the torturing night you just went through and that you had a chance for getting more practical experiences and knowing a lot more about 51CPU while struggling your way out of the s/w bugs (and/or h/w anomaly). ☺☺

[**Key** to the s/w debugging and h/w trouble shooting:

checking along the data path in a stage by stage manner, from the start: inside of 89c51 to the end: the target module.]

TASK 1

(b.1) Is the labwork running well? Either way, what did you SEE in this very first task work?

(b.2) While modifying the codes of the line marked with **==AA==**, how should care be taken allowing no chance of causing catastrophe?

(b.3) How should care be taken while using 51IDE to modify code bytes corresponding to a JMP instruction? If not properly handled, can you tell about what possible consequences may be?

TASK 2

(b.2) If not using edge-triggering mode, is it possible to capture **RST** signal resulting from the press-release of RESET switch in the circuit board? Why or why not?

[5] **Comprehension evaluation**

Ask yourself three questions regarding the lab-work you just went through.

(a) ?

(b) ??

(c) ???