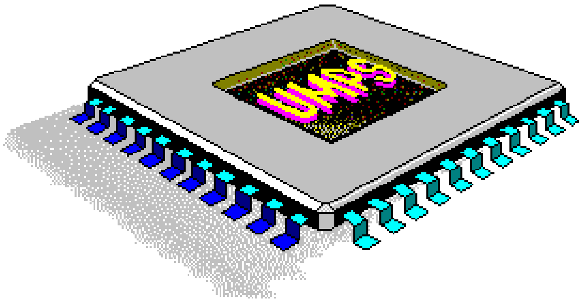
Virtual Micro Design

UMPS Support

Technopole Izarbel

E-Mail: umps@vmdesign.com

URL: http://www.vmdesign.com



UMPS Tutorial
Program simulation

1.BACKGROUND

* DEMO Version:

W-A-R-N-I-N-G

This demonstration version is for evaluation purpose, the limitation of this version allow the user to see ALL UMPS capabilities and no more. More than 40 examples are installed with this version that show UMPS in various situations and with different microcontroller, as well as a small tutorial to guide you in your first step.

Please contact our customer service to order a complete and full version.

* DEMO Version limitation

- some CPU librairies are missing in order to save disk space,
- simulation time is limited to about 800000 cycles
- CPU don't have the full ROM/RAM space (limited to about 256 bytes)
- CPU documentation is not complete
- Demo version does not contain all CPU Libraries.
- all save capabilities are inhibited and not implemented, excepted for:
    - save text,
    - save CPU Code as text.
- demo version will run about 3 months.

DEMO Addendum:

- Program counter is limited as follow:

68HC11:	$F000 <= PC <= $F3FF
HC705:	$0080 <= PC <= $01FF
8031:	$0000 <= PC <= $07F0
PIC:	$0000 <= PC <= $00FF
AVR:	$0000 <= PC <= $0120

- When the maximum CPU demo time is reached, CPU will stop even if there is no reason to stop (Break or INT).

- TUTORIAL:
  Tutorial will ask you to save the project, which is impossible with the demo version. Please forget to save file/project when using tutorial with **demo version.**

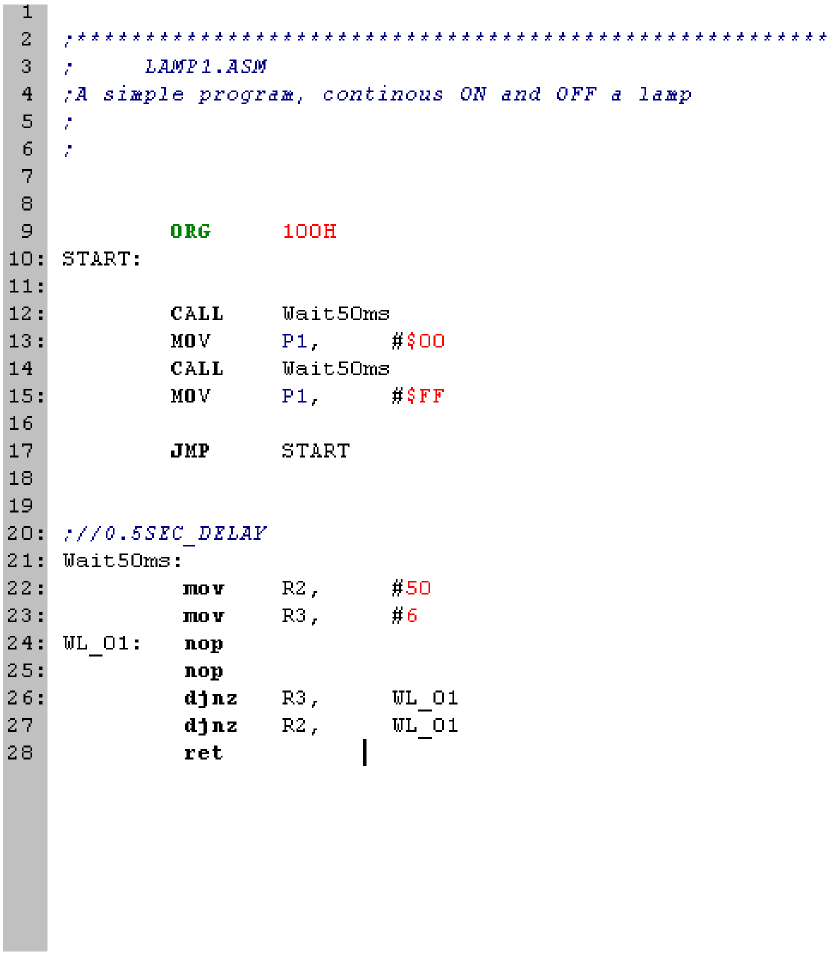
2.LAMP1.ASM

Compiling a program

1. In File menu, choose File Load.
2. Select **Text File** at file format section
3. Then choose LAMP1.ASM, click OK to close dialog box.

   You can new read the ASM source code of the program.
4. Click Compile button in tools bar.
5. In file dialog box, select LAMP1.ASM, then click OK.

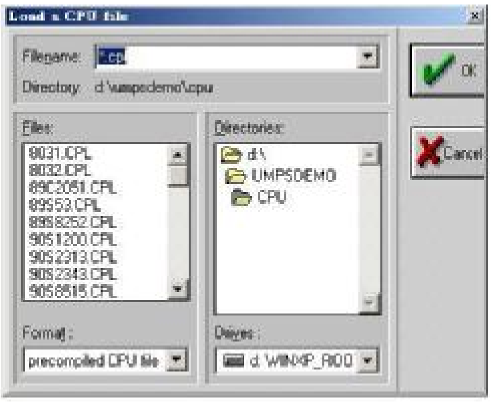
   Program is now compiled.



```

;*******************************************************
;     LAMP1.ASM
;A simple program, continous ON and OFF a lamp
;
;


        ORG     100H
START:

        CALL    Wait50ms
        MOV     P1,     #$00
        CALL    Wait50ms
        MOV     P1,     #$FF

        JMP     START


;//0.5SEC_DELAY
Wait50ms:
         mov    R2,     #50
         mov    R3,     #6
WL_01:   nop
         nop
         djnz   R3,     WL_01
         djnz   R2,     WL_01
         ret
```

Loading a microcontroller

1. In Configure menu, choose Configure Load CPU.
2. In file dialog box, choose **89C2051.CPL** file. Click OK to load



Load a CPU file

Filename: *.cpl

Directory: d:\umpsdemo\cpu

Files:
8031.CPL
8032.CPL
89C2051.CPL
89S53.CPL
89S8252.CPL
90S1200.CPL
90S2313.CPL
90S2343.CPL
90S8515.CPL

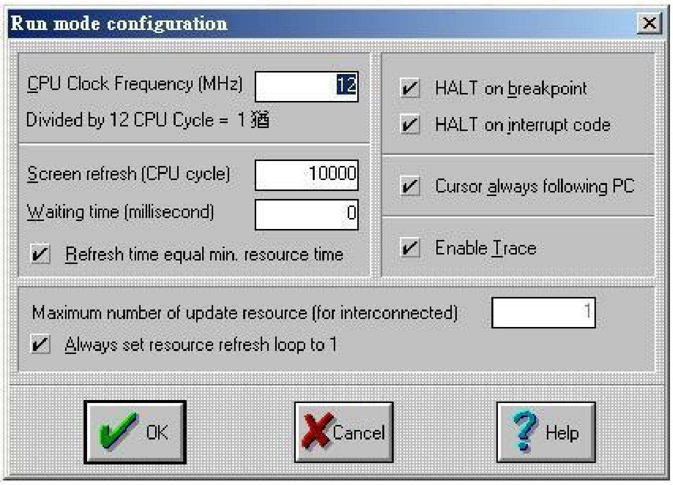
Directories:
d:\
UMPSDEMO
CPU

Format: precompiled CPU file

Drives: d: WINXP_R00

Run mode configuration

1. In Option menu, choose RUN Mode
2. Enter 12 at CPU Clock Frequency. This indicate clock frequency is 12 MHz
3. Enter 10000 at Waiting Time
4. Check Cursor always following PC. Click OK to close the dialog box
5. microcontroller is now configured



Run mode configuration

CPU Clock Frequency (MHz) 12

Divided by 12 CPU Cycle = 1 猶

Screen refresh (CPU cycle) 10000

Waiting time (millisecond) 0

☑ Refresh time equal min. resource time

☑ HALT on breakpoint

☑ HALT on interrupt code

☑ Cursor always following PC

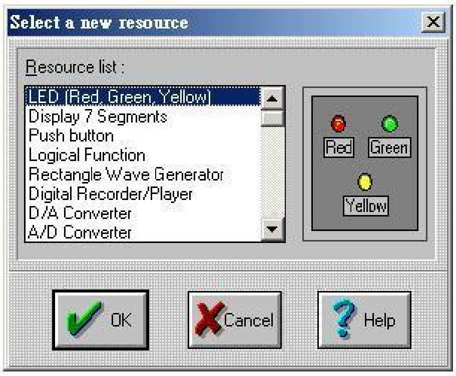
☑ Enable Trace

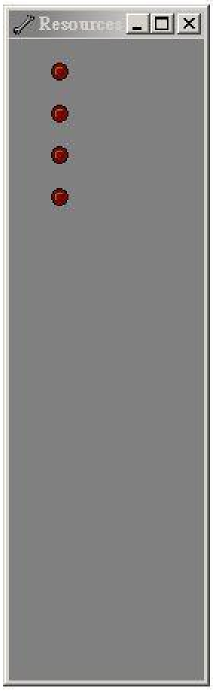
Maximum number of update resource (for interconnected) 1

☑ Always set resource refresh loop to 1

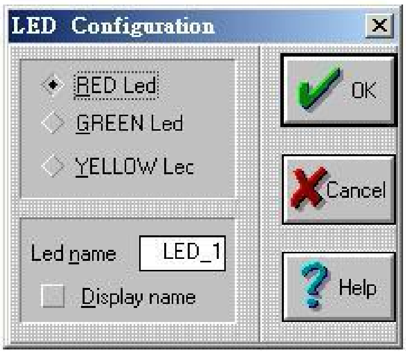
Creating resources

1. In Configure menu, choose Configure Resources.
2. Click Add button in the tools bar then choose **Push button**, click OK to close dialog box.
3. Select the LED and move it down while maintaining left mouse button down





Click Display button in tools bar to display configuration dialog box



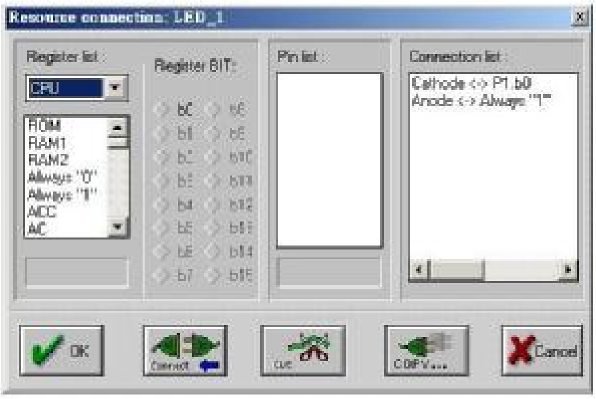
● RED Led
○ GREEN Led
○ YELLOW Lec

Led name LED_1

□ Display name

Connecting resources:

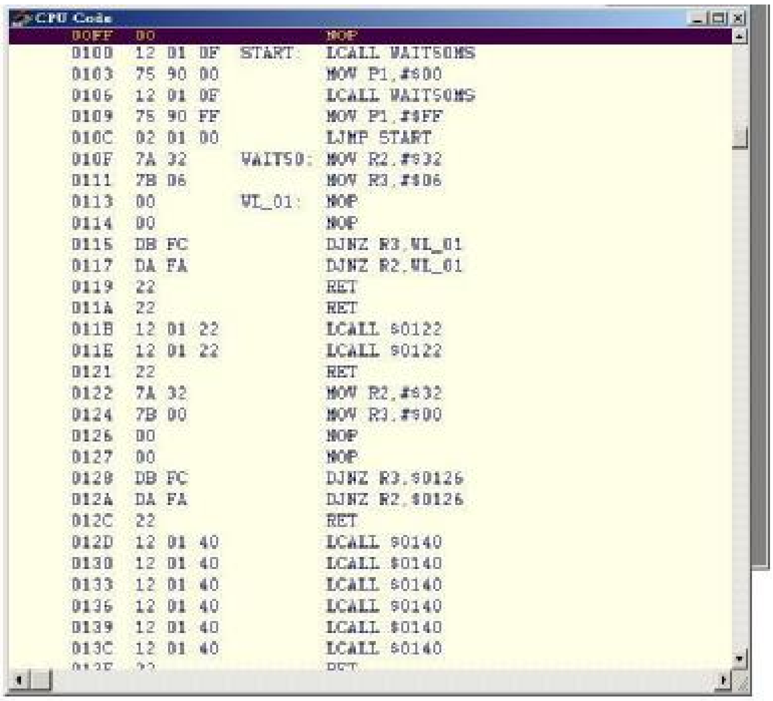
1. Click a LED to select it
2. Click Connect button to open connection dialog box
3. In Register List, select P1
4. In Register Bit, select Bit 0
5. In Pin List, select Cathode.
6. Click Connect
7. In Register List, select Always “1”
8. In Pin List, select Anode.
9. Click Connect , then OK to close dialog box.
10. Repeat step 1 to 9 for each LED
11. Click Exit button in tools bar to exit configuration mode.



Cathode <-> P1.b0
Anode <-> Always "1"

Display assembly code:

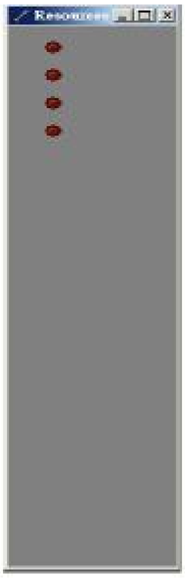
1. In View menu, choose CPU Code. ROM content is displayed with disassembled code.
2. Press ENTER key can change to overwrite mode.

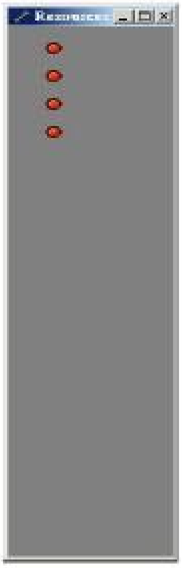


```
00FF  00                  NOP
0100  12 01 0F  START:    LCALL WAIT50MS
0103  75 90 00            MOV P1,#$00
0106  12 01 0F            LCALL WAIT50MS
0109  75 90 FF            MOV P1,#$FF
010C  02 01 00            LJMP START
010F  7A 32     WAIT50:   MOV R2,#$32
0111  7B 06               MOV R3,#$06
0113  00        WL_01:    NOP
0114  00                  NOP
0115  DB FC               DJNZ R3,WL_01
0117  DA FA               DJNZ R2,WL_01
0119  22                  RET
011A  22                  RET
011B  12 01 22            LCALL $0122
011E  12 01 22            LCALL $0122
0121  22                  RET
0122  7A 32               MOV R2,#$32
0124  7B 00               MOV R3,#$00
0126  00                  NOP
0127  00                  NOP
0128  DB FC               DJNZ R3,$0126
012A  DA FA               DJNZ R2,$0126
012C  22                  RET
012D  12 01 40            LCALL $0140
0130  12 01 40            LCALL $0140
0133  12 01 40            LCALL $0140
0136  12 01 40            LCALL $0140
0139  12 01 40            LCALL $0140
013C  12 01 40            LCALL $0140
013F  22                  RET
```

Execute Program

1. In Program menu choose Reset. CPU core is now reset
2. Click Trace into button to execute on instruction. Look at RAM content and also register window which are changing
3. Click GO button to execute the program

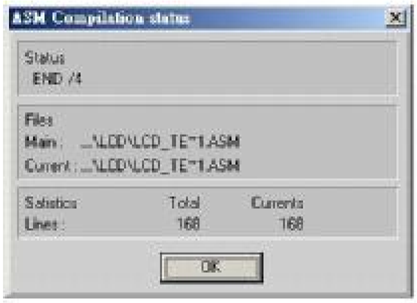




3.LCD_TESTV1.ASM

Compiling a program

1. In File menu, choose File Load.
2. Select **Text File** at file format section
3. Then choose LCD_TESTV1.ASM, click OK to close dialog box.
   You can new read the ASM source code of the program.
4. Click Compile button in tools bar.
5. In file dialog box, select LAMP1.ASM, then click OK.
   Program is now compiled.



Status
END /4

Files
Main : ...\LCD\LCD_TE~1.ASM
Current : ...\LCD\LCD_TE~1.ASM

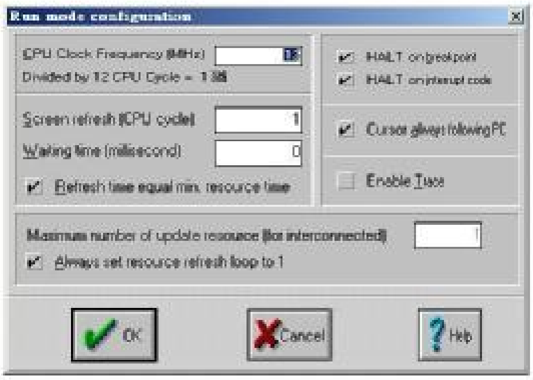
Statistics	Total	Currents
Lines :	168	168

Loading a microcontroller

1. In Configure menu, choose Configure Load CPU.
2. In file dialog box, choose **89C2051.CPL** file. Click OK to load

Run mode configuration

1. In Option menu, choose RUN Mode
2. Enter 12 at CPU Clock Frequency. This indicate clock frequency is 12 MHz
3. Enter 1 at Waiting Time
4. Check Cursor always following PC. Click OK to close the dialog box
   microcontroller is now configured



CPU Clock Frequency (MHz) 12
Divided by 12 CPU Cycle = 1 38

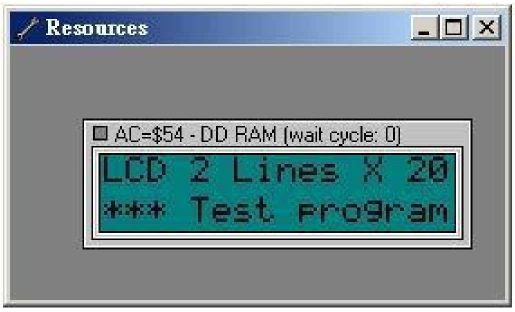
Screen refresh (CPU cycle) 1
Waiting time (millisecond) 0
☑ Refresh time equal min. resource time

☑ HALT on breakpoint
☑ HALT on interrupt code
☑ Cursor always following PC
☐ Enable Trace

Maximum number of update resource (for interconnected) 1
☑ Always set resource refresh loop to 1

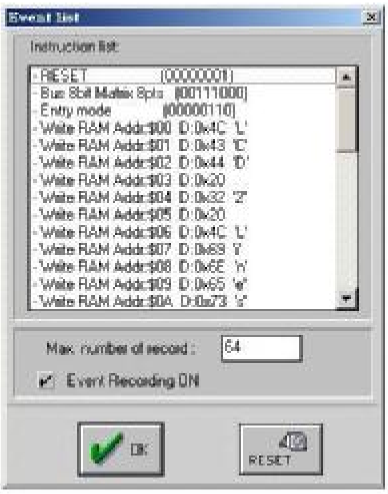
Execute Program

1. In Program menu choose Reset. CPU core is now reset
2. Click GO button to execute the program



□ AC=$54 - DD RAM (wait cycle: 0)

```
LCD 2 Lines X 20
*** Test program
```

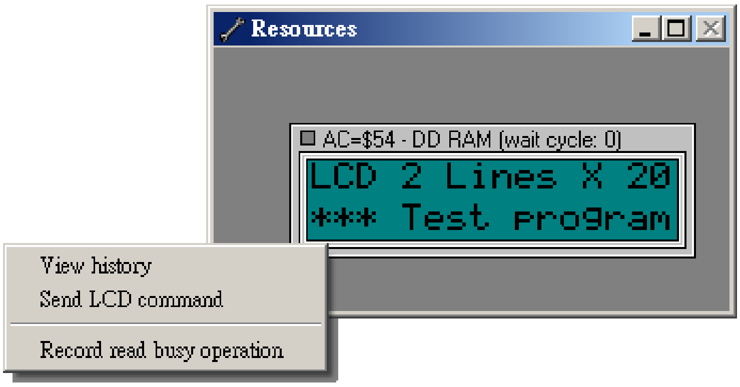


Instruction list:

```
- RESET                 (00000001)
- Bus 8bit Matrix 8pts  (00111000)
- Entry mode            (00000110)
- Write RAM Addr:$00  D:0x4C  'L'
- Write RAM Addr:$01  D:0x43  'C'
- Write RAM Addr:$02  D:0x44  'D'
- Write RAM Addr:$03  D:0x20
- Write RAM Addr:$04  D:0x32  '2'
- Write RAM Addr:$05  D:0x20
- Write RAM Addr:$06  D:0x4C  'L'
- Write RAM Addr:$07  D:0x69  'i'
- Write RAM Addr:$08  D:0x6E  'n'
- Write RAM Addr:$09  D:0x65  'e'
- Write RAM Addr:$0A  D:0x73  's'
```

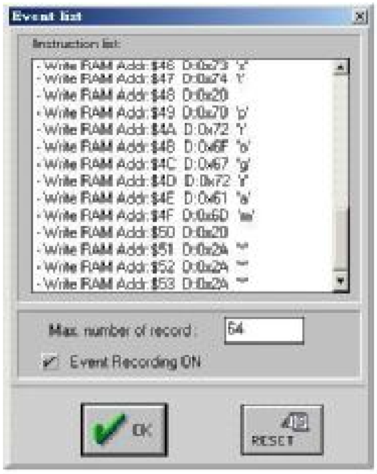
Max. number of record : 64

☑ Event Recording ON



□ AC=$54 - DD RAM (wait cycle: 0)

```
LCD 2 Lines X 20
*** Test program
```

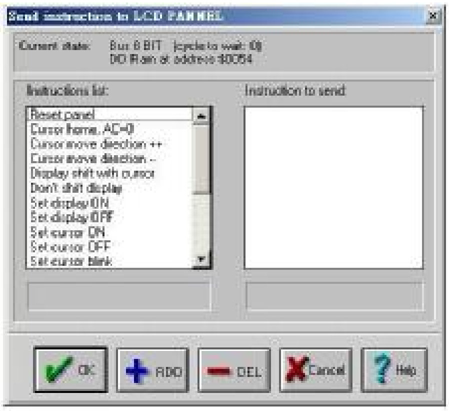


Instruction list:

```
- Write RAM Addr:$46  D:0x73  's'
- Write RAM Addr:$47  D:0x74  't'
- Write RAM Addr:$48  D:0x20
- Write RAM Addr:$49  D:0x70  'p'
- Write RAM Addr:$4A  D:0x72  'r'
- Write RAM Addr:$4B  D:0x6F  'o'
- Write RAM Addr:$4C  D:0x67  'g'
- Write RAM Addr:$4D  D:0x72  'r'
- Write RAM Addr:$4E  D:0x61  'a'
- Write RAM Addr:$4F  D:0x6D  'm'
- Write RAM Addr:$50  D:0x20
- Write RAM Addr:$51  D:0x2A  '*'
- Write RAM Addr:$52  D:0x2A  '*'
- Write RAM Addr:$53  D:0x2A  '*'
```

Max. number of record : 64

☑ Event Recording ON



Current state: Bus 8 BIT (cycle to wait: 0)
DD Ram at address $0054

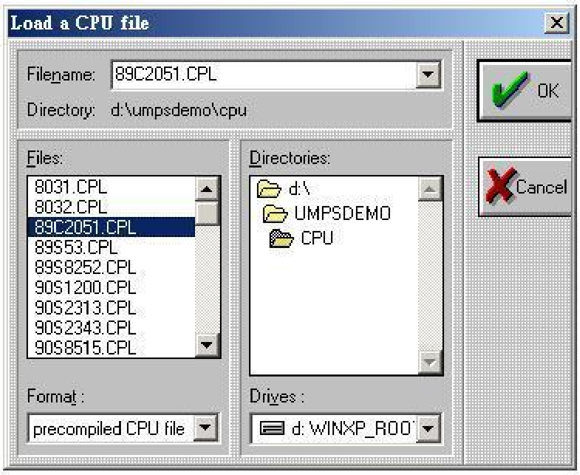
Instructions list:

```
Reset panel
Cursor home, AC=0
Cursor move direction ++
Cursor move direction --
Display shift with cursor
Don't shift display
Set display ON
Set display OFF
Set cursor ON
Set cursor OFF
Set cursor blink
```

Instruction to send

4.SPITEST.ASM

1. Loading a microcontroller
2. Compiling a program



Load a CPU file

Filename: 89C2051.CPL

Directory: d:\umpsdemo\cpu

Files:

```
8031.CPL
8032.CPL
89C2051.CPL
89S53.CPL
89S8252.CPL
90S1200.CPL
90S2313.CPL
90S2343.CPL
90S8515.CPL
```

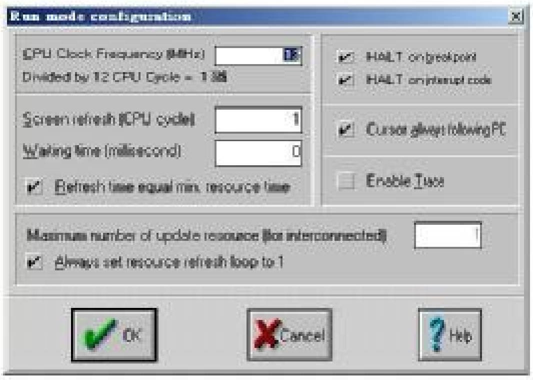
Directories:

```
d:\
  UMPSDEMO
    CPU
```

Format : precompiled CPU file

Drives : d: WINXP_ROO

3. Run mode configuration



Run mode configuration

CPU Clock Frequency (MHz) [illegible]

Divided by 12 CPU Cycle = [illegible]

Screen refresh (CPU cycle) 1

Waiting time (millisecond) 0

☑ Refresh time equal min. resource time

☑ HALT on breakpoint

☑ HALT on interrupt code

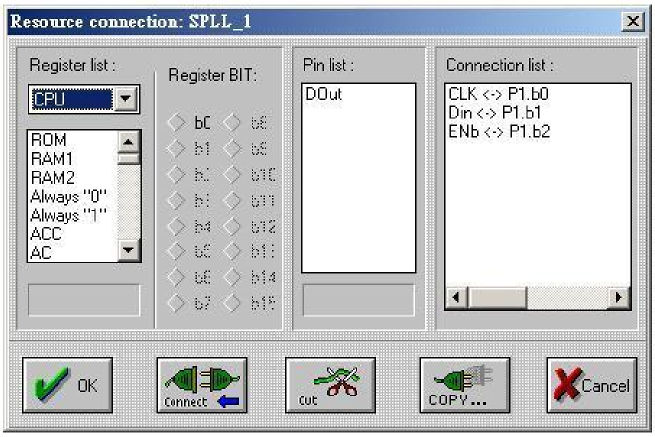
☑ Cursor always following PC

☐ Enable Trace

Maximum number of update resource (for interconnected) 1

☑ Always set resource refresh loop to 1

4. Creating resources
5. Connecting resources:



Resource connection: SPLL_1

Register list : CPU

```
ROM
RAM1
RAM2
Always "0"
Always "1"
ACC
AC
```

Register BIT:

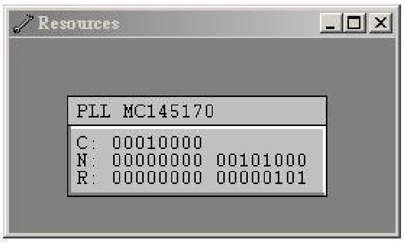
```
○ b0  ○ b8
○ b1  ○ b9
○ b2  ○ b10
○ b3  ○ b11
○ b4  ○ b12
○ b5  ○ b13
○ b6  ○ b14
○ b7  ○ b15
```

Pin list :

```
DOut
```

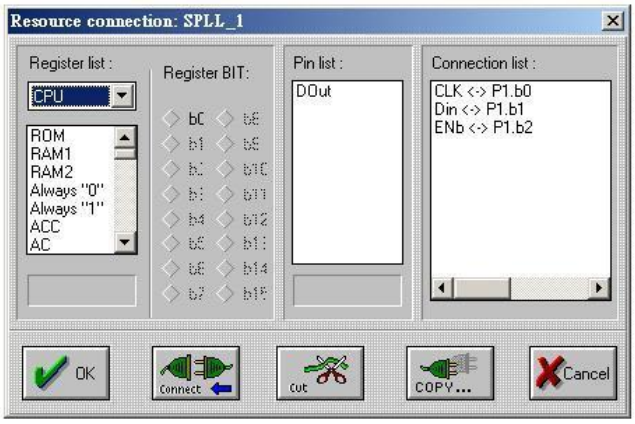
Connection list :

```
CLK <-> P1.b0
Din <-> P1.b1
ENb <-> P1.b2
```



Resources

```
PLL MC145170
C: 00010000
N: 00000000 00101000
R: 00000000 00000101
```



Resource connection: SPLL_1

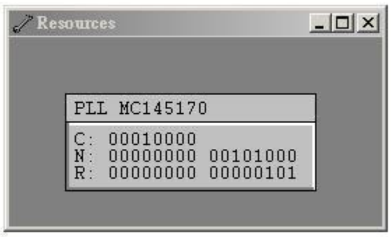
Register list : CPU

ROM
RAM1
RAM2
Always "0"
Always "1"
ACC
AC

Pin list : DOut

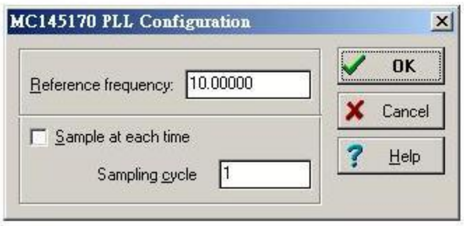
Connection list :

```
CLK <-> P1.b0
Din <-> P1.b1
ENb <-> P1.b2
```



Resources

```
PLL MC145170
C:  00010000
N:  00000000 00101000
R:  00000000 00000101
```



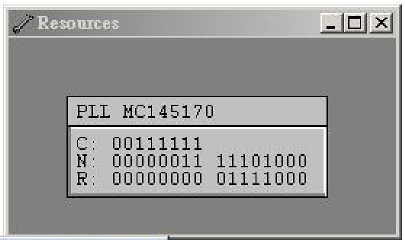
MC145170 PLL Configuration

Reference frequency: 10.00000

□ Sample at each time

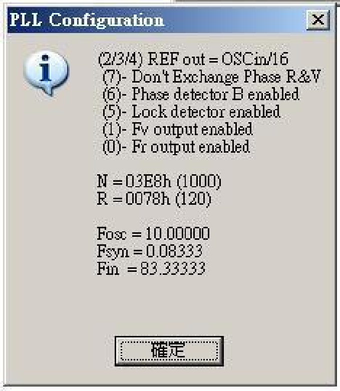
Sampling cycle 1

6. Execute Program



Resources

```
PLL MC145170
C:  00111111
N:  00000011 11101000
R:  00000000 01111000
```



PLL Configuration

```
(2/3/4) REF out = OSCin/16
 (7)- Don't Exchange Phase R&V
 (6)- Phase detector B enabled
 (5)- Lock detector enabled
 (1)- Fv output enabled
 (0)- Fr output enabled

N = 03E8h (1000)
R = 0078h (120)

Fosc = 10.00000
Fsyn = 0.08333
Fin  = 83.33333
```