

## 1 Introduction

To create a well formatted printout of your code or listing file is not always possible using some IDEs or editors (Notepad, CodeWarrior, etc.). One of the very best programs for printing out program code – or general text files – comes from the UNIX world. That is **enscript**. It takes a text file and formats it based on options you give it and converts it to postscript for printing. If desired, it also will send the output directly to the printer. Unfortunately, the printing configuration does not allow this in the EET labs so we'll have to send the output to a file and then convert the postscript file to pdf and print it via Acrobat.

Since this is a UNIX command, you'll have to use a little UNIX to use it. In the lab, this means using Cygwin – a free UNIX command distribution for windows.

## 2 Using Cygwin and Basic UNIX Navigation

Open the Cygwin window from the desktop or the Start menu. You will get a prompt like this:

```
toddm2@toleak ~  
$
```

You will now have to move to your project directory before using enscript. In UNIX, to change directories, you use **cd**. For example, to go to the folder 'z:\etec274\lab1' you would type:

```
$ cd z:\etec274\lab1
```

(Note the **\$** is the prompt, you do not type it)

Now you should be in the desired directory. To verify this, you can do two things – ask what your current working directory is, and view the contents of the directory.

To see what directory you are in type:

```
$ pwd
```

It will respond with your 'present working directory'.

To view the contents of the directory, use the **ls** command. For example:

```
$ ls  
Lab1.a12  Lab1.lst  Lab1.s19  Lab1.sym
```

That is all there is to the navigation and viewing required for using **enscript**.

# Printing Code and Listing Files Using Enscript (cont)

## 3 Generating Postscript for Source Code or Debug Listing

If you are printing assembly or C source code, or are printing compact listings for debugging use the following **enscript** command exactly:

```
$ enscript -2Grzc -o Lab1.ps Lab1.lst
```

Where, **Lab1.lst** is the file you want to print and **Lab1.ps** is the name of the generated postscript file. This command generates a compact printout as follows:

```
04/15/09 09:51:01 Lab1.lst 1
as12, an absolute assembler for Motorola MCU's, version 1.2h
;
; A simple demonstration program to see the 9512 run.
; It generates 60 1-second, active-low, pulses on PA0.
; MCU: 95120A, E=24MHz
; 01/15/08 Todd Morton
;
; Equates
;
0002 DORA equ $0002
0000 PORTA equ $0000
0001 BIT0 equ %00000001
176c TC1MS equ 5996 ;Delay count for 1ms
;
; Program
;
2000 org $2000
2000 4d 00 01 start bclr PORTA,BIT0 ;Initialize PORTA,BIT0
2003 4c 02 01 bset PORTA,BIT0
2006 18 0c 20 38 20 39 movb InitCnt,CurCnt ;Initialize pulse counter
; Main loop for output pulse generation
;
200c 4c 00 01 pulse bset PORTA,BIT0 ;Turn pulse on.
200f cc 00 fa ldd #250 ;Wait 250ms
2012 16 20 24 jsr WaitDms
2015 4d 00 01 bclr PORTA,BIT0 ;Turn pulse off.
2018 cc 02 ee ldd #750 ;Wait 750ms
201b 16 20 24 jsr WaitDms
201e 73 20 39 dec CurCnt ;Count pulses?
2021 26 e9 bne pulse ; No: Another pulse
2023 3f swi ; Yes: Return to monitor
;
; Subroutine WaitDms - A programmable delay in ms.
; Arguments: The number of mS is passed in ACCD.
; Registers: preserves all registers except CCR.
; Stack Reqs: 6 bytes stack space
; Req. Subs: Dly1ms
;
2024 3b WaitDms pshd
2025 16 20 2f msd1p jsr Dly1ms
2028 83 00 01 subd #1
202b 26 f8 bne msd1p
202d 3a puld
202e 3d rts
;
; Subroutine Dly1ms - 1ms delay loop.
; MCU: 9512C32, E=24MHz,no clock stretching, 16-bit bus
; Registers: preserves all registers except CCR.
; Stack Reqs: 2 bytes stack space
;
202f 34 Dly1ms pshx
2030 ce 17 6c ldx #TC1MS
2033 09 d1ms1p dex
2034 26 fd bne d1ms1p
2036 30 pulx
2037 3d rts
;
; Constants
;
2038 3c InitCnt dc.b 60 ;Initial pulse count
```

### 4 Generating Postscript for Final Listing to be Turned in

If you are printing final listings for turning in with write-ups the compact printout about wont work because some of the comments are truncated. So, for a final print of a listing file use the following **enscript** command exactly:

```
$ enscript -Grz -o Lab1.ps Lab1.lst
```

Where, **Lab1.lst** is the file you want to print and **Lab1.ps** is the name of the generated postscript file. This command generates a printout as follows:

```
04/15/09 09:51:01 Lab1.lst 1
as12, an absolute assembler for Motorola MCU's, version 1.2h
;*****
; A simple demonstration program to see the 9S12 run.
; It generates 60 1-second, active-low, pulses on PA0.
; MCU: 9S12Dx, E=24MHz
; 01/15/08 Todd Morton
;*****
; Equates
;*****
0002 DDRA equ $0002
0000 PORTA equ $0000
0001 BIT0 equ %00000001
176c TC1MS equ 5996 ;Delay count for 1ms
;*****
; Program
;*****
2000 org $2000
2000 4d 00 01 start bclr PORTA,BIT0 ;Initialize PORTA,BIT0
2003 4c 02 01 bset DDRA,BIT0
2006 18 0c 20 38 20 39 movb InitCnt,CurCnt ;Initialize pulse counter
;*****
; Main loop for output pulse generation
;*****
200c 4c 00 01 pulse bset PORTA,BIT0 ;Turn pulse on.
200f cc 00 fa ldd #250 ;Wait 250mS
2012 16 20 24 jsr WaitDms
2015 4d 00 01 bclr PORTA,BIT0 ;Turn pulse off.
2018 cc 02 ee ldd #750 ;Wait 750ms
201b 16 20 24 jsr WaitDms
201e 73 20 39 dec CurCnt ;Count pulses?
2021 26 e9 bne pulse ; No: Another pulse
2023 3f swi ; Yes: Return to monitor
;*****
; Subroutine WaitDms - A programmable delay in ms.
; Arguments: The number of mS is passed in ACCD.
; Registers: preserves all registers except CCR.
; Stack Reqs: 6 bytes stack space
; Req. Subs: Dly1ms
;*****
2024 3b WaitDms pshd
2025 16 20 2f msd1p jsr Dly1ms
2028 83 00 01 subd #1
202b 26 f8 bne msd1p
202d 3a puld
```

Notice that the comments were not truncated but the complete listing could not be printed on one page. So, the final print is complete but uses more paper. That is why you use the compact version for debugging.

### 5 Generating pdf and Printing

Now you will need to generate a pdf file to print. To do this use the **ps2pdf** command:

```
$ ps2pdf Lab1.ps Lab1.pdf
```

Where, **Lab1.ps** is the postscript file generated in the last step and **Lab1.pdf** is the generated pdf file. You could also use Acrobat Distiller for this but, hey, you're already doing UNIX.

Once you have the pdf file, you can open it with Acrobat and print. Remember to print two sided and, since this is landscape, set the pages to open along the top.

Do not close the Cygwin window. Now that you have it in your project directory, you can skip to step three or four immediately to make a new printout.

That's it, now you know UNIX (smile) and can make great looking printouts of your code.