

# Emily52

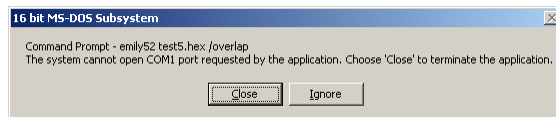
A Simulator/Emulator for the 8051 and 8052  
Dunfield Development Services Inc.  
<http://www.dunfield.com>

## Overview

- A simulator allows you to verify program correctness without operational embedded hardware. It gives you the ability to carefully examine program execution and memory/register usage.
- Emily52 is a simulator/emulator for the 8051 architecture
- Target hardware or SFR emulation code is required to accurately simulate the SFRs. We don't typically use this feature in this course.
- Steps for debugging your program:
  1. Create an assembly (or C) source file. Edit your source code, if corrections or enhancements are required.
  2. Assemble (or compile) the source to create a hex file (.hex extension)
  3. Load the hex file into Emily52 (started with the /overlap option)
  4. Use the features in Emily52 to execute your program and examine registers and memory contentsRepeat steps 1-4 as needed until your program is complete and correct
- For more detailed information, refer to the Emily52 and ASM51 documentation included with Emily52.

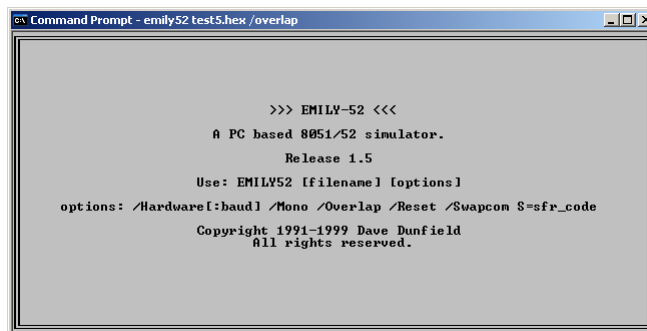
# Starting Emily52

- Emily52 must be used with the /overlap option in order for the simulator to act like the target hardware used in this course
- From a DOS command prompt, start Emily52 in one of the following ways (assuming Emily52 is in your computer's PATH environment variable):
  - emily52 file.hex /overlap
  - emily52 /overlap (and then load a hex file using the 'L' option)
- When you start Emily52, you may see a warning that the system cannot open the COM1 port. Click on the Ignore button to continue.



# Splash Screen

- When Emily52 starts, you see a splash screen.
- TIP: You can skip this screen (and save 5-10 seconds of your valuable time) by pressing any key when this screen appears.



# Main Window (Control Panel)

- The Main Window shows you the contents of program memory, registers, internal memory (IRAM), and simulator status
- To exit Emily52, press the escape key 'ESC'

```

0000 78 00  MOV  R0,#00
0002 E4  CLR  A
0003 F6  MOV  [R0],A
0004 04  INC  A
0005 08  INC  R0
0006 80 FB  SJMP 0003
0008 00  NOP
0009 00  NOP
000A 00  NOP
000B 00  NOP
000C 00  NOP
000D 00  NOP
000E 00  NOP
000F 00  NOP
0010 00  NOP
0011 00  NOP

A :00 0000: 00 00 00 00
B :00 0004: 00 00 00 00
DPTR:0000 [78] 000C: 00 00 00 00
SP :07 [00] 0010: 00 00 00 00
PC :0000 0014: 00 00 00 00
R0 :00 [00] 0018: 00 00 00 00
R1 :00 [00] 001C: 00 00 00 00
R2 :00 0020: 00 00 00 00
R3 :00 0024: 00 00 00 00
R4 :00 0028: 00 00 00 00
R5 :00 002C: 00 00 00 00
R6 :00 0030: 00 00 00 00
R7 :00 0034: 00 00 00 00
0038: 00 00 00 00
003C: 00 00 00 00

Code and data are overlapped, Hardware emulation is disabled
test5.hex: 8 bytes loaded
Press F1 for help
  
```

# Main Help Window

- Press 'F1' from the Main Window to get the Main Help Window
- Note: Pressing the space bar will single step one instruction
- Note: Pressing 'G' and a carriage return ('CR' or 'Enter') will cause the program to execute non-stop
- Press the escape key 'ESC' to close the help window

```

0000 78 00  MOV  R0,#00
0002 E4  CLR  A
0003 F6  MOV  [R0],A
0004 04  INC  A
0005 08  INC  R0
0006 80 FB  SJMP 0003
0008 00  NOP
0009 00  NOP
000A 00  NOP
000B 00  NOP
000C 00  NOP
000D 00  NOP
000E 00  NOP
000F 00  NOP
0010 00  NOP
0011 00  NOP

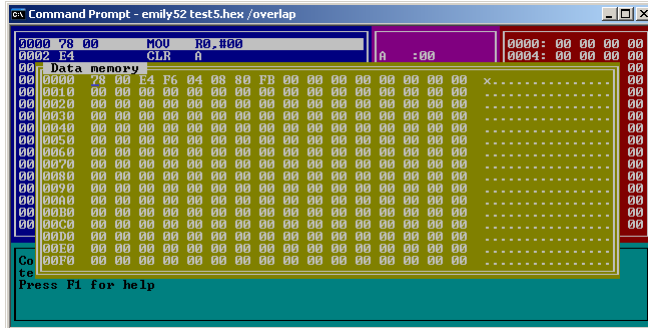
A :00 0000: 00 00 00 00
B :00 0004: 00 00 00 00
DPTR:0000 [78] 000C: 00 00 00 00
SP :07 [00] 0010: 00 00 00 00
PC :0000 0014: 00 00 00 00
R0 :00 [00] 0018: 00 00 00 00
R1 :00 [00] 001C: 00 00 00 00
R2 :00 0020: 00 00 00 00
R3 :00 0024: 00 00 00 00
R4 :00 0028: 00 00 00 00
R5 :00 002C: 00 00 00 00
R6 :00 0030: 00 00 00 00
R7 :00 0034: 00 00 00 00
0038: 00 00 00 00
003C: 00 00 00 00

Emulator commands =
A - Animate execution          F2 - Set disassembler address
B - set a Breakpoint          F3 - Setup serial port
C - Change a register         F4 - Breakpoint counters
D - edit Data memory          F5 - Reset processor
F - Function registers        F10 - Shell to DOS
G - Go: execute program       space - Step one instruction
I - edit Internal memory      Up - Backup IRAM one line
J - Jump to PC display        Down - Advance IRAM one line
K - Kill all breakpoints      PgUp - Backup IRAM one page
L - Load program file        PgDn - Advance IRAM one page
N - Next code page           Esc - Terminate EMILY52
P - edit Program memory
R - Remove a breakpoint
T - Traceback viewer
U - scroll code page Up
V - Vector to interrupt

Code a test5:
Press F1 for help
  
```

# External Data Window

- Press 'D' from the Main Window to open the External Data Window
- This window shows you the contents of external data memory (SRAM, accessed in hardware using the /RD and /WR signals)

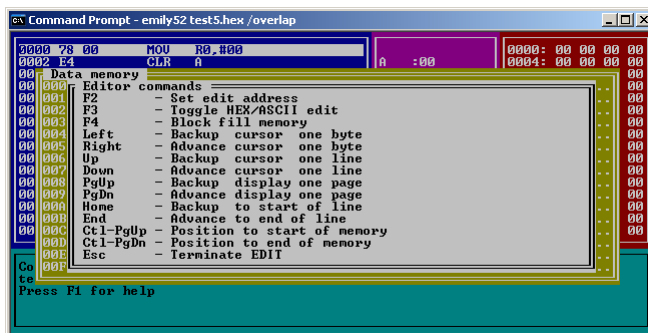


© Copyright 2004 Linden H. McClure, Ph.D.

7

# External Data Help Window

- Press 'F1' from the External Data Memory Window to get a help window specific to external data memory
- TIP: You can use 'Block fill memory' to clear a range of data memory to '00' or set the range to any other 8-bit value



© Copyright 2004 Linden H. McClure, Ph.D.

8