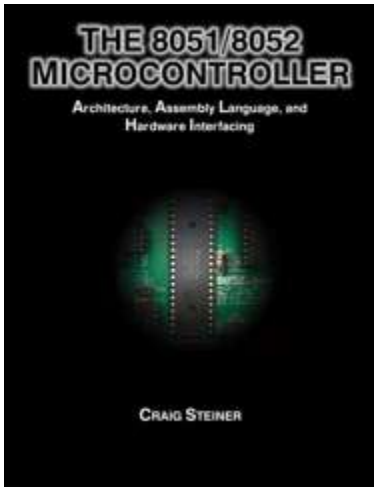
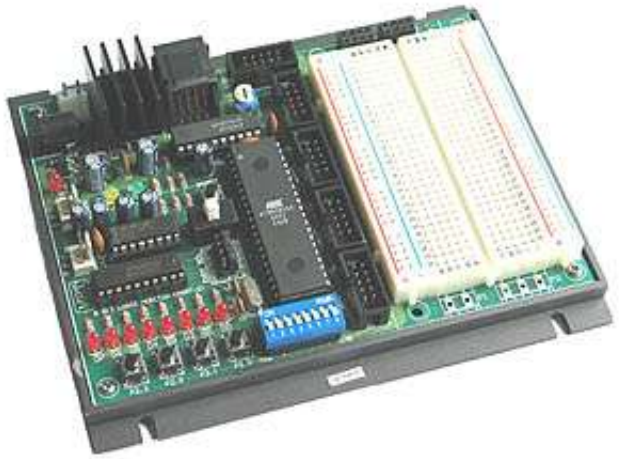
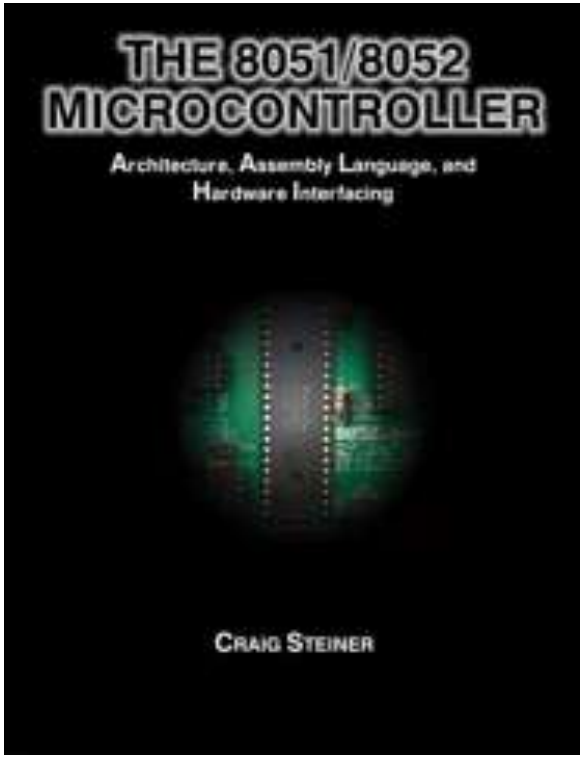
CSCI 255:
Introduction to Embedded Systems
Fall 2011





Keith Vertanen
Museum 102
496-4385
kvertanen@mtech.edu



http://www.8052.com/tut8051

Course web site (lecture slides, links to online materials):
http://katie.mtech.edu/classes/csci255

Moodle (grades):
https://moodlemtech.mrooms3.net/course/view.php?idnumber=72253

Expectations (what you should already know)

- How a computer works
  - At a high-level
- Write, compile, execute programs
  - High-level language, e.g. Java, C
- Variables, data types, assignments
- Control flow structures, e.g. if-then-else
- Repetition structures, e.g. for loop, while loop

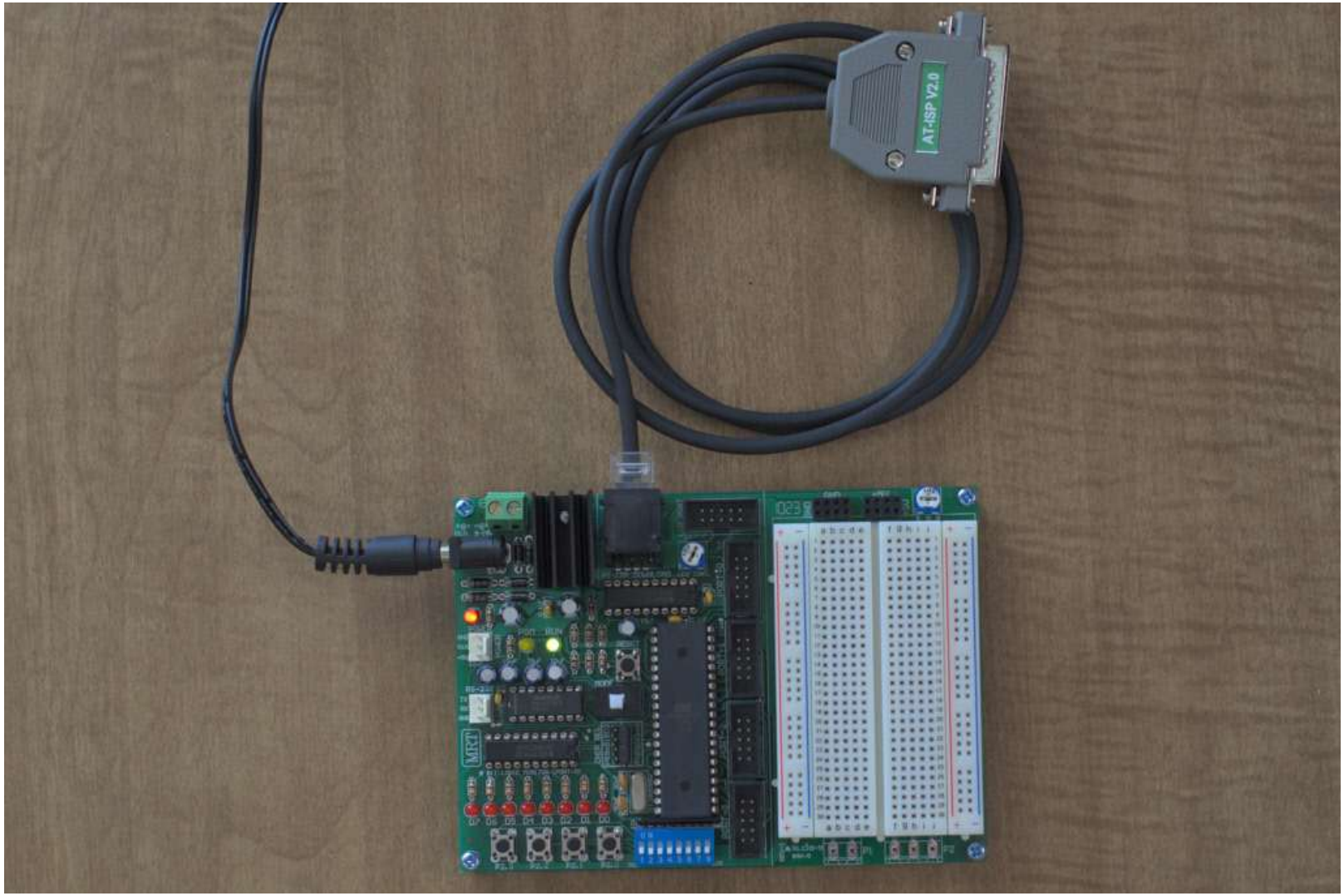
Course outcomes (what you'll learn)

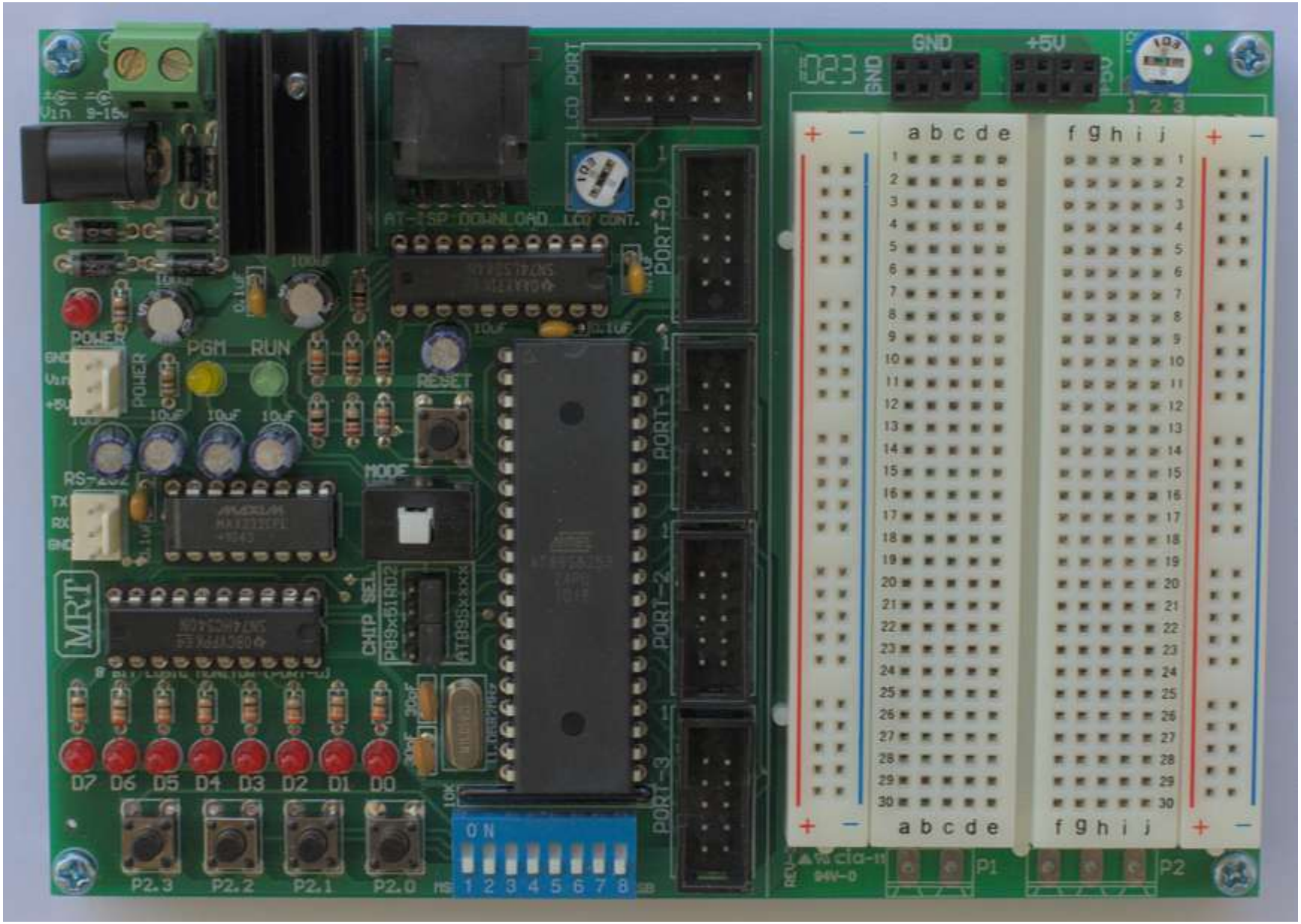
- Number systems
  - binary, octal, decimal, hexadecimal
- How computers represent numbers
  - integers (positive/negative), floating-point
- Boolean algebra, bit manipulations
- Combinational and sequential circuits
  - How these things build up to a computer
- Embedded programming
  - In assembly language and C
  - Subroutines, timers, interrupts

Lab intro day

- Goals
  - Meet your fellow students
  - Shake out software/hardware
  - Make some LEDs light up

MEB-2000P educational board




GND
+5V
LCD PORT
AT-ISP DOWNLOAD
LCD CONT.
PORT-0
PORT-1
PORT-2
PORT-3
POWER
PGM
RUN
RESET
MODE
CHIP SEL
P89x51RD2
AT89Sxxxx
MRT
D7 D6 D5 D4 D3 D2 D1 D0
P2.3 P2.2 P2.1 P2.0
ON
1 2 3 4 5 6 7 8
P1
P2
a b c d e
f g h i j

Desktop example

Source code:

Plain text file created in a high-level programming language

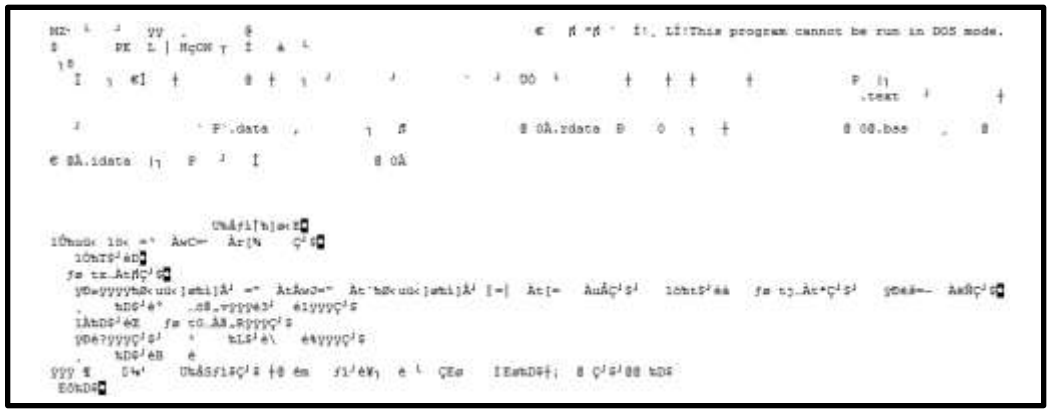
```
#include <stdio.h>

int main(int argc, char** argv)
{
   printf("Hello world!\n");
   return 0;
}
```

HelloWorld.c

```
% gcc HelloWorld.c -o HelloWorld
```

Machine language:

Actual binary run by a particular processor, not human readable/writeable



HelloWorld.exe

Desktop example

Source code:
Plain text file created in a high-level programming language

```
#include <stdio.h>

int main(int argc, char** argv)
{
   printf("Hello world!\n");
   return 0;
}
```

HelloWorld.c

```
% gcc -S HelloWorld.c -o HelloWorld.s
```

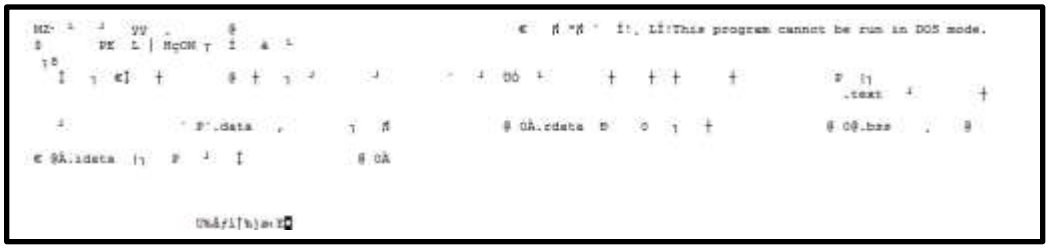
Assembly language:
Low-level, but still human readable/writeable

```
_main:
            pushl       %ebp
            movl        %esp, %ebp
            subl        $8, %esp
...
```

HelloWorld.s

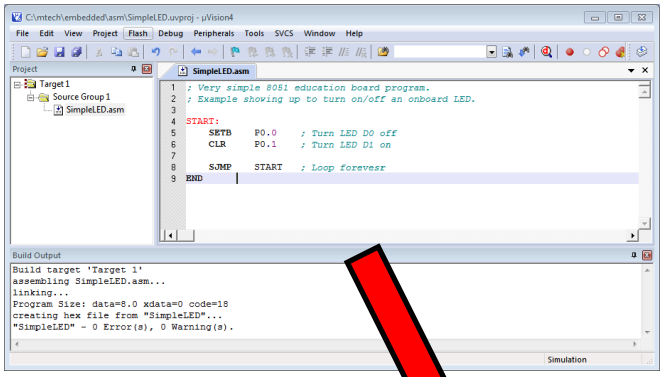
```
% gcc HelloWorld.s -o HelloWorld
```

Machine language:
Actual binary run by a particular processor, not human readable/writeable

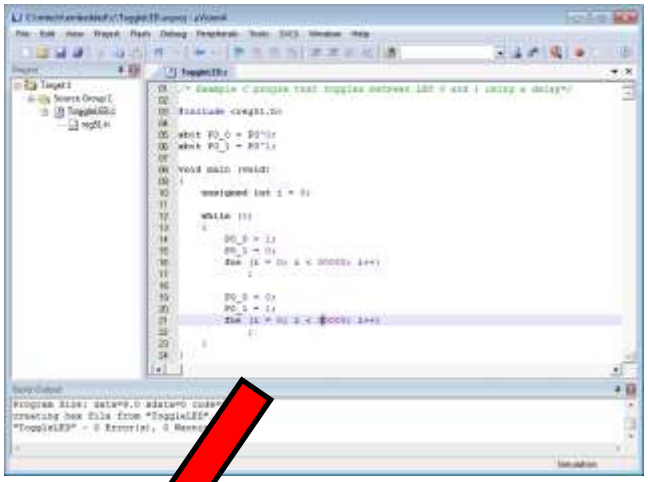


HelloWorld.exe

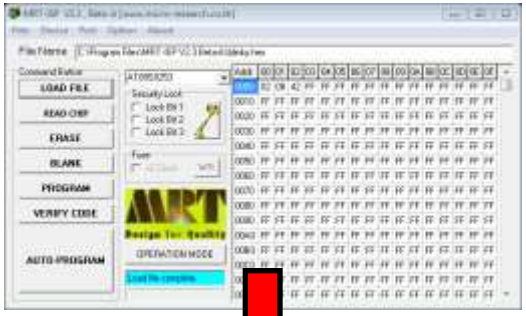
Assembly language



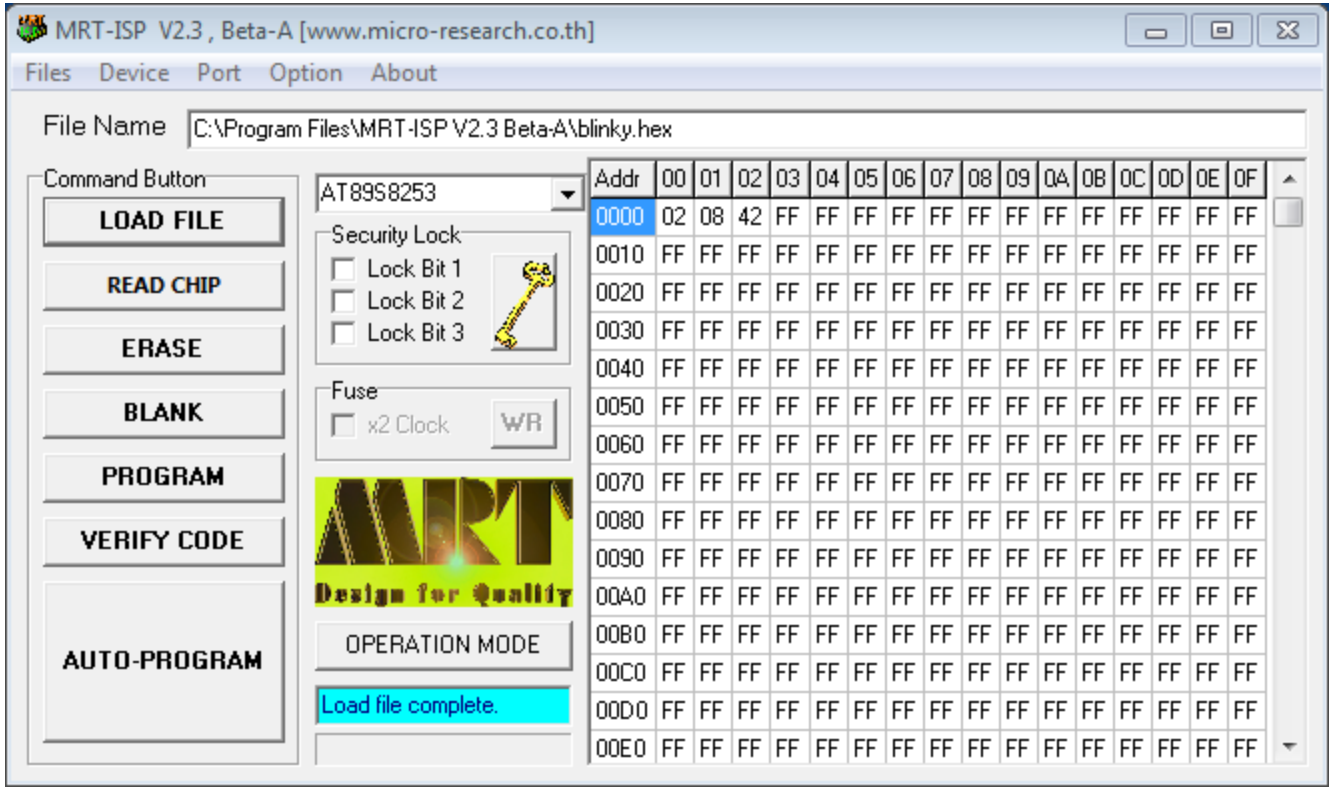
C

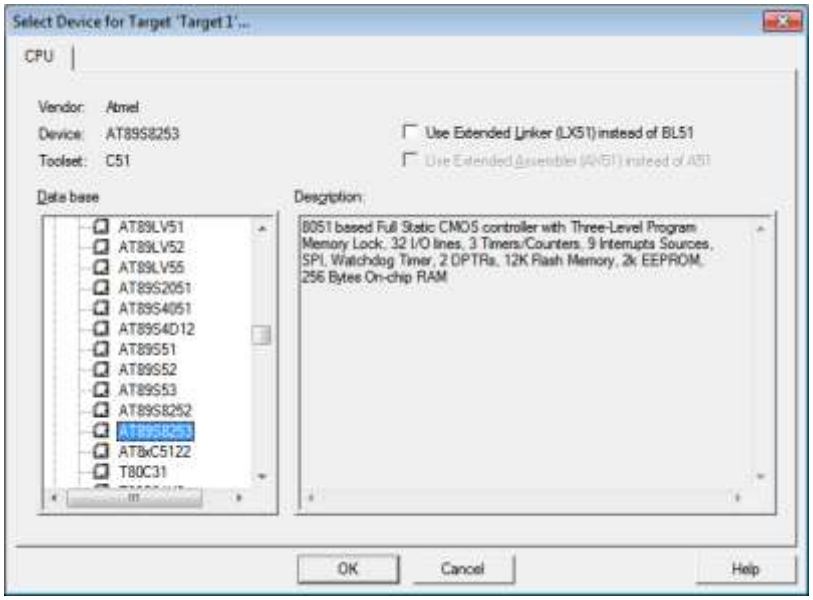


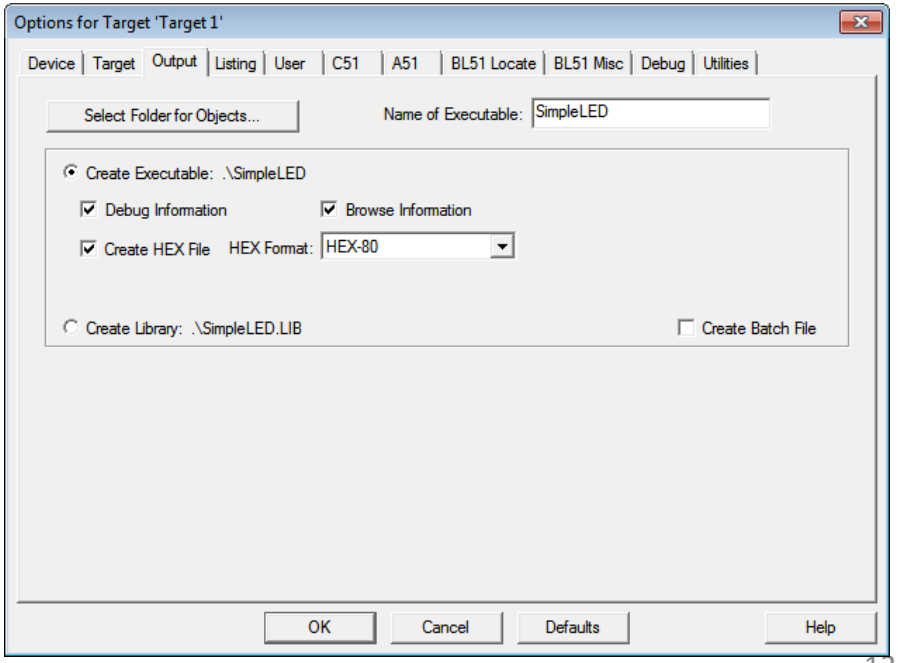
Hex file

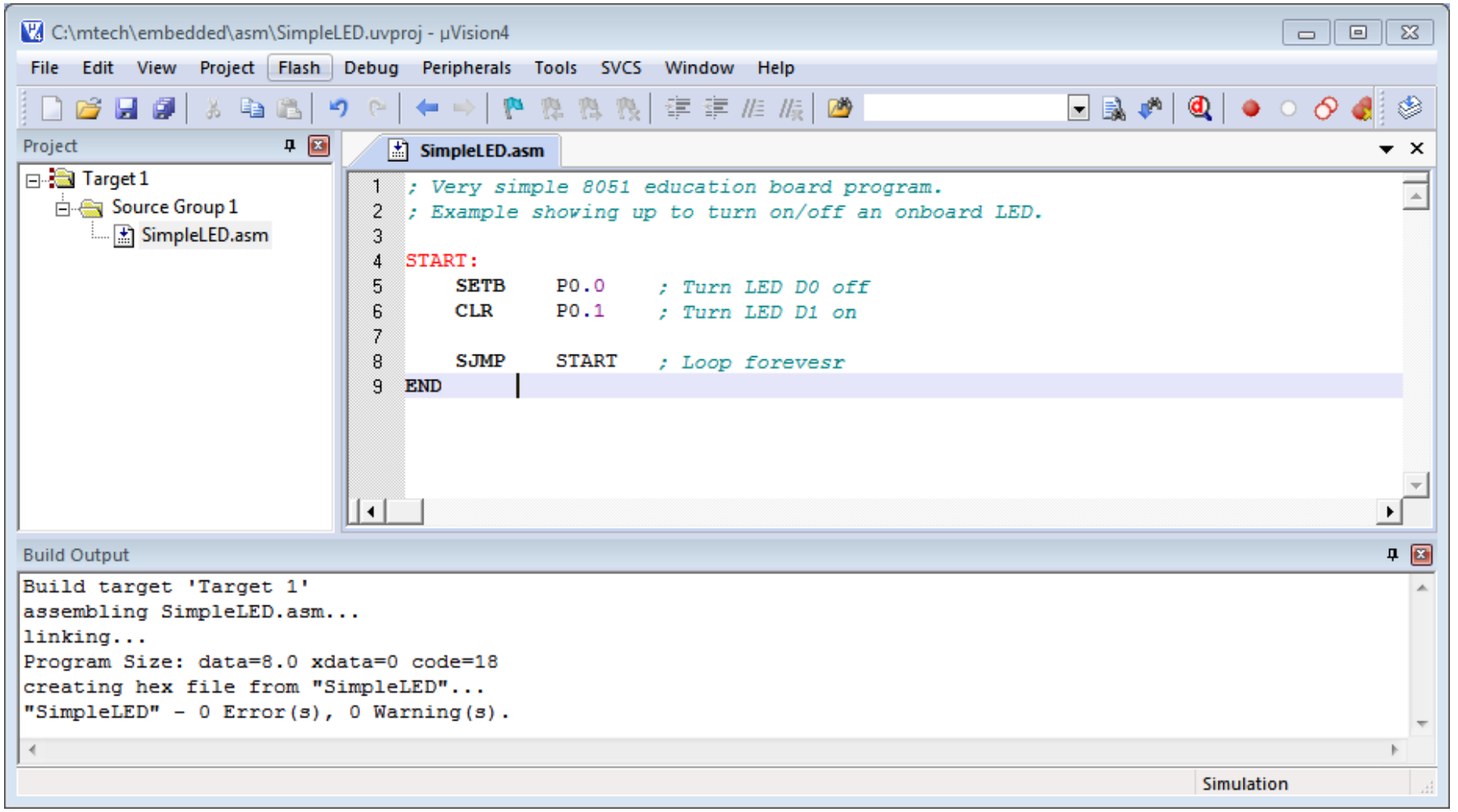


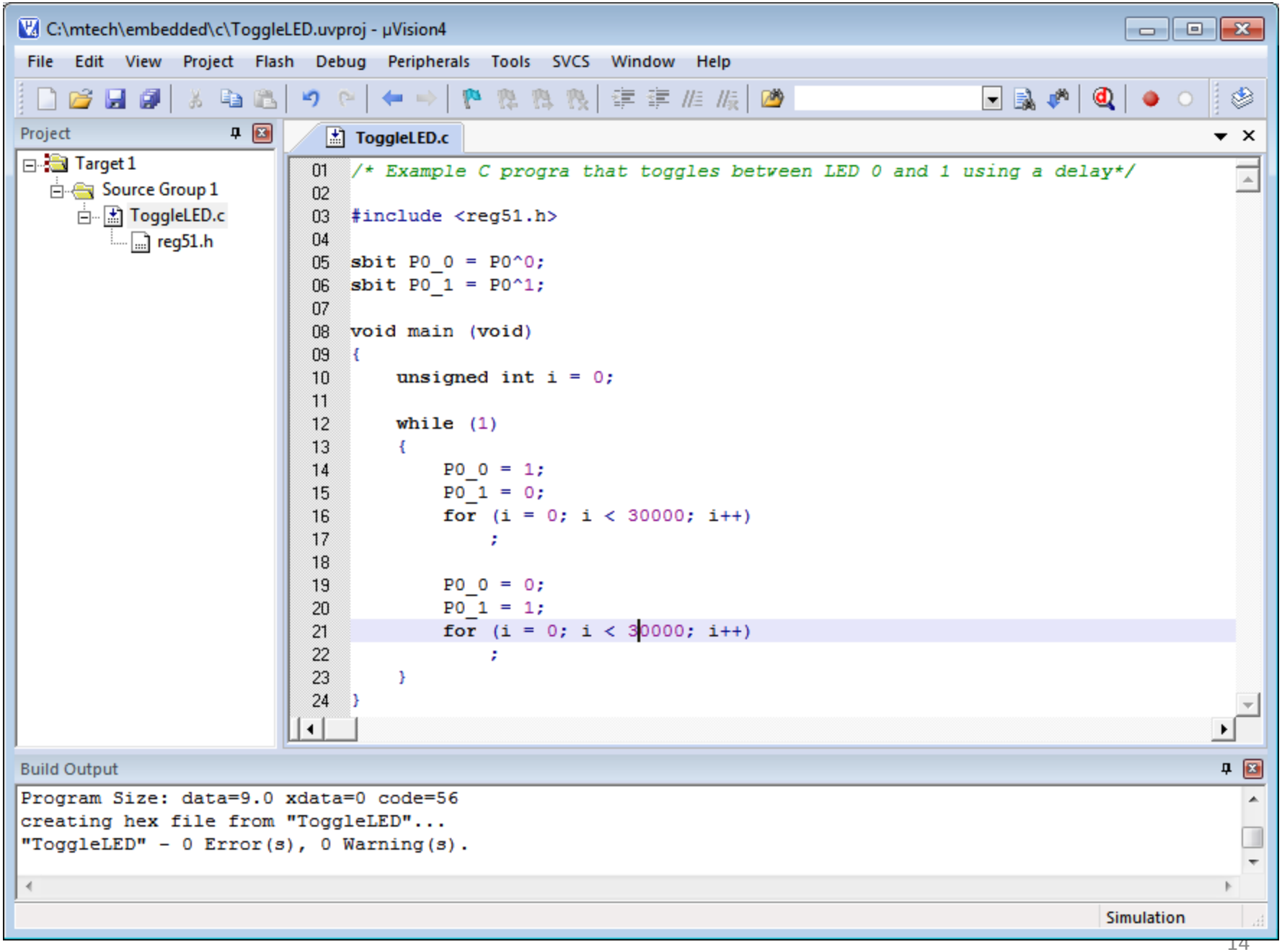
Loading machine code




Select Device for Target 'Target 1'...
CPU
Vendor: Atmel
Device: AT89S8253
Toolset: C51
Use Extended Linker (LX51) instead of BL51
Use Extended Assembler (AX51) instead of A51
Data base
Description:
AT89LV51
AT89LV52
AT89LV55
AT89S2051
AT89S4051
AT89S4D12
AT89S51
AT89S52
AT89S53
AT89S8252
AT89S8253
AT8xC5122
T80C31
8051 based Full Static CMOS controller with Three-Level Program Memory Lock, 32 I/O lines, 3 Timers/Counters, 9 Interrupts Sources, SPI, Watchdog Timer, 2 DPTRs, 12K Flash Memory, 2k EEPROM, 256 Bytes On-chip RAM
OK
Cancel
Help


Options for Target 'Target 1'
Device
Target
Output
Listing
User
C51
A51
BL51 Locate
BL51 Misc
Debug
Utilities
Select Folder for Objects...
Name of Executable: SimpleLED
Create Executable: .\SimpleLED
Debug Information
Browse Information
Create HEX File
HEX Format: HEX-80
Create Library: .\SimpleLED.LIB
Create Batch File
OK
Cancel
Defaults
Help


C:\mtech\embedded\asm\SimpleLED.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Project
Target 1
Source Group 1
SimpleLED.asm
SimpleLED.asm
1 ; Very simple 8051 education board program.
2 ; Example showing up to turn on/off an onboard LED.
3
4 START:
5 SETB P0.0 ; Turn LED D0 off
6 CLR P0.1 ; Turn LED D1 on
7
8 SJMP START ; Loop forevesr
9 END
Build Output
Build target 'Target 1'
assembling SimpleLED.asm...
linking...
Program Size: data=8.0 xdata=0 code=18
creating hex file from "SimpleLED"...
"SimpleLED" - 0 Error(s), 0 Warning(s).
Simulation


C:\mtech\embedded\c\ToggleLED.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Project
Target 1
Source Group 1
ToggleLED.c
reg51.h
ToggleLED.c
01 /* Example C progra that toggles between LED 0 and 1 using a delay*/
02
03 #include <reg51.h>
04
05 sbit P0_0 = P0^0;
06 sbit P0_1 = P0^1;
07
08 void main (void)
09 {
10 unsigned int i = 0;
11
12 while (1)
13 {
14 P0_0 = 1;
15 P0_1 = 0;
16 for (i = 0; i < 30000; i++)
17 ;
18
19 P0_0 = 0;
20 P0_1 = 1;
21 for (i = 0; i < 30000; i++)
22 ;
23 }
24 }
Build Output
Program Size: data=9.0 xdata=0 code=56
creating hex file from "ToggleLED"...
"ToggleLED" - 0 Error(s), 0 Warning(s).
Simulation