EE281

Embedded System
Design Laboratory

Lecture #5 Outline

- Announcements
- AVR-GCC (C compiler)
- Review example code

Announcements

- Have you finished Lab #2?
  - Was due: **Monday Oct 14th, 5pm**
- Lab#3 assigned
  - Due date: **Monday Oct 21, 5pm**
- Midterm survey

AVR-GCC

- AVR-GCC is a port of the GNU C compiler (GCC)
- Includes C library support for all AVR hardware features
- Low-Cost (FREE!!!)
- High-Performance (excellent optimizing compiler)
- Runs on Unix and Windows systems
  - EE281 will formally support AVR-GCC for WIN32
  - Also exists for UNIX →Linux, Sun, HP, etc
- Does not include a C code editor
  - Pick an editor you like:
    - Microsoft Visual Studio
    - AVR Studio
    - Emacs
- Compiling is automated using Makefiles

Installing AVR-GCC

- Download the latest AVR-GCC package from course web page "Materials" section or from www.avrfreaks.net/AVRGCC/
- Follow the installation instructions on the AVRfreaks site and/or refer to the course web page for installation help
- *AVR-GCC must be installed in a path without spaces*
- AVR-GCC requires an addition to your *path* and the AVR environment variable to be set. See the course web page for help making these changes.
- Check out the examples in GCCTEST
- Try compiling gcctest1 and others to test your installation

GCC TEST example code

- gcctest1 (flashing LED using hard-coded loop)
- gcctest2 (flashing LED using timer interrupt)
- gcctest3 (external interrupt example using INT0/1)
- gcctest4 (interrupt-driven UART example)
- gcctest5 (combined UART and EEPROM access)
- gcctest6 (floating-point math example)
- gcctest7 (printf-like access to the UART – example of multiple *.c and *.h file use)
- gcctest8 (accessing data in program memory **(replaces LPM)**)
- gcctest9 (complex timer/math/sprintf/UART example)

AVR-GCC Lib-C Functions

- Dozens of functions available for accessing memory, I/O, and peripherals more easily
- Compiler directives for making interrupt service routines, placing permanent data in FLASH, etc
- See the GCCTEST code for examples of the most common functions, directives, and usage
- See the library (LIBC) reference for a semi-complete listing of special C functions (available on the course website pdf format)

AVR-GCC Makefile Example

Embedded System
Design Laboratory

```
# Simple Makefile
# Volker Oth (c) 1999
include $(AVR)/include/make1
############ change this lines according to your project ##################
#put the name of the target mcu here (at90s8515, at90s8535, attiny22, atmega603 etc.)
    MCU = at90s8515
#put the name of the target file here (without extension)
    TRG    = gcctest7
#put your C sourcefiles here
    SRC    = uart.c $(TRG).c
#put additional assembler source file here
    ASRC    =
#additional libraries and object files to link
    LIB =
#additional includes to compile
    INC    =
#compiler flags
    CPFLAGS          = -g -O3 -Wall -Wstrict-prototypes -Wa,-ahlms=$(<:.c=.lst)
#linker flags
    LDFLAGS = -Wl,-Map=$(TRG).map,--cref
############ you should not need to change the following line #############
include $(AVR)/include/make2
###### dependecies, add any dependencies you need here ###################
uart.o     : uart.c  uart.h
$(TRG).o   : $(TRG).c
```