1 Introduction

In this lab you will familiarize yourself with the Atmel AT91SAM7L microcontroller, the IAR Embedded Workbench software, and the lab practices expected of this course.

1.1 Atmel AT91SAM7L

The Atmel AT91SAM7L128 is a low-power system-on-chip (SoC) from Atmel's smart ARM microcontroller (SAM) family which has been tailored for battery operated applications such as calculators, toys, remote controls, medical devices, mobile phone accessories and wireless sensors. It is based on the 32-bit ARM7TDMI RISC processor, which has had major success in low-power devices such as Apple's iPod, Nintendo DS, Nokia cell phones, Sirius Satellite Radio receivers, and iRobot's Roomba.

The AT91SAM7L runs at clock speeds up to 36MHz, contains 128 KB of high-speed flash memory, and 6 KB of SRAM. Additional peripherals include an LCD controller, USART, SPI, Timers, Analog to Digital, and PWM. Study the block diagram below to see how the ARM communicates with the many peripherals on the SoC.

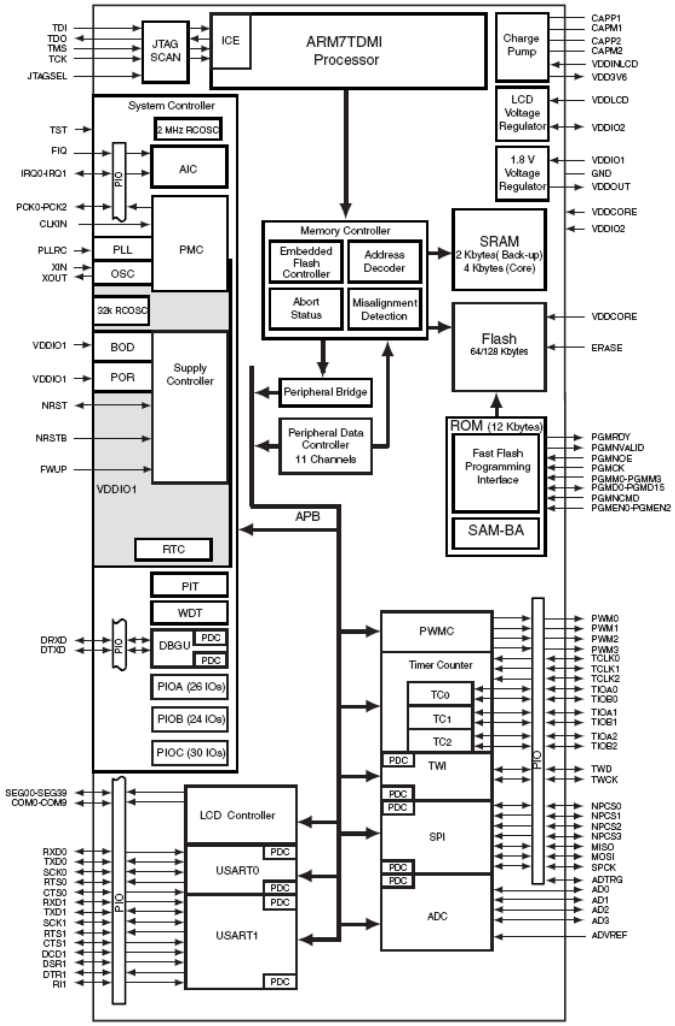

Figure 1: AT91SAM7L Block Diagram.

In this lab, we will be using the AT91SAM7L-STK (starter kit), which integrates the AT91SAM7L128 on a PCB with a 400 segment monochrome LCD, push buttons, 40-pin I/O header, AAA battery socket, JTAG/ICE debug interface, and an RS232 serial port.

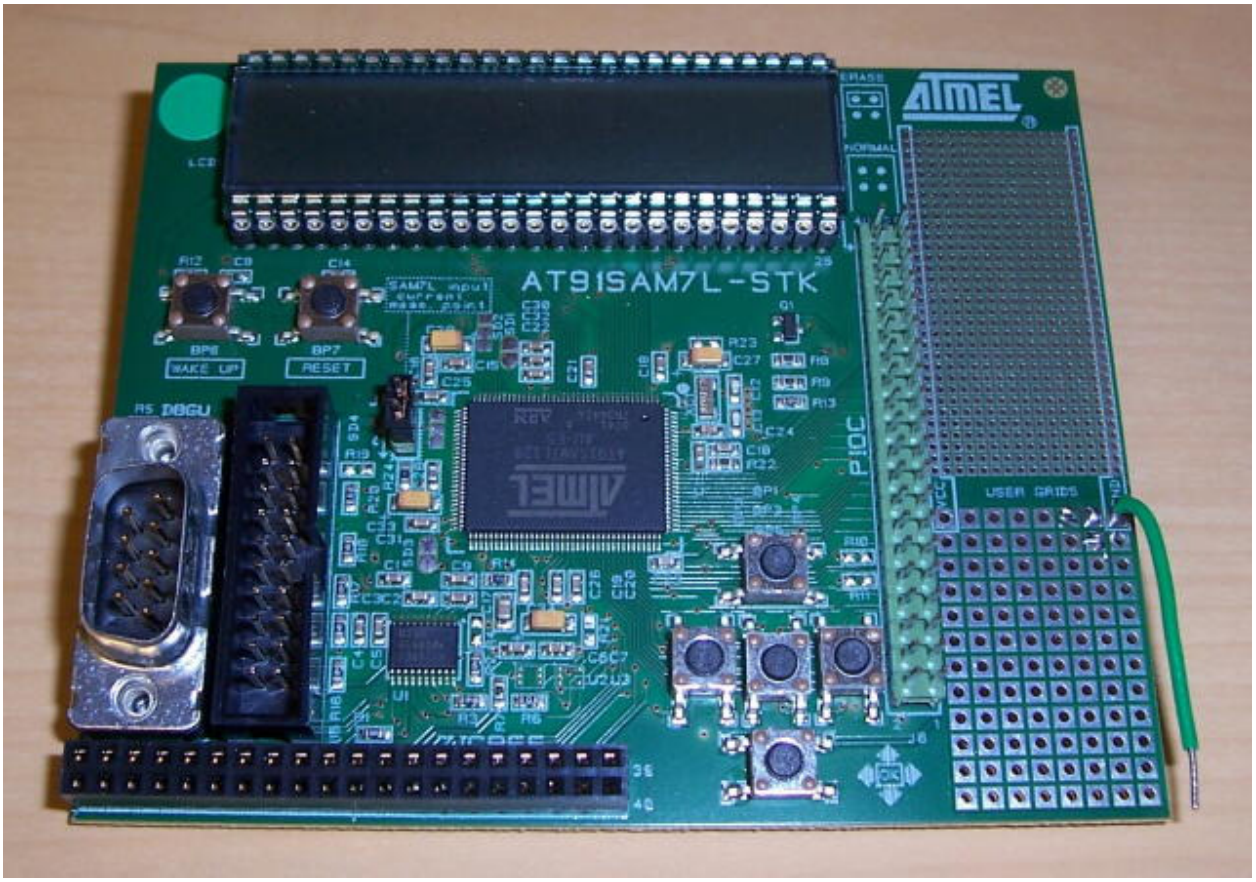

Figure 2: Photograph of the AT91SAM7L-STK.

References:

During the semester, you will frequently need to refer to the following documents for specific information on the AT91SAM7L128 microcontroller and the AT91SAM7L-STK starter kit.

AT91SAM7L128 Microcontroller

http://www.atmel.com/dyn/products/product_card.asp?part_id=4293

- ***AT91SAM7L128/64 Preliminary***
  This is the AT91SAM7L128 datasheet, which describes in detail the capabilities and use of the AT91SAM7L128 and its features.
- ***AT91SAM7L128/64 Preliminary Summary***
  This is the datasheet summary, which is an abbreviated version of the datasheet. It has less detail, but is easier to browse.
- ***ARM7TDMI Technical Reference Manual***
  This is the reference manual for the ARM7 architecture, on which the AT91SAM7L128 is based. It includes the Instruction Set Architecture (Assembly) description.

AT91SAM7L-STK Starter Kit
http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4336

- ***AT91SAM7L-STK Rev. A Starter Kit***
  This is user guide for the AT91SAM7L-STK starter kit board. It includes a schematic of the board and a description of the on-board peripherals.

1.2 IAR Embedded Workbench

IAR Embedded Workbench for ARM (EWARM) is an integrated development environment (IDE) for building and debugging embedded applications. The IDE is the framework where all necessary tools are seamlessly integrated:

- The highly optimizing IAR C/C++ Compiler
- The IAR Assembler
- The versatile IAR ILINK Linker, including accompanying tools
- A powerful editor
- A project manager
- A command line build utility
- IAR C-SPY® Debugger, a state-of-the-art high-level language debugger.

We will be using this development environment to manage, write, compile, download, and debug all of the labs that use the AT91SAM7L. IAR has some useful features that can speed up the time that it takes to develop a solution. We will be exploring those features in the first few labs.

The IAR EWARM software is installed on the ECE computers in the labs, but you may also install a free version on your own computer by downloading the 32KB KickStart edition from the following website:

IAR Embedded Workbench for ARM
http://www.iar.com/website1/1.0.1.0/68/1/index.php

- ***32KB KickStart edition***
  Here you can download a free version of the IAR Embedded Workbench software. A copy is already installed on the lab computers.
- ***Datasheet (PDF)***
  This is the datasheet for the IAR Embedded Workbench software. It is a pamphlet that gives a high-level summary of the IAR Embedded Workbench software and what it does.

We will also refer to the following documentation:

IAR Systems user and reference guides
http://www.iar.se/website1/1.0.1.0/78/1/index.php

- ***IAR Embedded Workbench IDE User Guide for ARM***
  This is the User Guide for the IAR Embedded Workbench software. It describes how to use the software.
- ***IAR C/C++ Development Guide for ARM***
  This describes in detail the process of how the IAR Embedded Workbench software compiles and links ARM software. You shouldn't need this for most of the ECE3884 labs, but its useful if you want to get the most out of the ARM microcontroller.

1.3 Weekly Lab Practices

Each week, you will have a lab description similar to this one. Read it completely before the start of each lab, and perform any pre-lab assignments. Then, print out a copy to bring with you to the lab period.

In each lab, you will be required to demonstrate results to your TA. The lab description indicates what results you must demonstrate. Your TA will observe your demonstration of correct results, then initial in a blank in the lab description. Keep the lab description printout, and submit it at the start of the next lab period.

Also, after the lab period, you must complete a lab report, and submit it at the start of the next lab. At the end of each lab description, there will be a "Lab Report" section, which describes what should be in the lab report for that lab. Some labs may take more than one lab period to complete, in which case you will turn in a single lab report at the completion of the lab.

Only a basic level of C programming skill is required for this course. However, if you feel that you need additional C/C++ reference, consult the following tutorials:

C Tutorial
http://www.cprogramming.com/tutorial.html#ctutorial

C++ Language Tutorial
http://www.cplusplus.com/doc/tutorial/

2 Lab Procedure

2.1 Open IAR

1. Click on start → ***Applications*** → ***IAR Systems*** → ***IAR Embedded Workbench for ARM Kickstart*** → ***IAR Embedded Workbench***
2. Select ***Open existing workspace*** on the Startup Screen.
3. Browse to the file location where you extracted the lab01.zip project files and open ***ECE3884 Labs\at91sam7l-stk\lab01\lab01.eww***

2.2 'Make' the Project

1. Click on Project → Make in the menu bar (or click the  icon from the Main toolbar).
2. Verify that there were no errors in the Messages window.

2.3 Connect the AT91SAM7L-STK

1. Remove the AT91SAM7L-STK from its anti-static packaging.
2. Ensure two AAA batteries are properly installed in the battery compartment.
3. Connect the RS232 serial cable between the PC and the microcontroller.
4. Connect the yellow USB IAR j-link device to the USB port of the PC.
5. Connect the JTAG ribbon cable to the microcontroller.
6. Press the 'Wake Up' button on the microcontroller.

2.4 Hyper Terminal

1. Open Hyper Terminal by clicking on 

→ Accessories → Communications → Hyper Terminal with your mouse.
2. Enter a name for this connection (Be creative!) and click on OK.
3. Select COM1 in the "Connect using" field and click on OK.
4. Choose the following Port settings and click on OK.

Bits per second	115200
Data bits	8
Parity	None
Stop bits	1
Flow control	None

5. Hyper Terminal is now configured to communicate with the Serial port of the AT91SAM7L-STK.

2.5 Download and Debug

1. Back in the IAR EWARM window, Click on Project → Download and Debug (or the  icon). This opens the IDE's debugging environment.
2. To run the program, click on Debug → Go (or the  icon).
3. The program should scroll some characters across the LCD screen and write to the Hyper Terminal window.
4. Stop the program by clicking on Debug → Break (or the  icon)
5. Leave the debugging environment by clicking on Debug → Stop Debugging (or the  icon)

2.6 Understand the Program

1. Read through the project files by expanding each folder in the Workspace window and double clicking on each file.
2. Right clicking on a function or variable in the editor allows you to navigate quickly to the definition of that function - ***go to definition of <function>***
3. Additionally, the  icon allows you to return quickly to your previous view of the source code (before clicking go to definition).
4. Using these tips, explore the program and be prepared to answer questions in **Section 3**. A complete understanding of the technical details is not expected.

2.7 Modify the Program

1. Change the string that gets printed to the LCD and Hyper Terminal window to your first and last name.
2. Make the message on the LCD screen scroll back and forth at twice the speed.

2.8 Control the Program

1. Read the section titled **Using breakpoints** on page 135 (Part 4 Debugging → Using breakpoints) of the ***IAR Embedded Workbench IDE User Guide for ARM*** .

2. Set a breakpoint such that the program breaks (stops executing) when the text on the LCD has scrolled all the way to the right and is just about to start scrolling back to the left.
3. Demonstrate the execution of the program with this breakpoint to the TA, and have the TA initial in the box.

TA Initial Lab01 Box 1

2.9 Other Examples (bonus)

1. IAR EWARM ships with more examples that can help you understand the AT91SAM7L and the pre-written functions supplied by IAR to interface to various peripherals.
2. Re-open the Startup Screen by clicking on Help → Startup Screen…
3. Click on Example applications
4. Select Atmel → at91sam7l-stk and choose among the nine available examples.
5. This lab was a customized version of basic-slcd-stk-project.
6. Explore the examples and practice what you have learned. Having working examples can speed up the time it takes to finish a project drastically.
7. When finished, press the RESET button on the microcontroller to enter standby mode.
8. Exit IAR and Hyperterminal.

3 Lab Report

Your lab report for Lab 1 should include answers to the following questions:

1. Several software routines/functions are called to initialize and operate the LCD controller. Use the techniques learned in **Section 2.6** to find which files (.c and .h) that contain these routines. List those file names.
2. What peripherals of the AT91SAM7L are used to output to the LCD screen? (Hint: Check the Datasheet and study the interrupt service routine)