# Lab 0: Introduction & Programming, Simulating, and Emulating with the Atmel XMEGA

## OBJECTIVES

In this lab you will meet your TA and give her/him the required information listed below. Your TA will present a short demonstration on soldering during this lab; these skills will aid you in building your **UF 3744 *μPAD 1.4* (ATXMega128A1U) Development Board** kit (including four other PCBs) After observing the TA demonstrate proper soldering techniques, you will be encouraged (i.e., required) to practice soldering on some practice boards.

Meet your TA. Get to know her/him. Understand the lab policies (as stated on the *Lab Rules and Policies* document).

In this lab you will also write your first program for the Atmel processor. You will gain practice in programming the processor, understanding how to simulate your program **before** programming the board, and how to debug/emulate your program **after** programming the board.

## REQUIRED MATERIALS

- As stated in the *Lab Rules and Policies*, do the following:
  - Submit your entire pre-lab report and **YOUR** asm file(s) through Canvas **BEFORE** entering the lab.
  - Print parts 14 a through d of the pre-lab report for submission to your TA as you enter lab.
- Read/save the following documents:
  - *Out of the Box uPAD Assembly Guide*
  - *uPAD 1.4 Parts List*
  - *Atmel Studio Installation Instructions*
  - *Electronic Assembly handout*
  - *Class syllabus*
  - *Create Simulate Emulate on Atmel* tutorial
  - *AVR instruction set (doc 0856)*
  - *Assembly Language Conversion: GCPU to Atmel Assembly*
- Toolbox, DAD/NAD, and multimeter (from 3701)
- Soldering iron (available for use **in lab**)
- EEL 3744 Board kit and tools (distrusted in lab)

## DISCUSSION

The *Electronic Assembly handout* discusses how to solder, a technique that makes a good electrical (and mechanical) connection between two locations.

If you have a soldering iron and solder from *Intro to ECE*, do **not** use it without explicit instruction from your TA. (The soldering irons from *Intro to ECE* have unregulated heating and may get too hot for our boards. Also, the solder that you get in this class is different than what we use. Mixing solder may lead to unreliable connections; poor soldering irons and soldering technique can burn PCB traces.)

Prior to this semester in 3744, we wire wrapped, another technique that makes a good electrical and mechanical connection between two locations by wrapping wires around pins. We will no longer wire wrap in 3744.

## PRELAB REQUIREMENTS

Read **all** of the documents listed in the sections. Answer the pre-lab questions and complete your first pre-lab report that you will submit to our Canvas account (as specified in the *Lab Rules and Policies* document). Note that your first (Lab 0) pre-lab report will be very short and will only include a few of the items that will be including in all future lab reports. You must print out sections a) through d) for this lab and all of the other labs (even if there is not much to say in a section). Answers to this lab's prelab questions can be found in the posted lab documents.

You **must** make a **flowchart** or write pseudo-code **before** writing **any** program in this course. This will help you formulate a plan of attack for the code. The flowchart or pseudo-code for part ~~B and~~ C should be included (as stated in the *Lab Rules and Policies*, part 14 g) in your pre-lab report.

Note: Pre-lab requirements **MUST** be accomplished **PRIOR** to coming to your lab.

### PART A. ATMEL Installation

Go through the *Atmel Installation Instructions* to install the software on your laptop (or tablet) computer. Obtain a screen shot of-Atmel Studio running on your computer that **also shows your name** in big letters on the same screen. To do a screen shot in Windows, press Ctrl-PrtScrn (i.e., select Ctrl and PrtScrn at the same time). Copy this screen shot into MS Word (or a similar program) and include this in Appendix (see section 14 i) of your pre-lab report that you will submit to Canvas.

### PART B. ATMEL Tutorial

Go through the *Create Simulate Emulate on Atmel* tutorial found on the website. Obtain a screen shot on your laptop of the results of step 11 that **shows your name** in big letters on the same screen. Save this screen shot in Appendix of your pre-lab report. Read the rest of the tutorial, since you will perform the emulation part of the tutorial during your lab.

### PART C. Write, Debug/Simulate a Project

Write a program (use filename `lab0.asm`) **using Atmel assembly language,** to filter data from an array in memory. For each byte in the data table, if the byte matches the filter condition, write the filtered data to the specified memory locations. Include this program in your pre-lab report (as stated in the *Lab Rules and Policies*, part 14 h). Your program will assume that the data is already placed in **program memory** prior to execution, i.e., you will use assembler directives like ".DB" to put the constant values into memory. See the following

webpage for more information on using assembler directives:

http://www.avr-tutorials.com/assembly/avr-assembler-directives

Section 4.5 of the below:
http://mil.ufl.edu/3744/docs/XMEGA/doc1022_Assembler_Directives.pdf

The array will be in hexadecimal and will contain the data in Table 1 starting at **word address 0xF000** in program memory (using assembler directives). ASCII is a 7-bit coded version of numbers, letters and symbols. An ASCII table can be found at www.asciitable.com.

Table 1: Memory Array

| | Data (ASCII) | Data | |
|---|---|---|---|
| | = | 0x3D | |
| | ~ | 0x7E | |
| | **T** | **84** | |
| | f | 102 | |
| | { | 0x7B | |
| | z | 0172 | |
| | space | 0x20 | |
| | d | 0x64 | |
| | ~ | 0x7E | |
| | ? | 0x3F | |
| | 0 | 060 | |
| | 3 | 0x33 | |
| | { | 0x7B | |
| | y | 121 | |
| | v | 118 | |
| | ! | 0x21 | |
| | x | 0x78 | |
| | w | 0x77 | |
| | NUL | 0 | |

Your program should filter the data in the array shown in Table 1 such that values above 0165 (= $165_8$ = 0x75 = 117) should be ignored, i.e., move on to the next value. Otherwise, if the value is greater than 37 (= 0x25 = 045), add 0x11; if the value is less than or equal to 37 do not add anything. Store each of the resulting values in successive **data memory** locations, starting at address **0x3744**. You must use a pointer to accomplish this task.

Your program should end after it detects the "NUL" character (0x00). The data in the original array should **not** be corrupted. The new table should end by adding the NUL character.

When you write your program, be sure to make your code modular and re-locatable, i.e., make it easy to change the location of both your input and output tables; use assembler directives appropriately. **Use the `.DSEG` assembler directive for the output table.** Remember to allocate enough memory for the output table. Try to make it easy to change the end of table value (NUL) and the filter values (0165 and 37), i.e., use assembler directives for these two values.

Test your program using the Atmel Studio Simulator. Verify that the program works. Take a screen shot of a memory view window that shows the filtered values located at the correct addresses in data memory. Also, include any registers you used in the watch window (this allows you to see the registers change when stepping through the code). Make sure the screenshot **displays your name** in big letters on the same screen. Save this screen shot in the Appendix of your pre-lab report (as stated in the *Lab Rules and Policies*, part 14 i).

## PRELAB PROCEDURE

1. It is required that you make a flowchart or write pseudo-code **before** writing **any** program in this course. This will help you formulate a plan of attack for the code.
2. Create your program for parts B and C (in a file on your computer) using the Atmel assembly language instruction set. Include this in your pre-lab report (as stated in the *Lab Rules and Policies*, part 14h).
3. Bring the required printed parts of your pre-lab report to turn in to your TA (as stated in the *Lab Rules and Policies*, part 14a-14d).

Note: All pre-lab requirements **MUST** be accomplished **PRIOR** to coming to your lab.

## PRELAB QUESTIONS

1. What minimum lab average is required in order to be **eligible** to pass the course?
2. Can you drop this lab if ... a) you overslept? b) project for other class due?
3. How late can you arrive for lab and still be admitted? How late can you arrive for lab and still be allowed to take the lab quiz?
4. What is the lab makeup policy if you miss a single lab?
5. When soldering a wire to a pin, what should the soldering iron touch and what should the un-melted solder touch?
6. What instruction can be used to read from program memory (flash)? Can you use any registers with this instruction?
7. What are the key differences between program and data memory? See section 7 in the ATxmega128A1U manual
8. When using RAM (not EEPROM), what memory locations can be utilized for the `.DSEG`? Why? What `.DSEG` did you use in this lab and why?

## LAB PROCEDURE

**Parts Kit:**
Obtain your 3744 lab kit from your TA. These parts will be used in the labs throughout the semester. **Verify** that your kit has **all the parts** listed on the checklist provided. **Immediately notify** your TA about any missing parts.

**Lab Rules and Policies and Introductions:**
Your TA will discuss the lab rules and policies (that you should have already read and agreed to as part of Homework 0) and then give you a general introduction to the laboratory and what will be expected from you for the semester.

**Simulation Demo:**
Demonstrate (to your TA) that your program from Prelab Part C creates the correct table. The TA will ask you to change the data values and/or filtering condition and then re-simulate/emulate your program. Your TA will also ask you to single step through your program and use breakpoints. Be prepared to answer questions about your program.

**Program and Emulate ASM Project**
In this part you will program your UF Board with the program you wrote in Prelab Part C and then run it on the uPAD.

Debugging with hardware is called **emulation**. Whenever possible, you should simulate your design first, before emulating; this eliminates any chance of mistaking hardware bugs for software bugs.

Load the program you created for Prelab Part C onto your board and verify that it functions properly. Demonstrate this emulation to your TA

**Board Construction:**
Your TA will demonstrate proper soldering techniques. Perform step 1 of the *uPAD Assembly Guide*, i.e., solder the female header through which you program your uPAD and emulate your program. For every major component, solder two pins (on opposite corners of the component) and **have your TA check your work before completing the soldering for that component**. Do **NOT** be overconfident; failure to follow this procedure will result in unnecessary errors that might cost significant **time, money, and lab points)**. If you are not sure, ask first **before** soldering.

After you finish the rest of your lab (or while waiting for the TA to get to you for check off) continue working through the *uPAD Assembly Guide*. As stated above, be sure to check with your TA regularly to verify that you are correctly building your PCBs.

You are expected to remain in lab until you are finished with the required soldering (or lab ends). If you do not finish, you need to finish before your Lab 1 begins (which probably means going to a TA office hour).