# ECE 375 PRELAB 3

Introduction to AVR Simulation with Atmel Studio

**Lab Time: Monday 12-1:50**

*Khuong Luu*

## ANSWER PRELAB QUESTIONS

1. **What are some differences between the debugging mode and run mode of the AVR simulator? What do you think are some benefits of each mode?**

   Debugging mode allows us to jump through our code line by line to examine our program's behavior closely while still compiling/assembling our code into machine code in the end to run on our board. In the other hand, run mode allows us to quickly compile/assemble our code into machine code to run on our board and see what happen, without having to step through our code line by line like in the debugging mode. The benefit of debugging mode is observing and examining our program closely step by step. This mode is suitable for finding bug in our program or analyzing our program usage's of memory, processors, etc. and thus help us optimize performance, strengthen security, etc.

   The benefit of run mode is quickly deployment our code onto our board, so that we can receive feedback from our board without being interrupted by breakpoints and other things. Thus via this mode, we can patch bug-fix, update, upgrade for our program quickly.

2. **What are breakpoints, and why are they useful when you are simulating your code?**

   Breakpoints are locations marked by programmer to (temporarily) halt the simulation at that location (typically a code line). This is useful because it allows us to observe all information about the program up to that point and we can also step over that one line afterward to see closely and clearly what have changed, which is the effect of the line, <u>without having to step over line by line from the beginning, and over and over again</u>.

   Breakpoints can also support conditional break and break count. These are very handy extended features of Breakpoints that allows us to skip a lot of manual step-over keystrokes while dealing with code lines that we are not always want to stop at it every time the program goes through it. With proper usage, these features would save us a lot of time and mental energy in debugging and analyzing our program.

3. **Explain what the I/O View and Processor windows are used for. Can you provide input to the simulation via these windows?**

   The I/O View window is used for not only displaying all the configuration registers but also simulating input on the I/O Ports associated with the simulated chip. We do this by clicking on the value being displayed on the I/O View window.

   The Processor window display the current contents of the Program Counter, Stack Pointer, the 16-bit pointer registers X, Y, Z, the Status Register, and the current values contained in each of the general purpose registers (register R00 – R31). Like the I/O View window, we can also simulate value of what the Processor window is displaying at the moment. We do this by clicking on the value being displayed on the Processor window.

4. **The ATmega128 microcontroller features three different types of memory: data memory, program memory, and EEPROM. Which of these memory types can you access by using the Memory window of the simulator?**

All three different types of memory - data memory, program memory, and EEPROM – can be access by using the Memory window of the simulator. To select which type of memory we want to access, on the Memory window, click on the dropdown menu and choose the type of memory. Specifically, to access the data memory, choose dropdown items that start with “data” e.g. “data registers”, “data IRAM”; to access the program memory, choose “prog FLASH”; to access EEPROM memory, choose “eeprom EEPROM”