

# Using GNU AVR Assembler with Atmel Studio

Aly El-Osery

Jan. 4th, 2018

## 1 Prefix

This document outlines how to use the GNU AVR Assembler `avr-as` with Atmel Studio. It requires that you have Atmel Studio already installed. The following steps outline how to create a GCC C/C++ project and use a custom Makefile. If you are familiar with Atmel Studio here are the key steps and you can skip most of the sections in this document

1. Create a **GCC C** project
2. Delete the `main.c` file that gets created automatically
3. Add an assembly file to the project
4. Download the Makefile from the following line and store it in the `debug` folder  
[http://www.ee.nmt.edu/~elosery/spring\\_2018/ee308/resource\\_files/](http://www.ee.nmt.edu/~elosery/spring_2018/ee308/resource_files/)
5. Edit the Makefile and change the project name and the file name to match your project
6. Open properties and select **Use External Makefile**
7. If you need the definition file for ATmega1284, it is also under  
[http://www.ee.nmt.edu/~elosery/spring\\_2018/ee308/resource\\_files/](http://www.ee.nmt.edu/~elosery/spring_2018/ee308/resource_files/)

## 2 Starting a New Project

The steps below outline how to start a new GCC project and add an assembly file to it. By default the creating a new project creates a `main.c` file which we are not using right now so you will have to remove/delete from the project.

1. Open Atmel Studio
2. Click **File**
3. Click **New**

4. Click **Project**
5. Select **GCC C Executable**
6. Change the **Name** to something representative of the project you are working on. (Note: Avoid using spaces in the name)
7. Unselect **Create directory for solution**
8. Select your **Device**
9. If it is not already open, click on **View** and select **Solution Explorer**
10. In the **Solution Explorer** right-click on the **main.c** and select **remove**. Select **delete** when prompted.
11. In the **Solution Explorer**, right-click on the project name and select **add then New item**
12. Select **Assembly File**
13. Rename the file to something representative

### 3 Using Custom Makefile

Now you have a project created with an assembly file ready to go. The following steps will show you how to use the provided Makefile rather than the defaults of Atmel Studio.

1. Download the Makefile file from the following link and save it to the **Debug** directory. It is located in your project folder.  
[http://www.ee.nmt.edu/~elosery/spring\\_2018/ee308/resource\\_files/](http://www.ee.nmt.edu/~elosery/spring_2018/ee308/resource_files/)
2. Now go back to Atmel Studio
3. Click **Project**
4. Click on **Properties**
5. Select **Build**
6. Select **Use External Makefile**
7. Browse to and select the Makefile you just downloaded

Now you have a Makefile you can open and manipulate. The first thing you need to do is to open the Makefile and change it to reflect the name of your project and the name of the assembly file you have chosen.

```
.nolist
.include "../m1284def.inc"
.list

5 .section .data
   .org 0x00
result: .byte

10 .section .text
   .global main
       .org 0x00
       rjmp main

15   ;can put some subroutines here if desired

main:
   ;put code to initialize the stack

20   ;some more code

   ;the subroutine

here:   jmp     here
```

Figure 1: Program skeleton

## 4 Definition File

The GCC assembler uses slightly different directives. If you need to use a definition file you can copy and modify the ones in the Atmel Studio. Or you can download the needed file for ATmega1284 from the following link to your project folder.

[http://www.ee.nmt.edu/~elosery/spring\\_2018/ee308/resource\\_files/](http://www.ee.nmt.edu/~elosery/spring_2018/ee308/resource_files/)

## 5 Test Your New Project

To make sure everything is setup correctly, try creating a simple code using the skeleton shown in Figure 1