ECE-304 Junior Design Project
Installing VSCode and the AVR-GCC toolchain on your computer
Spring 2022
Version 1.0

PART 1. INSTALL VISUAL STUDIO CODE

Visual Studio Code (aka VS Code) is an open source a text editor for writing source code. This is a popular editor among programmers because it has a number of features (eg, color coding, indentation, syntax, error checking, code folding, file management, etc..) that help to develop source code. Note that VS Code is not the same as another similarly-named Microsoft product, Visual Studio. Visual Studio is a full blown Integrated Development Environment, while VS Code is simply a text editor that happens to have many features.

You can find the instructions and the download files here:
Windows installation:
https://code.visualstudio.com/docs/setup/windows

macOS installation:
https://code.visualstudio.com/docs/setup/mac

You will likely only be using a small set of the full capabilities of VSCode.

PART 2. COMMAND LINE WINDOW

VS Code is used to create and edit source code. In PART 3 of this document, you will download the compiler toolchain, which is the set of software tools used to compile, link, and do other processing to convert your C source code into a binary executable file for download to your AVR microcontroller.

The toolchain is invoked from the command line. In windows, you will use the COMMAND PROMPT program to get a command line window. In macOS you will use the TERMINAL to get a command line window.

PART 2. INSTALL THE AVR-GCC TOOLCHAIN

Goto https://bytes.usc.edu/ee109/toolchain/ and follow the instructions there to download and install the avr toolchain. Our use of this link is courtesy of Prof. Mark Redekoop, Univ of Southern California (USC), with permission.

Windows Users:

If you're using a windows machine, this should be easy. Just follow the steps from the ee109 web site. You don't need to read further.

macOS Users:

If you're using a macOS, running macOS Catalina, Big Sur, or Monterey, the process is more complicated. The USC web site guides you through it, but you should pay extra attention in a few places as noted below:

Step 3: Install Link to AVR Toolchain: When you do this step, make sure you type avr after ...hardware/tools/avr, as shown by the arrow. This is easy to miss.

```
$ cd /usr/local
$ sudo ln -s /Applications/Arduino.app/Contents/Java/hardware/tools/avr avr
$ sudo mkdir bin
$ cd $HOME
```

Step 4:Adjust PATH Variable.

Your laptop probably is set up for the "zsh" shell by default, so you will be editing or creating the .zshrc file in your home directory. If your mac is using the "bash" shell, then you will be editing or creating the .bash_profile in your home directory. You can edit/create these files using VS Code.
When you add this line to you your .zshrc or .bash_profile shell, be careful to copy this line exactly as shown, taking note of all the /, :, $ and other characters.

```
export PATH=/usr/local/bin:/usr/local/avr/bin:$PATH
```

Step 5: Install the "avrdude" Program

Download the file "macos-avrdude.zip" from the USC EE-109 web site and follow the instructions for placing that downloaded file into /usr/local/bin

Step 6. Install a Working "make" Program

Three different ways of getting a working "make" program are given. Choose the third one: download a pre-compiled copy from the EE-109 web site then follow the instructions to move that file to /usr/local/bin.

Step 7: Test Installation

Here is a screenshot showing typical results of the "which" and "-- version" commands.