

## CS 341 – Lab 6

### Setting up Virtual Arduino

The goal in this lab will be to setup and get familiar with a program that mimics an Arduino Uno. The software is only runnable on Windows, so users of Mac and Linux will need to download a virtual machine. If you run into any issues please post your questions on Piazza under the “Lab” tab rather than emailing me.

## Step 1a: VM download for Mac OS

The iso file and its key are on the school’s user server. The files can be found under the path “/nobackup/faculty/cs341”.

### Method 1 (VMware)

1. Run “scp **username**@users.cs.umb.edu:/nobackup/faculty/cs341/\* .”
2. Open VMware and follow the prompts to add a new machine
3. provide the .iso file and the key in the .txt file when prompted
4. this [link](#) provides step by step instructions for Mac installing Windows
5. The Virtual Machine will need to start up and restart a few times to set itself up

### Method 2 (VirtualBox)

1. download VirtualBox if you don’t already have it
2. Run “scp **username**@users.cs.umb.edu:/nobackup/faculty/cs341/\* .”
3. Open VirtualBox and follow the prompts to add a new machine
4. This [YouTube video](#) does a step-by-step breakdown of the download process. You can start at 3:00 because we are providing the .iso file
5. provide the .iso file and the key in the .txt file when prompted
6. The Virtual Machine will need to start up and restart a few times to set itself up

At this point you should be able to startup the VM and access the internet via internet explorer. Next, we need **to get the Arduino simulator software**. We have found that you cannot download the software while inside the VM, nor download it on your host computer and email it. The instructions below are the simplest method, but you can also [set up a shared folder](#).

### Method 1 (Drag and Drop)

We have not been able to test this on a Mac but we believe it will work

1. On the host computer's finder navigate to UnoArduSimV2.7.1 (you should have gotten it from the scp command earlier)
2. Drag the folder onto the VM's desktop
3. If this doesn't work try method 2

### Method 2 (Filezilla)

1. Be inside the windows 7 virtual machine
2. Install [Filezilla 32-bit](#)
3. Connect to the school's server using your username + password:



4. Click quickconnect
5. On the right half of the screen you should have landed in your home directory. Navigate to “/nobackup/faculty/cs341/”
6. Drag the folder labelled “UnoArduSimV2.7” and drop it on the left half of the screen to transfer it to the VM
7. Exit Filezilla. Navigate to “UnoArduSimV2.7” using file manager.
8. Open the folder up and double click on “UnoArduSim.exe”

# Step 1b: Downloading without a VM

The software download is located [here](#). You should download 2.7 (the most recent version).

1. Windows users should open the .zip folder and save the files.
2. Double click “UnoArduSim.exe”

## Step 2: Introduction to Virtual Arduino



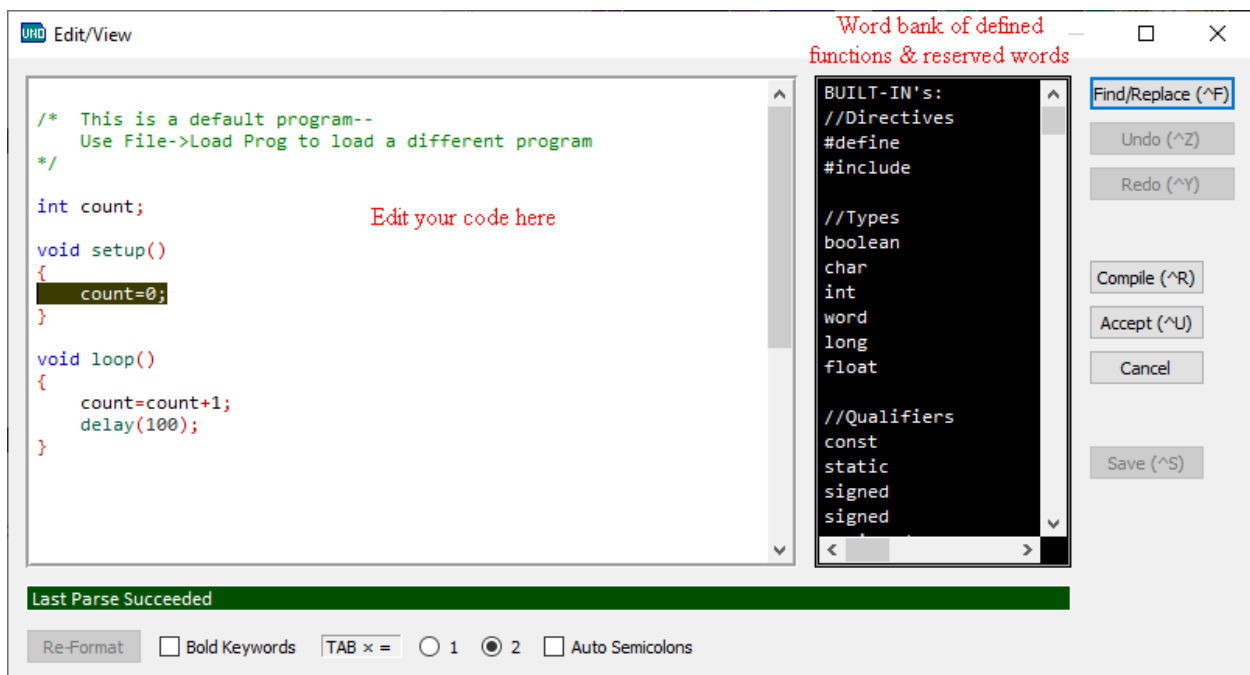
When you open up the software you should see something like the picture above. At the top you have your drop down menus. On the left is the code and variable tracker. On the right is the virtual hardware. You should see some familiar components (there are four 1K  $\Omega$  resistors on the left, and 2 LEDs in the top right). On the board in the middle we have digital pins (what we've used so far) and analog (which are unfamiliar).

This software does not allow us to do the circuitry. Instead we only have to specify the connections we want. For instance if we want to connect an LED to pin 3 it would look like this:



### Editing Code:

1. Open an existing .ino (arduino) file by going File > Load Ino
2. Double click on the code in the top left quarter of the screen (or ctrl+E)
3. You should see a window like below
4. Hit Compile, and Accept when you are done



### Useful:

- Execute > Animate Execution (will highlight the next-to-run line of code)
- Windows > Serial Monitor (opens the Serial Monitor)
- The documentation in the two provided PDFs is good

## Step 3: LEDs

Please make sure you have read all of Step 2 first. Now that you have an understanding of how to use UnoArduSim, your task is to blink an LED like we did in Lab 1. This will require you to do a bit of coding as well as connect the LED to the pin you decide to use. If you encounter any difficulty please post on Piazza. This isn't meant to be difficult or time consuming, we just want you to get comfortable with the software.

## Step 4: Lab Report

Please use the same lab template we have been using for this class. You can skip the “hardware” section if you would like. Because the majority of this lab was setup rather than new information you can keep your report short.

The plan moving forward will be for instructions like this to come out on Piazza once a week. The labs will be designed so that you can do them from home without any additional hardware. You can work in groups or by yourself. We will initially give you a week to complete the lab and submit a lab report by email ([jack.davis001@umb.edu](mailto:jack.davis001@umb.edu)). Please CC all group members so that my replies reach them as well. We will do our best to be flexible and understanding in this difficult time.