**CS 341 – Lab 1**

**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 and we will respond promptly.

VM download for Mac OS and Linux

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

**Method 1 (VMware)**

1. Use the command scp to copy files from the CS server onto your local computer. Here’s what you’ll need:

scp username@users.cs.umb.edu:/nobackup/faculty/cheungr/CD\_Key\_Windows7.txt .

scp username@users.cs.umb.edu:/nobackup/faculty/cheungr/Windows\_7\_Pro\_SP1\_x64\_VL.iso .

scp username@users.cs.umb.edu:"'/nobackup/faculty/cheungr/UnoArduSimV2.8.1 (1).zip'" .

**(mind the space and dot at the end of the command)**

1. Open VMware and follow the prompts to add a new machine
2. provide the .iso file and the key in the .txt file when prompted
3. this [link](https://kb.vmware.com/s/article/2128765) provides step by step instructions for Mac installing Windows if you get stuck
4. 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. Use the command scp to copy files from the CS server onto your local computer. Here’s what you’ll need:

scp username@users.cs.umb.edu:/nobackup/faculty/cheungr/CD\_Key\_Windows7.txt .

scp username@users.cs.umb.edu:/nobackup/faculty/cheungr/Windows\_7\_Pro\_SP1\_x64\_VL.iso .

scp username@users.cs.umb.edu:"'/nobackup/faculty/cheungr/UnoArduSimV2.8.1 (1).zip'" .

**(mind the space and dot at the end of the command)**

1. Open VirtualBox and follow the prompts to add a new machine
2. This [YouTube video](https://youtu.be/-TCgmY3kkC4?t=181) does a step-by-step breakdown of the download process. You can start at 3:00 because we are providing the .iso file
3. provide the .iso file and the key in the .txt file when prompted
4. 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](https://www.youtube.com/watch?v=iksh6SO9L5A&feature=emb_logo) on MacOS.

Get UnoArduSim on VM (MacOS and Linux)

**Method 1 (Drag and Drop)**

1. On the host computer’s finder navigate to UnoArduSimV2.8.x.zip (you should have gotten it from the scp commands earlier)
2. Right click to unzip and get UnoArduSimV2.8.x
3. Drag the folder UnoArduSimV2.8.x onto the VM’s desktop

**Method 2 (Filezilla)**

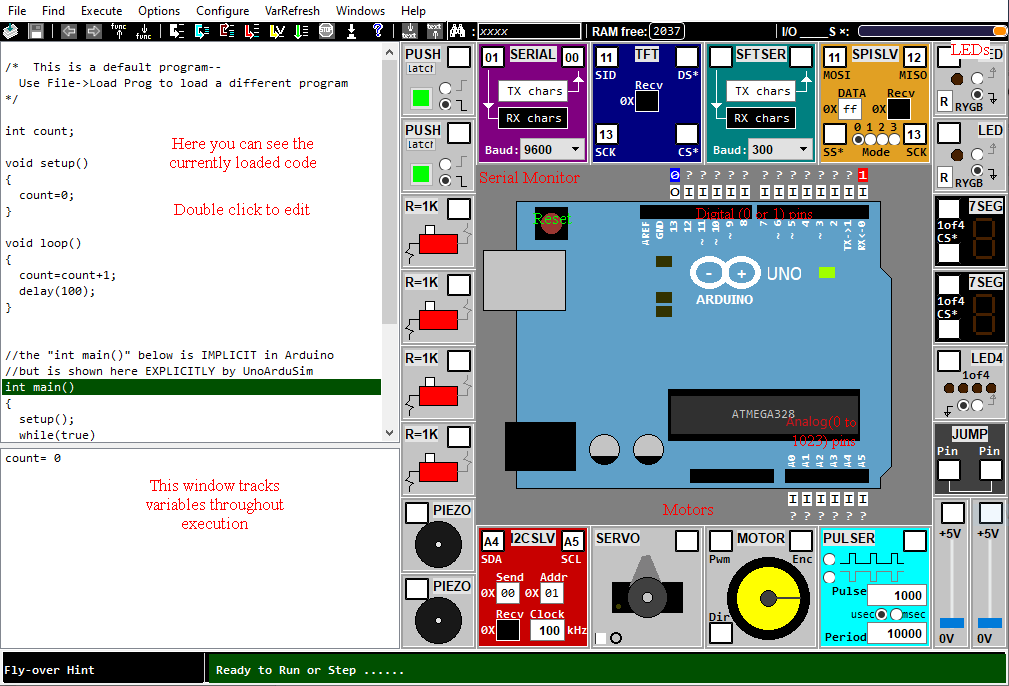
1. Be inside the windows 7 virtual machine
2. Install [Filezilla 32-bit](https://filezilla-project.org/download.php?platform=win32)
3. Connect to the school’s server using your username + password:
4. Click quickconnect
5. On the right half of the screen you should have landed in your home directory. Navigate to “/nobackup/faculty/cheungr/”
6. Drag the folder labelled “UnoArduSimV2.8.x.zip” and drop it on the left half of the screen to transfer it to the VM
7. Exit Filezilla. Navigate to “UnoArduSimV2.8.x.zip” using file manager.
8. Right click to unzip and get UnoArduSimV2.8.x
9. Drag the folder UnoArduSimV2.8.x onto the VM’s desktop

Windows Users

The software download is located [here](https://www.sites.google.com/site/unoardusim/services). You should download the most recent version available

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

Step 2: Introduction to Virtual Arduino



When you open 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 a variable tracker. On the right is the virtual hardware. You may recognize some of the components (there are four 1K Ω resistors on the left, and 2 LEDs in the top right). On the board in the middle we have digital pins (which read and right only high or low) and analog pins (which read and write on a range of 0-255).

A screenshot of a cell phone

Description automatically generatedThis 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

A screenshot of a cell phone

Description automatically generated

**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 can be quite helpful

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. Starter code is provided on [GitHub](https://github.com/jack17davis/cs341/lab1). Comments have been added to guide you along.

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 lab template found on [GitHub](https://github.com/jack17davis/cs341). Submit to [jack.davis001@umb.edu](mailto:jack.davis001@umb.edu), and CC all group members so that my replies reach them as well.