Practice Set 0 – Installing Arduino IDE

Flavor:

You cannot really code with the Arduino, if you have no way to upload your code to the board! The Arduino IDE allows you to do this.

Assignment:

Your assignment is to install the Arduino IDE using the CS50 appliance. Once this task is completed, you will plug your Arduino into your laptop or pc to make sure that your computer or laptop can recognize it. Afterwards, you will explore a few options inside of the IDE such as New, Save, Examples, and Serial Monitor. Afterwards, you will write a quick Arduino, “Hello World” program: forcing the Arduino onboard LED to blink.

Inventory:

1 Arduino

1 USB Cable

Schematic:

Library Function Reference

Setup()

Loop()

pinMode()

digitalWrite()

digitalRead()

delay()

Serial.println()

Instructions:

1. Start the CS50 Appliance and open the terminal window up.
2. Install the IDE Software by typing: <i>sudo yum install arduino</i>
3. After the software is installed, restart your appliance.
4. When your appliance has been restarted, grab an Arduino and USB Cable. Plug the larger male end of the USB into the Arduino’s power slot. Then, plug the smaller end into your computer. If you are using VMWare fusion, it will ask you where you want to route the USB Connection, in this case, choose Linux.
5. Click the menu in the button in the bottom left hand corner and proceed to the Programming Menu; click on the program name “Arduino”. It might be the case a message will come up asking for privileged access to some OS components - do grant access and logout of the appliance.
6. Log back into the Appliance and open the Arduino program once again. The program’s IDE should launch.
7. Take a moment and check out the article on <http://arduino.cc/en/Guide/Environment> to learn more about the environment, but if you are impatient like me, just go to step 8.
8. Download the Sketch in the next tab, and open it using the Arduino IDE.
9. The first <i>//TODO</i> requires understanding of pinMode(), which prepares the pin to be used. It accepts two parameters: first, the i/o pin number, which is the pin you want to utilize, and second, a value declaring whether the pin is used for input from the circuit (constant value <i>INPUT</i>) or output to the circuit (constant value <i>OUTPUT</i>).
10. Arduino UNO R3’s have a built in <i>LED</i> that is configured to pin 13. So, if we want the LED to light up, we need to tell the Arduino that it must listen on pin 13 because it will be <i>outputting</i> current. So, go ahead and set pinMode(13,OUTPUT).
11. The second <i>//TODO</i> requires understanding of digitalWrite(), which is used to interact with the pin that you configured using pinMode(). The first argument is the pin number that you are interacting with. The second argument is a constant that is either HIGH, meaning full voltage (on) or LOW, meaning no voltage (off).
12. We first want to turn the light on, to do this – we write: digitalWrite(13,HIGH).
13. Now, we want to turn the light off using digitalWrite(13,LOW) just under the previous command.
14. Upload the code to the Arduino board.
15. Why does the Arduino LED never blink? Funny enough, it is! We just don’t have super human vision (boo), so we need to slow things down to give us, “regular humans” time to see it blink. Let’s try the delay() function, which will stop the code from running based on the amount of time in milliseconds its parameter is given. For Arduino’s one second is equal to one thousand milliseconds.
16. Add delay(1000) after BOTH digitalWrite() function calls, and re-upload your code to the Arduino board.
17. And, that’s it! See <i>Challenges</i> for using the Serial Monitor, or go to Practice Set 1: Let me, ASCII <i>ya</i> name!

Challenges:

Use the Serial.println() to print in the Serial Monitor when the light is on, but remember – you need to first call Serial.begin(96000) some where in your program…. (I wonder where you put functions that need to load first? …hmm)

Sketch: