Summer 2019 Engineering Notebook

Jacob Smith

Jsmith2021@brandeis.edu

This engineering notebook contains my work during the summer of 2019 in the Automation Lab at Brandeis University. Project contained include

Arduino Class Generator:

A tool integrated to the Arduino IDE to create class body, header, keywords, and example sketch files https://github.com/jsmith2021Brandeis/Arduino Class-Maker

Arduino Education:

A set of workshops, tutorials, and background programs to get students started with Arduino https://github.com/jsmith2021Brandeis/ArduinoEd cation

(includes my helping two high school students refine their project https://github.com/AidenKunkler Peck/Tactile-Necklace)

Rasberry Pi Digital Window Project

Programs and setup for a series of digital windows that will allow the makerlab to communicate internally with itself

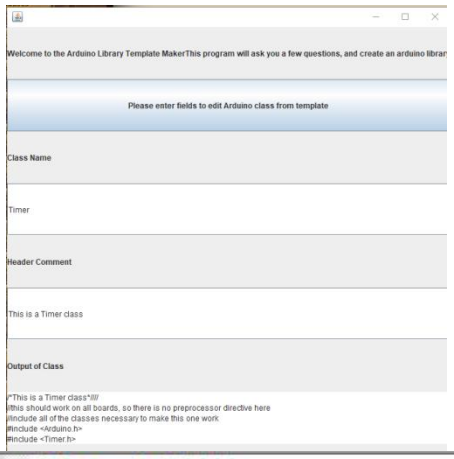
Table of Contents

Jacob Smith 5/16/2019 5:06 AM: I looked up how much information there is on Object oriented programming In Arduino, and https://forum.arduino.cc/index.php?topic=6127.0

Post 3 says that writing libraries is difficult, which helps to justify my automatic library creator

Jacob Smith 6:53 PM: I wrote a series of Java classes to allow for the user friendly creation of new Arduino Libraries. The ScriptEditor class allows for the simple reading and writing of files, the Arduino Class allows for the parsing and editing of Arduino Classes, and the GUI class creates a simple user interface.

Graphical User Interface of Library Creator→



I made these libraries because I found myself repeating a series of tasks every time I made a library like replacing the name and changing the header comment. The user interface accomplishes both of those actions by reading user input in textboxes and using it to modify a simply example class.

Template Class for Arduino Code generation→

```
1   /*Example*////
2   //this should work on all boards,
3   //include all of the classes nece
4   #include <Arduino.h>
5   #include <Template.h>
6   //Template constructor
7   Template::Template(){
8
9   }
10  //resets the initial time
11  long Template::methodl () {
12
13  }
```

While this isn't that helpful on its own, these classes show the possibilities for making it easier to create arduino classes, in the future the user could add a global variable and have both the header and cpp files automatically updated, currently only the cpp file is changed.

Jacob Smith 5/13/2019 5:19 AM: I am rewriting the Java ArduinoClass to build a cpp file from a template, and I am researching arduino libraries some more. In addition to my current scheme, auomatic conversion of an arduino sketch (.ino) to an arduino library (.cpp) would be very useful

[1] https://arduino.stackexchange.com/questions/32998/how-to-convert-arduino-example-sketch-to-a-complete-c-project

[2] https://community.platformio.org/t/tutorial-for-creating-multi-cpp-file-arduino-project/5830/13

5/14/2019 Jacob Smith: I wrote an inheritance hierarchy of common methods in ArduinoClassMaster, body methods in ArduinoClassCpp, and header methods in ArduinoClassH. Left is Autogenerated header and keywords file from ArduinoClassH

Automatically Generated Header file→

```
/* Written by Jacob Smith for Brandeis Robotics Club 2019/05/14
A timer class to allow the user to create loops and maintain program contro
Boards supported: ARDUINO_AVR_UNO ESP8266_WEMOSD1R1*/

//sets up the Timer Header file
#ifndef Timer_h
#define Timer_h

//only compile this class if the board is correct
if defined(ARDUINO_AVR_UNO) | defined (ESP8266_WEMOSD1R1)

//includes the libraries of code necessary to make this one work
#include <Timer.h>
#include <Apple.h>

class Timer{
  private:
    //the beginning time of the interval
    long initTime;
    //a test varible for the parser
    Apple test;
  public:
    //Creates a new Timer object
    Timer();
    //resets the Initial Time
    long resetTime();
    //returns the current time
    long getTime();
    //returns the current time and the initial time
    long getAndResetTime();
};
#elif defined (DONT_NEED_TIMER)
  #warning : May lead to error: 'Timer' does not name a type ; Program to Bo
#else
  #error : Program to Board Incompatibility ; One of the libraries you are u
#endif
#endif

//Generates ARDUINO KEYWORDS for Timer class
Timer   KEYWORD1
Timer   KEYWORD2
resetTime       KEYWORD2
getTime KEYWORD2
getAndResetTime KEYWORD2
```

Jacob Smith 5/15/2019 1:23 AM: I restructured the github folders, here are some toggle button schemes that Deveroah and I thought of a while ago: **Wireless Button Driving Schemes→**

I have been working on the user interface for Arudino class generator, and I am fidnign hat putting the required varibales names, prompts, and example formatting in an enum makes my client code easy to read and write. →

Motion-Stop

Dive F -stoppped

Drive B -stopped

Turn L-stepped

Turn R=stopped

Direction-Turn

Drive F-DriveB

Turn L-Turn R

Move- Stop

Pin Directed

Drive--Stop

Forward--Backward

Turn Right--Stop

Turn Left--Stop

```
public enum ArduinoClassField {

    CLASSNAME       ("Please enter the name of the class"
    AUTHOR          ("Please enter the name of the author"
    ORGANIZATION    ("Please enter your organization"
    HEADERCOMMENTS  ("Please enter a description of the class"
    SUPPORTEDBOARDS ("Please enter the boards your class can work with/ALL if any board
    VARIABLES       ("Please enter the type and name of class variables"
    PRIVATEMETHODS  ("Please enter the return type, name, description, and body of the
    PUBLICMETHODS   ("Please enter the return type, name, description, and body of the

    public final String prompt;//the prompt for the user
    public final String format; //the proper formatting of the field

    /** A private constructor to create an enum with prompt and format
     * */
    ArduinoClassField(String prompt, String format) {
        this.prompt = prompt;
        this.format = format;
    }
}

/** returns the name of the ArduinoClassField
 * */
public String getname() {
   return this.name();
}

/**Displays capabilities of the enum
 * */
public static void main(String[] args) {
    System.out.println("Prints out the different enums, prompts, and f
    for (ArduinoClassField a : ArduinoClassField.values()){
        System.out.println(a);
    }

}

/**returns a string representation of the enum, with name, prompt, and
 * */
public String toString(){
    String enumString="";
    enumString+=this.name();
    enumString+="\n\t"+prompt+"\n";
    //insert Tabs into any of the multiline formats so they print well
    String formatWithTabs=ArduinoParser.insertTabs(format,2);
    enumString+=formatWithTabs;
    return enumString;
}

    return enumString;
}
```

ArduinoFields Enum for Code Readability and to allow all field related prompts to happen In one place

My Journal Notes on the Project

Unknown Date:

had to reset project tested prompt method in prompt enum, going to display DONE prompt more cleanly and also, I have to break up internally parsed prompts better I am using recursion to handle the complex formatting→ is on the innermost level tokens of |, then tokens of newlines, then tokens of double newlines while this format is complicated, it is both computer and human readable

void|driveLeft|tells robot to Drive Left robot.driveLeft() void|driveRight|tells robot to Drive Right robot.driveRight()
Example of Method Formatting

I am working on a field by field prompter, handling initial conditions before loop does simplify code

Sunday: I am only allowing allowed characters, to hide which characters I am using for parsing and also exclude strange characters I haven't thought of

I am replacing all instances of scanner read next line with my validating looping method

It seems like using an enum for prompting is too complicated, I will just use methods

I am handling special cases manually, simplifies my methods and makes code more readable

I am using interface to save code enums can implement interfaces

Manually typing in user responses for testing is tedious, so with print streams and enums, I can now just run a test instead of manually typing in test input for ArduinoClassPrompts.

I am now at the point where I can prompt field by field and try to generate the Arduino class, but I found methods are missing | character set breakpoints as you work with debugger, maybe directly from runtime exception found a parsing error in variables, explains why only header file had it, I've got an extra newline in variables I fixed that error which was missing |, but methods had extra newline, I removed it with substrings Now test runs correctly, but public methods aren't showing up TO DO: not do substring by flipping with fencepost solution, fix public methods, unit test error corrections public methods was newline issue,

Now that I added constructor, the output essentially looks like an Arduino class. I switched order of private and public methods in body file, which puts constructor at top and puts most visible methods at top too line by line unit test passed.

Created Arduino class maker GitHub repository so I can actually delete unused code and know it is backed up

May 23 I am working on the program that converts an existing Arduino sketch to a library, which required a lot of background code work. I modified the MiniScanner class to display an error with the word that was looked for, and the Arduino Class generators to allow method parameters of null if blank. The methodParser class handles the work of converting a method into the format the code generator uses, and I want the program to automatically generate an example sketch. The sketch parser should

also be able to tell which methods are public vs private by looking for which methods are in the setup and loop methods The parsedMethod class has fields even though I could have the whole thing be one big toString. The SketchParser class currently can read a sketch into the code representing the header, variables, public and private methods, loop and setup, but they aren't formatted into the format the Arduino class generator needs

Now that the SketchParser method can convert my sketch into a fairly well formatted list of header comment, variables and methods,the next step is separating private and public methods and parsing variables. The variables parsing will become fairly interesting/complicated, because they should all go to a constructor. This means that the sketch to library program is forcing me to add some features like adding constructor and adding method parameters.

The method correctly decided that wifiConnect and run server where public methods I am using replaceAll String method a lot, very useful in reformatting sketch. I am writing ParseVariable method with test cases to convert code of variable declaration to parsable format. This is complicated because data types and values can be more than one word, and arrays are in form type name [], and variables don't have to be initialized .

5/24 10:51 PM: Working on Arduino parser and unit tests, I got test to pass when variable is declared not initialized, still failing Linked List switch nodes (I wrote linked list to iterate over variables which can have multiple words like const char) and eliminating extra vertical bars. passed extra vertical bars test using String. It seems that using String.replace works when String.replaceAll does not, I don’t know why.

5/25 10:48: FUTURE WORK: I want to integrate this project into the wider Arduino Environment by reformatting and compiling files before they are parsed. This would allow me to standardize text before I have to parse it, and return an error in advance if program doesn't compile. Also, I want to eventually make a pull request to have the automatic class generator be part of the Arduino ide. Relevant links:

[1] Arduino Command Line Interface
https://github.com/arduino/Arduino/blob/master/build/shared/manpage.adoc
[2] Jar files already part of Arduino https://github.com/arduino/Arduino/tree/master/arduino-core/lib
[3] How to create a jar file in eclipse http://www.skylit.com/javamethods/faqs/createjar.html

May 27, 2019 10:34 AM: I looked up who else is working on an Arduino Class Generator on the Arduino developers’ group, no one there is. I'm working on passing more of the VariableParser unit tests, so I wrote a MiniScanner getRest method to help.

CODE STYLE: Else statements are useful in avoiding something right before exiting a loop. In that scheme, use Boolean loop control variable as condition. Set variable in loop, and use else statement to do whatever should be done while looping is chugging, but not when it is about to exit. This is a more subtle version of a break statement.

12:58 PM: Working on LinkedList switch method for variable parser, I wrote a state method that handles null pointer exceptions, and made setter methods private to force its use

1:13 PM: Arduino IDE Auto format inserts spaces but doesn't remove them, I'll have to do that on my own, which should be using the MiniScanner class internally.

3:07 PM May 28, 2019: I got the MiniScanner to allow it to ignore multiple tokens in input, but now a lot of other unit tests fail

6:58 PM: I spent a lot of time today trying to get rid of scanner errors, and put the project on GitHub so I wouldn't have to redo my work again

11:00 PM I got all unit tests to pass, and having project on GitHub with commits on passing tests makes it much easier to keep project moving forward with unit tests passing, and creating branches is useful. In the SketchParser class, I am working on an autogenerated constructor based on variables. Before that, I am preparing the program to handle the https://www.arduino.cc/en/Hacking/LibraryTutorial Morse code sketch as is so I can post it to the Arduino developers forum https://groups.google.com/a/arduino.cc/forum/#!forum/developers. The challenges so far with that have been to reformat the newline brackets style of coding into the same line and to create to do comments when comments are missing, which is what I'm working on now.

May 29, 2019 10:37 AM: last night, I got the SketchParser class to produce keyword, example file, body, and header for the ESPServer and Morse classes. The unreliable parts are automatic constructor generation and how the sketch file doesn't call based on object. In addition, I posted my project idea on the GitHub developers forum, and I got a response from someone to share my GitHub link, although I haven't decided whether to keep my code private yet.

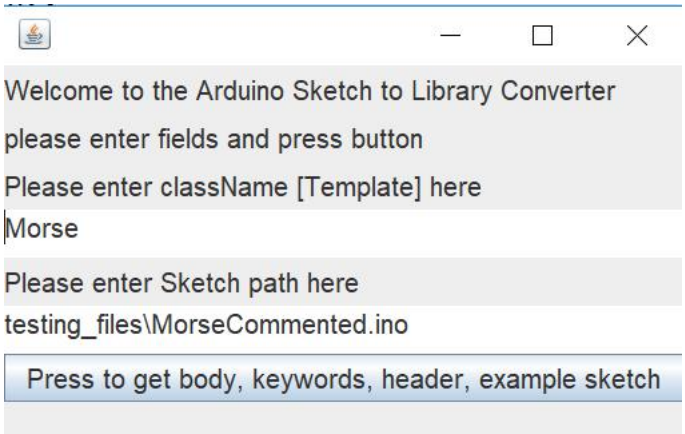
7:18 PM: I got the generator to change method calls in example sketches to be an object method, like converting dash(); to morse.dash(). Also, I got the GUI to be useful, now the user can click on

RUNme.jar, and this window pops up→

The user can enter the name of the class and location of the correct Arduino sketch.

Then, when the button is pressed, the following windows pop up (next page).

Future work: Integrate this into the Arduino IDE: https://github.com/arduino/Arduino/wiki/Building-Arduino, research whether I should publish this program.



User input interface, runMe.jar

Original Sketch

```
/*Written by Jacob Smith for Brandeis University.
 Allows arduino to cmmunicate with morse code*/
//the pin to flash on
int pin = 13;

//runs once, sets up pins
void setup()
{
  pinMode(pin, OUTPUT);
}

//runs many times, flashes a message
void loop()
{
  dot(); dot(); dot();
  dash(); dash(); dash();
  dot(); dot(); dot();
  //wait 3 seconds...
  delay(3000);
}

//plays a dot on the pin
void dot()
{
  digitalWrite(pin, HIGH);
  //wait quarter second...
  delay(250);
  digitalWrite(pin, LOW);
  delay(250);
}

//plays a dash on the pin
void dash()
{
  digitalWrite(pin, HIGH);
  //wait one second...
  delay(1000);
```

Class Example File

```
/*Written by Jacob Smith for Brandeis University.
 Allows arduino to cmmunicate with morse code 2019*/

//includes the libraries of code necessary to make this
#include <Morse.h>

//The object used to interfact with the class
Morse morse();

//runs once, sets up pins
void setup() {
   pinMode(pin, OUTPUT);

}

//runs many times, flashes a message
void loop() {
   morse.dot(); morse.dot(); morse.dot();
   morse.dash(); morse.dash(); morse.dash();
   morse.dot(); morse.dot(); morse.dot();
   //wait 3 seconds...
   delay(3000);

}
```

Class Header File

```
/*Written by Jacob Smith for Brandeis University.
 Allows arduino to cmmunicate with morse code 2019*/
//sets up the Morse Header file
#ifndef Morse_h
#define Morse_h

//this should work on all boards, so there is no preproc

//includes the libraries of code necessary to make this
#include <Morse.h>

class Morse{
 private:
   //the pin to flash on
   int pin;
 public:
   //Creates a new Morse object
   Morse();
   //plays a dot on the pin
   void dot();
   //plays a dash on the pin
   void dash();
};
#endif
```

Class keywords file

```
//Generates ARDUINO KEYWORDS for Morse class
Morse	KEYWORD1
dot	KEYWORD2
dash	KEYWORD2
```

(TOP) Generated Example, Header, keywords, and original sketch. From runme.jar

(BOTTOM) Generated JDK error message because FJD must be 32 bit

How to build Arduino IDEhttps://www.mkyong.com/ant/how-to-install-apache-ant-on-windows/

Why I need 32 bit JDK https://github.com/arduino/Arduino/issues/3276

May 30 1:00 AM: I can now compile and run the Arduino application; everything is in the Editor class. I can now add a menu option called generate Library, and I'm trying to use the tab creation feature to generate all the correct files at once. Then, I need to add the relevant classes into the Arduino folders.

May 30 2:52 PM: I am trying to automatically create files in Arduino ide, relevant functions: Addfile Sketch.java line 311, SketchFile.java line 94, EditorTab.java createTextArea line 146, EditorTab.

```
C:\Users\jsmit\Downloads\Arduino\build\windows\work>arduino_debug.exe
java.lang.UnsatisfiedLinkError: C:\Users\jsmit\Downloads\Arduino\build\wi
lib\listSerialsj.dll: Can't load IA 32-bit .dll on a AMD 64-bit platform
        at java.lang.ClassLoader$NativeLibrary.load(Native Method)
        at java.lang.ClassLoader.loadLibrary0(ClassLoader.java:1941)
        at java.lang.ClassLoader.loadLibrary(ClassLoader.java:1824)
        at java.lang.Runtime.load0(Runtime.java:809)
        at java.lang.System.load(System.java:1086)
        at processing.app.Platform.loadLib(Platform.java:162)
        at processing.app.Platform.<clinit>(Platform.java:157)
        at java.lang.Class.forName0(Native Method)
        at java.lang.Class.forName(Class.java:264)
        at processing.app.BaseNoGui.initPlatform(BaseNoGui.java:515)
        at processing.app.Base.<init>(Base.java:206)
        at processing.app.Base.main(Base.java:151)
```

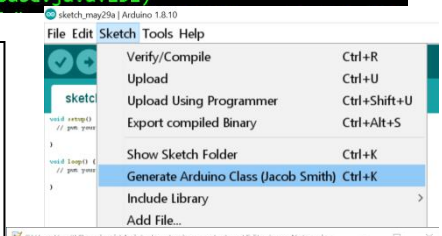
(Middle)→ The menu option to generate Arduino Class.

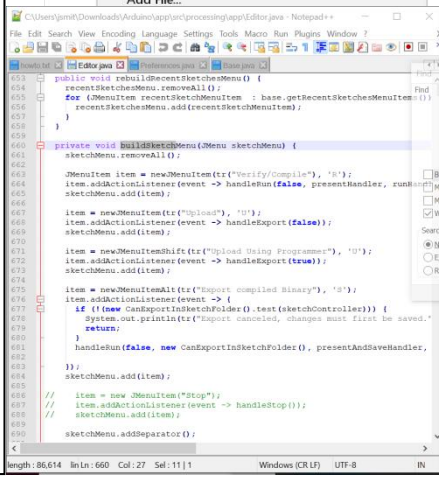
(Bottom)→ The area of the Editor class in the Arduino IDE that I will be modifying for the GUI.

SketchControler.java 136, nameCode

Sketch Controller Line 598 import Library

Sketch Controller upload line 598 (some boards have a password)





Relevant background:
https://github.com/sudar/Arduino-Makefile More control of Arduino Compiler
Arduino Tabs are just a way of breaking up a sketch
https://forum.arduino.cc/index.php?topic=206078.0,
Nice tutorial for library creation
http://arduino.land/FAQ/content/7/43/en/breaking-a-sketch-into-multiple-files.html
Using sketchContoller code avoids java exceptions and lets you create new tab.

11:00 PM: I am writing a script to compile the Arduino ide, so I don't have to repeat a manual task by going to a certain directory and deleting the dll file. The dll file gets a wired error access is denied, when the real error is that the file is already running

12:50 AM: I can now generate tabs for the .cpp,.h, keywords.txt, and example file.ino files. I am now getting the string of the file name and file contents to pass to my Arduino class generator.

2:41 AM: I can now set the text of the files and save automatically, now I will abstract the code into a method that takes the four strings of the body, text, keywords, and example files, the only unknown inputs
Useful method:
Build method sketch controller.java.

```
public class Actions {
  public final Action newTab = new SimpleAction(tr("New Tab"),
      Keys.ctrlShift(KeyEvent.VK_N),
      () -> editor.getSketchController().handleNewCode());
```

Editor Header.Java Line 88 , shows how new tabs are generated

```
199   // currently opened program
200   SketchController sketchController;
201   Sketch sketch;
```

Editor.java line 199 fields, sketchController allows new tabs to be created

```
public void edit(String message, String dflt) {
  changeState(EDIT);
  this.message = message;

  okButton.setVisible(true);
  cancelButton.setVisible(true);
  editField.setVisible(true);
  editField.setText(dflt);
  editField.selectAll();
  editField.requestFocusInWindow();

  repaint();
}
```

EditorStatus.java line 147, the method used to get the file name, which I want to set automatically

```
final EditorHeader header;
EditorStatus status;
EditorConsole console;
```

Editor.java line 193 status field which is used to display messages

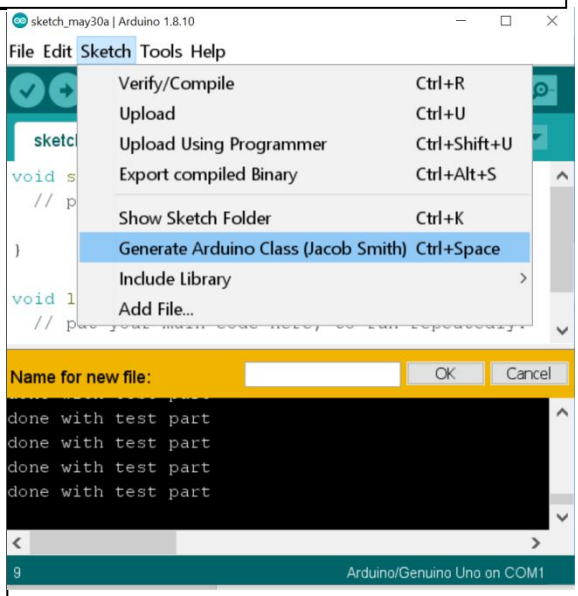
```
item = newJMenuItem(tr("Generate Arduino Class (Jacob Smith)"),' ');
item.addActionListener(event -> {this.generateLibrary();});
sketchMenu.add(item);
item.setEnabled(Base.openFolderAvailable());
```

```
okButton.addActionListener(e -> {
  // answering to rename/new code question
  if (mode == EDIT) {  // this if() isn't (shouldn't be?) necessary
    String answer = editField.getText();
    editor.getSketchController().nameCode(answer);
    unedit();
  }
});
```

↑Initialize Method of Editor Status.java line 240, shows how ok button is wired to create a new tab

Arduino IDE showing the new menu option, file name dialogue, and console printout→

← Editor Java Line 697, showing how keyboard shortcut finally works



Clockwise from left. These pictures show: how the menu option is shown, how new tabs are created in example IDE code, the menu option being used to display the generate class option which now works with keyboard shortcut, and the batch file that compiles the Arduino application.

This stage of the project was about reverse engineering the Arduino ide to connect it to the class generator API. To do this, I used GREP WIN to search for where the text of a menu option occurred in code to see how it is used and what source functions are important. This allowed me to write my own functions to create tabs for the class information.

```
rem **Builds and runs the arduino IDE for testing Jacob Smith
rem reference https://superuser.com/questions/757146/windows-7-cmd-command-to-keep-cmd-open-after-executing
rem reference https://www.wikihow.com/Delete-a-File-Using-Command-Prompt
rem reference https://superuser.com/questions/1189975/how-can-i-force-stop-a-program-without-using-the-mous
rem reference https://superuser.com/questions/1179510/wait-for-a-process-to-complete-in-cmd
rem **delete file that results in compilation error
cd C:\Users\jsmit\Downloads\Arduino\build\windows\work\lib
taskkill /F /IM "javaw.exe" /T
del jnidispatch-4.2.2-win32-x86.dll
rem  go to directory, compile run the arduino package
cd C:\Users\jsmit\Downloads\Arduino\build
call ant build
call ant run
```

RunArduino.bat file to compile and run the Arduino package automatically.

4:46 AM: I wrote this method→ to stop library generation if sketch doesn't compile, it is based on the build method of Sketch Controller class which returns null if the build failed.

Currently, the messages aren't displayed after the couple seconds it takes to compile the program, probably because I'm not calling a repaint method.

I am working on abstracting the inputs and outputs of the sketch generator interface in the same manner as this method, notice how it is one method that can be used easily and doesn't rely on global variables.

I'm trying to do this for setting the texts of the tabs, which was complicated because they are automatically alphabetized, but I'm creating them in alphabetical order now.

CODE STYLE: I'm making methods static before I put them in their own class. This will prove that they don't rely on instance variables, making them easy to put into their own class.

```
734  * compiles the sketch before arduino class is generated
735  * to force the user to submit a correct class
736  * @return true if class failed to compile
737  */
738 private boolean failedToCompile(SketchController controller) {
739     System.out.println("Compiling sketch to check for errors...");
740     String output="failed";
741     try {
742         output=controller.build(true, false);
743     }catch(Exception e) {}
744     if(output.equals("failed")) {
745         System.out.println("Error, the sketch failed to compile, ex
746         return true;
747     }else {
748         return false;
749     }
750 }
```

(TOP) My FailedToCompile method in Editor.java to stop generation of library if sketch doesn't compile. (BOTTOM) usage of method in generate library client code to stop method if compilation fails with screenshots of ide in both use cases.

```
private void generateLibrary() {
    if(failedToCompile(sketchController)) {return;}
```

sketch_may31d § | sketch_may31d.cpp § | sketch_may31d.h § | sketch_

```
//this will compile and make tabs
void setup() {}
void loop() {}
```

```
Compiling sketch to check for errors...
Sketch uses 444 bytes (1%) of program storage space. Maximum
Global variables use 9 bytes (0%) of dynamic memory, leaving
```

sketch_may31c §

```
This won't compile
```

```
sketch_may31c:1:1: error: missing ter

 This won't compile

 ^

sketch_may31c:1:1: error: 'This' does

exit status 1
Error, the sketch failed to compile,
```

WORKFLOW: The Arduino IDE lets me input my data directly from the sketch contents and file name and compile the sketch beforehand. This means that I deleted all the classes that prompt the user for input and validates the response. I also separated tests and source files from each other, which will make it easier to put the source files into the Arduino ide, possibly with a jar file to encapsulate and hide them. Potential bug: Some fields can be read to the Arduino class generators as null or "null", I should allow both.

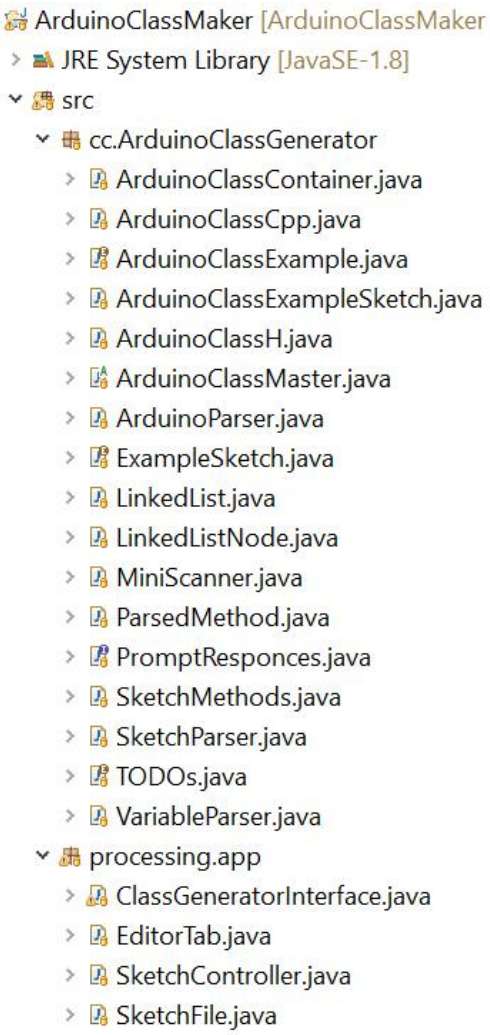
June 3 2019 10:15 PM: After trying out working environments, I settled on this structure→**,** where all of the classes to generate an Arduino class from a string are in the ArduinoClassGenerator package, and the interface is in the processing.app class.

The interface needs to be in its own class because it calls the nameCode method of the SketchController class to create tabs, which is a protected variable. →

The other members of the processing.app class are dummy classes so the referenced classes in the ClassGeneratorInterface don't result in errors.

Finally, I can build the arduino IDE with the main method of the ArduinoClassGenerator class, which calls my batch file. →

This setup means that all of the required files are in the ArudinoClassGenerator repository (as opposed to having to juggle classes in Arduino build and the batch file being in my desktop to run)



TOP: new Arduino File structure
Below: Error building arduino, sketchController var
Bottom: main method of ArduinoClassGenerator class calling batch file

```
error: nameCode(String) has protected access in SketchController
```

```java
/**
 * Builds and Runs the Arduino IDE by running batch Script RunArdu
 * @param args not used
 */
public static void main(String[] args) {
    try {
        Runtime.getRuntime().exec("cmd /c start RunArduino.bat");
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
```

The batch script is shown at left→
Line 15-18 automatically copy files from the ArduinoClassMaker workspace into the Arduino IDE.
Also note the use of function declaration line 33 to reduce redundancy.
The reason I didn't make my project a fork of the Arduino github project is because I couldn't import it as an eclipse project and because I can't make the fork private→
I looked up using .jar files to package the class generator, but I couldn't find an example of one being called from java code. Finally, I am including the only other additions to the Arduino IDE necessary to use the class generation feature.
The left shows the Editor class with the code necessary to add class generation as a menu option, which calls the generate library method of the ClassGenerator Interface class.
The last modification to the IDE was to allow .txt files→, so I can generate the keywords.txt file. This change is unlikely to be accepted by the developers of Arduino, but I could generate the keywords file without opening it.

Make this repository private
Public forks can't be made private. Please duplicate the repository.

TOP: Github website privacy message
LEFT: Editor.java line 708 showing addition of Line 22 of Sketch.java allowing .txt file for keywords
BOTTOM: Batch file to build Arduino IDE

```
707 +
708 +   item = newJMenuItem(tr("Generate Arduino Class"),' ');
709 +      item.addActionListener( event -> ClassGeneratorInterface.generateLibrary(sketchController,getCurrentTab(),getCurrentTab().getSketchFile(),tab
            s));
710 +   sketchMenu.add(item);
```

```
final List<String> OTHER_ALLOWED_EXTENSIONS = Arrays.asList("",".txt");
```

```
1  rem **Builds and runs the arduino IDE for testing Jacob Smith May 31 2019 Personal Study
2  rem reference 1 https://superuser.com/questions/757146/windows-7-cmd-command-to-keep-cmd-open-after-executing
3  rem reference 2 https://www.wikihow.com/Delete-a-File-Using-Command-Prompt
4  rem reference 3 https://superuser.com/questions/1189975/how-can-i-force-stop-a-program-without-using-the-mouse-in-windows-10
5  rem reference 4 https://superuser.com/questions/1179510/wait-for-a-process-to-complete-in-cmd
6  rem reference 5 https://www.tutorialspoint.com/batch_script/batch_script_if_statement.htm
7  rem reference 6 https://stackoverflow.com/questions/734598/how-do-i-make-a-batch-file-terminate-upon-encountering-an-error
8  rem reference 7 https://superuser.com/questions/101974/play-a-sound-maybe-way-from-windows-line-command/528541
9  rem reference 8 https://stackoverflow.com/questions/4983508/can-i-have-an-if-block-in-dos-batch-file
10 rem reference 9 https://stackoverflow.com/questions/40610241/automatically-answer-to-input-prompt-in-windows-batch
11
12 rem temp, sets computer to shutdown so I have to take a break
13 shutdown /s /t 1600
14 rem copy files from eclipse workspace to arduino ide and allow all files to be copied, see ref 9 for auto respond
15 call echo all|xcopy /E /I C:\Users\jsmit\Documents\ArduinoClassMaker\src\cc\ArduinoClassGenerator C:\Users\jsmit\Pictures\Arduino\app\src\cc\ArduinoClassGenerat
16 call all
17 call echo all|xcopy /E /I C:\Users\jsmit\Documents\ArduinoClassMaker\src\processing\app\ClassGeneratorInterface.java C:\Users\jsmit\Pictures\Arduino\app\src\pr
18 call all
19 rem **delete file that results in compilation error ref 3
20 cd C:\Users\jsmit\Pictures\Arduino\app\lib
21 taskkill /F /IM "javaw.exe" /T
22 rem delete the file, see ref 2
23 del jnidispatch-4.2.2-win32-x86.dll
24 rem  go to directory, compile run the arduino package
25 cd C:\Users\jsmit\Pictures\Arduino\build
26 rem see 4 for call, which forces commands to be sequential
27 call ant build
28 rem these two lines stop the script and play a sound, see reference 1,4,5,6,7,8
29 call :holdError
30 call ant run
31 call :holdError
32 exit
33 :holdError
34    if %errorlevel% neq 0 (
35       start wmplayer "C:\Windows\Media\Alarm10.wav" && timeout 5 && taskkill /im wmplayer.exe
36       pause
37       exit
38    )
39    EXIT /B 0
```

This is the ClassGenerator Interface class, which is all of the logic necessary to: read the file name and contents of a sketch, generate the class body, header, keywords, and example file, and display them to the IDE.

The failed to Compile method is 3 pages ago.

The getNameContentsPath method gets the filename and contents of the sketch without requiring the user's input.

The SetLibraryTabs method actually creates the tabs in the IDE to display the generated class files.

The IDE with current tabs is shown at lower right corner. They are currently not saved automatically so the user can look over them before saving.

FUTURE WORK: A) writing unit tests that cover the generator's behavior under different types of sketches, B) create interactive user prompts to get the author name, required boards, and organization, display a loading dialogue when class is being generated. C) research whether this project should become a public fork and pull request to Arduino, or whether I should keep it private.

6/10/2019 1:14 PM: Pito Salas documentation, gif creator, gui multithreading, prompt messages.

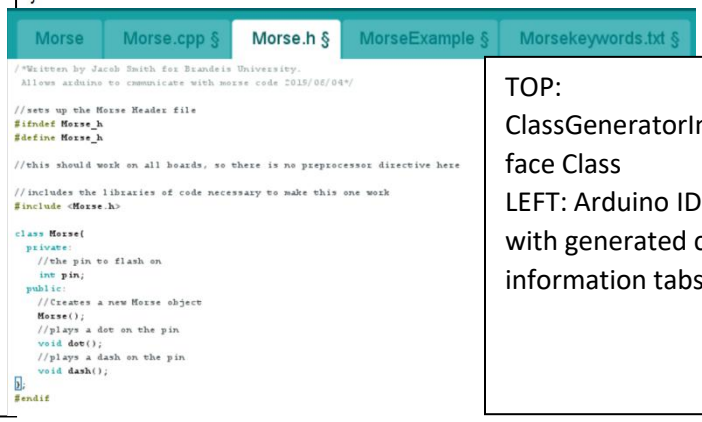
Because the Class Generator takes a couple seconds to complete, I wanted progress messages to be displayed to show the user something is happening. I can use the progressNotice method of the

```
public static void generateLibrary(SketchController controller, EditorTab tab, SketchFile sketchF
        ArrayList<EditorTab> tabs) {
    if (failedToCompile(controller)) {
        return;
    }
    // get the file name and contents of the sketch
    String[] sketchInfo = getNameContentsPath(tab, sketchFile);
    String className=sketchInfo[0];
    String contents=sketchInfo[1];
    String filepath=sketchInfo[2];
    //parse the sketch into format used to generate arduino classes
    SketchParser parser = new SketchParser(contents);
    ArduinoClassContainer cont = parser.getContainer(className, false);
    // create the files with strings
    setLibraryTabs(className, cont.getBody(),  cont.getExample(), cont.getHeader(),cont.getKeyworc
            controller);
}
```

```
private static String[] getNameContentsPath(EditorTab currentTab, SketchFile sketch
    // create array to return sketch information in
    String[] sketchInfo = new String[3];
    // load sketch file name
    String fileName = sketchFile.getFile().getName();
    // remove extension from fileName
    sketchInfo[0] = fileName.replaceAll(".ino", "");
    // load sketch contents
    sketchInfo[1] = currentTab.getText();
    // load sketch path
    sketchInfo[2] = sketchFile.getFile().getPath();
    return sketchInfo;
}
```

```
private static void setLibraryTabs(String className, String body, String example, Strin
        ArrayList<EditorTab> tabs, SketchController controller) {

    // create and set text of all tabs
    // the name of every tab starts with className, because tabs are alphabetized
    // create body tab
    controller.nameCode(className + ".cpp");
    // get the tab from the tabs field because nameCode doesn't return the tab
    // these indexes get complicated because of alphabetization
    tabs.get(1).setText(body);
    // create header tab
    controller.nameCode(className + ".h");
    tabs.get(2).setText(header);
    // create example tab
    controller.nameCode(className + "Example.ino");
    tabs.get(3).setText(example);
    // create keywords tab
    controller.nameCode(className + "keywords.txt");
    tabs.get(4).setText(keywords);
}
```



TOP: ClassGeneratorInterface Class
LEFT: Arduino IDE with generated class information tabs

Editor status method, but that only displays the messages at the end of execution. The reason is because graphical user interfaces are multithreaded. To find a solution, I found this source from codeRanch:

[1] https://coderanch.com/wiki/660058/JProgressBar-Doesnt-Update

And this on Worker Threads:

[2] https://docs.oracle.com/javase/tutorial/uiswing/concurrency/worker.html

[3] https://docs.oracle.com/javase/tutorial/displayCode.html?code=https://docs.oracle.com/javase/tutorial/uiswing/examples/concurrency/FlipperProject/src/concurrency/Flipper.java

I simplified the code from sources 1 and 3, but in the end I looked up how displaying is handles by the Arduino IDE (TOP)--> , and I saw how the Editor class is able to display interm results while compiling.

This method is to create an internal class that implements the Runnable interface, with a run method which performs computations and displays prompts, and then create a new Thread which calls the start method.

This approach makes sense, because it creates a thread other than the display thread to perform computations on, which was the cause of my problem.

I created the internal libHandler class, where the run method displays prompts just like at top of page, and the class generator now displays the prompts properly.

```
    // Cannot use invokeLater() here, otherwise it gets
    // placed on the event thread and causes a hang--bad idea all around.
    new Thread(verbose ? verboseHandler : nonVerboseHandler).start();
  }
  class BuildHandler implements Runnable {
    private final boolean verbose;
    private final boolean saveHex;
    public BuildHandler() {
      this(false);
    }
    public BuildHandler(boolean verbose) {
      this(verbose, false);
    }
    public BuildHandler(boolean verbose, boolean saveHex) {
      this.verbose = verbose;
      this.saveHex = saveHex;
    }
    @Override
    public void run() {
      try {
        removeAllLineHighlights();
        sketchController.build(verbose, saveHex);
        statusNotice(tr("Done compiling."));
      } catch (PreferencesMapException e) {
        statusError(I18n.format(
                tr("Error while compiling: missing '{0}' configuration para
                e.getMessage()));
      } catch (Exception e) {
        status.unprogress();
        statusError(e);
      }

      status.unprogress();
      toolbar.deactivateRun();
      avoidMultipleOperations = false;
    }
  }
```

(TOP) Editor Line 1630, Showing how compilation is able to display messages while thread is being compiled.

(Bottom) GenerateLibrary method of ClassGeneratorInterface, showing additions to display interm prompts to the user.

```
public static void generateLibrary(SketchController controller, EditorT
      ArrayList<EditorTab> tabs, EditorStatus status) {
    // create private LibHandler Class so graphics will be updated
    class LibHandler implements Runnable {
      @Override
      /**
       * generates library files and display status messages
       */
      public void run() {
        //compile sketch to check for errors
        status.progressNotice(tr("Compiling sketch..."));
        if (failedToCompile(controller)) {
          return;
        }
        // get the file name and contents of the sketch
        String[] sketchInfo = getNameContentsPath(tab, sketchFile);
        String className = sketchInfo[0];
        String contents = sketchInfo[1];
        String filepath = sketchInfo[2];
        // parse the sketch into format used to generate arduino cl
        status.progressNotice(tr("Creating Library Files..."));
        status.progressUpdate(50);
        status.clearState();
        SketchParser parser = new SketchParser(contents);
        ArduinoClassContainer cont = parser.getContainer(className,
        // create the files with strings
        status.progressNotice(tr("Creating Tabs..."));
        // getExample()
            controller);
        status.progressUpdate(100);
        status.unprogress();
        status.progressNotice(tr(""));
      }
    }
    // class, try line 1632
    new Thread(new LibHandler()).start();
}
```

June 14 2019 4:13 PM: I worked with Daniel to find a useful feature for the class generator. He created a class that set pin modes In the constructor, but they needed to be called in an initialization method at setup, because Arduino does background work before the setup method is called. References:

[1] Problem with using Arduino in constructorhttps://forum.arduino.cc/index.php?topic=243680.0

background of setup and loop methods[2]https://arduino.stackexchange.com/questions/4039/is-setup-and-loop-provided-for-convenience

[3] Use setup method https://stackoverflow.com/questions/18806141/move-object-creation-to-setup-function-of-arduino

[4] Background on Arduino main methodhttp://arduino.land/FAQ/content/2/2/en/can-i-use-int-main-with-arduino.html

[5] The Arduino Main method, files called are in same folder https://github.com/arduino/ArduinoCoreavr/blob/master/cores/arduino/main.cpp

The related methods called in Arduino main method is:
Attach in USBCore
Init in wiring
initVariant line 28
While looking through the Arudino source code, I found a new compiler directive, which may be useful to enforce scope (Consult with Tim).

A simple and efficient approach, you may have seen in other libraries is to add a meth that assigns the pins:

```
class LedControl {
  void attach(uint8_t pin1, uint8_t pin2, uint8_t pin3);
};

void LedControl::attach(uint8_t pin1, uint8_t pin2, uint8_t pin3) {
  this.pin1 = pin1;
  this.pin2 = pin2;
  this.pin3 = pin3;
  // do other setup type operations
  return;
}
```

Now your program, constructs the blank object, and assigns pins during setup():

```
LedControl lc; // not ready to use until attach pins

void setup() {
  lc.attach(11, 13, 12);// data, clock, latch;
  ...
}
```

This involves no temporary object construction, and no assignment operator. With res some people might fairly comment that the user might forget to call attach() and leave object unusable. For a desktop application, you might add some code to prevent that an embedded application, that is a risk you accept which is balanced by the gains in c memory savings.

(TOP) StackOverflow link 2 showing how an initialization method (called attach here) needs to be used inside setup method (BOTOM) Arduino Main method link 5

```
int main(void)
{
  init();
  initVariant();
#if defined(USBCON)
  USBDevice.attach();
#endif
  setup();
  for (;;) {
    loop();
    if (serialEventRun) serialEventRun();
  }
  return 0;
}
```

[5] C Processor directive with #undef
https://users.cs.cf.ac.uk/Dave.Marshall/C/node14.html

With this problem, I need to write the class generator to automatically create a begin method to be called in setup if anything complicated with the Arduino is being done (Consult with Tim).

DOCUMENTATION: I got an email from Professor Salas on how to better document the github page,--> and one useful feature was that I can add gifs in my readMe

Screen Capture to make readme[1]:https://www.laptopmag.com/articles/how-to-video-screen-capture-windows-10

Convert Video to GIF: [2]https://giphy.com/

Add Gifs to github readme: [3]https://medium.com/@josephcardillo/how-to-add-gifs-to-your-github-readme-89c74da2ce47

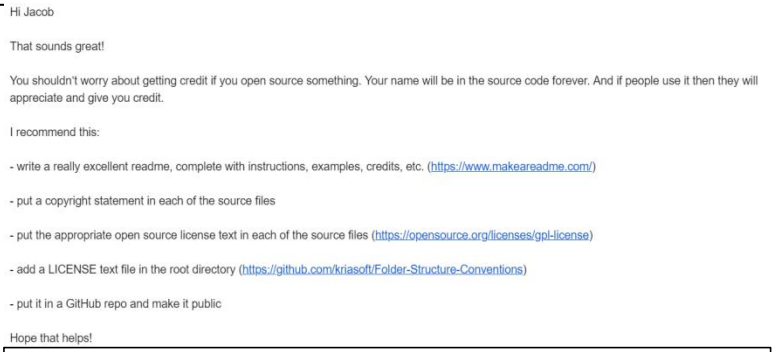
Rights information

[4]https://softwareengineering.stackexchange.com/questions/19649/copyright-notices-disclaimers-in-source-files

June 17 2019 Jacob Smith 4:07 PM: I am using a rasberry pi to run a live viewing server, which can help with publicity for the Arduino Outreach program

These links are to use the rasberry Pi

Getting Started https://projects.raspberrypi.org/en/projects/raspberry-pi-getting-started/5
FAQ https://www.raspberrypi.org/documentation/faqs/#networking



Hi Jacob

That sounds great!

You shouldn't worry about getting credit if you open source something. Your name will be in the source code forever. And if people use it then they will appreciate and give you credit.

I recommend this:

- write a really excellent readme, complete with instructions, examples, credits, etc. (https://www.makeareadme.com/)

- put a copyright statement in each of the source files

- put the appropriate open source license text in each of the source files (https://opensource.org/licenses/gpl-license)

- add a LICENSE text file in the root directory (https://github.com/kriasoft/Folder-Structure-Conventions)

- put it in a GitHub repo and make it public

Hope that helps!

(TOP) Documentation from Professor Salas

Useful commands for internet

Hostname-I to get IP address: https://www.raspberrypi.org/documentation/remote-access/ip-address.md

Ifconfig to get mac address: https://www.raspberrypi-spy.co.uk/2012/06/finding-the-mac-address-of-a-raspberry-pi/

These links show how to connect a webcam and run a server

Overall Tutorial

https://www.instructables.com/id/How-to-Make-Raspberry-Pi-Webcam-Server-and-Stream-/

Webcam usage

https://www.raspberrypi.org/documentation/usage/webcams/

Register rasberry pi brandeis open wifi networkhttps://netreg.brandeis.edu/index.php

Download noobs rasberry pi

https://www.raspberrypi.org/documentation/installation/noobs.md

A related project would be to use the rasberry pi to tweet a picture of the room every week:

Tweeting and taking picture tutorial
https://projects.raspberrypi.org/en/projects/the-all-seeing-pi
Just Tweeting tutorial
https://projects.raspberrypi.org/en/projects/getting-started-with-the-twitter-api/8
Scheduling a task in python:
https://stackoverflow.com/questions/373335/how-do-i-get-a-cron-like-scheduler-in-python
I talked to H about these ideas, and she suggested a doorbell feature. This means that by the end of this project, there will be monitors displaying live feeds of the makerlab rooms with some additional displays in the lobby, and the user can go to a website to view the streams or ring a doorbell in person.

This could then be connected to existing PIR sensors in used rooms to display when there is a person there (Daniel Lays idea).

While setting up the rasberry pi, I got an input/output error, which showed me that the Rasberry Pi uses its sd card as its storage so it must have an sd card in at all times.
https://www.raspberrypi.org/forums/viewtopic.php?t=26715

6:21 PM: The secret to download noobs on previous page is to copy contents of noobs folder to sd card.

6/18/2019 11:18 Am Jacob Smith: I set time on Rasberry Pi using startup menu, and went to Brandeis ITS help desk to handle the MAC address registration on Brandeis Open (see page 102 of Zumo notebook)

4:49 PM: I spent today looking through functions on the Rasberry Pi.
I can now tweet, read button presses, and take pictures from the rasberry Pi.

Some Errors I got along the way where

A folder not found error when the webcam was not plugged in
https://forum.arduino.cc/index.php?topic=424233.0
A parsing error with Python because methods need to be declared before they are called
https://stackoverflow.com/questions/3754240/declare-function-at-end-of-file-in-python/3754284
A server rejected error when remotely accessing IP address, Brandeis ITS gave me a secure file transfer protocol datasheet.
https://www.ssh.com/ssh/putty/putty-manuals/0.68/Chapter10.html#errors-connrefused
Python blocks need one element in bodyhttps://stackoverflow.com/questions/43189302/syntaxerror-unexpected-eof-while-parsing?rq=1
Defining a python function
https://www.tutorialspoint.com/python/python_functions.htm
https://www.codementor.io/kaushikpal/user-defined-functions-in-python-8s7wyc8k2

```python
#Jacob Smith Brandeis University Makerlab 6/18/2019
#A library of functions useful in interacting with a RasberryPi
#Along with a main method to send a tweet when a button is pressed

#sends a tweet with a picture when a button ispressed
def main():
    import random
    #messages to selectfrom when tweeting
    messages=["HiFolks! Come check out the makerlab!","Look at what
    hashtag="\nMeet us on first floor of Farber in the library\n#Br

    #when the button is pressed,a randomly selected message will be
    twitter=importKeys()
    while(True):
        if isButtonPressed():
            print("Taking Picture for Tweet")
            #generate the message for the tweet
            message=random.choice(messages)+hashtag
            tweet(twitter,message)
            print("Tweeted:%s"%message)
    return


#imports a twitter account's keys from auth.py file
#see tweetfunction for link
def importKeys():
    from twython import Twython

    from auth import(
    consumer_key,
    consumer_secret,
    access_token,
    access_token_secret
    )

    twitter=Twython(
    consumer_key,
    consumer_secret,
    access_token,
    access_token_secret
    )
    return twitter

#takes a picture and tweets it with a given message
#needs the twitter object and the message
#https://www.raspberrypi.org/documentation/usage/webcams/
#https://projects.raspberrypi.org/en/projects/getting-started-with-the-
def tweet(twitter,message):
    import subprocess
    subprocess.call(["fswebcam","/home/pi/toSend.jpg"]);
    image=open('/home/pi/toSend.jpg','rb')
    response=twitter.upload_media(media=image)
    media_id=[response['media_id']]
    twitter.update_status(status=message,media_ids=media_id)
    return

#returns true if a button wired to ports 18 and ground is pressed
#http://razzpisampler.oreilly.com/ch07.html
def isButtonPressed():
    import RPi.GPIO as GPIO
    import time

    GPIO.setmode(GPIO.BCM)

    GPIO.setup(18,GPIO.IN,pull_up_down=GPIO.PUD_UP)

    input_state=GPIO.input(18)
    #delay for 200 milliseconds so multiple button presses aren't detec
    time.sleep(.2)
    return not input_state

#actually call the main method to run the program
main()
```

Jacob Smith 6//21/2019
(Backlog of documentation)

<--This is the Twitter.py python program I wrote to: load my twitter account information, and tweet a newly taken picture when a button is pressed.

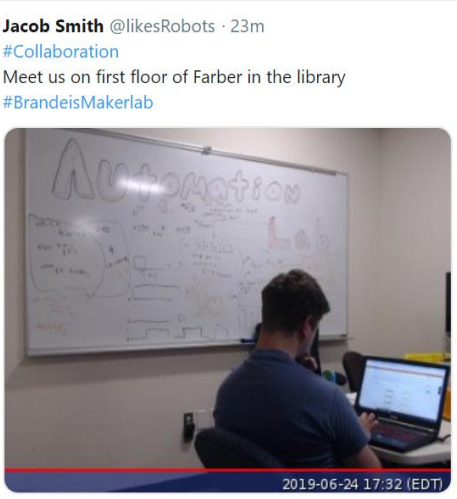
The tweet also includes a randomly selected message such as #collaboration.

The programs are modified from existing tutorials.

In this version, I am importing code in a method instead of at the top as I would in java, which is bad practice.
Error messages I encountered are presented here and on the next page

Tweet Generated fromTwitter.py Python program

While loop in python https://www.tutorialspoint.com/python/python_while_loop.htm



Run a bash script in python
https://unix.stackexchange.com/questions/190495/how-to-execute-a-bash-command-in-a-python-script
File Paths for imaging
https://stackoverflow.com/questions/11727598/pil-image-open-working-for-some-images-but-not-others

Sleeping in Python

https://stackoverflow.com/questions/510348/how-can-i-make-a-time-delay-in-python

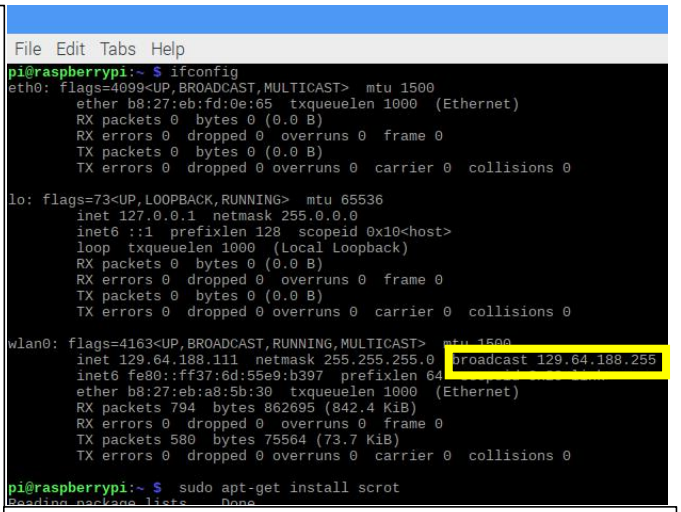
Randomly Select element from a list python

https://stackoverflow.com/questions/306400/how-to-randomly-select-an-item-from-a-list

Python whitespace error
https://stackoverflow.com/questions/26720841/python-indentationerror-unindent-does-not-match-any-outer-indentation-level/34100174

Jacob Smith 10:20 AM 6/19/2019: I am continuing on the Rasberry Pi Webserver and tweeting project, I connected to the ip address now, it was the broadcast ip address, I was using the inet ip address-->

https://www.raspberrypi.org/forums/viewtopic.php?t=172891



Result of ifConfig command in rasberry Pi, with correct IP adress for remote access highlighted

6/21/2019 1:02 PM Jacob Smith
I got fswebcam to display the server, using Tims help with step 6. That step involved using the Rasberry Pi Terminal to find the configuration file and editing it.
I needed to learn more about Linux commandshttp://linuxcommand.org/lc3_lts0020.php
Another command that was necessary was how to stop the motion stream:https://ubuntuforums.org/showthread.php?t=2145787

Output of fswebcam server

Unfortunately, the video is about one frame per second, which is the same as this post : https://raspberrypi.stackexchange.com/questions/29543/motion-filming-cant-get-on-real-speed, which recommends MJPEG

I tried using MJPEG
https://blog.miguelgrinberg.com/post/how-to-build-and-run-mjpg-streamer-on-the-raspberry-pi
MJPG Streamer download:
https://github.com/jacksonliam/mjpg-streamer
This results in a address already in use message, which seems that the other webserver I started isnt closing

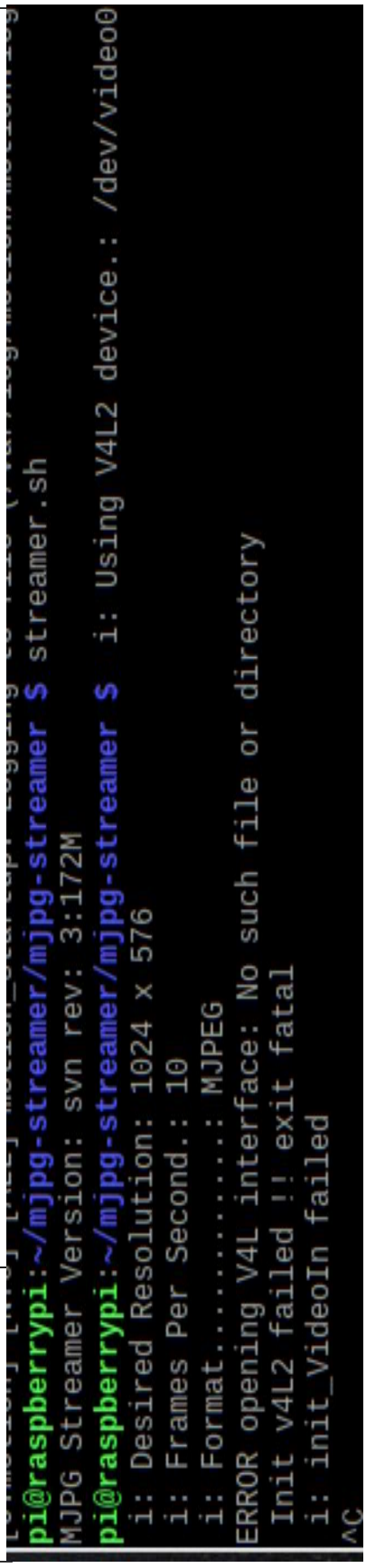
In addition, this forum claims to have a high enough frame rate to control a robot from the feed:
https://www.raspberrypi.org/forums/viewtopic.php?p=751735

The tutorial resulted in a error opening V4L interface error, **See Right-->**
https://raspberrypi.stackexchange.com/questions/77768/error-opening-v4l-interface-no-such-file-or-directory

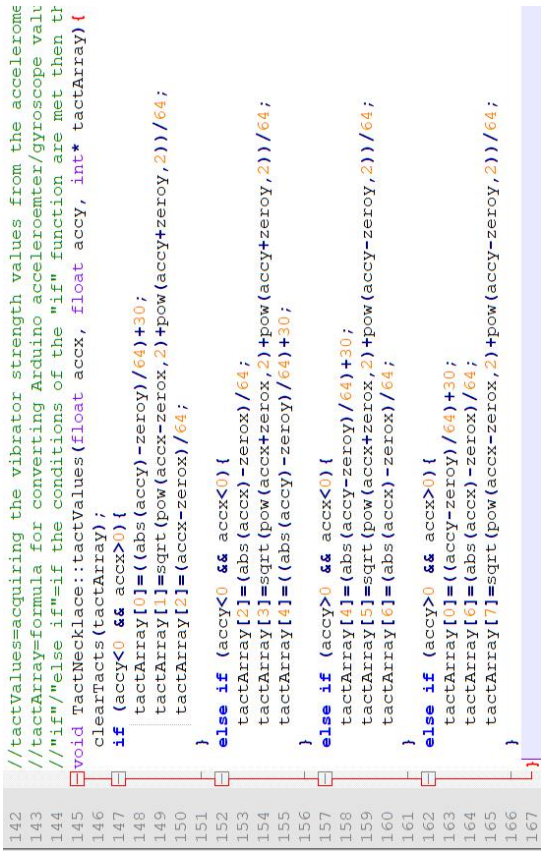
However, this stream is also very slow
I looked for other streaming platforms than motion, and tried this tutorial:
https://chriscarey.com/blog/2017/04/30/achieving-high-frame-rate-with-a-raspberry-pi-camera-system/
But it resulted in a failed to create component vc.ril.camera message

Below ↓ is an example of the feed not working

LEFT: Error with one of web server commands shows blank picture
RIGHT: Error when using streamer.sh



6/24/2019 Jacob Smith: also today and last week, I helped Aiden and Johnny, two rising college freshman engineering students who wanted to turn their large Arduino sketch into an Arduino library.
This has implications for my class generator project, as I want to show that it can handle different coding styles.
Tactile Necklace Project: https://github.com/AidenKunkler-Peck/Tactile-Necklace.
The sketch at right shows their condition checking to start, and the below image shows how I got them to sue methods and control flow to clean it up right. **RIGHT, high school tactarray method, BOTTOM same method after refactoring**



```
//tactValues=acquiring the vibrator strength values from the accelerome
//tactArray=formula for converting Arduino accelerometer/gyroscope valu
//"if"/"else if"=if the conditions of the "if" function are met then th
void TactNecklace::tactValues(float accx, float accy, int* tactArray){
  clearTacts(tactArray);
  if (accy<0 && accx>0){
    tactArray[0]=((abs(accy)-zeroy)/64)+30;
    tactArray[1]=sqrt(pow(accx-zerox,2)+pow(accy+zeroy,2))/64;
    tactArray[2]=(accx-zerox)/64;
  }
  else if (accy<0 && accx<0){
    tactArray[2]=(abs(accx)-zerox)/64;
    tactArray[3]=sqrt(pow(accx+zerox,2)+pow(accy+zeroy,2))/64;
    tactArray[4]=((abs(accy)-zeroy)/64)+30;
  }
  else if (accy>0 && accx<0){
    tactArray[4]=(abs(accy-zeroy)/64)+30;
    tactArray[5]=sqrt(pow(accx+zerox,2)+pow(accy-zeroy,2))/64;
    tactArray[6]=(abs(accx)-zerox)/64;
  }
  else if (accy>0 && accx>0){
    tactArray[0]=((accy-zeroy)/64)+30;
    tactArray[6]=(abs(accx)-zerox)/64;
    tactArray[7]=sqrt(pow(accx-zerox,2)+pow(accy-zeroy,2))/64;
  }
}
```

```
void tactValues(float accx, float accy, int* tactAr
  if (accy<0 && accx>0){
     tactArray[0]=((abs(accy)-zeroy)/64)+30;
     tactArray[1]=sqrt(pow(accx-zerox,2)+pow(accy+z
     tactArray[2]=(accx-zerox)/64;
     tactArray[3]=0;
     tactArray[4]=0;
     tactArray[5]=0;
     tactArray[6]=0;
     tactArray[7]=0;
  }
  else if (accy<0 && accx<0){
    tactArray[0]=0;
    tactArray[1]=0;
    tactArray[2]=(abs(accx)-zerox)/64;
    tactArray[3]=sqrt(pow(accx+zerox,2)+pow(accy+ze
    tactArray[4]=((abs(accy)-zeroy)/64)+30;
    tactArray[5]=0;
    tactArray[6]=0;
    tactArray[7]=0;
  }
  else if (accy>0 && accx<0){
    tactArray[0]=0;
    tactArray[1]=0;
    tactArray[2]=0;
    tactArray[3]=0;
    tactArray[4]=(abs(accy-zeroy)/64)+30;
    tactArray[5]=sqrt(pow(accx+zerox,2)+pow(accy-z
    tactArray[6]=(abs(accx)-zerox)/64;
    tactArray[7]=0;
  }
  else if (accy>0 && accx>0){
    tactArray[0]=((accy-zeroy)/64)+30;
    tactArray[1]=0;
    tactArray[2]=0;
    tactArray[3]=0;
    tactArray[4]=0;
    tactArray[5]=0;
    tactArray[6]=(abs(accx)-zerox)/64;
    tactArray[6]=(abs(accx)-zerox)/64;
    tactArray[7]=sqrt(pow(accx-zerox,2)+pow(accy-zer
  }
  else{
    Serial.println("uh oh! something failed!");
    tactArray[0]=0;
    tactArray[1]=0;
    tactArray[2]=0;
    tactArray[3]=0;
    tactArray[4]=0;
    tactArray[5]=0;
    tactArray[6]=0;
    tactArray[7]=0;
  }
  //return(tactArray[0,1,2,3,4,5,6,7]);
}

```

Also, the process of writing a library and documenting it made the students think more critically about their program, which I created this repository to hold example sketches for:https://github.com/jsmith2021Brandeis/ArduinoEducation

The skills I taught them would serve as a good second part of my Arduino outreach program. The topic we covered where:

- Using github desktop
- Creating a github repository with a readme
- Using loops to avoid repition and methods to keep code clear
- Creating a library out of a sketch
- Main ideas of object oriented programming, along with what in the new library to make private vs public.
- Class constructors

The areas we spent the most time on where for loops, the mindset of using private class methods to increase usability, and debugging the library.

While Aiden and Johnny where quick studies, they already had a summer of programming experience, and I need to make sure my tutorials work for someone who is completely new to programming.
Today, I worked with Aiden to make his project work for any number of pins and any board. This lead to working with arrays in Arduino.
Arduino Education Class-->

```
#ifndef ArduinoEducation_h
#define ArduinoEducation_h

//this should work on all boards, so there is no

#include <Arduino.h>
//create macro to get length of array
#define ARRAY_SIZE(arr)((sizeof(arr))/sizeof(arr

class ArduinoEducation{

    private:
        //an array containing pin numbers for an
        int* pins;
        //the length of the array
        int numPins;
    public:
        //creates a new ArduinoEducation object
        ArduinoEducation();
        //gives an array to the ArduinoEducation
        void ArduinoEducation::begin (int pins []

        //prints all the pins in the array to pr
        void ArduinoEducation::printPins();

};

#endif
```

(TOP) ArduinoEducation Header File (see next page)

(BOTTOM) ARDUINO EDUCATION EXAMPLE FILE

Shows how arrays can be used with classes in Arduino as cleanly as possible, array size macro highlighted

ArrayPrintingExample §

```
user to input an array without having to speci
June 24 2019
Sources:
https://forum.arduino.cc/index.php?topic=50812
https://blog.feabhas.com/2013/11/shock-horror-
*/

#include <ArduinoEducation.h>
ArduinoEducation test;
void setup() {
  Serial.begin(9600);
  int pins []={5,6,7,8,9,11,12};
  //pass the pins to the test program, and als
  test.begin(pins,ARRAY_SIZE(pins));

}

void loop() {
  test.printPins();
```

Jacob Smith 12:45 AM: I have been reading more about arrays in Arduino, and I've concluded that the closest I can get to how they work in java is to define an array size macro and pass it to the class.

I accomplish this with the **ArduinoEducation class, shown on this and previous page-->**

The nice part is the macro can be defined in the header file, away from the User in the sketch file, see previous page.
The reason I can't find the array size inside the class is because of array decay, as arrays turn into pointers when they are used by a function in C.

The only way to avoid the decay seems to require knowing the length in the first place, so it is not preferable:

Sources:

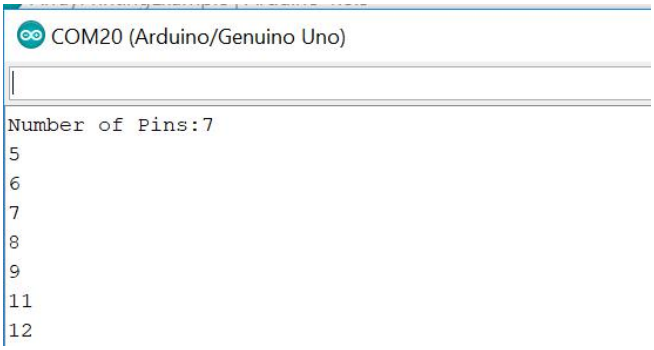
Array class Variable:[0]https://forum.arduino.cc/index.php?topic=508126.0

Background on Arrays in C [1]https://blog.feabhas.com/2013/11/shock-horror-i-learned-something-about-arrays-in-c/

FUTURE WORK: Tomorrow I will show Johnny this way of handling arrays to get dynamic length and high usability. In addition, I will show him how to display a warning if a board other than an uno or nano is used, saying that the necklace has not been tested with that board.

This will allow advanced users to ignore the warning, but still let new users follow the tutorial (Tim's advice).

After that, they just need a tutorial



(TOP) ArduinoEducation Example File Output

(BOTTOM) ARDUINO EDUCATION Body File

Shows how arrays can be used with classes in Arduino as cleanly as possible

```
//this should work on all boards, so there is no

//include all of the classes necessary to make th
#include <Arduino.h>
#include <Wire.h>
#include <ArduinoEducation.h>

//ArduinoEducation constructor
ArduinoEducation::ArduinoEducation(){
}

//gives an array to the ArduinoEducation class ar
void ArduinoEducation::begin (int pins[], int nur
    if(!Serial){
        //Serial.begin(9600);
        while(!Serial){}
    }
    //this->numPins=sizeof(pins)/sizeof(pins[0]);
    this->numPins=numPins;
    this->pins = pins;
    Serial.print("Number of Pins:");
    Serial.println(numPins);

}

//prints all the pins in the array to prove it wa
void ArduinoEducation::printPins(){
    for(int i=0;i<numPins;i++){
        //Serial.println(*(pins+i));
        Serial.println(pins[i]);
    }
    delay(5000);

}
```

Jacob Smith 6/25/2019 5:41 PM: I think the previous high frame rate tutorial is for rasberry pi camera

2012/5/7 post shows camera needs yuyv format:https://sourceforge.net/p/mjpg-streamer/discussion/739917/thread/dfc46d26/

6/26/2019 12:02 PM Jacob Smith: yesterday, I got the web stream to work at 60 frames per second!
First, I followed this tutorial to get the stream to work at all https://blog.cudmore.io/post/2015/03/15/Installing-mjpg-streamer-on-a-raspberry-pi/
Then, I used this:
2010/4/8 :https://sourceforge.net/p/mjpg-streamer/discussion/739917/thread/541a9b8c/ to change the frame rate from 6 fps to 60 fps.
The stream and command line results are shown right and bottom (note highlight box showing framerate)-->
The next goal is to allow streaming with audio and video, and YUYV can’t do that.
The Logitech c290 camera I am using supports the h264 video format, which includes audio
https://raspberrypi.stackexchange.com/questions/4412/streaming-h264-with-logitech-c920

```
pi@raspberrypi:/home $ ./stream.sh
MJPG Streamer Version: svn rev: 3:182
 DBG(input_uvc.c, input_init(), 136): argv[0]=UVC webcam grabber
 DBG(input_uvc.c, input_init(), 136): argv[1]=-f
 DBG(input_uvc.c, input_init(), 136): argv[2]=10
 DBG(input_uvc.c, input_init(), 136): argv[3]=-r
 DBG(input_uvc.c, input_init(), 136): argv[4]=320x240
 DBG(input_uvc.c, input_init(), 136): argv[5]=-d
 DBG(input_uvc.c, input_init(), 136): argv[6]=/dev/video1
 DBG(input_uvc.c, input_init(), 136): argv[7]=-y
 DBG(input_uvc.c, input_init(), 223): case 6,7
 DBG(input_uvc.c, input_init(), 201): case 4,5
 DBG(input_uvc.c, input_init(), 194): case 2,3
 DBG(input_uvc.c, input_init(), 231): case 8,9
 DBG(input_uvc.c, input_init(), 302): input id: 0
 i: Using V4L2 device.: /dev/video1
 i: Desired Resolution: 320 x 240
 i: Frames Per Second.: 10
 i: Format............: YUYV
 i: JPEG Quality......: 80
 i: TV-Norm...........: DEFAULT
 DBG(input_uvc.c, input_init(), 355): vdIn pn: 0
Unable to set format: 1448695129 res: 320x240
 Init v4L2 failed !! exit fatal
 i: init VideoIn failed
```

(TOP)yuyv format spec (LEFT)working mjpg stream at 60 fps 129.64.188.111:8080/stream_simple.htm (BOTTOM)message showing success

```
pi@raspberrypi:/home $ /usr/local/bin/mjpg_streamer -i "/usr/lo
MJPG Streamer Version: svn rev: 3:182
 i: Using V4L2 device.: /dev/video0
 i: Desired Resolution: 640 x 480
 i: Frames Per Second.: 60
 i: Format............: YUYV
 i: JPEG Quality......: 80
Adding control for Pan (relative)
UVCIOC_CTRL_ADD - Error: Inappropriate ioctl for device
Adding control for Tilt (relative)
UVCIOC_CTRL_ADD - Error: Inappropriate ioctl for device
Adding control for Pan Reset
UVCIOC_CTRL_ADD - Error: Inappropriate ioctl for device
Adding control for Tilt Reset
UVCIOC_CTRL_ADD - Error: Inappropriate ioctl for device
Adding control for Pan/tilt Reset
UVCIOC_CTRL_ADD - Error: Inappropriate ioctl for device
Adding control for Focus (absolute)
UVCIOC_CTRL_ADD - Error: Inappropriate ioctl for device
mapping control for Pan (relative)
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for Tilt (relative)
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for Pan Reset
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for Tilt Reset
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for Pan/tilt Reset
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for Focus (absolute)
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for LED1 Mode
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for LED1 Frequency
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for Disable video processing
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
mapping control for Raw bits per pixel
UVCIOC_CTRL_MAP - Error: Inappropriate ioctl for device
 o: www-folder-path...: /usr/local/www/
 o: HTTP TCP port.....: 8080
 o: username:password.: disabled
 o: commands..........: enabled
```

Jacob Smith 6/27/2019 8:13 PM
TACTOR PROJECT CODE
CLEANUP:Yesterday, I helped Johnny remove incorrect for loop conditions (which explains why the tactors where not working with incorrect array size two days ago).
Today, I helped Johnny print out detailed sensor data and write error messages in his code, while Tim gave us the idea to explicitly allocate array memory. This is the first time I've gotten an array in a class to work without the size of the array being specified in the class.
See modification of Arduino Education Sketch at left-->
Also, Tim's help got us to see that a for loop in SendVibration was causing a memory error, which was stopping the program. Whenever a program stops running, the main reasons are power and memory (two issues not encountered in a computer science course).
TACTOR LAYOUTS: In addition, Johnny's code now works for either 4 pins on an uno, or 8 on a nano (the uno necklace is simpler and will be used for the workshop).
FUTURE WORK:Tomorrow, I will have Johnny make a dozuki guide for the uno setup.

Jacob Smith 7/3/2019 12:46 PM:
PISERVER:This week, I have been helping Aiden set up the Rasberry Pi Cluster Computer. He found and registered the MAC dresser of the raspberry Pis and formatted their sd cards according to the specified node.
This is where the rasberry Pis will be mounted and connected to the internet with ethernet-->

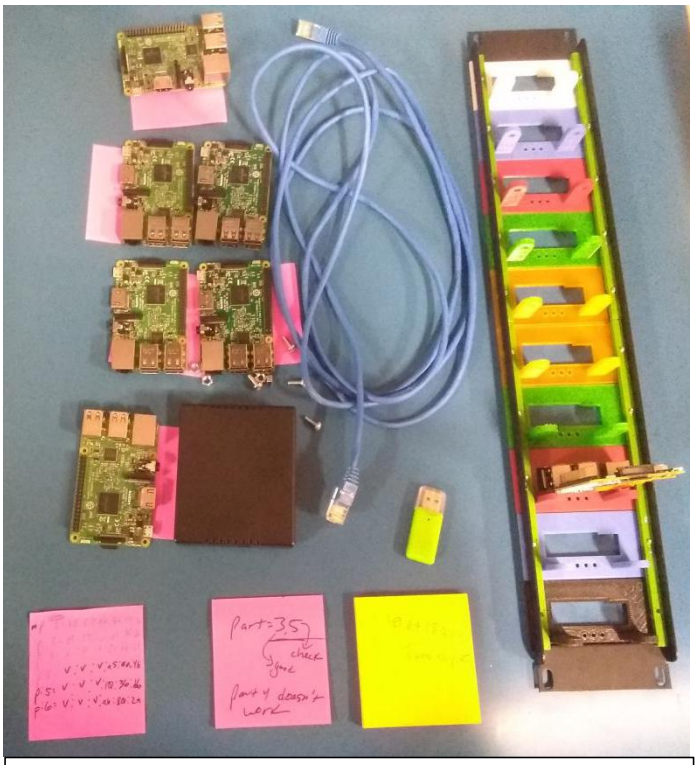
```
16  //create a list of pin numbers
17  //the arraysize has to be here, don't know why
18  int pins [7]={5,6,7,8,9,11,12};
19  //pass the pins to the test program, and also
20  test.begin(pins,ARRAY_SIZE(pins));
```

ArduinoEducation Example Sketch showing explicit array size. (replaces line 16 of bottom code 3 pages previous)

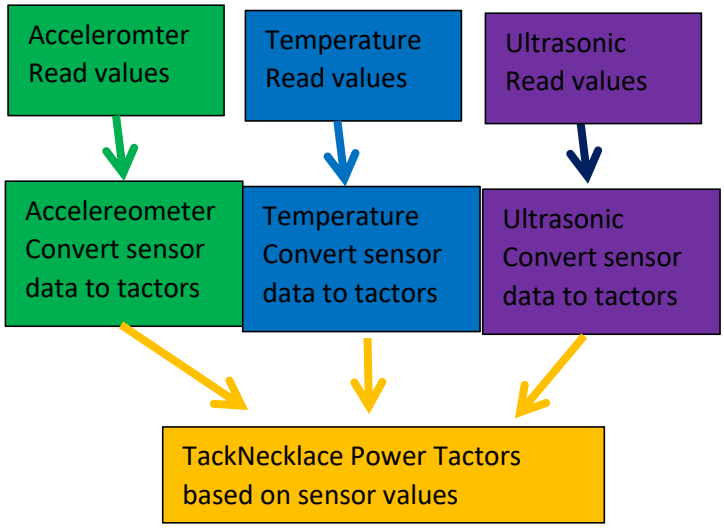
Empty Server Stack which will house the rasberry Pis.

This is the equipment that Aiden is using for the project, note the **colored rasberry pi mounting material at right -->**
Daniel and I inspected the server stack today, and if we want to use the Dell Poweredge Rackmount Console, we would need to connect it to a computer that uses analog display. If we want to only use rasberry Pi computers, we could either use a converter https://www.freetv.ie/vga-to-hdmi/ or replace the display with an hdmi display.

TACKNECKLACE: Also this week, I worked with Johnny to make the Tactile Necklace Program more modular so it can work with multiple sensors. Currently, the TackNecklace class just has all the sensor data, but to really follow good practice, I will have Johnny create classes for each of the sensors to be used, designed in a specified format for interfacing with the vibrators.
Class Diagram for new TackNecklace-->
It's hard to explain to the students why a multi-class approach is better than putting sensors and motors into one large class, but the reasons are organization, modularity, and reducing the number of lines of programs.
Also, the level of code clarity required for this project is higher than usual, both because these programs will be used by novice programmers and because they all need to send an output to the TackNecklace class. For true encapsulation, I would want the sensor classes to return a magnitude and direction of sensed data, which is closer to a universal format.



Rasberry Pis and Mounting for Rasberry Pi Linux



TackNecklace ClassFlow Diagram showing enhanced modularity.

DIGITAL WINDOW: While demonstrating the digital window tweeting to precollege students, I saw that the digital window wasn't able to take pictures because the camera was busy running the stream. Solutions for this would include automatically pausing the stream to take a tweet or making the tweet be a screenshot of the stream. I prefer the latter approach because it would allow the user to see themselves while taking the picture.
ARDUINO OUTREACH: Now that I finished my trainings and am officially hired by the makerlab, I'm getting ready to contact laboratories to gauge interest for the electronic course. I drafted an introductory email which was reviewed by Tim and will be reviewed by Ian next week. FUTURE WORK: That gives me this week to lay out the introductory materials of the course and prepare a publicity campaign.
analogMeter==makes a nice dial
arlCDTest=has touch screen
button=lets you press button
checkbox=has a checkbox
choice=displays a menu
circle=displays a nice drawing
cls= stes background color
colorid=shows three example colors, don't understand
dial=shows a touch dial
digitalMeter=shows a box withtext field
files=shows file reading, don't understant
fonts=shows fonts available
gauge=shows a bar that can be filled
linetype=shows lines for drawing
picture=allows screen to display a picture
pie= shows lcd's ability to display pie chart in different line styles
plot=displays a scatterplot
print example=shows lcd's ability to print
NOtes: interuppot options are avilable, like radio_interrupt
rect=draws rectangles
SerialMonitorTest=shows lcd's ability to send to Serial Monitor
simple=shows ability to display well formatted text
slider=shows ability to display a slider
static2=interactive display
tictactoe=plays a tic tac toe game with lcd
touch=shows feedback of screen, what coordinates user touched
touchzone=displays a keypad
towers1= displays the towers of hanoi
wstate=shows ability of button to be enabled or disabled
JUne 14 2019 begin featurehttps://github.com/earthmake/arLCD/issues/2
Working on class editor, setup and loop full of blank, blank lines
Batch file now uses variables for file paths and works, working on code reiew arduino pull request, build path remove multiple paramters, line 42 Arduino class master can be deleted why not array?

https://bugs.eclipse.org/bugs/show_bug.cgi?id=168208 descritpion windows
Connecting file paths
https://unix.stackexchange.com/questions/23208/building-paths-robustly
Advice of best way to handle long parameter lists
https://stackoverflow.com/questions/439574/whats-the-best-way-to-refactor-a-method-that-has-too-many-6-parameters