

Arduino Education and Development: Summer 2019

Jacob Smith¹, Tim Hebert¹, Vivek Vimal
¹Brandeis University



Brandeis
UNIVERSITY

Description

Over the summer as an Impact maker with the Brandeis Department of Research, Technology and Innovation, I had the opportunity to develop programs in the areas of electronics education and communication. Together, these programs combine to create a comprehensive electronics outreach platform for Brandeis. Individual project summaries.

Arduino Class Generator: A tool integrated to the Arduino IDE to create class body, header, keywords, and example sketch files. This is the only such application I know of. It allows advanced Arduino users to quickly create object oriented classes from programs and allows new users to learn what goes into a c++ class. I have used it for my own programs, to teach a high school student on the tactile necklace (see below), and to create a program for a neuroscience laboratory.

Arduino_Class_Maker Arduino Education: A set of workshops, tutorials, and background programs to get students started with Arduino. I am currently teaching this workshop at Brandeis, where it is the only way for students of all backgrounds to learn this content. This workshop will also be the foundation of a high school class at Waltham Public High School this year.

Tactile Necklace: I advised two students on a tactile sensor necklace with neuroscience research applications. The project allowed me to teach advanced programming concepts to college students and helped them simply and document their programs. I also helped a neuroscience laboratory develop a library for their tacto.

Raspberry Pi Digital Window Project Programs and setup for a series of digital windows that will allow the makerlab to communicate internally with itself. This system could also be extended to connect with other makerlabs.

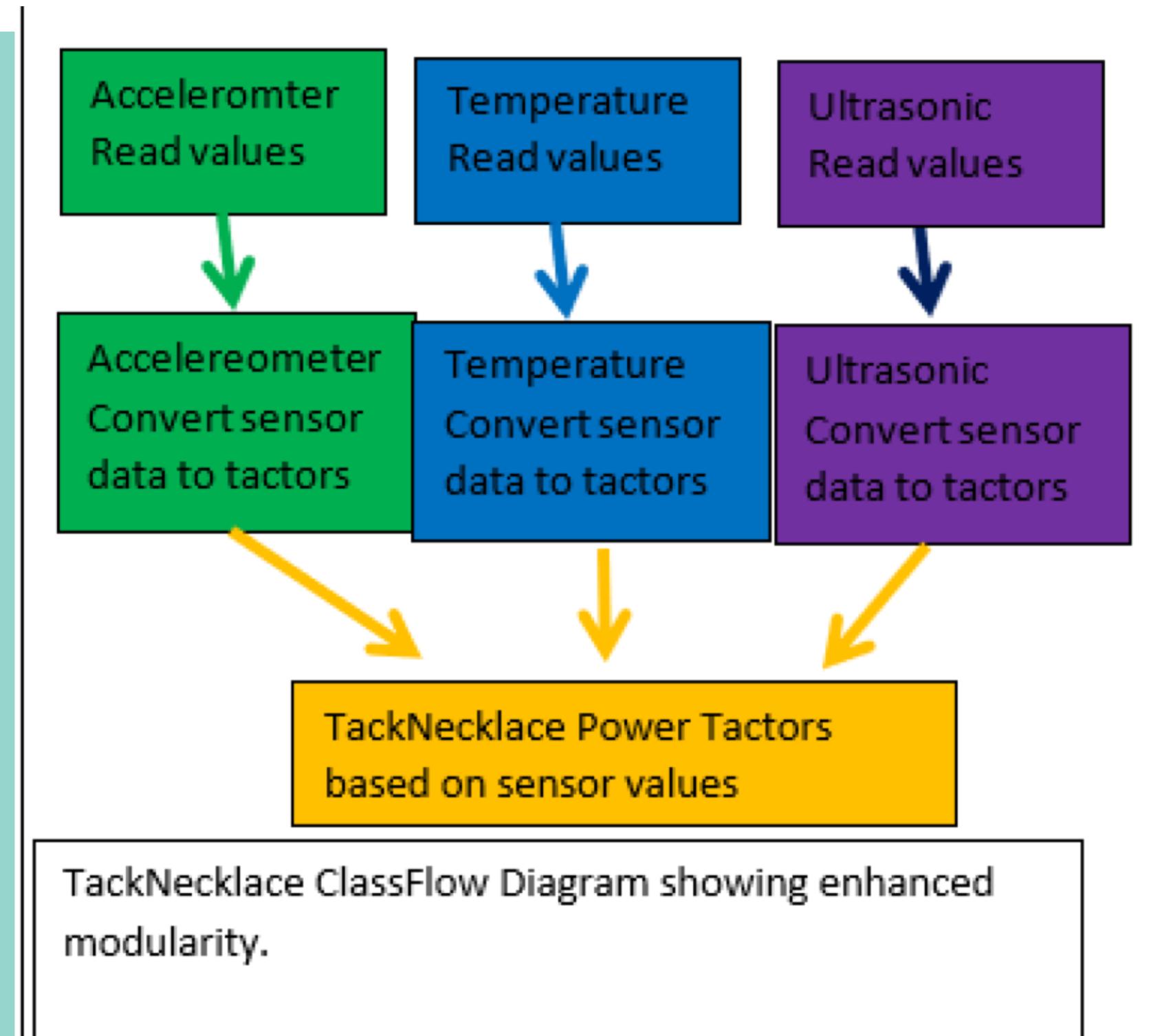
Design process

While I was writing Arduino Programs in the Brandeis Robotics Club, I found myself tediously organizing my programs so they could be presented properly and simply. The existing options were to write sloppy code that I couldn't teach new students in the robotics club with or to create an application to automate Arduino library creation, which became my Arduino Class Maker.

I turned my interest in technical communication and education to the other projects over the summer, and they developed with the support of my advisors.

My main technical challenge was to convert the sparsely documented electronics forums into a programming system that actually explains errors and automatically performs operations that I had to learn after a year of programming.

The education challenge was to break down a topic I was familiar with and make it accessible to students of any background. I made a presentation that explains the concepts with straightforward analogies, and had to practice the workshop multiple times, receiving feedback from my advisors. Fortunately, I know have many whiteboards of student's positive feedback, and I was even able to teach advanced concepts such as class modularity and encapsulation to the tactile necklace students



```

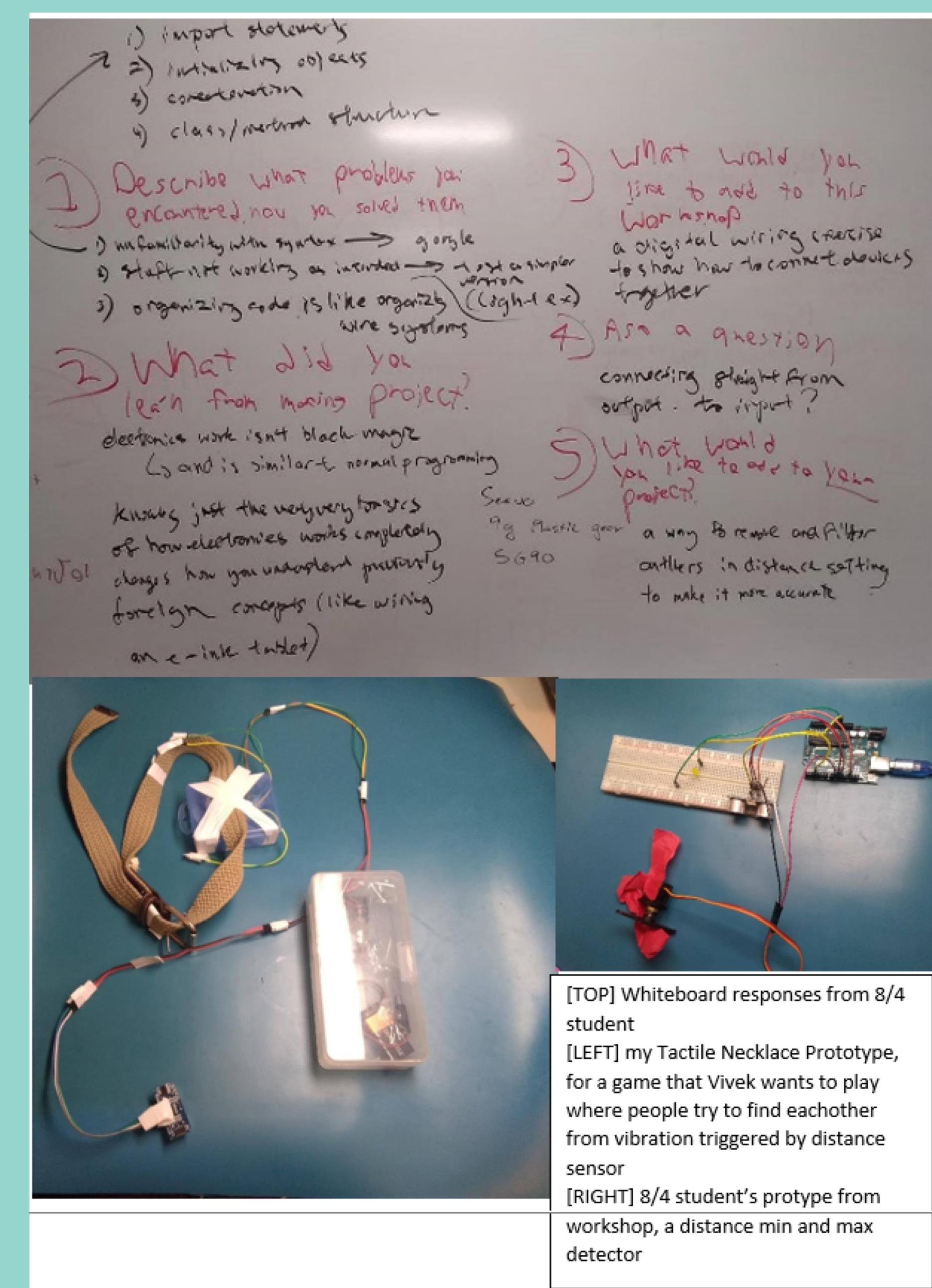
File Edit Sketch Tools Help
Morse Morse.cpp § Morse.h § MorseExample § MorseKeywords.txt §

void Morse::dash() {
    digitalWrite(pin, HIGH);
    delay(1000);
    digitalWrite(pin, LOW);
    delay(250);
}

#endif

```

Global variables use 9 bytes (0%) of dynamic memory, leaving 20



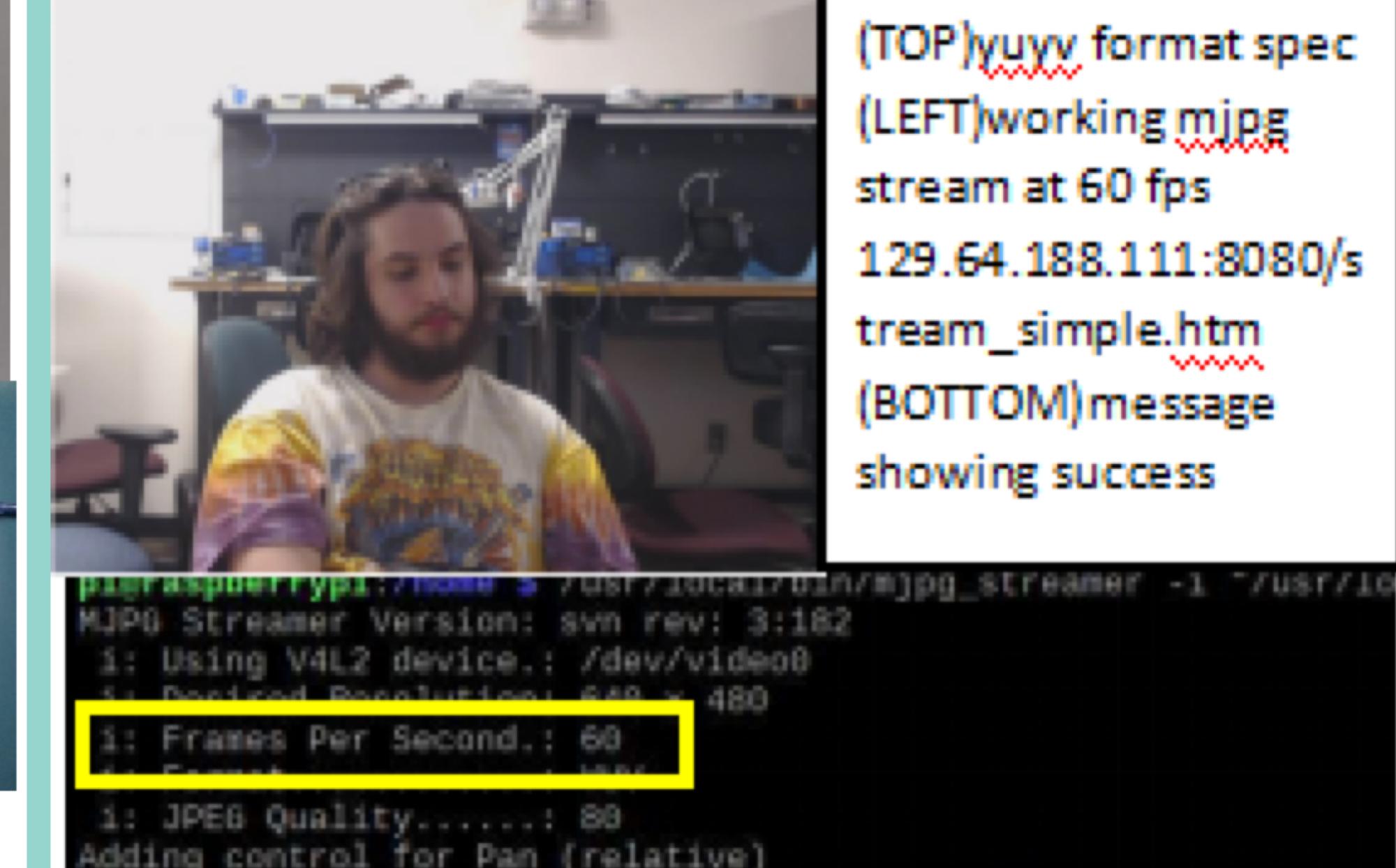
Results

I hope to make the class generator more rigorous to other students programming styles and integrate it into a larger Arduino application. This semester, I will develop an advanced electronics workshop for Brandeis community member who understand the background concepts but still need to learn about Arduino components. I also plan on allowing the digital window system to stream audio, it is video only now. Finally, the tactile Necklace Project will be modified to allow the necklace to work simply with a suite of sensors.

```

pi@raspberrypi:~/home $ ./stream.sh
MJPG Streamer Version: svn rev: 3:182
DBGInput_uvc.c, input_init(), 136: argv[0]=UVC webcam grabber
DBGInput_uvc.c, input_init(), 136: argv[1]=-f
DBGInput_uvc.c, input_init(), 136: argv[2]=10
DBGInput_uvc.c, input_init(), 136: argv[3]=-r
DBGInput_uvc.c, input_init(), 136: argv[4]=320x240
DBGInput_uvc.c, input_init(), 136: argv[5]=-d
DBGInput_uvc.c, input_init(), 136: argv[6]=-dev/video1
DBGInput_uvc.c, input_init(), 136: argv[7]=-y
DBGInput_uvc.c, input_init(), 223: case 6,7
DBGInput_uvc.c, input_init(), 201: case 4,5
DBGInput_uvc.c, input_init(), 194: case 2,3
DBGInput_uvc.c, input_init(), 231: case 8,9
DBGInput_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
: format: YUYV
i: JPEG Quality.....: 80
i: TV-Norm.....: DEFAULT
DBGInput_uvc.c, input_init(), 355: vdIn pn: 0
Unable to set format: 1448695129 res: 320x240
Init v4l2 failed !! exit fatal
i: init_VideoIn failed

```



Links

https://github.com/BrandeisMakerLab/Arduino_Class_Maker
https://github.com/BrandeisMakerLab/Arduino_Education
https://github.com/BrandeisMakerLab/Arduino_Tactile_Necklace
https://github.com/BrandeisMakerLab/Raspi_Digital_Window_Proj