**Idea**: “Smart” Room

**Description**: Create software for controlling various gadgets in a room (i.e. lights, speakers, deadbolt lock, etc.). Use a Python back-end and HTML front-end. HTML for GUI sends information to Python back-end which interfaces with the gadgets.

**Reasoning**: Because the high schoolers are inexperienced it will be difficult to design a project that can span the entirety of two weeks. However, using a beginner-level language such as Python can enable them to learn and write practical code in a short amount of time. A simple HTML GUI on the front-end will also be within the scope of the project. The nature of this design also allows for a large amount of stretch goals to be added depending on the amount of progress made over the two week period. For example, we could begin with a goal of simple turn a Wi-Fi light bulb on and off from a webpage. If this proves too trivial, we could add things such as change the brightness or color settings of the light. Creating a time setting for when the light turns on automatically, or monitoring how many hours a day the light is on (possibly estimating energy consumption) can also be added. Assuming all of this is achieved, then additional gadgets could be added on such as a Bluetooth speaker for music or an alarm sound, or even a small motor that can unlock a deadbolt.

This project essentially consists of a dozen smaller projects that should (hopefully) be within the scope of the high schoolers.

https://github.com/schollz/find3/blob/master/doc/doc.md

**Initial Goal**: Turn a light on and off through a webpage interface.

**Stretch Goals**:

* Change color and brightness settings of light
* Automate control of light (based on time of day)
* Monitor duration the light is on
  + Estimate energy consumption
  + Make suggestions to reduce energy consumption
* Set up wake-up alarm with Bluetooth speaker
* Play music through Bluetooth speaker
* Unlock/lock a deadbolt
* Add Bootstrap CSS to webpage

**Materials**:

* Raspberry Pi (maybe just run it off a computer/server instead?)
* Wi-Fi enabled lightbulb
* High School students

Optional:

* Bluetooth speaker
* Electronic deadbolt

**Tools**:

* Python
* HTML

Optional:

* Bootstrap

**Idea**: Stenography in Images

**Description**: Create software that can encode text into images and also decode the images to retrieve the original message. A couple options for what programming languages to use for this project. Possibly C++ with Qt for the GUI, also could use Python with an HTML front end and host it the webpage on a cloud.

**Reasoning**: Stenography is a cool idea, though not very practical in the real-world. However, it can be a good learning experience for the high schoolers because the project will involve things such as bit shifting, image manipulation, and also GUI elements. This project would be significantly more technical from a coding aspect compared to the above idea. However, it may still be possible for them to complete with direction. While this project would not be as extensible as the above, I feel like there will be a sufficient workload to span the two week period. The program could be written in either C++ or Python. I feel that writing the program in C++ would make the project easier for me to prototype, but Python *may* be easier for the students (it might not make a difference since they have no experience anyways). While I develop the prototype version of the program, I can make helper functions or a library that the students could access as an API for the more complex portions of the program (but I wouldn’t want to make it trivial).

**Initial Goal**: Encode/decode text to/from an image file.

**Stretch Goals**:

* Upload custom images for encoding.
* Use more than one color channel for encoding.
* Loop encoded text (watermark effect).

**Materials**:

* High school students

**Tools**:

* C++ and Qt **–OR-**
* Python and HTML

Optional:

* Bootstrap

**Idea**: Notification Web Service

**Description**: Web service that sends important updates to user’s phone or email (or both!). This can have an HTML front-end and Python back-end. Essentially offer a bunch of modules that send certain notifications to the user. The list of options could be weather alerts, traffic alerts, news excerpts, etc. The user could select which types of alerts they want to receive, how often they want them, what method it should be sent, etc.

**Reasoning**: This project is similar in style to the “Smart” room project. It is also a large project that consists of a dozen or so smaller projects that are hopefully within the scope of the students skills. The initial goal could be as simple as “send text alerts about the weather.” Stretch goals could add different types of alerts or include modifying the time and frequency alerts are sent, or even send different alerts to different devices.

**Initial Goal**: Send SMS alerts about the weather

**Stretch Goals**:

* Send email alerts as well
* Send alerts based on weather condition
* Send alerts at certain times
* Send alerts every *X* minutes
* Send traffic alerts based on location (hard coded)
* Send traffic alerts based on location (detected)
* Send news alerts
* Send news alerts filtered by topic
* Etc.

**Materials**:

* High school students

**Tools**:

* Python
* HTML
* SMS/Email service

Optional:

* Bootstrap

**Idea**: “Smart” room

**Pros**:

* Big project broken into small projects
  + Easy to break up in teams
* Highly extensible
* Real-world application
* Connection of hardware and software

**Cons**:

* Materials cost
* Possible complexity?

**Idea**: Stenography in Images

**Pros**:

* Fun project with interesting results
* Learn about low-level aspects of code
  + i.e. bit patterns
* Extensible

**Cons**:

* No hardware involved
* Difficult to break up in teams?
  + Maybe do one team front-end one team back-end?
* No real-world application
* Complexity

**Idea**: Notification Web Service

**Pros**:

* Big project broken into small projects
  + Easy to break up in teams
* Highly extensible
* Real-world application

**Cons**:

* No hardware involved
* Possible complexity?