**An Exploration into the Role of Touch in Computer Programming**

Diagram

Description automatically generated

Official Project Documentation

*Organization: HaRVI Lab*

*Director: Heather Culbertson, PhD*

1. **Preamble**

Welcome! The purpose of this document is to provide an overview of our tactile coding project. Namely, it will cover the locations of project resources, materials, and will detail the remaining assembly required for project completion. The document is organized into three major sections: *Overview*, *Resources*, and *Assembly*. The *Overview* section will offer a description of the project as a whole and its intent. The *Resources* section details the locations of, and other sources of documentation for, project components. Lastly, the *Assembly* section will highlight the current state of the project and will cover the remaining steps for finishing the implementation of the project.

If at any point you should have questions regarding the content of this or other project documentation, please don’t hesitate to reach out to either Heather Culbertson ([hculbert@usc.edu](mailto:hculbert@usc.edu)) or Alex Atcheson ([saatcheson@gmail.com](mailto:saatcheson@gmail.com)) for clarification or assistance.

1. **OVERVIEW**

In this section, we will briefly discuss the goal of the project and the programming tool under construction. Moreover, we will cover the workflow that programmers will encounter when using the novel programming tool.

The genesis for this project is all about understanding the role that touch may play in computer programming. Namely, we are interested in exploring how individuals code through when tasked with using a tactile tool in place of digital text as a representation of computer code. As such, we have devised a novel programming environment whereby programmers use physical coding blocks, each representing a unique code construct, for writing functioning computer programs. Each of these physical coding blocks are 3-D printed elements, about the size of one’s palm, and are consist of a 3-D design on their surface. Via recognition of surface design features, programmers might rely on touch-related stimuli for informing coding practice, an anticipated question of this project.

The system functions as follows: A user would sit down at a desk with the programming tool resting on the desk in front of them. The programming tool is a box shape and can be easily reached in the sitting the position. Next to the box on the table are a variety of coding blocks that the individual can use for writing their own program. Once ready, the individual can begin placing the coding blocks on the interactive area of the programming tool in a sequential order, nearly identical to how one would write text-based code. Once the individual has written completed their program by arranging the coding blocks on the interactive space of the programming tool, they may press the programming tool’s “Run” button for executing their program.

One of the potential target audiences for a tool such as this would be for those individuals who experience some form of visual impairment, lending added difficulty to the task of coding. As such, the programming tool uses non-visual output as a method for signifying successful or unsuccessful program execution, removing the need for sight. To support non-visual output, the programming tool relies on a speaker system for conveying messages and auditory signals to the programmer.

In summary, the system flow would allow for programmers to write code by arranging physical coding blocks on the interactive space of the programming tool. When ready to execute, the programmer may press the programming tool’s “Run” button. Through the internal speaker of the programming tool, the programmer will be alerted of either successful program completion or of an error in their code.

1. **RESOURCES**

In this section, we will highlight the various materials associated with the project. Namely, we will cover location and organization information. The *Assembly* section will cover construction steps for completing the project; as such, any build-related information will not be covered here.

**High-Level Description**

The programming tool works using a combination of hardware and software features. The programming tool has been designed and built entirely from scratch, which means that the hardware and software elements have been curated for the purposes of the programming tool. The below subsections will offer a breakdown of these hardware and software elements.

**GitHub**

The GitHub repository, titled *sympl*, houses all the non-hardware project materials. This includes items such as source code, technical drawings, design files, user study design, and Raspberry Pi configuration files. A pointer to the repository can be found here: <https://github.com/atchesonUSC/sympl>.

To clone this repository onto your local machine, you can click the green “Code” button on the repository home page, copy the SSH or HTTPS link to your clipboard, and run the following command from your command line to clone the repository folder to the directory of choosing:

$ git clone git@github.com:atchesonUSC/sympl.git

**Repository Breakdown**

The root of the repository consists of a README file and several directories. For increased clarity regarding the structure of the directory hierarchy, each directory consists of a README file describing its contents and related purpose. Please refer to these README files for getting sense for the contents of each directory.

1. **design**
   1. **3d\_print**
      1. **gcode**

*Source code files for printing blocks on the Prusa extrusion printer.*

* 1. **laser\_cut**

*Files for upload to the Glowforge application for cutting.*

* 1. **cad**

*All CAD files for the programming tool and physical programming blocks.*

* 1. **tech\_drawings**

*Technical drawings of the board components containing dimension information.*

1. **documentation**

*Informational material on the project.*

* 1. **pics**

*Project pictures and related media.*

1. **old**

*Decommissioned materials, not in use.*

1. **rpi**

*Materials related to configuration and setup of Raspberry Pi.*

* 1. **overlays**

*Device tree patches that allow for adjusting the number of available SPI chip selects.*

* 1. **rfid**

*Source code for testing reading and writing capabilities of RFID reader configurations.*

* 1. **wifi**

*Stores wpa\_supplicant.conf files for connecting Raspberry Pi to wireless network, useful if using headless Raspberry Pi (i.e. without connecting the Pi to a computer monitor and solely operating from the command line via SSH).*

1. **samples**

*Old examples for testing the block code parser; however, these are a bit outdated and not super useful.*

1. **source**

*Contains all the source code for making the programming tool operate! main.py is the key file for running the programming tool. ParseTool.py is the second most important file, and is responsible for parsing the block code, catching and reporting errors, and formatting the lines of Python code to be written to an output file (which will be executed – this is the Python translation of the block code).*

1. **study**

*Contains information for user study design.*

**Project Box**

The bulk of the project hardware can be found in the project cabinet located on the workshop side of the lab space. Here’s an image of the project cabinet (it is the rightmost cabinet located in the background of the image):



Upon opening the project cabinet, the project box can be found on the shelf that is second from the bottom. Here is a picture of the cardboard project box:



Other tools that might be necessary for working on the assembly (i.e., soldering iron, 3D printing materials, scissors, etc.) are organized and can be found in the workshop side of the lab space.

1. **ASSEMBLY**

This section will cover the remaining assembly required for completing the programming tool. As covered in the *Resources* section, generally speaking, the materials required for completing the assembly can be found in the project box and the tools for doing so can be found in locations around the lab workspace. The following subsections will cover these remaining assembly steps.

**Laser Cut Board Siding**

The physical frame of the programming tool is constructed from several pieces of unfinished draftboard. The programming tool is in the shape of a rectangular box, and the layout of the siding is shown below:

Icon

Description automatically generated

Of the several box pieces, the top side still needs to be cut. The outward-facing side of the top piece is where users will place and arrange blocks, whereas the inward-facing side is where the sensors and stability components will be connected. The stability components in this case are small magnets arranged on the inward-facing part of the top side. These magnets sit in small casings, as per following image:

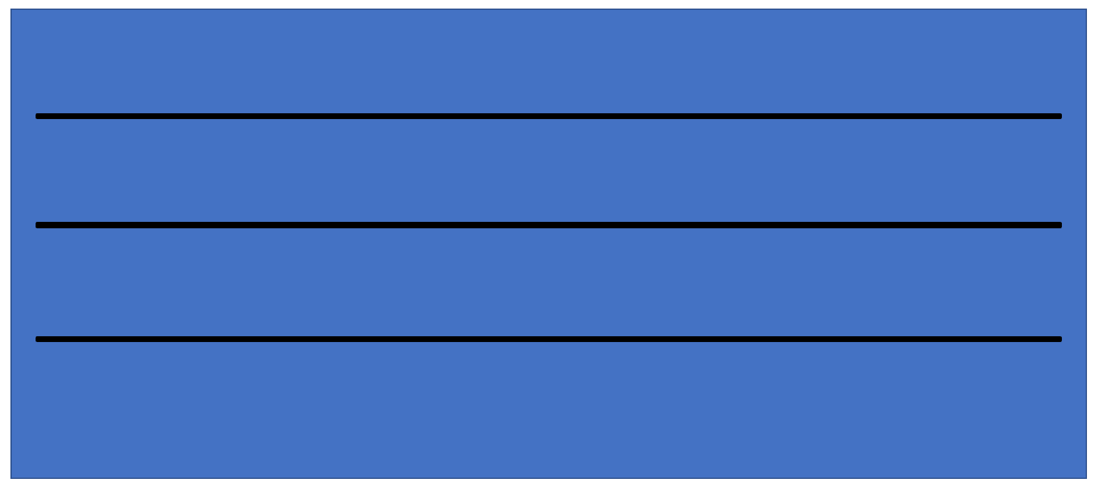
A picture containing funnel chart

Description automatically generated

Located in the /sympl/design/laser\_cut directory, there are two SVG files: top\_side\_upward.svg and top\_side\_inward.svg. Using a piece of draftboard from lab, the top portion of the board can be laser cut using the Glowforge (check in Sandeep for how to complete the Glowforge training). The Glowforge web app (app.glowforge.com) is where designs can be uploaded for lasering with the Glowforge. For top\_side\_upward.svg, set the oval shapes to be engraved and set the rectangular outline to be cut. Flip over the cut board piece and place back in the Glowforge bed. For top\_side\_inward.svg, set the circles to be engraved and set the outline to be ignored.

**Row Division**

The board consists of 4 rows, with each row supporting 6 block spaces. Dividers between the different rows should be added to reinforce the division between the different rows, as in the figure below:

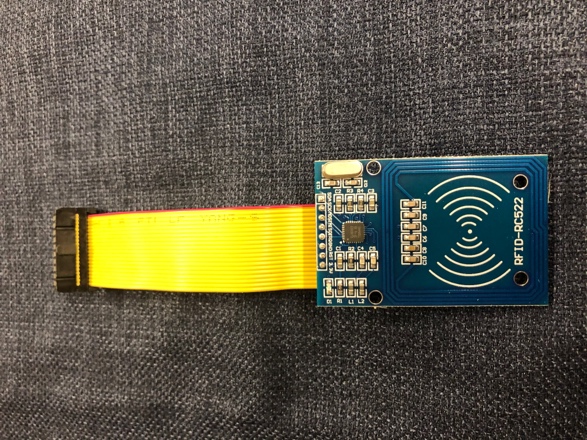


Three Dowel rods will work well for this, and they can be purchased on Amazon at the following page: [- https://www.amazon.com/Wooden-Square-Crafts-Woodworking-Projects/dp/B07T3F18BM/ref=sr\_1\_4?dchild=1&keywords=1%2F8"+square+rod&qid=1625078197&sr=8-4&th=1](-%20https:/www.amazon.com/Wooden-Square-Crafts-Woodworking-Projects/dp/B07T3F18BM/ref=sr_1_4?dchild=1&keywords=1%2F8%22%2Bsquare%2Brod&qid=1625078197&sr=8-4&th=1)/

The dowel rods can be cut to the appropriate size and connected to the board (a little bit of hot glue should do the trick here) in such a way to separate the rows as shown in the above figure.

**RFID Readers**

The programming tool utilizes RFID readers to detect the presence of the programming blocks placed on the surface of the tool. One of these RFID readers, with a yellow ribbon cable connected to it, is pictured here:



These RFID readers are connected to the inward-facing side of the top piece of the board box, as shown in the diagram below:



Altogether, since there are 24 total block spaces, there will be 24 total RFID readers connected to the inward-facing side of the top board piece, one for each block space. Some experimentation might be required, but some strong tape should do the trick in the way of connecting the readers to the top piece. The sensing side of the readers (shown in the picture of reader with the white markings) should be facing upward so that it can read the block sitting over its space.

Each of the RFID readers should have a yellow ribbon cable connected to their pins. Ultimately, this will allow for the readers to be easily plugged into the programming tool circuit board (a topic discussed more below).

**Programming Tool Audio – Speaker**

The programming tool conveys auditory output to the user via a small speaker mounted into the side of the board box. The speaker, along with its connecting cord for interfacing with the Raspberry Pi’s audio port, is pictured here:

A pair of black headphones on a wooden surface

Description automatically generated with medium confidence

If twisted, the speaker can extend in length, as in the following image:

A picture containing person, holding, hand, indoor

Description automatically generated

To insert the speaker into the board box, it is desired that the speaker be untwisted, and box side wedged in between the two ends, so that the audio-emitting portion face outward and the other portion face inward. The below figure gives a view of this design:

Chart

Description automatically generated

As shown in the diagram above, the speaker is wedged into the side piece of the board. This wedge opening must be carved out. To do this, you can take a piece of draftboard and, using the Glowforge, cut out the design entitled speaker.svg in the laser\_cut folder. Once the piece has been cut, you can carve out the opening to the whole using a dremel tool in lab. This step is pictured below:

A picture containing text

Description automatically generated

**Magnet Casings**

Magnets are used for providing a form of stability and control to block placement. These small magnets are inserted into 3D-printed casings, which are then attached to the inward-facing side of the top part of the board. An image of a magnet in its casing is shown below:

A black rock on a wooden surface

Description automatically generated with low confidence

The magnets can potentially be secured by using hot glue. These magnet casings can be 3D printed in lab using the Prusa extrusion printer. The gcode for the magnet casing can be found in the gcode folder in the 3d\_print directory. Once you have downloaded and installed the PrusaSlicer application (useful for 3D printing with our lab’s printer), you can simply use this gcode file. Here’s where you can download the PrusaSlicer tool: <https://www.prusa3d.com/prusaslicer/>

From the subsection *Laser Cut Board Siding*, it discussed the engraving process that occurs to the inward-facing side of the top board piece. These engraved holes are where the magnet casings will be connected, as illustrated in the second graphic of that section.

**Compile/ Execute Button**

The programming tool has a single button for executing code, which is located on the side of the tool. An image of this button is shown below:

A picture containing yellow, indoor

Description automatically generated

This button is “normally open,” which means pressing it will close the circuit.

**Power Supply**

To power the programming tool, it must be plugged into a wall outlet. The plug for the tool is pictured here:



The red wire is for power and the black wire is for ground.

**Circuit Board**

The programming tool consists of several electronic components: Raspberry Pi, speaker, button, and RFID readers. The Raspberry Pi is the brains of the tool, the speaker is for auditory output, the button is executing the block code, and the RFID readers are for sensing the type and location of the blocks. Internally, these components are connected to the same circuit board.

First let’s talk about the protoboards that comprise the circuit board. Each protoboard is consists of 30 columns (with label “A” on top and “D” on the bottom) and 48 rows, as shown in the picture below:

A picture containing text, cage, kitchen appliance

Description automatically generated

Each column is shorted to be a single node, meaning that all the holes in a column are connected. This makes it convenient for connecting a row of readers in parallel, where each RESET pin on a reader is connected a column designated for carrying the RESET signal, and so on.