SIMPLE TANGIBLE INTERFACE

PROJECT 001

TIME: A WEEKEND | LVL: BEGINNER | COST: <$100

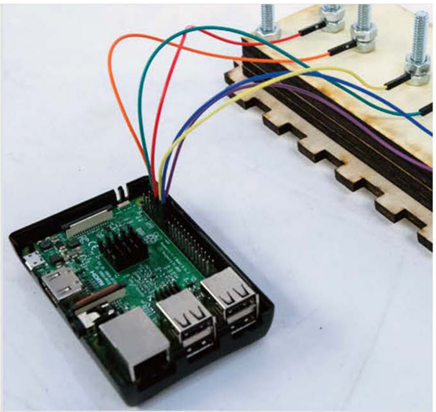
This project is a set of instructions on how to build a very simple, tangible "touch" digital interface for an interactive video exhibition. The user can hold, touch and examine physical pieces, and then insert them into the interface to start up videos related to each piece.

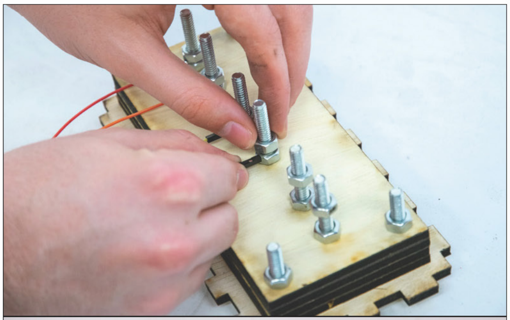
The example project here is generic -- the pieces are simple geometric volumes, with corresponding descriptive videos. This is a useful, quick, and inexpensive tutorial on how to start building such exhibits. Preferably the user has access to "maker" tools, like a 3D printer, laser cutter, and simple hand tools.

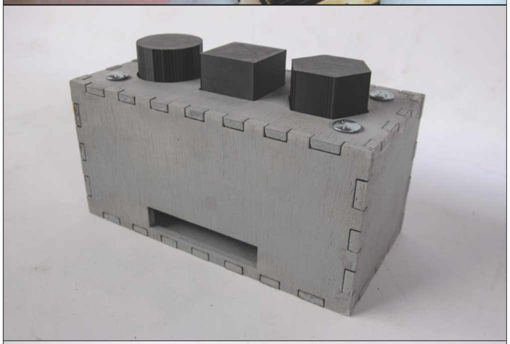
The same strategies and technologies can then be used to build another project with a specific theme, determined by the exhibition designer (you). The generic physical pieces can be replaced with other artifacts, or 3D printed "tangibles". The related videos can be designed or selected to support the interaction and theme.

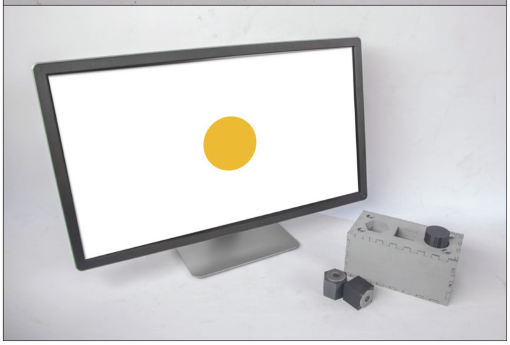
We used these exact techniques to design and build an interactive museum exhibition. The user could hold and touch 3D printed replicas of fragments of a Roman frieze (a decorative sculpture), and place them into the table like a puzzle, triggering narrative videos describing the artifacts. The technology used was extremely inexpensive, yet robust enough for a permnanent installation.

This is an open source project, so feel free to copy and alter it as you see fit -- just share your work with others!









WHAT YOU'LL NEED

GADGETS

- A Raspberry Pi computer, ideally the **Raspberry Pi 3 Model B**. **$35**
- An **8 GB (or more) micro-SD card**. This is the "hard drive" for the Pi. **$7.5**
- A **5V 2.4A power supply with micro-** **$10**
- **Jumper wires**. For three tangibles you'll need no fewer than six 12'' female-to-male jumper wires (two wires for each tangible), though it would probably be wise to have plenty of extras. **$7.5**

MATERIALS

- A **2' x 4' sheet of quarter-inch thick wood** for your box. Be wary that not all types of wood will laser cut well. If you're not sure what to get, **baltic birch plywood** is inexpensive and laser cuts well. **$10**
- **Wood glue**, for adhering the various cut-out pieces of your box together. **$5**
- **Epoxy**, for adhering your washers to the bottom of your 3D-printed tangibles. **$5**
- A bit of **steel wool** for polishing off your metal bits. This can help conductivity. **~$1**

HARDWARE

- 6 **quarter-inch hex tap bolts** with a length of around **1 inch**. **~$1**
- 4 **quarter-inch bolts** of your choosing, with a length of around **2 inches**. **~$1**
- 16 **quarter-inch hex nuts**. **~$1**
- 3 **12mm** x **37mm metric fender washers**. **~$3**

TOOLS

- A **computer with an internet connection** for the initial installation of the Raspian operating system onto your micro-SD card.
- A **keyboard and mouse** for using and setting up your Raspberry Pi. At the end of the process, the Pi will automatically launch the necessary script as it boots up, so you'll no longer need to use a keyboard and mouse unless you want to make adjustments.
- An **HDMI-compatible display**, as well as an **HMDI cable** to hook it up.

SERVICES

- Access to a **laser cutting machine** for cutting the box pattern into your wooden sheet.
- Access to a **3D printer** for fabricating your tangibles.

1. BUILDING THE BOX

First things first, you'll need to fabricate the wooden box which is going to act as a case for all your electronic components, as well as a base for the tangible interface.

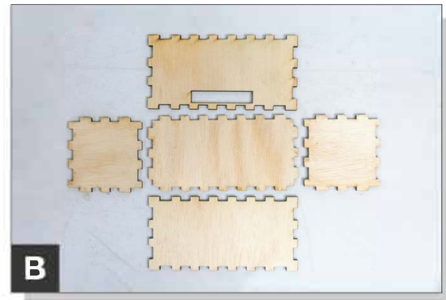
1A. LASER CUTTING

You could potentially build this box by hand, but for this project we recommend using a laser cutter.

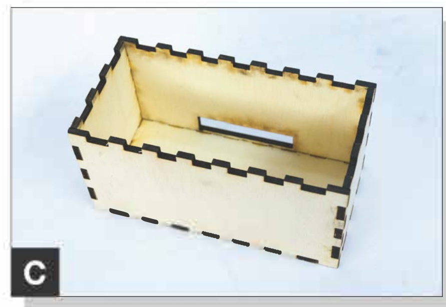
You can find the vector file with the laser cutting pattern at the following link:
`https://github.com/vnc-ncsu/simpletangible_prj_001/blob/master/files/boxplans.ai`

This process will differ based on your particular laser cutter machine and software. You'll have to use the power settings, etc. that works for your particular laser-cutting setup and choice of material. In all you should end up with ten cut-out wooden pieces. (Fig. **B** & **D**)

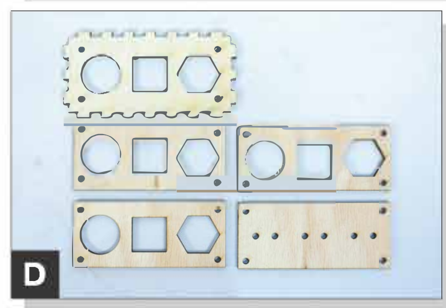
A



B

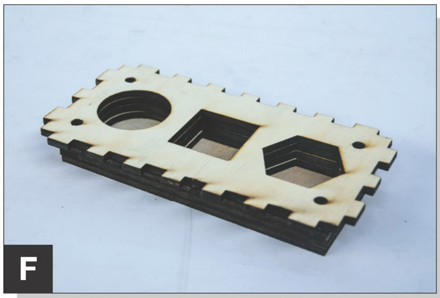


C

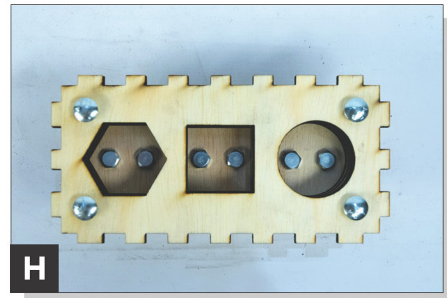


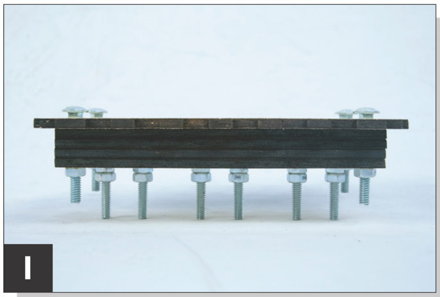
D

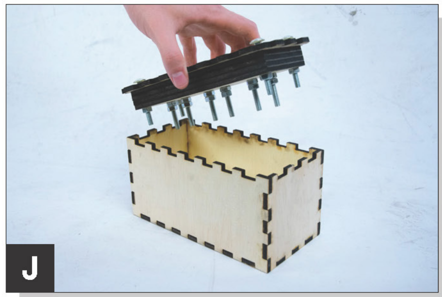
E

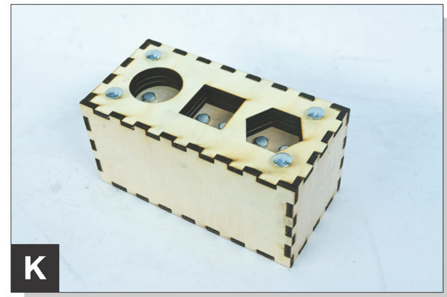

F


G


H


I


J


K

Of these ten pieces, five (Fig. **B**) will glue together to form the base of your box and the other five (Fig. **D**) will be pressed together using nuts and bolts to form the top of your box.

1B. WOOD GLUING

Take the five pieces shown in Figure **B**. Using wood glue, adhere these five pieces to one another to form a box shape with an open top that looks like Figure **C**. Let the glue dry in the manner specified on the wood glue container.

1C. NUTS AND BOLTS

While you wait for your glue to dry, take your remaining five pieces (Fig. **D**) and simply stack them on top of each other, with the tooth-edged wooden piece on the top of the stack and the piece with six small holes on the bottom (Fig. **E**). Arrange the pieces carefully so that the quarter-inch circular holes at each of the corners of the pieces are aligned (Fig. **F**).

Insert your four 1/4", 2-inch long bolts into each of the circular holes at the corners. Screw one nut onto each of these shafts so that the stack is pressed together tight (Fig. **G**).

1D. CONTACT POINTS

The box is nearly finished. The last thing to do is to add some more nuts and bolts to the top of your box. These nuts and bolts are intended to act as electrical contact points for the tangible pieces.

Insert your 6 quarter-inch, 1-inch long hex tap bolts into the 6 remaining quarter-inch holes in the top of your box. Tighten these in place with 6 nuts.

Lastly, with the 6 remaining nuts, add one to each of the bolts, so that each bolt has two nuts attached to it. The two nuts on each of the bolts will act as a sort of makeshift, metallic, conductive clamp which will keep the male ends of your jumper wires firmly in place, with no soldering required. The top of your box should now look something like Figure **I**.

At this point, your box is finished! If your glue has dried, you should be able to place the top-half of your box into the base-half so that the two "lock" together.

2. MAKING THE TANGIBLES

Now that you have a box with appropriately-shaped slots in it, you need to actually make the tangible objects which will fit inside these slots.

You will 3D print three plastic, polygonally-shaped base pieces, and then adhere a metallic washer onto the bottom of each piece with epoxy.

2A. 3D PRINTING

You can find the .obj file with the three tangible objects at the following link:

`https://github.com/vnc-ncsu/simpletangible_prj_001/blob/master/files/tangibles.zip`

As with the laser cutting, this process will differ based on your particular 3D printing equipment and software.

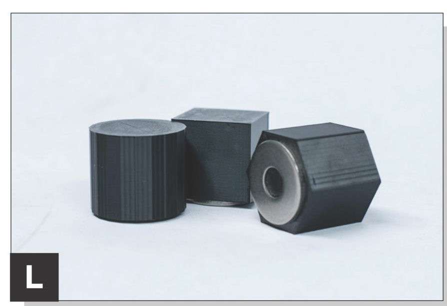
Here are the dimensions that will ensure a good fit with the slots in the box, as well as compatability with the 12mm x 37mm washers:
Cube: 38.1 mm wide, 38.1mm long, 39.7 mm tall
Hex: 47.7 mm wide, 41.3 mm long, 39.7 mm tall
Cylinder: 43.8 mm wide, 43.8 mm long, 39.7 mm tall
All: 160.0 mm wide, 43.8mm long, 39.7 mm tall

2B. INSTALLING THE WASHERS

After your tangible pieces are 3D printed, adhere one 12mm x 37mm fender washer onto the underside of each tangible using epoxy.

Note that the underside of each tangible is the side featuring a center-aligned circular peg.

Follow the instructions listed on your epoxy to adhere the washers to your tangibles. The end result should look something like Figure **L**.


L

3. SETTING UP THE PI

You've assembled the box and the tangibles. The only "hands-on" step left is installing the jumper wires. Before you get to that, it would be wise to perform the first-time setup for your Raspberry Pi, if you haven't already.

3A. INSTALLING RASPBIAN

Raspbian is a lightweight, linux-based operating system. It also comes with the scripting language Python pre-installed. In summary, Raspbian is an ideal OS choice for this project.

Loads of documentation already exists online to help you get Raspbian installed on your Pi. For easy-to-understand documentation of installing Raspbian onto your micro-SD card, check out the official Raspberry Pi documentation online at the following link:

raspberrypi.org/documentation/installation

3B. BOOTING YOUR PI

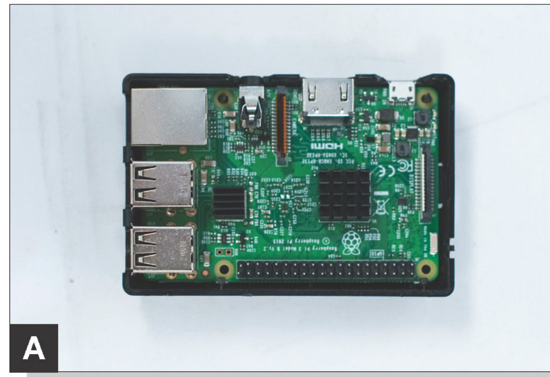
Once you've got Raspbian installed on your micro-SD card, you'll need to insert the card into the Raspberry Pi, as well as connect a USB mouse and keyboard, and external display. Finally your Pi should be ready to go. It will boot up automatically when you connect the power supply. Now, it's time to square away the last "hands-on" step -- the installation of the jumper wires.

4. INSTALLING THE WIRES

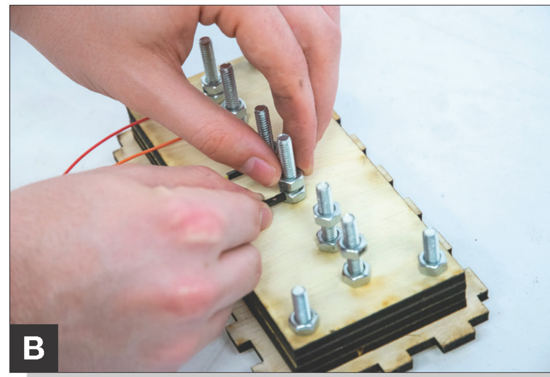
Now you'll use your 6 jumper wires to create a circuit that connects the Raspberry Pi to the tangible contact points in your box.

4A. UNDERSTANDING THE SYSTEM

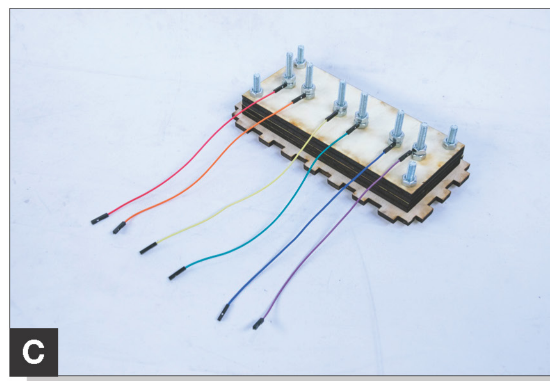
The Raspberry Pi is able to detect a completed electric circuit. If the tangible is absent from its slot, the circuit is incomplete (Left half of Fig. **D**). However, whenever the tangible is placed into its corresponding slot, the metallic washer touches each of the contact bolts, thus completing the circuit (Right half of Fig. **D**). The Raspberry Pi will be able to detect this and trigger the appropriate code.



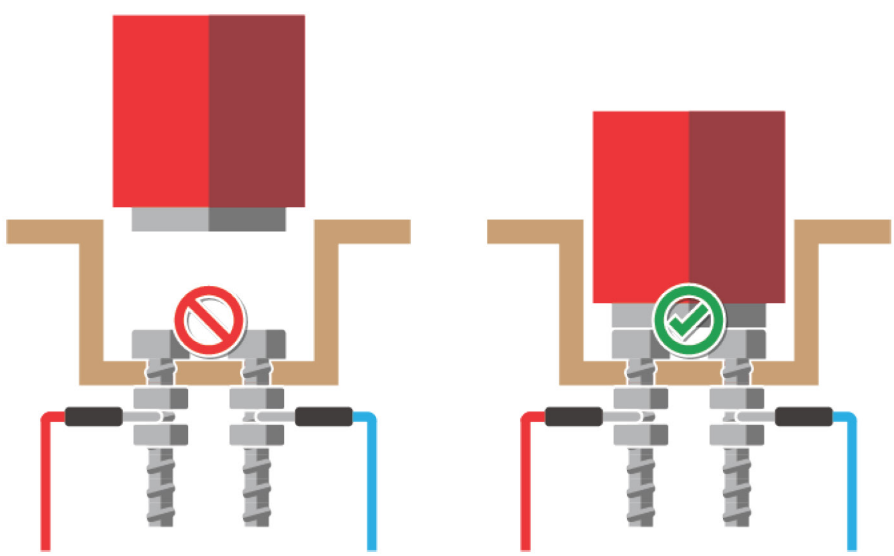
A



B



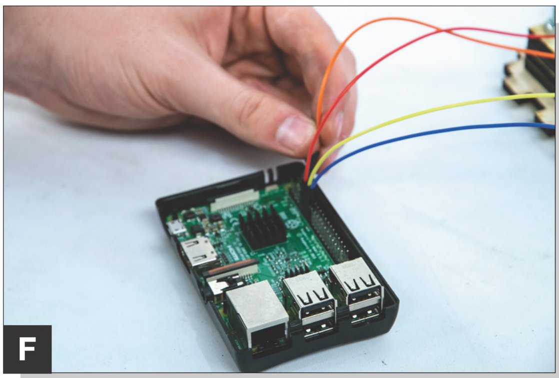
C



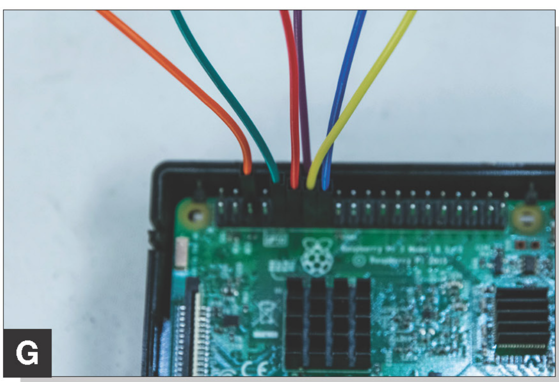
D

Pin	Description	Description	Pin
01	3.3v PWR	5v PWR	02
03	GPIO 02	5v PWR	04
05	GPIO 03	Ground	06
07	GPIO 04	GPIO 14	08
09	Ground	GPIO 15	10
11	GPIO 17	GPIO 18	12
13	GPIO 27	Ground	14
15	GPIO 22	GPIO 23	16
17	3.3v PWR	GPIO 24	18
19	GPIO 10	Ground	20
21	GPIO 09	GPIO 25	22
23	GPIO 11	GPIO 08	24
25	Ground	GPIO 07	26
27	Reserved	Reserved	28
29	GPIO 05	Ground	30
31	GPIO 06	GPIO 12	32
33	GPIO 13	Ground	34
35	GPIO 19	GPIO 16	36
37	GPIO 26	GPIO 20	38
39	Ground	GPI0 21	40

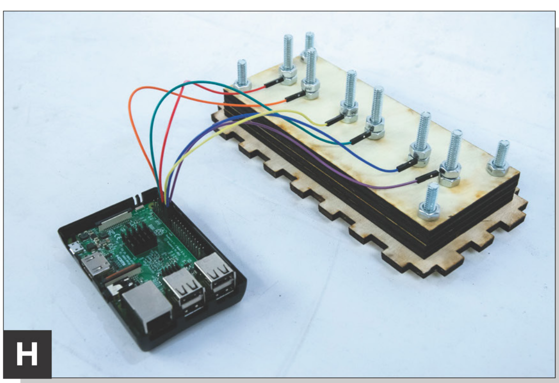
E



F



G



H

4B. MALE ENDS TO BOLTS

You'll need to start by attaching the male ends of each of your six jumper wires to the shafts of your six contact bolts. Simply place the male end of each of your jumper wires between the two nuts on each of the bolts, tightening the loose bolt with your free hand so as to clamp it against the other nut (as in Fig. **B**).

When you've performed this step with each of your six jumper wires the result should look something like Figure **C**.

4C. UNDERSTANDING GPIO

The final step is to press the female ends of your jumper wires onto the appropriate GPIO ports on your Raspberry Pi.

The GPIO pins are the 2 x 20 skinny metallic pins sticking out of your Raspberry Pi. These pins provide input, output, power, and ground pins for external devices and circuits. **Randomly placing wires onto these pins is a bad idea**, as the wrong configuration of wires could short circuit your Pi and potentially cause permanent damage.

The handy chart in Figure **E** is a quick summary of the function of each of the 40 GPIO pins on your Raspberry Pi. Note that in this diagram, the pins at the top (starting with 01 and 02) correspond to the GPIO pins closest to the micro-SD card slot on your Raspberry Pi.

For this project, the only two types of GPIO pin that you should concern yourself with are the standard-variety GPIO pins (depicted in Figure **E** as blue) and the ground pins (depicted in Figure **E** as black).

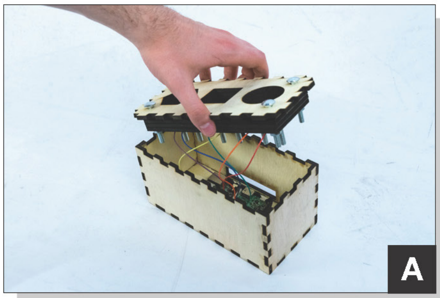
4D. FEMALE ENDS TO GPIO

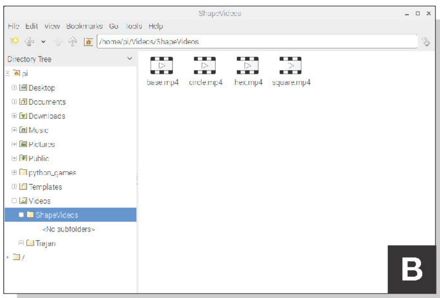
You can use any of the blue GPIO pins that you prefer for this process, but it would be easiest if you just use the same GPIO pins that we used, which are marked with a green-background pin number in Figure **E**: pins 11 (GPIO 17), 13 (GPIO 27), and 15 (GPIO 22), corresponding to the circle, square, and hex shape slots respectively. This will save you a bit of work down the line when you're setting up the code, since the default values will already be correct!

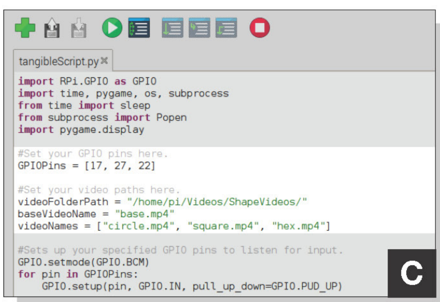
Connect either one of the loose jumper wires from each of your three shape slots to your correspondingly chosen GPIO pins, fitting the head of the female end over the pin and pushing it down slightly until it rests on the pin securely (Fig. **F**).

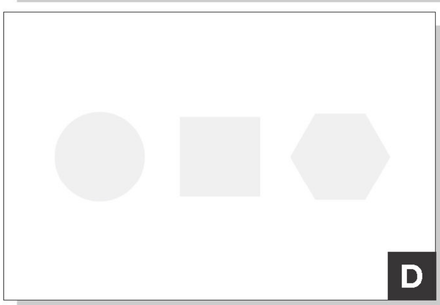
Repeat the process with your three remaining loose jumper wires, attaching them to any Ground pins. Which Ground pins you decide to use is unimportant, because each Ground pin is functionally identical -- but if you want to use our layout, we happened to use pins 06, 09, and 14 (depicted with a purple-background pin number in Figure **E**).

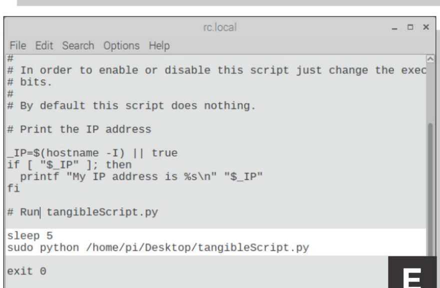
At the end of the process, the pins should look something like Figure **G**, and the entire setup (box-top included) should look something like Figure **H**.











5. LAST STEPS

Congrats, all the physical fabrication necessary for the project is complete! The last steps are simply a matter of booting up your Raspberry Pi, transferring and organizing a few media files, and modifying a couple of lines of code.

5A. TRANSFERRING THE VIDEO FILES

With three shape slots, you'll need at least four video files -- one to play in response to each tangible being placed into its corresponding shape slot, and one "base" video to play when no tangibles are currently placed.

We've provided a bundle of four appropriately-themed videos to use as a starting point at the following link:

`https://github.com/vnc-ncsu/simpletangible_prj_001/blob/master/files/ShapeVideos.zip`

Download this .zip and extract the videos into an appropriately-named folder in a file location you can remember -- we used the location **"home/pi/Videos/ShapeVideos/"**, but anywhere else you prefer is fine.

5B. SETTING UP THE SCRIPT

Now it's time to get your Python script setup and running. This program will wait until a tangible is placed in its corresponding shape slot, and then play the appropriate video.

The script can be found at the following link:

`https://github.com/vnc-ncsu/simpletangible_prj_001/blob/master/files/tangibleScript.py`

Save this script in another good file location you can remember. We simply placed it on the desktop -- **"/home/pi/Desktop/tangibleScript.py"**.

Open up the script using your Python editor of choice -- several are installed automatically with Raspian. You'll notice a few lines of code at the top of the script declaring a few variables, like `GPIOPins`, `videoFolderPath`, and `videoNames`.

For **`GPIOPins`**: if you chose the same GPIO pins that we did in Step 4D (GPIO 17, 27, and 22), great! You don't have to do anything here. Otherwise, replace these values with the GPIO values that you selected yourself. Remember to keep the same format, i.e. commas separating each value and the entire list enclosed on either side with brackets.

For **`videoFolderPath`**: if you used a different file location other than the default, change this value to be the location where you placed your videos.

For **`baseVideoName`** and **`videoNames`**: unless you changed the file names of the videos in ShapeVideos.zip, you should be able to leave these. However, simply understand that these are the file names of the videos which will play in response to the placement of a tangible. Note that the order of the filenames in `videoNames` actually does matter -- the order has to correspond to the order of the pins designated in `GPIOPins`.

The rest of the code should be able to stay the same. Indeed, you probably shouldn't try to change any of it, unless you're a Python programmer yourself and you're curious to see how it works or whether you can improve it.

Lastly it should be noted that this program is fully capable of supporting as many tangible/video pairings as you might care to have -- this is to say, in the future if you decide to expand the concept of this project to include even more than three tangibles and shape slots, this script can still accomodate that. Simply add any additional GPIO values and/or video names to the end of the appropriate list -- e.g. `[17,27,22,5,6,13]` for `GPIOPins` or `["circle.mp4","square.mp4","hex.mp4","triangle.mp4","star.mp4"]` for `videoNames`.

5C. RUNNING THE SCRIPT

Everything should be ready to go. Run the script and you should be greeted by a screen with looping playback of base.mp4. When you insert a given tangible into its slot, playback should switch to the corresponding video. Pressing the Escape key will exit the script.

Having problems? If nothing happens when you put a tangible into its slot, there are a couple simple solutions you can try:

First, double-check the GPIO pins you used, the pins you declared for the `GPIOPins` variable, and the video file names you declared in the `videoNames` variable.

Second, it's possible the metal surface of your contact points and/or metal washers need a bit of polishing in order to conduct an electrical current more consistently.

5D. AUTORUN AT BOOT

This step is optional, but recommended. Having your tangibleScript.py Python script run automatically when the Python boots up will save you the hassle of having to plug in a mouse and keyboard to the Pi everytime you turn it on.

First, open the Leadpad text editor as the super-user by issuing the following command in the command line: `sudo leafpad`

In the text editor, open the file named "rc.local" located at the following filepath: **"/etc/rc.local"**

Add the following two lines of code, making sure to place them before the line "`exit 0`", which always has to go at the end:

```
sleep 5
sudo python (Your file path here)
```

Now whenever your Pi boots, it will wait 5 seconds and then automatically run the script.