**Activity: Customized Useless Boxes**

BY: Connor, Daniel, Jared, Raina

GRADE AND CAMP: Tesla (7-9)

TOPIC(s): 3D modeling, soldering, electronics

|  |
| --- |
| TIME: ~6 hours |
| Lesson: ~45 mins |
| Project Building: ~5 hours |
| Clean Up: 15 mins |

**MATERIALS (what you’ll need for one week):**

Per camper:

3D character

- TinkerCAD website (1 account works for all children)

- 1 head or object to be used as decoration

Box

- 6 pre-cut ¼” wooden panels (2 of 4”x5”,2 of 3” x 5”, 2 of 3” x 4”)

- 8 to 12 ¼” screws

- Wood glue

- 4 3D printed corner hinges

- 2 3D printed lid hinges

- 3D printed bottom plate

Moving parts

- 1 arm to be 3D printed

- 1 box lifting arm

Electronic components

- 1 Raspberry Pi Pico W

- 1 breadboard

- 1 toggle switch

- 2 micro servo motors

- 3 LEDs

- 3 AA batteries

- 1 battery pack (for 3 AA batteries)

- 3 330-ohm resistors (optional, helps prevent burnout in LEDs, also cause we have like a million 330-ohm resistors)

- lots of wires (jumper or solid core)

**OVERVIEW:**

Mankind has spent thousands of years developing technology to help millions of people everyday. All tech has a use… except this one. This little box likes things how they were before and refuses to compromise. You want to change one little thing, flick one little switch, but the box will flick it right back. Children will create the little box through 3D printing, handyman work, soldering and creating circuits. Children will learn how to use CAD through TinkerCAD and their creations will come to life by 3D Printing. Wood will be supplied so kids will learn how to create housing for their electronics, which they will create by gluing and screwing a box together. Inside their box they will create a circuit to run the useless box using micro servo motors, a PICO W, a DC battery, switches and more. Once complete they will touch up the code for their PICO W to create varying patterns for the box to turn the switch back. But that’s not all, we will be attaching a speaker for the kids to program noises of their choice to play when the box is in operation.

**PROCEDURE:**

**Day 1: Tinker CAD**

Materials

**Campers**

-TinkerCAD website (all campers should be able to use same account

**Instructors**

-CURA software (the slicer; should only need one copy for instructor)

-3D printer and acrylic spools

3D modeling their character will be the first project as we need time to print the objects. They will use the program TinkerCAD

**Overview**

3D Modeling is the way of the future (and largely the present)! Using even the simplest 3D modeling software, we can create complex designs which can be viewed in all 3 dimensions, whether digitally or in real life through use of 3D printing technology.

In this project, campers will be given a brief introduction into how 3D models are made and how 3D printers are able to print STL files. Campers will create a small character (roughly 1”x1”x1.5”) which will act as the useless box inhabitant.

**How it works**

3D modeling is a combination of mathematics, geometry and design that allow creators to digitally construct items that can then be “printed” using a 3D printer. Unlike traditional modeling, where the model components are made out of clay or other physical material, 3D modeling allows the creator to be as detailed as they want and edit at their leisure, without having to spend on new materials or even leave the comfort of their chair.

Essentially, 3D modeling software allows the user to “sculpt” objects using precise specifications and measurements. When the user is done, the file can then be saved as a set of instructions on how to replicate the thing designed. These files are saved in STL format, where STL stands for “stereolithography”. This means that only the surfaces of the object are recorded, using a coordinate system and a lot of triangles.

From here, the STL file can be fed into a program called a Slicer. The slicer takes the 3D STL file and “slices” it into a stack of 2D layers.

The 3D printer reads each layer individually and prints them one on top of another. When these layers are stacked up correctly, the form the original 3D object, complete with a structured in-fill to prevent it from collapsing.

Using this technology, designers and engineers can create parts and products in hours instead of weeks. Large scale 3D printers can even create things such as rockets and camper trailers from raw materials.

**During Camp Day:**

March all campers to a computer lab (Delta works well as it has a big screen to demo on and all the chairs face the same way) and have them log on.

Have campers navigate to TinkerCAD.com and click “Sign in” in the top right corner. Then use the Gmail option and enter the required login information from below.

**Email:** [scifi.tesla.tinkercad@gmail.com](mailto:scifi.tesla.tinkercad@gmail.com)

**Password:** Sciencecamp22

Once logged on, have campers create a new file for themselves by clicking “create new design” in the top left corner.

NOTE: Each now new file will have a randomly generated name. This name can be found in the top left corner next to the TinkerCAD logo. Click on this to change file name to match camper.

NOTE: Campers WILL have access to each others files due to the joint account, so be sure to warn campers not to mess with each others files.

After they have created their new file, allow campers to begin trying to use basic shapes and others files from the right hand window to create a 3D model of their own head. Once the shapes have been put together, they **MUST** be grouped using the button in the upper right side of the screen.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Copy | Paste | Duplicate | Delete | Undo | Redo | Group | Ungroup | Export |
|  |  |  |  |  |  |  |  |  |

NOTE: For time and resource managerial purposes, ensure all models are no more than 25x30x38 mm in dimension. A box of this size can be copy and pasted from the “Head Template” File.

If students wish to use the demo box to judge their head size, be sure to DELETE THE BOX before saving the file.

While the students are working, you can talk about or put a video on some of the new advancements and industrial application of 3D printing.

Once the students have completed their design, show them how to save as an STL file. This can be accomplished by clicking on the head design, and then pressing the “Export” button on the top right corner. Make sure that “The selected shape” is the bubble selected, and then press “For 3D Print: .STL”. The campers will then be asked where they wish to save their file.

If time remains, have campers use “scribble” tool to write their own first name in the workspace. (This can and might be used at a later date). After this, they can make whatever they want.

All files save automatically to TinkerCAD, so when time is up, have campers simply close the web browser.

**After Camp Day:**

As soon as campers have left for the day, log into TinkerCAD and use the copy and paste tool to bring all completed heads into a single workspace. Ensure that all heads are within the 25x30x38 mm size limit. Once this has been completed, organize the heads into “X”s of 5. Group these “X”s and export the selection as an STL. These are the files that will be printed, so you may want to save them with the names of the campers so that you don’t forget whose head is whose.

An example of the grouped “X”s can be found in the TinkerCAD files, if you need an example. It is entitled “Groups 2 Print”.

**IF THE SCI-FI PRINTER IS WORKING:**

Talk to Sam or fill out 3D printing request form. Campers need them back by Thursday morning to complete the useless box.

**IF THE SCI-FI PRINTER IS NOT WORKING:**

Contact Rob Peace in the engineering building and ask very nicely if he can print the STL files for you using his machines. He is very nice and will (hopefully) agree. You may need to provide him with some of PLA filament found in the SCI-FI 3D printing drawer in the office as payment. NOTE: This is a stopgap until we can find a better in-house option. Mr. Peace is doing us a HUGE favour here.

**SAFETY CONSIDERATIONS:**

3D printers melt plastic. They are hot, be careful when touching parts of the 3D printer, AKA don’t touch nozzle or base right after it stops printing.

**Box Assembly**

**Materials**

Per camper

- 6 pre-cut ¼” wooden panels (2 of 4”x5”,2 of 3” x 5”, 2 of 3” x 4”)

- 12 ¼” screws

- 4 3D printed corner hinges

- 2 3D printed lid hinges

General supplies

- power drill

- wood glue

**Overview**

Electronics cannot just sit out in the open, they need some housing to protect them from the environment. Campers will be given the materials listed above and will be required to assemble the box. They will do this by using screws and hinges as well as the wood glue to create a firmly attached box with no lid. They will then attach the lid with 2 hinges so it can be opened and closed.

**Procedure**

1. Hand out 6 sheets of wood, 2 of each type, screws, glue, hinges, etc

**Breadboarding & Electronic Assembly**

By: Daniel Yu

**Materials:**

Per camper:

* 1 assembled plywood box
* 1 breadboard
* 1 Raspberry Pi Pico
* 2 micro servo motors & kit
* 1 toggle switch
* 3 LEDs
* 3 330-ohm resistors
* 3 AA batteries
* 1 battery pack (for 3 AA batteries)
* Tons of wires (jumpers and solid core ones are fine)
* 3D printed arm

General supplies:

* Soldering iron & solder
* Small screwdriver
* Hot glue

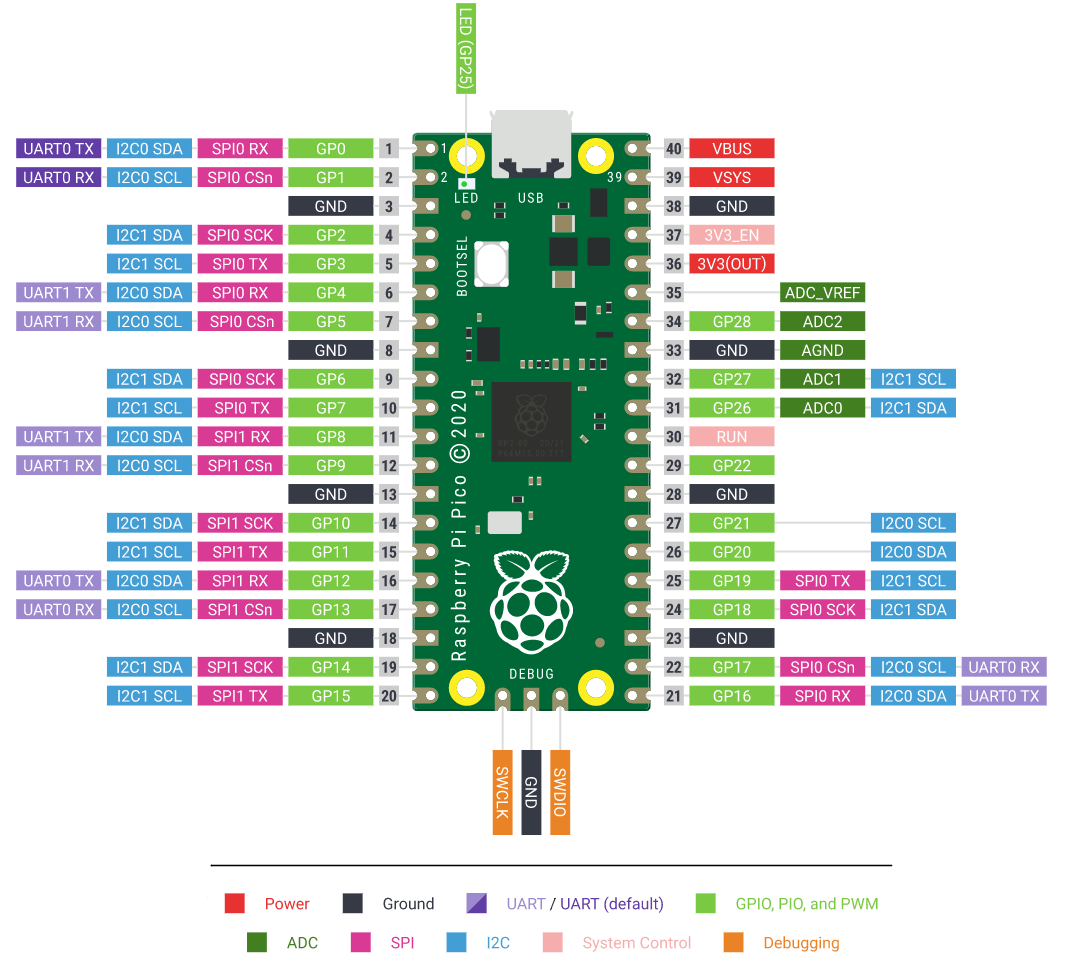
**Overview:**

Breadboard go brrrrr. Explain circuits and electricity and how they work!

**Procedure:**

1. Start by going over the basics inputs, outputs, power, and ground pins on a Raspberry Pi Pico and how connections work on a breadboard
   1. Pull up Raspberry Pi Pico pinouts
   2. All row numbers are electrically connected while all + and – columns are electrically connected on a breadboard
2. Get the campers to place their Picos into their breadboard (ideally, the MicroUSB side is on the edge of one of the short side faces)
3. Assemble your Micro Servo Motors; help the campers screw in their motor blades, ideally one servo has the short one-sided blade and the other has the long two-sided blade
4. Have some campers start soldering two solid core wires to their toggle switch while the others begin following the schematic on the board or have the instructors lead them step by step (see pictures below)
   1. One possibility is having the instructors pull up all the pinouts on the Pico and Daniel’s hand-drawn schematic on the projector and get the campers to wire everything themselves while going around and helping them around (see pictures below)
   2. Or have the instructors lead each step by step and have the campers follow along
   3. Meanwhile, campers should rotate between them to solder two wires to their toggle switch (does not matter which wire they solder, just two wires)
5. Listed below are all the connections required:
   1. GND (Pin 8) 🡨 🡪 Negative breadboard column
   2. 3V3(OUT) (Pin 36) 🡨 🡪 Positive breadboard column
   3. Arm Micro Servo (Sig, Orange Wire) 🡨 🡪 GP0 (Pin 1)
   4. Arm Micro Servo (VCC, Red Wire) 🡨 🡪 Positive breadboard column
   5. Arm Micro Servo (Gnd, Brown Wire) 🡨 🡪 Negative breadboard column
   6. Lid Micro Servo (Sig, Orange Wire) 🡨 🡪 GP1 (Pin 2)
   7. Lid Micro Servo (VCC, Red Wire) 🡨 🡪 Positive breadboard column
   8. Lid Micro Servo (Gnd, Brown Wire) 🡨 🡪 Negative breadboard column
   9. [FOR ALL THREE LEDs] LED Anode (Positive Side, Long Leg) 🡨 🡪 330 Ohm Resistor 🡨 🡪 GP14 (Pin 19), GP15 (Pin 20), and GP16 (Pin 21) respectively
   10. [FOR ALL THREE LEDs] LED Cathode (Negative Side, Short Leg) 🡨 🡪 Negative breadboard column
   11. Toggle Switch Edge (“Off”-Sided Leg) 🡨 🡪 GP13 (Pin 17)
   12. Toggle Switch Middle (“On”-Sided Leg) 🡨 🡪 Positive breadboard column
   13. Battery Pack Black Wire (Negative End) 🡨 🡪 Negative breadboard column
   14. Battery Pack Red Wire (Positive End) 🡨 🡪 Positive breadboard column
6. Note that all the GP connections to the Pico are not strictly those connections, it can be moved around to different pins but the code will need to reflect that as well
7. Help the campers out with connections, make sure that they are not short circuiting anything and that the battery pack should always be “OFF”

**Pictures:**

****

**A diagram of a circuit board

Description automatically generated with low confidence**

**Notes:**

* GP pin connections can be switched around with other pin connections
* Connections to the positive or negative breadboard columns can be connected anywhere along that column
* Pay close attention to detail when double checking wiring connections
* PLEASE REMIND THE CAMPERS TO KEEP THE WIRING AS CLEAN AS POSSIBLE, HELPS YOU DEBUG THE WIRING LATER
* Any wiring issues, send questions to Daniel and ping him on Discord

**Safety Considerations:**

* Watch the campers solder carefully
  + Make sure they don’t burn themselves or others
  + No pointing soldering irons
* Remind them to keep their battery packs on “OFF”

**Software & CircuitPython Coding**

By: Daniel Yu

**Materials:**

Per camper:

* 1 MicroUSB to USB-A cord
* 1 Assembled breadboard ready-to-go
* 1 Raspberry Pi Pico (should be in breadboard already)

General supplies:

* Computers
* Internet
* Thonny Application
* Adafruit Motor Library
* Camper Template Code
* Passion, Patience, & Persistence

**Overview:**

We need code to make our amazing wiring job work! ☺ Explain coding for beginners and its importance in our every day lives! Ask for other coding languages and talk about Python, the difference of CircuitPython, and how powerful Python and CircuitPython is because of its libraries which we will be using!

**Procedure:**

1. Plug in the MicroUSB to the Pico and the USB-A into your computer
2. Get the campers to follow the instructions at: <https://learn.adafruit.com/getting-started-with-raspberry-pi-pico-circuitpython/circuitpython>
   1. Make sure they complete THIS PAGE ONLY
   2. Help them download Thonny if their accounts don’t have it (probably first week only)
3. Download the .zip file from: <https://circuitpython.org/libraries>
   1. Specifically, “Bundle for Version 8.x”
4. **Unzip the file and find the “adafruit\_motor” folder in the “lib” folder**
5. **Copy the “adafruit\_motor” folder and paste it into the “lib” folder on your CIRCUITPY drive**
6. Give a basic rundown of Python, its variables, conditionals, etc. and start coding!
   1. Either give the campers a template of the code (found either on our GitHub or Sci-Fi camper folder) and explain how the code works and what it does and tell them which parts to fill in with their code lines
   2. Code the Example Code (see end of this lesson) with all the campers line by line on Thonny (must have a lot of patience with repeatedly debugging and a lot of time to kill 🡪 probably not in Tesla)

**Notes:**

* **This activity can be switched with the “Final Assembly” one (see last lesson) so they know what exactly is happening with their fully assembled box when they code**
* Ideally, all of these steps should be done be an instructor on the projector step by step so the campers can follow pretty easily
* Instructors should familiarize themselves with the two links above before teaching and leading this lesson
* Getting the BOOTSEL reset can be hard and confusing so feel free to help campers out with that
* **Every camper’s Pico must poop up as CIRCUITPY on their desktop after they’ve followed the tutorial**
* **Make sure that the “adafruit\_motor” folder is inside the “lib” folder on each camper’s CIRCUITPY drive**

**Safety Considerations:**

* Stay safe on the web kiddos
* Stranger danger

**Example Code:**

#

#

# Project: Useless Box CircuitPython Code

#

# Written By: Daniel Yu

# Date: June 8, 2023

# Edited: June 25, 2023

# Planned By: Connor Dawson, Jared Steier, Raina Owen, Daniel Yu

#

# Tesla – Useless Box Project

# Sci-Fi Science Camps 2023

#

#

# Basic library imports

import board, time, digitalio, pwmio, random

# Importing motor and audio libraries

from adafruit\_motor import servo

# Initializing the random integer

rand\_int = 1

# Setting input output for button

button = digitalio.DigitalInOut(board.GP13)

button.switch\_to\_input(pull = digitalio.Pull.DOWN)

# Setting input output for LEDs

led\_one = digitalio.DigitalInOut(board.GP14)

led\_one.direction = digitalio.Direction.OUTPUT

led\_two = digitalio.DigitalInOut(board.GP15)

led\_two.direction = digitalio.Direction.OUTPUT

led\_three = digitalio.DigitalInOut(board.GP16)

led\_three.direction = digitalio.Direction.OUTPUT

# Create PWMOut objects on Pin GP0 and GP1

pwm\_lid = pwmio.PWMOut(board.GP0, duty\_cycle = 2 \*\* 15, frequency = 50)

pwm\_arm = pwmio.PWMOut(board.GP1, duty\_cycle = 2 \*\* 15, frequency = 50)

# Create servo motor objects

lid = servo.Servo(pwm\_lid)

arm = servo.Servo(pwm\_arm)

# Various LED lighting effects determined by the given integer parameter

def lighting(effect):

# Effect 1: Quick flashing

if effect == 1:

for i in range(8):

led\_one.value = True

led\_two.value = True

led\_three.value = True

time.sleep(0.05)

led\_one.value = False

led\_two.value = False

led\_three.value = False

time.sleep(0.05)

# Effect 2: Slow blinking

elif effect == 2:

for i in range(5):

led\_one.value = True

led\_two.value = True

led\_three.value = True

time.sleep(0.1)

led\_one.value = False

led\_two.value = False

led\_three.value = False

time.sleep(0.1)

# Fast rotating flashes

elif effect == 3:

for i in range(5):

led\_one.value = True

led\_two.value = False

led\_three.value = False

time.sleep(0.05)

led\_one.value = False

led\_two.value = True

led\_three.value = False

time.sleep(0.05)

led\_one.value = False

led\_two.value = False

led\_three.value = True

time.sleep(0.05)

# Fast dual rotating flashes

elif effect == 4:

for i in range(5):

led\_one.value = True

led\_two.value = True

led\_three.value = False

time.sleep(0.08)

led\_one.value = False

led\_two.value = True

led\_three.value = True

time.sleep(0.08)

led\_one.value = True

led\_two.value = False

led\_three.value = True

time.sleep(0.08)

# Keeps all LEDs on when closing the lid

led\_one.value = True

led\_two.value = True

led\_three.value = True

# Main movement

def movements(num):

# Turns on all LEDs when lid opens

led\_one.value = True

led\_two.value = True

led\_three.value = True

# ---------- Movement 1: Regular Movement ---------- #

if num == 1:

lid.angle = 60

time.sleep(1) # Wait 1 second

arm.angle = 180

# --------------- End of Movement 1 --------------- #

# ---------- Movement 2: Fast Lid & Arm ---------- #

elif num == 2:

lid.angle = 60

arm.angle = 180

# --------------- End of Movement 2 --------------- #

# ---------- Movement 3: Lighting Show ---------- #

elif num == 3:

lid.angle = 60

lighting(random.randint(1, 4)) # Calls for a light show

time.sleep(0.05)

arm.angle = 180

# --------------- End of Movement 3 --------------- #

# ---------- Movement 4: Annoyed & Petty ---------- #

elif num == 4:

lid.angle = 90

time.sleep(0.5)

lid.angle = 180

time.sleep(1)

lid.angle = 30

time.sleep(0.5)

lid.angle = 180

time.sleep(3)

arm.angle = 180

time.sleep(2)

arm.angle = 90

time.sleep(0.5)

arm.angle = 150

# --------------- End of Movement 4 --------------- #

# ---------- Movement 5: Throwing A Fit ---------- #

elif num == 5:

for i in range(2):

for j in range(3):

lid.angle = 30

time.sleep(0.2)

lid.angle = 180

time.sleep(0.2)

for k in range(5):

lid.angle = 30

time.sleep(0.1)

lid.angle = 180

time.sleep(0.1)

lid.angle = 30

lighting(random.randint(1, 4))

for x in range(3):

arm.angle = 150

time.sleep(0.25)

arm.angle = 0

time.sleep(0.25)

arm.angle = 0

lighting(random.randint(1, 4))

arm.angle = 180

time.sleep(0.5)

arm.angle = 0

# --------------- End of Movement 5 --------------- #

# All closing animations function

def exit\_movement(num):

# Lid closing animation for Movements 1

if num == 1:

arm.angle = 0

time.sleep(1)

lid.angle = 180

# Lid closing animation for Movements 2, 3, & 5

elif num == 2 or num == 3 or num == 5:

arm.angle = 0

time.sleep(0.2)

lid.angle = 180

#Lid closing animation for Movement 4

elif num == 4:

time.sleep(0.75)

arm.angle = 0

time.sleep(0.2)

lid.angle = 180

# Main loop

while True:

# Starts the Useless Box if switch is flipped on

if button.value:

movements(rand\_int) # Calls the different movement function given a random integer

# Otherwise, keep all LEDs off and lid closed

else:

exit\_movement(rand\_int) # Calls the closing arm and lif animation given the same integer as the movement

# Turns off all the LEDs

led\_one.value = False

led\_two.value = False

led\_three.value = False

rand\_int = random.randint(1, 5) # Generates a random movement for the next switch flip

**Final Assembly**

By: Daniel Yu

**Materials:**

Per camper:

* 1 assembled box
* 1 connected breadboard

General supplies:

* Hot glue
* Duct tape

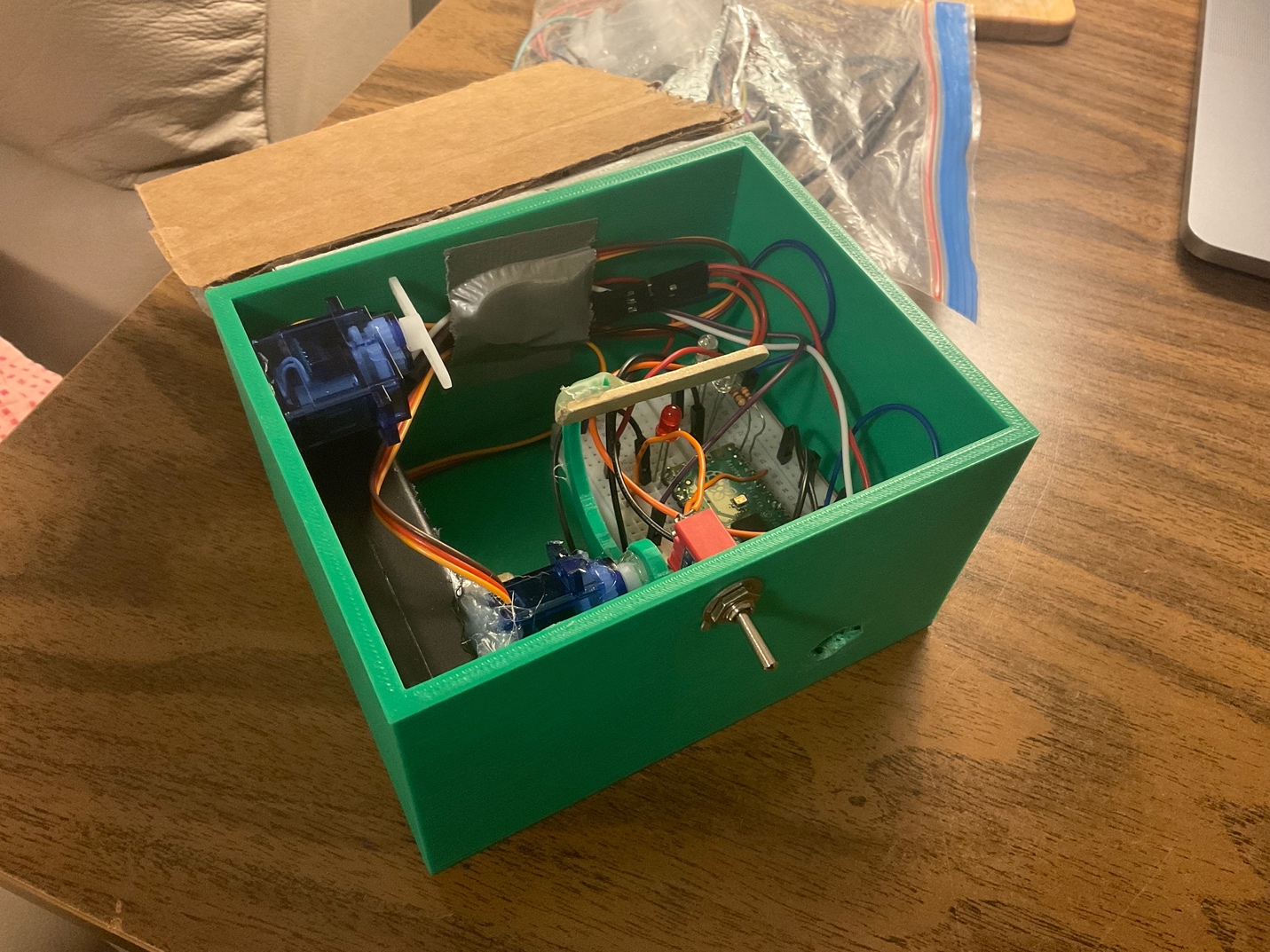
**Overview:**

Final step before we done babyyyyy!!! <3

**Procedure:**

1. Begin by thinking and testing how you want to organize the inside of your box
2. Once you’re ready, peel off the sticker on the back side of the breadboard and stick it into the botton of the box (align the Pico’s MicroUSB port with the opening so they can code more later if they have one)
3. Hot glue the battery pack making sure the open/close sliding lid is still moveable and the switch is accessible (through opening in the box or the box’s lid)
4. Hot glue the lid servo the in one of the top back corners of the box keeping it on the side of the hinges
5. Place the toggle switch carefully through the main centre hold at the front of the box and secure it using the provided washers and nuts
   1. **Make sure the “OFF” switch is facing the ground and the “ON” side is facing up**
6. Hot glue is the 3D printed arm onto the arm servo making sure its secure in place
7. **Test run the box multiple times while holding the arm servo to make sure it can hit the switch**
8. Hot glue the arm servo wherever you need and according to the tests in order for it to flip the switch
   1. Give campers cardboard if they need to build contraptions to help support the arm servo
   2. Arm servo must be secure so overload on the hot glue if necessary
9. See pictures below to see the prototype’s internal setup as a guide!
10. **Ta-da! Welcome to your very own USELESS BOX!!! :0**

**Internal Pictures:**

****

**A green box with wires and wires

Description automatically generated with low confidence**

**Notes:**

* **This activity can be switched with the “Software & CircuitPython Coding” one from earlier so they know what exactly is happening with their fully assembled box when they code**
* **Finding the correct position for the arm servo is extremely hard so get campers to help each other while instructors go around and help as a third hand (took myself >2 hours)**
* This is the part where many campers might realize their box doesn’t work and instructors might need to help find some designs later
  + Ping Daniel on Discord or bring it to him on EH in the afternoon as he can fix some boxes if there’s a lot not working
  + Explain to campers that this is simply the engineering process and that rarely does something you build the first time ever work perfectly
  + Try to create a box for ourself every week in case one camper has a box that is such a lost cause you can just give your extra one

**Safety Considerations:**

* Don’t burn or duct tape yourself
* **To Tesla Instructors: Have fun, don’t stress, you’re doing amazing!**