Inverted SLS Printer

Brock Taylor, John Douglas Whitehead

Fall 2022

Table of Contents:

1. Load Cell Integration

Overview

In order to ensure proper pressure placed on each layer of the print during use, a load cell was integrated into the machine to stop motor movement once a certain threshold of force was applied. This was an overhaul of the existing system for running the vertical movement of the pressure plate above the laser on the printer.

Load Cell

Load cell integration was done according to the following tutorial using the HX711 arduino library:
https://randomnerdtutorials.com/arduino-load-cell-hx711/

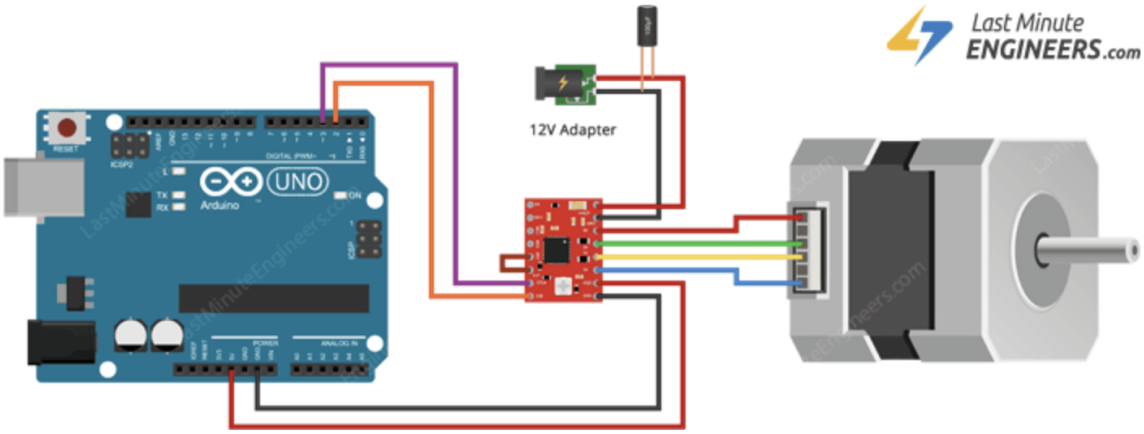
Load cell use is fairly straightforward. The following code snippet can be used to take measurements within the loop() section of arduino programming.

```
if (scale.is_ready()) {
   reading = scale.get_units();
}
```

The load cell reports data in native units by default. Therefore, the most important aspect of using the load cells is properly calibrating them for the setup that they are being used in. This is done by taking calibration measurements based on a known weight applied to the setup. From this, a calibration factor can be calculated as the ratio of native units reported to the known weight. set_scale() can be used to automatically convert readings to proper units based on this calibration factor.

Stepper Motor

Stepper motor wiring was done according to the following tutorial.
https://lastminuteengineers.com/a4988-stepper-motor-driver-arduino-tutorial/



Documentation for the AccelStepper arduino library can be found here:
https://www.airspayce.com/mikem/arduino/AccelStepper/

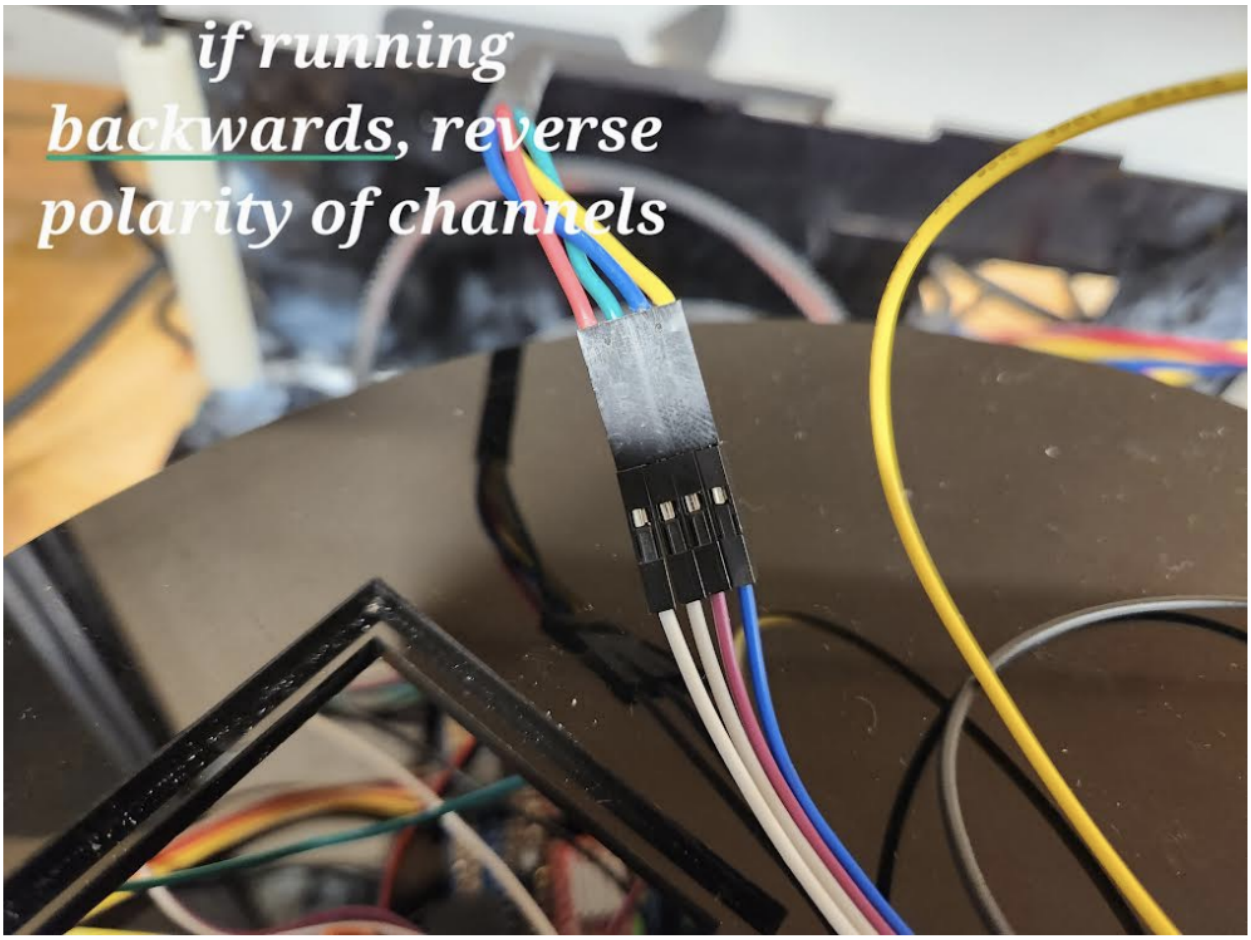
Instructions for Running

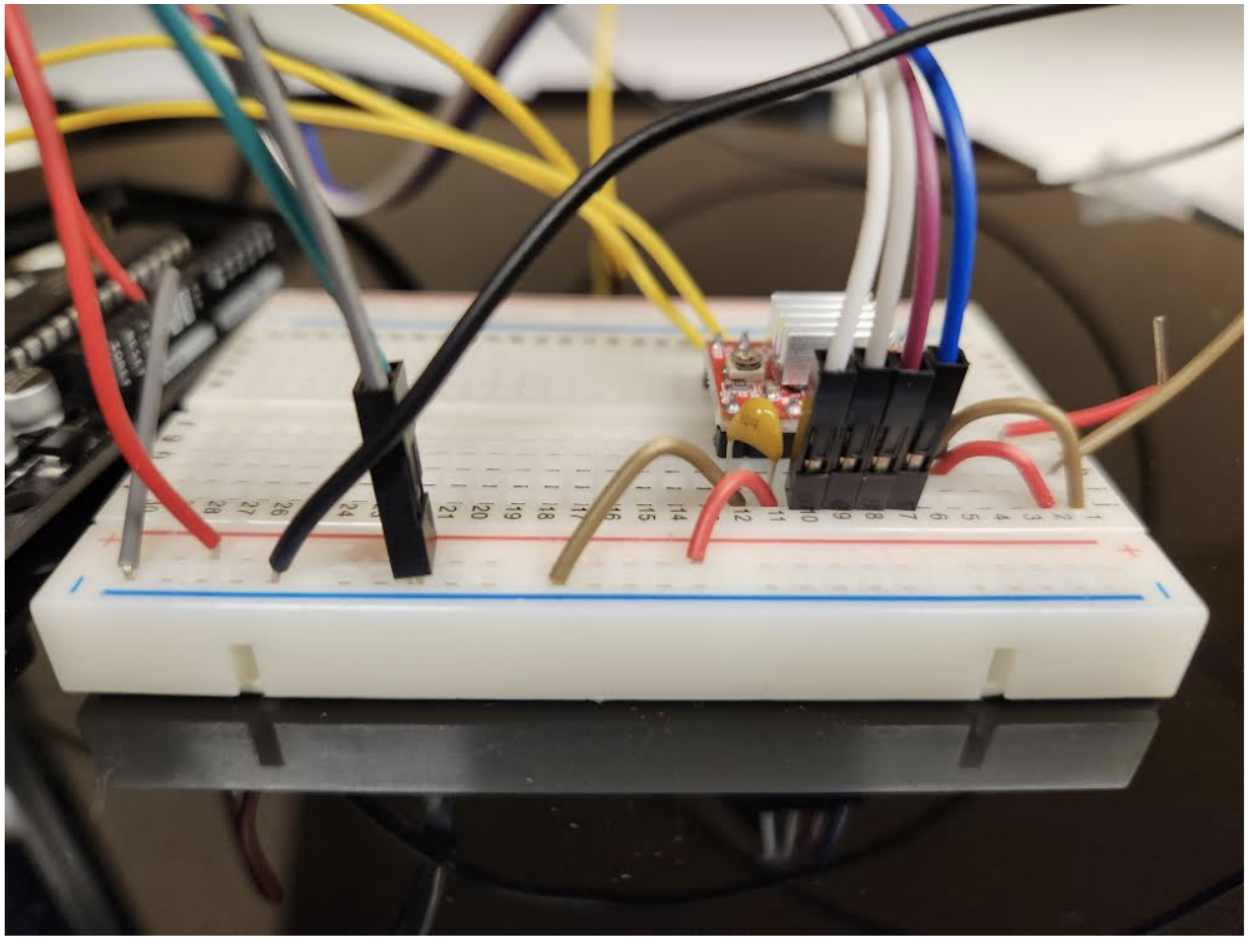
Wire the stepper motor and load cell according to the tutorials above. If the motor is running in the opposite direction (i.e. it starts by going up), you can either reverse the polarity of the wires or negate the values for controlSpeed() in the arduino program.

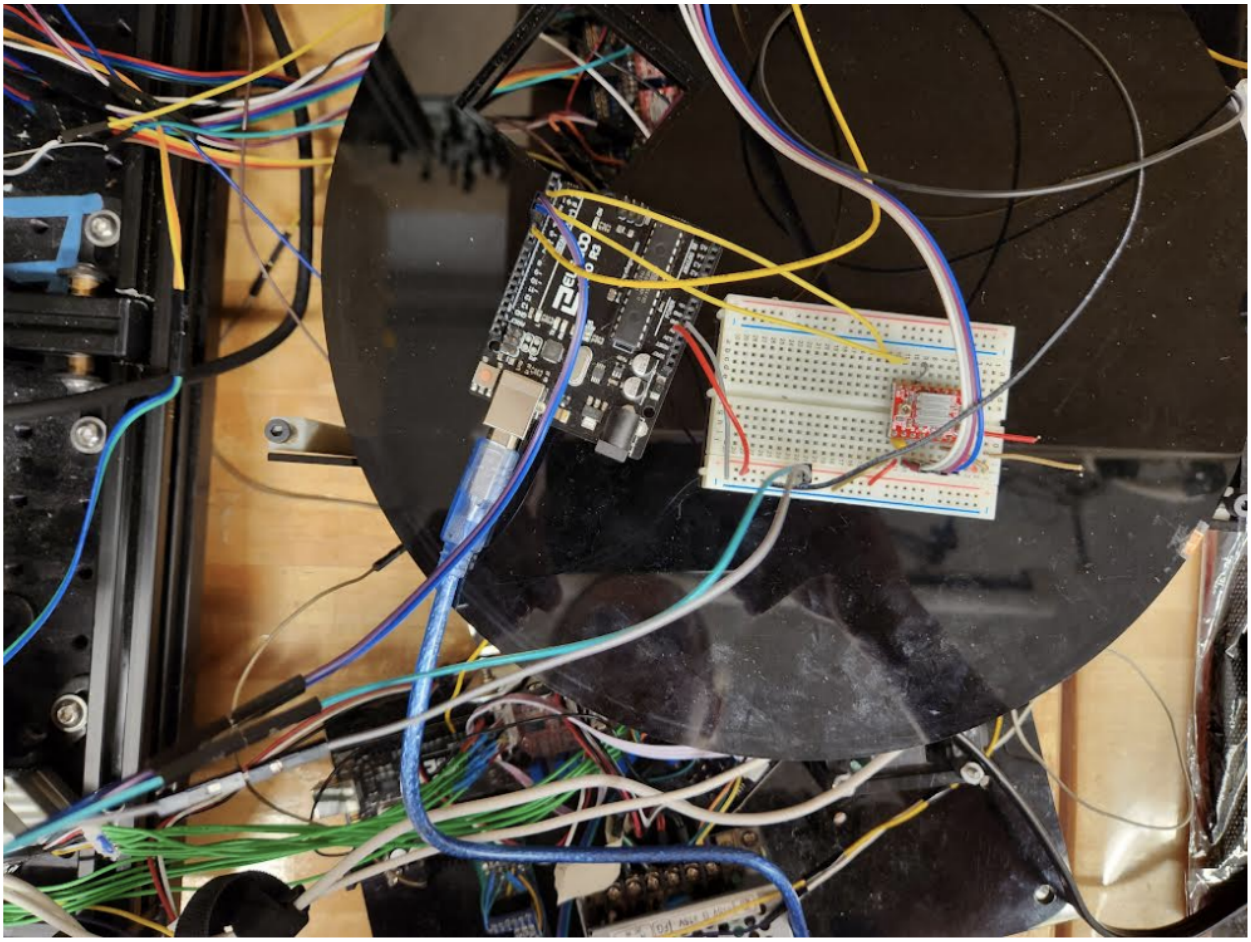
wires to plug in:
- motor wires (adjacent wires should connect to adjacent wires to match phases)
- load cell wires (color coded)
- motor power supply (12v as indicated)
- arduino usb cable

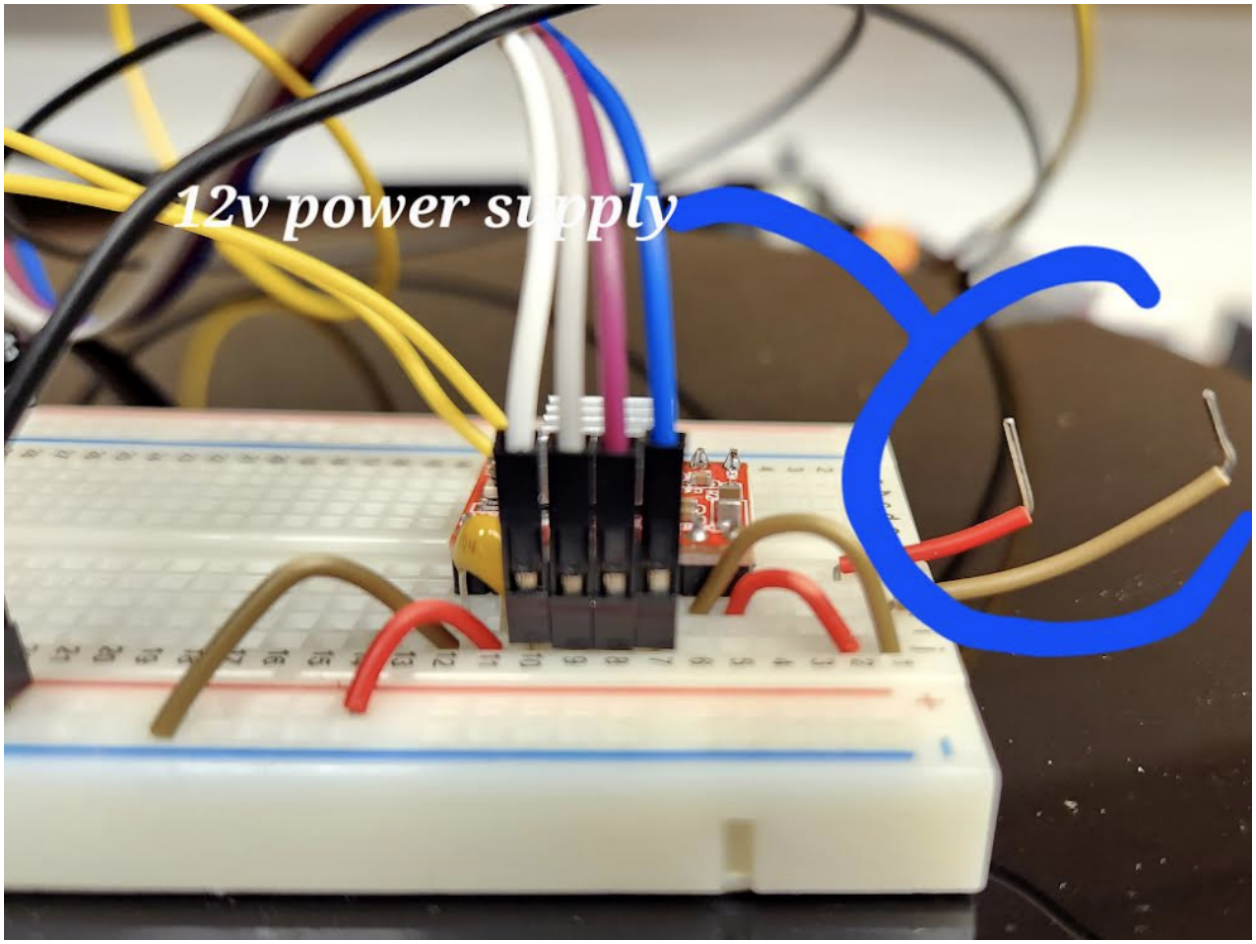
running procedure:

It should start in a clear position at the top where it can tare the load cell (load cell should equal zero at this point). It will then start moving down until a value of MAX_SCALE_VAL is read by the load cell. It will then move up until the limit switch is pressed. This process repeats. If it zeroes improperly, the program will need to be reuploaded or the arduino will need to be power cycled (it only zeroes in the setup() portion of the program)








12v power supply

Arduino Programs

Load Cell Calibration:

```
/*
  Rui Santos
  Complete project details at https://RandomNerdTutorials.com/arduino-load-cell-hx711/

  Permission is hereby granted, free of charge, to any person obtaining a copy
  of this software and associated documentation files.

  The above copyright notice and this permission notice shall be included in all
  copies or substantial portions of the Software.
*/

// Calibrating the load cell
#include "HX711.h"

// HX711 circuit wiring
const int LOADCELL_DOUT_PIN = 4;
const int LOADCELL_SCK_PIN = 5;

HX711 scale;

void setup() {
  Serial.begin(57600);
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
}

void loop() {

  if (scale.is_ready()) {
    scale.set_scale();
    Serial.println("Tare... remove any weights from the scale.");
      delay(5000);
    scale.tare();
    Serial.println("Tare done...");
    Serial.print("Place a known weight on the scale...");
    delay(5000);
    long reading = scale.get_units(10);
    Serial.print("Result: ");
    Serial.println(reading);
  }
  else {
    Serial.println("HX711 not found.");
  }
  delay(1000);
}
//calibration factor will be the (reading)/(known weight)
```

Stepper Motor and Load Cell:

```
// Include the AccelStepper library:
#include <AccelStepper.h>
#include <ezButton.h>
#include "HX711.h"

// Define stepper motor connections and motor interface type. Motor interface type must be set to 1 when using a driver:
#define dirPin 2
#define stepPin 3
#define motorInterfaceType 1
#define scale_constant 1

// HX711 circuit wiring
const int LOADCELL_SCK_PIN = 5;
const int LOADCELL_DOUT_PIN = 4;

const int MAX_SCALE_VAL = 3000;
long reading;
bool goingUp;
int controlSpeed;

HX711 scale;

// Create a new instance of the AccelStepper class:
AccelStepper stepper = AccelStepper(motorInterfaceType, stepPin, dirPin);
ezButton limitSwitch(6);

void setup() {
  Serial.begin(57600);
  goingUp = false;
  // Set the maximum speed in steps per second:
  stepper.setMaxSpeed(200);
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
   scale.set_scale(224.659); //calibration factor. 224659 native units per kilogram. 224.659 native units per gram
  scale.tare();
  reading = 0;
  limitSwitch.setDebounceTime(50);
}

void loop() {
  limitSwitch.loop();



   //in downward config, run down until scale reaches MAX_SCALE_VAL
   //in upward config, run up until limit switch is reached
  if (scale.is_ready()) {
      reading = scale.get_units();
    Serial.print("Y: "); Serial.print(reading); Serial.print("  ");
    Serial.print("Z: "); Serial.print(MAX_SCALE_VAL); Serial.print("  ");
    Serial.println("uT");
  }

  controlSpeed = -30;
  if(goingUp){
    controlSpeed = 30;
  }
```

```
  if(abs(reading) > MAX_SCALE_VAL){
    goingUp = true;
    delay(60000);
    reading = 0;
  }
  else if(limitSwitch.isPressed()){
    goingUp = false;
  }
  stepper.setSpeed(controlSpeed);
  stepper.runSpeed();

}
```

2. Substrate Cutting

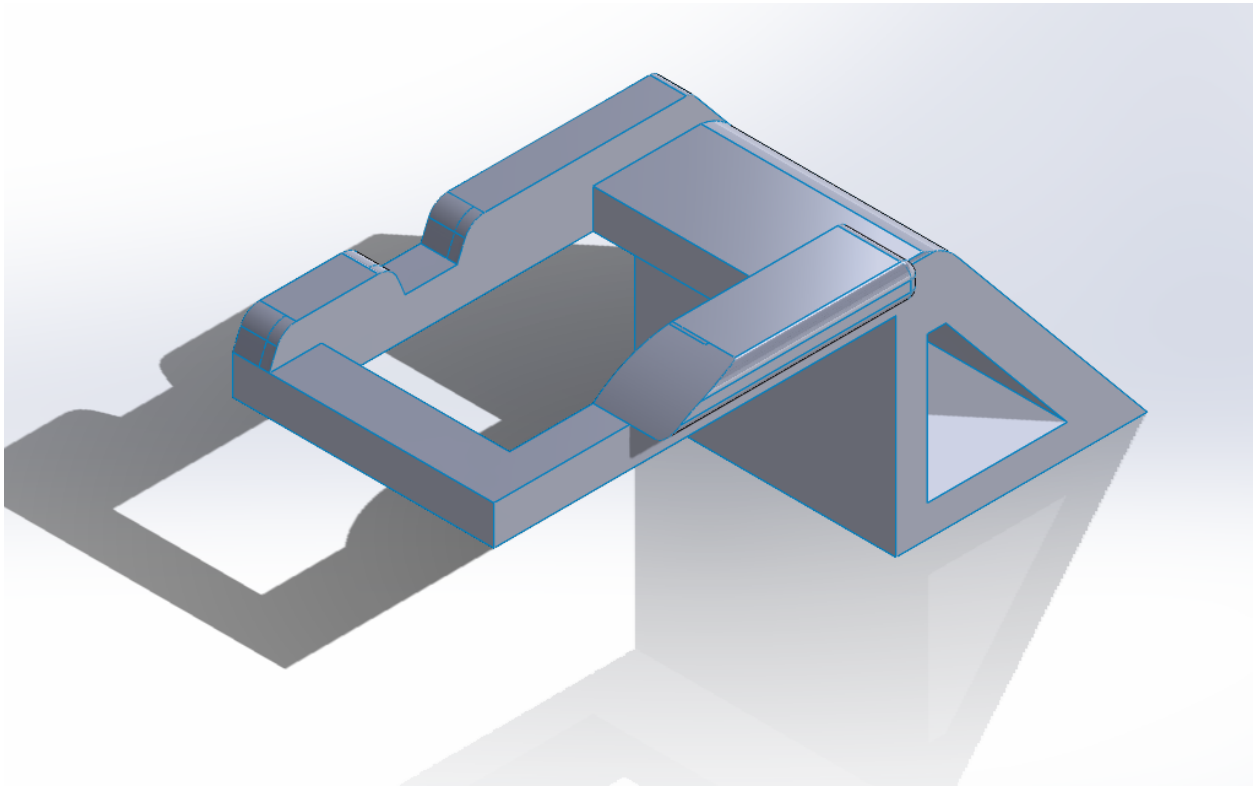
A total of 10 substrates were cut and glued from orange acrylic. The smaller pieces of the substrate were made larger to reduce movement during use and to increase the surface area on which printing can occur.

3. Glass Support

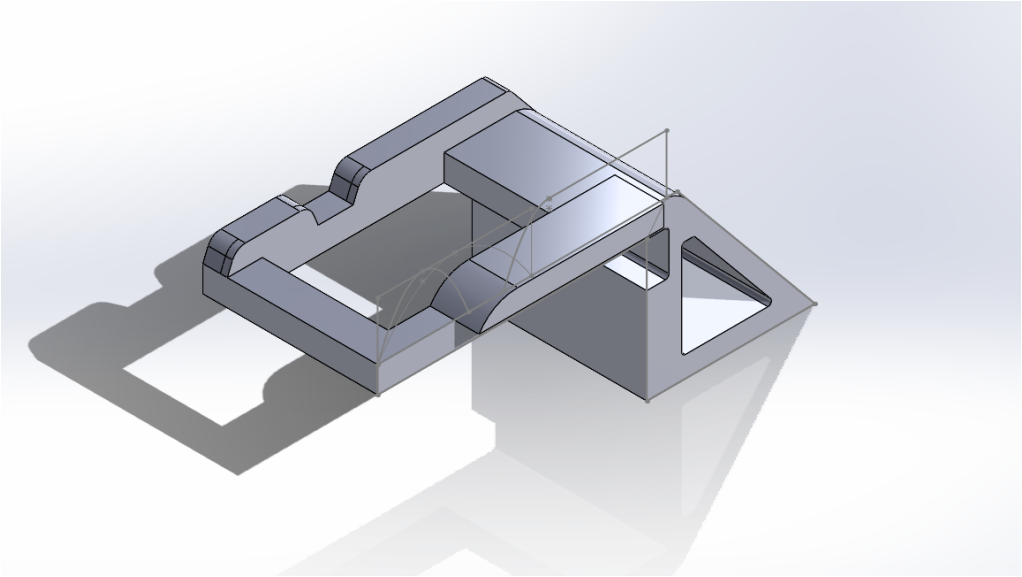
The design for the support piece under the glass was iterated on. The main use of this piece is to bring the glass up such that it is flush with the printer plate and to support it from the bottom while the laser is being run.

Laser support iterations:

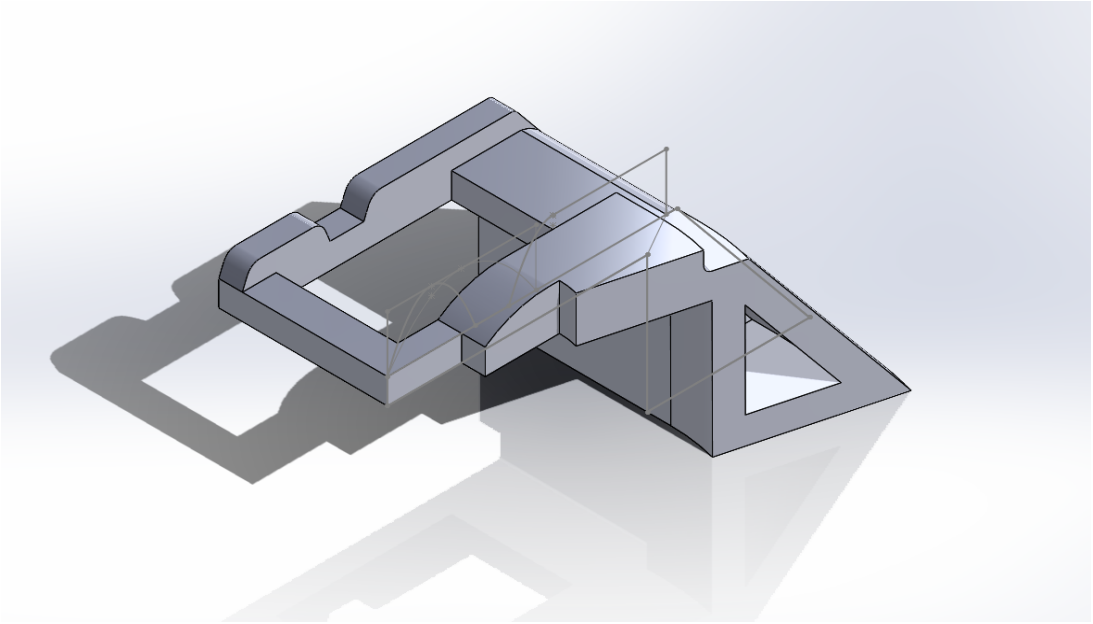
Original print:



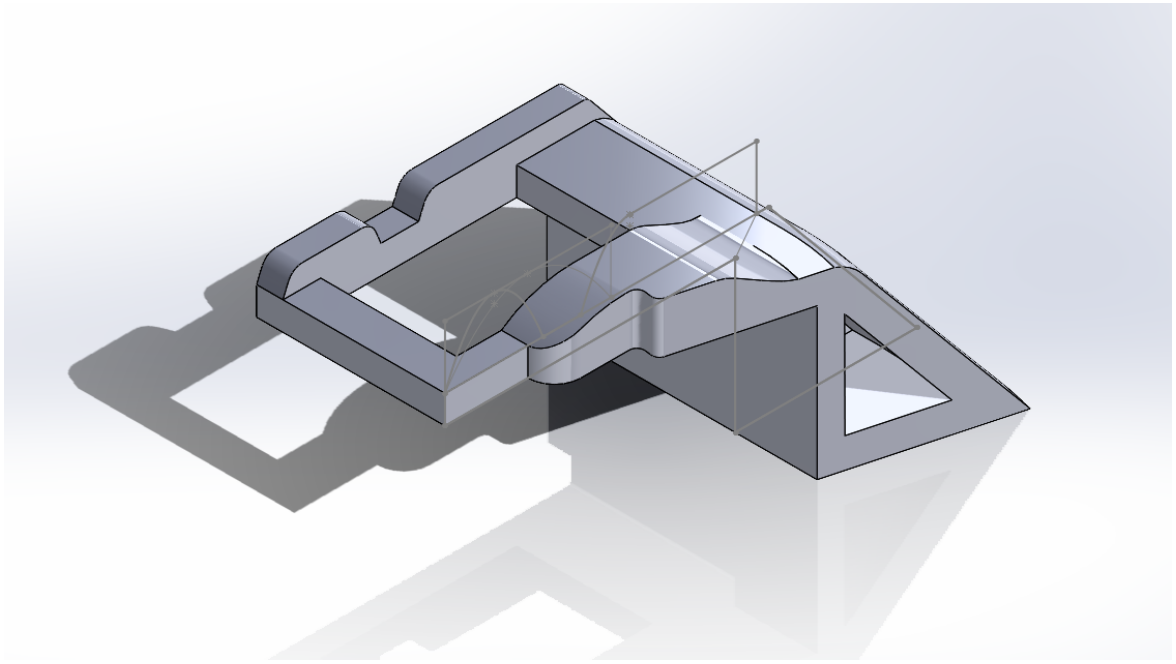
Edit 1:
The goal was to add accommodation for some bolt heads and to increase the amount of material supporting the glass on the right side of the piece (inner edge).
Results: bolt heads were no longer an issue. There was still an issue with catching the hinge on the ramp-side edge of the support structure.

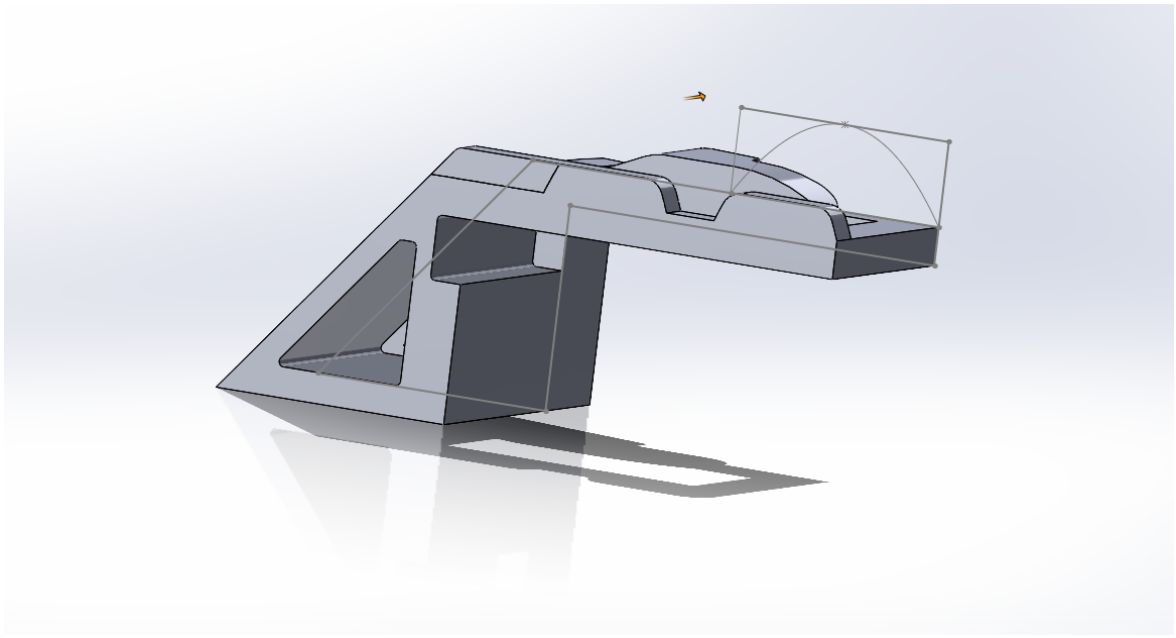


Edit 2: Attempted to match the ramp to the curvature of the plate to prevent the glass from catching on the bottom inner corner.
Results: Glass no longer caught on the corner.

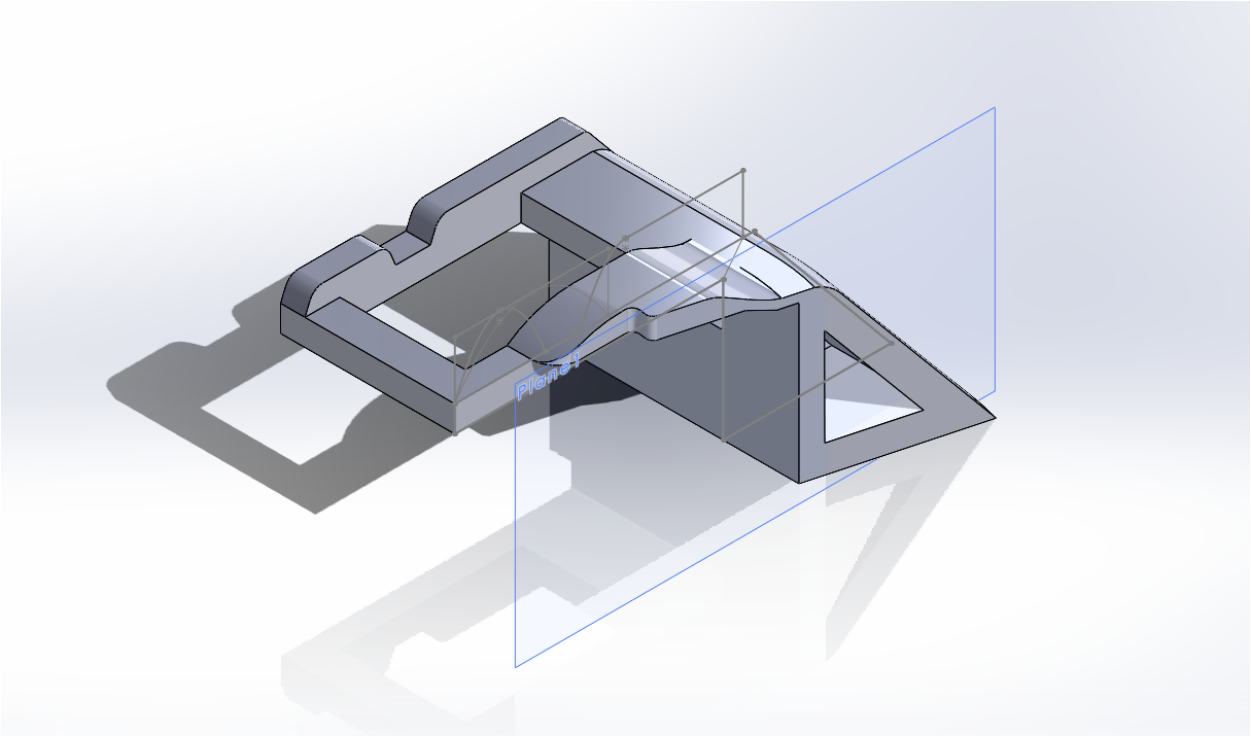


Edit 3: Added smooth surfaces to prevent hinge catching. There was a new issue of falling off due to relying on friction to stay on the main structure.





Edit 4: Previous iteration had a solid piece that interfered with bolts on the top of the mount. This one cuts a portion out to accommodate these bolt holes.



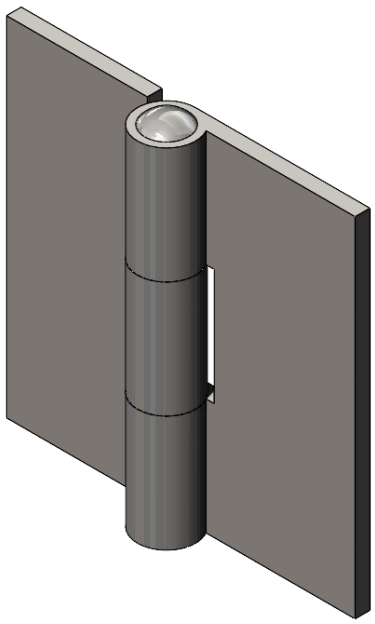
Potential improvement: add support to the edge opposite the ramp to prevent falling.

4. Printer Plate

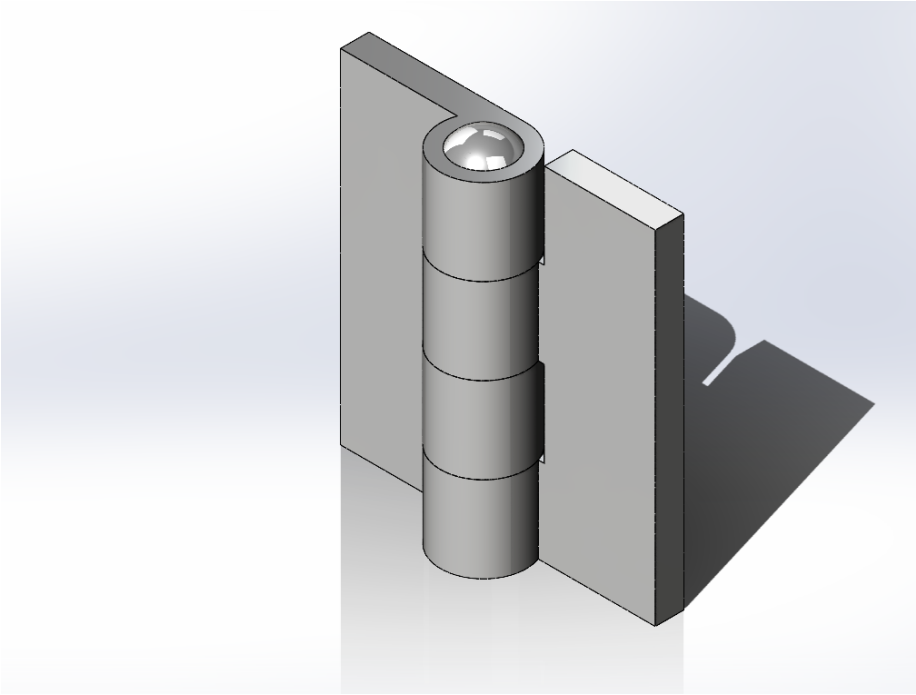
The printer plate was redesigned from scratch to match primary dimensions to the old plate with a few improvements. The main goal of the design change was to find a way to mount the hinge such that it did not protrude from the bottom of the plate. This was accomplished by attaching the hinge to a small piece of plastic that is itself attached to the top of the plate. The second design change was to use a thermal insert instead of tape to mark a full rotation of the plate.

Before modeling the new plate, a new hinge had to be chosen. We debated between plastic and metal hinges, with plastic being easier to bond to acrylic and 3D printed pieces, but metal being more robust and smaller.

Metal hinge McMaster

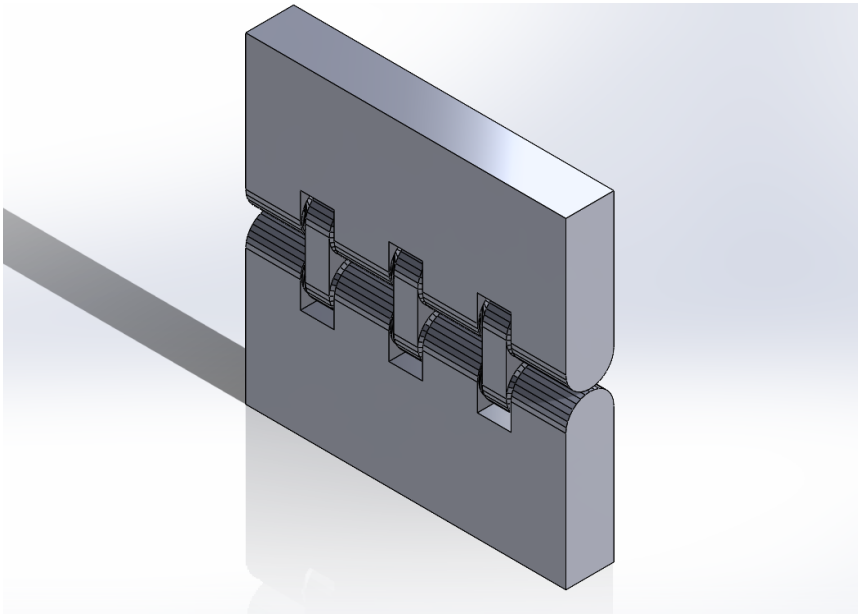


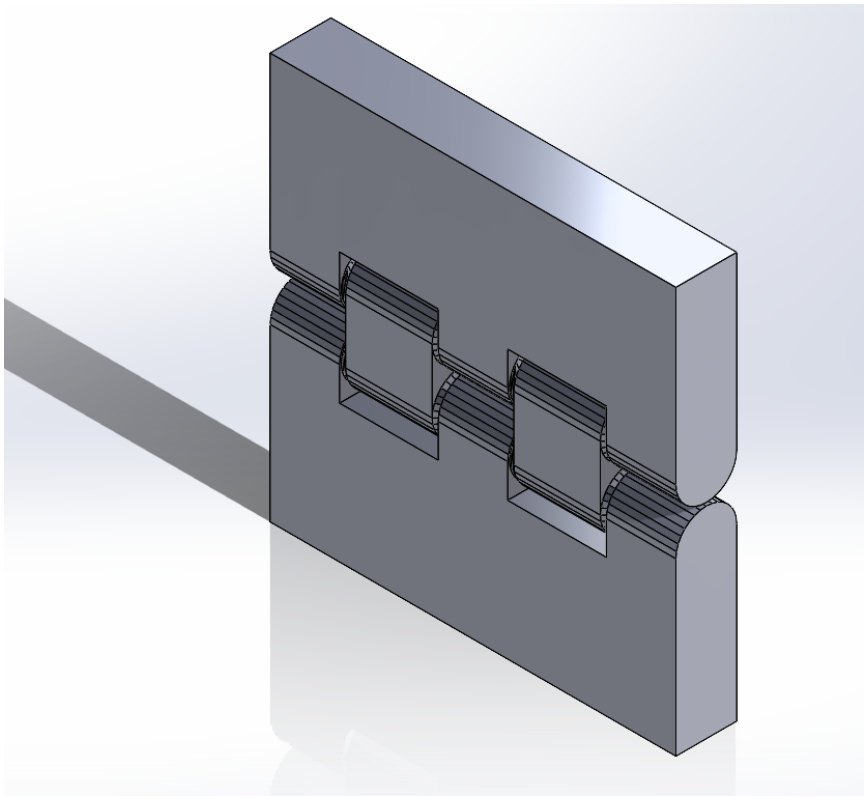
Plastic hinge McMaster:

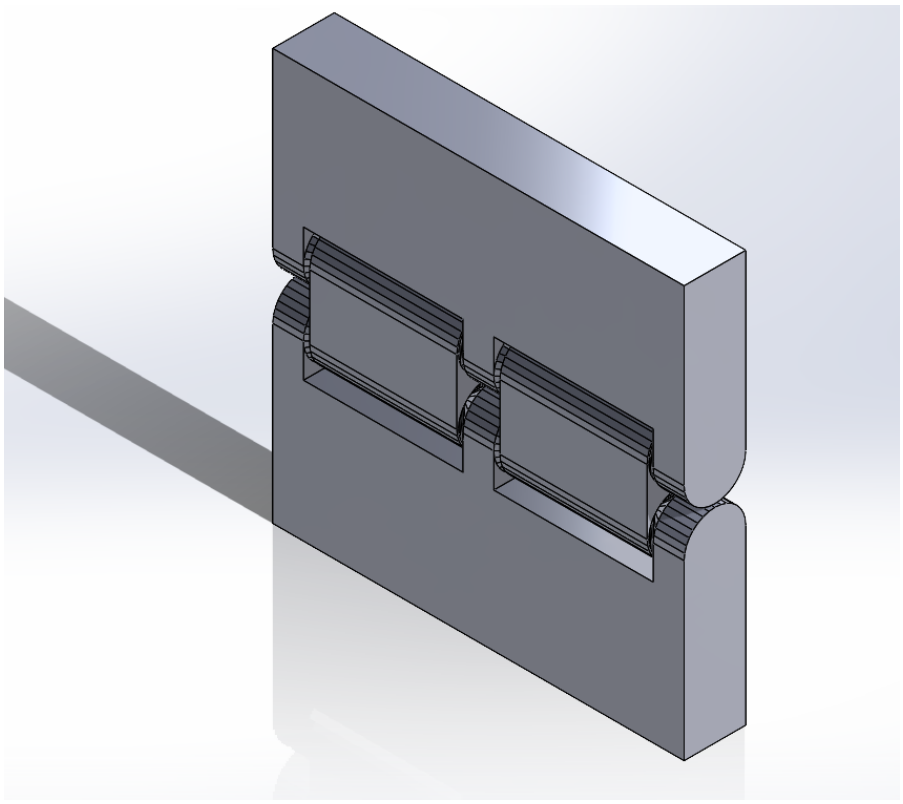


After not being able to find sufficiently small hinges on McMaster, we opted for 3D printed hinges which could be more easily redesigned.

Printable Hinges:

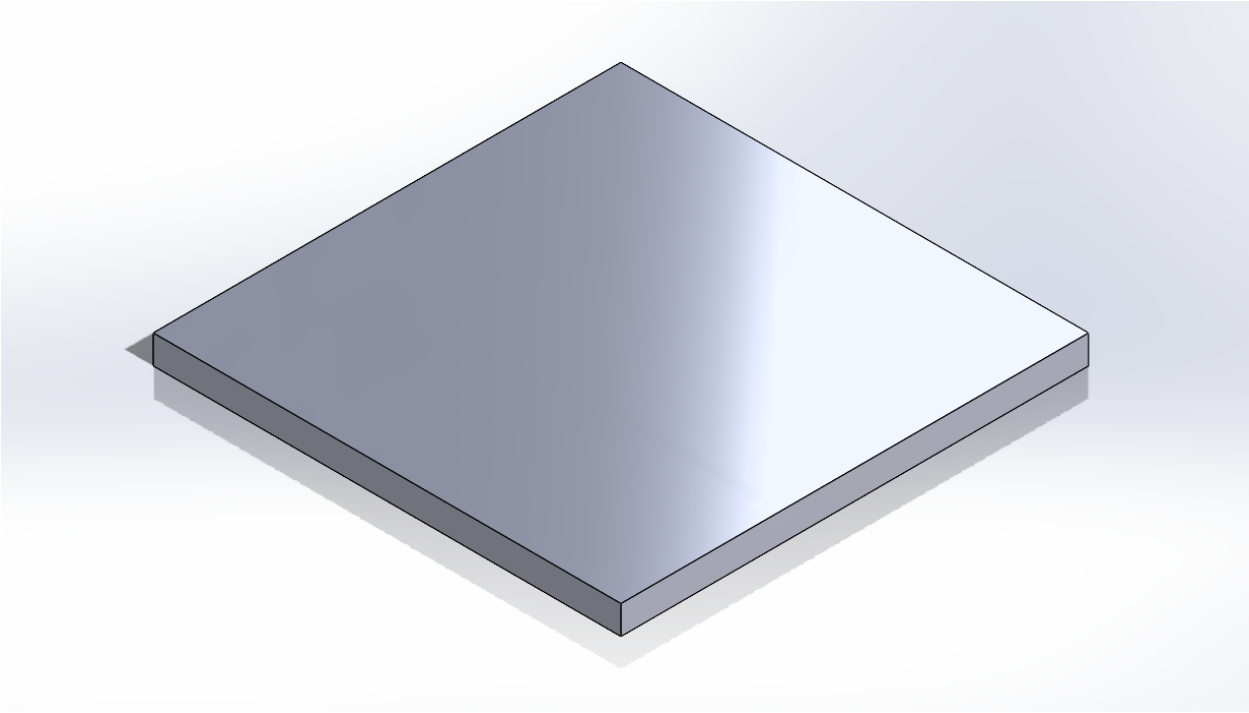




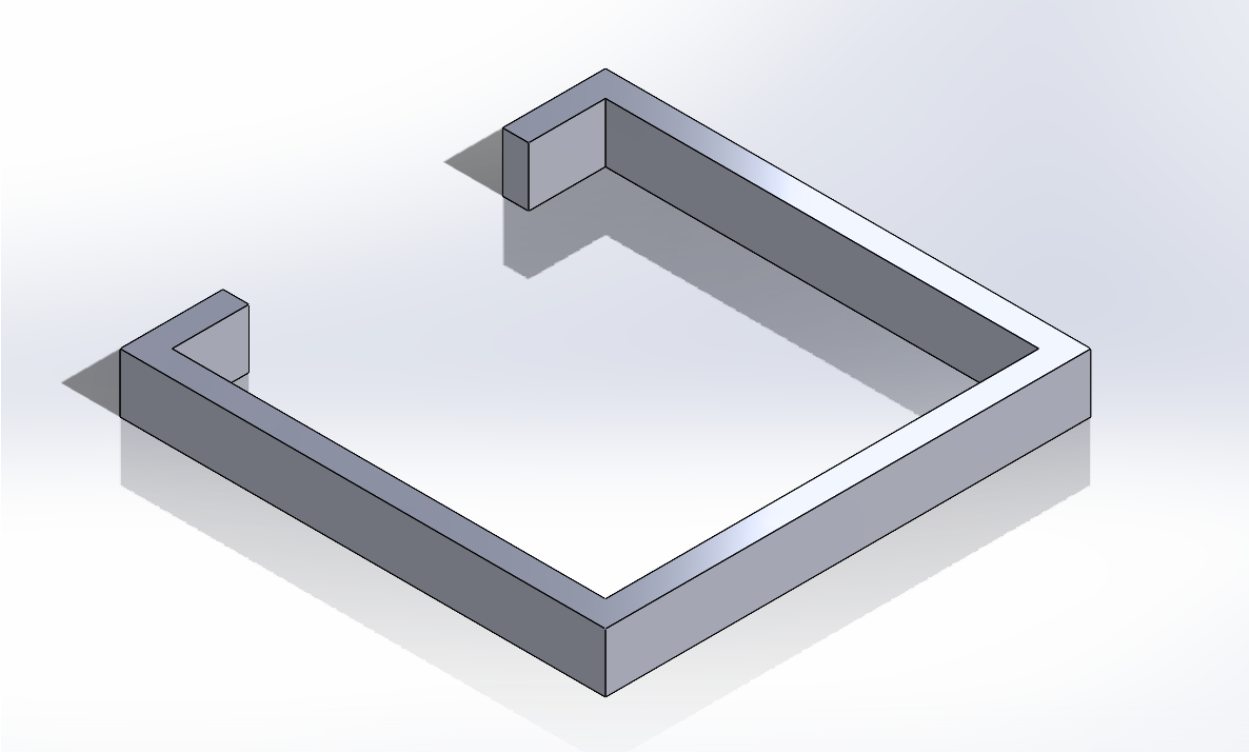


Scaled down to a width of 4 mm before printing.

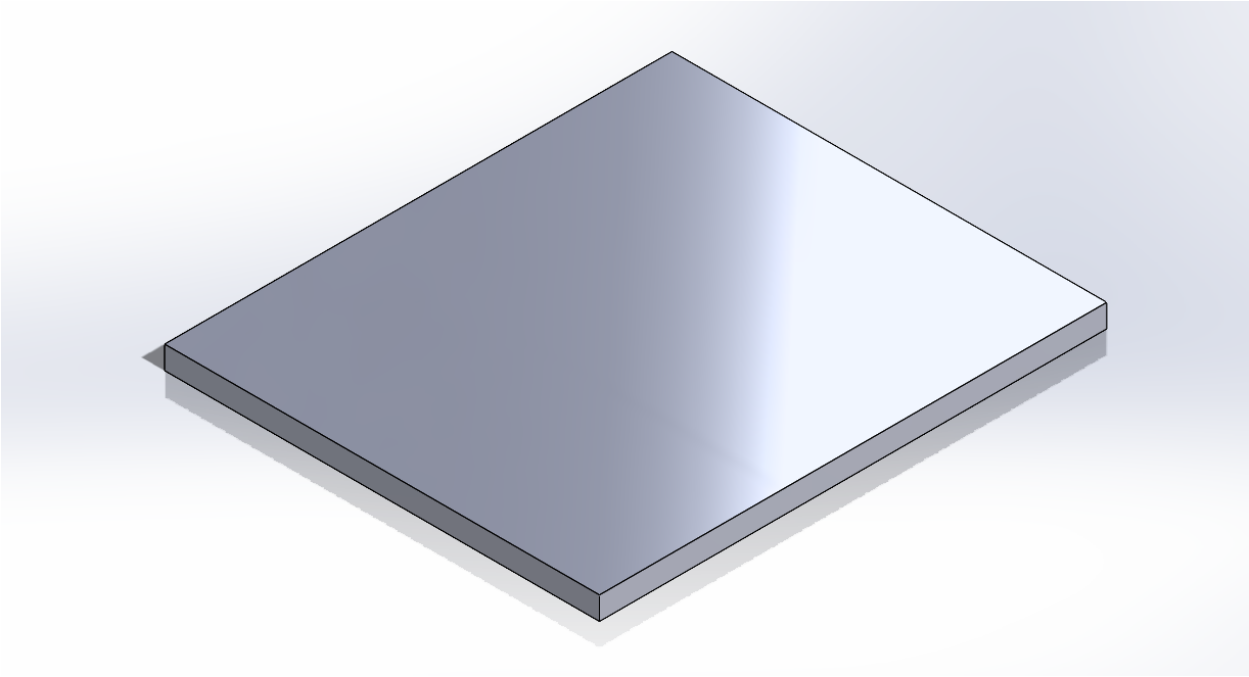
Glass model:



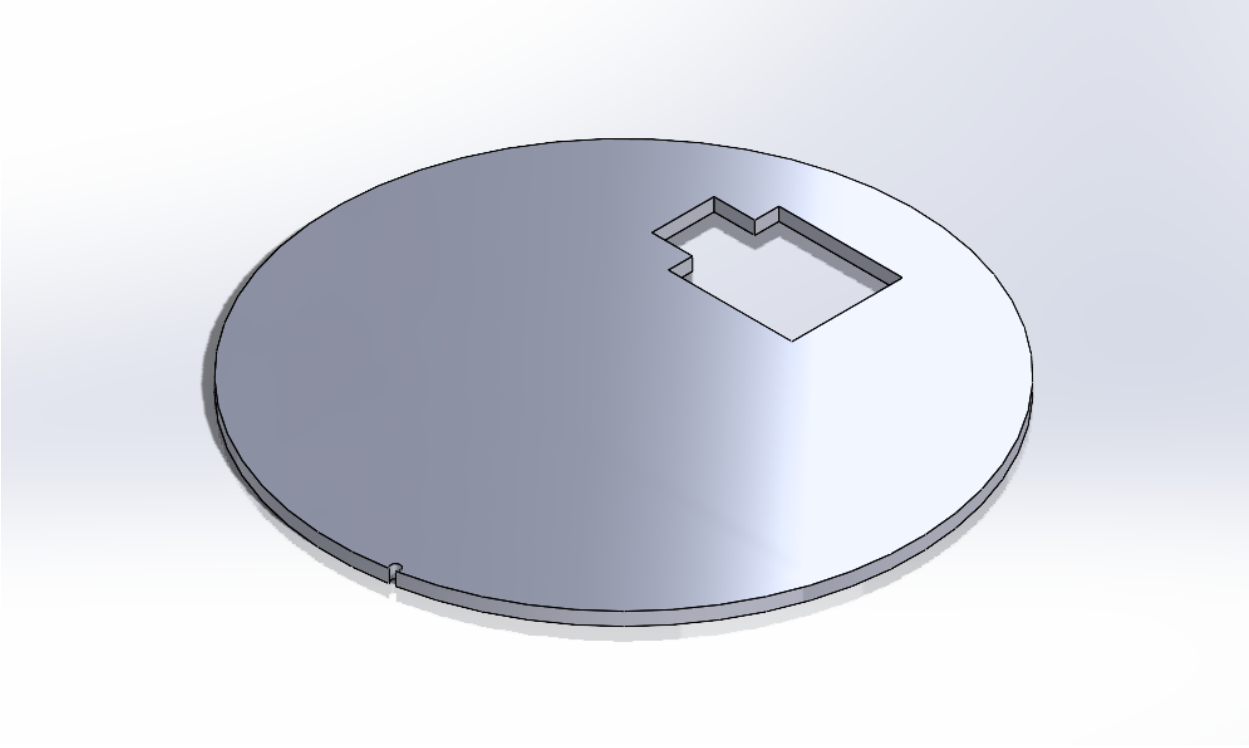
Glass attachment: Laser cut



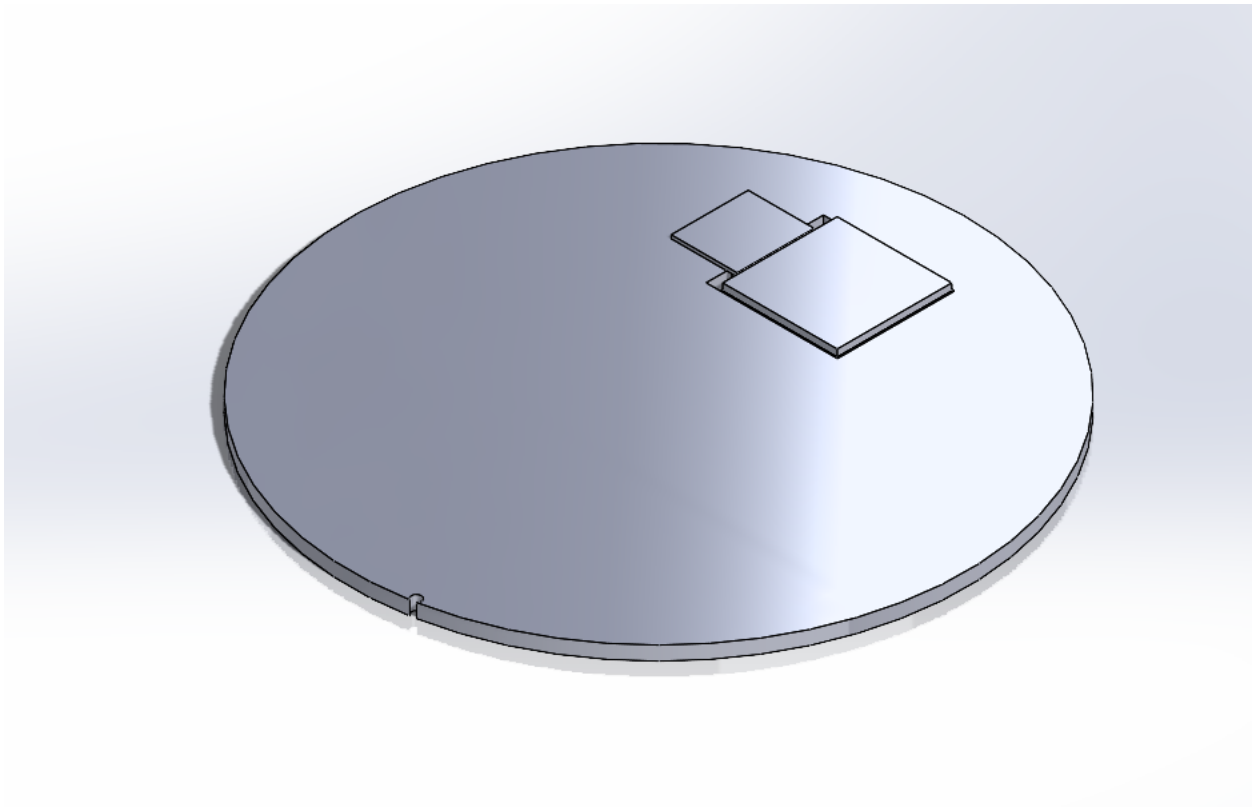
1/16 in attachment plate: 3D printed

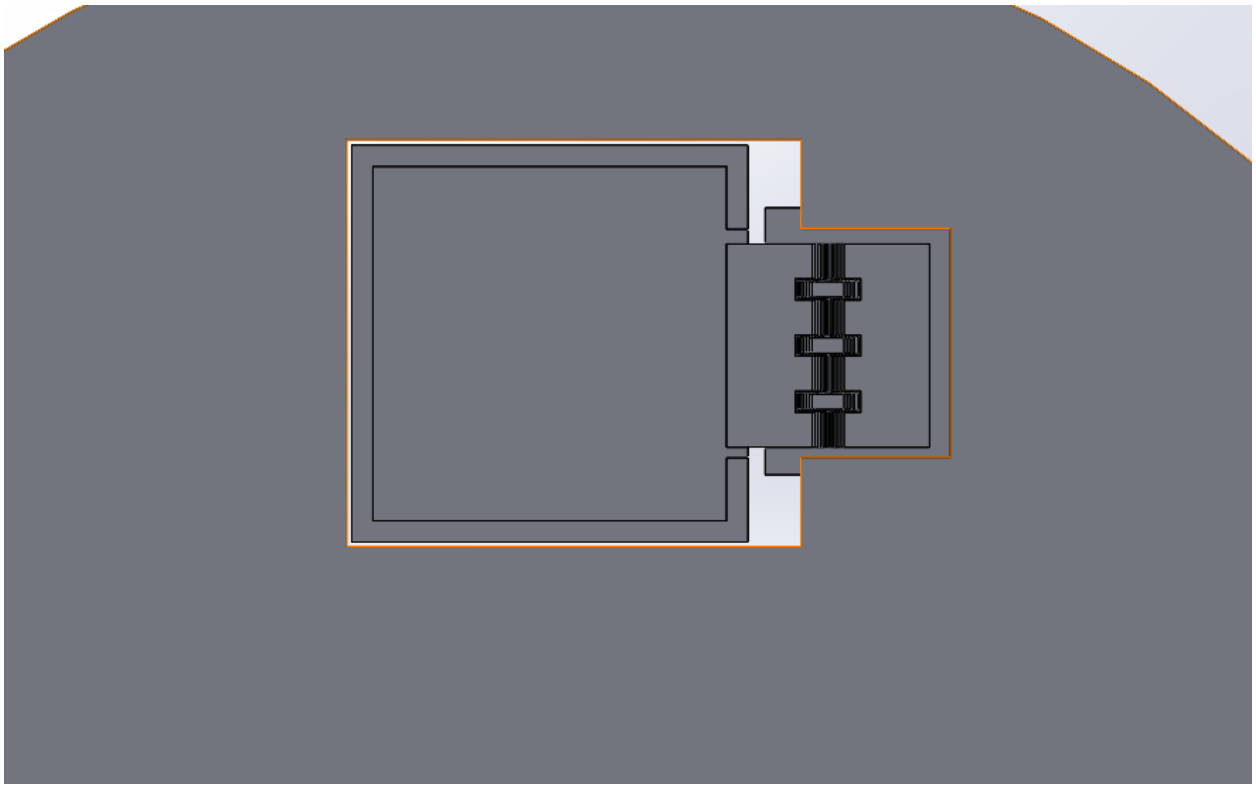


Base Plate:

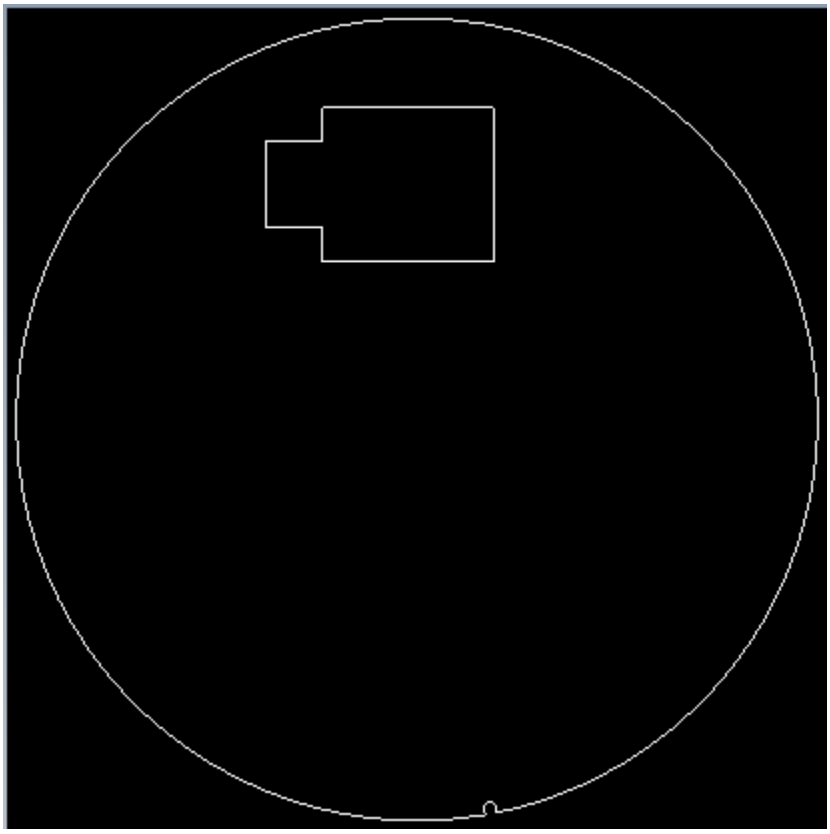


Assembly:





Lasercut dxf: This file was changed before lasercutting to cut the glass support piece out of the same material as the printer plate.



Upon assembly, I realized that I messed up on the glass attachment, making it too large for the glass which inhibited the rotation of the hinge. This could be fixed by cutting the glass attachment on the current iteration. On future iterations, a small change can be made to the lasercut.

5. References

Last Minute Engineers. “In-Depth: Control Stepper Motor with A4988 Driver Module & Arduino.” *Last Minute Engineers*, Last Minute Engineers, 25 Oct. 2022, https://lastminuteengineers.com/a4988-stepper-motor-driver-arduino-tutorial/.

McCauley, Mike. “AccelStepper Library for Arduino.” *Accelstepper*, AirSpayce, 31 Oct. 2022, https://www.airspayce.com/mikem/arduino/AccelStepper/.

Santos, Sara, and Rui Santos. “Arduino with Load Cell and HX711 Amplifier (Digital Scale).” *Random Nerd Tutorials*, 4 Sept. 2022, https://randomnerdtutorials.com/arduino-load-cell-hx711/.