

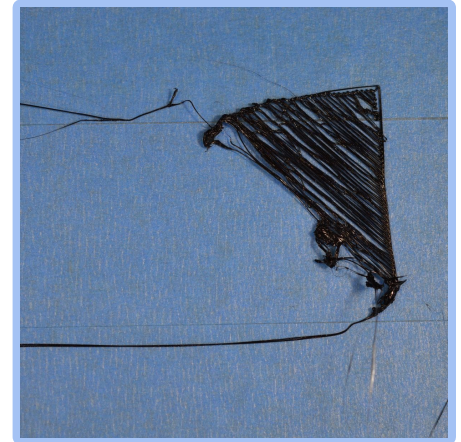
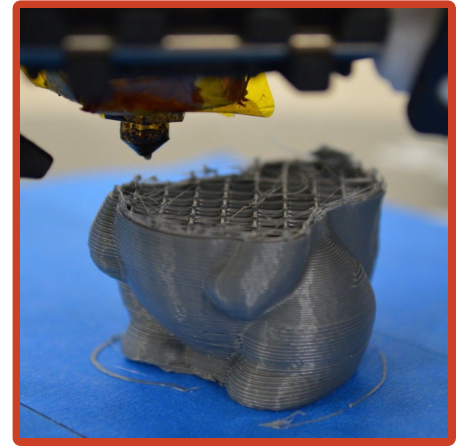
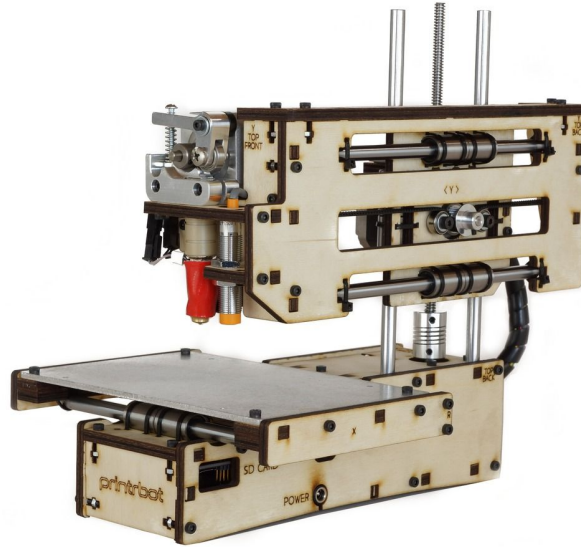
# 3D Printing Error Detection System

Team E1

Lucas Moiseyev, Hannah Preston, Joshua Bas

# Project Summary - Rescoped

- Monitor active 3D prints, detecting errors as they occur, and alert users of potential errors
- Errors to Detect:
  - Extrusion stops mid-print
  - Failing to adhere to the print bed
- Target Printer: PrintrBot Simple Wood

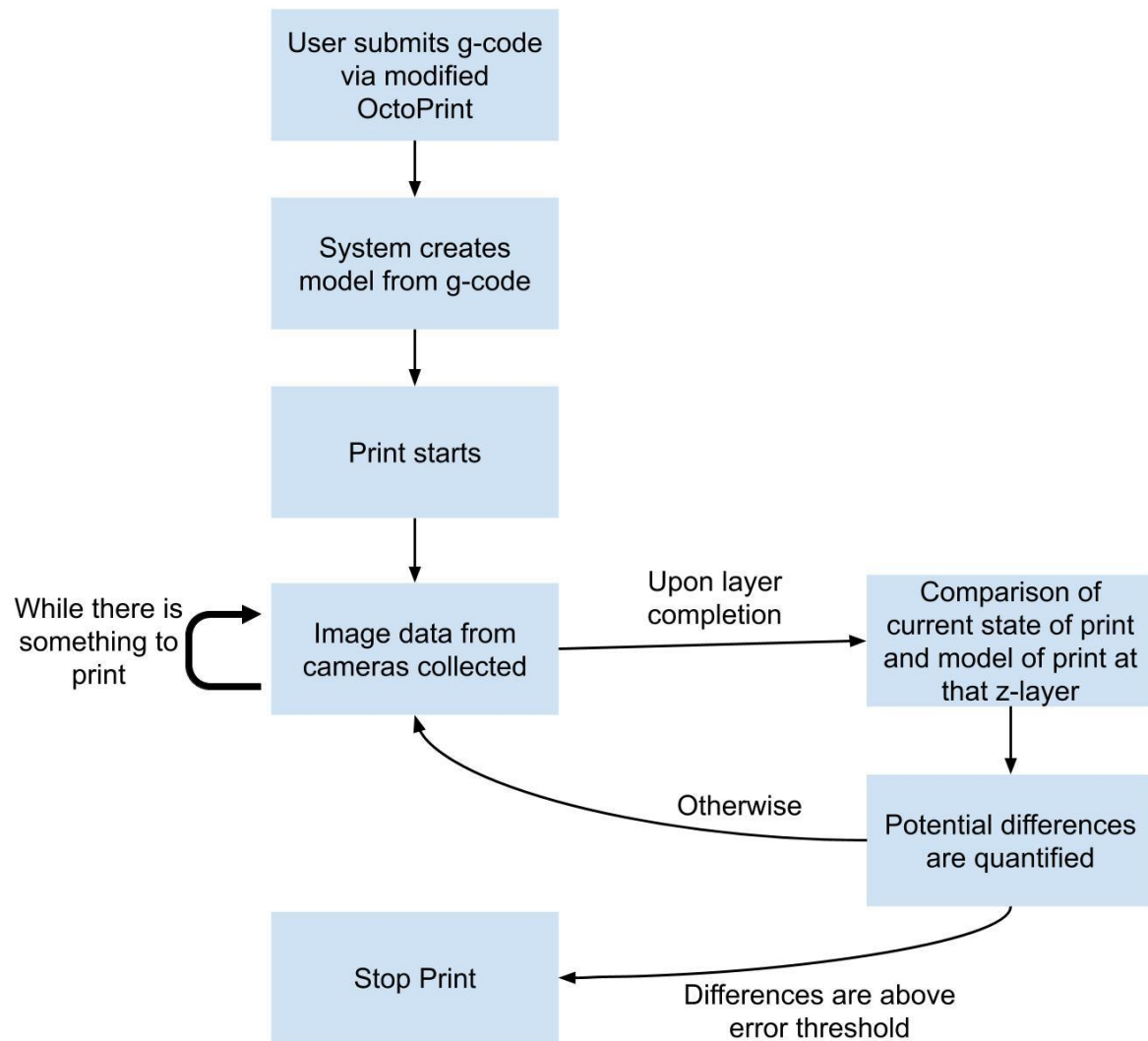


# System Requirements Review

01	Error Check Rate	<ul style="list-style-type: none"><li>• Calculate when a layer should be completed</li><li>• Check on layer completion</li><li>• Otherwise, check every second</li></ul>
02	Error Detection Rate	<ul style="list-style-type: none"><li>• Detected within 10 checks (~1mm)</li></ul>
03	Error Detection Accuracy (Average)	<ul style="list-style-type: none"><li>• 85% accurate</li></ul>
04	False Positive Rate	<ul style="list-style-type: none"><li>• 20% of each detected error is actually not an error</li></ul>
05	False Negative Rate	<ul style="list-style-type: none"><li>• 10% of each real error is not detected</li></ul>
06	Runtime	<ul style="list-style-type: none"><li>• Must run at least 6 hours uninterrupted</li></ul>
07	Size & Weight	<ul style="list-style-type: none"><li>• Within 6 x 3 inches</li><li>• Weights less than 4lbs</li></ul>
08	Sensor Coverage Region	<ul style="list-style-type: none"><li>• Covers a 8.9L x 6.7W x 6.7H inch space</li></ul>

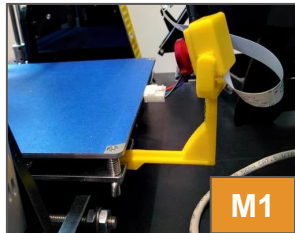
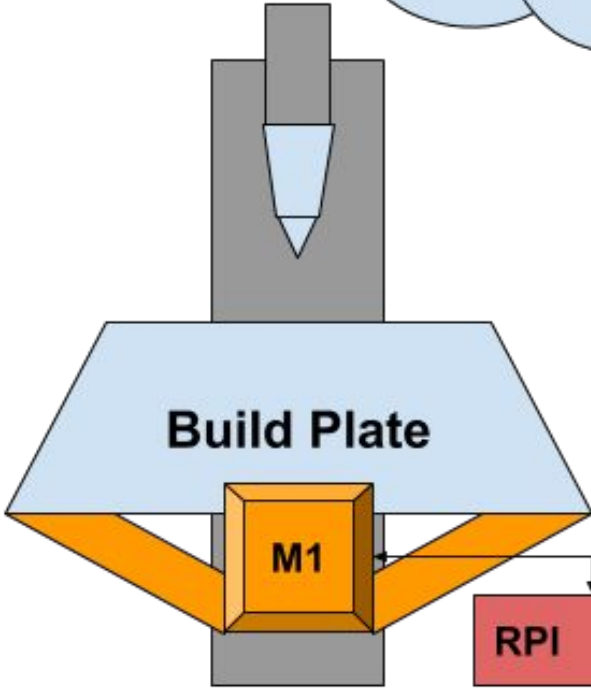
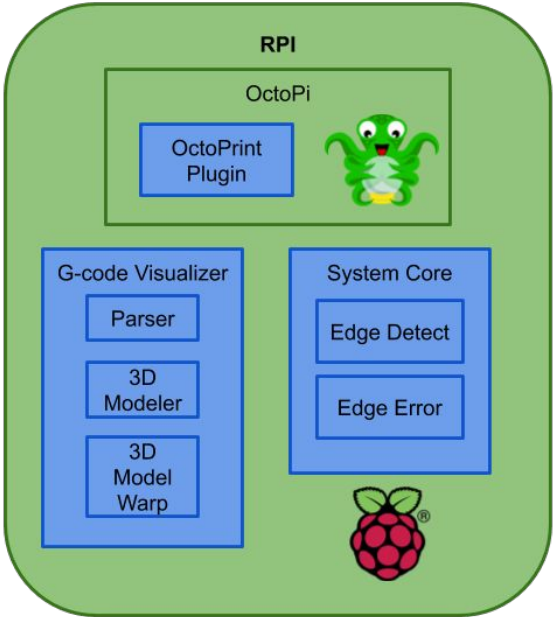
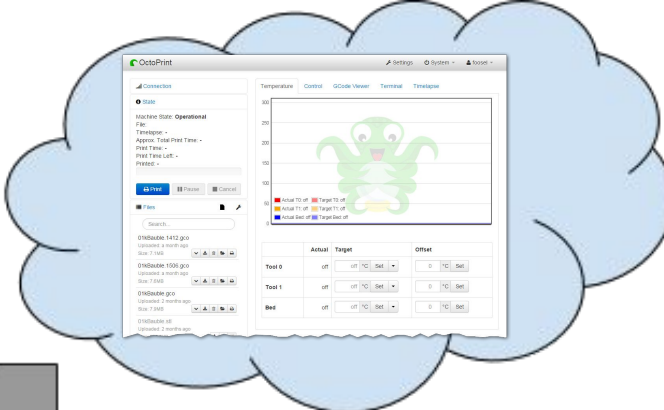
# Process Flow

- From g-code of print, we create 3d models of the print at each different z-layer
- At the end of printing a layer, the camera takes a photo of the current state
- Compare the two images to check for errors



# Solution - Overall System:

- Descoped to a single custom mount for the raspberry pi camera: M1
- Raspberry Pi runs OctoPi (a custom version of the Raspbian OS without a desktop)
- Connects to 3D printer via microUSB
- Connects to OctoPrint web command through WIFI

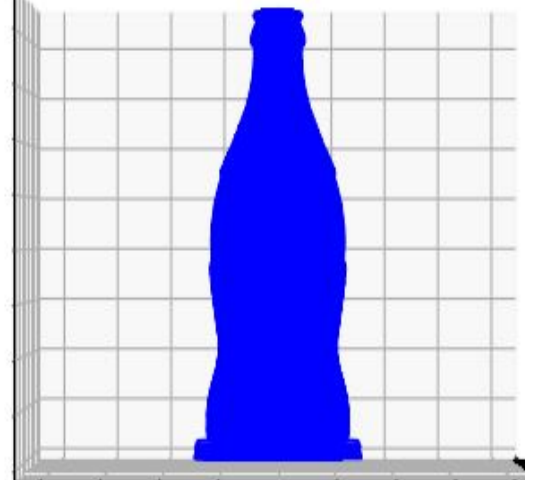
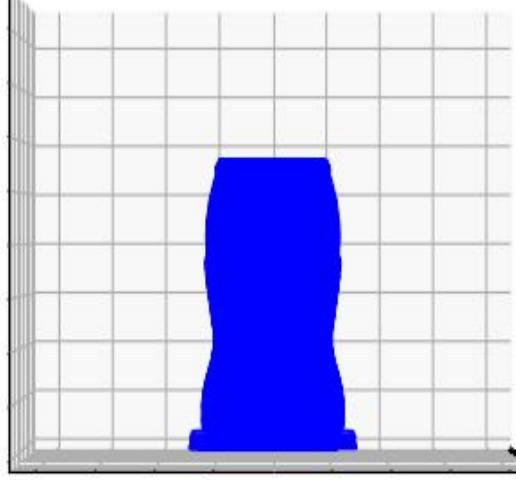
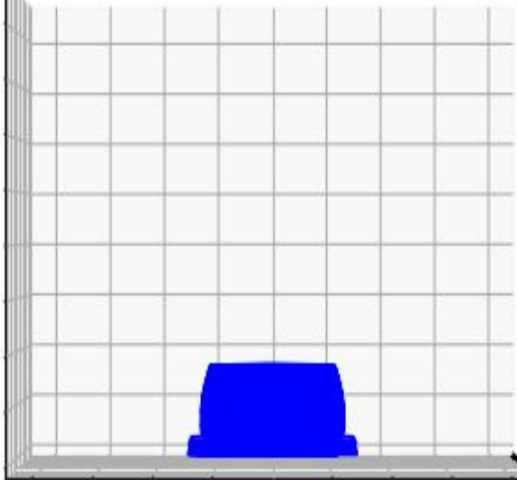


**Key**

Software	Off-the-Shelf
Hardware	Newly Designed

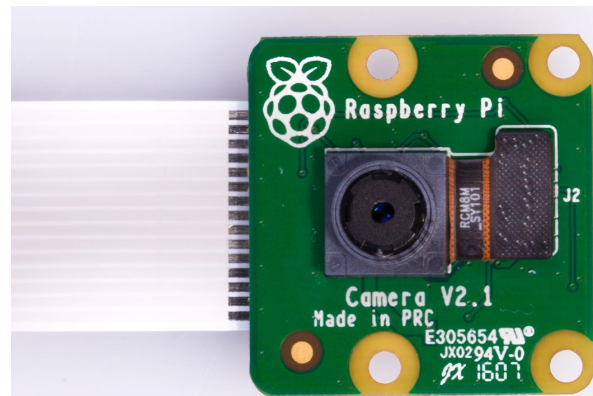
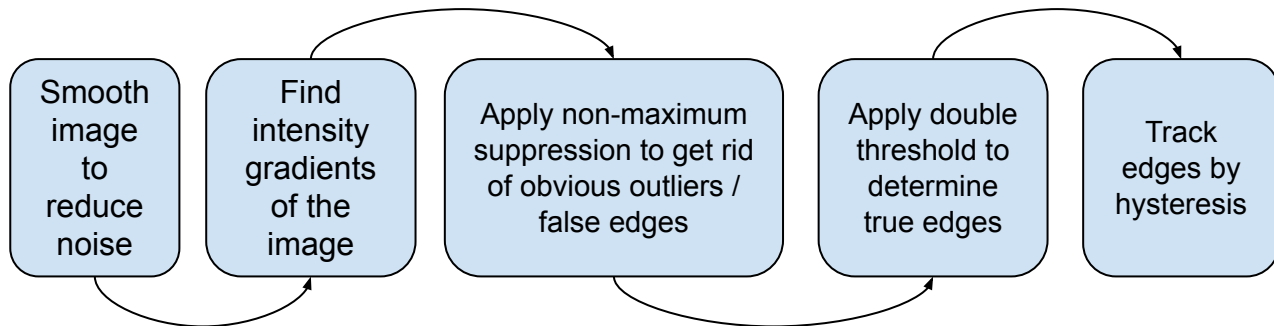
# Solution - G-Code Modeling

- Parser combs through g-code for movement commands (G0 and G1) and separates them into x, y, and z coordinate arrays
- Plot rotated model at different layers
- Mask out grid lines from python's plotter



# Solution - Edge and Error Detection:

- Match corresponding points between image and 3D model
- Project 3D model to image plane
- Find edges on image and projected 3D model
- Compare using Hausdorff distance
  - Using this distance metric because it is suitable for template matching



# A Series of Unfortunate Events:

- Printer broke
  - Design fault on Rev F4 PrinrBot boards
  - Mosfet input takes 12V without any resistor in between
- Laptop exploded
- Printer broke...again

Your computer restarted because of a problem. Press a key or wait a few seconds to continue starting up.

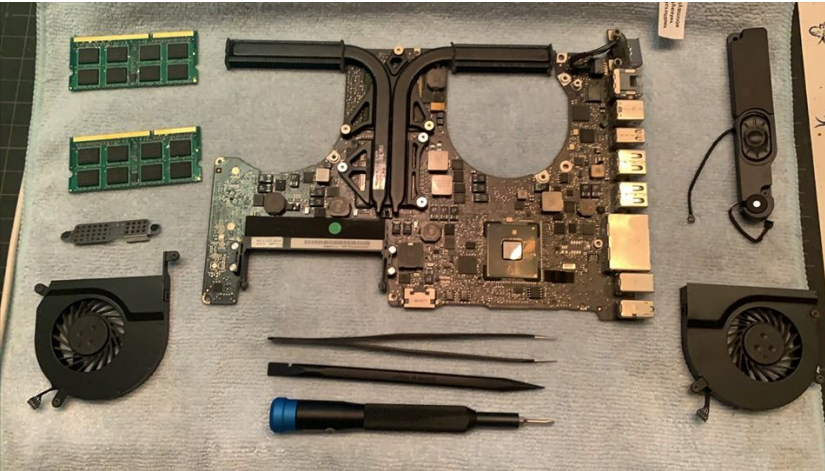
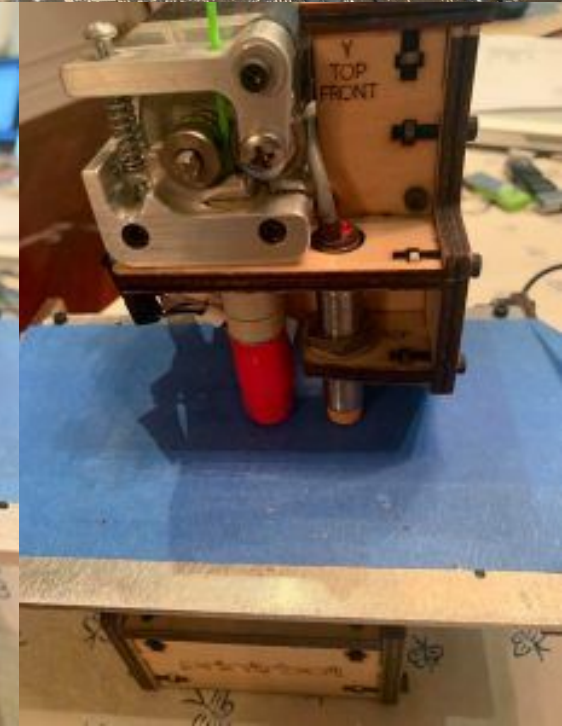
Votre ordinateur a redémarré en raison d'un problème. Pour poursuivre le redémarrage, appuyez sur une touche ou patientez quelques secondes.

El ordenador se ha reiniciado debido a un problema. Para continuar con el arranque, pulse cualquier tecla o espere unos segundos.

Ihr Computer wurde aufgrund eines Problems neu gestartet. Drücken Sie zum Fortfahren eine Taste oder warten Sie einige Sekunden.

問題が起きたためコンピュータを再起動しました。そのまま起動する場合は、いずれかのキーを押すか、数秒間そのままお待ちください。

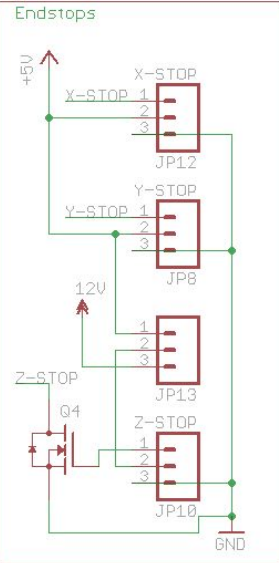
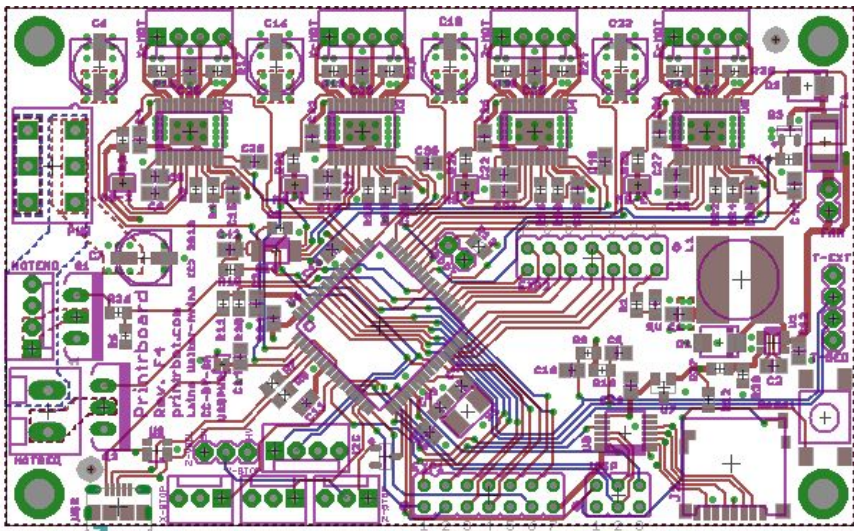
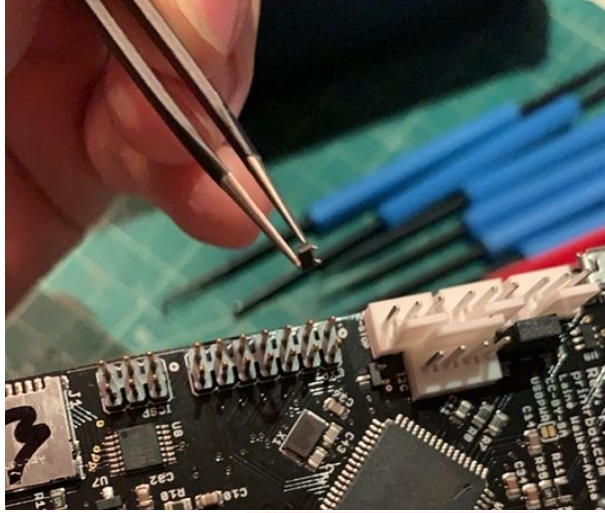
电脑因出现问题而重新启动。请按一下按键，或等几秒钟以继续启动。





# Fixing the Printer:

- SOP-23-3 package NPN BJT replaced the mosfet
- Original Mosfet was internally shorted
- Probe not producing enough voltage to flip BJT



# Design Trade Offs

- Software:
  - Point Cloud vs. Blob vs. Edge Detection
- Hardware:
  - Stereo cameras vs. single camera
  - TOF vs no TOF
- Design Goal:
  - Generalized vs. Specific target printer

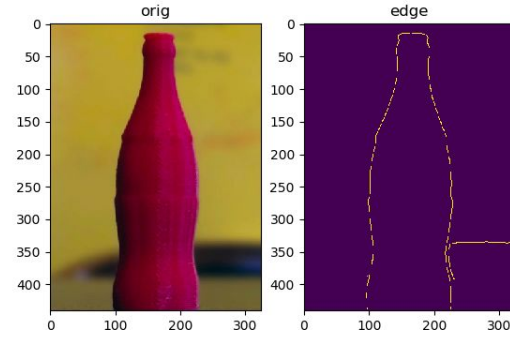
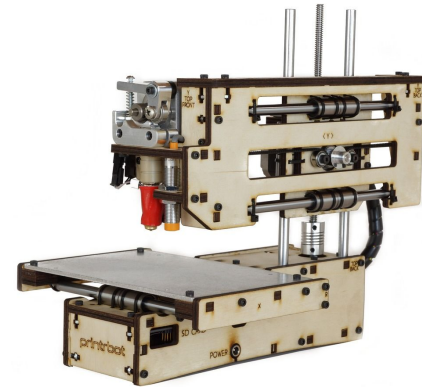
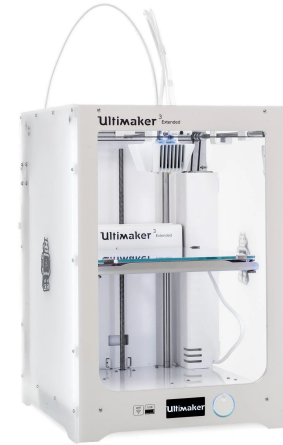
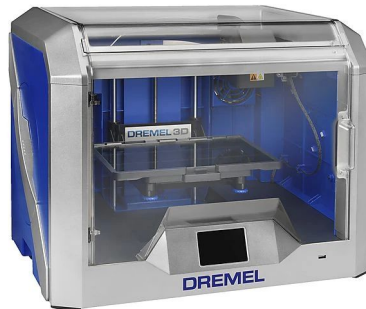
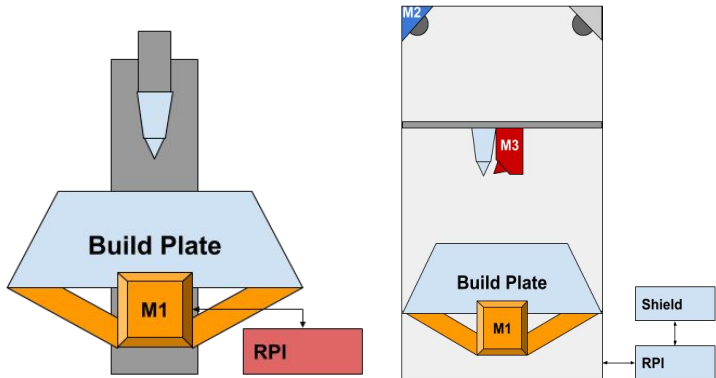


Figure 5. Extracted blob from blue object on printing-bed.



Figure 6. Blue object on printing-bed.



# Metrics and Validation Plan and Results

- Method: Programmatic Error
  - Load faulty g-code into parser / renderer and get series of images
  - Load correct g-code into parser / renderer and get series of images
  - Compare the two using error detector to see if errors are caught

