Print-a-Bot *MakinaMasters admission pool*

"Beyond the virtual: From code to component." Version: 1.0

Contents

Preamble

The digital world is one of pure logic. The physical world is one of constraints, tolerances, and imperfections. This project serves as the bridge between them.

You have learned to instruct a machine with code. Now, you will instruct a machine to create a physical object. This is not a project about creating a beautiful model; it is a project about precision, function, and **Design for Manufacturing (DfM)**.

You will move from downloading a test file to designing, slicing, and printing a functional component for a robotics project. Your goal is not just to print, but to print *correctly*, so that your part fits a real-world object.

General Instructions

- This project must be done individually.
- You must use an FDM (Fused Deposition Modeling) printer.
- The mandatory material for this project is **PLA** or **PETG**.
- You must use **Cura** or **PrusaSlicer** as your slicing software.
- Using a pre-made 3D model (e.g., from Thingiverse, Printables) for the final mandatory part is strictly forbidden. The design must be your own.
- You must turn in a git repository containing all required files.
- Failure to adhere to these rules will result in a grade of 0.

Mandatory Part

This project is divided into three exercises. Each must be completed perfectly to proceed to the next.

Exercise 00: The Calibrator

- **Goal:** Achieve dimensional accuracy.
- **Task:**
  1. Download a standard **20mm XYZ Calibration Cube**.
  2. Slice and print the cube.
  3. Using digital calipers, measure the X, Y, and Z dimensions.
  4. **Your goal:** All dimensions must be within ± 0.2mm of 20.00mm.
- **To Turn In:**
  - A folder named `ex00/`
  - The `.stl` file used.
  - The final `.gcode` file used to print the accurate cube.

Exercise 01: The Slicer

- **Goal:** Demonstrate control over the "compiler" (slicer).
- **Task:**
  1. Take the same `calibration_cube.stl` from Exercise 00.
  2. You must create three distinct slicing profiles and generate G-code for each.
  3. **Profile 1: "Draft"**
     - Layer Height: 0.28mm or higher
     - Infill: 10% (Grid or Lines)
     - Walls: 2
  4. **Profile 2: "Structural"**
     - Layer Height: 0.20mm
     - Infill: 40% (Gyroid or Cubic)
     - Walls: 4
  5. **Profile 3: "Support Test"**
     - Find a model specifically for testing supports (e.g., an "Overhang Test").
     - Slice it with "tree" or "organic" supports enabled.
- **To Turn In:**
  - A folder named `ex01/`
  - Sub-folders `draft/`, `structural/`, and `support_test/`.
  - Each folder must contain the `.gcode` file and a screenshot of the slicer settings used.

Exercise 02: The Component

- **Goal:** Design and print a functional robotic component.
- **Component:** A mount for an **HC-SR04 Ultrasonic Sensor**.
- **Task:**
  1. Obtain an HC-SR04 sensor and digital calipers.
  2. Measure the sensor's body, the two "eyes" (transducer and receiver), and the 4-pin header. Pay special attention to the diameter of the "eyes" and the distance between them.
  3. Using a design software of your choice, design a mount.
  4. **Design Constraints:**
     - The sensor must be held securely *only* by the "eyes" (a friction fit).
     - The 4-pin header on the back must be fully accessible.
     - You must account for **tolerance**. A 4mm peg does not fit in a 4.00mm hole. You must design your holes appropriately (e.g., 4.2mm).
  5. Slice the part (use your "Structural" profile as a base) and print it.
  6. **Test:** The sensor must fit snugly without glue. If it fails, *iterate your design*, not your slicer settings (e.g., do not use "horizontal expansion").
- **To Turn In:**
  - A folder named `ex02/`
  - Your CAD files (e.g., `.f3d`, `.step`)
  - The final, printable `.stl` and `.gcode` files.
  - A `README.md` detailing your design choices, specifically the tolerances you used for the sensor "eyes".
  - A photo named `assembly.jpg` showing the sensor successfully fitted into your printed part.

Bonus Part

- **Bonus 00: Motor Mount**
  - Design and print a functional mount for a standard **SG90 micro-servo** *or* a **NEMA 17 stepper motor**.
  - This is more complex than the sensor mount as it must be dimensionally accurate to fit the motor, and robust enough to handle potential torque and vibration.
  - **To Turn In:** A new folder `bonus00/` with all CAD, `.stl`, `.gcode`, and photo proof of the motor fitted in the mount.
- **Bonus 01: The Assembly**
  - **(Requires Bonus 00)**
  - Design and print an adapter that connects your `Exercise 02` sensor mount to the motor from `Bonus 00`.
  - The goal is to create a functional 1-axis "pan" or "tilt" mechanism for the sensor.
  - You will need to precisely measure the servo horn (if using an SG90) or the motor's mounting pattern.
  - **To Turn In:** A new folder `bonus01/` with all CAD/STL/G-code files for the *adapter part* and a photo (`full_assembly.jpg`) showing the sensor, adapter, and motor assembled.
- **Bonus 02: Captive Hardware**
  - Design and print a test piece that includes a **captive nut** pocket.
  - The design must allow an M3 nut to be inserted (either from the side or pressed in) and prevent it from spinning when a bolt is tightened.
  - This demonstrates an essential skill for creating robust, assemblable robotic parts without using glue.
  - **To Turn In:** A new folder `bonus02/` with the CAD/STL/G-code files and a photo or video showing a bolt successfully tightening into the captive nut.

Submission and Peer-Evaluation

- You will submit your work in a `git` repository.
- During peer-evaluation, you will check:
  - The presence of all required files.
  - The `calibration.txt` for Exercise 00. Is the tolerance met?
  - The screenshots for Exercise 01. Do they show the correct settings?
  - The `README.md` for Exercise 02. Does it explain the tolerances?
  - The `assembly.jpg`. Does the photo show a successful print and assembly? This is the final pass/fail test.
  - You will attempt to load their `.stl` file into a slicer to ensure it is a valid, manifold 3D model.