Functional Recommendations Document

Autodesk Collaborative Control (AKA "Project Escher")
Beta Release
Revision 2.2
161123

REMINDER: This document is property of Autodesk, Inc.

1. Purpose
   1.1. The purpose of this Functional Recommendations Document (FRD) is to define the functional recommendations for hardware and electronics to ensure compatibility with the current releases of Netfabb software, the "Project Escher" features of Netfabb software, and Machine Collaboration Utility (MCU) software.
   1.2. This FRD applies to the current release of Netfabb software and MCU software. We expect printers that comply with this specification to be compatible with future versions of software. However, hardware changes may be required to interface with future software features.

2. Overview of project
   2.1. Collaborative Control is a manufacturing process rooted in FFF/FDM that performs simultaneous and synchronized additive manufacturing processes on a single build platform. Collaborative Control compatible printers use discrete automated manufacturing units, also known as "bots", that can independently move over and along a central build platform. Bots can operate independently, concurrently, or in sequence. When coupled with Netfabb software and Machine Collaboration Utility software, bots can work to collaboratively manufacture a single object or can simultaneously create multiple objects on a build platform.
   2.2. By leveraging parallel processes, the control system coupled with properly configured hardware may be able to manufacture parts many times faster than a traditional 3D printer of equal resolution.
   2.3. The current goal of the system is to print large format objects at increased print speed without loss of resolution.
   2.4. Future releases may add additional functionality.
   2.5. Autodesk to provide software only.
      2.5.1. The control system provided by Autodesk consists of software only.
         2.5.1.1. Slicing/Splicing G-Code Generating software (Netfabb).
         2.5.1.2. MCU Printer Control software (open source).
      2.5.2. Compatible hardware and electronics to be provided by printer manufacturer.

3. Disclaimer
   3.1. The software is currently in Beta.
      3.1.1. This preview is an early version of the system.
      3.1.2. There may be software bugs.
      3.1.3. Functionality may be limited compared to future releases.

4. Build Platform and Build Volume Specifications

4.1. Build Volume

4.1.1. The software is configured for Cartesian build volumes. See Bot Packing recommendations (Section 6.4) for optimal bot width & bot quantity recommendations for build volumes.

4.1.2. Build volumes are not limited to specific dimensions.

4.1.2.1. NOTE: Due to potential slicing issues, problems may arise when slicing very large and complex parts during the BETA phase of the software. Complexity and size is geometry-dependent and therefore not readily quantified.

4.2. Build Platform

4.2.1. Material

4.2.1.1. To be determined by machine manufacturer.

4.2.2. Flatness

4.2.2.1. To be determined by manufacturer.

4.3. Heated Bed

4.3.1. Optional

4.3.2. Controlled independently of Netfabb or MCU.

4.4. Heated Build Chamber

4.4.1. Optional

4.4.2. Controlled independently Netfabb or MCU.

5. Coordinate systems

5.1. Coordinate system definition (Figure 1)

5.1.1. X-axis is parallel to “shared” (long) axis, parallel to build platform.

5.1.2. Y-axis is perpendicular to “shared” (long) axis, parallel to build platform.

5.1.3. Z-axis is perpendicular to build platform.

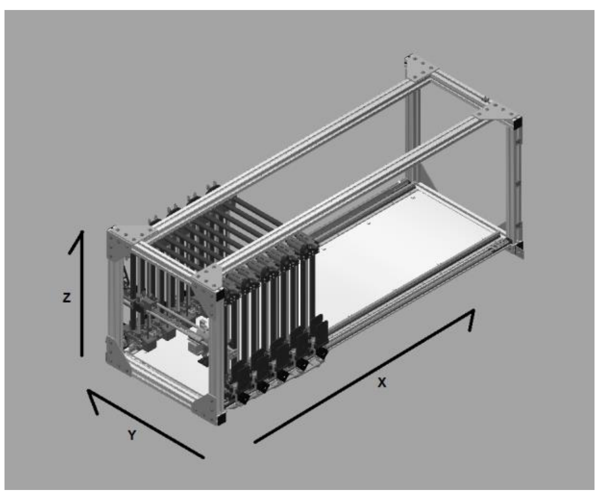

Figure 1 Coordinate System Definition

5.2. Axis signs follow “right hand rule,” with origin at the “front bottom left” of build platform (minimum x, minimum y, minimum z)

6. Bot Specifications

6.1. Definition: A “bot” is a printing gantry or structure with controllable and independent X, Y and Z axes, and Extrude.

6.2. Bot Architecture

6.2.1. The bots should be organized in a “1 x N” or “single-file” architecture with the bots having equal but independent Y and Z extents while independently moving along a shared X extent (Figure 2).

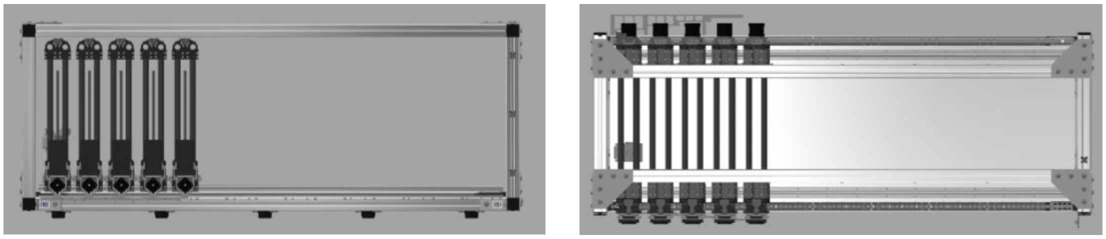

Figure 2 X Side (left) and Top (right) View of Single-File Bot Architecture along X Axis

6.3. Bot Quantity

6.3.1. The system can theoretically accommodate an infinite number of bots as determined by designer.

6.3.1.1. The highest tier of Netfabb is limited to 10 bots.

6.3.1.2. Different tiers of Netfabb subscriptions enable different numbers of bots.

6.3.1.2.1. To enable printing with more than 10 bots please contact the Project Escher team at Autodesk.

6.3.1.2.1.1. EscherTeam@autodesk.com

6.3.1.3. Note that for most part geometries, except long thin parts, 4-6 bots will likely achieve the greatest increase in print speed.

6.3.1.3.1. In many cases there are substantially diminishing returns for adding more bots.

6.4. Bot Packing

6.4.1. Shortest printing time is generally achieved by maximizing bot density (aka “packing”) along the X-axis.

6.4.1.1. Each Bot’s extruders should be able to move as close as possible to adjacent bots’ extruders without colliding.

6.4.1.2. RECOMMENDED: For printers with more than two bots, achieve effective parallelization on full-length prints(full length of the build platform by following the below formula:

6.4.1.2.1. Bot width in the X direction should be less than: (length of the build platform) / (number of bots * 3)

6.4.1.2.2. For example: for a 1500-mm build platform and five (5) bots, each individual effective bot width should be 100mm or less.

6.4.1.2.3. Narrower effective X dimension for bots will improve parallelization

6.5. Bot movement and features

6.5.1. Each Bot should have independent X-Y-Z-Extrude.

6.5.2. Four axis independent movement:

6.5.2.1. Movement must be Cartesian.

6.5.2.2. Current software supports single Extrude axis (single material) per bot.

6.5.3. Zeroing limit switch or sensor for each bot at end of X, Y, Z travel

6.5.3.1. Zeroing limit switch locations.

6.5.3.1.1. X-axis limit switch triggered at the minimum X location for each bot

6.5.3.1.1.1. All bots should be able to home at their respective minimum X positions at the same time.

6.5.3.1.1.2. Minimum X positions (X home) for bots should be separated by a buffer distance between each bot (Figure 3) that is greater than the trigger double-tap sequence for your specific system

6.5.3.1.1.2.1. NOTE: Minimum buffer distance maximizes usable build volume. This can also be accomplished by establishing a “parking” area where no or minimal parked bots extend into the build platform.

6.5.3.1.1.3. RECOMMENDED: Limit switch system for bots at minimum X positions should be designed to maximize position repeatability

6.5.3.1.2. Y-axis limit switch triggered at either the minimum or the maximum Y location.

6.5.3.1.3. Z-axis limit switch triggered at either the maximum or minimum Z location.

6.5.3.1.3.1. Note: in certain error situations it may be advantageous to have limit switch at maximum Z.

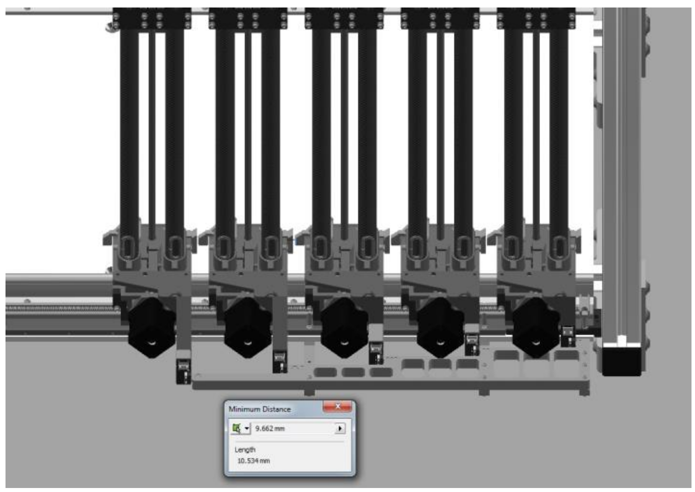

Figure 3 Five (5) Bots in Homed Position with a small buffer between each bot

6.6. Bot Collision Detection

6.6.1. Bot collision is defined as two adjacent bots attempting to simultaneously move into the same x-coordinate space of the machine.

6.6.2. RECOMMENDED: Bot-to-bot collision switches/shut offs on all bots

6.6.2.1. Functionally E-stops

6.6.2.2. RECOMMENDED: Shield switches to prevent accidental actuation by operator

6.6.3. NOTE: Bots only require collision detection capability in the X (long) axis; there should not be the possibility of adjacent bot collisions in either the Y or Z directions.

6.6.4. All bot components and subassemblies must not collide/interfere when bots are spaced at the minimum X bot spacing.

6.6.4.1. For example, adjacent bots at home position in the X (long) axis should be able to move Y and Z axes without collisions.

6.7. Bot alignment

6.7.1. Means should be provided to ensure bot coordinate systems are aligned.

6.7.2. For discussion of bot misalignment prevention (Skew/Racking) See Design Supplement Document.

6.8. Nozzle Purge and Wipe

6.8.1. Discussion: There are periods during many prints when individual bots are idle and moved to the “park” position. Prior to parked bots’ return to printing, their extruders should be purged and wiped to ensure high print quality.

6.8.2. Incorporate nozzle wiper and purged material handling for each bot

6.8.2.1. The wiper and means to handle purged material should be on the negative-Y side of the build platform and should travel with the X-axis such that it can be reached from any X position using only Y movement for a given bot (Figure 4).

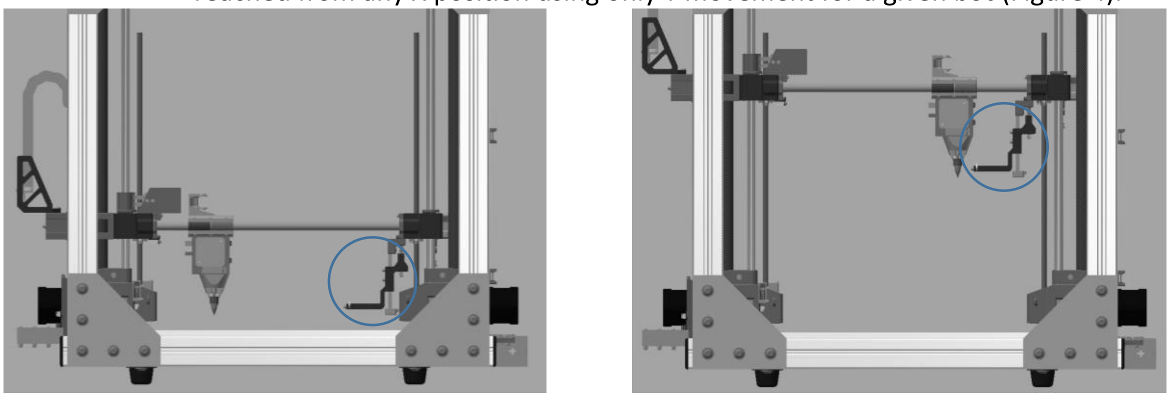

Figure 4 The Nozzle Purge Wipe Moves in Z with the Extruder

7. Per Bot Motion Controller

7.1. Description

7.1.1. The motion control system is composed of a dedicated “Conductor” computer (Raspberry Pi, supplied by OEM) and a quantity of motion controller “Player” computers (e.g. Smoothieboard, Arduino etc., supplied by OEM) equal to the number of bots. The “Conductor” is a separate computer dedicated to distributing and dispatching portions of G-Code as they become available for execution. Each “Player” computer is a dedicated motion control system capable of interpreting XYZE commands and communicating with the Conductor. The “Conductor” communicates with each “Player” either directly over Ethernet or using an Ethernet translator add-on (i.e. Raspberry Pi).

7.2. I/O

7.2.1. Boards that communicate over serial ports and use Marlin firmware, or similar flavor gcode interpreters, are currently supported. Other motion control systems can be adapted to work with MCU by creating a communication driver.

7.2.2. A dedicated Ethernet switch is recommended to connect and communicate between the players and the conductor over a network.

7.2.2.1. We recommend against using Wi-Fi for communication due to potential signal drop-out.

7.2.3. Digital Inputs

7.2.3.1. The system requires limit switches for X, Y, and Z axis. See Section 6.4.3. for further information.

7.2.4. Analog Inputs

7.2.4.1. System requires a thermistor or thermocouple for the extruder capable of sending a reliable temperature signal to the bot control system

7.2.4.1.1. Temperature sensor must be capable of measuring up to desired nozzle temperatures for the expected printing materials

7.2.5. Analog/PWM Outputs for extruder
7.2.5.1. Recommended: Two blower fans, one for the extruder's melt chamber, and one for cooling extruded filament.
7.2.5.1.1. One blower fan will be connected to the Printer board to be controlled by PWM.
7.2.5.1.2. RECOMMENDED: Connect the extruder chamber's blower fan to an external power supply independent of e-stop and collision switch system.
7.2.5.1.2.1. In case of a bot collision switch activation or e-stop, the bot's printer board and motors will power down but the fan will stay on.
7.2.5.1.2.1.1. Prevent overheating of the nozzle / development of material jams.
7.2.5.2. An extruder heater cartridge capable of reaching desired print temperatures.
7.2.5.3. RECOMMENDED: An electrical connection for landing the following:
7.2.5.3.1. Collision Kill-switch DC+V
7.2.5.3.2. Collision Kill-switch GND
7.2.5.3.3. Persistent Extruder Cooling Fan DC+V
7.2.5.3.4. Persistent Extruder Cooling Fan GND
7.2.5.4. RECOMMENDED: Collision Kill-switch
7.2.5.4.1. In the event of bot-to-bot collision, a normally closed limit switch connected to each bot, activated by adjacent bot.
7.2.5.4.1.1. Position should place limit switch and limit switch trigger as first point of contact between two bots, along the x-axis.
7.2.5.4.2. RECOMMENDED: Collision Kill-switch should have a non-momentary bypass for post-collision handling
7.3. Motor Drivers
7.3.1. Install motors and drivers capable of driving the mass of the bot at the desired speeds and accelerations.
7.4. Implementation
7.4.1. RECOMMENDED: The motion controller should hold motor in a locked position when it is not moving.
7.4.2. RECOMMENDED: If driving more than one Z axis motor, Z axis should have negligible slip when depowered.
7.4.3. RECOMMENDED: On command, the motion controller should be able to disable holding current on each individual axis to facilitate maintenance and possible manual bot movements.
7.4.4. The motion controller must have a configurable extruder acceleration profile.
7.5. Other Notes
7.5.1. In order to compensate for physical discrepancies between coordinate systems of each bot, the MCU software offsets G-code prior to sending it to each bot.
8. The computer with Netfabb on it will be used to upload the sliced '.esh' file to the "conductor" for printing. During this time, the computer will need to be on the same network as the printer.
8.1. Note: The "Conductor" and "Player" MCU instances are independent from Netfabb. Once a file is uploaded and a job is started, no computers outside of the Raspberry Pis running MCU will be needed.
9. If a user wishes to interact with a raspberry pi's MCU instance, it must be on the same network as the raspberry pi.
10. Other Electronics
10.1. E-Stop button

10.1.1. RECOMMENDED: non-momentary e-stop
10.1.2. RECOMMENDED: Cuts power to all motors
10.1.3. RECOMMENDED: Does not cut power to extruder cooling fans