Project Definition

Today's industries are demanding lighter and stronger materials. Carbon, glass, and Kevlar fibers are promising to deliver these benefits, however, they are typically more expensive and time-consuming to manufacture than traditional materials. A 3D braiding machine could combat some of these issues by reducing manual labor. Continuous Fiber capable 3D printers already exist, however they are limited to the planar application of fiber. Other 3D Braiding machines exist, but are far too large and expensive for everyday use or do not integrate additive manufacturing. Existing machines can be seen here:
https://www.youtube.com/watch?v=cKcvnwf4oDw
https://www.youtube.com/watch?v=iR9MEi9-EG8
https://www.youtube.com/watch?v=h5TbO1rNj6A

The goal of this project is to design and develop a prototype braider that can braid fiber across three dimensions, which meets space, cost, function, ergonomic, and safety constraints specific to our customer. In future iterations, it may integrate with additive manufacturing to produce one cohesive part. Our customer is Dr. Denis Cormier who operates in the AMPrint lab. The machine will be located in either the AMPrint or the Brinkman Lab at RIT. All funding comes from our customer and his budget. Dr. Cormier also gave us spare parts, filament and access to his high end 3D printers for this project.

Requirements

In the project document, we were given a list of initial requirements from Dr. Cormier. After several meetings and design reviews we added to this list and created engineering requirements from them as well. The master spreadsheet can be seen here:
https://docs.google.com/spreadsheets/d/19ZkXIih5ps13-FzijnVPbb-Dooc5hIuNzhpTlrU-790/edit?usp=sharing

To summarize, the main requirements are that the machine is:

- Safe
- Scalable up to a 25X25 grid (or has plans on what would need to change)
- Functional
- Able to create a prototype braid
- Has detailed documentation allowing easy transition to future teams

All other requirements are built off of these or are secondary in importance.

Subsystem Design

In order to meet these requirements, we identified a few subsystems that needed to be designed. They are:

- Mandrel Holder- holds the mandrel in place to be braided
- Mandrel Mover- moves the mandrel so that the braid can go along the piece
- Bobbin Passing- moves the bobbins around so that the braid can be made

- Tensioning- maintains tension in the bobbins so the braid is tight
- Bed Orientation- the location of the mandrel with respect to the bobbins

In order to determine a design for each sub system, each of us made morphological charts with potential designs. Then we made a decision matrix for each and determined our own ranking criteria. They can be seen here:
https://docs.google.com/spreadsheets/d/1IpaTevaC_M_Vv20L0imULOpVaidwoqGCdDjy2BO3Rkc/edit?usp=sharing

Safety and compatibility with other systems were the most important criteria as they relate to our initial requirements of safety and functionality. Integrating with 3D printing and low complexity were the next as we believe design considerations that allow for 3D printing, and are not tremendously difficult to implement are ideal. Cost, ease of use and weight were low priority as we felt it unlikely to go over our weight limit and our budget. And after speaking with our customer, he expressed that the budget can be increased if it leads to a better machine.

Design Decision Rationale

We ended up making these choices based on our decision matrices:

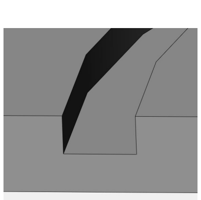
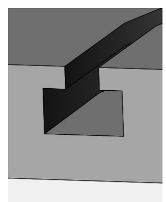
- Tensioning- 1 pulleys with two strings on both ends.
    - The bobbin will also have a knob on top to tighten the rotation of the spool; therefore, creating a bit of tensioning on the string. Ultimately, with every component moving accordingly the tension should be consistent throughout the braid.
- Bobbin Passing- Horn gear slots.
    - Horn gears will have openings that can grip the bobbins. Bobbin movement will be confined to a track with switches that allow for different paths to be followed.
- Mandrel Holder- Pegboard
    - Fibers will be attached to a pegboard on the top of the machine. This way fibers can wrap around the mandrel.
- Mandrel Mover- Belt & slider
    - The belt and slider would move the mandrel up and down to create different braid angles and allow the entire mandrel to be braided.
- Bed Orientation- Bobbins on bottom
    - The bobbins and horn gears will be on the bottom plane, the pegboard will be on the top, and the mandrel will be in the middle.

These were good starts for our designs but they changed throughout the semester.

Final designs are as follows:

- Tensioning
    - The bobbin would be placed into the bobbin carrier. A threaded rod will be going through the bobbin and the metal nut will be tightened on the top to control the tension. The tighter the nut the more tensions will be created.
- Bobbin Passing

    - Same as initial. Horn gears will spin and move the bobbin. Solenoids will move to change the tracks that the bobbins can move on. This allows for more control of the braid in a 3X3 matrix.
- Mandrel Holder
    - The mandrel holder will be pegboard style where threads go through the holes and wrap around a bead that prevents it from backing through.
- Mandrel Mover
    - A lead screw has replaced the belt system. Electric motor is mounted from above on the frame, driving a lead screw which interfaces with the mandrel moving board, which slides up and down on guide rails.
- Bed Orientation
    - Same as our initial design. Bobbins move on the horizontal plane and braid around a mandrel extending in the z (vertical) axis. The main benefit of this style was easier retention of bobbins in the horn gear bed. Currently, the boat path slot is not enclosed, which allows us to use a simple end mill on the CNC router, as opposed to purchasing a T Slot cutter. It also reduced friction and mechanical complexity. This will also be beneficial for future 3D printing integration.


(Open Slot) vs.

(T Slot)

Our controls also had many, many, many design decisions that will be covered in another section.

Final Designs

Our design is broken down into 4 major subsystems. Bobbin movement, mandrel movement and holding, bobbin tensioning, controls and electronics.

Bobbin Movement:

At the bottom of each bobbin there is a "boat" that fits into the track in the horn gear board. This track (shown orange, Figure 5) limits the bobbins to circular motion, except where the circular paths meet and bobbins pass from one circular path to another. This switching is facilitated by a 3D printed component (shown blue, Figure 5) that can rotate 30º to allow bobbin movement between any two adjacent horn gears rotating in opposite directions. The bobbins are loaded into the machine via a slot in the lower right corner which is plugged during normal operation.

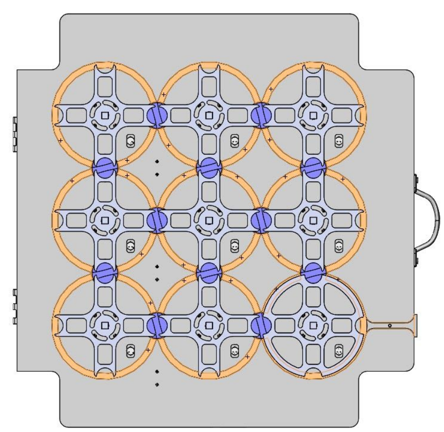
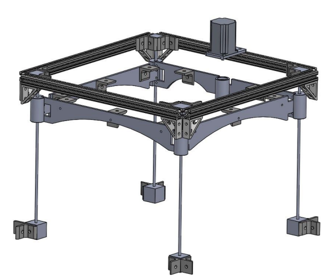


Figure 5: Horn Gear bed with installed boat path switchers and horn gears

Mandrel Movement / Holding:

The mandrel holder and mover works in conjunction with the bobbin tensioner to ensure that the fiber is held taut. As the braid is created, the mandrel mover steadily moves upwards through a stepper-controlled lead screw. The



mandrel holder provides a place to secure a mandrel and to hold each of the strands of fiber in a peg board. It is likely a universal clamp will work for most mandrel types, but in some cases a custom one may be needed based on mandrel geometry.

Figure 6: Mandrel Mover (not pictured: pegboard, lead screw, and mandrel-holding clamp)

Bobbin Tensioning:

The other end of the fiber is held by the bobbin tensioner. Beneficial friction can be regulated using a screw on top of the bobbin panel to prevent the bobbin from unraveling. Springs and a pulley help to counteract the varying length needed as the bobbin moves closer and farther away from the center of the mandrel. Previous designs did not use this spring and pulley mechanism (only the friction from the bobbin clamped with a hex nut) but the team noticed that the fiber may hang too low as it moves through the path if there is no support given for it to stay upright. Therefore, a supporting panel alongside springs would be able to lift the fiber up after it unravels from the bobbin.

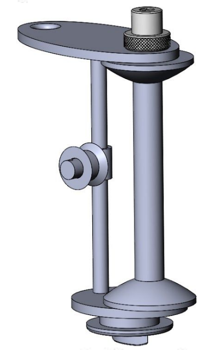


Figure 7: Bobbin tensioning and supporting components.

Controls and Electronics:

The control subsystem is responsible for coordinating and facilitating the movement of the other systems. The back bone of the controls system is a network of microcontrollers. This network is tied into a PC to receive commands and execute them with proper timing. The network is TCP/IP over ethernet and sends commands to micro controllers that store commands until they are told to execute the commands. To ensure proper timing, an interrupt connected GPIO signal is sent from a main microcontroller to the rest of the controllers. This single line ensures proper timing in the system.

The machine required two main electrical components to function. Stepper motors were used to rotate the horn gears and move the bobbins. Solenoids were used to allow the bobbins to switch between adjacent tracks. One additional Stepper motor was required to control the mandrel mover.

Both of these devices were controlled using the STM32H7 Nucleo-144 microcontrollers. A total of 4 microcontrollers were used for the 3*3 prototype. This allowed for the control of the 10 (3*3+1) stepper motors and the 12 solenoids. Stepper drivers and solenoid drivers were used to connect the components to the microcontroller.

The fourth microcontroller used to control only one motor is also designated as the main microcontroller. This will be used for the master once the microcontroller network is set up. It will also be the microcontroller communicating with the PC.

The stepper drivers were “hats” designed for the STM32 microcontrollers. Each microcontroller can have a stack of three hats which allows for each microcontroller to control 3 steppers.

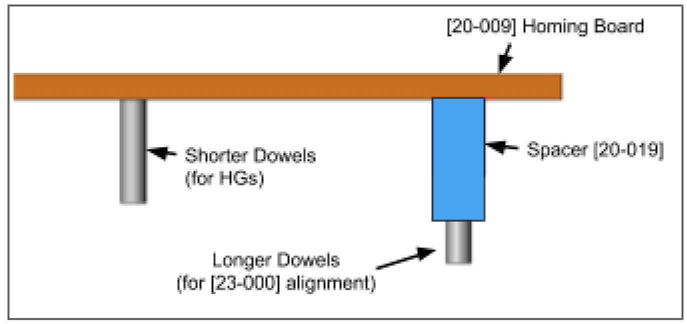
The solenoid drivers could each control 2 solenoids. This meant that each microcontroller was connected to two solenoid drivers. The solenoid drivers were controlled using pwm signals.

The machine is also built to be run on wall power. To accomplish this an AC-DC power supply was chosen that could provide 12V DC and 500W of power. This was necessary to be able to control all the motors and solenoids. While this is sufficient for the 3*3 and can be scaled up a little, it will not be sufficient for the 25*25 and a power supply with a higher power output will need to be chosen.

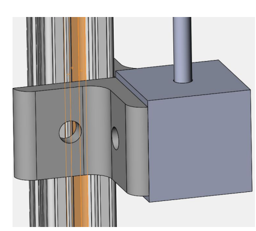
Open/Future Design Tasks

Due to time constraints, there were a few design tasks we were unable to completely finish.

- Horngear Homing board [23-000]
    - The original design for this was that the board would attach to the mandrel mover. However, due to the desire for ergonomic wiring access, we moved the bottom mandrel moving supports up so that there would be no collisions between mandrel mover and board. This means that there are approximately 8 inches between the mandrel mover and the HG Board at the mandrel mover's lowest position
    - Long story short, the new suggestion is to mount the homing board directly to the HG board rather than have more than 8 inch dowels. Have longer dowels on all 4 corners with 3D printed plastic spacers [20-019] (so that you can let go of the board), and shorter dowels to act as homing for each horn gear. You will need to orient the HGs in approximately the correct position before starting



    - You will need to add tight fit dowel clearance holes to [20-006] to accommodate the 4 corners
- Securing the HG Bed [20-000]
    - While working with the wiring, you're going to want something to hold the HG Bed/Board up. After the frame is built you could honestly just use a bungee cord or some rope through the HG Bed handle, just make sure you secure it properly so it doesn't come down and hit you in the head.
- Wire Management [91-000]
    - Please see "Wire placement & Routing" in this document, under Assembly Manual > Wiring > Wire placement & Routing.
    - In CAD, the wire clips [90-001] should be modified so there is a configuration for each type of wire being used.
        - This will make the CAD assembly easier because you can just switch which configuration is used rather than insert another part if you decide to make wiring changes later
- Solenoid Rail Spacer
    - This probably won't be necessary, but you could add 3D printed or cut tube spacers between the HG board [20-006] and the solenoid grid mounting structure [25-000] so that the solenoids aren't acting as the only stopping point when you tighten down the ¼"-20 bolts [92965A546].
- Mandrel Mover
    - Due to illness, one of our team members was unable to complete some aspects of the mandrel mover. One issue is that the lower brackets of the mandrel mover do not line up with the slots in the 80-20 (see image)



- "Design for Assembly" (DFA) Improvement

- 25-000 attachment to 20-006 could be improved.
- Bolts are assembled from step 5c but then need to be removed carefully since the head of the bolt needs to be above the HG Board. If the nut-side is above the board, it will be too tall and will collide with the HG as it rotates
- Possible Solutions:
    - Raise the HG by making the adapter taller and have bolt head below solenoid rails
    - Cut off the end of the bolt and have bolt head below solenoid rails
    - Add a spacer below the solenoid rails so the end of the bolt doesn't stick out so much. Have the bolt head below solenoid rails.
    - Only use thick rails that interlock so assembling without bolts is easier (and leave bolt head as the side above the HG Board.

CAD

General CAD Information

Our team decided to use Solidworks as our modeling software since members of the team already had it installed. This can be obtained for free as students by contacting Solidworks directly. If you contact them and describe what type of work you will do for your project, Solidworks will send you an email with codes for activating SOLIDWORKS Student Design Kit. For the Rochester area the phone number is +1(585)486-4740. Just as a side note if it still works, the current SDK – Student Design Kit –CADimensions code is (9SDK2019). This could change for a newer version, but it doesn't hurt to try. The code can be utilized at www.solidworks.com/SDK.

Although we strongly suggest that you use Solidworks, we have provided .step and .stl (for 3D printed components, file units are [mm]) in case you choose to use a different software. However, if you decide not to use Solidworks you will lose the ability to conveniently change dimensions, configurations, and assembly mates. Having Solidworks will be incredibly important for assembly since instructions are given based on part number and are divided by subassembly. Changing CAD packages would require a lot of unnecessary rework from you.

Due to the nature of the design and the goal of making it scalable, Global Variables were used extensively, as well as dimensioning functions, part/assembly configurations, and linear patterns. Please note that changing the value of a Global Variable in one location will change it everywhere in that part or assembly, but not in other part files or assembly files. Some of the parts or assemblies contain configurations that show some of our original design ideas. Please do not go through and delete suppressed items in the feature tree without ensuring they aren't used in a different configuration.

Note- File names correspond to their part ID numbers in the Bill of Materials.

Naming Convention

The first number (# _ - _ _ _) indicates which subsystem the part or assembly belongs to (see the following figure for details). Three zeros (_ _- 000) indicates that the file is an assembly. Upper level subassemblies should be named: _0 - 000 when possible, where the first number is their subassembly number. All parts should be (# 0 - _ _ _) when possible, where the number is the subsystem they belong to, and it should be done sequentially such that there are no gaps (ex: 20-001, 20-002, 20-003 etc.)

Subsystems	
top level	00
frame	1X
horn gear matrix/control	2X
carrier/tensioner	3X
mandrel moving / holding	4X
micro controller / solonoid driver	5X
power regulation	6X
Misc	9X

- 00-000 *Top Level*
- 10-000 *'80-20' Frame Asm*
    - 11-000 *HG Bed Hinge*
- 20-000 *HG Bed Asm*
    - 21-000 *HG & Adapter Asm*
    - 22-000 *Bobbin Switcher Asm*
    - 23-000 *Homing Asm*
    - 24-000 *1 Solenoid Rail*
    - 25-000 *Grid of Solenoid Rails*
- 30-000 *Bobbin Tensioner Asm*
- 40-000 *Mandrel Mover Asm*
- 50-000 *3 stepper drivers on microcontroller*
    - 51-000 *Microcontrollers on mounts*
- 91-000 *Wire Comb Clips Asm*

Exception to the above naming convention: If parts were purchased and unaltered, we usually didn't change the PN they came with from the supplier (ex. McMaster and Grainger PNs).

BOM

Here's a link to our BOM:
https://docs.google.com/spreadsheets/d/1FoziwT1IYr9S8veS1bGnfN_yufplJE7XOCOmCX8AEu0/edit?usp=sharing
comments/concerns:

Rough cost scaling can be found: Control Systems -> Design Scaling

Controls/software

Software Dependencies

- CubeMX
    - Board and project configuration. Not needed for day to day development.
    - Outputs Keil uVision project, makefile, or other project type
    - Proper selection of components and initialization is not to be trusted. If you have issues, write the init by hand.
- Keil uVision

    - For:
        - Development
        - Compiling
        - Microcontroller programming
    - Easiest way to develop
    - Code exceeds free version limit
        - There are alternatives
            - Make + GCC
            - STM32cubeIDE
                - May need some rework and kerjiggering
                - Seems to have issues with messaging.c

Software Composition

Code is located:

- Google Drive->Control Systems- Code
- AND (Just a copy, nothing new)
- Edge->web->private->Code

The software/project layout was created by CubeMX. It's the easiest way to get startup files, FreeRTOS, and LWIP. It is terrible at most other things, so don't trust it. CubeMX can generate many project types such as Keil uVision or a Makefile. Keil was chosen for its debugging abilities.

The software is made up of multiple parts:

- FreeRTOS - Imported from CubeMX
    - Real Time Operating System
    - For our microcontrollers
    - Allows concurrent tasks
- Stepper Driver BSP - Imported from STM's website
    - BSP = Board support package
        - Software layer (API) to control stepper driver hardware with microcontroller
    - For the stepper drivers.
    - Needed modification to work with this board.
- LWIP - Imported from CubeMX
    - Lightweight IP
    - Software IP Stack for Microcontroller
    - Best solution for TCP/IP
    - For Ethernet communication.
    - Not setup yet
    - WILL TAKE A LOT OF TIME
- TexMind Computer Controlled Braiding Machines Configurator
    - Models bobbin path and horn gears with swapping
    - Enables previewing of braid patterns
    - Allows for configurations of beds of any shape & size
    - Exports code for braiding path

    - PAID SOFTWARE - only trial version was tested
- Separate Logic and Motor Power
    - Allows disabling of motor power without disabling logic
        - Useful so a pause doesn't trigger recalibration
        - Safety

What We Did:

- Configured Stepper Driver BSP
- Basic Solenoid and Stepper Movement
- Basic Packet Communication Over USB
    - Note: Right now communication happens before the index is passed. In the future, communication should happen while motors are moving to cut down on wasted time. Care must be taken to prioritize estop messages and the like over normal packet communication.

Variations from Long Term Plan

- Communicated Via USB instead of Ethernet. Done to conserve time. It was known that there was no time to set up LWIP and TCP Client/Server. USB was already used for programming and debugging communication. While the medium may change, the packet based communication code should remain valid.

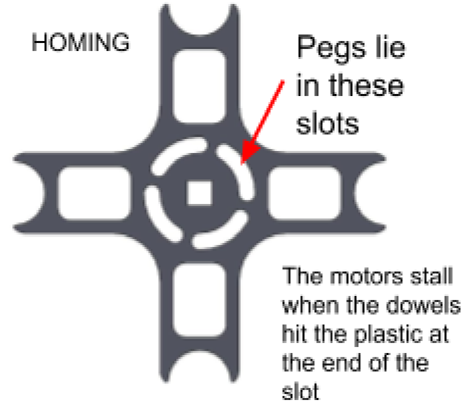
Important Documents:

- Control Systems Work Flow
    - What has been done and what needs to be done
- SW Flow -TCP-IP.drawio
    - Flow diagram of how software should work
    - Rough Guide, not set in stone
- Design Scaling
    - Power
    - Cost
    - Wires

Homing/Calibration

- Why:
    - When steppers turn on, it is unknown what step will be active.
    - Stalls
- When:
    - Power Up
    - After Stalls/skipped steps
- How:
    - Orient horngears in roughly the same position

        - Align homing bed [23-000] (bed of dowels/pegs) onto HG board [20-006] using the 4 corner dowels of the homing bed and the holes in the HG board
        - Move motors until they stall
        - Store known location
        - Remove Homing Bed
        - Go to a default position



Stopping the Machine

- Triggers:
    - E Stop
    - End of braid
    - SW Pause
    - Stall detected
    - Cabinet Opened
- How:
    - Broadcast a stop/pause packet. State reason
    - Each node will stop appropriately(Hard VS Soft)
    - PC will display the status

In Progress Issues

- PWM only works on TIM4, not TIM 1.
    - TIM1 is used for PWM1 and PWM2
    - TIM1 can generate PWM when not using the BSP.
    - If not resolved, just jump the PWM3 signal to PWM1&2
- Reset
    - Motors go into Hi-Z state upon first movement. See "Control Systems Work Flow" for more info

User Created Files

Located under src

- ControlMessages
    - Defines packet structures.
    - Makes communication independent of medium
- Coordination
    - Tells the Solenoid and Stepper task when to move
    - Coordinates Between different movement tasks.
    - Common point between solenoids and steppers
    - This is probably a good spot to add a state machine when things get more complex
- Freertos

    - Created by CubeMX
    - Not Modified
- Led_task
    - Mostly for debugging
    - Should be reworked to display errors
    - Was the “HelloWorld” task
- Main
    - Mostly created by CubeMx
    - Calls task init functions
- Messaging
    - Provides Printing and reading from UART.
    - Reads in data and uses the Registered PacketHandlers to send the data to
- SolenoidDriver
    - Controls Solenoid Movement
- StepperDriver
    - Works with the BSP to talk to Stepper Hat and control stepper movements.
    - Code is heavily based off of the BSP example
- stm*
    - Created by CubeMX
    - Not Modified

BSP Important Files

This is really just a huge heap of code that is hard to digest, but it’s better than writing it all again. This software set looks like it’s meant to work with many microcontrollers and many stepper drivers. The BSP files call board specific functions. We are using the STMH7XX files for our microcontroller. The stepper we use is ihm01a1.

The biggest advantage of use this package instead of writing our own is that the driver has its own communication standard and writing that would be a nightmare.

It may be worth it to strip out code we don’t use from this package.

Drivers/BSP/X-NUCLEO-IHMxx

- x_nucleo_ihmxx.*
    - Defines BSP functions that call out to the actual implementation elsewhere
    - Uses an odd pointer system to mimic object oriented code
        - Makes it hard to go to reference via an IDE
    - Not all functions that are defined here actually work with our stepper driver.
    - Try to avoid using this to see what functions are available and what they do under the hood.
- X_nucleo_ihm01a1_stm32l0xx
    - Actual implementation of stepper driver functions
    - Looks here for board capabilities
    - Interfaces with the HAL.
        - Needed to point towards the proper ones generated by CubeMX
        - Uses some of its own HAL

    - L6474_Board_Delay modified to use VTaskDelay.
        - Created dependency on FreeRTOS

Drivers/BSP/STM32H7xx_Nucleo

- Stm32h7xx_nucleo
    - Led and button control
    - These should not be used.
    - Look into removing

Drivers/BSP/Components/l6474

- L6474
    - Communication with stepper hat via SPI

Drivers/BSP/Components/Common

- Motor
    - Typedefs and enums for motor control. Used by the Drivers/BSP

Drivers/BSP/Components/lan8742

- Lan8742
    - No idea where this came from. Might not have been stepper driver related
    - May conflict with LWIP when that is used.

Current FreeRTOS Tasks

If weird behavior starts happening that may smell like memory issues. Try increasing the allocated memory for a task in questions. If the LEDs stop blinking, this is a good sign that there was a hard fault or issues with tasks.

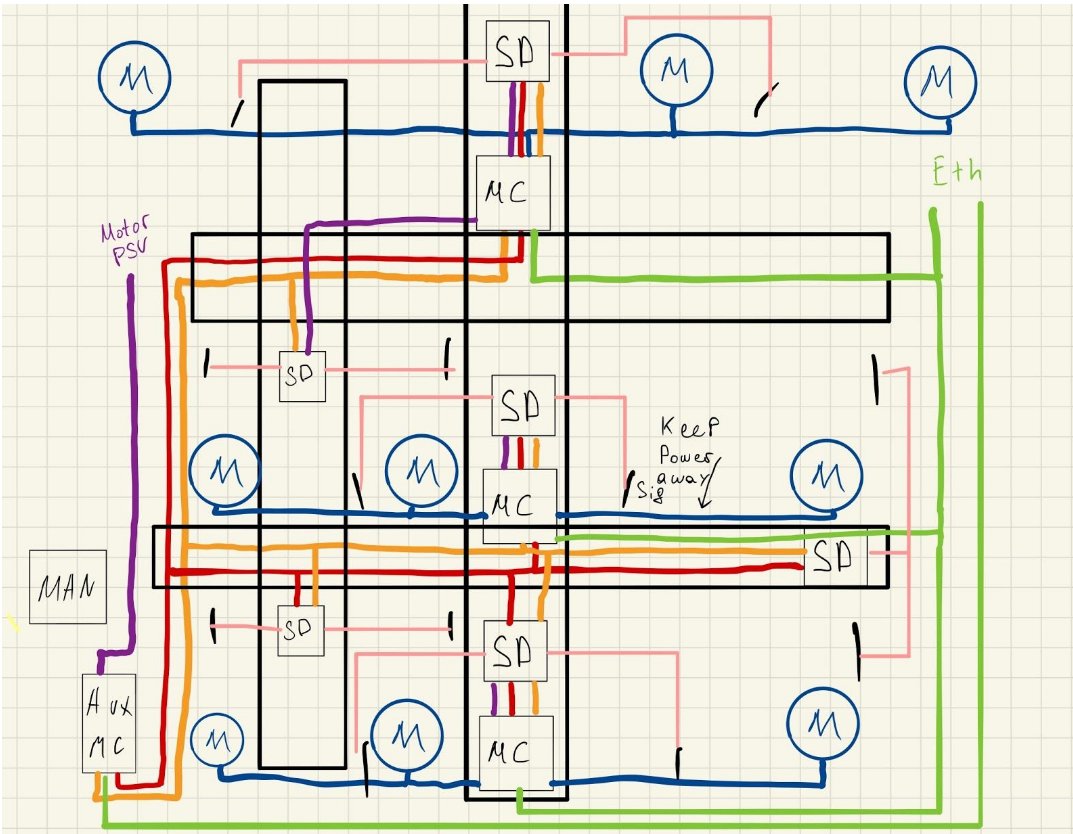
Make sure that tasks have some vTaskDelay call in them or it will starve other tasks.
Some blocking BSP functions had their delay turned into vTaskDelays to alleviate this problem.

- CoordinationTask
    - Starts movements
- Led_task
    - Blinks LEDs
- MessageTxTask
    - Prints out messages from a queue
- MessageRxTask
    - Reads data and sends it to the appropriate header
- SolenoidTask
    - Movement of Solenoids Upon Command
- StepperTask
    - Movement of Steppers Upon Command

Wires

See:

- Control Systems -> Design Scaling (it’s a google sheets doc in our Shared Google Drive)

- 

- See Control Systems -> Pics -> Wiring <x>

Assembly Manual

This will be a detailed manual on how to assemble the machine given all required parts. Parts will be referenced by their part or assembly number (should match the BOM and CAD). Note: HG means horn gear, uC means microcontroller. PLEASE ENSURE CAD IS OPENED ON A COMPUTER BEFORE TRYING TO ASSEMBLE THIS. Otherwise you will have a very bad time. The subassemblies specified below are each their own subassembly in CAD, it would help to have each open as you assemble them. If you don't have Solidworks on your laptop yet, you could try to get by with just the 3D PDFs or step files, but it would be significantly more difficult and take a lot more time.

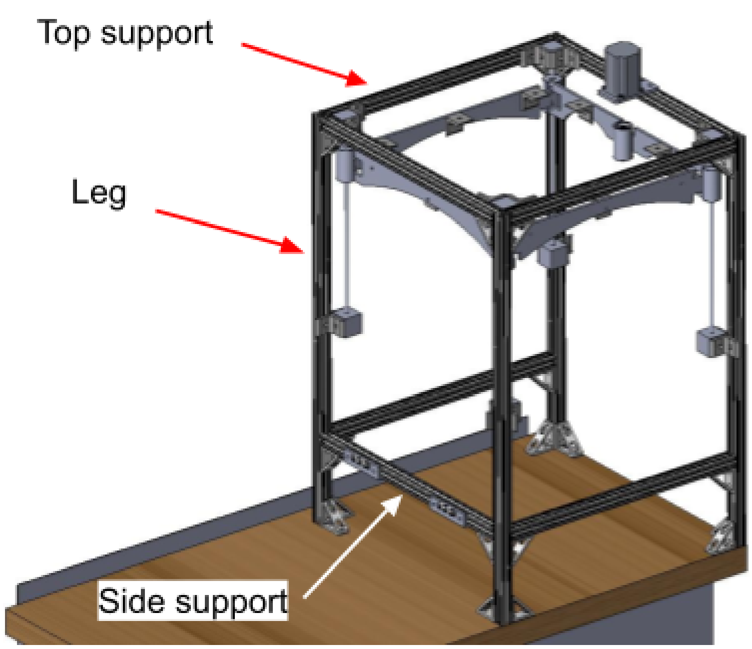
Tools needed

- Metric allen key set (no way around this sorry, the motors use them)
- Standard/English allen key set
- Standard/English 7/16" socket
- 7/16" open end wrench
- 11/32" open end wrench
- Phillips head screwdriver
- Tiny Phillips head screwdriver (for standoffs into microcontroller)
- Tiny flathead screwdriver (for loosening/tightening wire-to-board connectors)
- Drill & Drill Bits (if you need pilot holes for the hinge and handle wood screws into MDF. I was able to do the handle without it though). Also for drilling holes in the table/cart that the frame rests on

- If you haven't added the heat set inserts to [20-012] yet, you'll need a soldering iron
- Bungee cords or trustworthy rope (for holding the HG board up while wiring underneath)

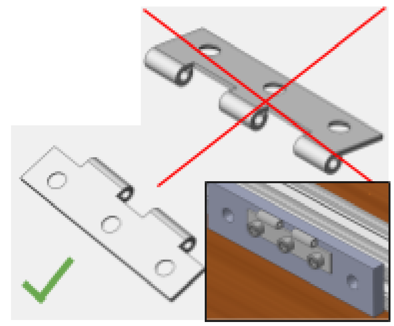
Assembly Steps

1. 3D print, purchase, and machine all necessary components. See "BOM" for purchased parts & material and "Mechanical Drawings" folder on EDGE for drawings. We do not have hardware specified for the mandrel mover yet, so this will have to be done before assembly. Some electrical components should have wires attached before assembling the component into a subassembly or full assembly. ***Examples***: Solenoids should have their 2 wires before going into 25-000; Solenoid drivers should have wires before going onto the rails; the microcontroller hats should have wires before assembly 50-000 and microcontrollers should be wired before [51-000] is finished in step 8c where they are attached to the HG Bed [20-000].
2. Test-fit boat switchers [20-005] motors [20-001] and boats [30-007] in the HG board [20-006]. If all components fit and there is low friction on the boat switchers in their slots and boats in the track, then coat the HG board in polyurethane [45870] to seal it. If they do not all fit smoothly, post-machining and/or sandpaper may be necessary. Disassemble the 2 sides of the hinges from each other (so you're left with 2 "3-prong" sides, 2 "2-prong" sides, and 2 connecting pins).
3. **Build subassembly 10-000** :
Frame



    a. Drill holes in the table in the correct locations for the brackets to hold the 4 legs, 4 holes per leg
    b. Assemble the 4 legs with mounting brackets to the table
    c. Mount the legs to the table by lining up the holes in the brackets with the holes in the table.
        i. Loosely tighten the mounting bolts to allow for adjustment of side and top supports

d. Assemble the side and top supports with connection brackets. ENSURE THE FOLLOWING BEFORE PROCEEDING, AS EXTRA HARDWARE MAY BE DIFFICULT TO MOUNT AFTER THIS STEP:
    i. T-nuts for the hinge adapter are inserted into the correct side support for mounting
    ii. Mandrel mover T-nuts are inserted into top support pieces, and the ones that attach to the frame legs are inserted BEFORE the top supports but AFTER the side supports
e. Slide the side supports between the legs, then slide the top supports between the legs.
f. Slide the top supports into position, then lightly tightening them in place
g. Slide the side supports into position, and securely tighten them in place
h. Securely tighten the top supports in place
i. Securely tighten the legs to the table
j. Assemble QTY2 hinge adapters [11-001] onto the correct side support (see CAD) using 80/20 bolts and the T-nuts (should already be in the 80/20 slots from step 3.c.i.)
k. Assemble QTY2 of the “2-prong” sides of the Hinge [11-000] onto the 2 hinge adapters using QTY6 #8-32 bolts [91251A190] and thread locker (loctite, etc.).

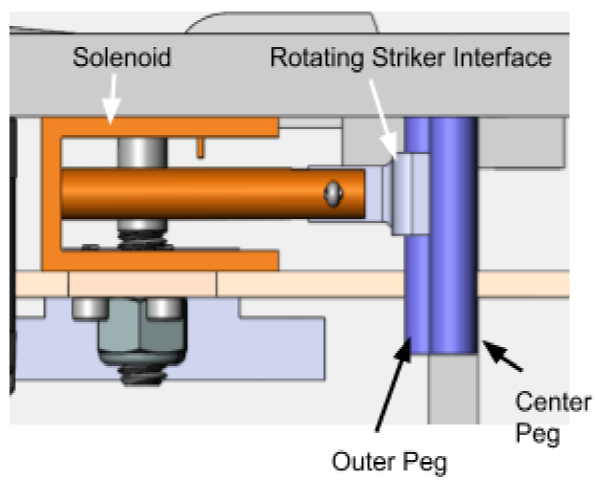


4. **Build Subassembly 23-000** :
   HG Calibration / Homing Board
   a. Lay empty calibration board [20-009] on a sturdy surface then hammer in QTY9 ¼” dowel pins [20-010] into the 9 innermost holes (outer 4 holes are clearance for ¼” mounting bolts) so that they are flush with one side
5. **Build Subassembly 25-000** :
   12 Solenoids on mounting structure
   a. Gather 6 Solenoid rails [20-011]. Half should be some variation of configuration “A” and the other half should be configuration “B”. Only 4 rails will have solenoids on them, the other 2 are for additional support and don’t need #4 clearance solenoid mounting holes in them (though they will still need ¼” clearance holes for mounting to the HG board [20-006] later.
   b. Assemble Solenoid Drivers [20-018] onto the Solenoid Driver Mounts [20-016] The mount pins slide on the driver on both ends. Then slide the mounts onto the Solenoid rails [20-011] in the proper location (see CAD).
   c. Assemble all rails as seen in CAD for 25-000, ¼”-20 bolts [92965A546] can be slipped through at this point for convenience, especially if thin rails are being used (without the machined slots that fit together). Notice that all 3 “B” rails are parallel, all 3 “A” rails are perpendicular to those but parallel to each other. Ensure the rail with no solenoid holes is closest to the end of the perpendicular rail it fits into (this is true for both “A” and “B”).
   d. Assemble 12 Solenoids [DSML-0630-12C] onto the rails using 2 #4-40 bolts [91251a106] for each solenoid. DO NOT OVERTIGHTEN the bolts, you may damage the solenoid since the end of the bolt sticks through and may contact the solenoid coil cover. NOTE: be sure that you’re assembling the solenoids on the correct side of the rails and correct orientation of the solenoid. Otherwise, you’ll

have to take them all out and try again. The mounting structure grid is not symmetric.

e. Assemble rotating solenoid striker interface [90-012_rev2] onto each solenoid using ⅛” cotter pin [90520a101 aka 30-013] and bend the cotter pin so the striker interface stays on

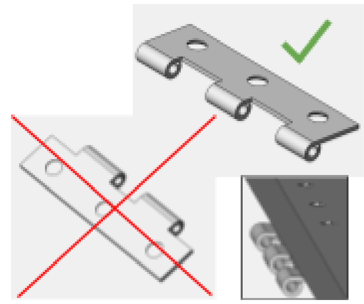
6. **Build subassembly 20-000** :
   HG Bed
   (for some of this assembly process, it might be convenient to set it on the frame)
   a. Press motors [20-001] into HG board [20-006] and bolt the motors to it from the top of the board using [94500A265] M3x0.5x16, 4 per motor
   b. Add heat set inserts [20-017] to motor-to-HG-adapters [20-012 > center only] using a soldering iron. Ensure that they don’t stick out past the outside edge of the adapter (since then the HG won’t slide on) and that the motor shaft still fits inside. More detailed instructions are found in the mechanical drawings folder.
   c. Assemble the solenoid mounting structure [25-000] onto the HG board [20-006] using ¼”-20 bolts [92965A546] (the bolts may already be assembled from step 5c but you will have to take them out since the head of the bolt needs to be above the HG Board. If the nut-side is above the board, it will be too tall and will collide with the HG as it rotates) and ¼”-20 nyloc nuts [grainger-3hdu1]. The solenoids should be flush against the HG board. The head of the bolt should be prevented from rotating by the machined pocket in the top of the HG Board, and the 7/16” socket or open end wrench should be used to tighten the nut.
   d. Add QTY12 boat switchers [20-005] to the HG board [20-006] such that the boat switcher’s outer peg fits in the HG Board’s slotted arc hole and the center peg fits in the center hole. In this step you will also have to hold and move the rotating solenoid striker interface [90-012_rev2] so that it slides over the outer peg. It may be easier to take the solenoid plunger out to get the rotating interface to fit. If the solenoid mounting rails and solenoids are in the correct locations, this should not be too challenging. If there is misalignment due to tolerance stackup, manufacturing errors, or other issues, minor design changes or adjustments may be necessary. Note: There are versions of the boat switchers made for testing that have a curved track on the top [20-005_test]. These will not work in the final assembly since they do not rotate.



   e. Slide motor-to-HG-adapter [20-012 > center only] on motor shaft, then HG [20-012 > without center] onto the adapter, then tighten set screws [91251a076] into heat set inserts [20-017]. Note that the HG nearest the loading slot should be the “Loader” configuration of 20-012.
   f. Using 2 wood screws [90031A196] assemble the handle [14865A410] roughly in the center of the HG board [20-006]. Make sure it’s the correct side of the HG board - the same side as the loading slot! If this is the first build, the handle is

probably still attached to our Rev 1 board and needs to be moved to the Rev 2 board.

g. Using 6 wood screws [90031A196] assemble QTY3 uC-rail-to-HG-bed-brackets [50-004] onto the HG Board [20-006]. There are already pilot holes in the MDF, so you shouldn't need a drill. Ensure the side of the bracket with countersunk holes is up against the HG Board since that fits the wood screws.
h. Using 6 wood screws [90031A196] assemble QTY2 of the "3-prong" sides of the Hinge [11-000] onto the HG Board (see images). Unfortunately, we do not have pilot holes in the MDF for these, so you will have to be very careful to ensure they are in the correct spot. If they are shifted too close or far from the center of the board, it may not hinge correctly and cause interference with the frame. If you make a Rev 3 HG Board, please make sure to include the pilot holes.



7. **Build 3 Subassembly 50-000s** :
   Microcontroller with 3 Stepper Drivers
   a. Carefully insert 3 of the microcontroller (stepper driver) hats [Fake_Hat aka 20-007] into 3 of the microcontrollers [Fake_Micro aka Node Microcontrollers aka 50-001], ensuring you do not bend the pins
8. **Build Subassembly 51-000** :
   3 of the microcontrollers on mounting structure
   a. Modify plastic bolts from the standoff kit [8002124922093] using sandpaper so that they can fit in the microcontroller mounting holes
   b. Assemble the QTY3 50-000 assemblies onto the microcontroller rail [20-013] using plastic standoffs from 8a.
   c. Assemble the uC rail [20-013] onto uC-rail-to-HG-bed-brackets [50-004] using nuts [90480A009] and bolts [91251A197]. The brackets should already be assembled in the HW bed [20-006]. Note: the bracket has countersunk holes that should be facing AWAY from the uC rail. Those are meant for countersunk wood screws that were used to assemble into the MDF
9. **Build subassembly 30-000** :
   Bobbins (you will need several of these assemblies)
   a. Bobbin [30-001] slides into threaded rods in bobbin carrier bottom [30-002].
   b. Pulley holder [30-006] slides into an unthreaded rod in bobbin carrier bottom [30-002].
   c. Pulley [30-004] attaches to the pulley holder [30-006] and is then pinned.
   d. Bobbin carrier top [30-003] then goes on top of the rods that are in bobbin carrier bottom [30-002].
   e. A knob [30-012] is then placed on top of the bobbin carrier top [30-003] to hold the bobbin carrier top in place.
10. **Build subassembly 40-000** :
    Mandrel Mover (will require design changes before assembly due to alignment conflicts between guide rod column blocks and 80/20 slots)
    a. The square part of the guide rod column [40-007] is attached to the rails [8020-1010 X 38] and brackets [40-002].
    b. Column slider blocker [40-005] slides on guide rod column [40-007].
    c. Guide rod base [40-006] is then attached to the column slider blocker [40-005].

d. Brackets then hold the guide rod base [40-006] in place.
e. The previous steps are then repeated for other rails.
f. Attach the two side arms [40-003] to the slider blocks [40-005] so that they are opposite each other.
g. Attach the front arm [40-004] to the slider blocks [40-005] between the two side arms [40-003].
h. Attach the back arms [40-010] to the other slider blocks [40-005].
i. Attach the lead screw adapter coupler [40-012] to the lead screw adapter [40-008].
j. Attach the assembled lead screw adapter [40-008] to the two back arms [40-010].
k. Thread the lead screw [40-012] partially through the lead screw adapter [40-008].
l. Mount the motor adapter plate [40-009] to the top support above the lead screw adapter [40-008].
m. Attach the driving motor [40-014] to the motor adapter plate [40-009].
n. Attach the 11 brackets to the holes in the front [40-004], side [40-003], and back [40-010] arms, such that the flat surface of the brackets are flush to the side and top of the arms.
o. Mount the mandrel mover plate to the brackets on the arms.
p. Couple the driving motor [40-014] to the lead screw [40-011] using the lead screw adapter [40-013].

11. **Build Assembly 00-000** :
Full Assembly
a. Lay subassembly 20-000 (HG Bed) onto the frame so that the halves of the hinges line up. You may have to loosen the 80/20 bolts and slide the HG Adapters [11-001] along the frame side supports.
b. Insert Hinge Pin and secure. Once the HG Bed is centered on the frame, tighten the 80/20 nuts holding the Hinge Adapters to the frame. Test hinging motion of board for safety & security
c. Place power supply [60-002] on table [10-002]
d. Insert desired number of bobbins [30-000] using the loading slot
e. Place the plug [90-011] into the slot in the HG Board [20-006] so bobbins don’t escape during operation
f. Finish attaching wiring (see Wiring section). Wire combs [90-000] should be used as needed. Wire comb design changes will likely be needed as you get a feel for how wiring works with this design.

Stepper Driver (20-007) Modifications

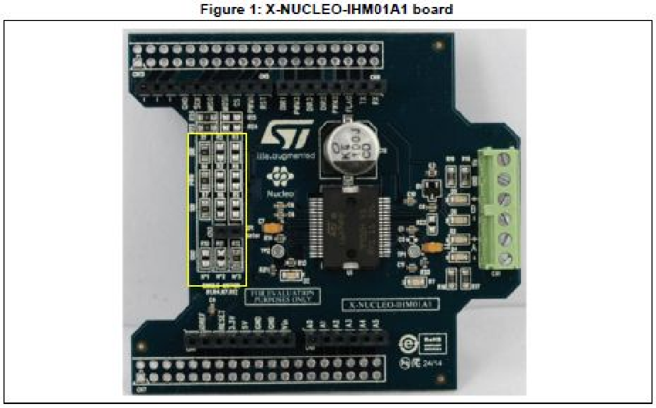
The stepper drivers are “hats” that can be stacked on top of the Microcontroller. To enable stacking of the stepper drivers the solder bridges on the drivers need to be modified as follows. Each microcontroller will be configured to control 3 motors (3 drivers per MCU). It is recommended to label the stepper drivers to ensure they are not stacked in the wrong order. They are labeled (0-2).

Table 5: Multi-motor setup table

Number of motors	Board	STCK\\DIR	Mounted resistors (0R)
1	-	PWM1\\DIR1	R1, R4, R7, R12
2	1 (bottom)	PWM1\\DIR1	R1, R4, R7, R10
	2 (top)	PWM2\\DIR2	R2, R5, R8, R12
3	1 (bottom)	PWM1\\DIR1	R1, R4, R7, R10
	2	PWM2\\DIR2	R2, R5, R8, R11
	3 (top)	PWM3\\DIR3	R3, R6, R9, R12

Table located in stepper driver manual

Figure 1: X-NUCLEO-IHM01A1 board



Stepper Driver image. The yellow square shows the solder bridge locations

Microcontroller configuration 12V VIN

The Microcontrollers will be powered by an external voltage of 12V from the power supply. To enable this the power jumper(JP2) will need to be moved. The default configuration has the jumper on USB power and this will need to be moved to the VIN pins. VIN is then connected to either CN8 pin 15 or CN11 pin 24.

see page 22 of edge\public\Design Documentation\Electrical Schematics\STM32H7 Nucleo-144 boards en.DM00499160

Wiring

Shielded wires should be used on any high speed signals (Communication and stepper pulses)

Stepper Wires:

Currently using the wires that came with the steppers. We have shielded wires that need to be used instead.

Wires using name on BOM:

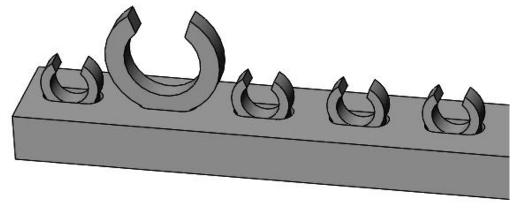
- Ethernet: Cat 6 (RJ45) STP Cables
- 18 gauge wire used to provide DC Power from the power supply to machine
  - Motor Power/Ground
  - Solenoid Power/Ground
  - Microcontroller Power/Ground
- Logic wire (22 gauge):
  - Logic Signals/Power/Ground:
  - Motor Signals

Power C13 cable:

AC Power wall to supply

Wire Placement & Routing

The mechanical / electrical integration plan was to route wires along the solenoid rails and microcontroller brackets. Preliminary 3D printed wire combs have been designed to help organize wires: [90-000, 91-000, 90-002, 90-001]. Different size clips [90-001] can be used for different wires or groups of wires and interchanged in the slots in [90-002] throughout testing. Labeling and routing along specific paths will be especially helpful for troubleshooting. Although cable sleeves would help clean up the final appearance, this would be a huge hassle for testing so the wire combs are our attempt to hold wires securely but also allow them to be taken quickly in and out and be traced from input to output if necessary. Additional design work is needed to fit the wire combs into the design where it makes sense (we were going to do this after we had started assembly -- it would be very difficult to judge the best locations from CAD alone)



Preliminary design for wire combs

Power Supplies

- One of Microcontrollers and Sensor
- One for Motors
  - Controlled by Main Microcontroller for safety
- AC Input
- DC Output
- Control Signal to selectively turn on/off PSU
- Common Ground
  - Both Supplies, ground AC to DC, ground frame

Power Strip

Requires a standard outlet plug (Part no 60-003).

Repairs and replacements

All parts that were 3D printed should be easy to replace with the corresponding STL files. There should be a page on the BOM with all of those organized.

Parts that were ordered can be reordered from the same supplier as detailed on the BOM.

If there is a motor malfunction, software should report the issue so you know which motor or solenoid needs repair/replacement.

When repairing:

- Machine should be powered down and unplugged
- External parts should be removed first to allow access to interior parts
- Keep all nuts, bolts, screws, as they are necessary to reassemble machine
- Electronics can be unplugged, but remember to plug them back in the same place when finished
- In case of a complete disassembly of the machine, follow the assembly guide to reconstruct

In the case of electronic damage, switches, motors, and boards will likely need to be replaced.

Scaling

Everything in our design works for a 3X3 and up to a 5X5 with added parts. Once you expand past that, some things might have to change. Dimensions can be easily adjusted if larger/smaller parts are requested.
The biggest concerns are in the controls scheme. The number of wires gets to be quite unwieldy pretty quickly as you scale up the machine. It might make sense to make adjustments to the controls plan like these:

A big change could be a way to control all horn gears with 1 strong motor instead of each having a dedicated motor. This, however, would require significant redesign for the controls and the bobbin passing subsystems. We believe that best would to have a sort of clutch that either allows for Clockwise or counterclockwise turning for each horn gear. Solenoid controlled switches would still be needed to steer the bobbins.

When scaling to above 10x10, it may alleviate wiring issues to orient the machine on its side relative to the current design. Our choice for vertical braiding orientation was to alleviate the mechanical design of the bobbin movers and to ease the eventual integration of actively 3D printed mandrel (during the braiding process). If a vertical HG Bed (horizontal braiding orientation) is selected, tensioning, mandrel mover, boats in tracks, and other smaller subsystems will have to be redesigned or changed. If a hinged system is still selected, consider automatic movement, or at least resistive springs / shocks. The automatic rear hatch on some cars may be a good example of an effective controlled hinge system. Another alternative is allowing space and sufficient support so that someone could be under the machine and work on wiring above them.

Future Recommendations

Electrical / Controls:

- If the ***power consumption*** of solenoids becomes significant and an issue, consider looking at latching solenoids since they are able to hold both in & out position without power, and only require power to change states
- Bus bars may be used to ***simplify power delivery***. For our application, it doesn't make sense since there are only 4 microcontrollers
- **Get more computer/software engineers to help on all the software requirements**

- Honestly, you could make a whole team of computer/electrical/software engineers just to integrate all the stuff we wanted to include.
- Make use of the dedicated PC. It might be able to handle more of the control aspects so that boards don't have to.
- Figure out how to move bobbins to a vertical plane instead of horizontal. Most braiders in industry are configured that way. It makes more sense as you scale up to a larger matrix.
- Schedule meetings with Dr. Cormier long in advance. He is very busy and can have a full schedule for weeks at a time.

Mechanical:

- You might be tempted to waterjet the HG board, but that prevents you from having a single piece (waterjet is "through all" only). You could make separate pieces and assemble them after, but getting them concentric within reasonable tolerances might be a challenge. A previous MSD team (P17432) did not have success with this endeavor. You could try a jig to help align them better, that's up to you. CNC was not chosen because the HG Bed is so large (24" x 24"). MDF is not an ideal material, but it allowed us to use the CNC router in the construct.
- One of the reasons we made the horn gears so large was so that the bobbin spools would be large enough to hold fiber for large mandrels (you cannot switch spools after the braid is started without starting the braid over) and the bobbins not hit each other. Another reason was to ensure we had enough room on the underside for electronics/controls. In the current size, we had a hard time fitting solenoids in (this is why we chose smaller solenoids, also why the solenoid mounting holes are not centered on the rail). With careful consideration you could push them closer together in future revisions. In CAD, many dimensions are referenced off a global variable, "HG_Spacing" to help make this change easier. This global variable will need to be changed in each part file and assembly in which it occurs. The current value is 6.1 inches.
- If a Horn Gear Board of a stronger material is required, CNC machined aluminum might be an adequate substitute, as this would increase its strength while keeping weight down. However, with switching to metal comes an increased cost, so be sure to talk with Dr. Cormier about budget options if this route is pursued.
- If a size larger than 24" x 24", or larger than the milling capacities can handle is needed for the bed, the bed design will need to be split up into pieces. With this comes the inherent issue of ensuring that the tracks will line up properly so tolerancing will have to be very precise.
- If the move to integrating with 3D printing is needed, the entire system may have to be flipped upside down, so that the bobbins are located at the top and the mandrel mover is at the bottom. Doing so would allow for additive manufacturing to take place at the same time as braiding, which moves from mounting points towards the bobbins. However, to allow for the print head to print in between and around the fibers, dynamic mounting points (i.e. mounting points that move) would be required. Possible ideas include, but are not limited to, attaching the mounting points directly to the print head (although this greatly impacts braid angle), or having an extendable print head, such as one that would extend the print head from the print-head drivers like a telescope (picture two motors driving a box at the top of the machine in the x and y directions, attached to the bottom of the box is a telescoping mechanism, which the print head is attached to). There are many other potential ways to solve this issue, but these could help in giving you some place to start.

Current Usage Instructions

Limiting Factors

1. Power Supply Wiring
   a. Was using a personal power supply

Instructions

1. Wire up microcontrollers and things you would like to move
2. Connect microcontroller to PC via USB
3. (Optional) Open Keil project if you wish to modify code
4. Connect to Virtual Com Port
   a. Use Putty, Realterm or any other Terminal
   b. You can find Virtual port in Device Manager->Ports
5. Send Index Cycle Instructions to Microcontroller
   a. Example located in Software->Python Scripts->SampleIndexPacket.txt
   b.
6. Press Blue Button on Microcontroller
   a. This triggers the index cycle to run
   b. Eventually this will be replaced with a signal sent by the main microcontroller to trigger the commands to execute
7. Repeat steps 5&6