HardwareX 6 (2019) e00081

Contents lists available at ScienceDirect

HardwareX

journal homepage: www.elsevier.com/locate/ohx

Hardware Article

HyperRail: Modular, 3D printed, 1–100 m, programmable, and low-cost linear motion control system for imaging and sensor suites

José M. Lopez Alcala, Marja Haagsma, Chester J. Udell, John S. Selker *

Openly Published Environmental Sensing (OPEnS) Lab, OR, USA
Department of Electrical Engineering and Computer Science, Oregon State University, OR, USA
Department of Biological and Ecological Engineering, Oregon State University, OR, USA

A R T I C L E I N F O

Article history:
Received 2 May 2019
Received in revised form 11 August 2019
Accepted 13 August 2019

Keywords:
Open source hardware
Hyperspectral Imaging
Greenhouse sensing
Environmental sensing
Linear motion

A B S T R A C T

Reliable, accurate, and affordable linear motion control systems for precision agriculture applications are currently not easily accessible due to their elevated cost. Most systems available to the public have price tags in the thousands of dollars and their dimensions cannot be easily customized. Current systems have a maximum length of about ten meters, and for a typical greenhouse application that length may not be sufficient. The price of the system increases with an increase in length, and with a base price in the thousands of dollars it becomes impractical to buy a system for such application. Our HyperRail is a modular linear motion system with a repeatability of 2 mm and current top speed of 200 mm/s. This is possible through a stepper motor driver that allows for 1/16th microstepping giving an average of 6180 steps per revolution. An advantage that this system has is its ability to increase or decrease the length of system with minimum effort and only nominal increase in price. The HyperRail can be mounted on a set of tripods or directly on the structure of a building such as a greenhouse. The base price for a three-meter system, on tripods is US$278 and only US$45 for each additional 1.5 m of length.

Specifications table

Hardware name	*HyperRail*
Subject area Hardware type	• Environmental, Planetary and Agricultural Sciences • Imaging tools • Field measurements and sensors • Electrical engineering and computer science

(*continued on next page*)

* Corresponding author at: Department of Biological and Ecological Engineering, Oregon State University, OR, USA..
E-mail address: john.selker@oregonstate.edu (J.S. Selker).

https://doi.org/10.1016/j.ohx.2019.e00081
2468-0672/

* (*continued*)

Hardware name	*HyperRail*
Open Source License	CERN Open Hardware License *GNU General Public License v3.0*
Cost of Hardware	HyperRail vTerra 3 m: $278.00 HyperRail v SupraTerra 3 m: $300
Source File Repository	https://doi.org/10.17605/OSF.IO/TQ3SB

1. Hardware in context

High-throughput optical scanning is becoming an essential technique for characterizing large plant samples. Our particular interest in this technique utilizes hyperspectral, push-broom cameras to capture images of pine seedlings. These cameras must move linearly at a constant speed to capture high quality images. Pushbroom, hyperspectral cameras are only capable of taking a snapshot that is one pixel wide, a time series of these snapshots can be stitched together to produce an image which covers the entire sample. These snapshots have embedded spectral data within every pixel and contain vast amounts of information useful for the characterization of the scanned plant's biological and physical state.

One method of operating a pushbroom camera is to use commercial linear motion control systems that are very accurate but extremely expensive. The base price range is in the thousands of dollars and only increases with an increment in length. The linear actuator usually comes with a great disadvantage in its modularity because the length cannot be easily shortened or extended after installation. Other open-source solutions such as carts have been made but lack the resolution that would be required for a push-broom camera [1]. Therefore, the problem with carts is the stability of the sensors with respect to height; some solutions even have variations in height to approximately 1 cm and vibrate with frequencies up to 20 Hz [1]. This type of oscillatory movement is not acceptable for the push-broom cameras as the movement would interfere with data capture.

Although the demand for technologies capable of rapidly and accurately imaging is high, the actual development of such technologies is behind the need of the demand [2]. It is for this reason that we decided to create a low-cost linear motion control system that is able to facilitate the collection of data using a hyperspectral camera. This paper details the design and construction of the HyperRail, a modular, low-cost, 3D-printed and programmable, linear motion control system; all of the components of this system are fully customizable to the user's specifications for the implementation of this system. It is similar to the design by Susko et al, but our system has no default sensor or camera for data acquisition as it is intended to be used with any system [3]. The implementation of the HyperRail is only concerned with the linear motion of the device it is carrying and thus does not do any kind of data transmission to any central hub or server of any kind in its current iteration.

Although the HyperRail was initially designed to be used for hyperspectral imaging of pine seedlings it has become a platform for other applications and implementations and therefore it is not limited to agriculture and can be implemented in other fields. One example of this is photography and video; a camera can be mounted on the system and used for sweeping shots. Another example is the use for art documentation where a camera is used to scan a painting in order to be able to identify a pigment or examine underdrawings. Lastly, it also has shown effectiveness in environmental sensing applications. A sensor package can be attached to the moving carriage to take measurements at different points to be able to spatially characterize a certain area such as a greenhouse.

2. Hardware description

The HyperRail is constructed with commercially available 1.5 m, V-slot, 40×20 mm aluminum extrusion available in lengths from 0.250 m to 1.5 m. Sections are joined to create longer track and the sensing equipment is then mounted on a carriage system that is sold by the same supplier. The carriage is driven by a fiber of ultra-high-molecular-weight polyethylene (UHMWPE) that coils around a 3D-printed shaft coupler. The electronics of the base are: Feather M0 with RFM95 LoRa radio microcontroller and Big Easy Driver stepper motor driver. For the cloud service functionality to work, the HyperRail must be equipped with a Wi-Fi or Ethernet FeatherWing which is an add-on "shield" for the microcontroller on the base side. For the electronics of the carriage, a Feather M0 with RFM95 LoRa radio will suffice for basic communication with the base for data transmission and movement commands.

The HyperRail also comes in two different forms, Terra and SupraTerra versions. The Terra version is attached to tripods and SupraTerra version is directly attached to the structure where it is being deployed. The SupraTerra was designed for static-long-term deployments over long distances, where the Terra version is designed for quick field deployments where portability is key.

- Hyperspectral imaging with a line scanner
- Spatial characterization of greenhouses and near real-time visualization of spatial data

3. Design files

Design files summary

Design file name	File type	Open source license	Location of the file
TripodConnector2	STL	CERN Open Hardware License	https://osf.io/tq3sb/
LineRoller3Base	STL	CERN Open Hardware License	https://osf.io/tq3sb/
LineRollerSpacerBottom	STL	CERN Open Hardware License	https://osf.io/tq3sb/
IdleRing	STL	CERN Open Hardware License	https://osf.io/tq3sb/
LineRollerSpacerTop	STL	CERN Open Hardware License	https://osf.io/tq3sb/
Line_Tie2	STL	CERN Open Hardware License	https://osf.io/tq3sb/
Motor_mount_top	STL	CERN Open Hardware License	https://osf.io/tq3sb/
Motor_mount_bottom	STL	CERN Open Hardware License	https://osf.io/tq3sb/
Spool_DoubleLIne	STL	CERN Open Hardware License	https://osf.io/tq3sb/
ElectronicsCase_Top	STL	CERN Open Hardware License	https://osf.io/tq3sb/
ElectronicsCase_Bottom	STL	CERN Open Hardware License	https://osf.io/tq3sb/
HyperRailGUI3	pde	GNU General Public License v3.0	https://osf.io/tq3sb/
HyperRail_GUI_M0	ino	GNU General Public License v3.0	https://osf.io/tq3sb/

TripodConnector: STL file used for 3D printing the connector between the tripod and aluminum extrusion.
LineRoller3Base: STL file used for 3D printing one of the components used in the idle pulley assembly.
LineRollerSpacerBottom: STL filed used for 3D printing one of the components used in the idle pulley assembly.
Roller4: STL file used for 3D printing the groove where the fiber will be wrapping around on the idle pulley. This piece will be press fit around the bearing.
LineRollerSpacerTop: STL file used for 3D printing one of the components used in the idle pulley assembly.
Line_Tie2: STL file used for 3D printing the piece that is attached to the carriage and has the fiber tied to it.
Motor_mount_top: STL file used for 3D printing the top half of the motor base assembly. This piece is the one that has the stepper motor attached to it.
Motor_mount_bottom: STL file used for 3D printing bottom half of the motor base assembly. This piece is the one that connects to the aluminum extrusion.
ElectronicsCase_Top: STL file used for 3D printing the top half of the electronics case.
ElectronicsCase_Bottom: STL file used for 3D printing the bottom half of the electronics case.
HyperRailGUI3: Processing file used for synthesizing the graphical user interface.
HyperRail_GUI_M0: Arduino file used as firmware for the microcontroller.

4. Bill of materials

Materials for three-meter systems vTerra

Designator	Component	Number	Cost per unit	Total cost	Source of materials	Material
V-Slot	*V-slot 20 × 40, 1.5 m aluminum extrusion*	2	$20.99	$41.98	OpenBuilds	Aluminum
Tripod	*Amazon 60-inch tripod*	2	$23.49	$46.98	Amazon	Aluminum
V-Slot Gantry Set	*Base to mount all sensors or cameras*	1	$37.99	$37.99	OpenBuilds	Aluminum
Stepper Motor	400 steps/rev motor	1	$16.95	$16.95	Sparkfun	Metal
Big Easy Driver	Stepper motor electronics driver with microstepping	1	$19.95	$19.95	Sparkfun	FR4
Feather M0 Proto	Adafruit M0 microcontroller	1	$19.95	$19.95	Adafruit	FR4
Threaded Inserts	M3 heat-set inserts for plastic	6*	$0.13	$0.78	McMaster-Carr	Brass
M3 Bolt	M3x8 bolts	10*	$0.9	$9.00	McMaster-Carr	18–8 Stainless Steel

(*continued on next page*)

* (*continued*)

Designator	Component	Number	Cost per unit	Total cost	Source of materials	Material
Barrel Jack	Panel mount 2.1 mm DC barrel jack	1	$2.95	$2.95	Adafruit	Plastic
M5 8 mm Screw	M5 8 mm low profile screw	13**	$0.14	$1.82	Openbuilds	Metal
M5 40 mm Screw	M5 40 mm low profile screw	1**	$2.09	$2.09	OpenBuilds	Metal
M5 25 mm Screw	M5 25 mm low profile screw	1**	$1.79	$1.79	OpenBuilds	Metal
Fiber	SPIDERWIRE fishing line 50 lb, 114 *m*	1	$18.99	$18.99	SpiderWire	UHMWPE
Power Supply	12 V DC power supply	1	$12.99	$12.99	Amazon	Other
Ball Bearing	22 mm × 8 mm × 7 mm	1**	$0.14	$0.14	Amazon	Metal
Makerlink	Quad tee nut	6**	$0.09	$5.40	Openbuilds	Metal
Double Tee Nut	Double tee nut	4	$0.69	$2.76	Openbuilds	Metal
Low profile M5 Nut	M5 2.7 mm nut	5*	$0.06	$0.30	McMaster-Carr	Metal
M4 Bolt	M4x10 bolt	1*	$0.07	$0.07	McMaster-Carr	Metal
M4 Nut	M4 3.2 mm nut	1*	$0.07	$0.07	McMaster-Carr	Metal
Spacer	13.2 mm aluminum spacer	4	$3.99	$15.96	Openbuilds	Aluminum
Jumper Wires	Female to male jumper wires	5	$5.99	$5.99	Amazon	Metal
Headers/Crimps	Connector kit	1	$5.99	$5.99	Amazon	Other
Computer	OSX/Windows laptop or desktop	1	N/A	N/A	N/A	N/A

Pricing Notes

- ** sold in packs of 100*
- *** sold in packs of 10*

Additional materials for three-meter system vSupraTerra, 20 cm from attached structure

Designator	Component	Number	Cost per unit	Total cost	Source of materials	Material type
C Beam	2″ × 2″1/16in thick, 12″ long	1	$7.43	$7.43	OnlineMetals	Alumium
Angle Bar	1in × 36in aluminum angle bar with 1/16in thick	1	$5.36	$5.36	HomeDepot	Aluminum
M10 Bolt	M10x25 bolt	3**	$0.39	$1.17	McMaster-Carr	Steel
M10 Washer	M10 washer	3***	$0.12	$0.37	McMaster-Carr	Steel
M10 Nut	M10 nut	3****	$0.11	$0.33	McMaster-Carr	Steel
M5 Bolt	M5x10 bolt	9	$0.09	$0.81	McMaster-Carr	Steel
L Bracket	Double L bracket	3	$1.29	$3.87	Openbuilds	Aluminum
Tee nuts	M5 tee nuts	6*	$0.03	$0.18	Openbuilds	Steel
M5 8 mm Screw	M5 8 mm low profile screw	13*	$0.14	$1.82	Openbuilds	Metal

Pricing Notes

- ** sold in packs of 10*
- *** sold in packs of 25*
- **** sold in packs of 50*
- ***** sold in packs of 100*

Approximate cost for 3D prints for three-meter system (Printed on Fusion3 F400 using 1.75 mm Hatchbox ABS filament

Designator	Components	Number	Cost per unit	Total Cost	Source of Materials	Material Type
Motor mount	Motor_mount_top & Motor_mount_bottom	1	$1.00	$1.00	Amazon	ABS
Tripod Connector	TripodConnector2	2	$0.50	$1.00	Amazon	ABS
Electronics Enclosure	ElectronicsCase_Top & ElectronicsCase_Bottom	1	$5.00	$5.00	Amazon	ABS
Line Connector	Line_Tie2	1	$0.10	$0.10	Amazon	ABS
Idle Pulley Base	LIneRoller3Base, LineRollerSpacerBottom, & LineRollerSpacerTop	1	$0.50	$0.50	Amazon	ABS
Line Driver	Spool_DoubleLine	1	$0.20	$0.20	Amazon	ABS

Any additional length for vTerra will cost ~$45 per 1.5 m of rail. This includes the price of the aluminum extrusion, tripod, and tripodConnector2 3D print.

5. Build instructions

5.1. Three-meter HyperRail vTerra

5.1.1. 3D printing

All of the parts required for this build have been linked in the design files summary section, they can also be directly access through our GitHub repository. The recommended part density, for optimal performance, is approximately 35% infill. Higher density will result in stronger parts but will consume more material. Some light sanding may be required due to 3D printing tolerances.

5.1.2. Track assembly

Start by sliding a TripodConnector2 (Fig. 1) through the end of each section and place it at the center of the beam. This is done to reduce the amount of deflection the beam will experience when the carriage is loaded. The beams will be connected using six MAKERLINKs (Fig. 2) that go on all sides of the connection between the two sections of aluminum extrusion; the MAKERLINKS will be bolted in place using the included M5 set screws. For a minimal-parts assembly, place as demonstrated in Fig. 3. This will assure that the rail is still structurally safe while reducing the amount of links used.

5.1.3. Motor base assembly

Press fit the shaft coupler on to the motor until the shaft is flush with the 3D printed part (Fig. 4). Attach Motor_mount_top to the motor using four M3x8 bolts (Fig. 5). Attach the other motor mount 3D printed piece, Motor_mount_bottom, using

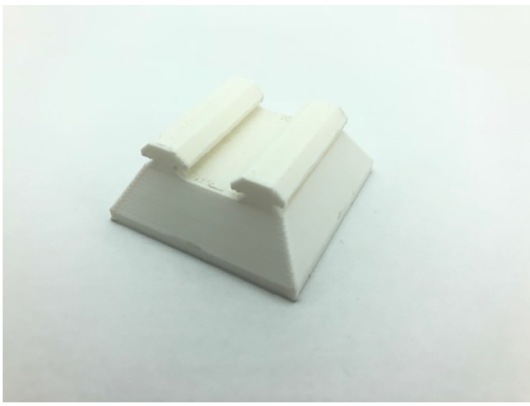

Fig. 1. TripodConnector2.

Fig. 2. Quad Tee Nut-MAKERLINK.

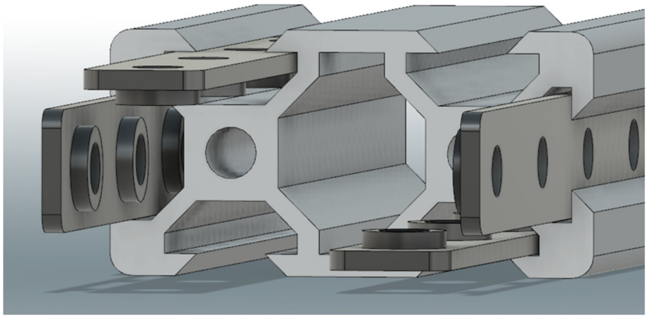

Fig. 3. CAD of the aluminum extrusion and the MAKERKLINKs in their correct locations to minimize the number of the links used. The M5 set screws are not included in the CAD for simplicity.

Fig. 4. 3D printed shaft coupler (white piece) mounted on the motor.

four M5x8 bolt and two double links, to the aluminum beam and place it at the beginning of the (Fig. 6). Connect the two pieces and bolt them using two M5x8 bolts and the final piece should look like Fig. 7. A video animation of the assembly can be found https://youtu.be/BjCgw2147VA.

5.1.4. Idle pulley assembly

First, press fit the IdleRing on to the bearing as shown in Fig. 8. Second, take a LineRollerSpacerTop, Fig. 9a, and LineRollerSpacerBottom, Fig. 9b, and press fit them inside the bearing. Third, take an M5x25 bolt and pass it through all three components. The assembly should look like Fig. 10. Fourth, insert four M5x8 bolts in their corresponding orifices on the LineRoller3Base and bolt them on their corresponding double Fig. 11. Last, place LinerRoller3Base on the bolt and tighten the corresponding nut on the assembly. The final product should look like this Fig. 12.

5.1.5. Carriage assembly

An instructional video for the assembly of the carriage is located here. Make sure to add the 13.2 mm spacer to the carriage, this will make sure that the carriage has enough clearance for the fiber to run underneath the carriage. After assembling the carriage, proceed with the following instructions. The Line_Tie2 (Fig. 13) will be added to the carriage so that the

Fig. 5. Motor_mount_top and the shaft coupler attached to motor using M3x8 bolts.

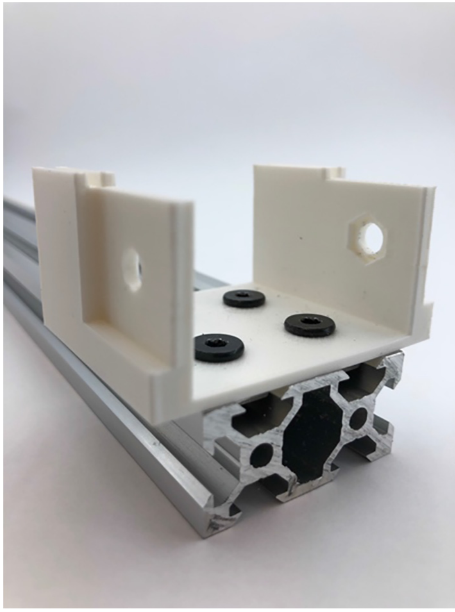

Fig. 6. Motor_mount_bottom attached to the aluminum beam using double links.

fiber can be used in the system. Take two M5x8 bolts and place them through the Line_Tie2 placing it four holes from the edge and center of the aluminum plate with the hook facing the motor side; tighten the bolts with the corresponding nuts. Add an M4x10 through the hole of the Line_Tie2 and add a nut to the bolt but do not tighten it yet. The assembly should look like Fig. 14.

5.1.6. Fiber winding

For easier understanding, this part of the assembly will also have a video, located here, demonstrating the process. With the hook of the Line_Tie2 (Fig. 13) facing away from the stepper motor, insert the carriage on the aluminum extrusion and place it near the end of the rail, the end with the idling pulley. Insert the idle pulley assembly at the end of the rail and tighten it down. Tie a knot around the hook of the Line_Tie2 (Fig. 13). Keeping everything taught, run the remaining line twice around the shaft coupler and then back to the idling pulley. Wind the remaining line around the bolt on the Line_Tie2′s M4 bolt and then pull on the fiber to make it as tight as possible while tightening the bolt (Fig. 15).

5.1.7. Electronics wiring

The wiring of the driver and stepper motor is detailed here on this webpage. The wiring configuration of the HyperRail is summarized in the following lists for a quick setup. Here is a wiring diagram (Fig. 16) and a list of connections:

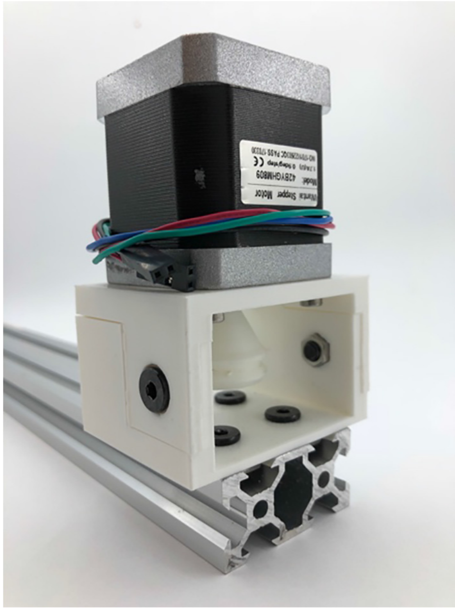

Fig. 7. Finished motor base assembly.

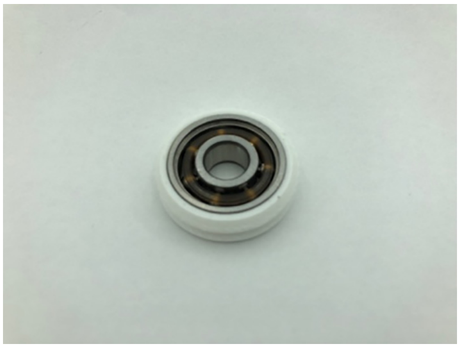

Fig. 8. IdleRing 3D print is press-fit on the ball bearing.

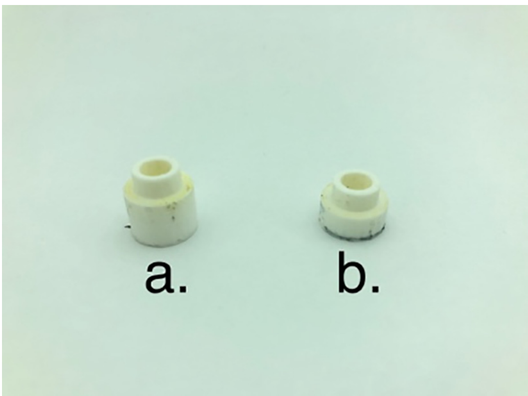

Fig. 9. LineRollerSpacerTop (a) and LineRollerSpacerBottom (b) 3D printed parts.

Fig. 10. Top part of the idle pulley assembly.

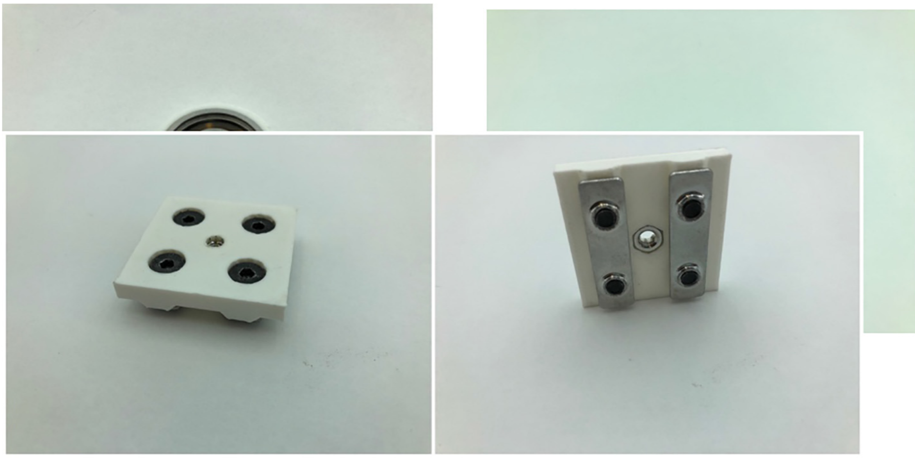

Fig. 11. LineRoller3Base with M5x8 bolts, double tee nuts, and M5 low profile (2.7 mm) nut in the center.

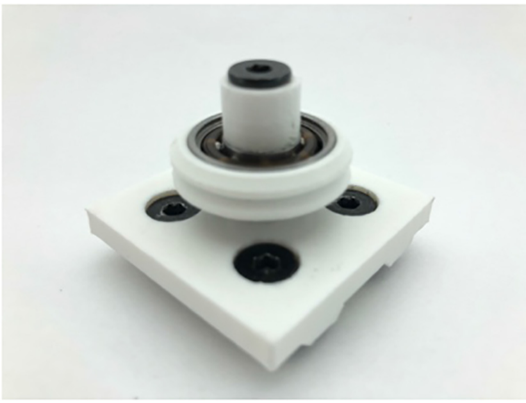

Fig. 12. Completed idle pulley assembly.

Motor → Big Easy Driver

- Green Wire → A
- Black Wire → A
- Red Wire → B
- Blue Wire → B

Fig. 13. Line_Tie2 3D printed piece that will have the fiber attached to it.

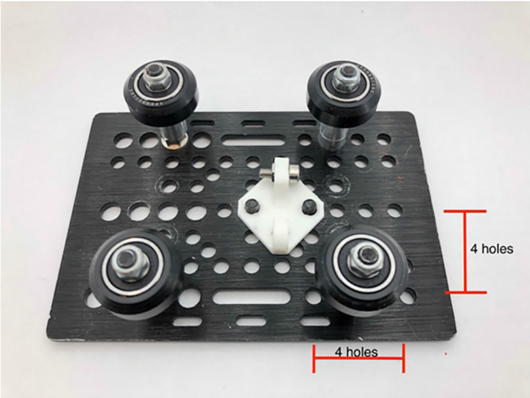

Fig. 14. Line_Tie2 3D printed piece has been placed in its corresponding location. It also has an M4x10 bolt that will be used to tighten the fiber once it has been installed.

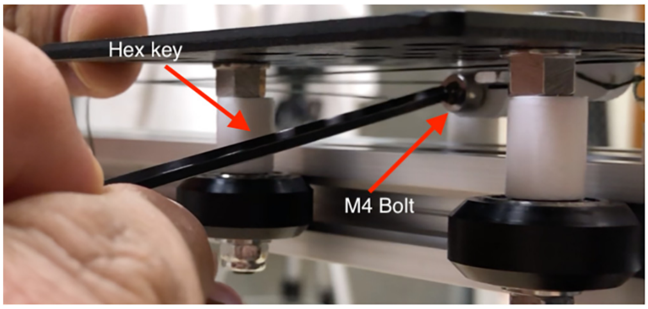

Fig. 15. Fiber tightening on the HyperRail. The hex key is tightening the M4 bolt on which the fiber wraps around.

Microcontroller → Big Easy Driver

- Pin 9 → STEP
- Pin 10 → DIR
- Pin 11 → ENABLE
- GND → GND

Barrel Jack → Big Easy Driver

- V+ → M+
- GND → GND

- GND -> GND

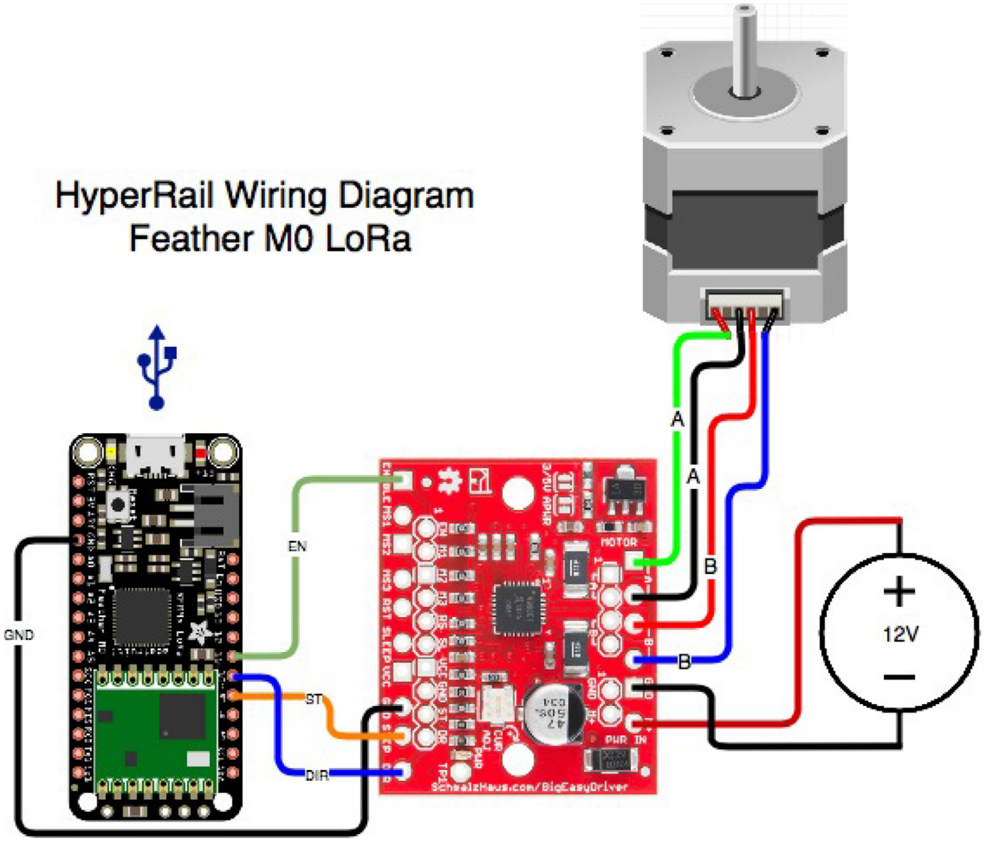

Fig. 16. Wiring diagram of the HyperDrive showing the microcontroller, stepper motor driver, and stepper motor.

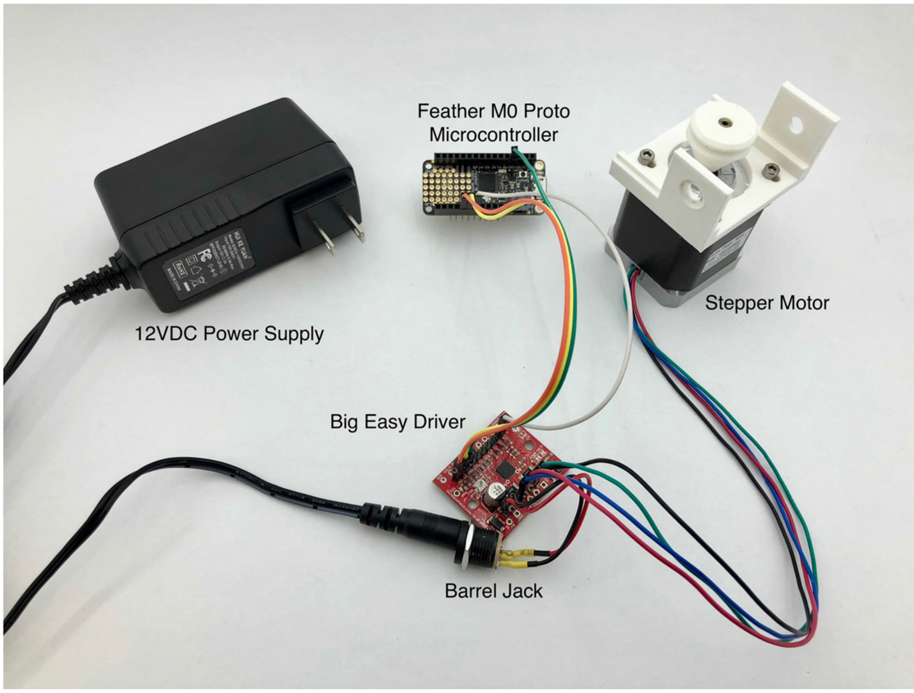

Fig. 17. Wired setup of the HyperRail.

When all the wiring is done, it should look like Fig. 17. The feather MO, barrel jack, and Big Easy Driver will all go to inside the electronics enclosure.

If you start testing the system and the direction of the carriage movement is backwards, swap the wires that are coming from the motor to the Big Easy Driver. In other words, put the wires that go to A in the B location and the wires that go to B now will go to the A location. This is assuming the carriage starts at the side with the motor and it is intended for the carriage to move to the side where the pulley is located.

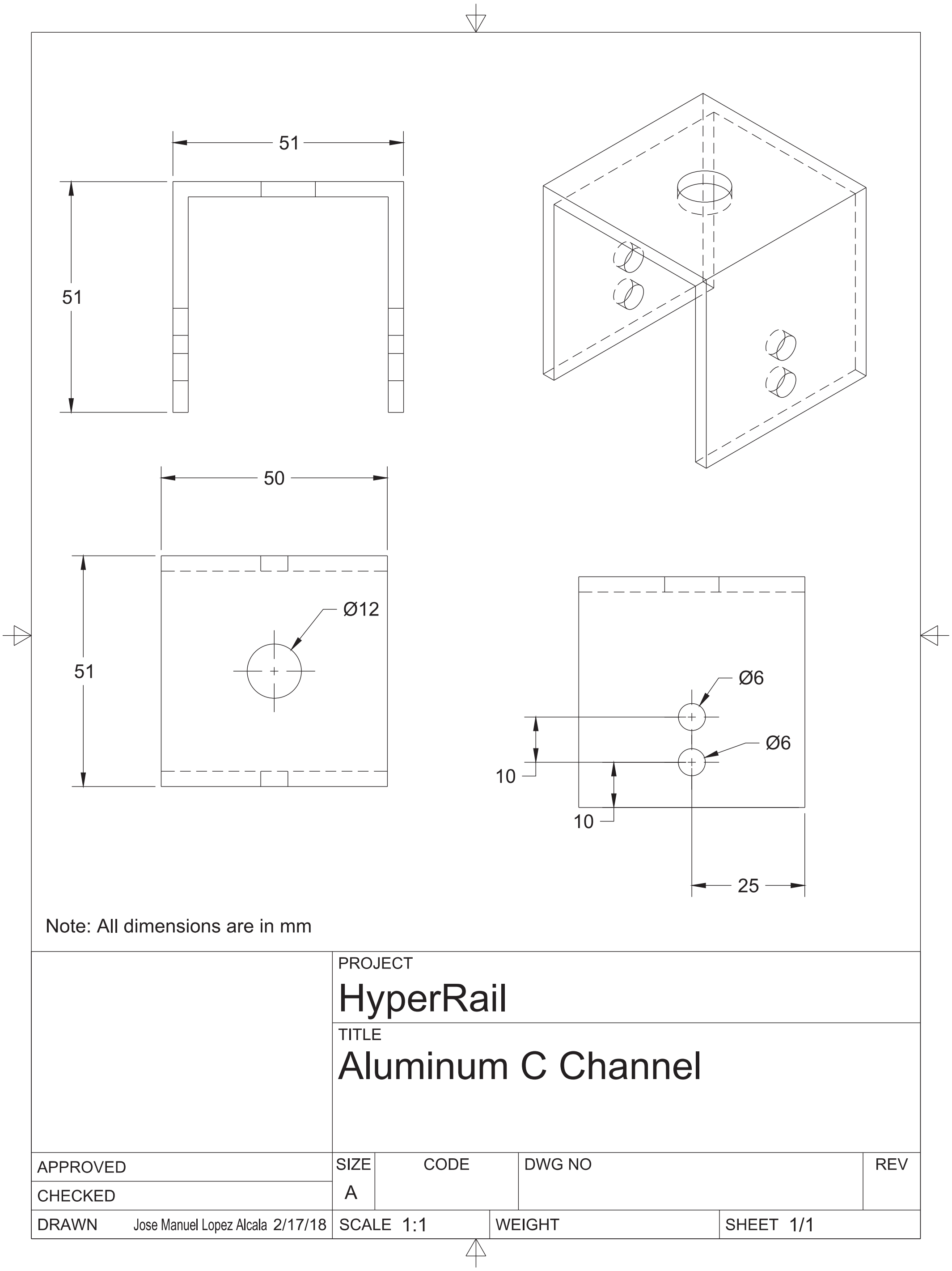

Fig. 18. 2D drawing of the layout of the holes of the aluminum C channel attachments.

5.2. Three-meter HyperRail vSupraTerra

The assembly process for this version of the HyperRail is very similar to vTerra; the only difference is how the aluminum extrusion is attached to the building structure. The following instructions are for a setup in a typical greenhouse.

5.2.1. Aluminum C channel structure attachment

Cut the aluminum C channel in sections of 50 mm; for a three-meter HyperRail you will need three. Then, drill out a 12 mm hole on the backbone of the channel. Finally, drill two 6 mm holes centered on the side of the channel spaced 10 mm from the bottom and each other; do the same on the other side. These instructions are visually detailed in Fig. 18.

After cutting the attachments, begin attaching them on the greenhouse structure starting at the location where the first section of aluminum extrusion will be placed; use an M10 bolt, washer and nut. The next attachments should be space according to the length of the aluminum extrusion. The attachment should resemble Fig. 19.

5.2.2. Aluminum angle bar

The aluminum angle bar will be used to connect the track to the building structure; this will be used in conjunction with the C channel part from the last section and will set the distance the track hangs from the building structure. The aluminum angle bar can be cut to the necessary length but for the purposes of this example we will cut at a length of 170 mm. There are two versions of this piece, the left and right; both of the pieces will be necessary for this build. Three holes are necessary for this piece. Two holes, 6 mm in diameter, are spaced 10 mm apart from each other and centered from the edge of the aluminum piece orientated the long way with the second hole 16.5 mm from the edge. The third hole is located on the opposing side and on the other flap of the L bar; this one is located 9.75 mm from the edge and 15.31 mm from the edge orientated the long way. These dimensions are for the left L bracket; the right bracket dimensions are similar and are further detailed in the following figures. Figs. 20 and 21 show the appropriate diameters and locations for the holes to be drilled at for both the left and right-angle bars.

These angle bars will be attached to the C channel sections in alternating order, i.e. left, right, left, right, etc. Use the M5x10 bolt and corresponding nut to attach the angle bars to the aluminum C channel. The assembly should look similar to Fig. 22.

5.2.3. Track assembly

This section is very similar to the track assembly in Section 5.1.2 but it will not use tripod attachments; therefore, this section will not go into detail about the track connections. For reference or clarification go to Section 5.1.2.

Start by attaching an L bracket, using an M5 tee nut and bolt, near the start of the first section of aluminum extrusion but don't tighten it hard as it might be necessary to move the bracket. See Fig. 23 for reference for the positioning and alignment of the L bracket.

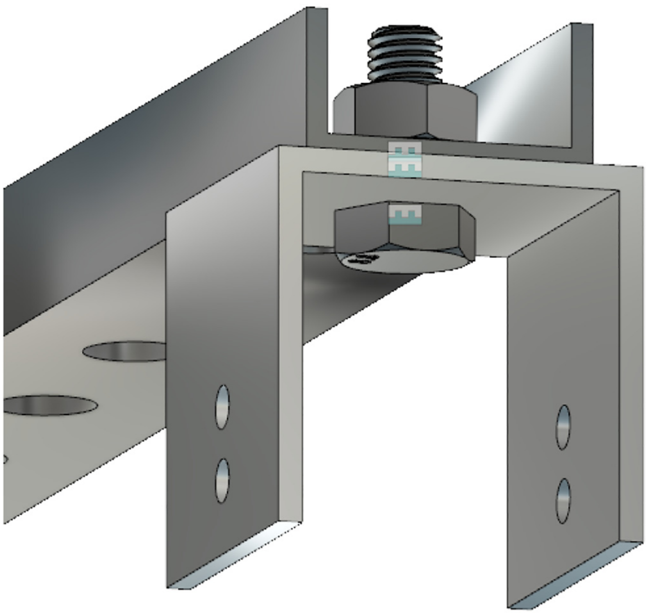

Fig. 19. CAD model of aluminum C channel section attached to greenhouse structure using an M10 bolt, washer, and nut.

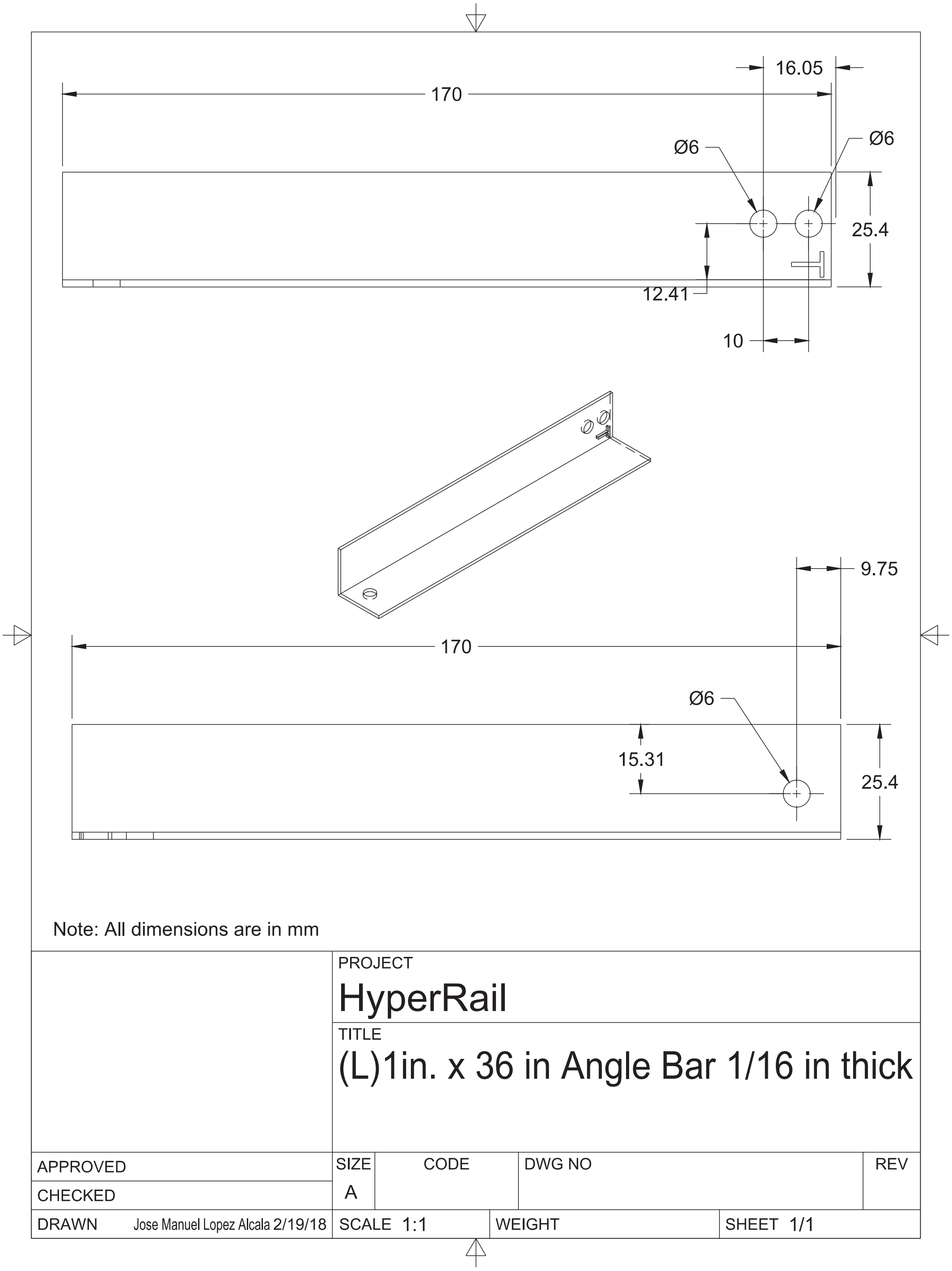

Fig. 20. 2D drawing of left aluminum angle bar. This will be directly attached to the aluminum C channel section.

Next, add the MAKERLINKs to the end of the aluminum beam and attach another L bracket directly on the MAKERLINKs using an M5 bolt. The assembly should now look like Fig. 24.

Now, attach the aluminum beam to the C channel attachments using M5 bolts and the corresponding nuts. Fig. 25 demonstrate the first assembled aluminum beam attached to the building structure.

For the following aluminum beams, only attach the MAKERLINKs and L bracket at the end. Insert the beam into the one that is already attached to the structure and then bolt it using the MAKERLINK's bolts and the angle brackets.

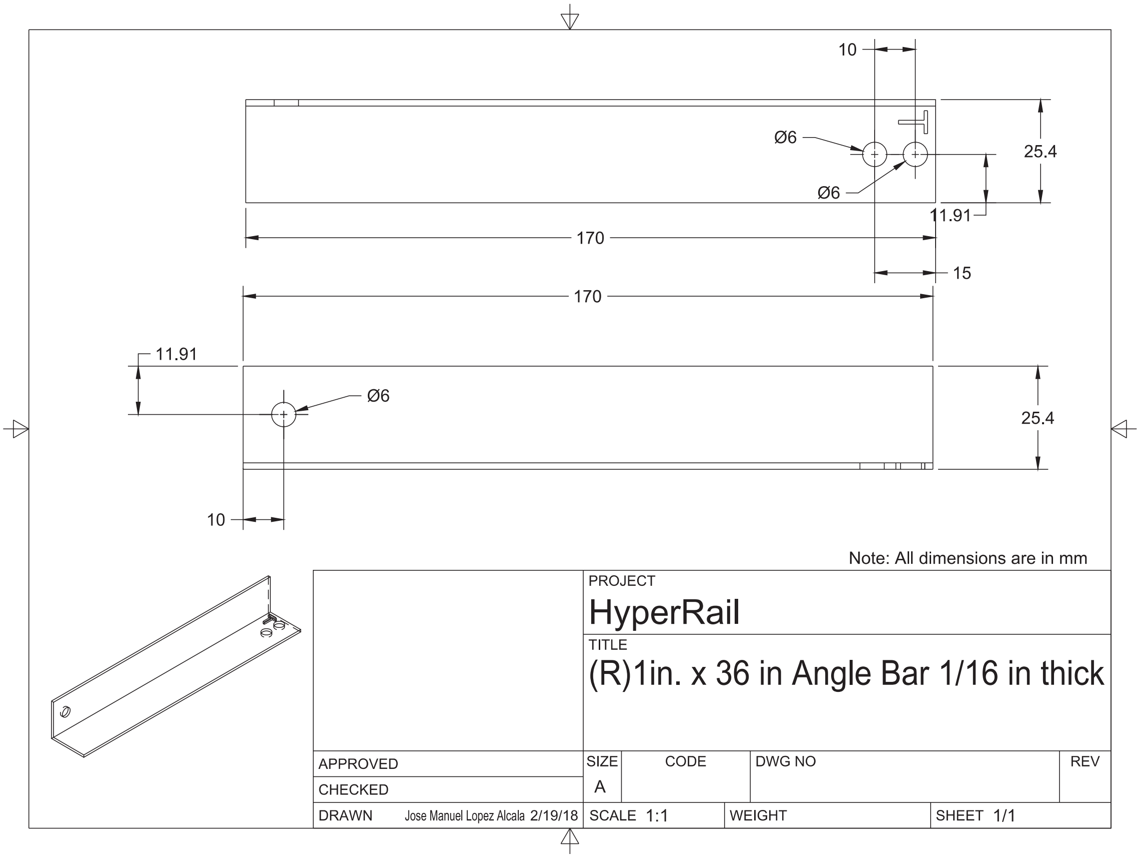

Fig. 21. 2D drawing of right aluminum angle bar. This will be directly attached to the aluminum C channel section.

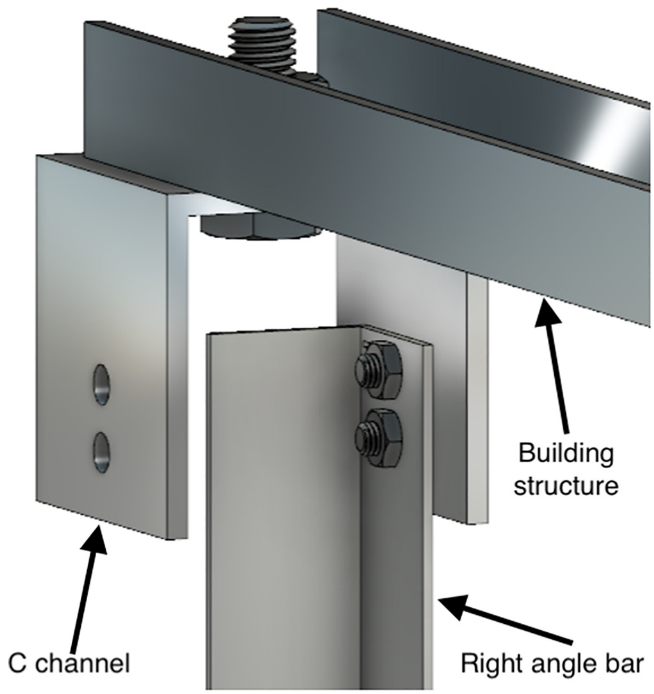

Fig. 22. CAD model of the assembly that includes the C channel and angle bar.

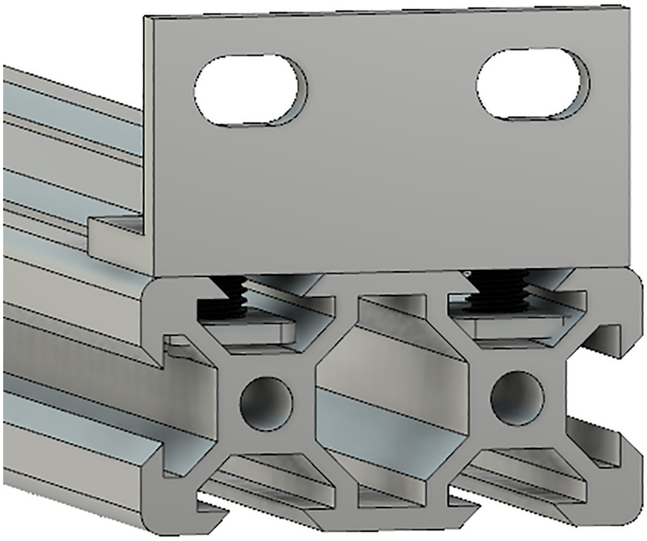

Fig. 23. CAD model of the L bracket attached to the beginning of the aluminum extrusion.

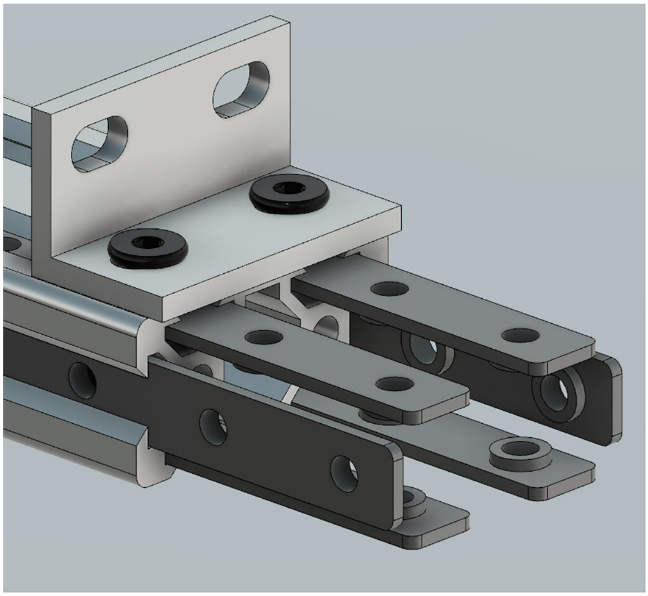

Fig. 24. CAD model of the assembly of the aluminum extrusion with an L bracket attached to the MAKERLINKs.

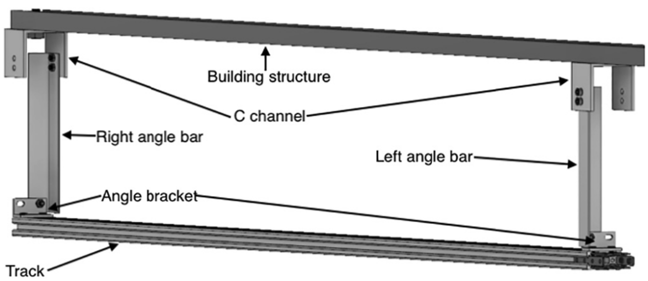

Fig. 25. CAD model of the assembly of the aluminum extrusion attached to the structure of the greenhouse.

Fig. 26. CAD assembly of 1-meter HyperRail demonstrating the assembly of the carriage, motor, and idling pulley.

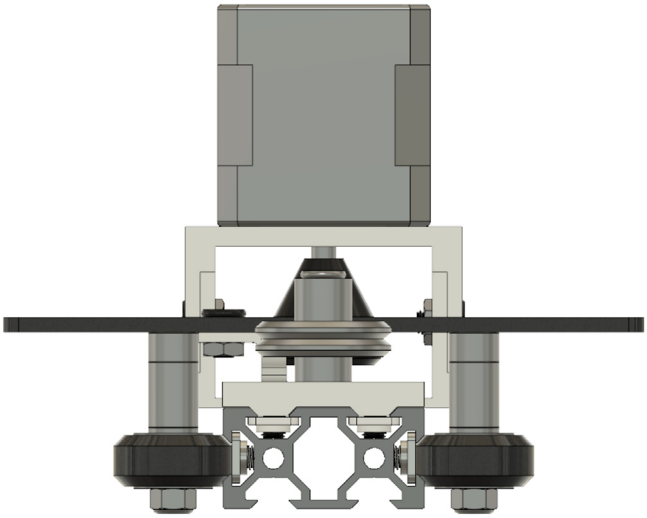

Fig. 27. This CAD model illustrates the positioning of the wheels with respect to the rail. The design of the wheels was created so that they could roll on the aluminum without running into the MAKERLINKs or the bolts that hold them in place.

5.3. Final assembly

Regardless of which version or length of the HyperRail gets implemented, the carriage, stepper motor, and idling pulley will have the same configuration. Fig. 26 is a CAD rendering of a 1-meter version HyperRail showing the complete assembly of the parts previously mentioned.

The actual implementation of the HyperRail will either be on tripods or connected to a structure of the building and the carriage will either be facing up or down, respectively.

Fig. 27 is the back view, the end that has the idling pulley, and it demonstrates the alignment of the pulleys and wheels on the track. In addition to that, it also displays the positioning of the MAKERLINKS that connect the two pieces of aluminum extrusion; these links are also used to connect the idling pulley base to the end of the aluminum extrusion.

6. Operation instructions

6.1. Software download

Download and install both Processing and Arduino IDE's if the software is not already installed on the computer. After installing the software, download the Arduino and Processing files listed in the last two rows of the design file summaries list; these are the source files, also linked here, that are required for the graphical user interface and the firmware that runs on the microcontroller.

6.2. Loading firmware on microcontroller

Loading the firmware on to the microcontroller will be done by using the Arduino IDE. First open the ino file, connect the microcontroller, and then under Tools > Port select the port that indicates the microcontroller, in this case it would be the Adafruit Feather M0.

Fig. 28 shows where to find the tools menu and the ports available. If for some reasons the board is not appearing in the ports list, give your computer a couple seconds before checking again. After selecting the correct port, upload the firmware to the microcontroller by pressing the upload button that has a right-facing-blue arrow inside a white circle in the upper left corner. After uploading the firmware to the microcontroller, the Arduino IDE console should say "Done uploading" (Fig. 29).

The console might have orange text in the black area, but that is not a reason for concern. This text is just extra information that the compiler is supplying for the programmer. At this point, everything should be configured properly on the microcontroller side.

6.3. Loading graphical user interface

The first time that the user interface file is run on the Processing IDE it will produce an error (Fig. 30).

The way to fix this problem is by changing the number "1" in line 89 to the corresponding port number on which the microcontroller is connected to. What this does is select the serial port through which the graphical user interface is expecting the microcontroller to communicate through. The indexing starts at zero and goes up sequentially looking for all the possible connected devices to the computer. In this example the device at index zero is "dev/cu.Bluetooth-Incoming-Port" and "/dev/cu.ThunderFlash-SPPDev" located at index one and so forth; this is shown in Fig. 31 with a blue number. Setting the correct index will require to locate the device that contains "cu.usbmodem" followed by some number that indicates the port on the computer.

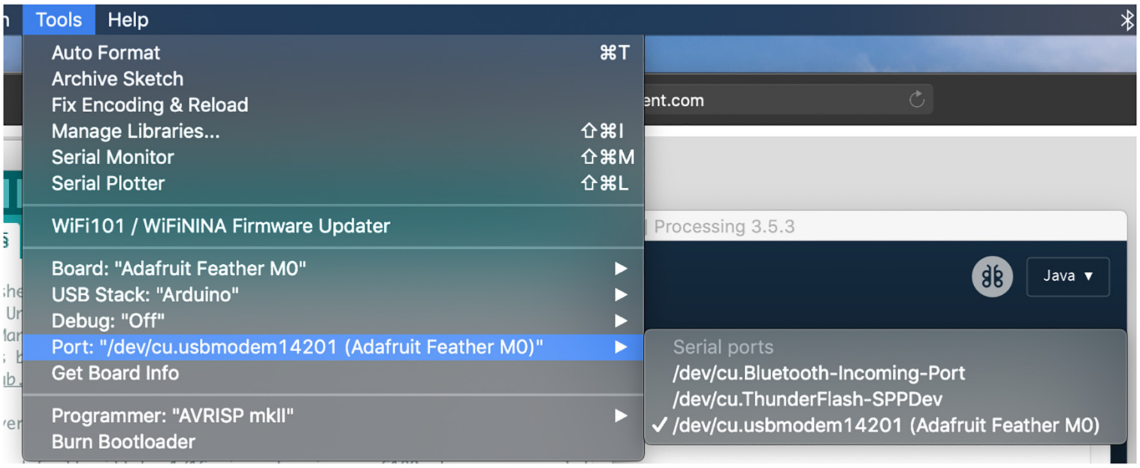

Fig. 28. Tools drop down menu for the Arduino IDE showing the selection of the port to be able to upload the firmware to the microcontroller.

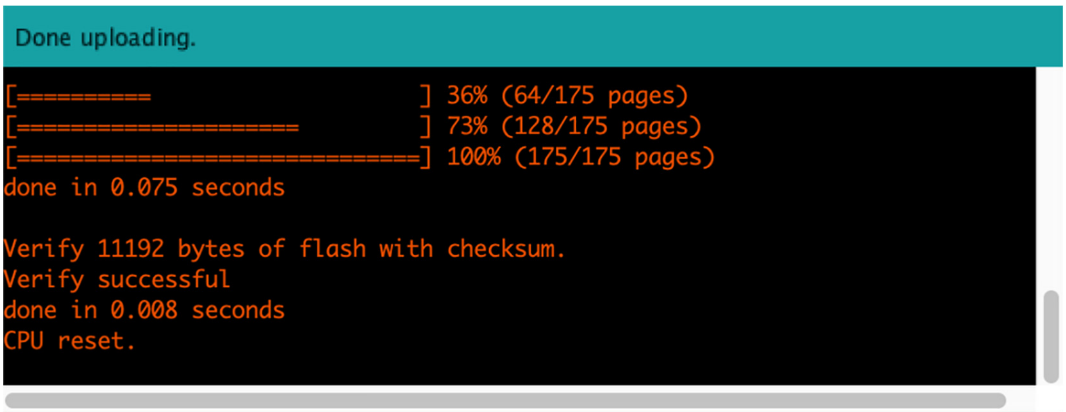

Fig. 29. Arduino IDE console show that successful upload of the firmware to the microcontroller.

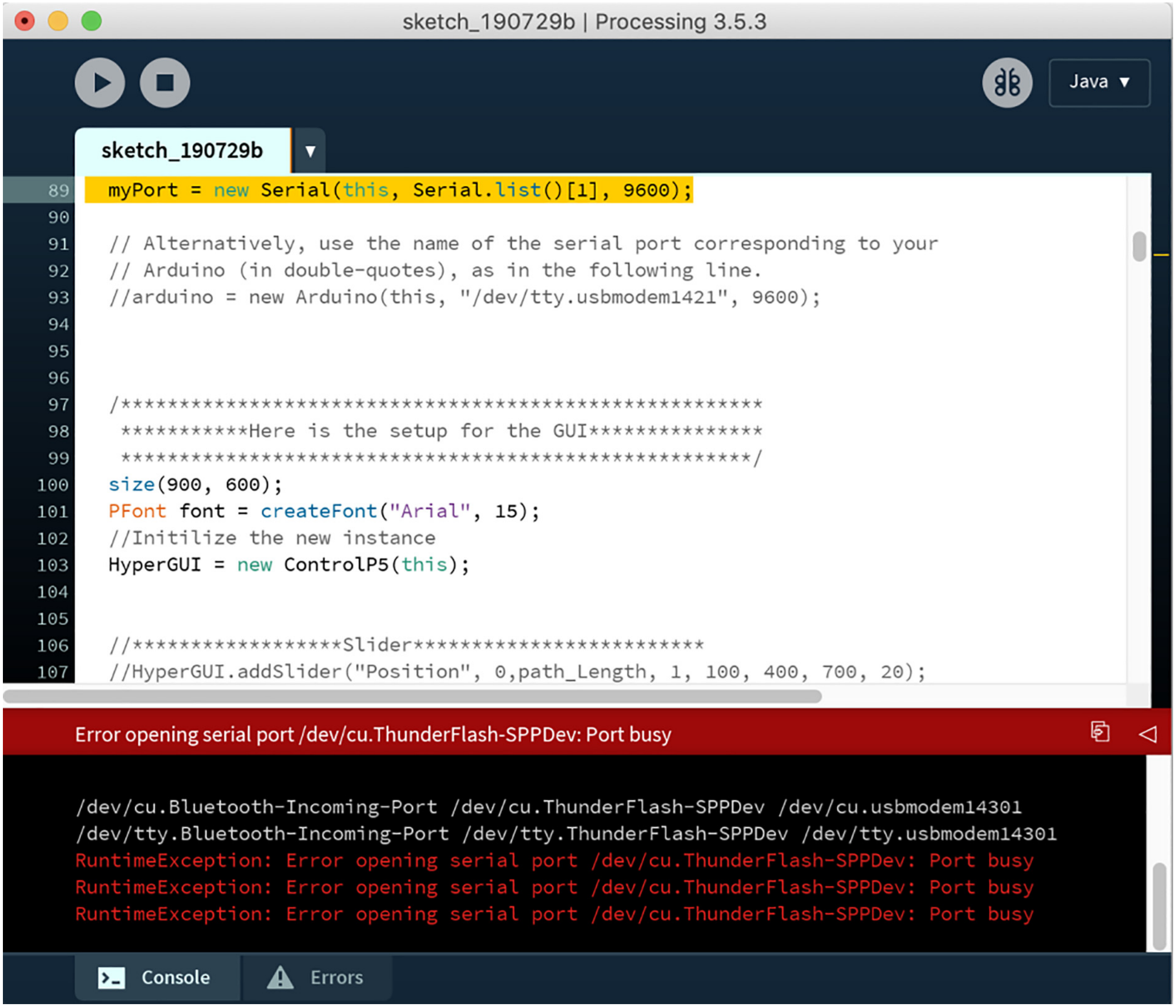

Fig. 30. Processing IDE running the graphical user interface for the first time and producing a communication error.

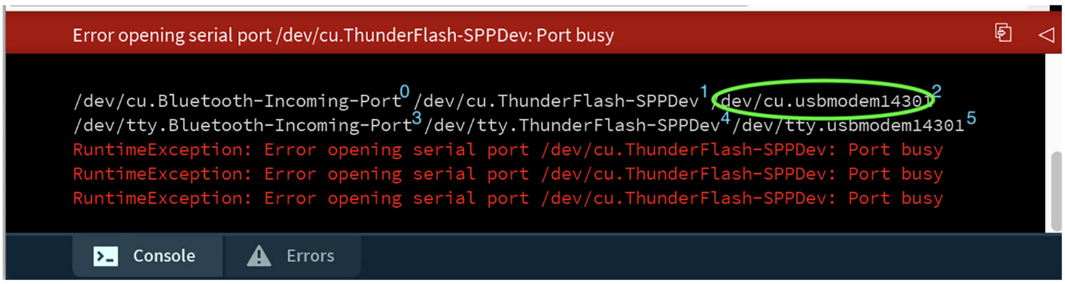

Fig. 31. Processing serial port with list of potential serial ports for incoming devices enumerated starting from zero.

In this example, the port that needs to be selected is "dev/cu.usbmodem14301" and this will tell the graphical user interface that this port will be used to communicate with the microcontroller. To select the port, change line 89 to "myPort = new Serial(this, Serial.list()[2], 9600)". The difference here is the number "2", it replaced number "1". After this change, re-run the graphical user interface and the console should say "GUI setup complete." (Fig. 32).

At this point the graphical user interface is complete and should be ready to go. Fig. 33 shows the graphical user interface when it is up and running.

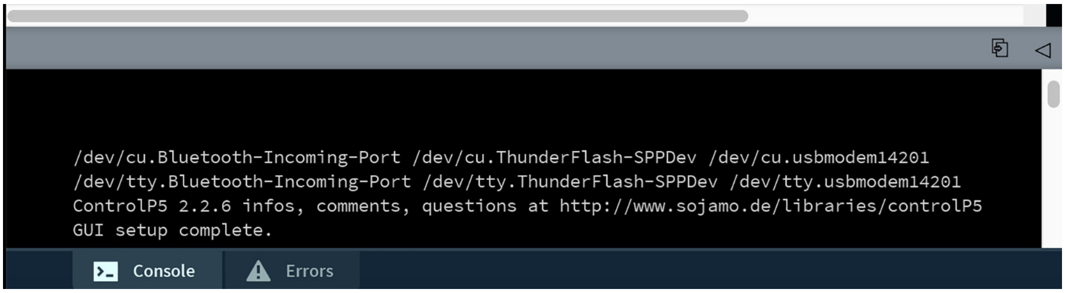

Fig. 32. Graphical user interface console output when the port has been properly selected.

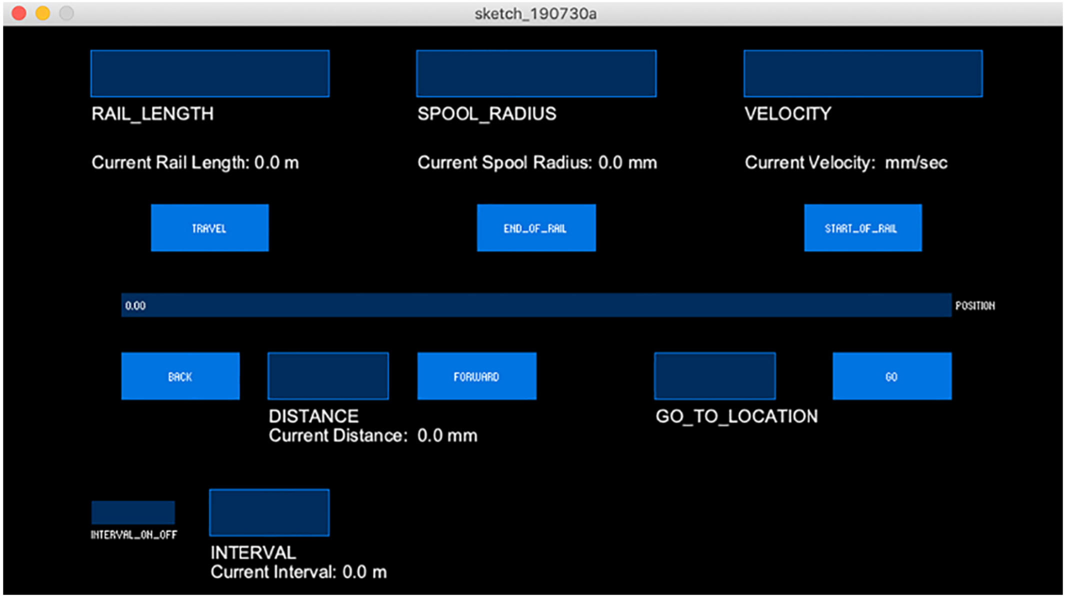

Fig. 33. Graphical user interface successfully running.

After having the firmware uploaded on the microcontroller and the graphical user interface running the system is ready to be used.

6.4. Getting started with HyperRail

First, make sure the HyperRail is powered on. This is indicated with an LED light on the stepper motor driver. The next step is to determine and enter the input parameters; 1) rail length [m], 2) spool radius [mm], and 3) velocity [mm/s] in the graphical user interface. There are three options for the direction of movement of the carriage; 1) travel, a return flight, 2) to the end of rail, and 3) to the start of the rail. Once you hit on the three options the carriage will start moving. During the first trial runs keep an eye on the carriage. It is possible that the carriage is obstructed by manufacturing errors such as a faulty connector between the metal extrusions or wrong assembly of the fiber line. If the carriage is obstructed the fiber line will slip on the wheel of the motor base. There are no known safety hazards.

7. Validation and characterization

7.1. Displacement validation

The first step in validating the performance of the HyperRail was to compare the actual displacement of the carriage to the desired displacement. Since we are using a 400 steps/revolution motor and a 1/16th microstepping driver, we expected to see that one revolution of the motor would result in approximately 6400 steps, but the results showed otherwise. On average, the motor was making one revolution with 6180 steps, which implies that that the theoretical average error is

0.216 mm/revolution. Our testing used a UNI-T UT390B+ range finder which only had an accuracy of 2 mm. This limited our validation of the error, but the performance of the HyperRail did always fall within the 2 mm accuracy of the laser range finder; therefore we can assure that the HyperRail will have a repeatability of at least 2 mm.

7.2. Speed validation

Our second step in the validation was to characterize the speed to be able to compare actual versus theoretical speeds. Since our system did not contain any kind of feedback for the motor rotation, it was only possible to calculate the average speed of the system. Without any compensation, the minimum percent deviation from theoretical speed was 2.89%. The characterization was done on a 1000 mm long rail and with speeds from 5 to 100 mm/s. After a first-degree compensation, on the same range of speeds and distance, the maximum deviation grew to 5.55%. This was because the compensation was not appropriate for the speeds less than or equal to 10 mm/s. In the following trials, the speeds less than 10 mm/s were not considered and had their own compensation curve. For speeds 15 to 100 mm/s, we used second-degree compensation curve, that resulted in a max deviation for 0.75%. Fig. 34 depicts the HyperRail's performance in that range.

Using a separate second-degree compensation curve, the speeds between 0.5 and 10 mm/s had a max deviation of 0.10%, this max deviation does not consider the deviation of 0.5 mm/s otherwise the deviation would increase to just over 1.6%. Fig. 35 illustrates the HyperRail's performance in the this range.

From this data we are able to determine that the HyperRail is able to maintain average speeds very consistently across its good range of speeds and is able to do so within 0.75%. Although the data is not for instantaneous speed, it is still a good

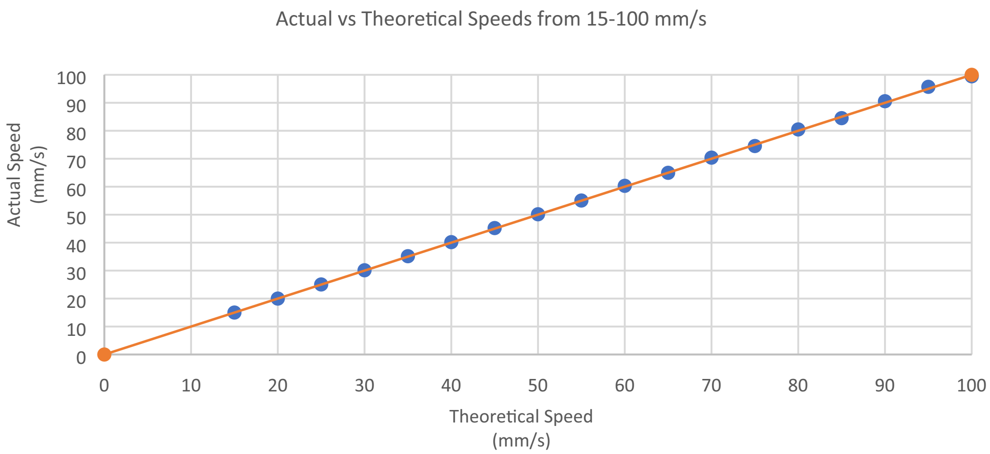

Fig. 34. Graph comparing actual versus theoretical speeds of the HyperRail between 15 and 100 mm/s.

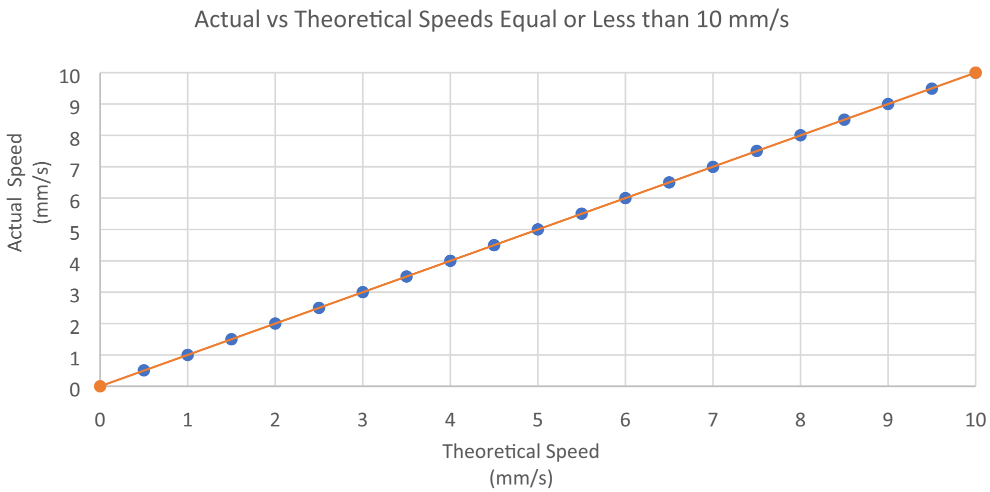

Fig. 35. Graph comparing actual versus theoretical speeds of the HyperRail between 0.5 and 10 mm/s.

indicative of the capabilities of the HyperRail to be able to move a system at a very consistent speed. Trials with the hyperspectral camera have shown its effectiveness and the system has carried out its job as expected. Therefore, we can determine that the HyperRail is a good instrument for jobs that require slow and constant speeds.

8. Future directions

Further calibration is required for the speeds lower or equal to 0.5 mm/s to assure that the average speed is the desired speed. All of the calculations were done on average speeds, this leaves a lot of room for error in the actual speed of the HyperRail. To improve upon our initial design, we will look to implement a shaft encoder to be able to obtain the instantaneous speed of the HyperRail; this will ensure that our calculations reflect the performance of the HyperRail at a more precise level. In addition to this, wireless-bidirectional communication capabilities are being developed to able to relay data back to the hub to be able to upload data directly to a cloud-based system.

9. Human and animal rights

The work does not use any human or animal subjects.

Funding

This work was supported by the USDA National Institute of Food and Agriculture, Hatch Project N118HFPXXXXXG055.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A.L. Thompson, A. Conrad, M.M. Conley, H. Shrock, B. Taft, C. Miksch, T. Mills, J.M. Dyer, Professor: a motorized field-based phenotyping cart, HardwareX 4 (2018) e00025, https://doi.org/10.1016/j.ohx.2018.

[2] U. Lee, S. Chang, G. Putra, H. Kim, D. Kim, An automated, high-throughput plant phenotyping system using machine learning-based plant segmentation and image analysis, PLOS One 13 (4) (2018), p. e0196615. Available at: http://proxy.library.oregonstate.edu.ezproxy.proxy.library.oregonstate.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=aph&AN=129351017&site=ehost-live.

[3] A.Q. Susko, F. Gilbertson, D.J. Heuschele, K. Smith, P. Marchetto, An automatable, field camera track system for phenotyping crop lodging and crop movement, HardwareX 4 (2018) e00029, https://doi.org/10.1016/j.ohx.2018.e00029.