What next?

This project has a lot of remaining possible ToDo's which could be done. While doing the project a lot of ideas came up and even more ideas were found by other creative heads on the web. This page contains a collection of unfinished things and good ideas for general Hangprinter and Trikarus improvements. This site might not be complete and it can happen that you find the one or the other additional idea / information on some of project's sub pages.

Recent ToDo's

Software / Firmware / Hardware

- finish IMU 9250 monitoring and trigger routines (alerting) for unwanted effector tilts
- finish MPU 6250 monitoring of vibrations
- fix the filament feed encoder at the tool head (not working mechanically right now)
- test out the Z limit switch and try to measure maximum movable Z height with it
- create some script to perform dry-run movements within the build envelope to validate the reachable coordinates (generate dryrun gcode to move circles with different diameter in different heights)
- monitoring / statistics
  - create dashboard from datasources to collect information about total print time, finished print jobs, printed filament amount (we can use information like Feed Encoder and Repetier Server job management). Add statistics for per day/week/month
  - make influxdb aggregation files to reduce the process load (push sets of data instead of single data lines)
  - add some alert for non-moving filament while printer is printing - compare feed encoder data (time between pulse, length, ...) with targeted extrusion value from Repetier or Duet API
  - Add bluetooth connection monitoring
  - Add I2C monitoring
- make more convenient scripts using less bash and more python / keep it consistent
- create some horizontal cable holder to force main cable movement (protect cable to collide with print object below effector in case the cable falls down the effector)
- Grafana: fix line buildup formula for calculation of position
- Fix possible python USB serial write timeouts

- Add some physical buffer between webcam and machine frame to reduce wobbling while making snapshot images (time lapse)
- Repetier Server 1.0.4 - add GPIO operations

Calibration/documentation

- document the anchor point measurement methods
- document how to find best line buildup compensation (Q and R factors)

Slicing

- Build volume
  - Make a parametric OpenSCAD 3D model which displays the maximum theoretical build volume shape of Hangprinter, based on the entered anchor point locations (kind of (hyperbolic) tetrahedron).
  - Add some cutoffs for the existing bed shape + the slicer shape (cylinder) + effector/transportation height cut off (dead height)
  - make a feature request for PrusaSlicer (and other slicers) to support Hangprinter build volume shape
- Check out 3doptimizer.com
- adjust MVS (max. volumetric speed)
  - https://grabcad.com/tutorials/dialing-in-a-filament-and-specifying-the-max-volumetric-e-xtrusion-value
  - https://help.prusa3d.com/en/article/max-volumetric-speed_127176
  - https://projects.ttlexceeded.com/3dprinting_techniques_calibrating_volumetric_rate.html

Printing

- Check out the recover print functions and the different possibilities using Repetier Server and DWC
- Check out filament changing routine
- add some heating for print (higher layer might warp excessively)

Testing/monitoring

- check out how long lines live until they have to be replaced
- long-term monitoring of prints and correlating values (Smart Stepper errors, vibrations, tilts, room temperature, ...)

Recent hardware

- find out life time of UPS batteries

Ideas for better prototype hardware/ electronics

Because Trikarus is installed at a public place we cannot change most things now ("never touch a running system") but we can think about some generic things. Things listed below are topics to be done in future and hopefully by a large bunch of people

- Measuring / calibration
  - ~~check out the~~ integrate auto calibration routine ~~and what's wrong with it~~ → use tobben's hp-mark implementation
  - Make it easier to get anchor point coordinates manually (might require new anchor hardware)
- Wifi / Switch / Router
  - use a smaller router or some FritzBox. Those can be used to repeat Freifunk network or any other SSID with total ease
  - buy one with SIM card slot → maybe use Freenet Funk or netzclub.net
  - do not use Freifunk as main network because it is limited to the city
  - use one with 2.4G + 5G or 5G only
  - use one with more than 2 meters Wifi range (when a lot of other nets are around) than the used TP-Link - connection between LifeTab Tablet und Router is nearly impossible and unstable
  - add a more recent Android tablet to Trikarus which allows to attach a USB LAN Interface. So we can switch off the second SSID created at our top Freifunk router
- Line guiding
  - omit self-twisting lines while installing them on spools and anchors
  - reduce the maximum wrap-around length of the line over V shape bearing
  - generally switch from V shape bearings to U shaped ones (better for life time)
  - search for smaller ceramic inlets (or make some custom parts using PTFE sticks)
  - reduce friction to let it behave more smooth like in https://vimeo.com/227891846 (maybe it has to be stronger because we use doubled line feature and a lot of ceramic inserts and bearings)
- Drives and spools
  - monitor temperatures of Smart Steppers
  - monitor temperatures of Nema 17 motors
  - add some small protection frame for the connected wires at the terminals at Smart Steppers

    - add thin grooves on the spool surface so line buildup is more controlled
    - make spools wider so the line cannot buildup on each other but only next to each other (only deviation: line is switching from left to right → angle must be calculated)
    - make drives with brakes → Drive motor brakes
    - redesign the spools to have coils which can be charged with line independently (gives better control over tension). We could install one motor per line and mirror the step signals. Would be a heavy printer ceiling module we but could control all motors precisely regarding to line errors (tension)
    - another idea: design coils which are so thin that the line cannot lay over each other with criss-cross (0.55 mm thin spool - just 0.05 mm thicker than the line) → large buildup but maybe more exact
    - another idea: make drive which does not require buildup compensation because we store the line on a separate spool with retraction system (like draw wire encoders or VR retraction system for main cable). Let the line move over a well shaped spiral on a shaft which exists to transmit the line straight forward (line must we tensioned all the time so the winding on the spiral does not get lost. maybe some buffer (sleeving) could be pushed over the spiral)
- Filament feed system
    - make some active filament feed system for large spools (> 5 kg) for the ground (because mounting over head is not nice) → Bluetooth Arduino Filament Feeder, Weigh and Monitor
        - make all known spools compatible
            - Redline Filament spool 8.5 kg → 335 x 168 mm, 13 mm thick border, inner diameter 36 mm
            - Das Filament spool 2.6 kg → 300 x 100 mm, ?? mm thick border, inner diameter 50 mm
            - Prusa Filament spool 1 kg → 200 x 90 x 4.5 mm thick border, inner diameter 50 mm
- General electronics / cable management
    - do some well prepared cabling system
    - reduce system to basic machine requirements (remove LED stripes and spots) or if installed LED things we could use PWM stuff to dim the light
    - add a better emergency halt wiring for portable use
    - replace relays to 3.3 V optocouplers (better compability with Raspberry Pi GPIO)
    - rethink the situation with long cables and data synchronization using I2C → see this hackaday.com post
    - use fans with tacho signal to monitor fan speed (security feature)
- Portable machine frame

  - fix tolerances in steel construction
  - change gas compression spring (lost their pressure)
  - add additional springs to the gas springs to have some force at the lower positions
  - replace quick releases with some mechanism to fold the frame without needing a ladder - current quick releases can easily be destroyed by folding frame without releasing them first (shear-off)
  - use less different neodymium magnets for clips
- Raspberry Pi's
  - add second Raspberry Pi which only handles monitoring (InfluxDB). This will reduce the load on the print server Raspberry Pi and this reduces overload peaks which result in halted printed (for some seconds)
  - upgrade to use Raspberry Pi 4 with more performance (recent Pi has much system load when rendering time lapse videos. This makes the system unresponsible for longer time)
  - add active cooling fans (temperature > 60°C reached really often)
  - add active (fans) or passing cooling (air slots in ceiling plate) for stepper motors which get hot after time
  - add remote Reset/Power Down of Raspberry Pi using PEN/RUN pins
  - use a non-backfeeding USB hub for connected devices →

    https://elinux.org/RPi_Powered_USB_Hubs
- UPS and USB wiring
  - omit undervoltage events using appopriate thick short cables and well rated UPS (we were not possible to suck out the required power from the recent UPS)
- Extruder
  - add better cooling. When printing quick (60 mm/s and more) with SuperVolcano the cooling is unsuffiecent. 2 or more fans should help to get better surface quality
  - add system for quick tool changing (e.g. magnetic) → + make pluggable wiring for extruder
  - add redundant PT100 sensor (second)
- Effector
  - change amount of fixing screws from 2 to 1 thumb screws for "felt fixators" (easier installation)
  - add rubber buffers at the corners which secure effector in case it's in free fall (tearing lines)
  - add system for quicker line changing
- Bottom anchors
  - add spirit levels to each fork for quick leveling at the ground
  - add one guitar tuner per anchor (directory to the anchor) instead of wiring A1/A2, B1/B2 and C1/C2 to a common tuner. That would give better manual

    tension control and helps to easier adjust rotation of effector around Z axis.
  - avoid lines jumping off the bearing when untensioned (fixate using another ceramic inlet for example)
  - we should test out what could better: one or two fixed tuners for the lines or "floating tuner" which has a combined knot for A1/A2, B1/B2 and C1/C2. Guessing the last would decrease accuracy because lines per pair will normalize their tension but that might reduce positioning exactness.
- Heated bed
  - design one. Maybe use underfloor heating of the building itself or use something with hot water circulatory
- Laser pointers
  - make the beam adjustable remotely using deflection mirros instead of screwing down with set screws
- Z limit switch
  - simulate switch using Smart Stepper (sensorless homing using error +/-) or perform contactless using IR for example
- Security
  - add PIR sensor or laser sensors to react on people entering the working area (build volume) - e.g. halt print, make some sound or print slower for example
- Duet / RepRapFirmware
  - add some routine to RepRapFirmware to disallow movements which are not in the build volume. At the moment we can try to reach unreachable coordinates. This results in line mess on spools)
  - make I2C/TWI compatible with Duet + Smart Steppers → use TWI support of Duet (TWD0, TWCK0 pins on expansion header) and send and receive I2C commands with M260 see also (forum.duet3d.com)
    - M114 S1 should work then which would help a lot for Building basics, checks, maintenance - we could convert the Repetier Server Smart Stepper control mode scripts into GCode aliases (see Smart Stepper - calibration and control modes (sPID mode, pPID mode and torque mode)
  - Udapte to RRF 3 → see https://forum.duet3d.com/topic/14215/hangprinter-help-on-duet-3/10
  - have a look into Klipper Firmware which seems to support cable winch driven printers too → https://github.com/KevinOConnor/klipper
  - upgrade hardware to HangPrinter v4 which supports native stock firmware soon instead custom firmware → "Later on, HP4 support will hopefully stabilize and be merged into all major firmwares, and this directory can simply contain instructions and configuration recommendations." (https://gitlab.com/tobben/hangprinter/-/tree/version_4/firmware)

  - wire Gyro/IMU directly with Duet and implement algorithms to control the printer using I2C data (real time data)
  - **Home position using IMU (idea from https://www.appropedia.org/Clerck,_a_RepRap_3D_printer_hanging_from_the_ceiling)**
    1. If printer is not horizontal, tighten D-lines until it is **(even if the effector was configured to be horizontal it could change its levelness while printing because the D spool spools up the 3 D lines unevenly)**
    2. Lower printer (extend D-lines) until hot end crashes into print surface
    3. Set D-length. D-axis is now calibrated
    4. Extend D-lines an additional 2 mm
    5. While not horizontal:
       1. Calculate direction of inclination
       2. Tighten A, B or C to counteract inclination
    6. Tighten D-lines 2 mm
    7. We are now at home position, all axes calibrated
  - **Automatic slack line compensation idea using IMU (idea from https://www.appropedia.org/Clerck,_a_RepRap_3D_printer_hanging_from_the_ceiling)**
    1. Compute some quantity describing "expected jerk" from gcode (Marlin and other firmware already does this. In Marlin the value would be tucked into the block_t struct)
    2. Use IMU to record "observed jerk" (again tucking it into block_t in Marlin)
    3. Write a function `F(expected jerk, observed jerk)` → `Move_to_tighten_strings`, and call it once in a while
    4. since cheap 6-axis IMUs are really good at recording rotation, I expect the notion of "jerk" to include rotation (with expected rotation always 0). Timing will be really important, but Hangprinter-Marlin already has five `nop` operations in the middle of the stepping code (see the workhorse function in `stepper.cpp`), that could be utilized for IMU readings at precisely the right times. The `nop`s are just there to wait for drv8825 stepper driver chips anyways.
- Force / tension sensing
  - add some kind of tension sensing system for all nine ABCD lines. The information can be used to do emergency halt if force exceeds limits (like line tripping) or to compensate the error
  - add some constant force springs to the ABC lines to create better tension
  - we cannot use the Smart stepper torque for this because we cannot guarantee the root cause of that event (e.g. line jumped off the bearing). We

    can use error pin of smart steppers for emergency halt. default errorlimit is 1.8 degrees. exceeding the configured limit will turn the yellow LED lighting on the PCB and the error pin will be set to 1
  - maybe add some encoders at each pivot bearing. We could sync the measurement values and could detect issues in motion system like teared lines, greater offsets, etc.
- Absolute positioning
  - ~~integrate draw wire encoder per drive for absolute positioning (in case the bottom anchors are rigid we can use this for absolute XYZ positioning maybe)~~
  - ~~"April Tag" Localization System~~
  - ~~libsurvive (Open Source Lighthouse Tracking System)~~
  - ~~vivetracker~~
  - ~~hivetracker~~
  - ~~pozyx.io~~
  - ~~camera tracking system with LEDs on the effector~~
  - tobben's hp-mark implementation

Version #1
Erstellt: 16 Mai 2025 13:55:45 von Mario Voigt
Zuletzt aktualisiert: 16 Mai 2025 13:58:00 von Mario Voigt