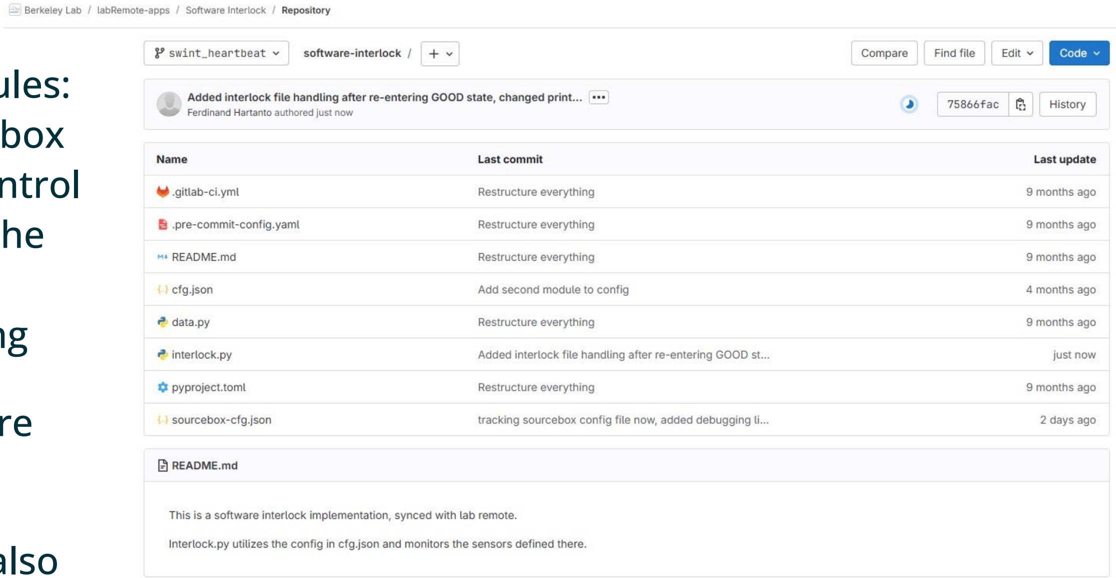
Software Interlock Slides

From Feb 7, 2025 DICE Meeting

Software Interlock (Sourcebox)

Ferdinand

- Need to do module QC for ~400 modules:
  - will have 4 coldboxes + 1 sourcebox
  - need to be able to monitor & control each setup so they don't break the modules:
    - HW interlock (hard breaking point)
    - we have SW interlock before reaching HW interlock
- Working on code mostly in software-interlock repo BUT have to also change scripts and configs in pixelmoduleqc repo
- Made my branch 'swint_heartbeat' in both repos (up to date)



Added interlock file handling after re-entering GOOD state, changed print...
Ferdinand Hartanto authored just now
75866fac

Name	Last commit	Last update
.gitlab-ci.yml	Restructure everything	9 months ago
.pre-commit-config.yaml	Restructure everything	9 months ago
README.md	Restructure everything	9 months ago
cfg.json	Add second module to config	4 months ago
data.py	Restructure everything	9 months ago
interlock.py	Added interlock file handling after re-entering GOOD st...	just now
pyproject.toml	Restructure everything	9 months ago
sourcebox-cfg.json	tracking sourcebox config file now, added debugging li...	2 days ago

README.md

This is a software interlock implementation, synced with lab remote.

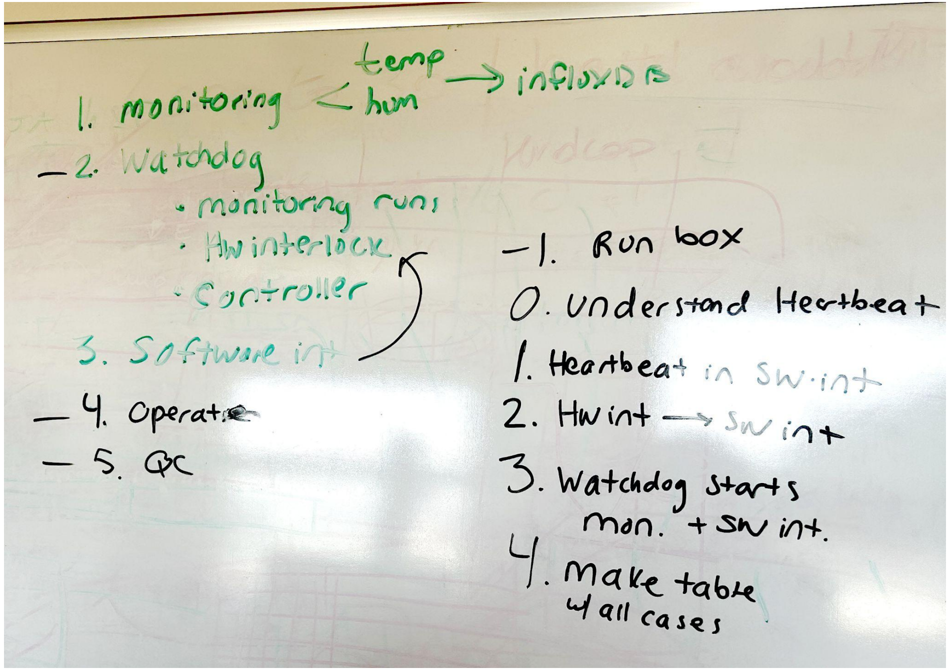
Interlock.py utilizes the config in cfg.json and monitors the sensors defined there.

- Intro to SW interlock:
  - Functionality
  - Improvements (what I worked on)
  - Goals

Software Interlock - Intro

Ferdinand

- Monitor, Controller, & Watchdog are part of `coolman.py` (pixelmoduleqc repo)
- Run SW interlock using `interlock.py` (software-interlock repo)
- `coolman` & `interlock` are labRemote apps
- SW interlock currently monitors 3 quantities:
  - Module NTC
  - Chuck dewpoint (HYT + NTC)
  - Box dry air flow
- Need to run monitoring, wd, and swint before operating the box (script enforced)



1. monitoring < temp, hum → influxDB

– 2. Watchdog
  - monitoring runs
  - HW interlock
  - Controller

3. Software int

– 4. Operate

– 5. QC

–1. Run box

0. Understand Heartbeat

1. Heartbeat in SW·int

2. HW int → SW int

3. Watchdog starts mon. + SW int.

4. Make table w/ all cases

Software Interlock - Functionality

Ferdinand

- Code in `coolman.py` (pixelmoduleqc repo in labRemote-apps):
  - main script (lots o' imports)
  - has different options based on what you want to run ("monitor", "watchdog", "operate", so on)
- HeartBeat class indicates that a process is live:
  - creates & locks files for each process
  - issues timestamps (beats)
  - can record status/data
- Watchdog class watches process health:
  - retrieves heartbeat from processes (monitor, controller, HW interlock)
  - turn off controller when HW int triggers
  - restarts dead processes (e.g. monitoring)

- Code in `interlock.py` (software-interlock repo in labRemote-apps):
  - sets up configs (2 config files: labremote & interlock)
  - has Quantity, Sensor, & SoftwareInterlock classes
  - defines quantities/sensors to monitor from config
  - checks quantities via InfluxDB queries
  - compares quantities with bounds from config
  - shuts down system if quantity out of bounds (or lost)
  - sends email alert

Software Interlock - Improvements (My Work)

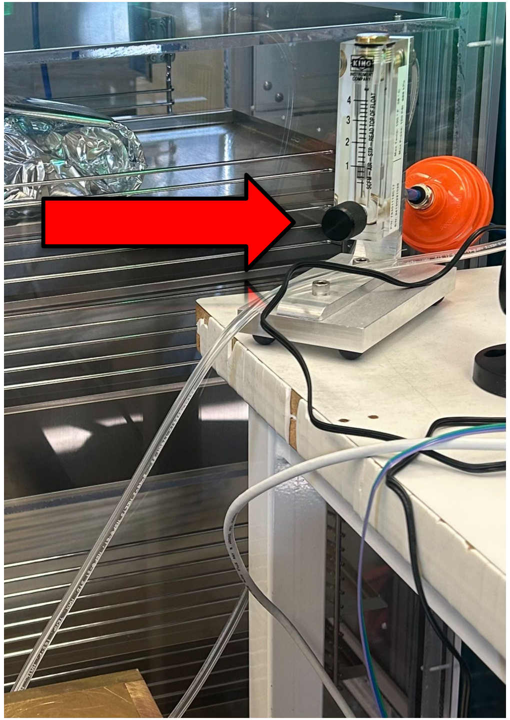
Ferdinand

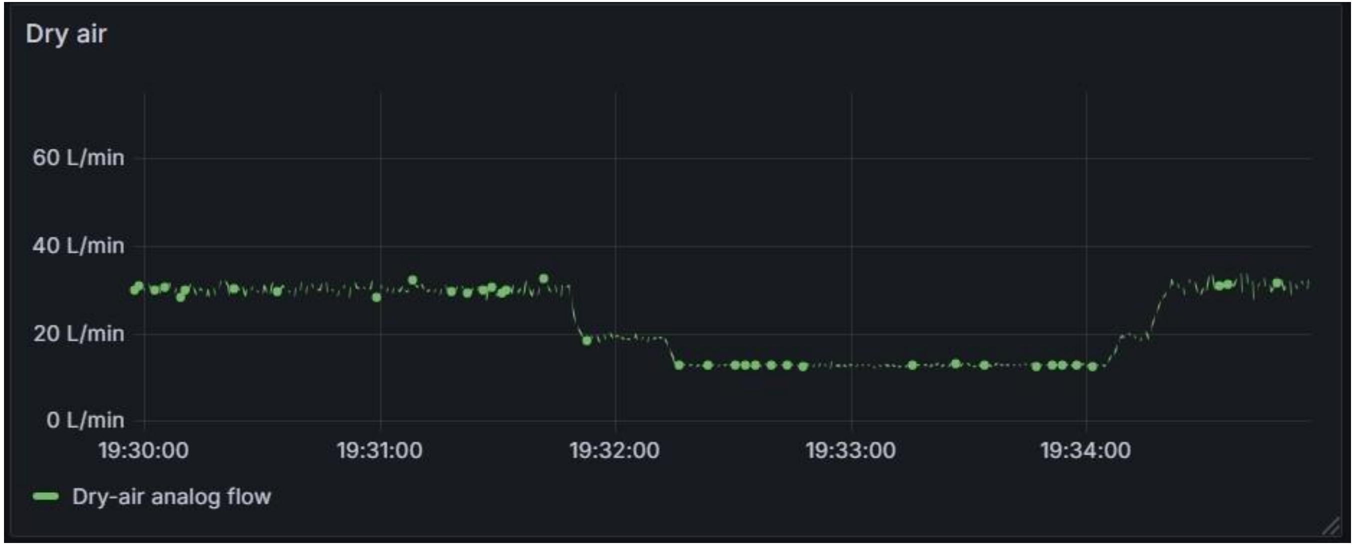
- SW interlock script incomplete:
  - no heartbeat, not easy to monitor
  - different status codes & number of status codes for SoftwareInterlock class & Quantity class (can be confusing to read)
  - minimal debugging output
  - requires manual reset even after fixing your setup (enters infinite loop once interlock is triggered)
- All of the above have been fixed (as of 2/6/25)
- NOTE: lockdown is ON when interlock file is UNLOCKED, so it needs to LOCK the file to lift interlock and allow operation

Software Interlock (Sourcebox) - Test

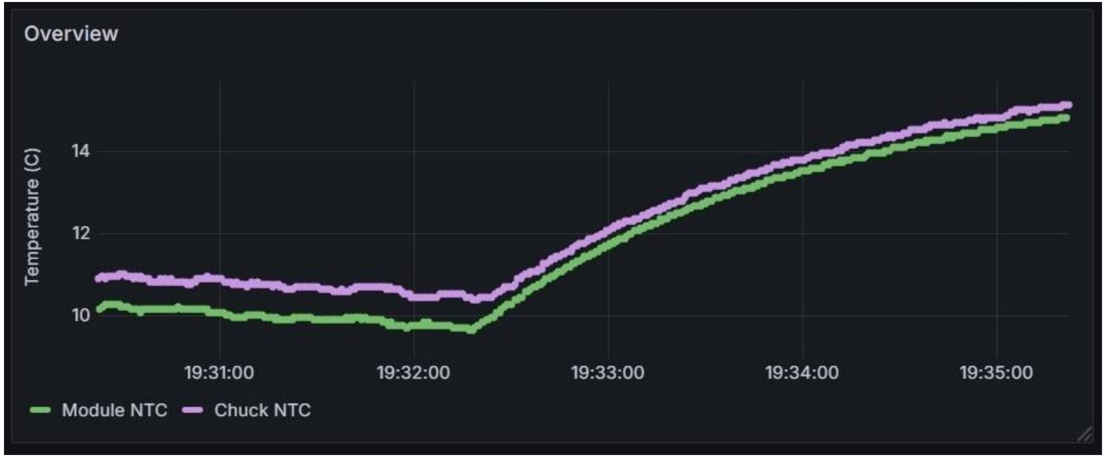
Ferdinand

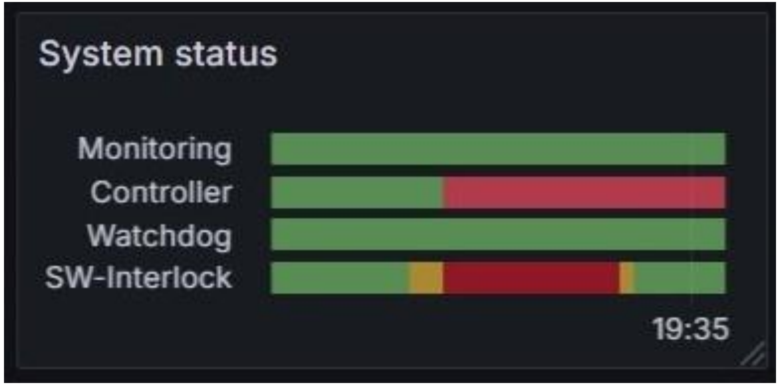
- To see SW interlock working, we want to trigger shutdown & lockdown
- Dry air flow easy to control (turn knob)
- Test - trigger SW interlock while box is operating:
  - operate at 10 C (`python ./scripts/coolman.py -c conf/sourcebox_config.json operate 10`)
  - reduce dry air flow first to ‘lower warning’ (20 Lpm), then to ‘lower cutoff’ (15 Lpm)
  - hopefully see:
    - controller gets shut down
    - email alerts are sent (for warning and cutoff)
    - interlock file being locked
  - then manually recover good air flow and see:
    - SW interlock updates status automatically
    - interlock file lockdown lifted automatically





- Upper left: box dry air flow dipping then increasing (manually done)
- Lower left: module NTC first in lower temp (controller on) then increasing temp (controller off)
- Lower right: SW status visible on Grafana
- Test successful!





Software Interlock - Goals

Ferdinand

- Summary of improvements:
  - added heartbeat to SW interlock process (monitored by controller & visible on Grafana)
  - standardized status codes for swint, quantities, and heartbeat (1 = GOOD, 2 = WARNING, 3 = CRITICAL)
  - SW interlock now queries DB every 2s even in lockdown mode to update status
  - interlock file locking/unlocking is handled automatically (no need to restart interlock script after reaching GOOD/WARNING state)
- Next steps:
  - add HW interlock monitoring to SW interlock
  - have Watchdog start monitoring & SW interlock + have it monitor SW interlock status
  - make it work for the coldbox
    - just need to create new config file (ideally)
  - add HW interlock & box lid monitoring to Grafana
- More details on software and the sourcebox setup: Sourcebox Wiki