Release Report: CEREZA_V1.4.2

Release Date: 2025-12-23 11:19:05+00:00

Total Issues: 2

Release Description

Changes ## Fix latest issues: - #205 - #203 ___ ## Summary of changes * `HTTPHandler` additional that sends logs to local server to aggregate logfiles * `Timeout` handling correctly before each I/O on RS bus to prevent scheduler crash * Experiment run for 19 days without crash ## Metrics [![pipeline status](https://gitlab.com/cherrydev/cubix/badges/main/pipeline.svg?ref=a887dc3ee0b885efd34117d135440a756ee3b30a)](https://gitlab.com/cherrydev/cubix/-/commits/main?ref=a887dc3ee0b885efd34117d135440a756ee3b30a) [![code coverage](https://gitlab.com/cherrydev/cubix/badges/main/coverage.svg?ref=a887dc3ee0b885efd34117d135440a756ee3b30a)](https://gitlab.com/cherrydev/cubix/-/commits/main?ref=a887dc3ee0b885efd34117d135440a756ee3b30a) [![linting](https://gitlab.com/cherrydev/cubix/-/jobs/artifacts/a887dc3ee0b885efd34117d135440a756ee3b30a/raw/public/badges/pylint.svg?job=pylint)](https://gitlab.com/cherrydev/cubix/-/commits/main?ref=a887dc3ee0b885efd34117d135440a756ee3b30a) [![Documentation](https://img.shields.io/badge/docs-Passing-green.svg)](https://cherrydev.gitlab.io/cubix/) ## Changes since last update: [Changes since v1.4](https://gitlab.com/cherrydev/cubix/-/compare/CEREZA_V1.4...CEREZA_V1.4.1?from_project_id=42324438)

Tasks

#177646230: Can we post the log on a port through a socket ? (closed) - Assignee: Unassigned

Description:

Hook up the logfile to textstream and socket at the same time ?

#177196053: Problem after 24H (closed) - Assignee: jean-robin peiteado

Description:

Problem while hitting timeout Not getting to idle then closing and re-opening the inlet... **logs_2025-10-27_10_00_43.log** : error where we keep the inlet opened ``` [INFO] 2025-10-31 06:12:12 mfc.main.Axetris.simplified_loop_/dev/ttyUSB1() Flow values received: [2.0, 1.998, 1.996, 1.996] [INFO] 2025-10-31 06:12:12 mfc.main.Axetris.simplified_loop_/dev/ttyUSB1() Temperature values received: [37.999, 38.255, 38.035, 37.535] [INFO] 2025-10-31 06:12:12 Valves.open: CH[1] VA[12] [INFO] 2025-10-31 06:12:13 Valves.open: CH[2] VA[0] [INFO] 2025-10-31 06:12:13 Valves.open: CH[3] VA[19] [INFO] 2025-10-31 06:12:13 Valves.open: CH[4] VA[13] [ERROR] 2025-10-31 06:12:16 mfc.main.Axetris._close_safely_mfc(): Could not get idle worker Event at port /dev/ttyUSB0, Hit timeout [INFO] 2025-10-31 06:12:16 Valves.close: CH[1] VA[1] [INFO] 2025-10-31 06:12:16 Valves.close: CH[2] VA[25] [INFO] 2025-10-31 06:12:16 Valves.close: CH[3] VA[16] [INFO]

2025-10-31 06:12:16 flow_ch 0: 2000.0flow_ch 1: 1998.0flow_ch 2: 1996.0flow_ch 3: 1996.0Iflow_ch 1: 0.0Iflow_ch 2: 0.0Iflow_ch 3: 0.0 O2:3.82 CO2:3.85 N2:4.03 GM:2.12 P1:-0.01 P2:-0.02 P3:-0.01 P4:-0.00 T:882.70 Tref:37.00 T_well = 882.695157 [INFO] 2025-10-31 06:12:16 Valves.open: CH[1] VA[6] [INFO] 2025-10-31 06:12:16 mfc.main.Axetris.simplified_loop_/dev/ttyUSB1() Flow values received: [2.0, 1.996, 2.0, 1.996] [INFO] 2025-10-31 06:12:16 Valves.open: CH[1] VA[6] [INFO] 2025-10-31 06:12:16 mfc.main.Axetris.simplified_loop_/dev/ttyUSB1() Temperature values received: [37.999, 38.267, 38.035, 37.559] [INFO] 2025-10-31 06:12:16 Valves.open: CH[1] VA[12] [INFO] 2025-10-31 06:12:17 Valves.open: CH[2] VA[0] [INFO] 2025-10-31 06:12:17 Valves.open: CH[1] VA[6] [INFO] 2025-10-31 06:12:17 mfc.main.Axetris.simplified_loop_/dev/ttyUSB0() Flow values received: [3.996, 0.796, 15.2] [INFO] 2025-10-31 06:12:17 mfc.main.Axetris.simplified_loop_/dev/ttyUSB0() Temperature values received: [39.72, 38.218, 39.708] [INFO] 2025-10-31 06:12:17 mfc.main.Axetris.simplified_loop_/dev/ttyUSB0() set_flows: [4.0, 0.8, 15.2] [INFO] 2025-10-31 06:12:17 Valves.open: CH[1] VA[1] [INFO] 2025-10-31 06:12:17 Valves.open: CH[3] VA[19] [INFO] 2025-10-31 06:12:17 flow_ch 0: 2000.0flow_ch 1: 1996.0flow_ch 2: 2000.0flow_ch 3: 1996.0Iflow_ch 1: 3996.0Iflow_ch 2: 796.0Iflow_ch 3: 15200.0 O2:3.83 CO2:3.85 N2:4.03 GM:2.18 P1:0.00 P2:-0.02 P3:-0.01 P4:-0.00 T:882.70 Tref:37.00 T_well = 882.695157 [INFO] 2025-10-31 06:12:17 Valves.open: CH[1] VA[6] [INFO] 2025-10-31 06:12:17 Valves.open: CH[2] VA[25] [INFO] 2025-10-31 06:12:17 Valves.open: CH[4] VA[13] [INFO] 2025-10-31 06:12:17 Valves.open: CH[1] VA[6] [INFO] 2025-10-31 06:12:18 Valves.open: CH[3] VA[16] [INFO] 2025-10-31 06:12:18 Valves.open: CH[1] VA[6] ``` **logs_2025-10-27_09_59_47.log** : where we don't let the inlet opened ``` [INFO] 2025-11-02 22:42:23 mfc.main.Axetris.simplified_loop_/dev/ttyUSB0() managed to set flows normally [INFO] 2025-11-02 22:42:25 mfc.main.Axetris.simplified_loop_/dev/ttyUSB0() Flow values received: [1.996, 1.996, 0.0, 0.0] [INFO] 2025-11-02 22:42:25 mfc.main.Axetris.simplified_loop_/dev/ttyUSB0() Temperature values received: [29.173, 29.185, 27.842, 28.013] [INFO] 2025-11-02 22:42:25 mfc.main.Axetris.simplified_loop_/dev/ttyUSB0() set_flows: [2, 2, 0, 0, 0, 0] [INFO] 2025-11-02 22:42:25 Valves.open: CH[1] VA[12] [INFO] 2025-11-02 22:42:26 Valves.open: CH[2] VA[0] [INFO] 2025-11-02 22:42:26 Valves.close: CH[3] VA[19] [INFO] 2025-11-02 22:42:26 Valves.close: CH[4] VA[13] [INFO] 2025-11-02 22:42:26 mfc.main.Axetris.simplified_loop_/dev/ttyUSB0() managed to set flows normally [ERROR] 2025-11-02 22:42:28 mfc.main.Axetris._close_safely_mfc(): Could not get idle worker Event at port /dev/ttyUSB1, Hit timeout [INFO] 2025-11-02 22:42:28 Valves.close: CH[1] VA[1] [INFO] 2025-11-02 22:42:28 Valves.close: CH[2] VA[25] [INFO] 2025-11-02 22:42:28 Valves.close: CH[3] VA[16] [INFO] 2025-11-02 22:42:28 flow_ch 0: 1996.0flow_ch 1: 1996.0flow_ch 2: 0.0flow_ch 3: 0.0Iflow_ch 1: 0.0Iflow_ch 2: 0.0Iflow_ch 3: 0.0 O2:4.15 CO2:3.90 N2:4.15 GM:2.13 P1:0.01 P2:0.00 P3:-0.00 P4:-0.00 T:882.70 Tref:37.00 T_well = 882.695157 [INFO] 2025-11-02 22:42:28 Valves.open: CH[1] VA[6] [INFO] 2025-11-02 22:42:28 mfc.main.Axetris.simplified_loop_/dev/ttyUSB0() Flow values received: [1.998, 1.996, 0.0, 0.0] [INFO] 2025-11-02 22:42:28 mfc.main.Axetris.simplified_loop_/dev/ttyUSB0() Temperature values received: [29.185, 29.197, 27.842, 28.025] [INFO] 2025-11-02 22:42:28 mfc.main.Axetris.simplified_loop_/dev/ttyUSB0() set_flows: [2, 2, 0, 0, 0, 0] [INFO] 2025-11-02 22:42:28 Valves.open: CH[1] VA[12] [INFO] 2025-11-02 22:42:28 Valves.open: CH[1] VA[6] [INFO] 2025-11-02 22:42:29 Valves.open: CH[2] VA[0] [INFO] 2025-11-02 22:42:29 Valves.open: CH[1] VA[6] [INFO] 2025-11-02 22:42:29 Valves.close: CH[3] VA[19] [INFO] 2025-11-02 22:42:29 Valves.close: CH[4] VA[13] ``` --- ## Timing issue Maximum time spent in the loop can go up to 7s without triggering messages inside MFC, our current timeout for idle is 5s. Logic would advise to increase the timeout and add a strict check before each measure_flow/set_flow/get_temp in order to ensure going to the correct handling without triggering the timeout ### Evaluation of maximum time per operation O.15 s before, between and after the single read, performed a maximum of 15 times. ### Non interruptible IOs When the IOs are impossible to interrupt without damage, we need to check for changes in the security flag before and in-between those operations, as following: ```diff index 3351c0b0..1773c172 100644 --- a/mfc/axetris_aux.py +++ b/mfc/axetris_aux.py @@ -28,7 +28,7 @@ from files.main import CONFIG # Main parameters to change: -IDLE_TIMEOUT = 5.0 +IDLE_TIMEOUT = 10.0 TIMEOUT = 1.5 MFC_TYPE = "CBX001" # line to change: Configuration changes here and file.main.config.json ``` ```diff index f8efe774..349de0da 100644 --- a/mfc/main.py +++ b/mfc/main.py @@ -533,8 +533,14 @@ class

Axetris: self.initialize_event.clear() try: # only handles timeoutError and RecoveryError self.parent.logger.log(2,f"Regular readings {self.port}") + if self.security_closed or self.shutdown_event.is_set(): + continue flows = self.measure_flows() + if self.security_closed or self.shutdown_event.is_set(): + continue temperatures = self.measure_temperatures() + if self.security_closed or self.shutdown_event.is_set(): + continue self.mfc_device.update_data({"flow": flows}) self.mfc_device.update_data({"temperature_mfc": temperatures}) except Exception as re_: @@ -554,6 +560,8 @@ class Axetris: f"mfc.main.Axetris.simplified_loop_{self.port}() Temperature values received: {temperatures}",) try: + if self.security_closed or self.shutdown_event.is_set(): + continue tmp_ = (copy(self.x) if not self.paused ``` Replicate on Burkert ```diff --git a/burkert/main.py b/burkert/main.py index b5d3c960..16e03d65 100644 --- a/burkert/main.py +++ b/burkert/main.py @@ -643,7 +643,11 @@ class Burkert: self.initialize_event.clear() try: # only handles timeoutError and RecoveryError self.parent.logger.log(2,f"Regular readings {self.port}") + if self.security_closed or self.shutdown_event.is_set(): + continue flows = self.measure_flows() + if self.security_closed or self.shutdown_event.is_set(): + continue temperatures = self.measure_temperatures() self.mfc_device.update_data({"flow": flows}) self.mfc_device.update_data({"temperature_mfc": temperatures}) @@ -664,6 +668,8 @@ class Burkert: f"burkert.main.Burkert.simplified_loop_{self.port}() Temperature values received: {temperatures}",) try: + if self.security_closed or self.shutdown_event.is_set(): + continue tmp_ = (copy(self.x) if not self.paused ``` ___ # Fixed for the moment After 5 days, no more issues with fix Remove extensive logging and cleanup before merging on main