Air LiHa (ZaapMotion) Investigation for Tecan EVO 150

Background

The Tecan Freedom EVO 150 can be equipped with two types of Liquid Handling Arms (LiHa):

- **Syringe LiHa** — positive displacement using XP2000/XP6000 syringe dilutors
- **Air LiHa** — air displacement using ZaapMotion BLDC motor controllers

The existing pylabrobot `EVOBackend` was written for syringe-based LiHa. This document describes the investigation into what changes are needed to support Air LiHa with ZaapMotion controllers.

Investigation Method

1. **EVOware firmware logs** — captured EVOware's command log during initialization and liquid handling operations
2. **USB packet capture** — used USBPcap + Wireshark to capture raw USB traffic between EVOware and the TeCU, revealing commands that EVOware's log does not show
3. **DLL string analysis** — extracted strings from `zaapmotiondriver.dll` to understand its configuration model
4. **Iterative testing** — tested individual firmware commands from pylabrobot to isolate the initialization problem

Finding 1: ZaapMotion Boot Mode

Problem

After every power cycle, all 8 ZaapMotion dilutor controllers boot into **bootloader mode** (`XP2-BOOT`, mode `ZMB`). In this state, the Z-axis motors cannot perform homing, so `PIA` (Position Initialization All Axes) always fails with error code 1 on all Z-axes.

Root Cause

The ZaapMotion controllers have firmware in onboard flash but default to bootloader mode on power-up. EVOware's `zaapmotiondriver.dll` sends an `X` (exit boot) command via the transparent pipeline (`T2xX`) to jump each controller to application mode.

Evidence

```
# After power cycle - bootloader mode
> C5,T20RFV0  → XP2-BOOT-V1.00-05/2011, 1.0.0.9506, ZMB
```

```
# After sending T20X - application mode
> C5,T20RFV0  → XP2000-V1.20-02/2015, 1.2.0.10946, ZMA
```

Fix

Send `C5,T2{0-7}X` to each tip before attempting PIA. Wait ~1 second after each for the application firmware to start.

Finding 2: ZaapMotion Motor Configuration

Problem

Even after exiting boot mode, PIA still fails. The Z-axis BLDC motors don't have their PID gains, current limits, encoder settings, or init parameters configured.

Root Cause

EVOware's `zaapmotiondriver.dll` sends ~30 motor configuration commands per tip during its 30-second "Scanning for and configuring ZaapMotion Axes" phase. These commands are sent via the transparent pipeline (`T2x`) and are **not logged in EVOware's firmware command log** — they were only visible in the USB packet capture.

Configuration Sequence (per tip, via transparent pipeline T2x)

```
RFV         - check firmware version (verify app mode)
CFE 255,500 - configure force/current enable
CAD ADCA,0,12.5 / CAD ADCB,1,12.5 - ADC configuration
EDF1        - enable drive function 1
EDF4        - enable drive function 4
CDO 11      - configure drive output
EDF5        - enable drive function 5
SIC 10,5    - set init current (10 amps, 5 ?)
SEA ADD,H,4,STOP,1,0,0 - set encoder/axis config
CMTBLDC,1   - configure motor type = Brushless DC
CETQEP2,256,R - configure encoder type = QEP2, 256 counts, reversed
CECPOS,QEP2 - connect position feedback to QEP2
CECCUR,QEP2 - connect current feedback to QEP2
CEE OFF     - controller encoder enable off
STL80       - set torque limit = 80%
SVL12,8,16 / SVL24,20,28 - set voltage limits
SCL1,900,3.5 - set current limit
SCE HOLD,500 / SCE MOVE,500 - set current for hold/move modes
CIR0        - clear integral reset
PIDHOLD,D,1.2,1,-1,0.003,0,0,OFF   - PID D-axis hold mode
```

```
PIDMOVE,D,0.8,1,-1,0.004,0,0,OFF   - PID D-axis move mode
PIDHOLD,Q,1.2,1,-1,0.003,0,0,OFF   - PID Q-axis hold mode
PIDMOVE,Q,0.8,1,-1,0.004,0,0,OFF   - PID Q-axis move mode
PIDHOLD,POS,0.2,1,-1,0.02,4,0,OFF  - PID position hold mode
PIDMOVE,POS,0.35,1,-1,0.1,3,0,OFF  - PID position move mode
PIDSPDELAY,0 - PID switch delay = 0
SFF 0.045,0.4,0.041 - set force factors
SES 0       - set encoder setting
SPO0        - set position offset = 0
SIA 0.01, 0.28, 0.0 - set init acceleration
WRP         - write all parameters to flash
```

Evidence

USB capture file: `keyser-testing/tecan-2/tecan.pcap` — 11,727 USB packets showing the complete init sequence including all T2x commands.

Fix

Send the above 33 commands to each of the 8 tips (T20-T27) before PIA. Validated in `keyser-testing/zaapmotion_init.py` — PIA succeeds with all 8 Z-axes after configuration.

Finding 3: Safety Module Commands

Context

EVOware sends safety module (O1) commands before PIA:

```
O1,SPN      - power on
O1,SPS3     - set power state = 3 (full power)
```

These ensure the motor drivers have full power. Without them, Z-axis homing may be unreliable. The existing pylabrobot `EVOBackend.setup()` does not send these commands.

Finding 4: Air LiHa Plunger Conversion Factors

Problem

The existing EVOBackend uses `volume * 3` for plunger steps and `speed * 6` for plunger speed — these are specific to syringe-based XP2000/XP6000 dilutors.

Correct Conversion for Air LiHa

Derived from USB captures correlating `CalculateProfile` log entries with actual `PPR`/`MTR` firmware commands:

Volume (µL)	Command	Steps	Steps/µL
2.5	PPR -266	266	106.40
5.0	PPR 532	532	106.40
10.0	PPR 1065	1065	106.50
15.0	PPR 1597	1597	106.47
18.8	MTR -1996	1996	106.17
25.0	PPR 2662 / MTR -2662	2662	106.48
30.0	MTR -3195	3195	106.50
31.3	MTR 3328	3328	106.33
42.3	MTR -4499	4499	106.41
47.9	MTR 5096	5096	106.39
62.9	MTR -6693	6693	106.41

Air LiHa conversion factors: - **106.4 steps/µL** (vs 3 for syringe LiHa — 35x difference) - **213 half-steps/sec per µL/s** for speed (vs 6 for syringe — 35x difference)

Evidence

- EVOware log: `keyser-testing/multidispense pro/EVO_20260327_125527.LOG`
- USB capture: `keyser-testing/simple pro/tecan2.pcap`, `keyser-testing/multidispense pro/tecan3.pcap`

Finding 5: ZaapMotion Per-Operation Commands

Problem

EVOware sends ZaapMotion-specific commands before and after every plunger operation (aspirate, dispense, tip discard). These are not present in the existing pylabrobot code.

Command Pattern

Before each plunger move:

```
T2xSFR133120    - Set force ramp (high value for acceleration)
SEP/SPP         - Set plunger end/stop speed (standard LiHa command)
T2xSFP1         - Enable force mode
```

After each plunger move:

```
T2xSFR3752      - Set force ramp (low value for hold/idle)
T2xSDP1400      - Set dispense parameter (default/idle)
```

This pattern is identical for aspirate, dispense, and tip discard operations.

Additional: Tip Discard

EVOware sends `C5,SDT1,1000,200` (Set DiTi discard parameters) before `AST` (drop tip). The existing pylabrobot code calls `AST` directly without this setup command.

Finding 6: Liquid Class Data

Source

EVOware liquid class XML files captured from `C:\ProgramData\Tecan\EVOware\database\`: - `DefaultLCs.XML` (1.4 MB) — 93 ZaapDiTi liquid class entries - `CustomLCs.XML` (437 KB) - Location: `keyser-testing/multidispense pro/`

Key Data Points per Liquid Class

```
<SubClass tipType="ZaapDiTi" min="10.01" max="50.01">
  <Aspirate>
    <Single speed="50" delay="400" />
    <LAG volume="10" speed="70" />        <!-- leading air gap -->
    <TAG volume="1" speed="20" />         <!-- trailing air gap -->
    <LLD detect="True" position="3" offset="0" />
    <Retract speed="5" position="4" offset="-5" />
  </Aspirate>
  <Dispense>
    <Single speed="600" breakoff="400" />
    <Retract speed="50" position="1" offset="0" />
  </Dispense>
  <Calibration>
    <Single offset="0.36" factor="1.04" />
  </Calibration>
</SubClass>
```

Existing Support

`TipType.AIRDITI = "ZaapDiTi"` already exists in pylabrobot's `TipType` enum (`pylabrobot/resources/tecan/tip_creators.py`), but no liquid class mapping entries use it.

Summary of Required Changes

1. Initialization (EVOBackend.setup)

- Exit ZaapMotion boot mode: `T2{0-7}X`
- Send 33 motor configuration commands per tip
- Send safety module commands: `O1,SPN` and `O1,SPS3`
- Then proceed with existing PIA sequence

2. Plunger Conversions

- Replace hardcoded `* 3` (steps) with `* 106.4` for Air LiHa
- Replace hardcoded `* 6` (speed) with `* 213` for Air LiHa
- Affects: `_aspirate_airgap()`, `_aspirate_action()`, `_dispense_action()`, `pick_up_tips()`

3. Per-Operation ZaapMotion Commands

- Add `SFR133120` + `SFP1` before each plunger operation
- Add `SFR3752` + `SDP1400` after each plunger operation
- Add `SDT1,1000,200` before tip discard

4. Liquid Classes

- Parse ZaapDiTi entries from DefaultLCs.XML into pylabrobot's `mapping` dict
- Map to existing `TecanLiquidClass` fields

Hardware Details

- **TeCU**: TECU-V1.40-12/2007
- **LiHa CU**: LIHACU-V1.80-02/2016, 8 air channels
- **ZaapMotion boot**: XP2-BOOT-V1.00-05/2011
- **ZaapMotion app**: XP2000-V1.20-02/2015
- **Safety module**: SAFY-V1.30-04/2008
- **RoMa**: ROMACU-V2.21-09/2007
- **Tips**: DiTi 50µL SBS LiHa (ZaapDiTi tip type)

Files

File	Description
`keyser-testing/zaapmotion_init.py`	Working init script (boot exit + motor config + PIA)
`keyser-testing/tecan-2/tecan.pcap`	USB capture of EVOware init sequence
`keyser-testing/simple pro/tecan2.pcap`	USB capture of aspirate/dispense/discard
`keyser-testing/multidispense pro/tecan3.pcap`	USB capture with multiple volumes
`keyser-testing/multidispense pro/DefaultLCs.XML`	EVOware liquid class definitions
`keyser-testing/multidispense pro/EVO_20260327_125527.LOG`	EVOware log with volume calculations
`keyser-testing/Tecan Manuals/text/`	Firmware command set manuals (LiHa, RoMa, MCA, PnP)