Turnpack, LLC
22425 S Scotland Ct, Suite 104
Queen Creek, AZ 85142
Tel: +1 (480) 331-1878

System-Level Design

1.1 Scope

This spec covers:

- UF-4000A reflow power supply integration
- ClickPLC integration (stage heater, secondary thermocouple, Futek/IAA100 load cell, machine I/O)
- Modbus/TCP data model between HMI (PySide6 app) and PLC
- Basic cycle sequence and responsibilities

1.2 Responsibilities Split

PC / PySide6 Application

- HMI / UI and recipe selection
- UF-4000A serial protocol (RS-232)
- High-level cycle control (start, abort, results logging)
- Polling PLC via Modbus/TCP and combining:
    - UF-4000 bond profile summary
    - Stage + secondary temp + force
    - Interlock status / errors

ClickPLC

- Stage heater closed-loop control (replaces Omron E5CN)
- Secondary thermocouple measurement
- Futek/IAA100 force measurement (analog → engineering units)
- Machine I/O and safety:
    - E-stop, door, head up/down, etc.
    - UF-4000 digital handshake (INITIATE, Ready, Alarm, End-of-Reflow, etc.)
- Provide summary process data to PC via Modbus/TCP

UF-4000A

- Runs the actual reflow profile on the thermode
- Provides:
    - Digital handshake signals to PLC

- Serial “RR” bond report to PC at the end of each cycle

---

2. UF-4000A Integration Spec (Software Developer)

2.1 Physical / Electrical

- Interface: RS-232 (UF-4000A COM port)
- Connection: PC via USB-RS232 adapter
- Serial settings:
  - Baud: per UF manual (default is typically **9600 bps**, 8 data bits, no parity, 1 stop bit); make these configurable in app settings.
- UF address: typically `01` (configurable; read from UF setup and mirror in app config).

2.2 Protocol Overview

- UF uses framed ASCII messages:
  - `<SOH><ADDR><CMD><COUNT><DATA><CHK><EOT>`
- All messages are 7-bit ASCII with checksum.
- PC must support:
  - Sending commands to UF (optional but strongly recommended)
  - Receiving both **unsolicited** and **solicited** responses

At minimum, the HMI must handle:

1. **Automatic end-of-reflow report**
   - UF sends `REPORT "RR"` automatically at the end of each reflow, containing:
     - Profile number used
     - Thermode temp metrics (start, peak, final, average)
     - Auxiliary temp metrics
     - Cool times
     - Status / result code (OK/NG, alarms)
   - HMI must:
     - Listen continuously on the serial port
     - Detect and parse RR messages
     - Associate each RR report with a “cycle” (see §2.4)
2. **Query last result (optional)**
   - Command to request the last RR-like data on demand (e.g., after reconnection).
   - Use when:
     - PC missed the automatic RR because of transient errors
     - PC is restarted while UF is running
3. **Profile control (optional)**

    - Commands to:
        - Read current profile
        - Select profile
    - Use if the HMI is expected to control profile selection instead of operator doing it on the UF front panel.

2.3 Software Responsibilities

2.3.1 Serial Port Lifecycle

- On application startup:
    - Load serial config (port name, baud, parity, stop bits, UF address).
    - Open port, start a **background reader thread** that:
        - Reads bytes
        - Frames messages on `<SOH>...<EOT>`
        - Validates checksum
        - Dispatches them to a UF protocol handler
- On failure (port not present, checksum errors, etc.):
    - Mark UF as “Not Connected” or “Faulted”
    - Expose this status to the UI and to logging
    - Do not block entire UI – failures are non-fatal but must be visible.

2.3.2 RR Message Handling

For each `RR` (reflow result) message:

- Parse into a **BondResult** structure containing at least:
    - `uf_cycle_id` (if available)
    - `profile_number`
    - `thermode_start_temp`, `thermode_peak_temp`, `thermode_final_temp`, `thermode_avg_temp`
    - `aux_start_temp`, `aux_peak_temp`, `aux_final_temp`, `aux_avg_temp`
    - `cool_time1`, `cool_time2`
    - `uf_status_code` / pass–fail flag
    - Raw string message for debug (store entire frame in logs)
- Timestamp the reception (`pc_timestamp_uf_result`).
- For each cycle:
    - Combine this BondResult with PLC data (see §3) and UI context (recipe, operator, lot, serial #).
    - Persist it in datastore/log.

2.3.3 Cycle Correlation

Because UF and PLC are independent:

- The HMI defines a **Cycle ID**:
    - Increment a local `cycle_counter` each time the operator starts a new bond from the UI.

- When:
  - UI sends Start command (via PLC coil) → record `cycle_id` and `start_time`.
  - PLC sets “End-of-Reflow” flag → UI logs `end_time_plc`.
  - UF RR report arrives → UI associates RR with **most recent in-progress cycle**.
- If timing mismatch or multiple RR with no PC-initiated cycle:
  - Mark as “unassigned UF result” and log for diagnostics.

2.3.4 Error Handling Requirements

- Loss of serial connection:
  - UI should continue to operate but:
    - Block new cycle starts, or
    - Allow starts but flag every result as missing UF data.
  - Show clear status: “UF-4000 COMM FAULT”.
- UF status codes:
  - Map non-OK codes to “bond failed” and show reason if available.
  - Store codes raw for traceability.

---

3. PLC / Modbus Integration Spec

This section is for **both** SW dev and PLC engineer. It defines the Modbus/TCP model and expected PLC behavior.

3.1 Modbus/TCP General

- PLC acts as a **Modbus/TCP server**.
- PC (HMI) is the **Modbus client**.
- Standard port: 502
- Polling:
  - HMI polls at **50–100 ms** cycle time for fast status/measurements.
  - Writes (commands) on demand.

3.2 PLC Responsibilities Overview

- Stage heater closed-loop control (PID)
- Secondary TC measurement
- Force measurement from Futek/IAA100 (0–10 V or 4–20 mA)
- I/O for:
  - Head up/down
  - E-stop
  - Door interlock
  - UF Ready / Alarm / End-of-Reflow (digital)
  - UF INITIATE command
- Internal cycle management:

    - Local cycle counter
    - Peak force per cycle
    - Stage/secondary temp at start/end of cycle
- Expose above via **fixed register map** as defined below.

3.3 Proposed PLC I/O and Tag Allocation

Names below are logical; PLC engineer can adapt to Click naming, but Modbus mapping must match.

3.3.1 Discrete Inputs (read-only to HMI)

Modbus DI	PLC Tag	Description
10001	DI_ESTOP_OK	1 = E-stop OK, 0 = E-stop pressed
10002	DI_DOOR_CLOSED	1 = Door closed
10003	DI_HEAD_UP	1 = Head up
10004	DI_HEAD_DOWN	1 = Head down
10005	DI_UF_SYSTEM_READY	1 = UF Ready relay ON
10006	DI_UF_ALARM	1 = UF Alarm relay ON
10007	DI_UF_END_OF_REFLOW	1 = End-of-Reflow relay (latched by PLC)
10008	DI_SPARE1	Reserved

3.3.2 Coils (writeable by HMI; some echoed to outputs)

Modbus Coil	PLC Tag	Description
00001	COIL_CMD_START_CYCLE	HMI requests start of bond cycle
00002	COIL_CMD_ABORT_CYCLE	HMI requests abort
00003	COIL_CMD_RESET_ALARMS	Reset PLC and UF alarms (PLC also pulses UF reset)
00004	COIL_CMD_HEAD_DOWN	Command head down (if controlled via PLC)
00005	COIL_CMD_HEAD_UP	Command head up
00006	COIL_ENABLE_HEATER	Enable output stage heater control

00007	COIL_SPARE1	Reserved

PLC behavior: these coils may be latched or treated as pulses; see §3.5.

3.3.3 Input Registers (Analog PVs, read-only)

Scaling convention:

- Temperatures: 0.1 °C (value = actual × 10)
- Force: 0.1 N (or 0.01 kgf – pick one and document)

Modbus IR	PLC Tag	Units	Description
30001	IR_STAGE_TEMP_PV	0.1 °C	Current stage temperature
30002	IR_SEC_TEMP_PV	0.1 °C	Current secondary (top-of-part) temperature
30003	IR_FORCE_PV	0.1 N	Current bonding force
30004	IR_STAGE_TEMP_AT_START	0.1 °C	Stage temp at cycle start
30005	IR_SEC_TEMP_AT_START	0.1 °C	Secondary temp at cycle start
30006	IR_FORCE_PEAK_LAST	0.1 N	Peak force during last completed cycle
30007	IR_STAGE_TEMP_AT_END	0.1 °C	Stage temp at end-of-reflow
30008	IR_SEC_TEMP_AT_END	0.1 °C	Secondary temp at end-of-reflow
30009	IR_SPARE1		Reserved

3.3.4 Holding Registers (Setpoints, commands, statuses)

Modbus HR	PLC Tag	Units	R/W	Description
40001	HR_STAGE_TEMP_SP	0.1 °C	R/W	Active stage temp setpoint
40002	HR_STAGE_TEMP_SP_CMD	0.1 °C	R/W	HMI-requested setpoint (PLC can copy after check)

40003	HR_SEC_TEMP_MAX_LIMIT	0.1 °C	R/W	Max allowed secondary temp during bond
40004	HR_FORCE_MIN_LIMIT	0.1 N	R/W	Minimum acceptable peak force
40005	HR_FORCE_MAX_LIMIT	0.1 N	R/W	Maximum acceptable peak force
40006	HR_CYCLE_STATUS	enum	R	Cycle state (see below)
40007	HR_CYCLE_RESULT_CODE	bitfield	R	PLC-level result bits
40008	HR_CYCLE_COUNTER	count	R	Incremented each completed cycle
40009	HR_ALARM_CODE	code	R	PLC alarm code (0 = OK, >0 = fault)
40010	HR_INTERLOCK_STATUS	bitfield	R	Safety/interlock bits
40011–20	HR_RESERVED			Reserved for future use

HR_CYCLE_STATUS (40006) – suggested states:

- 0 = IDLE
- 1 = READY (all interlocks OK)
- 2 = RUNNING (bond in progress)
- 3 = COMPLETE_OK (cycle done, within PLC limits)
- 4 = COMPLETE_NG (cycle done, PLC limit violation)
- 5 = ABORTED
- 6 = FAULT (hardware/safety fault)

HR_CYCLE_RESULT_CODE (40007) – bitfield example:

- Bit 0: Force too low
- Bit 1: Force too high
- Bit 2: Secondary temp too high
- Bit 3: Temp not at setpoint at start
- Bit 4: UF alarm occurred
- Bit 5: E-stop during cycle
- etc.

HR_INTERLOCK_STATUS (40010) – bitfield example:

- Bit 0: E-stop OK
- Bit 1: Door closed
- Bit 2: Head up
- Bit 3: UF Ready
- Bit 4: Stage heater enabled
- Bit 5: Secondary temp sensor OK

- etc.

3.4 PLC Cycle Logic (Informal)

High-level PLC sequence per cycle:

1. **Idle / Ready**
    - HR_CYCLE_STATUS = IDLE or READY
    - Evaluate interlocks:
        - E-stop, door, UF Ready, no alarms, etc.
    - When all OK, set HR_CYCLE_STATUS = READY.
2. **Start Command**
    - HMI sets `COIL_CMD_START_CYCLE = 1`.
    - PLC:
        - Verifies HR_CYCLE_STATUS == READY, interlocks OK.
        - Latches an internal `PLC_CYCLE_ACTIVE`.
        - Clears peak force register, start/end temps, alarm/result codes.
        - Captures `IR_STAGE_TEMP_PV` / `IR_SEC_TEMP_PV` into “AT_START” registers.
        - Sets HR_CYCLE_STATUS = RUNNING.
        - Turns on UF INITIATE output, starts motion/head-down as needed.
    - PLC resets `COIL_CMD_START_CYCLE` back to 0 after acknowledging.
3. **During Cycle**
    - Track `IR_FORCE_PV` and maintain `IR_FORCE_PEAK_LAST` as max during active cycle.
    - Optionally monitor secondary temp and abort if above HR_SEC_TEMP_MAX_LIMIT.
    - Any hardware fault (E-stop, door open, UF alarm, etc.) → abort logic:
        - Cut outputs, set HR_CYCLE_STATUS = FAULT or ABORTED.
        - Set HR_ALARM_CODE and HR_CYCLE_RESULT_CODE accordingly.
4. **End-of-Reflow**
    - When UF End-of-Reflow DI activates:
        - Capture “AT_END” temperatures to IR_STAGE_TEMP_AT_END / IR_SEC_TEMP_AT_END.
        - Freeze `IR_FORCE_PEAK_LAST`.
        - Increment HR_CYCLE_COUNTER.
        - Evaluate limits (force, temp, etc.) and set HR_CYCLE_RESULT_CODE.
            - Passed limits → HR_CYCLE_STATUS = COMPLETE_OK.
            - Violated → HR_CYCLE_STATUS = COMPLETE_NG.
        - Latch `DI_UF_END_OF_REFLOW` internally and provide as DI for HMI until next cycle or reset.
5. **Reset / Next Cycle**
    - HMI reads all values once HR_CYCLE_STATUS is COMPLETE_XXX.
    - HMI may call:
        - `COIL_CMD_RESET_ALARMS = 1` to clear latched DI / alarms and reset status to READY (if interlocks OK).

    - PLC resets `COIL_CMD_RESET_ALARMS` to 0 after handling.
6. **Abort Command**
    - If HMI sets `COIL_CMD_ABORT_CYCLE = 1` during RUNNING:
        - PLC stops outputs, sets HR_CYCLE_STATUS = ABORTED.
        - Clips motion/head and ensures safe state.
    - Reset coil back to 0 after handling.

---

4. HMI–PLC–UF Interaction (For Both)

4.1 Nominal Cycle Timeline

1. Operator selects product in HMI.
2. HMI:
    - Sets HR_STAGE_TEMP_SP_CMD (if profile-specific).
    - Optionally writes HR_FORCE limits, SEC temp limit.
    - Polls until HR_CYCLE_STATUS = READY and INTERLOCK bits show OK.
3. HMI:
    - Sets `COIL_CMD_START_CYCLE` = 1.
    - Locally assigns a new `cycle_id`.
4. PLC:
    - Accepts command, transitions to RUNNING, starts motion and asserts UF INITIATE.
5. UF:
    - Runs reflow profile.
    - At completion:
        - Activates End-of-Reflow relay → PLC sees DI_UF_END_OF_REFLOW.
        - Sends RR profile report over RS-232 → HMI parses BondResult.
6. PLC:
    - Captures end temps and peak force, updates status/result, increments counter.
7. HMI:
    - Detects HR_CYCLE_STATUS COMPLETE_xx.
    - Reads:
        - PLC analogs (start/end temps, peak force).
        - Status/result bits.
    - Associates these with the RR BondResult and recipe/operator.
    - Logs + displays result.

---

5. Deliverables / Acceptance

5.1 PLC Deliverables

- Ladder program implementing:
    - Stage heater PID with Modbus setpoint and enable.
    - Secondary TC and force scaling with documented scaling factors.
    - Interlock and cycle logic as above.
    - UF digital handshake.
- Verified Modbus/TCP server with **exact** register map:
    - List addressing for the actual Click memory (e.g., DS1/CT/C bits) mapped to above Modbus addresses.
- Short document or comments:
    - Scaling (e.g., “IR_FORCE_PV = 10 × N”).
    - Meaning of each HR/IR/coil/DI.

5.2 Software Deliverables

- Working UF-4000 serial integration:
    - Robust framing, checksum handling, and RR message parsing.
    - Configurable port / address / baud.
- Modbus client implementation:
    - Periodic polling of defined registers and coils.
    - Read–modify–write operations for commands and setpoints.
- HMI wiring:
    - Cycle start, abort, reset wired to PLC coils.
    - Status/alarms from PLC/UF visible in UI.
    - Per-cycle record combining:
        - UF RR data
        - PLC temps and force
        - Recipe / operator / timestamps.
- Minimal config file documenting:
    - Modbus IP/port
    - UF serial parameters
    - Scaling factors (temp, force)
    - mapping of PLC result codes ↔ UI messages