5. Live Operation

Hardware manual

www.pinetek-networks.com

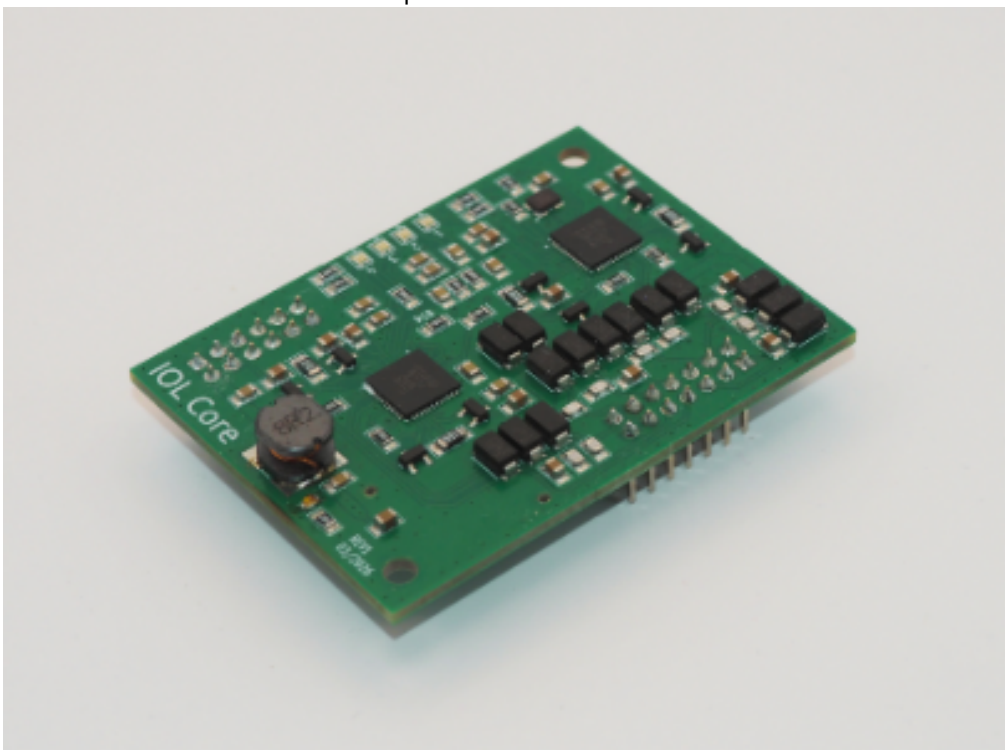


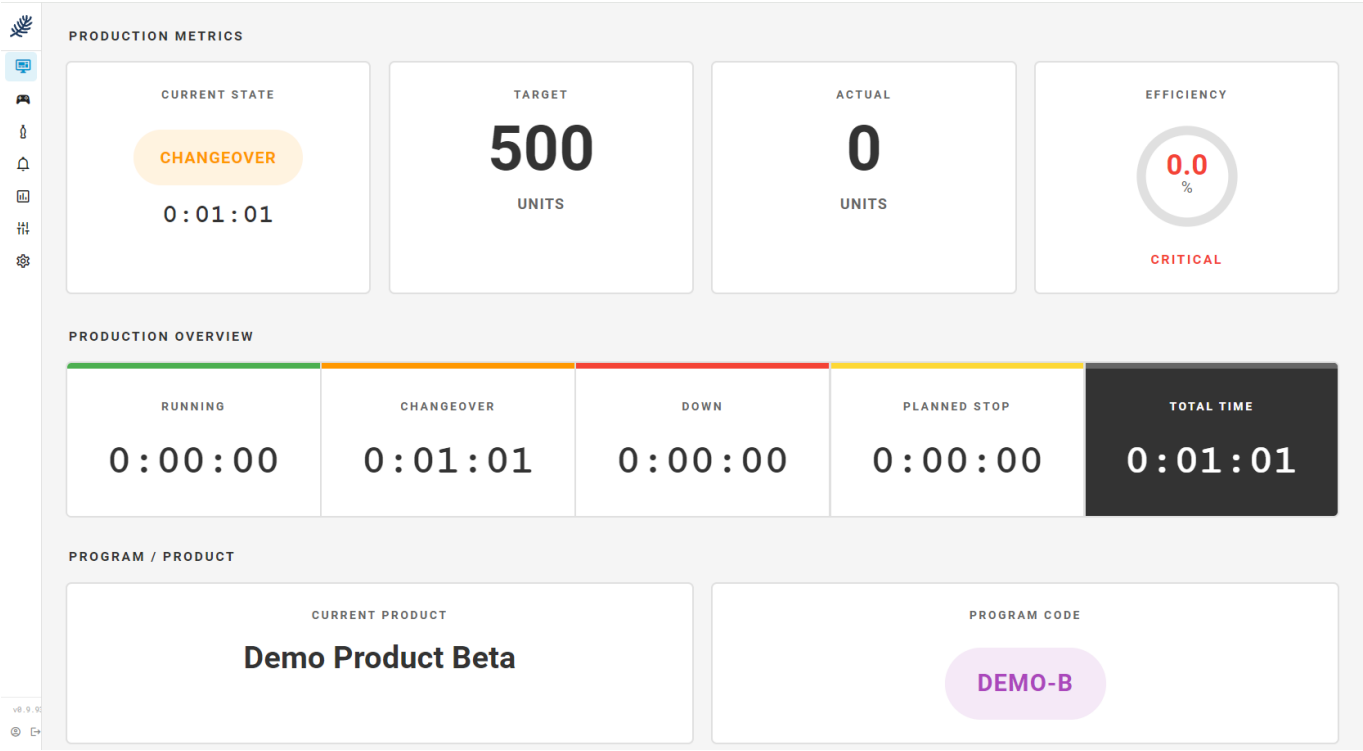
Table of Contents

5. Live Operation 5
5.1 Factory Scoreboard 5
5.2 Production Timeline 5
5.2.1 Starting a run (changeover) 6
5.2.2 During the run 7
5.2.3 Ending a run 7
5.2.4 Run-end summary and bad parts 7
5.3 Live Bottleneck Analysis 8
5.4 Recording downtime reasons 8

5. Live Operation

This chapter describes the day-to-day views used at the line. Required role: **Operator** (Scoreboard: any signed-in user).

5.1 Factory Scoreboard

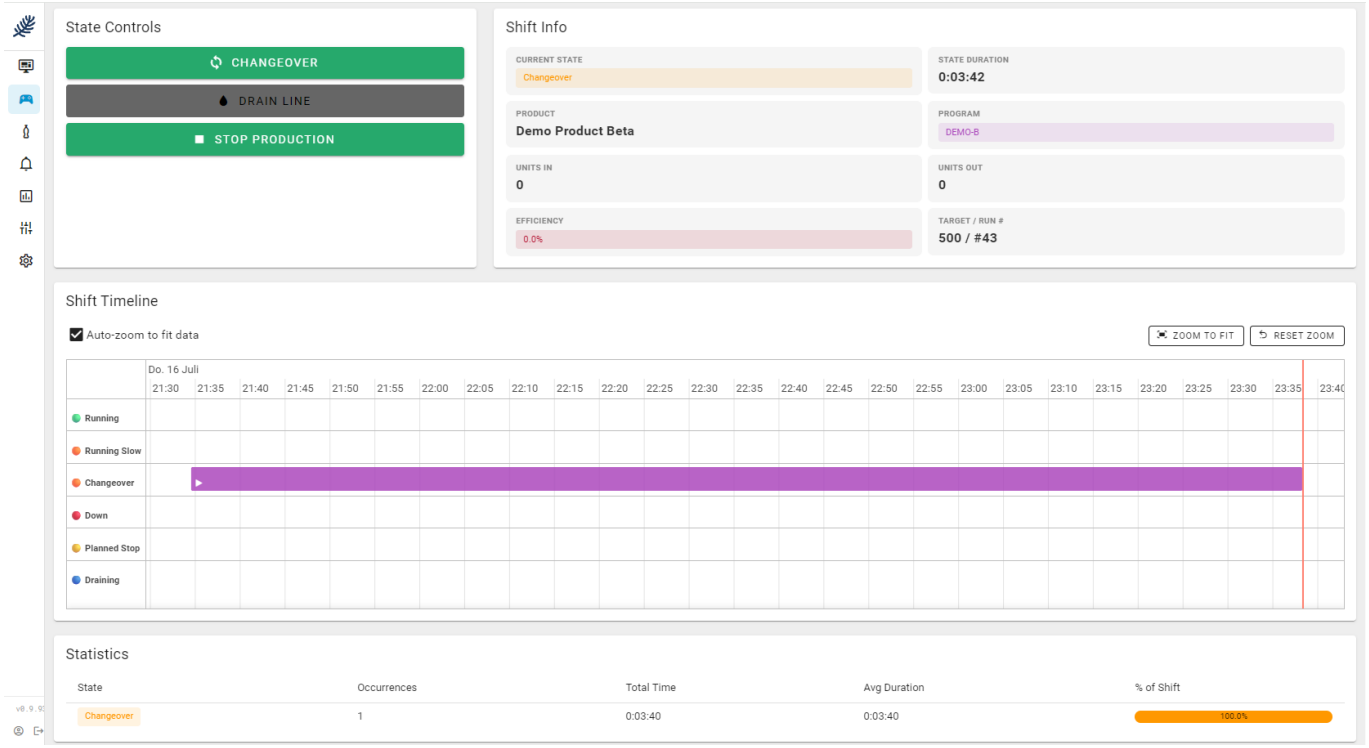


The Scoreboard is a large-format live display intended for wall monitors. It shows:

- **Current State** with state color and duration,
- **Current Product** and **Program Code**,
- Production Metrics: **Target, Actual, Efficiency**,
- Production Overview of the current shift: time in **Running, Changeover, Down, Planned Stop**, and **Total Time**.

The Scoreboard requires no interaction and refreshes automatically. A Basic account (no password) can be used to leave it permanently open on a wall display.

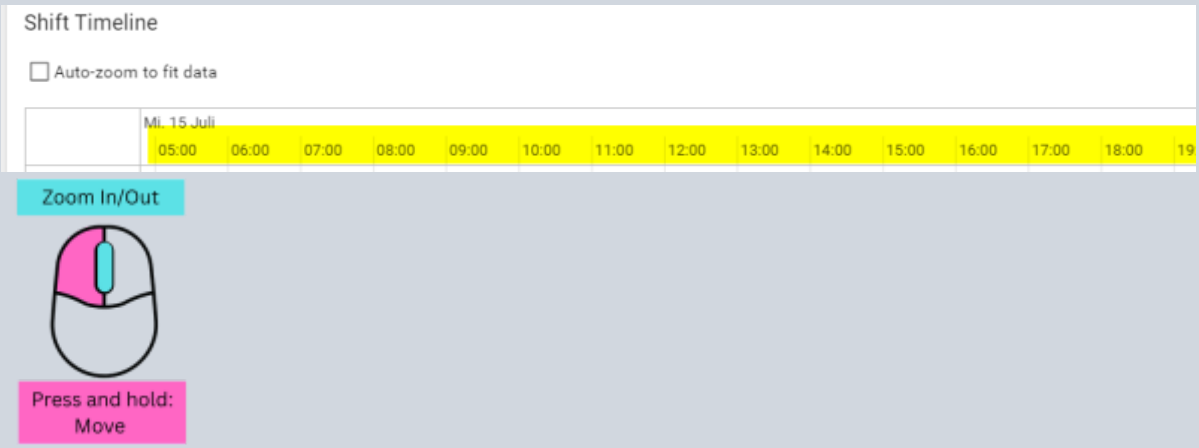
5.2 Production Timeline



The Production Timeline is the operator's main working view. It combines:

- **State Controls**: buttons to drive the production run (below),
- **Shift Info**: current state, state duration, program/product, Units In, Units Out, Efficiency, Target/Run #,
- **Shift Timeline**: a horizontal timeline of all state segments; use *Zoom to Fit* / *Reset Zoom* or enable *Auto-zoom to fit data*,
- **Statistics**: per state: occurrences, total time, average duration, % of shift,
- **Stage table**: per process stage: stage #, name, total units, throughput.

To zoom in and out with the mouse wheel, you need to put the mouse pointer over the timeline (yellow marked):



5.2.1 Starting a run (changeover)

1. Press **Start Changeover**. The line state changes to **Changeover** (purple).
2. Select the **Product** (mandatory): the program is determined by the product.
3. Enter the **Target Quantity** (leave blank for an unlimited run).
4. Confirm. The production run is created.

As soon as the line starts counting parts [PBX], the state changes automatically to **Running**; no further operator action is required.

5.2.2 During the run

The state is maintained automatically from the counter signals (chapter 2.3.2):

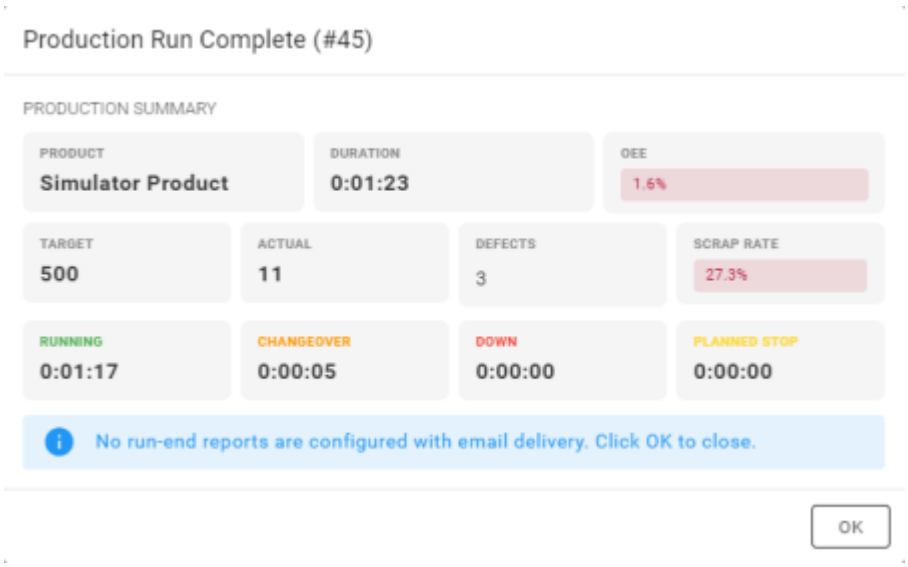
- If a stage counter stalls beyond its timeout, the state becomes **Down** and a downtime reason is requested (section 5.4).
- If throughput drops without a full stop (single stations stall), the state becomes **Running Slow**.
- For breaks or planned interventions, use **Planned Stop**; press **Resume** to continue. Planned stops do not reduce Availability.

5.2.3 Ending a run

There are two ways to end a run:

- **Drain Line**: the input stage stops while the rest of the line runs empty (state **Draining**, blue). When the output counter times out, the run ends automatically.
- **Stop Production**: ends the run immediately (confirmation required).

5.2.4 Run-end summary and bad parts



When the run ends, the **Production Run Complete** dialog shows the run summary: Target, Actual, Defects and Scrap Rate.

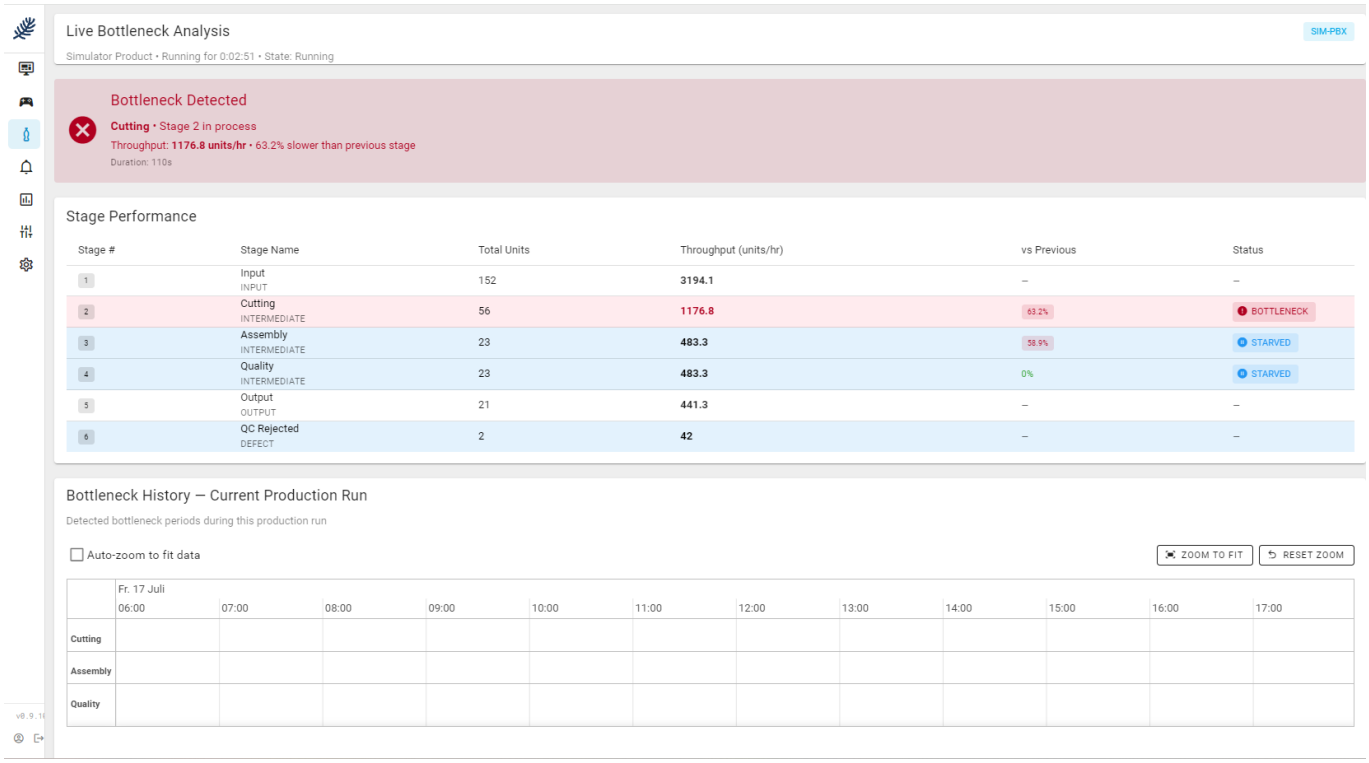
The defect parts are calculated as *input parts - output parts* after the line was drained or production was stopped.

Actions on run-end:

- Correct the defect quantity here if the automatic defect count does not match reality (e.g. manually sorted-out parts). The OEE Quality factor is recalculated with the corrected value.
- If run-end reports with e-mail delivery are configured (chapter 8), the dialog lists the reports and recipients. Choose **Send Email** to dispatch them, or **OK, do not send email** to skip.

The finished run is then available in all analysis views (chapter 6).

5.3 Live Bottleneck Analysis



The Bottleneck view compares the live throughput of all intermediate process stages of the current run and identifies the stage that limits the line.

How detection works: a stage is flagged as a bottleneck if its throughput is more than a configured percentage slower than the fastest stage, and this condition persists longer than a configured minimum duration. Both thresholds are set per program (chapter 7.1). Analysis begins after a warm-up of at least 60 seconds of running time.

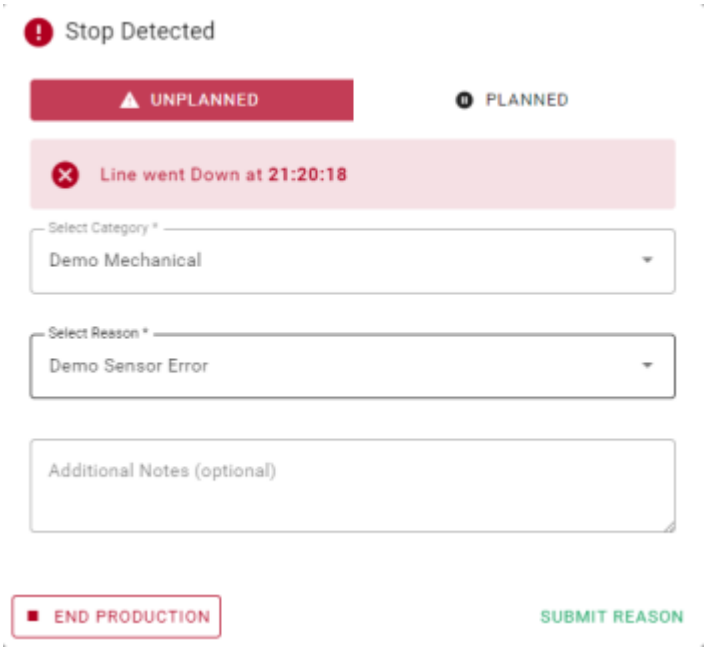
Each stage is shown with total units, throughput (units/h), deviation vs. the previous stage, and a status:

Status	Meaning
NORMAL	Stage operating within threshold
OBSERVING	Deviation detected; a countdown shows when it becomes a confirmed bottleneck if it persists
BOTTLENECK	Deviation persisted beyond the minimum duration; confirmed bottleneck
STARVED	Downstream of the bottleneck, waiting for upstream material

When a bottleneck is detected, all downstream stages are marked *STARVED*, since they cannot run faster than the bottleneck allows. Only the earliest slow stage is the true bottleneck. Input and output stages are shown for reference but are not analysed.

The lower part of the view lists the bottleneck periods detected during the current run. Historical analysis across runs is available under *Analysis → Bottleneck History* (chapter 6.2).

5.4 Recording downtime reasons



Every **Down** period must be justified. When a stop is detected, the timeline view shows a **Downtime Occurred** dialog (“Line went Down at …”):

1. Choose whether the stop was **Planned** or **Unplanned**.
2. Select the **Category** and then the **Reason** (both mandatory; the catalogue is configured under *Line Config → Downtime*, chapter 7.7).
3. Optionally add **Notes**.
4. Press **Submit Reason**.

If a stop is not classified immediately, it remains **pending** and can be assigned or corrected later in *Analysis → OEE History* (chapter 6.1). Complete reason data is what makes the Downtime Pareto (chapter 6.4) meaningful.