Laser Marking Systems Case Study

Embedded operator HMI for industrial laser marking systems, with a purpose-built protocol driver for direct laser control.
Industry: Machinery and OEM (Laser Marking)

Quick Facts

Attribute	Value
Customer	SIC Marking (OEM Partner)
Product	FrameworX 10.1.5
Markets	Canada, United States, and France
Focus	Laser marking and part traceability
Laser Controller	SCAPS USC card, driven through SAMLight
Integration	Custom protocol driver over TCP

The Challenge

Challenge: Replace the laser marking operator interface with a modern HMI that controls the laser directly, without device logic scattered through the application.

Specific pain points:

- Laser configuration was reachable only through the marking software's own interface, which was never designed for a production operator
- Marking content, barcode symbology, counters and date fields had to be changed per part without opening an engineering tool
- Marking parameters such as power, speed, frequency and delays are held per pen and needed operator-level exposure
- An earlier prototype drove the laser from application scripts, which spread device behaviour across the solution and made it difficult to support

Impact: Longer operator training, slower changeovers between parts, and more marking errors, all caused by an interface that was not built for production use.

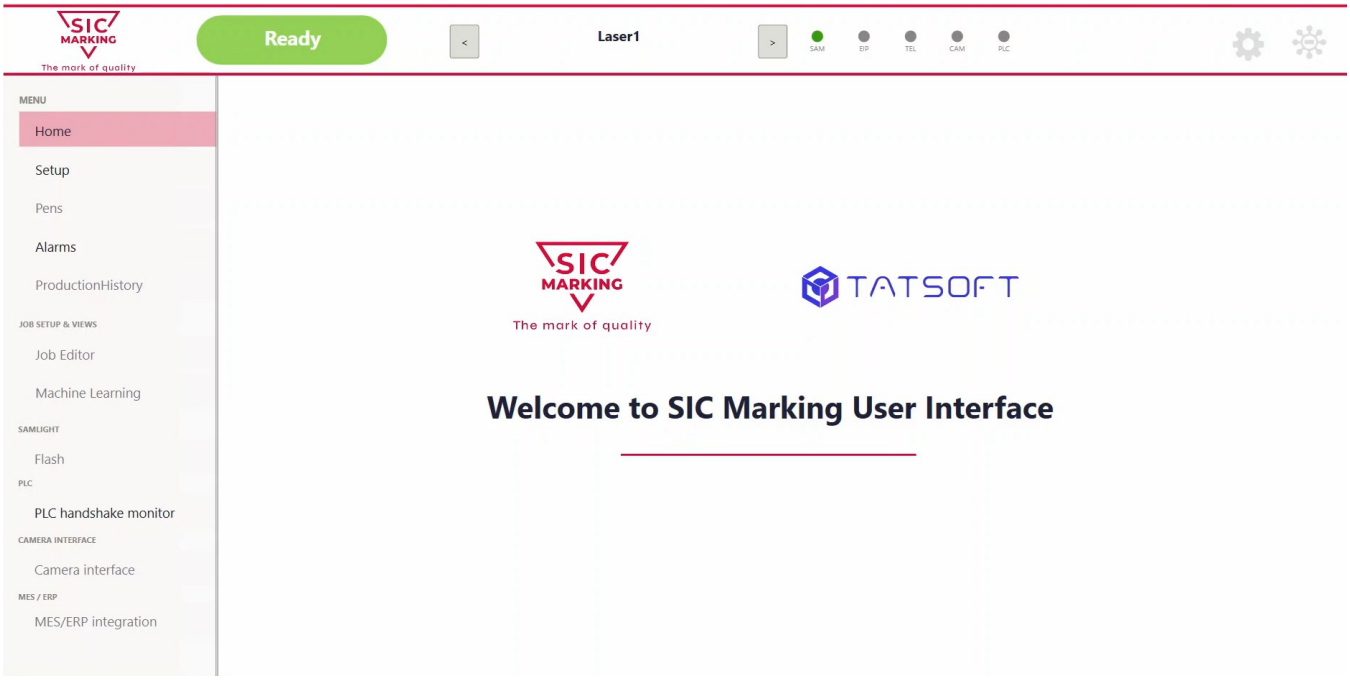
The Solution

Architecture

Component	Implementation
Base Platform	FrameworX 10.1.5
Laser Driver	Custom protocol driver communicating over TCP
Operator HMI	Three purpose-built screens: Setup, Job Editor, and Pens
Pen Library	Dataset of material and parameter presets
Job Preview	Live render of the marking layout inside the display

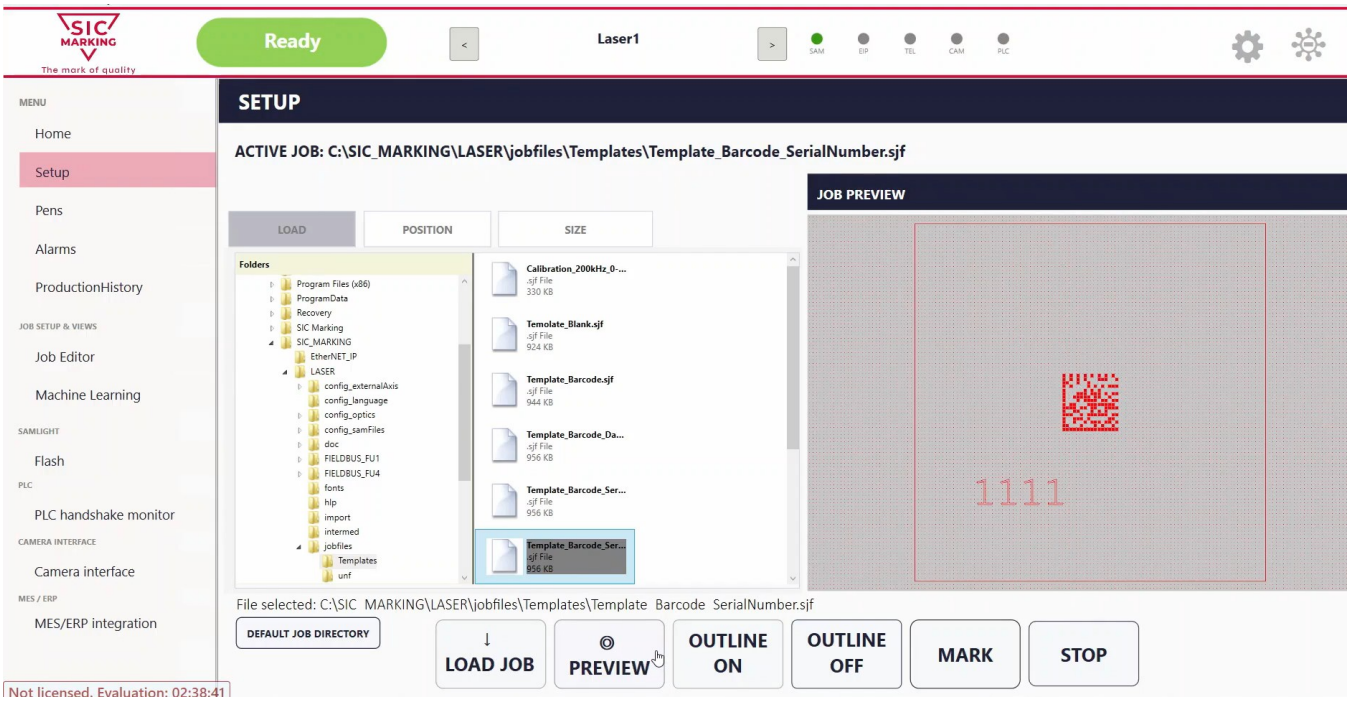
Main Screen

Initial display.



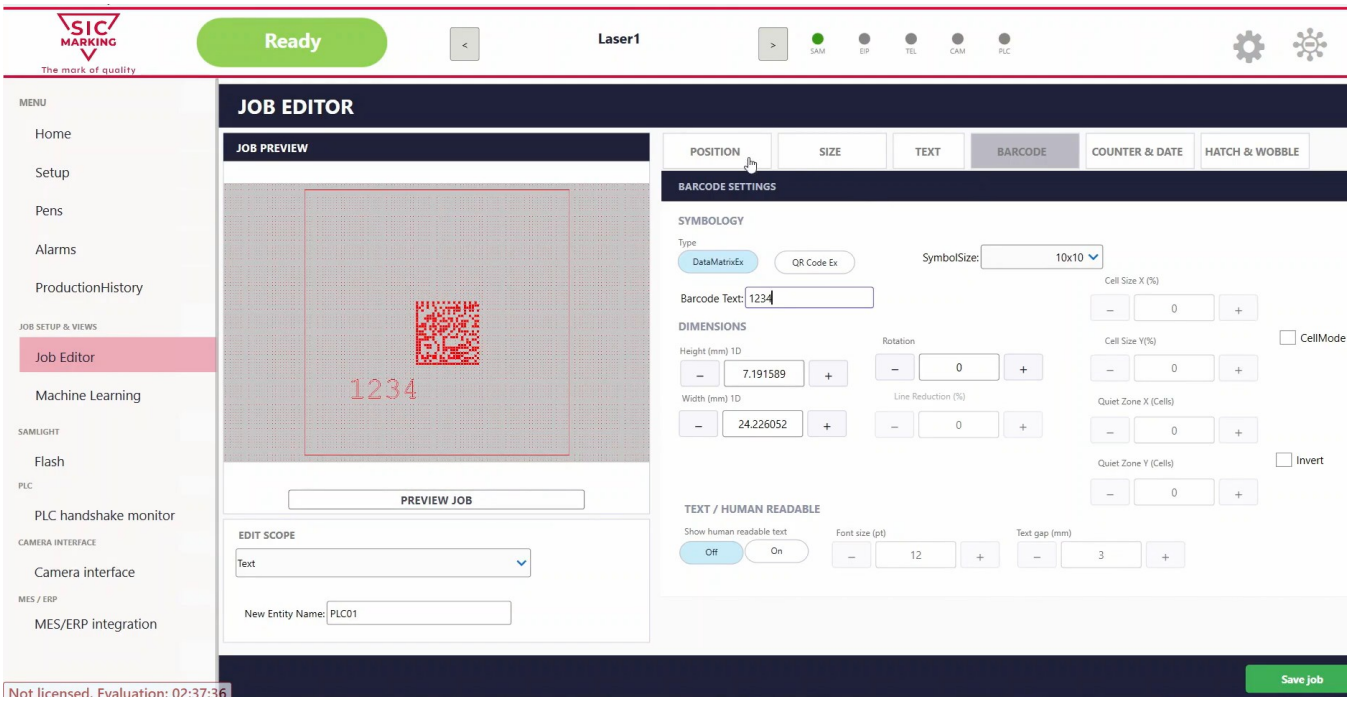
Setup Screen

The production screen. The operator selects the job, adjusts the position and size of the mark, and runs the marking cycle.



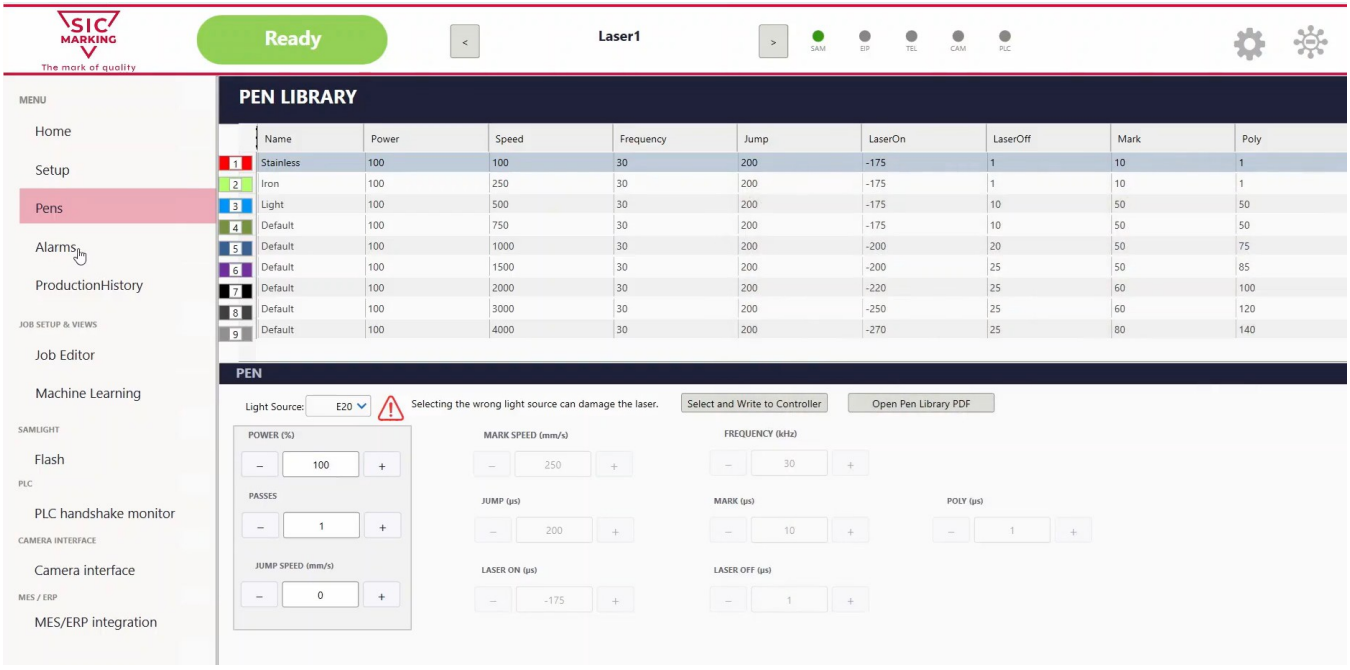
Job Editor

Editing the content of the loaded job: text, barcode position, and the parameters of each object in the job.



Pens

Marking parameters held per pen, including power, speed, frequency, and the jump, mark and laser delays. A dataset of presets lets an operator recall known-good settings per material.



Marking Control

- Text, barcode, DataMatrix, QR, counter and date/time objects
- Symbology, cell size, quiet zone, error correction and human-readable text
- Serial number format, step, start value, reset and beat count
- Per-pen power, speed, frequency, and jump, mark and laser delays
- Whole-mark position, rotation and size applied through the optic, without altering the job itself

Device Integration

Every laser operation goes through a single protocol driver. Nothing in the application talks to the laser directly, which keeps all device behaviour in one supported component.

Capability	Detail
Structured Addressing	One scheme covering selection, values, per-object data, flags, commands and status
Guide Pointer	The outline retraces automatically when the mark geometry changes
Flash Jobs	Store and load jobs held in the controller's flash memory
Job Preview	Captures the marking layout and renders it in the operator screen

Key Enablers

Platform capabilities that made this solution possible:

Capability	Application
Driver Toolkit	Purpose-built driver for the laser controller
Unified Namespace	Laser configuration exposed as a structured tag model
Display Engine	Operator screens with live preview rendering
Datasets	Pen library of material presets
Access Types	On-demand polling, so only the screen in use communicates with the laser

The Results

- **Single Integration Point** — all laser communication through one driver, with no device logic in the application
- **Operator-Level Control** — marking content and parameters changed from the HMI, with no engineering tool
- **Shorter Training** — three focused screens in place of a general-purpose engineering interface
- **Faster Changeovers** — job selection, content and position adjusted directly from the production screen
- **Fewer Marking Errors** — the operator sees a live preview of the layout before running the cycle
- **Reusable** — the driver serves any laser using the same controller software, not a single machine
- **Maintainable** — device behaviour documented and versioned in one component

This case demonstrates an embedded OEM machine HMI with a custom protocol driver for direct laser control.

In this section...