



Choose certainty.
Add value.

CERTIFICATE

Marc Lenoir

Has successfully completed the Functional Safety
Certification Program requirements for

Functional Safety Professional

In accordance with
IEC 61511:2016

Date issued: 11-February-2026
ID number: TP26051994
Field: Process Industry
Subject: Safety Instrumented Systems

Three handwritten signatures in blue ink, likely representing the certifier, the candidate, and the trainer.

G. Greil
Certifier

Dr. Michel Houtermans
Trainer

C U R R I C U L U M

- **Functional safety management:**
Life cycle concept, documentation requirements, verification, validation, assessments and audits, modifications
- **Hazard & Risk Analysis:**
Hazard identification, hazard analysis, risk reduction, safety function definition using FTA, HAZOP, LOPA, Risk matrix, risk graph
- **Planning the safety system:**
Planning for end users, integrators, and realization of safety systems, safety plan, verification plan, validation plan, safety requirement specification, requirements for suppliers, FAT, SAT
- **Hardware design:**
Hardware lifecycle, energize vs de-energize, low demand, high demand, demand mode, continuous mode, redundancy, diversity, voting, hardware fault tolerance, safe failure fraction, type A/B, architectural constraints, proof testing, diagnostic tests, measures to avoid and control failures
- **Hardware reliability:**
Reliability modeling, FMEDA, simplified equations, block diagrams, FTA, Markov, failure data, HFT, SFF, DC, CCF, SIL, PFH, PFD
- **Software design:**
Software lifecycle, embedded software, application software, utility software, fixed programming languages, limited variability languages, full variability languages, software architecture, V-model, measures to avoid failures
- **Operation and maintenance:**
Installation and commissioning, safety validation, operation, maintenance and repair, modification and retrofit, maintenance override
- **Experience:**
The holder of this certificate has more than 6 years experience in more than 2 functional safety projects