Common mistakes in Electronic Safety Systems
Welcome

- Tijmen Molema
- Liftinstituut: NOBO and AECO:
  - Worldwide Certification partner
- Product Specialist Certification
- Specialized in the IEC – 61508
  - Electronic safety
This presentation

- About **Programmable Electronic Safety Systems**: PESS
- Involves some new technology’s,
- Sharing our experience of last years
  - Of course, Safety is the base line
The modern lift

- ...Is full of electronics
- ...Is connected to a network
- ...Self Diagnoses for any faults
- But can all this “touch” the safeties?
World v.s. Elevator
The modern lift

- Programmable Electronic Safety Systems: **PESS**
- Electronic components / Software make safety decisions
  - Cheaper
  - Faster
  - More flexible
  - Safer?

  …*Show with an example*..
Example: overspeed governor

- **Cheaper**
  - Electronics are very price competitive

- **Faster**
  - Detection in software, no “tooths” on governor which cause delay
  - Can also act on Jerk ($v^3$)

- **Flexible**
  - Only 1 model needed for multiple speeds (configuration)

- **Safer?**
Other markets with PESS

- Self driving cars
  - Google: 2.1 million km driven before first accident
- Planes
  - More and more computers:
    - Hence, no good comparison possible
- For lifts, no data available
Trouble in paradise

- New technology's give new problems
  - Some due to minor understanding
  - Lift-cultural problems
  - Mixing of old and new thinking
Project planning

- Normal projects: first drafts and end product differ a lot
  - Mechanics: uncluttered
- PESS, this is a safety hazard
  - Black Box: hard to test, impacts are hard to see

*Think before you start! However…*
Options, options, options

- If we have safe information about speed and position, we can make
  - Overspeed
  - Final limed switch
  - Inspection end
  - UCMP
  - Short head/ Pit
  - ….

- All plans start small, but with the immense potential…

  *Stick to your plan!*
Documentation

- We stick to our plan with documentation
- But as engineers, we don’t like documentation
  - Value is not directly seen
- Requirements will get forgotten
  - And it was hard to test so…
  - Safety hazard!

*Use requirement tracking!*
Mixing safety and non safety

- Safety controller has position information of the car
  - So I can use it for non-safety functions too! Or not?...

- It is possible, but:

- Non safety code can and will influence safety code (if not, proof it)
  - Interrupts
  - Death-locks
Mixing safety and non safety

- All software becomes safety software…
- CRC of software is on certificate, so…
  - Every update: contact the NoBo
  - Long testing time for each update
  - Risk of hacking, security becomes a thing
- Only interesting in very big projects

Do not mix safety and non-safety software
Calculations

- A safety system always consist out of:
  - Sensor
  - Processing
  - Actuator
- For PESS, it is the same:
Calculations

- In PESS we have to calculate safety value of our system
- Everything in safety function shall be put into the math...

...Everything...
Calculations

- With the Sensor Processor and Actuator we’re not complete!
  - Power Supply is also part of safety!
- Is the actuator, really our actuator?
Calculations

- The actual actuator is… The brake!

_In the update of the EN81, you probably have to reserve a certain value for the brake_
Conclusion

- Think before you start → Make a proper **plan**
- **KISS**, and stick to your plan
  - Use requirement tracking
- **Do not mix** safety and non-safety software
  - Or be very aware of the impact
- Reserve a certain value of your **safety factor** for the brake to satisfy the new EN81-20+A1
  - Stil under discussion
Thank you for the attention!

Any questions?

Plan
Design
Make

What is not in here?
System limed
Mixing systems
Safety factors

What do I want to make?
Req Tracking