Università degli studi di Modena e Reggio Emilia

Dipartimento di Ingegneria “Enzo Ferrari”

Corso di Laurea Magistrale in Ingegneria Informatica



Modellazione di un sensore di rilevamento fuga di gas naturale, fault analysis e riduzione del rischio

pROGETTO DI CORSO

Nicola Corbo | Fabrizio Di Blasi | Informatica industriale ed IoT | a.a 2018/2019

# Sommario:

1. Introduzione
   1. Obiettivo del lavoro e spiegazione dell’ambito di progetto
2. Fault Analysis
   1. Analisi del rischio
   2. Fault tree
   3. Descrizione e cause del fault
3. Riduzione del rischio
   1. Calcolo del SIL
   2. PLR
   3. Mean Time to failure
4. Script Codesys
   1. Schema a blocchi dell’applicazione
   2. Script
5. Conclusioni
   1. Ottimizzazioni e miglioramenti futuri

# Introduzione:

In questo elaborato presentiamo l’analisi del rischio e degli errori di un sistema di rilevamento della temperatura all’interno di una cisterna di GPL.

Nel momento in cui la temperatura supera una soglia critica, simulata manualmente, il sistema andrà ad azionare una pompa di raffreddamento. Il raffreddamento della cisterna è stato modellato con un timer. In caso in cui la pompa è in fault si è scelto di effettuare un rilascio controllato in atmosfera per diminuire la pressione, e quindi la temperatura, all’interno della cisterna.

L’obiettivo è quello di modellare tale caso reale con uno script codesys che permetta di controllare e gestire eventuali problematiche ed effettuare tutte le analisi del caso in modo da prevenire ed aumentare il livello di sicurezza.

# Fault analysis:

per il progetto da noi esposto, si deve evitare il più possibile il rilascio, pur controllato, del gas in atmosfera poiché è un evento molto delicato

Un sistema si definisce sicuro solamente quando non reca danno alla vita umana o all’ambiente, perciò è bene studiare le conseguenze di un guasto.



Nella nostra analisi si vuole che un errore o malfunzionamento sia molto raro, più precisamente “occasionale” oppure “remoto”.

Fissata la categoria del malfunzionamento è possibile quindi ottenere la **classe di rischio**.

La classe più adeguata al nostro progetto è la *seconda* poiché il fault del sistema è sempre un evento indesiderabile, ed il costo da sostenere per ottenere un miglioramento è molto elevato

Vediamo però da cosa possono scaturire i malfunzionamenti del sistema con la *fault tree analysis*:

