Aziende: Overnet – Adecco

Corso: Ingegneria del Software

Documento: Specifiche per project work

Autore: Cosimo Adavastro

Data: 29/05/2019

Versione: 1.0

Sommario

[Introduzione 2](#_Toc9589270)

[Il Problema. 2](#_Toc9589271)

[Motore 2](#_Toc9589272)

[Tunnel congelamento 3](#_Toc9589273)

[Interfaccia fornita 4](#_Toc9589274)

[Programma da realizzare 4](#_Toc9589275)

[Gestione input bottoni 4](#_Toc9589276)

[Log 5](#_Toc9589277)

[Prodotti 5](#_Toc9589278)

# Introduzione

Il project work rappresenta una sperimentazione attiva dei contenuti appresi durante il percorso didattico formativo fin qui seguito.

Si è cercato di creare una simulazione della progettazione di un Sistema Embedded cercando di avvicinarsi il più possibile ad un caso reale risolvibile in 24 ore.

Il docente avrà sia il ruolo di Cliente da soddisfare che Project Manager con cui interloquire. Ad ogni domanda fatta da una persona singola del team verrà fornita una risposta pubblica allo scopo di chiarire il/i dubbio/i o colmare eventuali omissioni informando tutto il team.

Durante il project work i membri del team potranno interagire ma è importante che poi ognuno si concentri per fornire i prodotti richiesti nel tempo previsto.

Al termine i risultati verranno discussi e commentati in gruppo

# Il Problema.

Un’azienda che produce gelati ci chiede un sistema di controllo delle temperature per la fase di “Indurimento” del processo. L’indurimento avviene dopo la fase di Congelamento tramite un tunnel di raffreddamento tra -23/-25°C. Il gelato arriva nel tunnel in vaschette trasportate da un nastro trasportatore che, attraversato il tunnel, andranno alla fase di confezionamento del prodotto su un altro nastro trasportatore.

Tunnel Congelamento

Da

Congelamento

verso Confezionamento

Motore

Il tratto di nastro trasportatore che comprende il Tunnel di Congelamento è gestito da un motore elettrico apposito per poter garantire la permanenza di ogni vaschetta per il periodo necessario.

Il tutto attualmente funziona ma il Cliente si lamenta di sbalzi di temperatura non accettabile nel tunnel e del surriscaldamento del motore che spesso blocca la linea.

I nostri tecnici hanno analizzato il problema ed indicato quanto segue.

## Motore



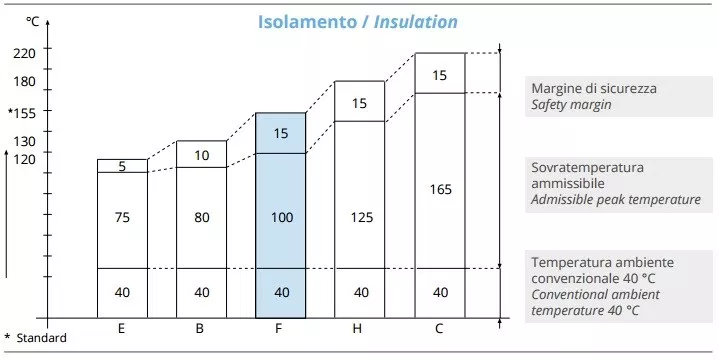
Il punto critico è la classe di isolamento.

La classe di isolamento definisce la qualità e l’insieme di trattamenti degli avvolgimenti usati nella costruzione di un motore elettrico. Per evitare dispersioni di corrente e cortocircuiti tra le fasi, l’avvolgimento di un motore elettrico viene impregnato di speciali resine.

In base alla qualità della resina utilizzata, questa garantisce il corretto funzionamento di un motore elettrico fino al raggiungimento di una certa temperatura.

Il parametro fondamentale è la sovratemperatura, cioè il Δ T. Maggiore è la classe di isolamento del motore, maggiore è la sovratemperatura alla quale il motore elettrico potrà resistere.

Il nostro motore è in classe F e può funzionare fino a 155°C secondo gli standard indicati nella figura sottostante.



Da prove effettuate ci si è accorti che il motore dovrà funzionare accanto ad un Refrigeratore che eroga calore e che la temperatura che produce il miglior rendimento si ottiene se non si superano mai gli 80°C – 85°C.

**Per fare ciò è stato aggiunto un sensore di temperatura che ne rileva la temperatura istantanea ed una ventola che dovrà essere usata per raffreddare l’ambiente quando la temperatura è nel range [70°, 80° +/- 5°C].**

## Tunnel congelamento

Il tunnel aveva un unico sensore di temperatura che automaticamente staccava la parte refrigerante, sono stati aggiunti altri tre sensori ed un apparato hardware che trasmette la media delle tre temperature misurate. **Tale media dovrà sempre essere tra -23°C e -25°C con una tolleranza di 0.25 °C.**

L’ipotesi di soluzione consiste nel fornire un Sistema Embedded in ambiente Windows Embedded con un programma C# che effettua tutti i controlli e le regolazioni.

Durante le prove manuali sono state scritte e testate le interfacce di controllo del Refrigeratore, del Motore e dei sensori di temperatura.

Il modello del sistema è il seguente.

Motore

Refrigeratore

Interfaccia Attuatori, Sensori, Input bottoni

Start Stop

Visualizza

# Interfaccia fornita

L’interfaccia viene fornita come .dll nel file SensoriAttuatori.dll.

La classe di interfaccia è **DevInterface** essa offre le seguenti proprietà e i seguenti metodi.

|  |  |
| --- | --- |
| Proprietà/Metodo | Descrizione |
| public DevInterface() { } | Costruttore vuoto per istanziare l’oggetto DevInterface |
| public string Xcmd = ""; | Una stringa che contiene il comando digitato su tastiera:   * S = Avvia il sistema (bottone Start) * F = Stoppa il sistema (Bottone Stop) * V = visualizza stato (Bottone Visualizza)   Per i dettagli vedere descrizione programma da implementare. |
| public void Init() | Permette di inizializzare l’interfaccia. Deve essere chiamata una sola volta prima di ogni altro metodo. |
| public void AttivaMotore() | Permette di attivare il motore (il motore non deve essere attivato prima che il Refrigeratore vada a regime). |
| public void StopMotore() | Permette di arrestare il motore. |
| public void AttivaVentolaPerMotore() | Permette di accendere la ventola di raffreddamento per il motore. |
| public void DisattivaVentolaPerMotore() | Spegne la ventola di raffreddamento per il motore. |
| public float DammiTemperaturaMotore() | Restituisce la temperatura corrente del motore. |
| public void AttivaRefrigeratore() | Accende il refrigeratore che impiegherà del tempo per arrivare al range di temperatura da controllare. |
| public void SpegniRefrigeratore() | Permette di spegnere il refrigeratore. |
| public float DammiTemperaturaRefrigeratore() | Restituisce la temperatura corrente (media dei tre sensori di temperatura) del tunnel di raffreddamento. |

L’interfaccia quando:

1. la temperatura del motore supera il valore della fascia di appartenenza F=155°C usa l’altoparlante del PC per emettere due beep contigui e a frequenza e durata diversa;
2. la temperatura del tunnel di congelamento sfora i range di temperature permesse {**-23°C e -25°C con una tolleranza di 0.25 °C.**} emette un beep, tale beep sarà emesso periodicamente fino a quando la temperatura del tunnel dopo l’accensione del refrigeratore non raggiunge la soglia minima.

# Programma da realizzare

Si chiede di realizzare un programma (Progetto console) che utilizzando l’interfaccia fornita avvii il sistema e controlli la temperatura del tunnel e del motore mantenendoli nei range su descritti.

Il programma dovrà fornire un log su un file .csv per analisi dei risultati.

L’interfaccia dovrà essere istanziata all’avvio del programma e subito dopo (o prima di iniziare qualsiasi operazione su di essa) si dovrà chiamare il metodo Init.

## Gestione input bottoni

L’input da tastiera che simula i bottoni è fornito dall’interfaccia dopo l’inizializzazione nella variabile globale d’istanza Xcmd direttamente leggibile dal programma. Xcmd viene compilata in modo asincrono dall’interfaccia leggendo i caratteri digitati da tastiera, essa può contenere:

* S = Avvia il sistema (bottone Start) – devono essere avviate tutte le operazioni di avvio dell’impianto si consiglia di avviare il motore solo quando il refrigeratore ha una temperatura nel range ammesso;
* F = Stoppa il sistema (Bottone Stop) – tutte le operazioni pendenti debbono essere concluse ed il programma dovrà comunicare a video che sta per chiudersi attendendo un invio per uscire;
* V = visualizza stato (Bottone Visualizza) – dovrà visualizzare i seguenti dati: Temperatura del motore, Temperatura del tunnel, indicazione se è stato effettuato l’avvio, indicazione se il motore è attivo o disattivo, indicazioni se la ventola è attiva o disattiva, indicazioni se il refrigeratore è attivo o disattivo.

# Log

Il log dovrà essere scritto nella cartella C:\Temp, nel caso la cartella non sia presente si consiglia di crearla prima di eseguire il programma manualmente (volendo può essere anche creata da programma).

Il nome del file di log dovrà essere logSensoriAttuatori.csv.

All’avvio del Log bisognerà verificare se è già presente un file di log e se presente copiarlo in un nuovo file con il seguente nome annomesegiornooraminutisecondi\_ logSensoriAttuatori.csv (es. 2019523155842\_logSensoriAttuatori.csv), i valori da usare come anno, mese, giorno, ora, minuti, secondi dovranno essere ricavati da una variabile DateTime inizializzata al valore DateTime.Now del momento in cui si va a scrivere.

Dopo aver copiato il file presente è necessario crearne uno nuovo inserendo la prima riga con il seguente formato:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DataOra | Tm | Tr | Avviato | Motore | Ventola | Frigo |

Dove:

1. DataOra è la data ed ora della scrittura della riga nel formato gg/mm/aaaa hh:mm:ss
2. Tm è la temperatura del motore;
3. Tr è la temperatura del tunnel (refrigeratore);
4. Avviato true se il sistema è avviato altrimenti false;
5. Motore true se il motore è avviato altrimenti false;
6. Ventola true se la ventola è avviata altrimenti false;
7. Frigo true se il sistema di raffreddamento è avviato altrimenti false;

Esempio

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DataOra | Tm | Tr | Avviato | Motore | Ventola | Frigo |
| 23/05/2019 15:58:42 | 40 | 25 | False | False | False | False |
| 23/05/2019 15:58:45 | 40 | 25 | False | False | False | False |

# Prodotti

Si chiede di effettuare l’analisi tecnica prima della codifica realizzando il diagramma statico delle classi, i casi d’uso più significativi e la descrizione di quello che si intende implementare considerando le precondizioni, le post condizioni, l’algoritmo e le eccezioni.

Ci si attende un .exe funzionante.