



UNIVERSITA' POLITECNICA DELLE MARCHE

FACOLTA' DI INGEGNERIA

Corso di Laurea Triennale in Ingegneria Informatica e dell'Automazione

"HMI nell'Industria: Progettazione di un'Interfaccia Intuitiva e User-Friendly per il Controllo dei Processi Produttivi"

"HMI in Industry: Design of an Intuitive and User-Friendly Interface for Process Control"

Relatore:

Prof. **Gianluca Ippoliti**

Correlatore:

Ing. **Stefano Fuligna**

Tesi di Laurea di:

Geremia Mattioli

A.A. 2022/2023

Ringraziamenti

Vorrei usare queste poche righe per ringraziare tutte le persone che hanno fatto parte di questo percorso per me lungo e difficile, passando attraverso Milano e poi Ancona.

Ringrazio la mia Famiglia per avermi sempre sostenuto e appoggiato, credendo in me ogni singolo giorno.

Ringrazio mia sorella Virginia, che mi ha sempre detto le cose come stavano, aiutandomi ad uscire dai momenti più duri.

Ringrazio i miei nonni che mi hanno insegnato cosa sia la passione per la vita e per il lavoro.

Ringrazio Giorgia, Francesca ed Emma, amiche da una vita, non servono altre parole, grazie.

Grazie a tutte le persone che ho incontrato al museo: Nicole, Sabrina, Elena, Sofia, Marco e Tommaso, siete stati i migliori colleghi che potessi desiderare e siete degli amici incredibili.

Grazie a Sofia, Ludovica e Alice siete le persone che mi hanno ridato aria nel momento in cui ne avevo bisogno.

Grazie ai ragazzi del '98, amici da quando abbiamo iniziato a gattonare e compagni di mille esperienze.

Grazie a Stefano, la tua guida e il tuo supporto hanno reso l'esperienza del tirocinio molto formativa per me.

Grazie a Francesca, mi hai aiutato a credere nel fatto che potevo farcela, e mi hai reso la versione migliore di me stesso.

Per ultimo, ringrazio me stesso per non aver mai abbandonato il sogno di terminare questo percorso e per essermi sempre rialzato dopo ogni caduta.

Indice

Ringraziamenti	3
Introduzione	6
Elenco delle figure	7
1 INTERNET OF THINGS	8
2 LINEA 12 PER IL ROTTAME	13
3 HMI 2.0	16
4 CRITICITA' DA RISOLVERE	24
5 SCRIPT PER IL TRACCIAMENTO DEL PESO CARICATO	27
6 SCRIPT PER IL FERMO ORARIO	29
7 SCRIPT SALVATAGGIO STATI RUN E FERMO	31
Bibliografia	33

Introduzione

L'elaborato inizia con un'analisi delle caratteristiche salienti dell'Industria 4.0 e dell'Internet of Things (IoT), focalizzandosi sulla loro rilevanza ed impatto. Successivamente approfondisce l'analisi della linea di produzione assegnata, evidenziando la necessità di progettare un'interfaccia conforme alle principali caratteristiche di queste tecnologie avanzate.

Il cuore del lavoro si concentra sulla progettazione di un Human-Machine Interface (HMI) mirato a semplificare l'utilizzo della macchina, contemporaneamente alla raccolta di dati operativi cruciali per la linea produttiva. L'obiettivo principale è migliorare l'efficienza del settore aziendale e facilitare un'analisi remota e completa dei processi produttivi.

Attraverso la progettazione di un HMI intuitivo e orientato al miglioramento delle prestazioni, si mira a favorire una gestione più efficiente e a consentire un monitoraggio dettagliato delle attività produttive. Tale approccio riflette l'impegno verso una implementazione pratica e strategica delle tecnologie dell'Industria 4.0, contribuendo così al progresso ed all'ottimizzazione dei processi industriali.

Elenco delle figure

- 1 Numero di dispositivi connessi in miliardi*
- 2 Linea 12 della produzione*
- 3 Linea 12 della produzione*
- 4 Magazzino della zorba*
- 5 Pianta della linea 12*
- 6 Header del HMI*
- 7 Box di selezione dell'ordine nella home*
- 8 Bottone di scelta del tipo di lavorazione nelle fasce orarie*
- 9 Led per il peso e pulsanti per il fermo impianto*
- 10 Pianta della linea 12 con led degli allarmi*
- 11 Tabella con elenco degli allarmi passati*
- 12 Grafico con funzionamento della linea 12 nel tempo*
- 13 Finestra di causalizzazione dei fermi*
- 14 Tabella dei fermi da causalizzare*
- 15 Elenco dei fermi causalizzati con motivazioni e durata*
- 16 Codice dello script per la creazione dello stato di caricamento*
- 17 Codice dello script per la fine dello stato di caricamento*
- 18 Codice dello script per il tracciamento del peso caricato*
- 19 Codice dello script per il fermo orario*
- 20 Codice dello script per il salvataggio degli stati sul server*

1 INTERNET OF THINGS

Nella mia esperienza presso Profilglass s.p.a., sono stato assegnato al settore IT dell'azienda, nello specifico all'ufficio dedicato all'Internet delle cose, cioè IoT, un termine che, per esteso, rappresenta l'Internet of Things. Questa denominazione fu coniata nel 1999 da Kevin Ashton, ricercatore presso il Massachusetts Institute of Technology di Boston, per descrivere una rete di comunicazione tra oggetti intelligenti, ovvero oggetti digitali capaci di fornire servizi tramite l'uso organizzato di sensori e attuatori, collegati da microcontrollori come smartphone e tablet.

L'idea di sviluppare una rete parallela a quella utilizzata dagli utenti, in cui i dispositivi possono autonomamente ricevere o trasmettere informazioni attraverso i sensori, ha aperto la strada a nuove forme di fabbricazione industriale, nonché all'interpretazione dell'ambiente metropolitano, al trasporto intelligente e al concetto di casa. In sostanza, ha inaugurato un nuovo concetto di vita.

Mentre i Cyber-Physical Systems hanno consentito un'integrazione profonda con l'ambiente fisico, l'IoT rappresenta la rete attraverso la quale i CPS possono comunicare e cooperare tra loro senza richiedere azioni umane. Nella visione di una Smart Factory, ogni macchinario utilizzato nella produzione deve poter comunicare con gli altri dispositivi e contribuire alle attività di produzione. L'integrazione di dispositivi connessi tramite tecnologie IoT apre le porte a nuove possibilità, come il controllo remoto della produzione e dei macchinari, o il monitoraggio in tempo reale dei dati operativi delle macchine e delle prestazioni produttive, permettendo una gestione automatizzata di tutti gli aspetti della produzione, e rendendola dunque più efficiente.

Tra le caratteristiche abilitanti dell'Industria 4.0, l'IoT è quella destinata ad avere nel breve periodo un ruolo ed un impatto maggiori sulla vita quotidiana delle persone, come evidenziato dal grafico che indica il numero di dispositivi connessi per anno.

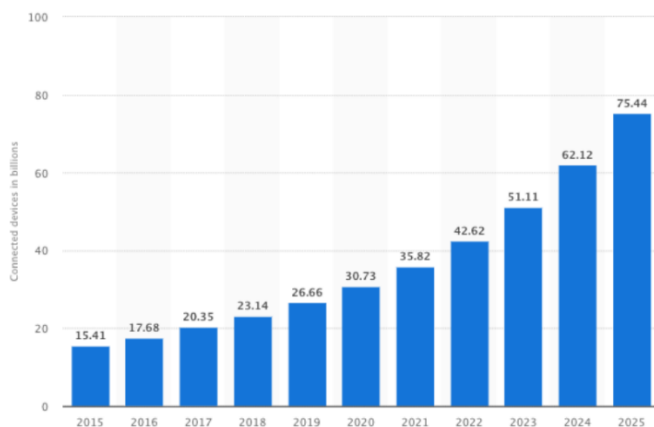


Figura 1

Uno dei motivi che ha portato a questi risultati è stato il progressivo abbassamento dei costi dei sensori, attuatori e microcontrollori, unito ad una continua ingegnerizzazione, che ne ha ridotto le dimensioni, espandendo così le possibilità di utilizzo e la funzionalità di tali dispositivi.

Un'altra caratteristica importante è la varietà di tipologie di connettività differenti che l'IoT racchiude al suo interno. Infatti, le possibilità di comunicazione di un dispositivo sono estremamente varie:

- **WIRELESS PERSONAL NETWORKS:** racchiudono le tecnologie a corto raggio (<100m) e attualmente sono le più diffuse
- **WIRELESS LOCAL AREA NETWORKS (WLAN):** raggio di comunicazione fino ad 1 km. La principale è il wi-fi, che ha aumentato ancora la sua diffusione grazie alla crescente presenza di assistenti vocali per la casa
- **LOW-POWER WIDE AREA NETWORKS:** tecnologie che promettono una grande attenzione alla durata della batteria dei dispositivi e massimizzano il raggio di comunicazione, fino a 20 km
- **TECNOLOGIE 5G:** categoria ancora in fase di sviluppo che promette una banda elevata (circa 1Gbit/s), affiancata ad una latenza estremamente bassa (circa 1ms)
- **RETE CABLATA:** anche se l'IoT è comunemente pensata come una connessione wireless, ci sono vari scenari nei quali una rete cablata risulta preferibile, per esempio all'interno dell'ambiente industriale, nel quale la tecnologia ethernet o fieldbus risulta ancora importante

- ALTRI: tecnologie come la rete cellulare, la rete satellitare e altre reti sono destinate a lasciare il passo a tecnologie più moderne e adatte al caso d'uso dell'IoT

La cooperazione tra dispositivi intelligenti rappresenta l'idea fondamentale dell'Internet delle cose (IoT). Tuttavia, la vasta quantità di dati gestita da queste reti, comunemente definita come BIG DATA, rende cruciale l'accessibilità di tali dati dall'esterno. Ciò solleva una questione di sicurezza significativa: cioè che i dati siano accessibili attraverso la rete Internet, ma al contempo non voglio che diventino di pubblico dominio, ma siano fruibili solo da coloro che ne hanno il diritto.

Esistono scenari comuni in cui la privacy e la sicurezza degli utenti possono essere compromesse. Ad esempio, in una smart home, il compromesso anche di un singolo dispositivo potrebbe consentire l'accesso non autorizzato ad ogni altro dispositivo, che potrebbe variare da un innocuo microonde a una serratura smart di casa, oltre a dati di accesso memorizzati nei dispositivi.

Ancora più preoccupante è la possibilità di accedere a dispositivi medici smart, come un pacemaker, con il rischio di causare danni all'utente. Con l'introduzione sul mercato di una vasta gamma di dispositivi eterogenei, si presentano nuove sfide riguardo alla segretezza, integrità dei dati, accesso e autenticazione. Va notato che questi problemi non sono limitati alle intrusioni a livello software, poiché la natura dell'IoT prevede l'installazione di dispositivi anche in zone remote e non controllate; di conseguenza, un utente senza adeguate misure di sicurezza potrebbe accedere direttamente all'hardware ed estrarne dati che potrebbero includere informazioni sensibili.

Tutte queste problematiche complicano la questione dell'installazione e della gestione, rendendola tutt'altro che banale e semplice. Ciò rende necessario l'inserimento di misure contenitive durante la progettazione di un'architettura di questo tipo.

Come già accennato, esiste un sottoinsieme specifico dell'IoT che riveste un particolare interesse, ovvero l'Industrial Internet of Things (IIoT). La differenza tra questa e altre categorie non è sempre legata alle funzionalità offerte, poiché un dispositivo consumer può svolgere lo stesso compito di uno industriale. Gli obiettivi dell'IIoT includono l'ottimizzazione del processo produttivo, la raccolta di dati di campo, l'analisi dei parametri di funzionamento dei macchinari e il tracciamento dei tempi di produzione.

Per questo motivo avrà dei parametri progettuali leggermente diversi che possiamo così riassumere:

- SICUREZZA: cruciale per l'IoT, ma ovviamente ancora più importante nel caso dell'IIoT, infatti la manomissione o l'alterazione di un processo produttivo può portare alla perdita di somme importanti; la manomissione della rete elettrica può causare danni fisici anche a molte persone. Per questo motivo le soluzioni devono integrare al loro interno meccanismi di sicurezza all'avanguardia, meccanismi di autenticazione e crittografia.
- INTEROPERABILITA': le tecnologie IoT devono co-esistere con un numero importante di tecnologie legacy già esistenti nello stabilimento. Per cui è fondamentale la capacità di integrarsi e supportare un ampio ventaglio di protocolli e dati diversi.
- SCALABILITA': le reti industriali devono essere in grado di supportare un elevato numero di dispositivi ed è necessario che siano in grado di scalare per supportare a loro volta un elevato numero di sensori differenti.
- PRECISIONE: i processi industriali richiedono nella maggior parte dei casi un'elevata sincronizzazione ed allo stesso tempo i sistemi di controllo della qualità devono essere in grado di rilevare ogni minima variazione nella fattura del prodotto.
- BASSA LATENZA: collegato al punto sopra, è necessario che i dispositivi possano captare le variazioni senza ritardi in modo da rendere possibili le soluzioni correttive quasi in tempo reale; ogni ritardo, sia di rilevamento, sia di valutazione o di esecuzione può portare ad una perdita in termini di qualità o, nei casi peggiori, di sicurezza per il lavoratore.
- PROGRAMMABILITA' E MANUTENIBILITA': una caratteristica dei sistemi IoT è quella di essere quasi completamente riprogrammabili, nel campo industriale queste operazioni devono essere possibili con la massima flessibilità e adattabilità, per esempio con interventi da remoto o sul campo durante il funzionamento. È inoltre fondamentale che il dispositivo sia facilmente manutenibile, sia per la sostituzione di un pezzo che per la riconfigurazione.
- AFFIDABILITA': i sistemi industriali hanno la necessità di continuare ad operare per lunghi periodi ed in condizioni non ideali, come caldo, freddo, vibrazioni o pressione. Per cui è necessario che le soluzioni IIoT riescano ad operare nelle stesse condizioni e con gli stessi requisiti di longevità.

- RESILIENZA: processi e sistemi critici sono realizzati in modo tale che in caso di errore o fallimento, l'intero sistema non smetta di funzionare; in caso di fallimento di un'unità operativa i suoi compiti possono essere rindirizzati ad una di backup oppure ad altre unità che sono in grado di gestirli, allo stesso modo per le soluzioni IIoT è necessario che siano fault tolerant: la perdita di un sensore o di connettività non deve pregiudicare il completamento dei processi di sistema

2 LINEA 12 ROTTAME

Durante il mio tirocinio, mi è stato assegnato il compito di semplificare e rendere più comprensibile l'Interfaccia Uomo-Macchina (HMI) della Linea 12, una parte del processo produttivo dedicato alla gestione del rottame di alluminio. Questo materiale viene riciclato all'interno dell'azienda per essere successivamente trasformato in prodotti di recupero. La mia responsabilità si è estesa anche alla Linea 12Bis, che segue immediatamente la Linea 12 ed è considerata come la sua continuazione. L'obiettivo è migliorare l'accuratezza con cui il materiale viene suddiviso, aumentando così la certezza sulla composizione del materiale recuperato.



Figura 2 e 3

Il processo operativo della Linea 12 può essere descritto nel modo seguente: all'ingresso della linea, troviamo della "zorba" (consultare la figura 4), un termine utilizzato per indicare i rifiuti metallici che sono stati precedentemente sminuzzati. Questi rifiuti sono classificati in tre dimensioni: grande, media e piccola. Tale differenziazione comporta la necessità di adottare programmi distinti per la macchina, poiché, in base alla dimensione, il portello di ingresso deve essere aperto in modo diverso.



Figura 4 l'operatore sceglie la ricetta da eseguire, ovvero seleziona quale tipo di materiale riceverà in ingresso la linea, carica la tramoggia con tutto il materiale da separare, circa 20 tonnellate e poi avvia il ciclo di lavorazione tramite l'HMI che si trova nella adiacente cabina di controllo. Sotto la tramoggia è presente un nastro metallico (numero 10) che conduce il materiale verso un nastro "a collo di cigno" (numero 4) il quale poi scarica a sua volta il materiale su un piano vibrante (numero 9); quest'ultimo serve, mediante il movimento sussultorio, a stendere in maniera omogenea la zorba, ed a suddividerla in singoli pezzi che poi potranno essere analizzati. Successivamente il nastro si divide in due piani vibranti (numeri 7 e 8), posti a monte delle macchine ad X-ray, questo per far sì che il materiale in ingresso nelle macchine non sia eccessivo, e quindi per assicurarsi una analisi precisa e corretta dei rifiuti. Le macchine ad X-ray permettono la suddivisione dei rifiuti in base al tipo di metallo di cui sono composti: all'uscita positiva troveremo i rifiuti di alluminio, mentre all'uscita negativa troveremo tutto il resto dei metalli, tra cui anche rifiuti di alluminio legati ad altri materiali, come ferro, zinco o rame. L'uscita positiva sversa il materiale in un box d'uscita, tramite i nastri 1 e 2, dai quali può essere trasportato verso i laminatoi, per essere lavorato, mentre l'uscita negativa converge in un altro box, da cui i rifiuti vengono ancora una volta presi, dal nastro 5, passati al nastro 6 ed infine al "collo di cigno" (numero 3); per poi essere passati in ingresso alla linea 12bis, con un funzionamento del tutto identico alla 12.

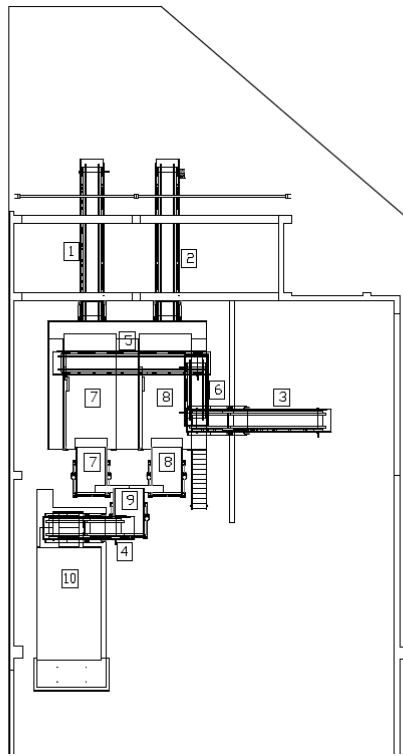


Figura 5

Questa linea era già fornita di un HMI, ed il mio compito è stato quello di modificarlo e di renderne più semplice l'utilizzo per tutti gli operatori che avrebbero dovuto poi lavorarci; inoltre, durante il tempo dedicato in azienda, sono emersi altri problemi sulla macchina, che la vecchia interfaccia

non era predisposta a risolvere, per cui ho provveduto a modificare anche le funzionalità che il nuovo HMI implementava, per rendere più sicuro l'utilizzo della linea e per migliorare la quantità e la qualità dei dati che venivano immagazzinati.

Alcuni esempi di queste modifiche applicate sono:

- Differenziazione degli stati macchina (carico, lavoro, fermo).
- Possibilità di spegnere la linea con un fermo orario prestabilito.
- Inserimento di una tabella che viene automaticamente riempita con i dati di carico della tramoggia iniziale per tenere traccia di quanto materiale viene lavorato.
- Layout di gestione dei fermi, con possibilità di casualizzarli e di dividerli in più parti.

3 *HMI 2.0*

Il periodo iniziale del mio lavoro è stato dedicato ai corsi di formazione su Ignition disponibili sul sito di Inductive Automation University. Questi corsi mi hanno fornito le competenze necessarie per iniziare a lavorare sull'Interfaccia Uomo-Macchina (HMI) che stavo sviluppando.

La prima fase del mio intervento è stata dedicata all'inserimento di tutti i tag su Ignition, ognuno dei quali è stato collegato a un singolo ingresso o uscita dei PLC presenti sulla linea in fabbrica. Questa operazione è fondamentale sia per monitorare in tempo reale lo stato della macchina e di ogni sua parte, che per avere la possibilità di operare su di essa da remoto, evitando la necessità di spostarsi direttamente negli spazi dedicati alla produzione.

Complessivamente, ho creato 247 tag per entrambe le linee di lavoro (Linea 12 e Linea 12Bis); questi sono suddivisi tra tag di comunicazione dal PLC allo SCADA, quelli di comunicazione dallo SCADA al PLC e infine quelli inseriti durante le fasi successive di progettazione, che sono tag ad uso interno del programma, e non sono collegati a nessun ingresso o uscita dei PLC.

Prendendo in considerazione la sola linea 12, l'HMI si divide in 4 macro-componenti

- L'header
- La pagina di home
- La pagina dello stato macchine
- La pagina dei charts (non sviluppato da me)

L'header della pagina, come mostrato nella figura 6, è creato utilizzando gli strumenti messi a disposizione da Ignition, è composto dal logo dell'azienda, da tre pulsanti che consentono la navigazione tra le pagine e da un menù a tendina che consente di selezionare la linea di cui si desiderano visualizzare i dati. Inoltre, è presente un pulsante di accesso (log in), che consente all'operatore di entrare nel programma con le proprie credenziali. Questo è particolarmente importante, poiché alcune funzioni dell'Interfaccia Uomo-Macchina (HMI) sono accessibili solo ai responsabili di reparto.

Oltre a ciò, è presente un LED che indica se l'interfaccia è attualmente connessa o disconnessa dalla macchina. In altre parole, fornisce un indicatore visivo sullo stato della connessione, segnalando se stiamo ricevendo dati in tempo reale o se l'interfaccia è offline.



Figura 6

Sulla pagina home ho inserito tutti i comandi di movimentazione della linea:

- Il box di inserimento del lavoro, che rappresenta il lavoro da effettuare e successivamente elaborato dalla macchina, è progettato per registrarsi, sia a fianco della macchina che nel server, attraverso la creazione di una tabella su SQL. Questo box è composto da tre fattori principali: la selezione dell'ordine, la selezione della portata e la selezione della ricetta, come illustrato nella figura 7.

Home / Movimentazione

CARICAMENTO RICETTA	DATI IN MACCHINA
SELEZIONA ORDINE <input type="text"/>	ORDINE IN MACCHINA <input type="text"/>
SELEZIONA PORTATA <input type="text" value="3"/> <input type="button" value="Carica"/>	PORTATA <input type="text" value="3"/>
SELEZIONA RICETTA <input type="text" value="Zorba Media"/> <input type="button" value="Carica"/>	RICETTA <input type="text" value="Zorba Media"/>

Figura 7

- Ci sono due pulsanti di selezione, che consentono di variare il tipo di lavorazione da diurno a notturno. La principale differenza tra i due, consiste nel fatto che, se la macchina è configurata per operare in regime notturno e la tramoggia non termina il materiale in ingresso (ovvero se la bilancia segna meno di 30 kg), il nastro si ferma e la linea si spegne. Invece, in regime diurno, la macchina continua a lavorare, inviando un segnale che permette agli operatori di riempire nuovamente la tramoggia con la zorba.

TIPO DI LAVORAZIONE	TIPO DI LAVORAZIONE
<input type="button" value="DIURNO"/>	<input type="text" value="DIURNO"/>
<input type="button" value="NOTTURNO"/>	

Figura 8

- Un LED che indica il peso in tempo reale misurato dalla tramoggia (vedi figura 9).
- Una rotella di selezione dell'orario, che consente di impostare un momento specifico in cui si desidera che l'impianto si spenga. Accanto ad essa, c'è un pulsante corrispondente che attiva o disattiva la funzione di fermo

programmato della linea. Se il fermo è disattivato, l'impianto continuerà a lavorare anche se si raggiunge l'orario stabilito nel menu (vedi figura 9).

- Un pulsante di fermo rapido dell'impianto che consente di interrompere rapidamente la lavorazione. Premendolo nuovamente, l'impianto riprenderà l'attività. Questo pulsante è visibile e utilizzabile da qualsiasi operatore (vedi figura 9).
- Un pulsante di blocco dell'impianto visibile solo se il login è stato effettuato dal responsabile. Questo pulsante consente di fermare immediatamente la linea. Sebbene possa comportare rischi e potenziali danni per la linea, è inserito come misura di sicurezza. Serve a garantire che, se il responsabile ferma la linea per manutenzione o altre ragioni, nessun operaio può accidentalmente farla ripartire (vedi figura 9).



Figura 9

- Una mappa della linea con gli allarmi che monitorano se tutti i componenti delle macchine stiano lavorando in maniera corretta, altrimenti si illuminano e avvertono l'operatore che c'è la necessità di un intervento (figura 10). In questo caso, risultano in errore, perché la linea era scollegata dalla rete aziendale, motivo per cui gli allarmi risultano tutti rossi.
- Due led che indicano in quale stato si trova la linea: se sta lavorando, oppure se è ferma

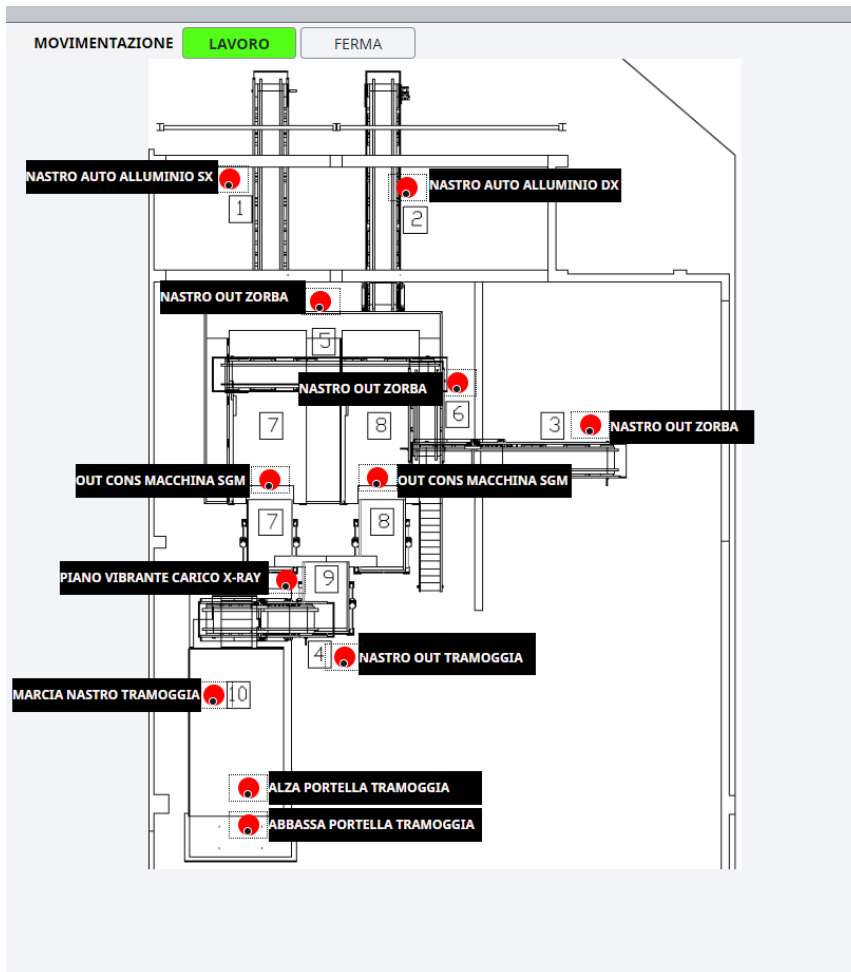


Figura 10

Nella pagina “Stato macchina e allarmi” è visibile una suddivisione più ordinata, ovvero una parte destra ed una sinistra con funzioni ben distinte.

Nella parte destra del layout:

- La mappa della linea con i bottoni d’allarme presenti anche nella home.
- Un bottone che apre la finestra dello storico degli allarmi, ovvero prende i dati che sono stati salvati sul server che gestiamo con SQL e riempie la tabella in ordine cronologico con i vari allarmi, indicando l’orario di inizio e fine dell’allarme (figura 11).

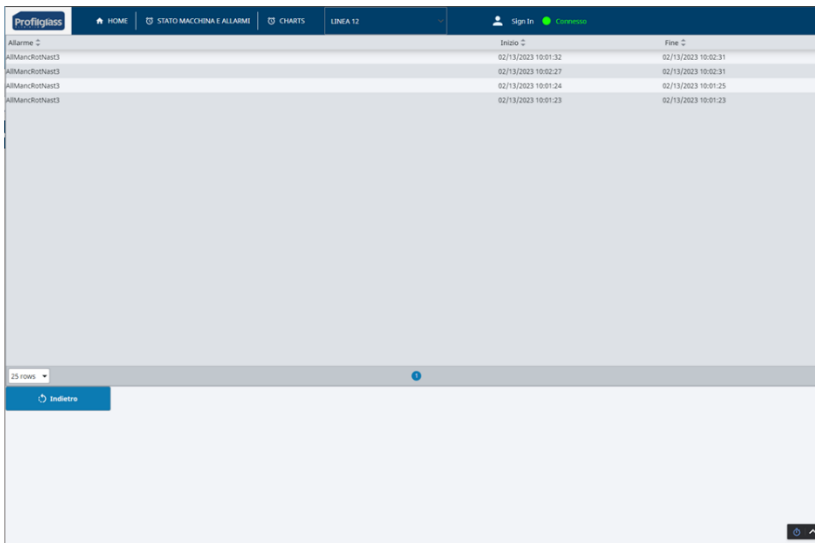


Figura 11

La parte sinistra si compone di:

- Un grafico che illustra lo stato della linea nel tempo: cioè si può selezionare il periodo di tempo che si vuole osservare, ed il grafico mostrerà con 3 colori diversi (verde, rosso e arancione) in quale momento la linea era rispettivamente al lavoro, ferma o nello stato di carico (figura 12).

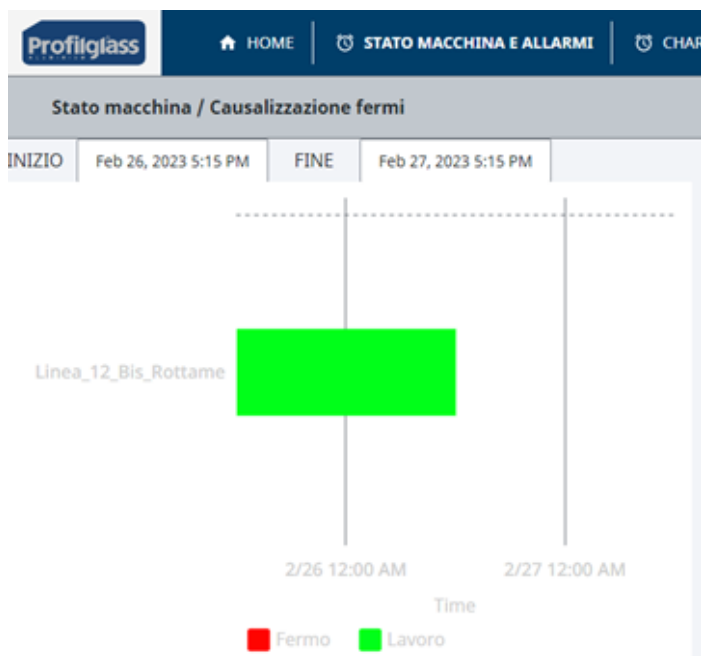


Figura 12

- Nell'HMI, è presente una tabella automaticamente compilata, grazie a uno script, con tutti i fermi subiti dalla macchina che devono ancora essere giustificati (vedi figura 14). Un pulsante "Causalizza Fermo" appare una volta selezionato un fermo dalla tabella; utilizzando questo pulsante, è possibile inserire la motivazione per cui l'impianto non ha operato in quel determinato periodo. La selezione della causa avviene attraverso un menù a tendina che offre diverse opzioni di problemi. In un secondo menù a tendina, è possibile specificare quale parte della linea ha riscontrato il problema (vedi figura 13). In alternativa, è possibile inserire direttamente la descrizione del problema in un box di testo che compare quando si seleziona "Altro" nel primo menù. Riguardo a questa funzione, mi è stato richiesto di apportare modifiche al codice per aggiungere una funzionalità particolarmente importante, che illustrerò più avanti.

The screenshot shows a web form titled "CAUSALIZZAZIONE FERMI". It includes the following fields and controls:

- ORA INIZIO:** Wed Feb 15 2023 17:01:16
- ORA FINE:** Fri Feb 17 2023 15:18:50
- ORA FINE:** 20/2/2023 11:28:00 (text input field)
- DESCRIZIONE:** FERMIV OPERATIVI (dropdown menu)
- TIPOLOGIA:** MANCANZA MEZZI (dropdown menu)
- Buttons:** A blue button labeled "Causalizza" is located at the bottom left of the form.

Figura 13

- Un pulsante “storico fermi” che ci mostra una tabella con tutti i fermi che la macchina ha subito con le relative motivazioni e momenti di inizio e fine (figura 13 e 15).

CAUSALIZZA FERMI	
Inizio ↕	Fine ↕
02/17/2023 16:55:21	02/20/2023 08:12:37
02/15/2023 17:01:16	02/17/2023 15:18:50
02/17/2023 15:41:02	02/17/2023 15:57:54
02/17/2023 15:57:58	02/17/2023 16:55:20
02/20/2023 08:14:42	

1

CAUSALIZZA FERMO STORICO FERMI

Figura 14

Inizio ↕	Fine ↕	Ordine ↕	Tipologia ↕	Descrizione ↕
02/20/2023 08:14:34	02/20/2023 08:14:42	rtx 4050		
02/17/2023 16:55:21	02/20/2023 08:12:37	rtx 4050		
02/17/2023 15:57:58	02/17/2023 16:55:20	rtx 4050		
02/17/2023 15:41:02	02/17/2023 15:57:54	rtx 4050		
02/15/2023 17:01:16	02/17/2023 15:18:50	rtx 4050		
02/15/2023 16:59:11	02/15/2023 17:01:16	rtx 4050	PIANO VIBRANTE	GUASTO ELETTRONICO/AUTOMAZIONE
02/15/2023 16:31:08	02/15/2023 16:31:32	rtx 4050		
02/15/2023 16:07:14	02/16/2023 16:09:55	rtx 4050		
02/15/2023 14:16:02	02/15/2023 15:17:07	rtx 4050	SPENTA	SPENTA
02/15/2023 11:20:06	02/16/2023 12:54:31	rtx 4050	ELETTRICA	MANUTENZIONE PROGRAMMATA
02/15/2023 08:25:52	02/15/2023 16:55:41	rtx 4050	PIANO VIBRANTE	GUASTO ELETTRICO
02/13/2023 16:40:53	02/13/2023 16:48:48	rtx 4050	PIANO VIBRANTE	GUASTO ELETTRICO
02/13/2023 16:36:16	02/13/2023 16:40:47	rtx 4050		
02/13/2023 16:36:51	02/13/2023 16:39:11	rtx 4050		
02/13/2023 16:18:54	02/13/2023 16:20:24	rtx 4050		
02/13/2023 16:05:06	02/13/2023 16:05:03	rtx 4050	PIANO VIBRANTE	GUASTO ELETTRONICO/AUTOMAZIONE
02/13/2023 16:04:59	02/13/2023 16:05:03	rtx 4050		
02/13/2023 16:04:32	02/13/2023 16:04:32	rtx 4050		
02/13/2023 15:58:22	02/13/2023 15:59:27	rtx 4050		
02/10/2023 10:30:29	02/10/2023 11:08:43	rtx 4050	ERRORE	FERMI PIANIFICATE NON PER MANUTENZIONE
02/10/2023 10:06:10	02/10/2023 10:30:19	rtx 4050	PIANO VIBRANTE	GUASTO ELETTRONICO/AUTOMAZIONE

25 rows

1

Indietro

Figura 15

4 CRITICITA' DA RISOLVERE

Ignition offre diverse opzioni per l'inserimento di comandi di programmazione. È possibile creare un "gateway event", ossia uno script che si esegue ogni volta che si apre il progetto, valido quindi in ogni pagina del progetto. In alternativa, è possibile creare uno script che si esegue in relazione a un preciso comando di input, come ad esempio la pressione di un bottone o lo scroll della rotella del mouse.

Per affrontare le sfide emerse nella progettazione di questa specifica Interfaccia Uomo-Macchina (HMI), ho dovuto utilizzare entrambi questi metodi, partendo dalla comprensione del linguaggio di programmazione Python, che è il linguaggio interpretato da Ignition negli script di testo. Non avendo seguito alcun corso specifico su Python durante la triennale, ho dovuto acquisire competenze in modo autonomo. Fortunatamente, Ignition fornisce diversi aiuti per completare il codice, oltre a offrire modi per utilizzare funzioni senza scrivere codice, sfruttando trasformazioni e collegamenti tra variabili.

La prima modifica richiesta è stata l'inserimento di un terzo stato della macchina della linea, denominato "caricamento", che indica un periodo in cui la macchina non sta lavorando ma non è completamente ferma. Per implementare questa funzionalità, ho creato una nuova voce nella query che inserisce i dati nel server, denominata appunto "caricamento", la quale viene salvata se alcuni tag hanno un determinato valore, come si evidenzia nella figura 16.

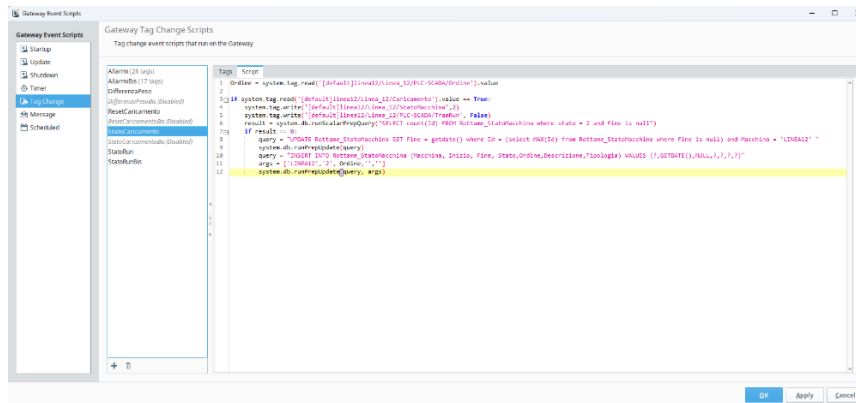


Figura 16

Questo script è parte dei gateway event scripts e si attiva ogni volta che il tag specificato durante la sua creazione cambia valore. Nel mio caso, il tag in questione è il tag di memoria chiamato "caricamento". Questo tag non è direttamente collegato a nessun valore della macchina ma è stato creato manualmente. Ogni volta che l'operatore sulla linea preme il pulsante di caricamento sull'HMI, questo tag viene impostato a 1, risultando quindi positivo.

Quando lo script viene eseguito, legge il valore del tag "caricamento". Se questo è vero (1), il tag dello stato della macchina viene modificato e impostato a 2, associato al caricamento del materiale. Inoltre, lo script spegne la tramoggia, mettendo il valore del tag relativo al suo movimento su false.

Come già descritto in precedenza, per l'Industria 4.0 è di fondamentale importanza il Data Collecting. Le ultime righe di codice in questo script controllano il query per verificare se ci sono stati "caricamento" con fine NULL, ovvero se ci sono caricamenti aperti ma non chiusi per qualche errore. Se non ne trova nessuno, crea una nuova riga nella tabella inserendo la data e l'ora di inizio del caricamento, insieme a tutti i dati dell'ordine che sta caricando.

I query sono funzioni di collegamento utilizzate per leggere o scrivere dati nelle tabelle dei server, selezionando specifici fattori di interesse, come ad esempio la linea di cui si vuole conoscere lo stato o solo gli orari di inizio lavorazione di tutte le linee, senza avere un eccesso di informazioni.

Per terminare lo stato di caricamento, si utilizza un secondo evento, molto più semplice del precedente (vedi figura 17); questo evento viene attivato sempre dal tag "TramInRun", che indica il cambio di stato della tramoggia iniziale.

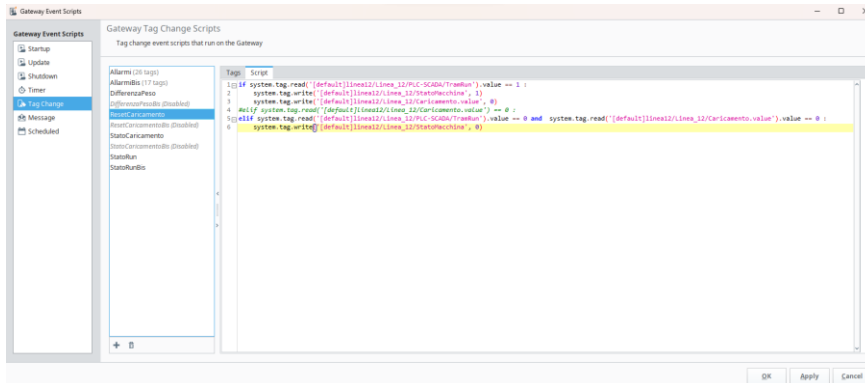


Figura 17

Se la tramoggia risulta in lavorazione, il tag dello stato macchina viene impostato su 1, mentre quello relativo al caricamento diventa 0, indicando la fine dello stato di caricamento. Se la tramoggia risulta spenta e lo stato di caricamento è già 0, ciò significa che la linea è spenta. In questo caso, il valore del tag stato macchina viene impostato su 0.

5 SCRIPT PER IL TRACCIAMENTO DEL PESO CARICATO

Per l'azienda è di fondamentale importanza tenere traccia della quantità di materiale lavorato in una specifica linea, sia per stimare i costi di produzione, sia per valutare se la linea può essere migliorata per aumentarne l'efficienza. A questo scopo, è stato implementato un gateway event, che monitora il peso misurato dalla tramoggia in ingresso alla linea e lo registra come dato nel server.

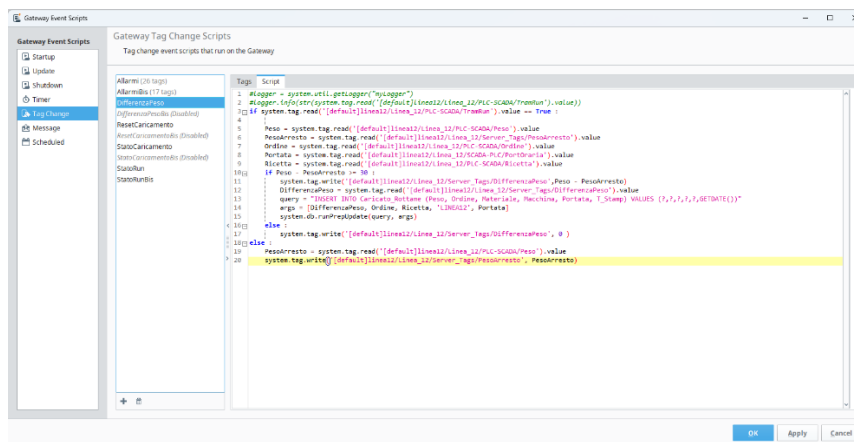


Figura 18

L'attivazione di questo script è gestita dal tag "TramInRun", il quale indica lo stato della tramoggia all'inizio della linea: 1 se la tramoggia sta lavorando, 0 se è ferma. Lo script viene eseguito ad ogni cambio di valore del tag.

Se il valore del tag è 1, ciò indica che la tramoggia è appena partita nella lavorazione, il sistema legge il valore del peso misurato in quel preciso istante e lo salva nella variabile "Peso"; inoltre, viene letto il peso di arresto precedentemente salvato, come sarà illustrato in seguito. Se la differenza di peso tra le due variabili "Peso" e "PesoArresto" è maggiore di un valore che rappresenta l'errore di misura o comunque un aumento di peso considerevole (in questo caso, considerato 30 chilogrammi), il valore della differenza viene salvato nella variabile "DifferenzaPeso"; quest'ultima sarà poi aggiunta al

query e, di conseguenza, al server. Se la differenza di peso non supera il valore di scarto impostato, il query viene completato inserendo uno zero come differenza peso, indicando che in quel periodo di fermo della linea non è stato caricato altro materiale.

Nel caso in cui la tramoggia sia ferma, lo script leggerà il valore del peso misurato sulla tramoggia e lo salverà nella variabile "PesoArresto".

6 SCRIPT PER IL FERMO ORARIO

Questo script è estremamente semplice e breve, ma rappresenta il primo problema di implementazione che ho affrontato durante il tirocinio. Era necessario incorporare nell'HMI la possibilità di programmare lo stop della linea in maniera automatica e il più chiaramente possibile.

Allo stesso tempo, tale mi ha messo in difficoltà perché ho dovuto utilizzare molte funzioni di Python, un linguaggio di cui non ho avuto istruzione nei programmi di studio.

A differenza dei precedenti script, questo non è un tag change event, ovvero che si attiva in corrispondenza del cambio di una variabile indicata, ma è un timer event, ovvero è sempre attivo all'interno dell'HMI

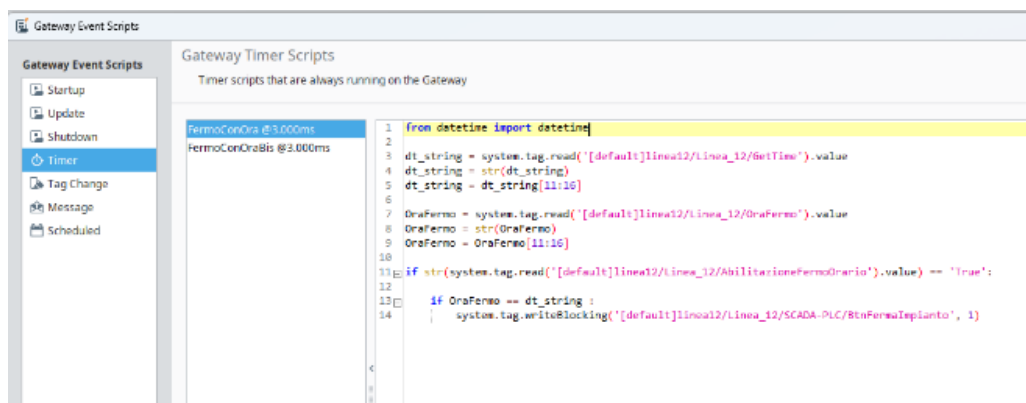


Figura 19

Il problema principale è stato trovare un modo per confrontare l'orario impostato nel menu dell'HMI con l'orario in real-time, poiché è possibile uguagliare e quindi confrontare solo due stringhe o due numeri, mentre il tempo viene scritto in formato data. È stato quindi necessario allungare il codice con passaggi più complessi:

- Assumere l'orario in real-time fornito da un query che salva l'orario nel tag "GetTime" e salvarlo in una stringa, quindi con data e ora in formato esteso. Ciò fornisce più informazioni di quelle strettamente necessarie.

- Eliminare le parti in eccesso dalla stringa utilizzando la funzione substring, che consente di creare una stringa più breve a partire dalla stringa iniziale.
- Ripetere i primi due passaggi anche con l'orario preso dalla selezione dell'orario di stop inserito nell'HMI.
- Verificare se il tag "AbilitazioneFermoOrario" risulta true; in tal caso, significa che l'operatore ha attivato il bottone che vediamo nella schermata Home per il fermo.
- Se il bottone è su true, verificare che le due substring create coincidano: in caso affermativo, modificare il valore del tag "BtnFermoImpianto" per spegnere l'impianto e tutte le sue parti. Se il tag "AbilitazioneFermoOrario" è su false, lo script non ha alcun effetto sul programma e sulla linea.

7 SCRIPT SALVATAGGIO STATI RUN E FERMI SUL SERVER

Questo script è stato implementato per rispondere alla necessità di raccogliere tutti i dati possibili sui cambi di stato della linea, liberando contemporaneamente la memoria locale dell'HMI, ed inserendo i dati direttamente nel server. Per raggiungere questo obiettivo, è stata necessaria l'implementazione e l'utilizzo di un query che ho creato ed inserito nel progetto.

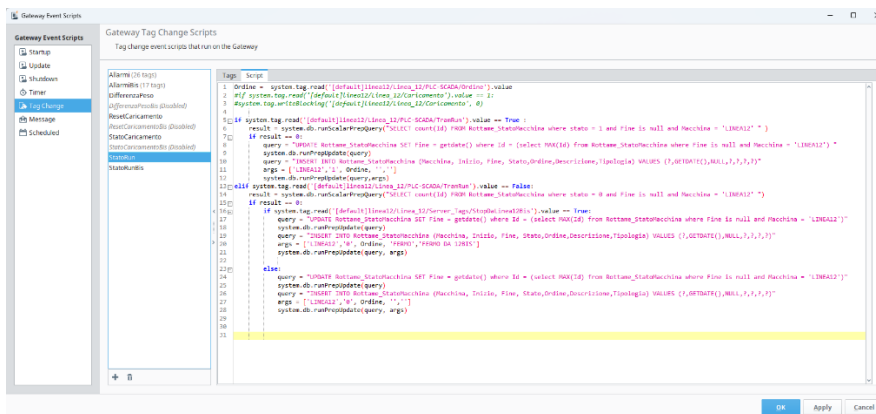


Figura 20

Anche questo script è attivato dal cambio di valore del tag “TramInRun”, infatti, come prima cosa, verifica se esso si trova in true o in false. Nel caso in cui lo trovi true, quindi la tramoglia è in movimento, cerchiamo nel query l’ultimo dato inserito che ha come stato della macchina true e che, come orario di fine, è nullo, cioè controlliamo se esiste già uno stato aperto che non è stato ancora chiuso; se questa ricerca ci restituisce zero come numero di risultati, creiamo lo stato inserendo il tempo di inizio, il tipo di ordine e la linea a cui ci stiamo riferendo, visto che il server non è esclusivo per la sola linea 12, ma per tutte le linee del rottame.

Nel caso in cui il valore della tramoglia sia in false, eseguiamo la stessa ricerca di uno stato lasciato aperto, e se non otteniamo risultati, in questo frangente lo script cambia: si legge il valore del tag “StopDaLinea12Bis” che ci dice se la linea 12Bis, ovvero quella direttamente successiva alla 12, ha

ordinato di fermarsi. Nel caso in cui ciò risulti vero, inseriamo nel server il fermo della linea, giustificandolo come voluto dalla linea 12 bis, e che quindi non apparirà nella tabella della pagina “Stato Macchine”, visto che è già stato causalizzato durante la sua creazione; altrimenti, inseriamo il fermo senza la motivazione, lasciando così il compito all’operatore di scegliere la motivazione di tale fermo direttamente sull’HMI.

Bibliografia

- "Industria 4.0. Uomini e macchine nella fabbrica digitale" - [Annalisa Magone, Tatiana Mazali]
- "La fabbrica connessa. La manifattura italiana (attra)verso Industria 4.0" - [Luca Beltrametti, Nino Guarnacci, Nicola Intini, Corrado La Forgia]
- "User Manual for Ignition" by Inductive Automation
- Sito web: <https://inductiveuniversity.com/>

Spiegazione: "Inductive University è il sito di formazione online di Inductive Automation, progettato per offrire una vasta gamma di corsi per imparare a utilizzare il framework Ignition. Questo sito fornisce risorse educative e guide pratiche per coloro che desiderano acquisire competenze nell'implementazione e nell'uso di Ignition, una piattaforma software per l'automazione industriale e la gestione dei dati."

