

Il compito prevede le seguenti attività:

1. Configurazione dell'hardware:

- Inserire i componenti hardware secondo il numero di articolo e la versione firmware indicati nella tabella.
- Configurare le proprietà di ciascun dispositivo: nome, indirizzo IP, maschera di sottorete e byte delle schede di input/output.

Sigla	N° Articolo	Versione Firmware	Telaio	Posto Conn.	Nome	Indirizzo IP	Maschera di sottorete
CPU 1215C DC/DC/Rly	6ES7 215-1HG40-0XB0	V4.5	0	1	PLC_1	192.168.10.1	255.255.255.0
SM 1221 DI16 x 24VDC	6ES7 221-1BH32-0XB0	V2.0	0	2	DI 16x24VDC_1		
SM 1222 DQ16 x relay	6ES7 222-1HH32-0XB0	V2.0	0	3	DQ 16xRelay_1		

2. Configurazione delle variabili PLC:

- Creare due tabelle delle variabili: una per gli ingressi e una per le uscite.
- Inserire le variabili di ingresso e uscita seguendo la tabella, rispettando gli indirizzi e il tipo di dato indicati.

Variabili PLC			
Nome	Tipo di dato	Indirizzo	Commento
CPU 1215C DC/DC/Rly - Uscite			
Out_SegLum_Normale	Bool	%Q0.0	Comando per azionare la segnalazione luminosa verde che indica lo stato normale dell'impianto. (Colonnina delle segnalazioni).
Out_SegLum_Anormale	Bool	%Q0.1	Comando per azionare la segnalazione luminosa arancione che indica una condizione anormale dell'impianto. (Colonnina delle segnalazioni).
Out_SegLum_Pericolosa	Bool	%Q0.2	Comando per azionare la segnalazione luminosa rossa che indica una condizione pericolosa dell'impianto. (Colonnina delle segnalazioni).
Out_Buzzer	Bool	%Q0.3	Comando per azionare il buzzer. (Colonnina delle segnalazioni).
CPU 1215C DC/DC/Rly - Ingressi			
In_Pul_Start_Ciclo	Bool	%I0.0	Pulsante di start ciclo (contatto N.O.).
In_Pul_Stop_Ciclo	Bool	%I0.1	Pulsante di stop ciclo (contatto N.C.).
In_Pul_Stop_Fine_Ciclo	Bool	%I0.2	Pulsante di stop a fine ciclo (contatto N.C.).
In_Pul_Pausa_Ciclo	Bool	%I0.3	Pulsante di pausa ciclo (contatto N.O.).
In_Sel_Aut_Man	Bool	%I0.4	Selettore per la modalità di funzionamento " automatica(1)/manuale(0)".
SM 1222 DQ16 x relè - Uscite			
Out_Nastro_Avanti	Bool	%Q4.0	Comando che attiva il contattore per gestire la marcia oraria del motore asincrono trifase (M1), consentendo il movimento avanti del nastro trasportatore.
SM 1221 DI16 x DC24V - Ingressi			
In_Pul_Nastro_Avanti	Bool	%I4.0	Pulsante per attivare la marcia avanti del nastro trasportatore (contatto N.O.).
In_Sens_Pezzo_Inizio_Nastro	Bool	%I4.2	Sensore per rilevare il pezzo all'inizio del nastro trasportatore (contatto N.O.).
In_Sens_Pezzo_Fine_Nastro	Bool	%I4.3	Sensore per rilevare il pezzo alla fine del nastro trasportatore (contatto N.O.).
In_FB_Nastro_Avanti	Bool	%I4.4	Segnala lo stato del contattore che comanda il motore per la marcia avanti del nastro trasportatore (contatto N.O.).
In_FB_Salvatore_Nastro	Bool	%I4.5	Segnala lo stato del salvatore che protegge il motore del nastro trasportatore. (contatto N.C.).

Esercizio 2: Gestione degli input/output del PLC tramite blocchi dati e funzioni

Descrizione: In questo esercizio, dovrai creare blocchi dati (DB) e funzioni (FC) per gestire gli ingressi e le uscite fisiche del PLC. L'obiettivo è separare la logica di lettura/scrittura dei segnali fisici dalla logica applicativa, rendendo il codice più modulare, gestibile e facile da aggiornare. Questa separazione riduce il rischio di errori e permette di modificare la configurazione hardware senza dover intervenire sulla logica applicativa, migliorando l'efficienza e la manutenibilità del sistema.

Obiettivo: Separare la logica di lettura/scrittura dei segnali fisici dalla logica applicativa per facilitare aggiornamenti, ridurre il rischio di errori e migliorare la modularità del codice.

Il compito prevede le seguenti attività:

1. Creare il blocco dati "DB0001_Configurazione_In":

- Inserire le variabili corrispondenti agli ingressi fisici del sistema, assicurando che il PLC interpreti correttamente i segnali dal campo.

2. Creare il blocco dati "DB0002_Configurazione_Out":

- Inserire le variabili corrispondenti alle uscite fisiche, fungendo da collegamento tra le decisioni logiche del PLC e le azioni fisiche sul campo.

3. Creare e Implementare la funzione "FC0001_Configurazione_In":

- Leggere gli ingressi fisici e assegnare i valori logici alle variabili corrispondenti nel blocco dati "DB0001_Configurazione_In" per mantenere il sistema allineato con l'ambiente operativo.

4. Creare e Implementare la funzione "FC0002_Configurazione_Out":

- Leggere le variabili del blocco dati "DB0002_Configurazione_Out" e assegnare i valori logici alle uscite fisiche del PLC per garantire un funzionamento preciso e controllato del sistema.

5. Compilazione, verifica e caricamento delle funzioni nel blocco main (Program Cycle - OB1) del PLC:

- Verificare e compilare i blocchi dati e le funzioni, assicurandosi che non ci siano errori di sintassi o logica.
- Caricare le funzioni nel blocco principale del PLC per garantire il corretto funzionamento durante l'esecuzione.

Nota: Le variabili nei blocchi dati "DB0001_Configurazione_In" e "DB0002_Configurazione_Out" devono corrispondere ai nomi degli ingressi e delle uscite fisiche, ma senza il prefisso "In_" o "Out_". Ad esempio, "In_Pulsante_XXX" diventa "Pulsante_XXX".

Esercizio 3: Integrazione e sincronizzazione dei comandi tra PLC e HMI

Descrizione: Questo esercizio si focalizza sull'integrazione e la sincronizzazione dei comandi tra l'HMI e il PLC per garantire un controllo preciso e affidabile del sistema. L'HMI funge da interfaccia per l'operatore, inviando comandi che devono essere gestiti insieme a quelli della pulsantiera fisica. È necessario sviluppare una logica di controllo nel PLC che assicuri la gestione sincronizzata dei comandi, prevenendo conflitti e garantendo una risposta coerente del sistema, indipendentemente dalla fonte del comando.

Obiettivo: Implementare una logica nel PLC che gestisca in modo sincronizzato i comandi dall'HMI e dalla pulsantiera, garantendo un controllo fluido e coerente del sistema.

Il compito prevede le seguenti attività:

1. Creare il blocco dati "DB0003_hmi":

- Inserire le variabili specifiche per i comandi dall'HMI, come indicato nella tabella, per consentire la gestione del sistema attraverso l'interfaccia grafica.

Nome	Tipo di dato	Commento
Pul_Start_Ciclo	Bool	Pulsante per attivare dall'HMI l'avvio del ciclo.
Pul_Stop_Ciclo	Bool	Pulsante per attivare dall'HMI l'arresto del ciclo.
Pul_Stop_Fine_Ciclo	Bool	Pulsante per attivare dall'HMI l'arresto del ciclo al termine delle operazioni.
Pul_Pausa_Ciclo	Bool	Pulsante per attivare dall'HMI la pausa del ciclo.
Pul_Nastro_Avanti	Bool	Pulsante per attivare dall'HMI la marcia avanti del nastro trasportatore

2. Creare il blocco dati "DB0004_Comandi_Generali":

- Inserire le variabili che riflettono l'unione dei segnali di controllo provenienti sia dall'HMI che dalla pulsantiera, , come indicato nella tabella, garantendo un controllo centralizzato e affidabile.

Nome	Tipo di dato	Commento
Pul_Start_Ciclo	Bool	Comando generale del pulsante di start ciclo.
Pul_Stop_Ciclo	Bool	Comando generale del pulsante di stop ciclo.
Pul_Stop_Fine_Ciclo	Bool	Comando generale del pulsante di stop a fine ciclo.
Pul_Pausa_Ciclo	Bool	Comando generale del pulsante di pausa ciclo.
Sel_Aut_Man	Bool	Comando generale del selettore per la modalità di funzionamento " automatica(1)/manuale(0)".
Pul_Nastro_Avanti	Bool	Comando generale del Pulsante per attivare la marcia avanti del nastro trasportatore

2. Creare e Implementare la funzione "FC0004_Comandi_Generali"

- Leggere gli input dal blocco dati "DB0001_Configurazione_In" (input dalla pulsantiera) e dal blocco "DB0003_hmi" (input dall'HMI).
- Assegnare i valori logici alle variabili nel blocco "DB0004_Comandi_Generali" per gestire i comandi in modo sincronizzato e senza conflitti.
- Consolidare tutti i comandi, inclusi quelli non duplicati, nel blocco dati "DB0004_Comandi_Generali" per evitare inconsistenze.

3. Compilazione, verifica e caricamento delle funzioni nel blocco main (Program Cycle - OB1) del PLC:

- Verificare e compilare i blocchi dati e le funzioni, assicurandosi che non ci siano errori di sintassi o logica.
- Caricare le funzioni nel blocco principale del PLC per garantire il corretto funzionamento durante l'esecuzione.

Esercizio 4: Gestione del processo in modalità manuale

Descrizione: Questo esercizio si concentra sul controllo manuale del sistema di automazione gestito direttamente dal PLC. La modalità manuale permette all'operatore di interagire con i dispositivi e le fasi operative del sistema attraverso comandi predefiniti. È necessario sviluppare una logica che consenta di attivare e disattivare queste funzioni in modo sicuro ed efficace, garantendo l'integrità operativa del sistema.

Obiettivo: Sviluppare la logica nel PLC per gestire le operazioni manuali, assicurando che il sistema risponda in modo sicuro e coerente ai comandi dell'operatore.

Il compito prevede le seguenti attività:

1. Creare il blocco dati "DB0005_Manuali":

- Inserire le variabili specifiche per il controllo manuale del sistema, come indicato nella tabella.

Nome	Tipo di dato	Commento
Comando_Condizioni_Ok_per_Manuale	Bool	Comando che verifica che tutte le condizioni siano rispettate per il funzionamento in modalità manuale.
Comando_Nastro_Avanti	Bool	Comando in modalità manuale per attivare la marcia avanti del nastro trasportatore.

2. Creare e Implementare la funzione "FC0005_Manuali" per la gestione del ciclo manuale:

- Leggere lo stato logico di ogni pulsante o selettore dal blocco dati "DB0004_Comandi_Generali".
- Assegnare i valori logici determinati alle variabili corrispondenti nel blocco dati "DB0005_Manuali".
- **Nota:** La variabile "DB0005_Manuali".Comando_Condizioni_Ok_per_Manuale è attiva solo se la macchina si trova in modalità manuale, ovvero quando "DB0004_Comandi_Generali".Sel_Aut_Man è negato e assume il valore 0.
- **Nota:** Le variabili Comando_xxxx possono essere attivate solo se è soddisfatta la condizione "Condizioni_Ok_per_Manuale".

3. Compilazione, verifica e caricamento delle funzioni nel blocco main (Program Cycle - OB1) del PLC:

- Verificare e compilare i blocchi dati e le funzioni, assicurandosi che non ci siano errori di sintassi o logica.
- Caricare le funzioni nel blocco principale del PLC per garantire il corretto funzionamento durante l'esecuzione.

Esercizio 5: Configurazione dell'hardware e sviluppo delle dashboard nell'HMI per il controllo manuale del processo.

Descrizione: In questo esercizio, configurerai l'hardware dell'HMI e svilupperai le dashboard per il controllo manuale dei processi industriali. Dovrai impostare i componenti fisici e creare interfacce utente intuitive con pulsanti e indicatori visivi per facilitare l'interazione degli operatori con il sistema. L'obiettivo è migliorare la gestione e il monitoraggio del processo.

Obiettivo: Configurare l'HMI e sviluppare dashboard per il controllo manuale, creando pagine di controllo che ottimizzino l'interazione e la gestione del sistema.

Il compito prevede le seguenti attività:

1. Configurazione dell'hardware:

- Inserire i componenti hardware seguendo la tabella sottostante.
- Configurare ogni dispositivo impostando il nome, l'indirizzo IP e la maschera di sottorete, come indicato.

Sigla	N° Articolo	Versione Firmware	Telaio	Posto Conn.	Nome	Indirizzo IP	Maschera di sottorete
KTP700 Basic PN	6AV2 123-2GB03-0AX0	V17.0.0.0			HMI_1	192.168.10.12	255.255.255.0

2. Creare le seguenti pagine nell'HMI:

- Pagina di home: La pagina principale di navigazione.
- Pagina di controllo manuale: Consente di eseguire operazioni in modalità manuale sui componenti del sistema.
- Pagina allarmi: Per monitorare e gestire gli allarmi di sistema.
- Pagina di impostazioni: Per personalizzare e configurare i parametri di sistema.
- Pagina di sistema: Fornisce informazioni sullo stato generale e sulla diagnostica. *(Nota: Le pagine di sistema predefinite devono essere abilitate e non create ex novo.)*

3. Gestione e supervisione tramite HMI:

- Sviluppare una dashboard nella pagina "Manuale" **ispirata** all'immagine fornita, mantenendo una disposizione intuitiva delle funzioni principali.

3.1 Pulsanti di controllo: Implementare i pulsanti necessari per il controllo manuale dei componenti del sistema.

- **Nastro trasportatore 1:** Pulsante per attivare dell'HMI la marcia avanti del nastro trasportatore.

3.2 Lampade di segnalazione, retroazione visiva e logica di controllo:

- Implementare una retroazione visiva chiara per gli operatori, monitorando lo stato degli attuatori e dei sensori tramite la dashboard HMI. Includere logiche di animazione per migliorare la comprensione visiva.

Tipi di Animazioni Applicabili:

- **Conformazione - Colorazione delle Icone:** Cambia il colore o l'aspetto delle icone per riflettere lo stato.
 - **Attivo (stato logico "1"):** L'icona diventa verde.
 - **Inattivo (stato logico "0"):** L'icona diventa grigia.
- **Visibilità - Visibilità delle Immagini:** Mostra o nasconde icone in base allo stato del componente. Ad esempio, l'icona di un motore appare solo quando il motore è in funzione.
- **4. Associazione variabili HMI-PLC:**
 - Associare ciascun pulsante, attuatore e lampada dell'HMI con la variabile corrispondente nei blocchi dati (DB) del PLC, in modo da riflettere lo stato del componente controllato.

Esercizio 6: Gestione del processo in modalità automatica.

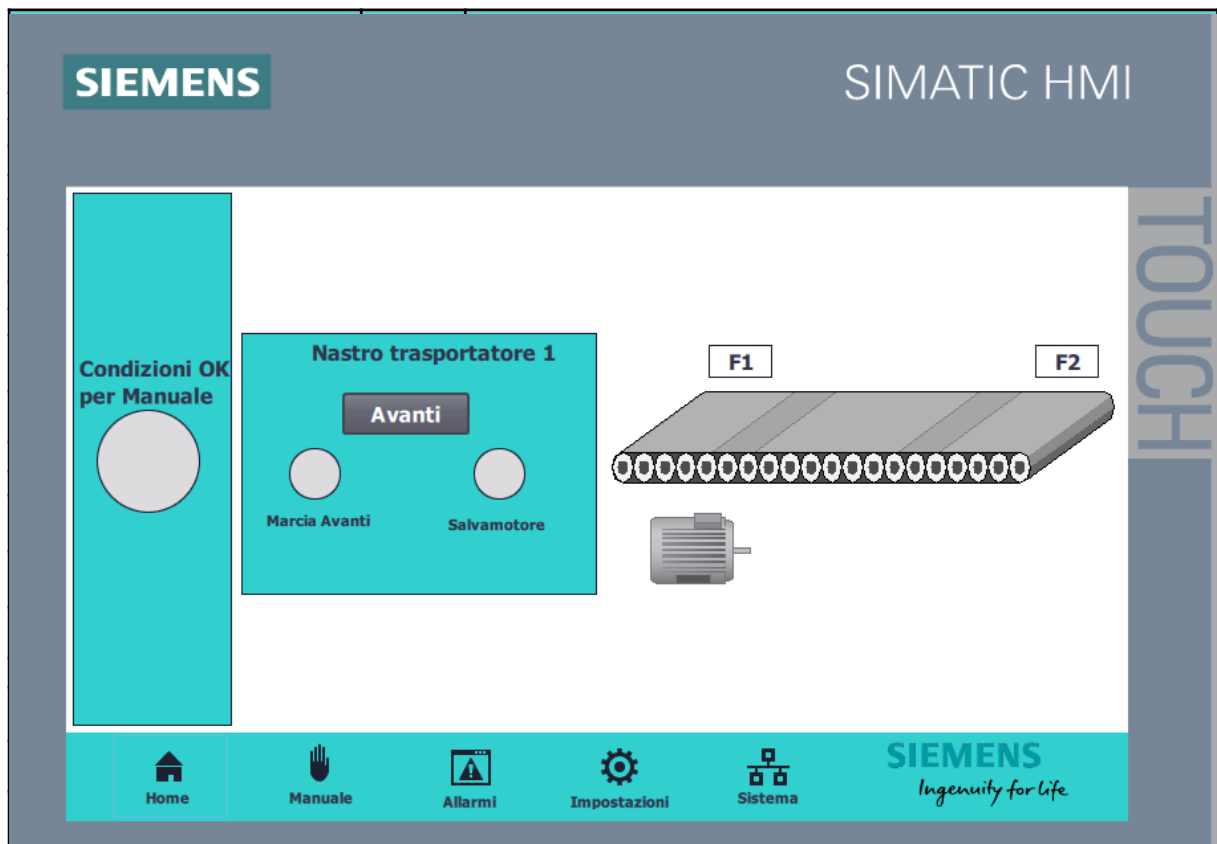
Descrizione: In questo esercizio, implementerai un blocco dati dedicato e una funzione specifica per gestire i comandi necessari per ogni fase del ciclo automatico. Il ciclo automatico permette al sistema di operare autonomamente, seguendo una sequenza predefinita di operazioni fino al completamento del processo. La logica di controllo attiverà i dispositivi e gli attuatori in base alle condizioni operative, garantendo un'esecuzione fluida e coordinata.

Obiettivo: Progettare e realizzare un ciclo operativo automatizzato che gestisca tutte le fasi del processo in modo autonomo, creando un blocco dati e una funzione di controllo per attivare e coordinare i dispositivi secondo una sequenza predefinita, senza intervento manuale.

Il compito prevede le seguenti attività:

1. Creare il blocco dati "DB0006_Ciclo_Automatico":

- Inserire le variabili necessarie per il controllo automatico del sistema, come indicato nella tabella.



2. Creare e Implementare la funzione "FC0006_Ciclo_Automatico"

- Gestire la logica operativa del ciclo automatico, coordinando le fasi del ciclo e l'attivazione degli attuatori in modo sicuro e preciso.
- Garantire la sincronizzazione di tutte le operazioni, incluse le fasi del ciclo e i dispositivi coinvolti, per ottimizzare l'efficienza del sistema.

2.1 Analizzare e implementare i passaggi del processo:

- Fare riferimento alla tabella allegata per i passaggi e le condizioni operative specifiche del ciclo automatico.
- Implementare il programma in base ai passaggi definiti, assicurandosi che ogni fase e comando siano correttamente eseguiti.

2.2 Implementare le funzionalità del ciclo automatico:

- Gestione delle fasi con Set e Reset:

- Comando Set: Attivare le fasi del ciclo quando le condizioni specifiche sono soddisfatte.
- Comando Reset: Disattivare le fasi del ciclo quando le condizioni di reset sono rispettate.
- Utilizzare le condizioni di attivazione/disattivazione definite nella tabella per gestire le operazioni in modo sicuro e coerente.
- Usare le variabili dal blocco dati "DB0006_Automatico" per gestire l'esecuzione del ciclo automatico, coordinando le fasi e gli attuatori coinvolti.
- Integrare segnali provenienti da pulsanti e selettori utilizzando le variabili del blocco dati "DB0004_Comandi_Generali".
- Monitorare i sensori e i feedback utilizzando le variabili dal blocco dati "DB0001_Configurazione_In" per mantenere la logica operativa allineata con lo stato attuale del sistema.

3. Passaggi del processo:

- Fare riferimento alla tabella che riporta i passaggi del ciclo automatico per implementare correttamente le fasi operative.

Variable: "Condizioni_OK_per_Partenza_Ciclo"
Condizione di attivazione/disattivazione: "Sel_Aut_Man" AND NOT "Comando_Ciclo_In_Corso"
Variable: "Condizioni_OK_per_Ripartenza_Ciclo"
Condizione di attivazione/disattivazione: "Sel_Aut_Man" AND "Comando_Ciclo_In_Corso"
Variable: "Richiesta_Partenza_Ciclo"
Condizione di attivazione/disattivazione: "Pul_Start_Ciclo" AND ("Condizioni_OK_per_Partenza_Ciclo" OR "Condizioni_OK_per_Ripartenza_Ciclo")
Variable: "Comando_Partenza_Ciclo"
Condizione di attivazione/disattivazione del Timer: "Timer_Comando_Partenza_Ciclo".TON (IN:= "Richiesta_Partenza_Ciclo", PT:5S).
Condizione di attivazione/disattivazione: "Timer_Comando_Partenza_Ciclo".Q.
Variable: "Comando_Ciclo_In_Corso".
Condizione di Attivazione (SET): "Comando_Partenza_Ciclo".
Condizione di Disattivazione (RESET): "Comando_Cancella_Ciclo".
Variable: Comando_Reset_Fasi.
Condizione di attivazione/disattivazione: NOT "Comando_Ciclo_In_Corso" OR "Fase_4".
Variable: Fase_0.
Descrizione: Inizio del ciclo.
Condizione di Attivazione (SET): "Comando_Ciclo_In_Corso" AND NOT "Comando_Ciclo_In_Pausa".
Condizione di Disattivazione (RESET): "Comando_Reset_Fasi".
Variable: Fase_1.
Descrizione: Il pezzo viene caricato sul nastro trasportatore e viene movimentato verso la destinazione prefissata.
Condizione di Attivazione (SET): "Fase_0" AND NOT "Comando_Ciclo_In_Pausa" AND NOT "Sens_Pezzo_Inizio_Nastro" AND NOT "Sens_Pezzo_Fine_Nastro" AND "Richiesta di scarico da stazione precedente".
Condizione di Disattivazione (RESET): "Comando_Reset_Fasi".
Variable: Fase_2.
Descrizione: Il pezzo si trova alla destinazione prefissata, in attesa di essere scaricato alla stazione successiva
Condizione di Attivazione (SET): "Fase_1" AND NOT "Comando_Ciclo_In_Pausa" AND "Sens_Pezzo_Fine_Nastro".
Condizione di Disattivazione (RESET): "Comando_Reset_Fasi".
Variable: Fase_3.
Descrizione: Il sistema attende che il pezzo venga scaricato dal Nastro trasportatore.
Condizione di Attivazione (SET): "Fase_2" AND NOT "Comando_Ciclo_In_Pausa" AND "Richiesta di carico da stazione successiva".
Condizione di Disattivazione (RESET): "Comando_Reset_Fasi".
Variable: Fase_4.
Descrizione: Termine del processo e reset di tutte le fasi.
Condizione di Attivazione (SET): "Fase_3" AND NOT "Comando_Ciclo_In_Pausa" AND NOT "Sens_Pezzo_Fine_Nastro".
Condizione di Disattivazione (RESET): "Comando_Reset_Fasi".

4. Attivazione degli attuatori per fase:

- Fare riferimento alla tabella per definire l'attivazione degli attuatori in base alla fase del ciclo e alle condizioni operative.

Variabile: Comando _Nastro_Avanti.
Condizione di attivazione/disattivazione: /(("Fase_1" AND NOT "Fase_2") OR ("Fase_3" AND NOT "Fase_4")) AND NOT "Comando Ciclo In Pausa".

5. Compilazione, verifica e caricamento delle funzioni nel blocco main (Program Cycle - OB1) del PLC:

- Verificare e compilare i blocchi dati e le funzioni, assicurandosi che non ci siano errori di sintassi o logica.
- Caricare le funzioni nel blocco principale del PLC per garantire il corretto funzionamento durante l'esecuzione.

Esercizio 7: Gestione avanzata di arresto, pausa e cancellazione del ciclo nel processo automatico

Descrizione: In questo esercizio, implementerai funzioni di controllo avanzate nel ciclo operativo automatizzato per migliorare la flessibilità e la sicurezza del sistema. Saranno gestite situazioni critiche come l'arresto sicuro del ciclo, la cancellazione delle operazioni in corso e la messa in pausa in risposta a condizioni specifiche.

Queste funzionalità saranno integrate nella funzione "FC0006_Ciclo_Automatico" per aggiungere livelli di controllo, ottimizzando la gestione operativa e garantendo una risposta adeguata a situazioni inattese.

Obiettivo: Migliorare la gestione del processo automatizzato implementando funzioni di controllo avanzate per rispondere in modo efficace a condizioni critiche e ottimizzare la sicurezza operativa.

Il compito prevede le seguenti attività:

1. Implementare la logica di controllo per la gestione avanzata di arresto, pausa e cancellazione del ciclo nella funzione "FC0006_Ciclo_Automatico":

- **Integrare la logica di controllo avanzata come dettagliato nella tabella allegata:**
 - **Arresto alla fine del Ciclo:** Garantire che il ciclo si fermi in modo sicuro, completando tutte le operazioni in corso ed evitando danni o malfunzionamenti.
 - **Pausa del Ciclo:** Permettere di mettere in pausa il ciclo in modo controllato, con la possibilità di riprenderlo esattamente dal punto di interruzione.
 - **Cancellazione del Ciclo:** Consentire la cancellazione di tutte le operazioni in corso, riportando il sistema a uno stato iniziale definito.

2. Implementare le funzionalità del ciclo automatico:

- **Gestione delle variabili con Set e Reset:**
 - Comando Set: Attivare le variabile per la gestione avanzata quando le condizioni specifiche sono soddisfatte.
 - Comando Reset: Disattivare le variabile per la gestione avanzata quando le condizioni di reset sono rispettate.
- Utilizzare le condizioni di attivazione/disattivazione definite nella tabella per gestire le operazioni in modo sicuro e coerente.
- Utilizzare le variabili del blocco dati "DB0006_Automatico" per gestire le condizioni operative specifiche del ciclo automatico, assicurando una corretta esecuzione delle condizioni avanzate di arresto, pausa e cancellazione.
- Integrare segnali provenienti da pulsanti e selettori utilizzando le variabili del blocco dati "DB0004_Comandi_Generali"
- Monitorare i sensori e i feedback utilizzando le variabili dal blocco dati "DB0001_Configurazione_In" per mantenere la logica operativa allineata con lo stato attuale del sistema.

3. Passaggi del processo:

- Fare riferimento alla tabella che definisce le condizioni operative per gestire correttamente le fasi di arresto, pausa e cancellazione del ciclo.

Variabile: Comando_Arresto_a_Fine_Ciclo.
Descrizione: Arresta il ciclo in modo sicuro una volta completato il processo attuale.
Condizione di attivazione (Set): NOT "Pul_Stop_Fine_Ciclo".
Condizione di disattivazione (Reset): NOT "Comando_Ciclo_In_Corso"
Variabile: Comando_Cancella_Ciclo
Descrizione: utilizzata per interrompere e cancellare un ciclo di lavoro in corso, garantendo un arresto controllato dell'impianto e prevenendo movimenti inaspettati degli attuatori, per la sicurezza dell'operatore e dell'impianto.
Condizione di attivazione/disattivazione del Timer: "Timer_Comando_Cancella_Ciclo".TON (IN: NOT "Pul_Stop_Ciclo", PT: 5s).
Condizione di attivazione/disattivazione: ("Fase_4" AND "Comando_Arresto_a_fine_Ciclo") OR "Timer_Comando_Cancella_Ciclo".Q
Variabile: Comando_Ciclo_In_Pausa
Descrizione: Utilizzata per mettere il ciclo in pausa temporanea, consentendo di gestire interruzioni o verifiche. Il sistema sospende l'operazione in corso, mantenendo lo stato attuale del processo.
Condizione di attivazione (Set): "Pul_Pausa_Ciclo" OR (NOT "Condizioni_OK_per_Partenza_Ciclo" AND NOT "Condizioni_OK_per_Ripartenza_Ciclo").
Condizione di disattivazione(Reset): "Comando_Partenza_Ciclo" OR NOT "Comando_Ciclo_In_Corso"

4. Compilazione, verifica dei blocchi dati e funzioni del PLC:

- Verificare e compilare i blocchi dati e le funzioni, assicurandosi che non ci siano errori di sintassi o logica.

Esercizio 8: Gestione delle segnalazioni operative nel processo automatico

Descrizione: Questo esercizio si concentra sulla gestione delle segnalazioni nel ciclo operativo automatizzato utilizzando la funzione "FC0006_Ciclo_Automatico". Le segnalazioni forniscono feedback in tempo reale sulle condizioni operative del sistema, consentendo agli operatori di intervenire rapidamente in caso di anomalie. Verranno implementate logiche per distinguere le condizioni operative normali, anomale e pericolose, e un feedback sonoro tramite buzzer segnalerà l'inizio del ciclo, migliorando la chiarezza delle segnalazioni.

Obiettivo: Garantire una comunicazione chiara e tempestiva delle condizioni operative del sistema attraverso una gestione efficace delle segnalazioni.

I compito prevede le seguenti attività:

1. Implementare la logica di controllo delle segnalazioni operative nella funzione "FC0006_Ciclo_Automatico":

Integrare la logica di controllo delle segnalazioni come dettagliato nella tabella allegata:

- **Stato Normale:** Indica il corretto funzionamento del sistema.
- **Stato Anomalo:** Segnala l'attivazione della modalità manuale.
- **Stato Pericoloso:** Indica una condizione critica che richiede attenzione immediata.
- **Buzzer:** Segnala l'inizio del ciclo e si disattiva automaticamente.

2. Implementare le funzionalità del ciclo automatico:

- **Gestione delle variabili con Set e Reset:**
 - Comando Set: Attivare le variabile per la gestione avanzata quando le condizioni specifiche sono soddisfatte.
 - Comando Reset: Disattivare le variabile per la gestione avanzata quando le condizioni di reset sono rispettate.
- Utilizzare le condizioni di attivazione/disattivazione definite nella tabella per gestire le operazioni in modo sicuro e coerente.
- Utilizzare le variabili del blocco dati "DB0006_Automatico" per gestire le condizioni operative specifiche delle segnalazioni, assicurando che le condizioni di stato normale, anomalo e pericoloso siano correttamente identificate e comunicate.
- Integrare segnali provenienti da pulsanti e selettori utilizzando le variabili del blocco dati "DB0004_Comandi_Generali"

- Monitorare i sensori e i feedback utilizzando le variabili dal blocco dati "DB0001_Configurazione_In" per mantenere la logica operativa allineata con lo stato attuale del sistema.

3. Passaggi del processo:

- Fare riferimento alla tabella che definisce le condizioni operative delle segnalazioni per implementare correttamente gli stati normali, anomali e pericolosi, garantendo una comunicazione efficace e tempestiva delle condizioni operative.

Variabile: Comando_SegLum_Normale
Descrizione: Indica che l'impianto è in condizioni normali e tutte le funzionalità operano correttamente.
Condizione di attivazione/disattivazione: " Sel_Aut_Man " AND "Comando_Ciclo_In_Corso" AND ((NOT "Comando_Ciclo_In_Pausa") OR ("Comando_Ciclo_In_Pausa" AND "Clock_5Hz"))
Variabile: Comando_SegLum_Anormale
Descrizione: Indica che il sistema è passato alla modalità manuale, segnalando un'anomalia rispetto al normale funzionamento automatizzato.
Condizione di attivazione/disattivazione: NOT " Sel_Aut_Man "
Variabile: Comando_SegLum_Pericolosa
Descrizione: Indica che il sistema è in uno stato critico o pericoloso, richiedendo un intervento immediato per garantire la sicurezza.
Condizione di attivazione/disattivazione: NOT "In_FB_Salvatore_Nastro" AND "Clock_0.5Hz"
Variabile: Comando_Buzzer
Descrizione: Attiva il buzzer all'avvio del ciclo di lavoro per segnalare l'inizio delle operazioni. Si disattiva automaticamente dopo un tempo prestabilito tramite un timer di tacitazione.
Condizione di attivazione (SET): "Comando_Partenza_ciclo".
Condizione di attivazione Timer: "Timer_tacitazione_buzzer".TON(IN: "Comando_Segnalazione_Buzzer", PT 5S)
Condizione di disattivazione (RESET): "Timer_tacitazione_buzzer".Q.

4. Compilazione, verifica dei blocchi dati e funzioni del PLC:

- Verificare e compilare i blocchi dati e le funzioni, assicurandosi che non ci siano errori di sintassi o logica.

Esercizio 9: Gestione integrata delle utenze per il controllo delle modalità manuale e automatica nei processi di automazione

Descrizione: Questo esercizio si concentra sulla gestione integrata delle utenze in un sistema di automazione industriale. La funzione "Gestione_Utenze" sarà sviluppata per leggere e interpretare simultaneamente i comandi provenienti dalle modalità automatica e manuale, instradando i risultati verso le uscite fisiche del sistema. Questo approccio garantisce che le operazioni vengano eseguite in modo coerente e sicuro, evitando conflitti e assicurando che ogni comando sia eseguito secondo le specifiche del processo.

Obiettivo: Sviluppare la funzione "Gestione_Utenze" per un controllo ottimizzato e integrato del sistema, migliorando l'efficienza e la sicurezza operativa.

Il compito prevede le seguenti attività:

1. Creare la tua funzione "FC0007_Gestione_Utenze":

- La funzione dovrà leggere simultaneamente le variabili di comando degli attuatori dal blocco dati "DB0006_Automatico" (modalità automatica) e dal blocco "DB0005_Manuali" (modalità manuale).
- Assegnare il risultato della logica di comando alle variabili corrispondenti nel blocco dati "DB0002_Configurazione_Out", che controlla le uscite fisiche del sistema.
- Assicurarsi che la logica di controllo gestisca in modo coerente le interazioni tra i comandi delle due modalità, prevenendo conflitti e garantendo che le uscite fisiche riflettano correttamente le operazioni richieste.

2. Compilazione, verifica e caricamento delle funzioni nel blocco main (Program Cycle - OB1) del PLC:

- Verificare e compilare i blocchi dati e le funzioni, assicurandosi che non ci siano errori di sintassi o logica.
- Caricare le funzioni nel blocco principale del PLC per garantire il corretto funzionamento durante l'esecuzione.

Esercizio 10: Utilizzo di PLCSIM per simulazione e test del PLC.

Descrizione: Questo esercizio introduce all'uso di PLCSIM, un software che emula un PLC virtuale, permettendo di testare i programmi senza hardware fisico. I partecipanti configureranno PLCSIM per simulare un PLC, testare i programmi e gestire variabili in tempo reale.

Obiettivo: Verificare il funzionamento dei programmi PLC in un ambiente virtuale con PLCSIM prima del trasferimento su hardware fisico, sviluppando competenze nella simulazione e nel test dei programmi per garantire software privo di errori.

Di seguito, una descrizione passo-passo del processo che i partecipanti seguiranno:

1. Avvio del simulatore PLCSIM:

- Apri il software PLCSIM, che fornirà un ambiente virtuale per emulare il comportamento di un PLC reale. Questo simulatore sarà utilizzato per tutte le operazioni di test.

2. Reset del PLC virtuale:

- Esegui un reset del PLC virtuale tramite gli accessi online di TIA Portal. Questo passaggio garantisce che il simulatore inizi da una configurazione pulita, eliminando residui di test precedenti e assicurando un ambiente stabile.

3. Caricamento del progetto nel simulatore:

- Carica nel simulatore PLCSIM il progetto PLC precedentemente sviluppato. Questa operazione permette di testare le funzionalità programmate senza rischi per l'hardware fisico.

4. Collegamento online:

- Stabilire una connessione online con il PLC simulato per monitorare e interagire direttamente con il programma caricato. Questa connessione consente di osservare e modificare il comportamento del programma durante la simulazione.

5. Creazione di una tabella di controllo:

- **TIA Portal:** Crea una tabella di controllo che includa gli ingressi definiti nel blocco dati "DB0001_Configurazione_In" e le uscite fisiche. Questo permette di monitorare gli ingressi e le uscite in tempo reale.
- **PLCSIM:** Carica gli ingressi fisici dalla tabella delle variabili e le variabili del blocco dati "DB0002_Configurazione_Out" per forzare gli ingressi e le uscite durante la simulazione. Questo setup consente di vedere in tempo reale i cambiamenti degli stati nella tabella di controllo del PLC grazie ai blocchi funzione (FC) che gestiscono le letture e le scritture delle variabili.

Nota: Per effettuare la battitura degli I/O, è necessario eliminare temporaneamente dal progetto i blocchi dati "FC0002_Configurazione_Out" e "FC0001_Configurazione_In".