

Esercitazione 4

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		

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.).

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".

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.

