

Structured Text : C:\Users\emili\OneDrive - unibg.it\UN\Anno III\Automazione Industriale\GitHub\Laboratorio-di-Automazione-Industriale\ST\Logical\Cosberg\Cyclic.st

```
1
2 PROGRAM _CYCLIC
3
4 CASE STATO OF
5     STATO_SPENTO: // attendiamo l'avvio
6         // attivazione del nastro trasportatore
7         IF NASTRO_ON THEN
8             STATO := STATO_ACCESO;
9             NASTRO_ON:= FALSE;
10        END_IF;
11
12    STATO_ACCESO:
13
14        // In caso di emergenza spegnamo tutto senza pensarci due volte e cambiamo stato
15        IF STOP_EMERGENZA THEN
16            ALIMENTAZIONE := 0;
17            SALDATURA_ON := 0;
18            FORATURA_ON := 0;
19            AVVITATURA_ON := 0;
20            QUALITY_CHECK := 0;
21            // cambiamo stato
22            STATO := STATO_SPENTO;
23        ELSE // se non c'è emergenza lavoriamo normalmente
24            // se è presente un pezzo in staz_1
25            IF STAZ_1 AND C_MAN_IN < CILCI_MANUTENZIONE THEN
26                // chiamo la funzione per la stazione 1
27                GESTIONE_STAZ_1(TIPO:= TIPO_TRIGGER , TEMPO := T#0s , TRIGGER := PRESENZA_PEUZZO, AZIONE := ALIMENTAZIONE);
28                // se si verifica l'edge negativo disattivo la stazione e conto il pezzo
29                IF EDGENEG(ALIMENTAZIONE) THEN
30                    PRESENZA_PEUZZO := FALSE;
31                    STAZ_1 := FALSE;
32                    C_MAN_IN := C_MAN_IN + 1;
33                END_IF;
34            END_IF;
35            // se è presente un pezzo in staz_2
36            IF STAZ_2 THEN
37                // chiamo la funzione per la stazione 2
38                GESTIONE_STAZ_2(TIPO:= TIPO_TEMPO , TEMPO := T#5s , TRIGGER := FALSE , AZIONE := SALDATURA_ON);
39                // se si verifica l'edge negativo disattivo la stazione
40                STAZ_2 := NOT EDGENEG(SALDATURA_ON);
41            END_IF;
42            // se è presente un pezzo in staz_3
43            IF STAZ_3 THEN
44                // chiamo la funzione per la stazione 3
45                GESTIONE_STAZ_3(TIPO:= TIPO_TEMPO , TEMPO := T#3s , TRIGGER := FALSE, AZIONE := FORATURA_ON);
46                // se si verifica l'edge negativo disattivo la stazione
47                STAZ_3 := NOT EDGENEG(FORATURA_ON);
48            END_IF;
49            // se è presente un pezzo in staz_4
50            IF STAZ_4 THEN
51                // chiamo la funzione per la stazione 4
52                GESTIONE_STAZ_4(TIPO:= TIPO_TEMPO , TEMPO := T#4s , TRIGGER := FALSE, AZIONE := AVVITATURA_ON);
53                // se si verifica l'edge negativo disattivo la stazione
54                STAZ_4 := NOT EDGENEG(AVVITATURA_ON);
55            END_IF;
56            // se è presente un pezzo in staz_5
57            IF STAZ_5 THEN
58                // chiamo la funzione per la stazione 5
59                GESTIONE_STAZ_5(TIPO:= TIPO_TRIGGER , TEMPO := T#0s , TRIGGER := QUALITY_CHECK_DONE, AZIONE := QUALITY_CHECK);
60                // se si verifica l'edge negativo disattivo la stazione e incremento il contatore finale
61                IF EDGENEG(QUALITY_CHECK) THEN
62                    QUALITY_CHECK_DONE := FALSE;
63                    STAZ_5 := FALSE;
64                    C_MAN_OUT := C_MAN_OUT + 1;
65                END_IF;
66            END_IF;
67            // verifichiamo se l'ultimo pezzo è uscito dalla stazione
68            IF C_MAN_OUT >= CILCI_MANUTENZIONE THEN
69                STATO := STATO_MANUTENZIONE;
70            END_IF;
71        END_IF;
72
73    STATO_MANUTENZIONE:
74        MAI := TRUE;
75        IF MAIR THEN
76            C_MAN_IN := 0;
77            C_MAN_OUT := 0;
78            MAI := FALSE;
79            MAIR := FALSE;
80            STATO := STATO_ACCESO;
81        END_IF;
82    END_CASE;
83 END_PROGRAM
84
85
86
```