

Automatico EXT con IO esterno

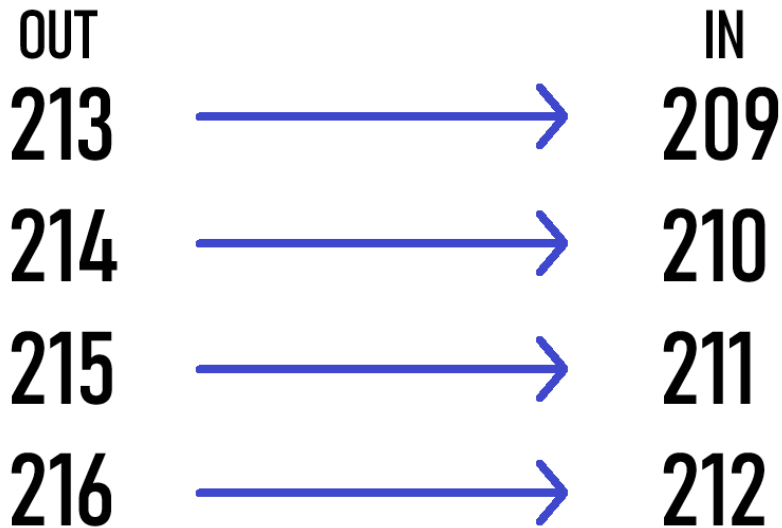
Modulo IO Esterno

Il sistema utilizza un modulo di I/O esterno Beckhoff per la gestione dei segnali del robot KUKA.



Logica dei ponticelli

I segnali di ingresso richiesti dal controllore KUKA vengono ottenuti tramite ponticellatura delle uscite selezionate sul modulo di I/O esterno. La numerazione degli I/O segue lo schema illustrato nell'immagine seguente.



Configurazione dei segnali di sistema

Come primo passo, è necessario configurare il file:

STEU → Mada → `$machine.dat`

Andando ad assegnare ai vari segnali, gli ingressi EtherCAT:

```
SIGNAL $EXT_START $IN[209] ;EXTERNER START
SIGNAL $MOVE_ENABLE $IN[210] ;FAHRFREIGABE GESAMT
SIGNAL $CONF_MESS $IN[211] ;EXTERNE QUITTUNG
SIGNAL $DRIVES_ON $IN[212] ;ANTRIEBE EIN
```

Configurazione delle uscite

Le uscite utilizzate per la gestione dei segnali EXT devono essere definite nel file `I0.dat`, come mostrato di seguito.

File `I0.dat`:

```
GLOBAL SIGNAL out_SYS_ExtStart          $OUT[213]
GLOBAL SIGNAL out_SYS_MoveEnable        $OUT[214]
```

GLOBAL SIGNAL out_SYS_ConfMess	\$OUT[215]
GLOBAL SIGNAL out_SYS_DrivesOn	\$OUT[216]

A questo punto, la configurazione degli ingressi e delle uscite risulta completata.

Gestione SPS e modalità EXT

All'interno del controllore KUKA è presente un programma SPS che gestisce, tramite interrupt, i segnali di *Start*, *Stop* e le altre funzioni di controllo in modalità EXT.

Questo programma è denominato **ExtManager.sub** e contiene il seguente codice:

```
&ACCESS RVP
&COMMENT USER specified PLC program
DEF ExtManager ( )
;FOLD DECLARATIONS
;FOLD USER DECL
; Please insert user defined declarations

;ENDFOLD (USER DECL)
;ENDFOLD (DECLARATIONS)
;FOLD INI
;FOLD USER INIT
; Please insert user defined initialization commands
    INTERRUPT DECL nStopBtn WHEN inStopRobot D0 EXT_STOP() ;Nel caso in cui l'ingresso del
pulsante di stop sia negato, basta mettere NOT
    INTERRUPT ON nStopBtn
    INTERRUPT DECL nStartBtn WHEN inStartRobot D0 EXT_START()
    INTERRUPT ON nStartBtn
    INTERRUPT DECL nIntExtSelect WHEN $EXT D0 EXT_SELECT()
    INTERRUPT ON nIntExtSelect
;ENDFOLD (USER INIT)
;ENDFOLD (INI)

LOOP
    IF NOT out_SYS_MoveEnable THEN
        out_SYS_MoveEnable = TRUE
    ENDIF
ENDLOOP
END
```

```

DEF EXT_INI ()
    out_SYS_ExtStart = FALSE
    out_SYS_ConfMess = FALSE
    out_SYS_Drives0n = FALSE
END

DEF EXT_STOP()
    DECL STATE_T STAT
    DECL MODUS_T MODE

    INTERRUPT OFF nStartBtn

    MODE=#SYNC

    IF $PRO_STATE1 <> #P_FREE THEN
        CWRITE($CMD,STAT,MODE,"STOP 1")
    ENDIF

    INTERRUPT ON nStartBtn
END

DEF EXT_START()
    INTERRUPT OFF nStartBtn
    IF (NOT $PRO_ACT)THEN
        EXT_INI()

        $TIMER[nTimeOutExt] = - tTimeOutExt
        $TIMER_STOP[nTimeOutExt] = FALSE
        nExtState = 0

        WHILE NOT $TIMER_FLAG[nTimeOutExt]
            SWITCH nExtState
                CASE 0
                    IF NOT $PERI_RDY THEN
                        out_SYS_Drives0n = TRUE
                        nExtState = 0
                    ELSE
                        out_SYS_Drives0n = FALSE
                        nExtState = 1

```

```
ENDIF
```

```
CASE 1
```

```
IF $STOPMESS THEN
```

```
    out_SYS_ConfMess = TRUE
```

```
    nExtState = 1
```

```
ELSE
```

```
    out_SYS_ConfMess = FALSE
```

```
    nExtState = 2
```

```
ENDIF
```

```
CASE 2
```

```
IF NOT $PRO_ACT THEN
```

```
    out_SYS_ExtStart = TRUE
```

```
    nExtState = 2
```

```
ELSE
```

```
    out_SYS_ExtStart = FALSE
```

```
    nExtState = 3
```

```
ENDIF
```

```
CASE 3
```

```
    $TIMER_STOP[nTimeOutExt] = TRUE
```

```
    nExtState = 0
```

```
EXIT
```

```
ENDSWITCH
```

```
ENDWHILE
```

```
EXT_INI ()
```

```
ENDIF
```

```
INTERRUPT ON nStartBtn
```

```
END
```

```
DEF EXT_SELECT ()
```

```
    DECL STATE_T STAT
```

```
    DECL MODUS_T MODE
```

```
INTERRUPT OFF nIntExtSelect
```

```
MODE=#SYNC
```

```
IF $PRO_STATE1 == #P_FREE THEN  
    CWRITE($CMD,STAT,MODE,"RUN /R1/MAIN()")  
    CWRITE($CMD,STAT,MODE,"STOP 1")  
    CWRITE($CMD,STAT,MODE,"RESET 1")  
ENDIF  
  
INTERRUPT ON nIntExtSelect  
END
```

In questo contesto, il timer utilizzato per la gestione dello Start fa riferimento alla variabile **nTimeOutExt**, definita nel file `Globals.dat`.

Revision #6

Created 2026-01-23 08:45:39 UTC by Tiziano

Updated 2026-01-23 14:00:49 UTC by Tiziano