Elementi

Di programmazione

Domotica

Sistema Evolus

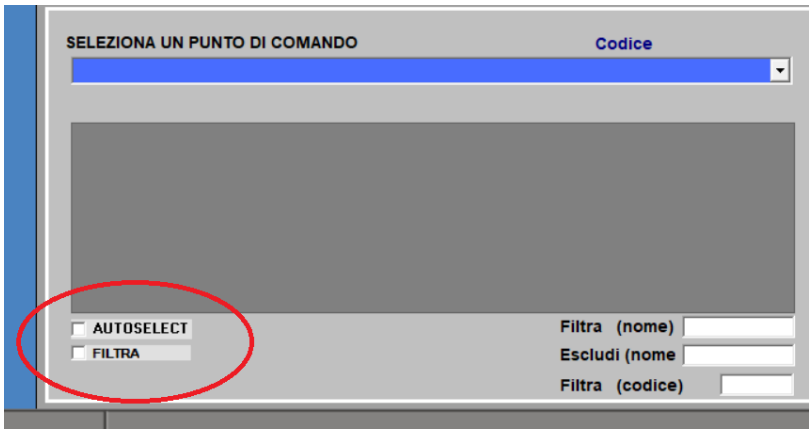
Guida all'uso di # E-bus

7

- **autoselect**
- **Il comando passa solo se**
- **Associazioni**

In attesa di prendere confidenza con la diagnostica fornita da E-bus, vediamo una funzione interessante che ci aiuta a capire a cosa fa capo un ingresso magari già chiuso nella scatoletta portafrutto senza dover smontare tutto

autoselect



Si tratta di una comoda funzione che serve per rintracciare velocemente un ingresso: ciccando su ***autoselect,*** il sistema attende il primo segnale sul bus proveniente da un ingresso: non appena intercettato se:

- non è stato ancora collegato apre window3 della centralina ove risiede l'ingresso, evidenziano in chiaro la label dell'ingresso corrispondente
- se già collegato, lo mostra sulla combo dei punti di comando.

L'opzione ***filtra*** serve per escludere gli ingressi di sicurezza o i sensori ti temperatura, che possono interferire sulla ricerca. Attenzione. Autoselect è in grado di trovare ingressi di tutte le centraline che sono collegate al bus, anche quelle che non fanno parte del progetto, ma ***filtra*** funziona solo con elementi che già fanno parte del progetto, perché E bus, per poterli bloccare, ne deve controllarne le caratteristiche ed i settaggi.

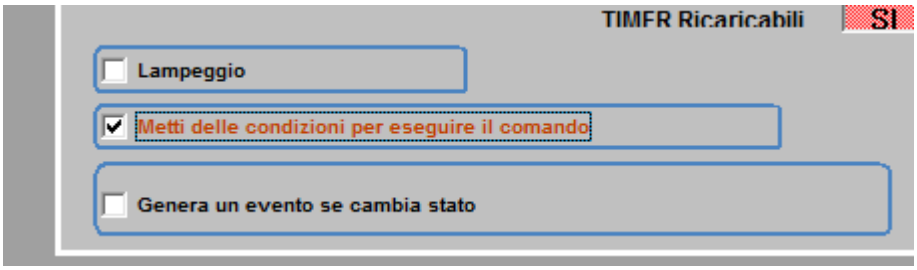
Facciamo una prova

Apriamo un progetto già fatto

Andiamo in ***collega*** e proviamo, dopo aver spuntato ***autoselect***, a premere un pulsante: vedrete che si aprirà window3 della centralina corrispondente, o se ha già un nome, l'ingresso vi comparirà nei punti di collegamento. (ricordatevi che, per poter collegare un ingresso, occorre dargli un nome, se no non lo troveremmo nella lista degli ingressi da collegare). È anche possibile quindi usare autoselect per rimediare a dimenticanze (collegamenti non "battezzati" o persi)

Condizioni per eseguire un comando

ora cominciamo ad affrontare argomenti un po' più difficili, non tanto per la loro complessità, che come vedremo non esiste, ma perché necessitano di ragionamenti che non siamo abituati a fare nel nostro lavoro (mentre, come vedremo, sono sempre presenti nella vita di tuti i giorni.



Mediante questa sezione possiamo fare in modo che un comando possa essere processato ***solo se*** siano in essere, all'interno della centralina, determinate condizioni (flag. Relè e relè virtuali). Questa funzione permette di sfruttare la ***logica a transiti***, ovvero la logica applicabile a situazioni transitorie, come, per esempio i comandi e gli eventi.

Immaginiamo che qualcuno vi dica di chiudere un interruttore del quadro: lo farete solo se non ci sono condizioni pericolose, ovvero un vostro collega che ci sta lavorando. Non appena cessa il pericolo, aspettate che vi ripetano la richiesta per eseguire l'azione; nella vita di tutti i giorni è quasi banale, mentre se provate a farne lo schema di funzionamento vedrete che non è poi così semplice (ricordate che ho detto che l'azione viene fatta solo dopo l'ordine, se non ci sono pericoli, non eseguita dopo l'ordine e non appena cessato il pericolo: quello è il finecorsa).

Sono concetti un poco difficili da assimilare in quanto nuovi, ma nelle lezioni avanzate avremo modo di capirli ed usarli con scioltezza; per ora limitiamoci ad un paio di semplici esempi.

Esempio 1

Abbiamo la luce di una scala che ha 2 diversi modi di accensione

Il primo, per la normale utenza, è temporizzato

Il secondo è passo passo per le manutenzioni

Ora immaginiamo che un utente, nonostante veda la luce accesa, riprema il pulsante per essere sicuro di avere tutto il tempo di accensione luce disponibile; se la luce fosse accesa per manutenzione, la sua manovra, avendo il sopravvento, farebbe spegnere la luce dopo il tempo stabilito, lasciando al buio i manutentori. In questo caso, per scongiurare questa situazione, il comando di accensione dell'utente potrebbe essere eseguito solo se la luce della scala è spenta: nel caso fosse già accesa non avrebbe nessun effetto.

Voi direte, ma cosi si toglie la possibilità di ripristinare il tempo della lampada già accesa: vero, ma era solo per capire il concetto. Se volessimo mettere in pratica l'esempio sopra dovremmo

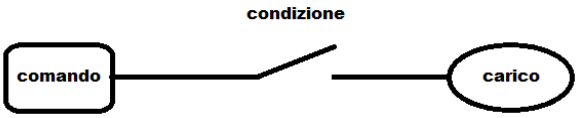
- Il comando di manutenzione passo-passo accende e spegne la luce della scala con un evento
- L'evento comanda un virtuale o flag in modalità segue lo stato

- Il comando dell'utente potrà essere processato solo se il relè virtuale comandato dalla luce scala è spento; il comando utente, solo accende la luce della scala per un tempo, non genera l'evento, per cui non affligge il virtuale.

Come possiamo vedere c'è una soluzione a tutto.

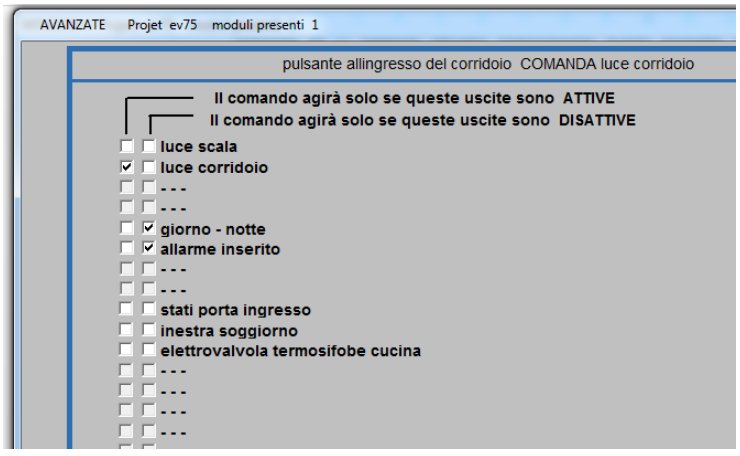
Ma prima di fare il secondo esempio……

Pensiamo però a 360°



Come in molte cose la potenza di un metodo ne determina anche il confine di utilizzo: facciamo un esempio: vogliamo che un campanello possa ricevere il comando di attivazione e disattivazione solo se la condizione ***non disturbare*** non sia presente: nella prima figura vediamo lo schema del circuito equivalente: il comando raggiunge il carico solo se la condizione lo permette.

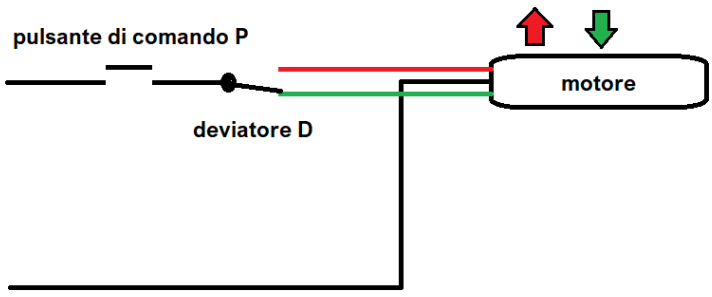
Ma se ***mentre*** suona il campanello si attivasse l'opzione ***non disturbare*** (condizione), questo non potrebbe più essere disattivato in quanto non potrebbe ricevere il comando di tacitazione; l'assenza della condizione non permetterebbe al relè di essere comandato, per cui il carico resterebbe nello stato dettato dal precedente comando "valido". Ovviamente ci sono parecchi metodi per evitare ciò, come per esempio usare le ***associazioni*** o fare in modo che il comando *non disturbare* taciti il campanello, usare timer etc., ma questo esempio è per farvi capire che è necessario tener conto di tutte le conseguenze possibili, quando si programma. Ricordiamoci che le macchine sono, seppur instancabili e veloci, "stupide", o per lo meno con l'intelligenza del programmatore, che **deve** perciò prevedere tutte le possibili interazioni.



Passiamo adesso alla realizzazione di un programmino che ci serva per capire questo concetto. Premetto che si tratta di un esempio che, pur semplice, può impegnarvi in quanto ancora una volta occorre ragionare in maniera più ampia, come purtroppo l'utilizzo degli impianti cablati non vi ha mai permesso. Una doverosa premessa: non c'è cosa che un impianto cablato non possa fare, solo che con la domotica risulta tutto più semplice, più modificabile e gestibile; per fare un esempio si potrebbe fare un computer ad acqua, dove rubinetti e valvole idrauliche modulano le sezioni di calcolo, ma un sistema in grado di fare una semplice addizione occuperebbe lo spazio di una corriera e costerebbe altrettanto, mentre utilizzando l'elettronica è decisamente più conveniente.

Problema da risolvere per il secondo esempio.

Con un solo pulsante dobbiamo comandare una tapparella: vediamo prima lo schema elettrico corrispondente.



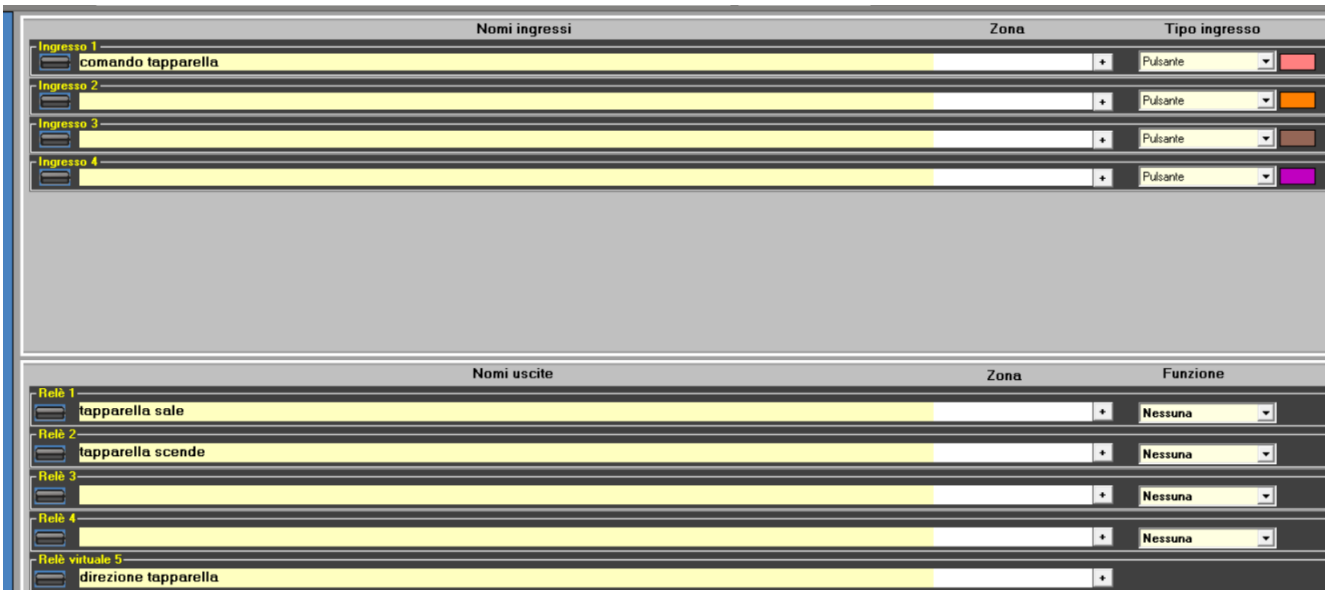
Il deviatore D collega, a seconda della sua posizione, il pulsante di comando P al filo che alimenta il motore in avvolgimento (apertura) svolgimento (chiusura della tapparella.

Cominciamo con il ragionamento

Nella figura a fianco, azionando il pulsante p il motore della tapparella viene alimentato attraverso il filo verde, e la tapparella si chiude.

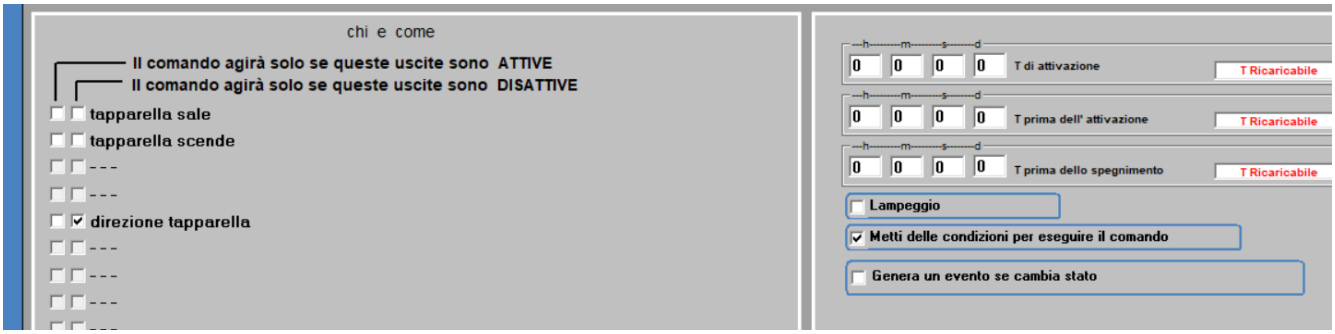
Cambiando la posizione del deviatore D, il pulsante p, se azionato, alimenterà il filo rosso facendo aprire la tapparella.

Quindi: il comando è dato da un unico pulsante e gestito da un deviatore. Facciamo un nuovo programma che chiameremo ***tapparella un comando***, aggiungiamo la solita 00B1 e riempiamo le label come da figura sotto. Come possiamo vedere abbiamo un solo pulsante di comando, due relè veri ed un relè virtuale chiamato ***direzione tapparella*** (il deviatore D). Naturalmente prima di salvare attiviamo gli ***interblocchi***, come abbiamo già visto.



Ora passiamo alla programmazione, andando su ***collega***

Abbiniamo il ***comando tapparella*** a ***tapparella sale***, in modo tipo ***campanello***, ma poniamo delle condizioni, ovvero che il comando possa essere processato ***solo se*** il ***relè direzione tapparella*** (D) sia spento.



Ora colleghiamo il pulsante di comando al relè ***tapparella scende***, ma a condizione che il virtuale ***direzione tapparella*** sia ***attivo***.

Abbiamo fatto in modo che la corrente virtuale del nostro impianto comandi la salita o la discesa a seconda di come si trovi il virtuale ***direzione tapparella***.

Resta ora da capire come possiamo comandare questo virtuale. Ci sono più modi, ma il ragionamento da seguire è lo stesso: quando deve cambiare posizione questo relè?

Dovrà farlo al ***rilascio*** del pulsante, in modo che al prossimo comando tutto sia pronto per il nuovo funzionamento. Vediamo come:

Basterà collegare il pulsante di comando al relè ***direzione tapparella*** in modalità ***passo-passo invertito*** che, come abbiamo visto, provoca il cambio stato di un'uscita al **rilascio** del pulsante di comando.

Potremmo però, invece che pilotare il deviatore da pulsante, far generare un evento ai relè sale e scende, ed usare:

- l'evento di scende per pilotare il deviatore in modo ***solo accende quando spegne*** e
- l'evento di sale in modo ***solo spegne***.

Questo metodo è da preferire perché permette il pilotaggio delle tapparelle sia uomo presente che passo-passo (ricordiamoci sempre i tempi e l'interblocco).

Come avete potuto constatare, gli esempi diventano sempre un po' più arzigogolati, ma non più difficili; se ci pensate abbiamo fatto una cosa semplice; stiamo semplicemente scoprendo un mondo che finora non pensavamo così facile e non lo abbiamo mai affrontato. Ricordiamoci inoltre che queste serie di incontri sono di un primo livello, dove tutto è nuovo per molti di voi, per cui possono sembrare più ostici qu quanto in realtà siano.

Abbiamo visto un comando logico fatto per la macchina a transiti, ovvero per i comandi che passano sul bus, ma esiste un altro modo, molto più potente, che è l'utilizzo delle **associazioni**, ovvero la macchina a stati di Evolus.

Le associazioni

anche qui ci sono davvero pochi concetti da imparare nonostante si tratti di uno strumento potentissimo che vi permetterà di risolvere in modo facile problemi anche complessi. Innanzitutto consideriamo che questa attività viene svolta dal firmware delle centraline come ultima cosa, per cui hanno priorità su tutto quello fatto prima dal programma.

Le associazioni sono la macchina a stati di Evolus. Rivediamo il concetto di macchina a stati e macchina a transiti, giusto per rinfrescarci la memoria.

Macchina a transiti

Io parlo con voi, voi ascoltate; una volta finita la frase che ho detto questa non esiste più, è scomparsa nel nulla ma tutti quelli che l'hanno sentita hanno l'informazione necessaria per fare qualcosa; avete captato una informazione in transito. Su questo principio si fondano le principali azioni nella vita comune, come l'invio di dati da un telecomando, sapere che è passata un'auto perché l'abbiamo vista, una telefonata etc.

Macchina a stati

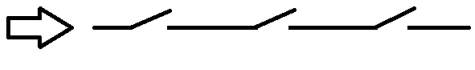
Io metto un oggetto sul tavolo e voi dovete dirmi quando sul tavolo ci sono tre oggetti dello stesso colore. Come potete immaginare, non mettendo tutti gli oggetti in contemporanea, devo sempre valutare la situazione di insieme e confrontare gli oggetti appena messi con quelli già presenti sul tavolo, con una visione di insieme. Questa è (molto semplicemente) una macchina a stati.

Come è facile intuire una può risolvere in modo facile problemi che con l'altra potrebbero essere complicati, per cui Evolus le gestisce entrambe in modo evoluto, anche facendo generare uno stato da un transito e viceversa.

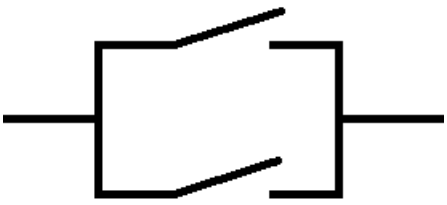
Prima di proseguire vediamo di comprendere i 2 concetti fondamentali della logica (si chiama logica booleana, da Boole, un matematico inglese che l'ha pensata nella prima metà dell'800). Le funzioni logiche fondamentali sono:

- AND
- OR

La funzione AND (e) può essere descritta come una serie di interruttori in serie. Chiamando condizione ***vera*** l'interruttore chiuso e ***falsa*** l'interruttore aperto, la corrente nell'esempio della figura fluirà solo quando ***tutte le condizioni sono vere***, ovvero tutti gli interruttori sono chiusi.



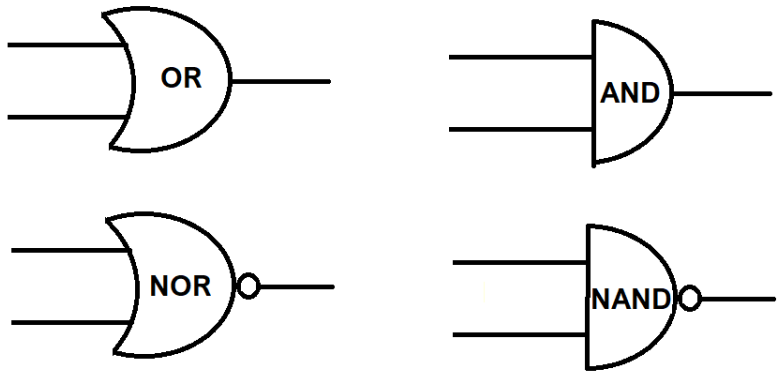
AND



OR

- La funzione OR (oppure) invece sarà vera quando ***almeno uno degli interruttori che la compongono è vero***. esistono altre varianti di porte logiche, ma soffermiamoci sulle AND, OR e le complementari NAND (not and) e NOR (not or) che si comportano in modo inverso, ovvero l'uscita è invertita.

Ovviamente questo è solo un esempio per memorizzare il concetto. Sotto la rappresentazione grafica delle porte logiche descritte; come possiamo vedere, la “negazione”, ovvero il funzionamento invertito, è rappresentata dal cerchiolino.



Facciamo qualche esempio di utilizzo delle associazioni

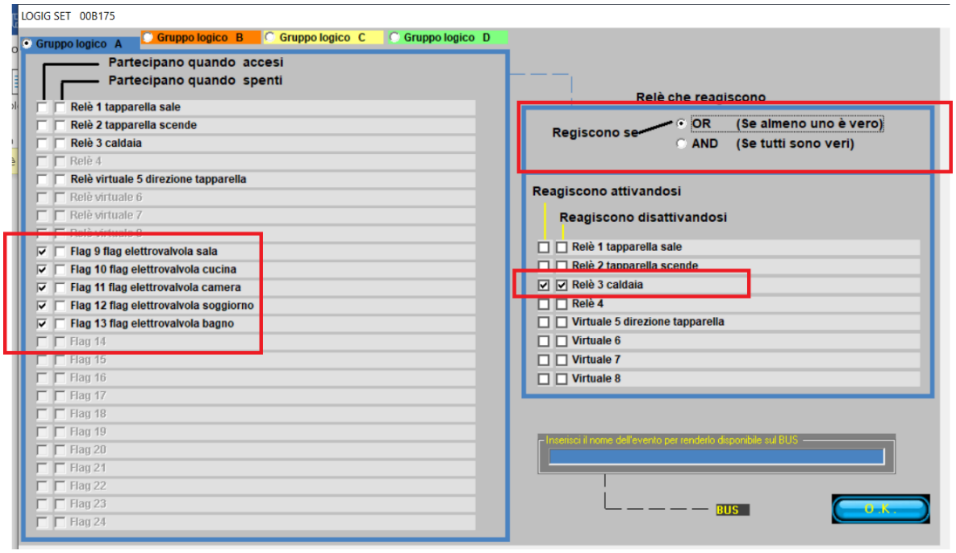
In un normale impianto di riscaldamento supponiamo di avere dei termostati che chiudano un contatto quando la temperatura è più bassa di quella impostata:

collegando i termostati agli ingressi, potremo comandare le elettrovalvole collegate a relè di uscita di un EV75 per esempio in modo campanello, ovvero a contatto chiuso il relè si attiva e viceversa (è solo per esempio)

immaginiamo ora di avere una decina di stanze e di dover attivare la caldaia se almeno una elettrovalvola è attiva:

basterà:

- far generare un evento alle elettrovalvole (non importa dove si trovino)
- con questo evento comandare, in modo segue ***lo stato***, dei flag su un'unica centralina (per esempio nella centralina dove comandiamo la caldaia
- fare un’associazione OR con la caldaia



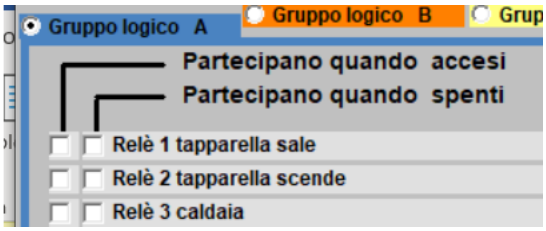
Come possiamo vedere nella figura a fianco, se almeno uno dei flag (funzione OR) che rispecchiano lo stato delle elettrovalvole è attivo (quindi almeno una elettrovalvola è attiva), la caldaia si attiva, altrimenti si disattiva. Ora che abbiamo qualche elemento per discutere approfondiamone il funzionamento.

Innanzitutto, ogni centralina ha 4 pagine di associazioni (o gruppi logici) identificati con colori (in alto a sinistra)

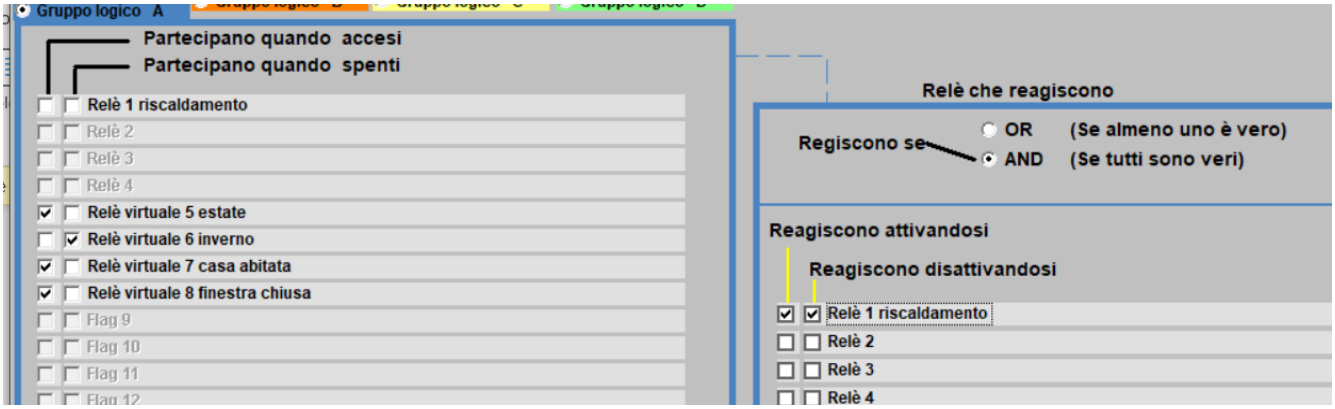
- Gruppo A – blu –
- Gruppo B – arancione –
- Gruppo C – giallo-
- Gruppo D – verde -

Occorre anche tener conto delle priorità: il gruppo D ha priorità sul C che a sua volta ho priorità sul B etc.

Questo, come vedremo più avanti, per risolvere eventuali conflitti. Se per esempio esiste una condizione sul gruppo A che accende una lampadina, se nel gruppo C la stessa lambada venisse spenta da un'altra condizione, la lampada sarebbe spenta in quanto il Gruppo C ha una priorità maggiore sul gruppo identificato con una lettera "più bassa".



A sinistra troviamo la lista dei relè veri, virtuali e Flag presenti nella centralina (le associazioni logiche possono essere fatte solo all'interno di una stessa centralina); ognuno di questi elementi può essere selezionato tramite due caselle: una per la condizione vera ed uno per la condizione falsa. Se nessuna delle caselle è selezionata, l'elemento è escluso dal processo.



Nell'esempio di figura sopra, il riscaldamento si attiva solo se:

- è estate AND
- non è inverno condizione NOT AND, ovvero NAND)
- la casa è abitata AND
- la finestra è chiusa AND

Quando queste condizioni sono tutte vere, nel modo stabilito dalla parte destra in alto (il deviatore disegnato che sceglie la funzione) i relè che devono reagire si attivano.

Ora chiariamo il comportamento di questa sezione:

1) Spuntando, nelle caselle dei carichi controllati (a destra) sia ON che OFF, il relè segue sempre la condizione, ovvero

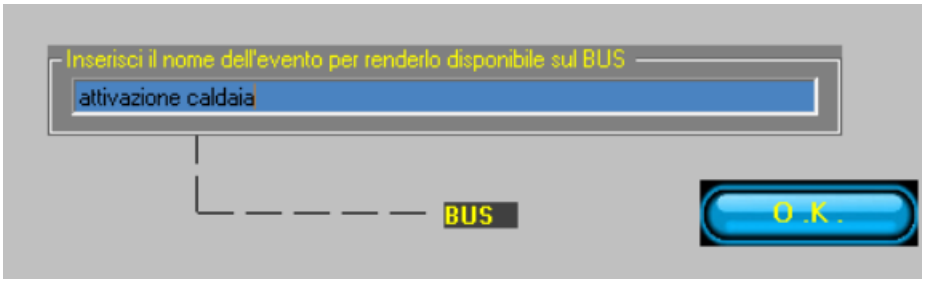
- Attivo se la condizione illustrata nella parte sinistra è vera
- Disattivo se falsa

2) Spuntando una sola casella sul relè che deve reagire, quando le condizioni sono vere sarà forzato in quella condizione, mentre se le condizioni sono false, potrà essere comandato normalmente dal programma fatto con E-bus.

Facciamo un esempio. (1) Io comando una luce normalmente in modo passo-passo da un punto di comando, ma, nelle associazioni, dico che se è non è vera la condizione notte (un virtuale o un flag che si attiva e disattiva con un crepuscolare), la luce viene forzata spenta. In questo caso di notte posso decidere se accendere o spegnere la luce con il comando programmato, ma quando la situazione cessa (non è più notte), la luce verrà forzata spenta dall'associazione e non sarà più comandabile in alcun modo (fino al ripristino della condizione "notte"); è come se la condizione logica, se presente, "schiacciasse"il relè impedendone l'attivazione, mentre, se non presente lo lasciasse libero di fare le funzioni per cui è stato programmato.

Da stato a transito

Il risultato di una pagina delle associazioni può anche essere trasformato in un comando sul bus; possiamo cioè trasformare il risultato della macchina a stati in una informazione della macchina a transiti. Se diamo un nome



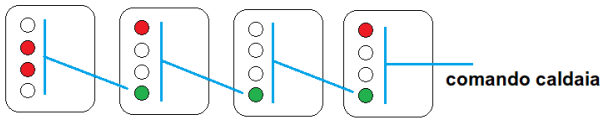
alla label in basso a destra, tutte le volte che la condizione passa da vera a falsa e viceversa, avremo un messaggio sul bus simile a quello generato da un pulsante o da un interruttore (con il nome assegnato alla label); ON se passa da falsa a vera, mentre, quando la condizione non è più vera (passa da vera a falsa), un messaggio di OFF. In realtà avremo due tipi di messaggi, uno se è vera la condizione OR ed uno per la condizione AND

Nella figura sopra vediamo un esempio di messaggi generati da un'associazione. Il vantaggio che ci offre è che il comando sul bus potrà essere trattato come si vuole, con tempi, condizioni etc. come se fosse un normale comando. Per esempio, potrei accendere la caldaia solo quando le condizioni siano vere, e spegnerla 5 minuti dopo che siano false in modo stabile (modalità ***segue lo stato*** con ritardo allo spegnimento).

Possiamo anche utilizzare i messaggi generati dalle condizioni per comandare elementi da riutilizzare nelle associazioni; con questo metodo, i confini si ampliano: facciamo un altro esempio.

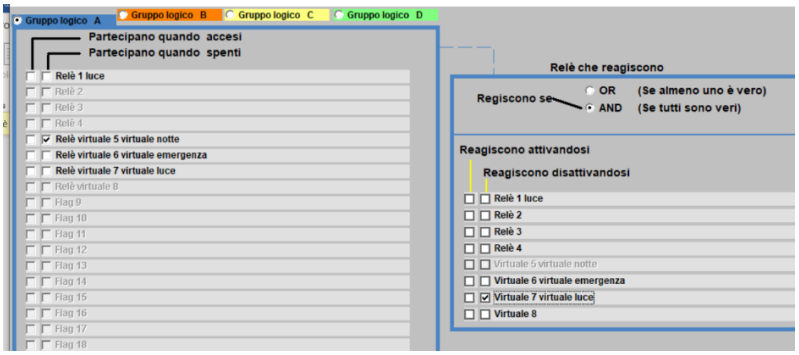
Nell' impianto di riscaldamento visto prima abbiamo 30 elettrovalvole sparse in, per esempio 22 centraline; abbiamo però solo 16 flag per ogni centralina, per cui la centralina che deve fare l'OR non avrebbe, in teoria, risorse sufficienti.

Nella figura sotto troviamo la soluzione al problema. La prima centralina ha 2 elettrovalvole accese; manda un messaggio (associazioni OR) alla centralina 2 che accende un flag in modalità segue lo stato, ovvero il flag sarà acceso solo se almeno una elettrovalvola della centralina precedente (quella che genera il messaggio) è attivo. A sua volta, la centralina 2 manda un messaggio di OR alla 3 che non ha nessun relè elettrovalvola attivo, ma ha comunque il flag di riporto (verde) attivo coinvolto nella condizione OR; facendo questi parte delle associazioni (almeno un elemento è vero), manda un messaggio alla centralina 4 e così via. Con questo sistema è chiaro che il numero di elementi coinvolti in un'associazione può virtualmente essere infinito; anche in questo caso è solo un esempio, vedremo che ci sono anche sistemi più snelli. Vediamo ora, per finire le nozioni basiche, come combinare AND e OR



Nell'esempio della luce di prima (esempio (1)), che può essere comandata solo di giorno, voglio aggiungere un comando di emergenza che la accenda comunque;

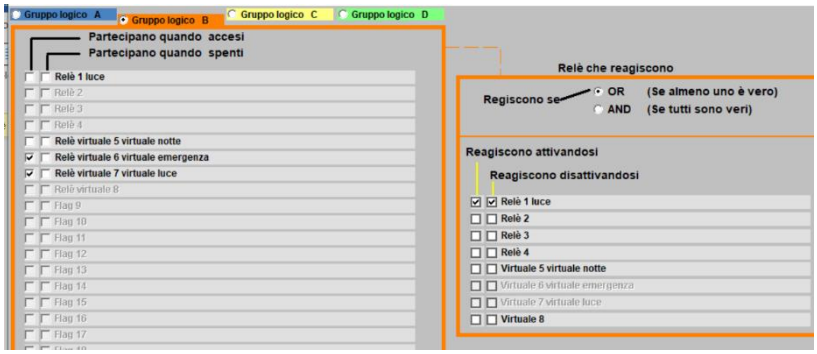
- il comando comanda un relè virtuale che chiamiamo virtuale luce
- il comando di emergenza comanda un relè virtuale che chiameremo virtuale emergenza
- il relè virtuale luce potrà essere forzato off dalla condizione ***non notte*** di: pagina BLU



nella pagina dopo (arancione) diciamo che:

- se è acceso il virtuale luce OR (oppure)
- è acceso il relè virtuale emergenza

la luce si deve accendere



Come è facile intuire si tratta di una funzione molto potente che, come ho detto, per adesso vi lascerà un po' spiazzati, ma, potendola usare negli esercizi risulterà più semplice di quanto possiate immaginare. È vero che per molti di voi la logica potrebbe risultare, di primo acchito, ostica, ma perché state

ancora pensando condizionati dagli impianti cablati; ora, con un semplice esempio vi dimostro che, nella vita di tutti i giorni, l'avete sempre usata.

Logica AND

- Condizioni: devo uscire, piove
- Azioni (risultato) prendo l'ombrello

Casi

- Devo uscire **e** piove – CONDIZIONE **VERA** PER PRENDERE L'OMBRELLO
- NON Devo uscire **e** NON piove – CONDIZIONE **FALSA** PER PRENDERE L'OMBRELLO
- NON Devo uscire **e** piove – CONDIZIONE **FALSA** PER PRENDERE L'OMBRELLO
- Devo uscire **e** NON piove – CONDIZIONE **FALSA** PER PRENDERE L'OMBRELLO

Logica OR

- Condizioni: andare a lavorare, passeggiata
- Azioni: uscire di casa

Casi OR

- Esco di casa sia che debba andare al lavoro o a fare una passeggiata

Come vedete è anche facile pensare a situazioni di tutti i giorni per fare combinazioni -AND e OR (esco solo se devo fare una passeggiata AND non piove Or andare al lavoro) (esco se voglio andare a fare una passeggiata se non piove, ma devo comunque uscire se devo andare al lavoro)

Quindi tranquilli. Sono forse cose nuove ma non difficili, solo viste in un altro contesto.

Per ora limitiamoci all'infarinatura, che già però vi permetterà di risolvere problemi che vedremo nelle prossime lezioni. Finito il ciclo base di incontri, (11 in tutto, dalla 0 alla 10) ci sarà un ciclo intermedio dedicato agli esercizi che ci aiuteranno a prendere confidenza con Evolus.

Per il prossimo incontro procuratevi 2 potenziometri da 22 KoHm A (lineari): chi lo volesse potrebbe aggiungere alla valigetta una centralina per dimmerare, cosa che vi consiglio vivamente in quando il mercato, con l'avvento dei led, sta andando in quella direzione.