



Manuale degli script xxter

Elementi base	2
Gestionescript	3
Editordi script	4
Comandi	6
Controllo dei componenti	6
Controllo di (altri) script	7
Utilizzo dei ritardi	8
Scenari, avvisi e pianificatori	8
Comandi	9
Valori	9
Variabili e calcolo	10
Variabili	10
Calcolo	11
Funzioni	12
Funzionimatematiche	12
Funzionitemporali	13
Funzionidi testo	14
Strutture di comando	15
IF e WHILE	15
INIZIO e FINE	17
Altro	18



Script xxter

Con gli script xxter è possibile creare i tuoi piccoli programmi all'interno di xxter. Questi script sono molto flessibili e possono essere utilizzati per aggiungere un'ampia varietà di funzionalità a un'installazione domotica. È possibile creare funzioni logiche, ritardare azioni, estendere i tuoi scenari con sequenze RGB e molto altro.

Gli script possono essere attivati utilizzando uno scenario, una pianificazione, un'azione (trigger) o un altro script.

Oltre a questo manuale, diversi esempi sono disponibili per il download sul nostro sito web per creare funzionalità specifiche con script xxter. Questi esempi potrebbero essere molto utili non solo per queste caratteristiche specifiche, ma anche per imparare come funzionano gli script xxter.

Elementi base

Gli script xxter sono programmi scritti che consistono in una o più righe. Ogni riga deve contenere un solo comando specifico. Non è possibile inserire più comandi su una riga. È possibile aggiungere commenti a uno script scrivendo un carattere "#". Questo può essere fatto su una riga separata o alla fine di un comando.

Un esempio di script xxter:

```
1 #Example script
2 #The following line sets a dimmer to 100%
3 SET BYTE(542/Example light) TO 100
4 #The following line disables a schedule
5 SET SCHEDULER(2/Alarmclock) TO 0
6 #The following line has an error
7 STOP [SCRIPT]
8
```

Gli script possono essere avviati e arrestati con molti mezzi, ad esempio con uno scenario o una pianificazione. Gli script possono essere script "unending" che eseguono determinate azioni a intervalli ripetuti o possono essere definiti come una sequenza di comandi che vengono eseguiti una sola volta, ogni volta che lo script viene avviato.



Gestione script

Gli script xxter possono essere creati e aggiornati utilizzando l'editor xxter. Questo editor può essere trovato nella configurazione del progetto di "My xxter". Selezionare il progetto per il quale si desidera gestire gli script e selezionare "Script" nella barra dei menu. Tutti gli script esistenti verranno mostrati qui e possono essere modificati ed eliminati.

Test ProjectProject has changed

Settings | Components | Cameras | Commands | Statistics/Chart data | Presence detection | Actions | **Scripts**

Filter ×

Enabled	Name		
Yes	Testscript 1		✎ ✕
No	Testscript 2	In scenario's and schedulers: Not available	✎ ✕

Add Script Add Group

Importante: Quando si aggiunge o si modifica uno script, questo deve essere caricato sul dispositivo xxter prima di poter essere utilizzato.

Ogni volta che è stata apportata una modifica, è possibile visualizzare il testo "Project has changed" nell'angolo in alto a destra. Quando si fa clic su questo e successivamente su "Push configuration to xxter device", il dispositivo scaricherà la nuova configurazione. Questa funzione può essere utilizzata solo quando xxter è configurato per l'uso all'esterno (vedere il capitolo 13 del manuale di installazione). In alternativa è possibile ricaricare il profilo tramite l'app (vedi capitolo 11 del manuale utente). Naturalmente è possibile anche ricaricare il profilo dal dispositivo xxter stesso, accedendo al dispositivo e facendo clic sul pulsante "Load configuration" nell'angolo in alto a sinistra del menu.

Quando si è connessi al dispositivo, è possibile visualizzare gli script caricati facendo clic sull'opzione "Script" nel menu. Qui uno script può anche essere avviato, riavviato o arrestato manualmente a scopo di test.

Scripts

To change actions and scripts, go to '[My xxter](#)'.

Active	Name		Status	
Yes	Testscript 1		Stopped	Start Restart Stop
No	Testscript 2	In scenario's and schedulers: Not available	Not active	Start Restart Stop



Editor di script

Quando si aggiunge o si modifica uno script, è possibile selezionare se deve essere abilitato e se lo script deve essere disponibile per l'utente finale, da utilizzare negli scenari o nello scheduler. È inoltre possibile selezionare se lo script deve essere avviato automaticamente, ogni volta che il dispositivo xxter viene (ri) avviato.

IMPORTANTE: Gli script disabilitati non possono essere eseguiti e non verranno triggerati quando inclusi in uno scenario, un'azione o uno scheduler. Questo può essere utile per le proposte di test e risoluzione dei problemi. Tuttavia, tieni presente che uno script può essere abilitato o disabilitato da un altro script!

Test ProjectProject has changed

Settings | Components | Cameras | Commands | Statistics/Chart data | Presence detection | Actions | **Scripts**

Script - Editor

Enabled: Yes
Name: Testscript 1
After boot: -
In scenario's and schedulers: Available

Help

COMMANDS
RESTART
START [script]
START AT BEGIN [script]
STOP
STOP [script]
WAIT [time]
WAIT [time] RANDOM [time]
CALL [scenario]
CALL [alert]
CALL [alert] WITH [value]
SET [script] TO [on/off]
SET [scheduler] TO [on/off]
SET [bit component] TO [value]
SET [byte component] TO [value]
SET [2byte component] TO [value]

1
2

Save and check | Save and close | Cancel

I numeri di riga visualizzati accanto allo script sono solo informativi e non vengono utilizzati negli script stessi. I comandi possono essere aggiunti digitandoli direttamente o utilizzando lo strumento di selezione dei comandi, situato nell'angolo in alto a destra. Assicurati di essere nella posizione giusta nell'editor di script quando aggiungi un comando con lo strumento comandi. Quando si aggiunge un comando con lo strumento comandi, i campi appropriati relativi alla configurazione xxter verranno aggiunti automaticamente come collegamenti. Questo permette di avere un modo rapido e semplice per aggiungere comandi corretti utilizzando l'editor.

1 SET SCHEDULER(1/Away) TO OFF
2
3



Dopo aver aggiunto uno o più comandi nell'editor, è possibile verificare se sono validi facendo clic sul pulsante "Save and check". Lo script corrente verrà controllato e visualizzato nuovamente nell'editor. Tutte le linee corrette verranno visualizzate in nero. Tutte le linee in rosso indicano che contiene un errore da qualche parte su quella linea. Il commento verrà sempre visualizzato in verde.

```
1 # Delayed turn everything off
2 WAIT 5000
3 SET BIT(601/EVERYTHING OFF) TO [0-1]
```

Dopo aver corretto o aggiunto righe, puoi facilmente verificare nuovamente lo script utilizzando il pulsante "Save and check". Quando tutto è corretto è possibile utilizzare il pulsante "Save and close" per chiudere l'editor.

Nota: gli script che contengono errori possono essere salvati e avviati. Tuttavia le righe che contengono errori (e sono visualizzate in rosso) non verranno eseguite durante l'esecuzione dello script. xxter le salterà e continuerà con la riga successiva.

```
1 #Example script
2 #The following line sets a dimmer to 100%
3 SET BYTE(542/Example light) TO 100
4 #No need to add a STOP command at the end of a script.
5
```

Gli script si interromperanno automaticamente dopo l'esecuzione dell'ultimo comando dello script.

Quando si utilizza lo strumento di selezione dei comandi, parti del comando verranno visualizzate tra parentesi quadre []. Queste parti devono essere sostituite da un input valido. Il testo predefinito mostrato tra queste parentesi ti darà un'indicazione del valore richiesto.

Gli script sono scritti utilizzando comandi e valori. Inoltre, gli script possono essere estesi utilizzando variabili, calcoli, funzioni e determinate strutture di controllo come istruzioni IF e loop WHILE.



La sezione dei comandi è suddivisa nei diversi tipi principali di comandi disponibili.

Controllo dei componenti

```
SET [component] TO [value]
```

Con questo comando è possibile impostare i componenti bit, byte e float (2 byte) su un valore specificato.

Utilizzare il carattere "." come separatore per i valori in virgola mobile. È inoltre possibile utilizzare altri componenti come parte del valore di questo comando. Questo permette di impostare un componente sul valore di un altro.

```
SET [RGB component] TO [red],[green],[blue]
```

Questo comando consente di impostare un componente RGB su un colore specificato. Il colore deve essere descritto con i valori rosso, verde e blu, il tutto utilizzando un range da 0 a 255. Invece di utilizzare valori statici, è possibile sostituire una qualsiasi delle tre parti di colore primario con un componente BYTE. È inoltre possibile sostituire tutte le parti componenti insieme a un singolo altro componente RGB. Questo permette di impostare un componente RGB su un altro colore dei componenti RGB.

```
FADE [component] TO [value] IN [time]
```

Questo comando è abbastanza simile ai due comandi precedenti; è possibile utilizzarlo su byte, float e valori RGB. Invece di impostare direttamente il valore, questo comando modifica i valori a passi dal valore corrente al valore specificato. È possibile specificare il tempo come valore statico o utilizzare il valore di un componente. Le unità di tempo possibili sono MSEC/SEC/MIN/ORA/GIORNO. Se non vengono specificate unità per il valore di tempo, si assumono millisecondi. Questo comando modifica il valore del componente ogni secondo. Analogamente al comando SET, è anche possibile utilizzare questo comando in combinazione con i componenti RGB.

```
FADE [component] TO [value] in [time] STEP [time]
```

Questo comando è simile al comando precedente; aggiunge solo la specifica del tempo di passaggio. Quando il tempo di passo non è specificato, il comando FADE cambia il valore ogni secondo. Specificando il tempo di passaggio, è possibile modificarlo, ad esempio, ogni 30 minuti o 100 millisecondi. Il valore minimo è di 100 millisecondi. Analogamente al comando SET, è anche possibile utilizzare questo comando in combinazione con i componenti RGB.



Con questo comando è possibile richiedere lo stato attuale del componente sul bus. È inoltre possibile utilizzare il comando per gli scenari. Questo eseguirà una LETTURA per ogni componente incluso nello scenario.

Controllo di (altri) script

STOP

Questo comando interrompe semplicemente lo script corrente.

RESTART

Questo comando riavvia lo script corrente. In altre parole, ricomincia a correre dall'inizio. **IMPORTANTE:** il riavvio di uno script senza utilizzare un comando WAIT (vedere la sezione successiva) porterà lo script in un loop infinito diretto, che dovrebbe essere evitato.

START [script]

Questo avvierà un altro script se non è in esecuzione al momento. Sostituisci [script] con lo script effettivo che desideri avviare.

START AT BEGIN [script]

Questo avvierà un altro script se non è in esecuzione. Se è in esecuzione, dirà allo script di ricominciare dall'inizio (riavvio). Sostituire [script] con lo script effettivo che si desidera (ri)avviare.

STOP [script]

Questo interromperà un altro script se è in esecuzione. Sostituisci [script] con lo script effettivo che desideri interrompere.

SET [script] TO [on/off]

Questo comando consente di abilitare o disabilitare uno script. Sostituire [script] con lo script effettivo che si desidera abilitare/disabilitare. Tutti i valori maggiori di 0 vengono interpretati come ON, tutti i valori di 0 o inferiori vengono interpretati come OFF. Questo può essere utile quando si utilizza un determinato componente per attivare/disattivare gli script.

La disabilitazione di uno script attualmente in esecuzione arresterà anche direttamente lo script.



Utilizzo dei ritardi

WAIT [time] o WAIT [time] MSEC/SEC/MIN/HOUR/DAY

Questo comando metterà lo script in attesa per il periodo di tempo specificato. Se non vengono fornite unità di tempo, vengono assunti millisecondi. Il tempo massimo di attesa è di circa 20 giorni. Invece di immettere un valore statico, è possibile utilizzare anche un valore di un componente. (BYTE o 2BYTE)

```
1 #Example script
2 #The following line waits for the value of the BYTE in minutes.
3 WAIT BYTE(542/Wait time) MIN
4
```

WAIT [time] RANDOM [time]

Questo comando è simile al comando precedente, tuttavia, questo comando attende il tempo specificato con l'aggiunta di un tempo generato casualmente nell'intervallo indicato come secondo intervallo di tempo. Per esempio: **WAIT 1 HOUR RANDOM 30 MIN** aspetterà da qualche parte tra 1 ora e 1,5 ore. Ogni volta che il comando viene eseguito, questo verrà ricalcolato con un risultato diverso.

Scenari, avvisi e pianificatori

Scenari, avvisi e pianificatori esistenti possono essere utilizzati negli script, con i seguenti comandi:

CALL [scenario]

Questo chiamerà lo scenario specificato. Sostituire [scenario] con lo scenario effettivo che desideri chiamare.

LEARN [scenario]

Questo modificherà lo scenario. Tutti i componenti nello scenario saranno impostati al loro stato attuale nell'installazione domotica.

CALL [alert] o CALL [alert] WITH [value]

Questo permette di utilizzare il servizio di avviso dall'interno degli script. Se non viene specificato alcun valore, viene utilizzato 0. Sostituisci [alert] con l'avviso effettivo che desideri chiamare.



SET [scheduler] TO [on/off]

Questo comando consente di abilitare o disabilitare uno scheduler. Sostituisci [Scheduler] con l'effettivo Scheduler che desideri attivare o disattivare. Tutti i valori maggiori di 0 vengono interpretati come ON, tutti i valori di 0 o inferiori vengono interpretati come OFF. Tenere presente che questo funzionerà solo su pianificatori che possono essere abilitati o disabilitati manualmente. Le utilità di pianificazione che vengono abilitate o disabilitate automaticamente in base a un componente non possono essere abilitate o disabilitate da uno script.

Comandi

EXECUTE [network command]

Questo permette di utilizzare un comando di rete dall'interno degli script. Sostituire [network command] con il comando effettivo che si desidera utilizzare.

EXECUTE [ir command]

Questo permette di utilizzare un comando a infrarossi all'interno degli script. Sostituire [ir command] con il comando effettivo che si desidera utilizzare.

EXECUTE ON [upnp device] [upnp command]

Questo permette di utilizzare un comando su un dispositivo uPnP dall'interno degli script. Sostituire [upnp device] con il dispositivo uPnP desiderato e [upnp command] con il comando effettivo che si desidera utilizzare.

EXECUTE ON [diva] [diva macro]

Questo permette di utilizzare un comando su un dispositivo diva dall'interno degli script. Sostituire [diva] con il server diva desiderato e [diva macro] con la macro diva che si desidera utilizzare.

Valori

Quando si inserisce un comando nell'elenco dei comandi dello script, i valori e i componenti vengono visualizzati tra parentesi quadre []. Questi vengono visualizzati per indicare cosa inserire nella posizione appropriata.

Quando possibile, l'editor creerà un collegamento all'elenco corretto di componenti, script, avvisi, pianificazioni o scenari. È quindi possibile selezionare il valore corretto.

I componenti validi sono scritti ad esempio come **SCENARIO(123/name)**. In questo esempio si intende uno scenario con l'identificatore "123". Il nome viene visualizzato solo



per la leggibilità. Non è necessario inserire il nome corretto, perché il nome corretto verrà inserito automaticamente dopo la convalida dello script.

Per molti comandi si utilizzerà normalmente un valore statico, ad esempio **WAIT 20 SEC**. Tuttavia, in alcune circostanze potrebbe essere utile utilizzare il valore di un componente. Con tutti i comandi è possibile sostituire il valore statico con un componente.

I valori di tempo vengono assunti in millisecondi se non viene specificata alcuna unità di tempo. Le unità valide sono: MSEC, SEC, MIN, ORA e GIORNO.

Variabili e calcolo

I comandi e i valori di base possono essere ulteriormente estesi con l'uso di variabili e calcoli.

Variabili

È possibile definire le proprie variabili negli script, che sono definiti come variabili globali, utilizzabili in tutti i diversi script sul dispositivo xxter. Una variabile inizia con il simbolo del dollaro "\$", ad esempio: \$AVERAGE Il nome della variabile può essere costituito da caratteri alfanumerici (A-Z e 0-9).

È possibile utilizzare le variabili per memorizzare qualsiasi valore, ad esempio:

```
$LIGHTPERCENTAGE = 0  
SET BYTE(21/Spots) TO $LIGHTPERCENTAGE
```

Oppure

```
$LIGHTSPOTS = BYTE(21/Spots)  
SET BYTE(24/Spots Kitchen) TO $LIGHTSPOTS
```



Calcolo

È possibile utilizzare le variabili per calcolare i valori utilizzando gli operatori aritmetici. I calcoli sono scritti nel modo seguente:

```
$MAXLIGHT = ( 2 * 10 ) + 10
```

Naturalmente è possibile utilizzare anche i valori dei componenti, ad esempio:

```
$MAXLIGHT = ( 2 * BYTE(21/Spots) ) + 10
```

IMPORTANTE: quando si utilizzano i calcoli si dovrebbe sempre fare uso di staffe. Quando non vengono utilizzate parentesi, il calcolo viene elaborato nell'ordine in cui è scritto! Ad esempio, $2 * 10 + 10$ darà come risultato 30, ma $10 + 2 * 10$ darà come risultato 24. Può essere evitato scrivendo: $10 + (2*10)$. Tenerlo sempre a mente quando si usano i calcoli.

I valori a virgola mobile possono essere utilizzati utilizzando "." come separatore, ad esempio 2,42

Si noti che tutte le variabili sono larghe xxter. Quindi, se si imposta una variabile specifica in uno script, anche un altro script può utilizzare quel valore. Se si desidera utilizzare una variabile specifica solo in uno script, assicurarsi che altri script non utilizzino lo stesso nome di variabile.

Possono essere utilizzati i seguenti operatori:

+	più	
-	meno	
*	per	
/	diviso	
%	modulo (avanzo di divisione)	
^	potenza	
&	e	(risultato = 1 se entrambi i valori sono maggiori di 0, 0 in caso contrario)
	o	(risultato = 1 se uno dei valori è maggiore di 0, in caso contrario 0)
=	è uguale a	(risultato = 1 se vero, 0 in caso contrario)
<	è minore di	(risultato = 1 se vero, 0 in caso contrario)
>	è maggiore di	(risultato = 1 se vero, 0 in caso contrario)
<>	non è uguale a	(risultato = 1 se vero, 0 in caso contrario)

Una radice quadrata può essere fatta con l'operatore di potenza: $^ (1/2)$ = alla potenza di $\frac{1}{2}$ = la radice quadrata del valore.



Funzioni

Ci sono diverse funzioni disponibili negli script per calcolare i valori. Queste funzioni possono essere utilizzate in combinazione con comandi o con variabili.

Funzioni matematiche

`ROUND (value)` o `ROUND (value, precision)`

Restituisce il valore arrotondato del valore dato. Quando non viene specificata alcuna precisione, viene utilizzata una precisione di 0. Ad esempio `ROUND(4.3) = 4` e `ROUND(4.321, 1) = 4.3`

`FLOOR (value)` o `FLOOR (value, precision)`

Restituisce il valore a pavimento (arrotondato per difetto) del valore dato. Quando non viene specificata alcuna precisione, viene utilizzata una precisione di 0. Ad esempio `PIANO(4.7) = 4` e `PIANO(4.987, 1) = 4.9`

`CEIL (value)` o `CEIL (value, precision)`

Restituisce il valore massimo (arrotondato per eccesso) del valore dato. Quando non viene specificata alcuna precisione, viene utilizzata una precisione di 0. Ad esempio `CEIL(4.3) = 5` e `CEIL(4.321, 1) = 4.4`

`ABS (value)`

Restituisce il valore assoluto del valore dato, quindi `ABS(-4) = 4` e `ABS(6.13) = 6.13`.

`MIN (valore, valore, valore, ..)`

Restituisce il valore minimo di tutti i parametri dati. Richiede almeno un parametro.

`MAX (valore, valore, valore, ..)`

Restituisce il valore massimo di tutti i parametri dati. Richiede almeno un parametro.

`AVG (valore, valore, valore, ..)`

Restituisce il valore medio di tutti i parametri dati. Richiede almeno un parametro.

`RANDOM (valore massimo)`

Restituisce un valore casuale compreso tra 0 e il valore massimo dato.



Funzioni temporali

NOW ()

Restituisce l'ora corrente come timestamp in secondi dal 1° gennaio 1970 (data unix). Questo timestamp può essere confrontato con un valore precedente per determinare quanto tempo è passato.

DAYTIME () di DAYTIME (timestamp)

Restituisce 1 se il sole è alto e 0 se il sole è basso. Quando viene fornito un timestamp, viene utilizzato questo tempo, altrimenti viene utilizzato l'ora corrente.

SECOND () di SECOND (timestamp)

Restituisce la quantità di secondi (0-59). Quando viene fornito un timestamp, viene utilizzato questo tempo, altrimenti viene utilizzato l'ora corrente.

MINUTE () di MINUTE (timestamp)

Restituisce la quantità di minuti (0-59). Quando viene fornito un timestamp, viene utilizzato questo tempo, altrimenti viene utilizzato l'ora corrente.

HOUR () di HOUR (timestamp)

Restituisce la quantità di ore (0-23). Quando viene fornito un timestamp, viene utilizzato questo tempo, altrimenti viene utilizzato l'ora corrente.

DAYOFWEEK () di DAYOFWEEK (timestamp)

Restituisce il giorno della settimana (1-7). Quando viene fornito un timestamp, viene utilizzato questo tempo, altrimenti viene utilizzato l'ora corrente. 1 = Domenica, 2 = Lunedì... 7 = Sabato.

DAYOFMONTH () di DAYOFMONTH (timestamp)

Restituisce il giorno del mese (1-31). Quando viene fornito un timestamp, viene utilizzato questo tempo, altrimenti viene utilizzato l'ora corrente.



DAYOFYEAR ()

di DAYOFYEAR (timestamp)

Restituisce il giorno dell'anno (1-366). Quando viene fornito un timestamp, viene utilizzato questo tempo, altrimenti viene utilizzato l'ora corrente.

MONTH () di MONTH (timestamp)

Restituisce il mese (1-12). Quando viene fornito un timestamp, viene utilizzato questo tempo, altrimenti viene utilizzato l'ora corrente.

YEAR () di YEAR (timestamp)

Restituisce l'anno, ad esempio "2015". Quando viene fornito un timestamp, viene utilizzato questo tempo, altrimenti viene utilizzato l'ora corrente.

Funzioni di testo

CONCAT ("abc", "def",)

Concatena i testi insieme, ad esempio CONCAT ("abc", "def", "ghi") produce "abcdefghi". In alternativa è possibile utilizzare il punto (" . ") come operatore di testo per concatenare i testi, ad esempio "abc"."def"."ghi" risulta in "abcdefghi".

FORMAT (format, parameters)

Può essere utilizzato per formattare il testo. Il campo formato descrive come, successivamente, possono essere aggiunti i parametri da utilizzare. Il testo standard verrà copiato direttamente, inoltre sono disponibili le seguenti opzioni di formato speciali:

- Utilizzare %s per includere un valore come testo, ad esempio: FORMAT("Now it is %s.", "dry") produce "Now it is dry."
- Usa %d per includere un valore intero nel testo, ad esempio: FORMAT("It is %d degrees.", 10) produce "It is 10 degrees."
- Utilizzare %f per includere un valore a virgola mobile nel testo, ad esempio: FORMAT("It is %f degrees.", 10.12345) produce "It is 10.12345 degrees." %f ha un parametro opzionale per la precisione decimale, usa %.xd come parametro di formato dove x è il numero di decimali. Prestare attenzione al "." Dopo il segno %. Ad esempio: FORMAT("It is %.2f degrees.", 10.12345) produce "It is 10.12 degrees."
- Utilizzare %% per visualizzare un simbolo %, ad esempio: FORMAT("The window is %d%% closed.", 10) produce "The window is 10% closed."



`SUBSTR(text, startindex)` or
`SUBSTR(text, startindex, length)`

Può sottrarre una parte di un altro testo. Si noti che l'indice di avvio inizia da 0. È possibile utilizzare questa funzione con o senza lunghezza. Senza il parametro `length`, la parte di testo verrà utilizzata fino alla fine del testo.

Ad esempio: `SUBSTR("abcdef", 3)` produce "def"

Ad esempio: `SUBSTR("abcdef", 3, 1)` produce "d"

`STRLEN(text)`

Questa funzione restituisce la lunghezza del testo dato, ad esempio `STRLEN("abcdef")` produce 6.

Strutture di comando

IF e WHILE

Le istruzioni loop `IF` e `WHILE` possono essere utilizzate per eseguire azioni specifiche solo se viene soddisfatta una condizione. La condizione può essere definita utilizzando tutti gli operatori aritmetici come spiegato sopra.

Ad esempio:

```
IF $LIGHT > 33.33
  $LIGHT = 33.33
```

L'azione che viene scritta sulla seconda riga viene eseguita solo quando il calcolo dietro l'istruzione `IF` è VERO (maggiore di 0). L'azione verrà saltata se il risultato è FALSO (0 o inferiore).

Se viene utilizzata un'istruzione `IF`, è possibile aggiungere anche un'istruzione `ELSE`, in questo modo:

```
IF $LIGHT > 33.33
  $X = 100
ELSE
  $X = 0
```

L'istruzione `ELSE` dovrebbe essere sempre su una riga separata, direttamente sotto il comando da eseguire quando viene soddisfatta la condizione dell'istruzione `IF`.



Oltre all'istruzione IF, è possibile utilizzare anche un'istruzione WHILE. La differenza tra un'istruzione IF e un'istruzione WHILE è che un'istruzione IF viene eseguita una volta e un'istruzione WHILE verrà ripetuta fino a quando la condizione non viene più soddisfatta, ad esempio:

```
$COUNT = 0  
WHILE $COUNT < 10  
    $COUNT = $COUNT+1
```

L'esempio precedente aumenterà il CONTEGGIO delle variabili fino a quando non sarà 10 o superiore. Come eseguire più comandi sarà spiegato nella prossima sezione.

IMPORTANTE: quando la condizione non finisce mai, il loop WHILE continuerà indefinitamente. Consentire questo senza utilizzare un comando WAIT porterà lo script in un loop infinito diretto, che dovrebbe essere evitato.



INIZIO e FINE

Se è necessario eseguire più di un comando dopo un'istruzione IF, WHILE o ELSE, è possibile utilizzare le istruzioni BEGIN e END.

Anche questi dovrebbero essere scritti su righe separate, ad esempio:

```
$X = 0
WHILE $X < 11
  BEGIN
    $Y = $X * 10
    SET BYTE(22/EXAMPLE BYTE) TO $Y
    $X = $X + 1
  END
```

Un altro esempio:

```
IF $LIGHT > 33.33
  BEGIN
    $X = 100
    $Y = 0
  END
ELSE
  BEGIN
    $X = 0
    $Y = 100
  END
```



Altro

`SEND WAKEONLAN TO [ma:c :ad:re:ss]`

Con questo comando è possibile inviare un comando wake on lan a un dispositivo di rete. Sostituire [ma:c :ad:re:ss] con l'indirizzo MAC del dispositivo desiderato.

`OPEN KNXTUNNEL`

Con questo comando è possibile abilitare l'accesso al tunnel knx sul dispositivo xxter per la programmazione con ETS per un periodo di 8 ore.

`CLOSE KNXTUNNEL`

Con questo comando è possibile disabilitare l'accesso al tunnel knx sul dispositivo xxter per la programmazione con ETS

L'utilizzo di tutti i comandi, i valori, le funzioni e le strutture di controllo descritti ti consentirà di creare quasi tutte le funzionalità richieste.