



Manual de scripts xxter

Lo básico	2
Gestiónde scripts	3
Editor de scripts	4
Mandos	6
Componentes de control	6
Controlar (otros) scripts	7
Uso de retardos	8
Escenarios, alertas y programadores	8
Mandos	9
Valores	9
Variables y cálculo	10
Variables	10
Cálculo	10
Funciones	12
Funciones matemáticas	12
Funciones de tiempo	13
Funciones de texto	14
Estructuras de control	16
IF y WHILE	16
BEGIN y END	17
Otros	18



scripts de xxter

Con los scripts de xxter puede crear sus propios programas pequeños dentro de xxter. Estos scripts son muy flexibles y pueden utilizarse para añadir una amplia variedad de funciones a una instalación domótica en una vivienda. Puede crear funciones lógicas, retrasar acciones, ampliar sus escenarios con secuencias RGB y mucho más. Los scripts se pueden activar utilizando un escenario, una programación, una acción (desencadenante) u otro script.

Además de este manual, en nuestro sitio web hay varios ejemplos para descargar y crear funciones específicas con scripts xxter. Estos ejemplos pueden ser muy útiles no solo para estas funciones específicas, sino también para aprender a utilizar los scripts xxter.

Lo básico

los scripts xxter son programas escritos que consisten en una o más líneas. Cada línea debe contener solo un mando específico. No puede poner varios mandos en una línea. Puede añadir comentarios a un guión escribiendo un carácter «#». Esto se puede hacer en una línea separada o al final de un mando.

Un ejemplo de un script de 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
```

Los scripts se pueden iniciar y detener por muchos medios, por ejemplo, por un escenario o una programación. Los scripts pueden ser scripts «interminables» que realizan ciertas acciones en intervalos repetidos o pueden definirse como una secuencia de mandos que se ejecutan solo una vez, cada vez que se inicia el script.



Gestión de scripts

los scripts xxter se pueden crear y actualizar utilizando el editor xxter. Este editor se puede encontrar en la configuración del proyecto de «Mi xxter». Seleccione el proyecto para el que desea gestionar los scripts y seleccione «Scripts» en la barra de menú. Todos los scripts existentes se mostrarán aquí y se pueden editar y eliminar.

Test Project Project has changed

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

×

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

Importante: Cuando añade o edita un script, este debe cargarse en el dispositivo xxter antes de poder usarlo.

Cada vez que se ha realizado un cambio, puede ver el texto «Project has changed» en la esquina superior derecha. Al hacer clic en esto y posteriormente en «Push configuration to xxter device», el dispositivo descargará la nueva configuración. Esta función solo se puede utilizar cuando xxter está configurado para su uso en el exterior (consulte el capítulo 13 del manual de instalación). Alternativamente, puede volver a cargar el perfil utilizando la aplicación (consulte el capítulo 11 del manual del usuario). Por supuesto, también puede volver a cargar el perfil desde el propio dispositivo xxter, iniciando sesión en el dispositivo y haciendo clic en el pulsador «Load configuration» en la esquina superior izquierda del menú.

Cuando haya iniciado sesión en el dispositivo, puede ver los scripts cargados haciendo clic en la opción «Scripts» en el menú. Aquí también se puede iniciar, reiniciar o detener manualmente un script con fines de prueba.

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 de scripts

Cuando añada o edite un script, puede seleccionar si debe estar habilitado y puede seleccionar si el script debe estar disponible para el usuario final, para usar en escenarios o para el programador. También puede seleccionar si el script debe iniciarse automáticamente, siempre que se (re) inicie el dispositivo xxter.

IMPORTANTE: Los scripts deshabilitados no se pueden ejecutar y no se activarán cuando se incluyan en un escenario, acción o programador. Esto puede ser útil para pruebas y propuestas de solución de problemas. Sin embargo, ¡tenga en cuenta que un script puede ser habilitado o deshabilitado por otro 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

Filter

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

Los números de línea que se muestran junto al script son solo informativos y no se utilizan en los propios scripts. Los mandos se pueden añadir escribiéndolos directamente o utilizando la herramienta de selección de mandos, situada en la esquina superior derecha. Asegúrese de estar en la posición correcta en el editor de scripts al añadir un mando con la herramienta de mandos. Al añadir un mando con la herramienta de mandos, los campos apropiados con respecto a su configuración xxter se añadirán como enlaces automáticamente. Esto permite una forma rápida y fácil de añadir mandos correctos utilizando el editor.

1
2
3

SET [SCHEDULER\(1/Away\)](#) TO OFF



Después de haber añadido uno o más mandos en el editor, puede verificar si son válidos haciendo clic en el pulsador «Save and check». Su script actual se verificará y se volverá a mostrar en el editor. Todas las líneas correctas se mostrarán en negro. Todas las líneas en rojo indican que contiene un error en alguna parte de esa línea. El comentario siempre se mostrará en verde.

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

Después de corregir o añadir líneas, puede verificar fácilmente el script nuevamente utilizando el pulsador «Save and check». Cuando todo esté correcto puede utilizar el pulsador «Save and close» para cerrar el editor.

Nota: los scripts que contienen errores se pueden guardar e iniciar. Sin embargo, las líneas que contienen errores (y se muestran en rojo) no se ejecutarán durante la ejecución del script. xxter las omitirá y continuará con la siguiente línea.

```
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
```

Los scripts dejarán de ejecutarse automáticamente después de que se haya ejecutado el último mando del script.

Cuando se utiliza la herramienta de selección de mandos, las partes del mando se mostrarán entre corchetes []. Estas piezas deben reemplazarse por una entrada válida. El texto predeterminado que se muestra entre estos corchetes le dará una indicación del valor requerido.

Los scripts se escriben usando mandos y valores. Además, los scripts se pueden extender utilizando variables, cálculos, funciones y ciertas estructuras de control como sentencias IF y bucles WHILE.



Mandos

La sección de mandos se divide en los diferentes tipos principales de mandos disponibles.

Componentes de control

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

Con este mando puede establecer los componentes bit, byte y float (2byte) en un valor especificado.

Utilice el carácter «.» como separador para los valores de coma flotante. También puede usar otros componentes como parte de valor de este mando. Esto le permite establecer un componente en el valor de otro.

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

Este mando le permite establecer un componente RGB en un color especificado. El color debe describirse con los valores rojo, verde y azul, todos utilizando un rango de 0-255. En lugar de usar valores estáticos, puede reemplazar cualquiera de las tres partes de color primario con un componente de BYTE. También puede reemplazar todos los componentes junto con otro componente RGB. Esto le permite establecer un componente RGB en otro color de componentes RGB.

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

Este mando es bastante similar a los dos mandos anteriores; puede usarlo en bytes, flotantes y valores RGB. En lugar de establecer el valor directamente, este mando cambia los valores en pasos del valor actual al valor especificado. Puede especificar el tiempo como un valor estático o usar el valor de un componente. Las unidades de tiempo posibles son MSEG/SEG/MIN/HORA/DÍA. Si no se especifican unidades para el valor de tiempo, se asumen milisegundos. Este mando cambia el valor del componente cada segundo. Al igual que el mando SET, también puede usar este mando en combinación con componentes RGB.

```
FADE [component] A [value] EN [time] PASO [time]
```

Este mando es similar al mando anterior; solo añade la especificación del tiempo de paso. Cuando no se especifica el tiempo del paso, el mando FADE cambia el valor cada segundo. Al especificar el tiempo del paso, puede cambiarlo, por ejemplo, cada 30 minutos o 100 milisegundos. El valor mínimo es de 100 milisegundos. Al igual que el mando SET, también puede usar este mando en combinación con componentes RGB.



`READ [component]`

Con este mando puede solicitar el estado actual del componente en el bus. También puede usar el mando para escenarios. Esto realizará un READ para cada componente que se incluya en el escenario.

Controlar (otros) scripts

`STOP`

Este mando simplemente detiene el script actual.

`RESTART`

Este mando reinicia el script actual. En otras palabras, comienza a funcionar de nuevo desde el principio.

IMPORTANTE: reiniciar un script sin usar un mando WAIT (consulte la siguiente sección) llevará el script a un bucle infinito directo, que debe evitarse.

`START [script]`

Esto iniciará otro script si no se está ejecutando en este momento. Reemplace [script] con el script real que desea iniciar.

`START AT BEGIN [script]`

Esto iniciará otro script si no se está ejecutando. Si se está ejecutando, le dirá al script que comience de nuevo desde el principio (reiniciar). Reemplace [script] con el script real que desea (re)iniciar.

`STOP [script]`

Esto detendrá otro script si se está ejecutando. Reemplace [script] con el script real que desea detener.

`SET [script] TO [on/off]`

Este mando le permite habilitar o deshabilitar un script. Reemplace [script] que el script real desea habilitar/deshabilitar. Todos los valores mayores que 0 se interpretan como ON, todos los valores de 0 o inferiores se interpretan como OFF. Esto puede ser útil cuando se utiliza un determinado componente para activar/desactivar scripts.

Deshabilitar un script que se está ejecutando actualmente también detendrá directamente el script.



Uso de retardos

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

Este mando pondrá el script en espera durante el periodo de tiempo dado. Si no se suministran unidades de tiempo, se asumen milisegundos. El tiempo máximo de espera es de aproximadamente 20 días. En lugar de introducir un valor estático, también puede utilizar un valor de 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]

Este mando es similar al mando anterior, sin embargo, este mando espera el tiempo dado con una adición de un tiempo generado aleatoriamente en el intervalo dado como el segundo rango de tiempo. Para ejemplo: **WAIT 1 HOUR RANDOM 30 MIN** esperará en algún lugar entre 1 hora y 1,5 horas. Cada vez que se ejecute el mando, este se recalculará con un resultado diferente.

Escenarios, alertas y programadores

Los escenarios, alertas y programadores existentes se pueden utilizar en scripts, con los siguientes mandos:

CALL [scenario]

Esto llamará al escenario especificado. Reemplace [escenario] con el escenario real al que desea llamar.

LEARN [scenario]

Esto modificará el escenario. Todos los componentes del escenario se establecerán en su estado actual en la instalación de domótica.

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

Esto le permite utilizar el servicio de alertas desde dentro de los scripts. Si no se especifica ningún valor, se utiliza 0. Reemplace [alert] con la alerta real a la que desea llamar.



```
SET [scheduler] TO [on/off]
```

Este mando le permite habilitar o deshabilitar un programador. Reemplace [Scheduler] con el programador real que desea activar o desactivar. Todos los valores mayores que 0 se interpretan como ON, todos los valores de 0 o inferiores se interpretan como OFF. Tenga en cuenta que esto solo funcionará en programadores que se puedan habilitar o deshabilitar manualmente. Los programadores que se habilitan o deshabilitan automáticamente en función de un componente no se pueden habilitar o deshabilitar mediante un script.

Mandos

```
EXECUTE [network command]
```

Esto le permite usar un mando de red desde dentro de los scripts. Reemplace [network command] con el mando real que desea usar.

```
EXECUTE [ir command]
```

Esto le permite usar un mando de infrarrojos desde dentro de los scripts. Reemplace [ir command] con el mando real que desea usar.

```
EXECUTE ON [upnp device] [upnp command]
```

Esto le permite usar un mando en un dispositivo uPnP desde dentro de los scripts. Reemplace [upnp device] con el dispositivo uPnP deseado y [upnp command] con el mando real que desea usar.

```
EXECUTE ON [diva] [diva macro]
```

Esto le permite usar un mando en un dispositivo diva desde dentro de los scripts. Reemplace [diva] con el servidor diva deseado y [diva macro] con el diva macro que desea utilizar.

Valores

Cuando inserta un mando a través de la lista de mandos de script, los valores y componentes se muestran entre corchetes []. Estos se muestran para indicar lo que debe introducir en la ubicación apropiada.

Cuando sea posible, el editor hará un enlace a la lista correcta de componentes, scripts, alertas, horarios o escenarios. A continuación, se puede seleccionar el valor correcto.

Los componentes válidos se escriben, por ejemplo, como **SCENARIO(123/name)**. En este ejemplo, se pretende un escenario con el identificador «123». El nombre se muestra solo para facilitar su lectura. No tiene que introducir el nombre correcto, porque el nombre correcto se insertará automáticamente después de que se valide el script.



Para muchos mandos, normalmente utilizará un valor estático, por ejemplo, **WAIT 20 SEC**. Sin embargo, en algunas circunstancias podría ser útil usar el valor de un componente en su lugar. Con todos los mandos puede reemplazar el valor estático con un componente.

Los valores de tiempo se asumen en milisegundos si no se especifica ninguna unidad de tiempo. Las unidades válidas son: MSEG, SEG, MIN, HORA y DÍA.

Variables y cálculo

Los mandos y valores básicos se pueden ampliar aún más con el uso de variables y cálculos.

Variables

Puede definir sus propias variables en scripts, que se definen como variables globales, utilizables en todos los scripts diferentes en el dispositivo xxter. Una variable comienza con un signo de dólar «\$», por ejemplo: \$AVERAGE El nombre de la variable puede consistir en caracteres alfanuméricos (A-Z y 0-9).

Puede utilizar variables para almacenar cualquier valor, por ejemplo:

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

O

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

Cálculo

Puede utilizar variables para calcular valores utilizando operadores aritméticos. Los cálculos se escriben de la siguiente manera:

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

Por supuesto, también puede usar valores de componentes, por ejemplo:

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

IMPORTANTE: al usar cálculos, siempre debe hacer uso de corchetes. Cuando no se utilizan corchetes, el cálculo se procesa en el orden en que está escrito. Por ejemplo, $2 * 10 + 10$ resultará en 30, pero $10 + 2 * 10$ tendrá un resultado de 24. Puede evitar esto escribiendo: $10 + (2*10)$. Tenga siempre esto en cuenta al utilizar los cálculos.



Los valores de coma flotante se pueden usar usando el «.» como separador, por ejemplo 2.42

Tenga en cuenta que todas las variables son xxter anchas. Entonces, si establece una variable específica en un script, otro script también puede usar ese valor. Si desea usar una variable específica solo en un script, asegúrese de que otros scripts no usen el mismo nombre de variable.

Se pueden utilizar los siguientes operadores:

+	sumar	
-	restar	
*	multiplicar	
/	dividir	
%	módulo (resto de la división)	
^	potencia	
&	y	(resultado = 1 si ambos valores son mayores que 0, 0 si no)
	o	(resultado = 1 si uno de los valores es mayor que 0, 0 si no)
=	es igual a	(resultado = 1 si es verdadero, 0 si no)
<	es menor que	(resultado = 1 si es verdadero, 0 si no)
>	es mayor que	(resultado = 1 si es verdadero, 0 si no)
<>	no es igual a	(resultado = 1 si es verdadero, 0 si no)

Se puede hacer una raíz cuadrada con el operador de energía: $^{(1/2)}$ = a la potencia de $1/2$ = la raíz cuadrada del valor.



Funciones

Hay varias funciones disponibles en los scripts para calcular valores. Estas funciones se pueden utilizar en combinación con mandos o con variables.

Funciones matemáticas

`ROUND (valor)` o `ROUND (valor, precision)`

Devuelve el valor redondeado del valor dado. Cuando no se especifica precisión, se utiliza una precisión de 0. Por ejemplo `ROUND(4.3) = 4` and `ROUND(4.321, 1) = 4.3`

`FLOOR (valor)` o `FLOOR (valor, precision)`

Devuelve el valor mínimo (redondeado hacia abajo) del valor dado. Cuando no se especifica precisión, se utiliza una precisión de 0. Por ejemplo, `FLOOR(4.7) = 4` and `FLOOR(4.987, 1) = 4.9`

`CEIL (valor)` o `CEIL (valor, precision)`

Devuelve el valor máximo (redondeado hacia arriba) del valor dado. Cuando no se especifica precisión, se utiliza una precisión de 0. Por ejemplo, `CEIL(4.3) = 5` y `CEIL(4.321, 1) = 4.4`

`ABS (valor)`

Devuelve el valor absoluto del valor dado, por lo que `ABS(-4) = 4` y `ABS(6.13) = 6.13`.

`MIN (valor, valor, valor, ...)`

Devuelve el valor mínimo de todos los parámetros dados. Requiere un mínimo de un parámetro.

`MAX (valor, valor, valor, ...)`

Devuelve el valor máximo de todos los parámetros dados. Requiere un mínimo de un parámetro.

`AVG (valor, valor, valor, ...)`

Devuelve el valor medio de todos los parámetros dados. Requiere un mínimo de un parámetro.



RANDOM (valor máximo)

Devuelve un valor aleatorio entre 0 y el valor máximo dado.

Funciones de tiempo

NOW ()

Devuelve la hora actual como marca de tiempo en segundos desde el 1 de enero de 1970 (fecha Unix). Esta marca de tiempo se puede comparar con un valor anterior para determinar cuánto tiempo ha pasado.

DAYTIME () de DAYTIME(marca de tiempo)

Devuelve un 1 si el sol está arriba y un 0 si el sol está abajo. Cuando se da una marca de tiempo, se utiliza esta vez, de lo contrario se utiliza la hora actual.

SECOND () de SECOND(marca de tiempo)

Devuelve la cantidad de segundos (0-59). Cuando se da una marca de tiempo, se utiliza esta vez, de lo contrario se utiliza la hora actual.

MINUTE () de MINUTE(marca de tiempo)

Devuelve la cantidad de minutos (0-59). Cuando se da una marca de tiempo, se utiliza esta vez, de lo contrario se utiliza la hora actual.

HOURL () de HOURL(marca de tiempo)

Devuelve la cantidad de horas (0-23). Cuando se da una marca de tiempo, se utiliza esta vez, de lo contrario se utiliza la hora actual.

DAYOFWEEK () de DAYOFWEEK(marca de tiempo)

Devuelve el día de la semana (1-7). Cuando se da una marca de tiempo, se utiliza esta vez, de lo contrario se utiliza la hora actual. 1 = Domingo, 2 = Lunes,.. 7 = Sábado.

DAYOFMONTH () de DAYOFMONTH(marca de tiempo)

Devuelve el día del mes (1-31). Cuando se da una marca de tiempo, se utiliza esta vez, de lo contrario se utiliza la hora actual.



`DAYOFYEAR ( )` de `DAYOFYEAR(marca de tiempo)`

Devuelve el día del año (1-366). Cuando se da una marca de tiempo, se utiliza esta vez, de lo contrario se utiliza la hora actual.

`MONTH ( )` del `MONTH(marca de tiempo)`

Devuelve el mes (1-12). Cuando se da una marca de tiempo, se utiliza esta vez, de lo contrario se utiliza la hora actual.

`YEAR ( )` del `YEAR(marca de tiempo)`

Devuelve el año, por ejemplo «2015». Cuando se da una marca de tiempo, se utiliza esta vez, de lo contrario se utiliza la hora actual.

Funciones de texto

`CONCAT ("abc", "def", . . . . )`

Concatena textos juntos, por ejemplo `CONCAT("abc", "def", "ghi")` da como resultado "abcdefghi". Alternativamente, puede usar el punto (".") como operador de texto para concatenar texto, por ejemplo, "abc"."def"."ghi" da como resultado "abcdefghi".

`FORMAT (format, parameters. . . .)`

Se puede utilizar para dar formato al texto. El campo de formato describe cómo, después, se pueden añadir parámetros para ser utilizados. El texto estándar se copiará directamente, además están disponibles las siguientes opciones de formato especial:

- Utilice %s para incluir un valor como texto, por ejemplo: `FORMAT("Now it is %s.", "dry")` da como resultado "Now it is dry."
- Utilice %d para incluir un valor entero en el texto, por ejemplo: `FORMAT("It is %d degrees.", 10)` da como resultado "It is 10 degrees."
- Utilice %f para incluir un valor de coma flotante en el texto, por ejemplo: `FORMAT("It is %f degrees.", 10.12345)` da como resultado "It is 10.12345 degrees."

%f tiene un parámetro opcional para la precisión decimal, use %.xd como parámetro de formato donde x es el número de decimales. Preste atención a la "." Después del signo %. Por ejemplo:

`FORMAT("It is %.2f degrees.", 10.12345)` da como resultado "It is 10.12 degrees."

- use %% para mostrar un símbolo %, por ejemplo: `FORMAT("The window is %d%% closed.", 10)` da como resultado "The window is 10% closed."



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

Puede restar una parte de otro texto. Tenga en cuenta que el índice de inicio comienza en 0. Puede usar esta función con o sin longitud. Sin el parámetro `length`, la parte `text` se utilizará hasta el final del texto.

Por ejemplo: `SUBSTR("abcdef", 3)` da como
resultado "def" Por ejemplo: `SUBSTR("abcdef", 3, 1)`
da como
resultado "d"

`STRLEN(text)`

Esta función devuelve la longitud del texto dado, por ejemplo `STRLEN("abcdef")` da como resultado 6.



Estructuras de control

IF and WHILE

Las instrucciones IF y los bucles WHILE se pueden utilizar para ejecutar acciones específicas solo si se cumple una condición. La condición se puede definir utilizando todos los operadores aritméticos como se explicó anteriormente.

Por ejemplo:

```
IF $LIGHT > 33,33  
  $LIGHT = 33,33
```

La acción que se escribe en la segunda línea solo se ejecuta cuando el cálculo detrás de la instrucción IF es TRUE (mayor que 0). La acción se omitirá si el resultado es FALSE (0 o menos).

Si se utiliza una instrucción IF, también se puede añadir una instrucción ELSE, como esta:

```
IF $LIGHT > 33,33  
  $X = 100  
ELSE  
  $X = 0
```

La instrucción ELSE siempre debe estar en una línea separada, directamente debajo del mando a ejecutar cuando se cumple la condición de la instrucción IF.

Además de la instrucción IF, también se puede utilizar una instrucción WHILE. La diferencia entre una instrucción IF y una instrucción WHILE es que una instrucción IF se realiza una vez y una instrucción WHILE se repetirá hasta que la condición ya no se cumpla, por ejemplo:

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

El ejemplo anterior aumentará el COUNT de variables hasta que sea 10 o más. En la siguiente sección se explicará cómo ejecutar varios mandos.

IMPORTANTE: cuando la condición nunca termina, el bucle WHILE continuará indefinidamente. Permitir esto sin usar un mando WAIT traerá el script en un bucle infinito directo, lo que debe evitarse.



BEGIN y END

Si se necesita ejecutar más de un mando después de una instrucción IF, WHILE o ELSE, se pueden utilizar las instrucciones BEGIN y END.

Estos también deben escribirse en líneas separadas, por ejemplo:

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

Otro ejemplo:

```
IF $LIGHT > 33,33
  BEGIN
    $X = 100
    $Y = 0
  END
ELSE
  BEGIN
    $X = 0
    $Y = 100
  END
```



Otros

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

Con este mando puede enviar un mando wake on lan a un dispositivo de red. Reemplace [ma:c :ad:re:ss] con la dirección mac del dispositivo previsto.

OPEN KNXTUNNEL

Con este mando puede habilitar el acceso al túnel knx en el dispositivo xxter para programar con ETS durante un periodo de 8 horas.

CLOSE KNXTUNNEL

Con este mando puede desactivar el acceso al túnel knx en el dispositivo xxter para programar con ETS

El uso de todos los mandos, valores, funciones y estructuras de control descritos le permitirá crear casi cualquier funcionalidad requerida.