NETx BMS Studio

Interfaz XCON.EMAIL

Uso y script ejemplo

MANUAL DE CONFIGURACIÓN DE EMAIL EN XCON

El sistema NETx BMS Studio utiliza la interfaz XCON para integrar diferentes plataformas de comunicación. Sirve tanto para el envío de emails como para la configuración de comunicación vía TCP/IP.

El objetivo de este manual es describir el funcionamiento y configuración de la opción EMAIL dentro del interfaz XCON.

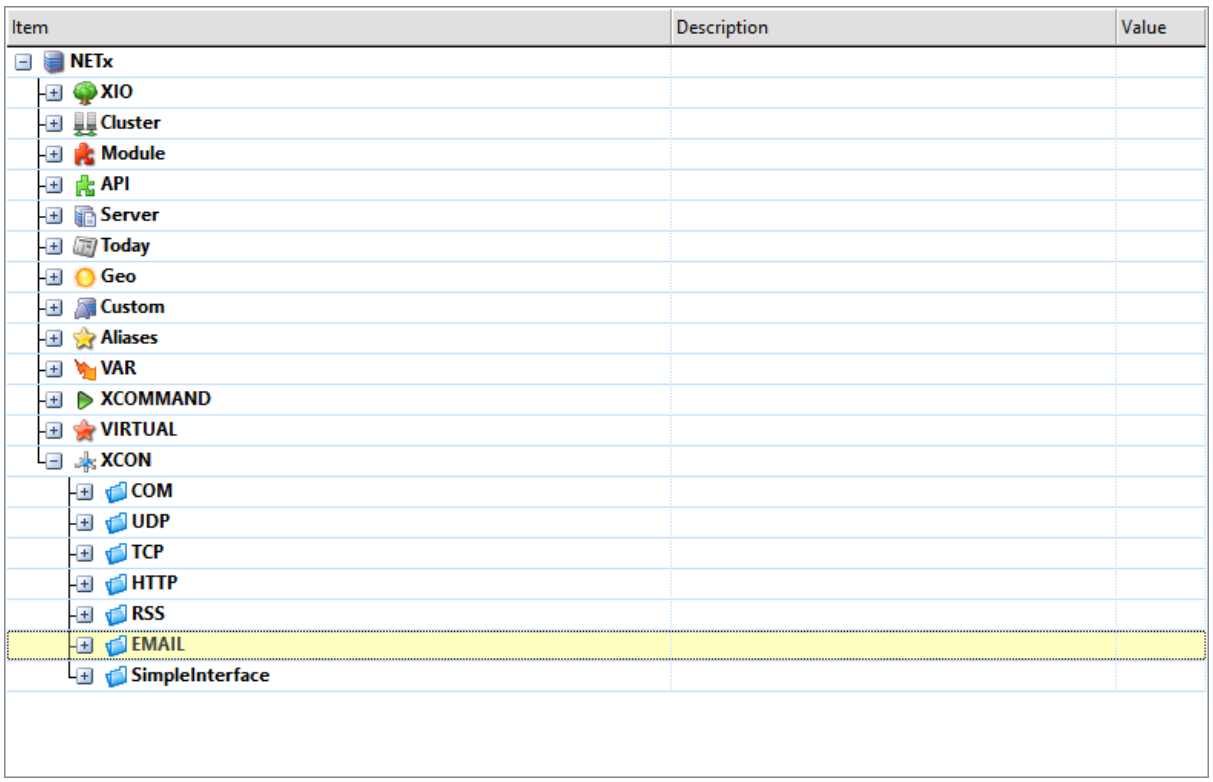

Imagen 1

Para poder utilizar la función primero deberemos configurar el servidor de correo SMTP en la pestaña de configuración del NETx.

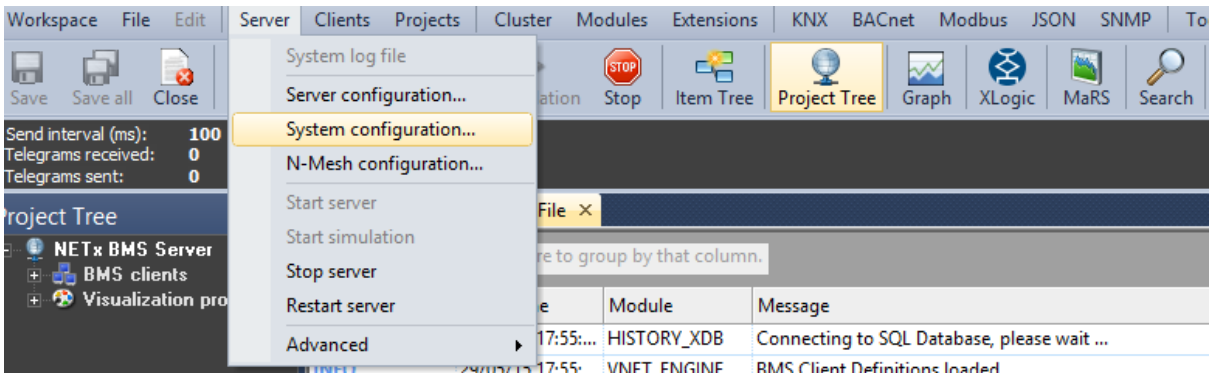

Imagen 2

En este caso, vamos a modificar los parámetros para un servidor de correo Gmail. En la *Imagen 3* hay que tener en cuenta que el puerto 465 corresponde con el sistema de encriptación SSL y sustituir usuario por el email desde el que queremos mandar los emails.

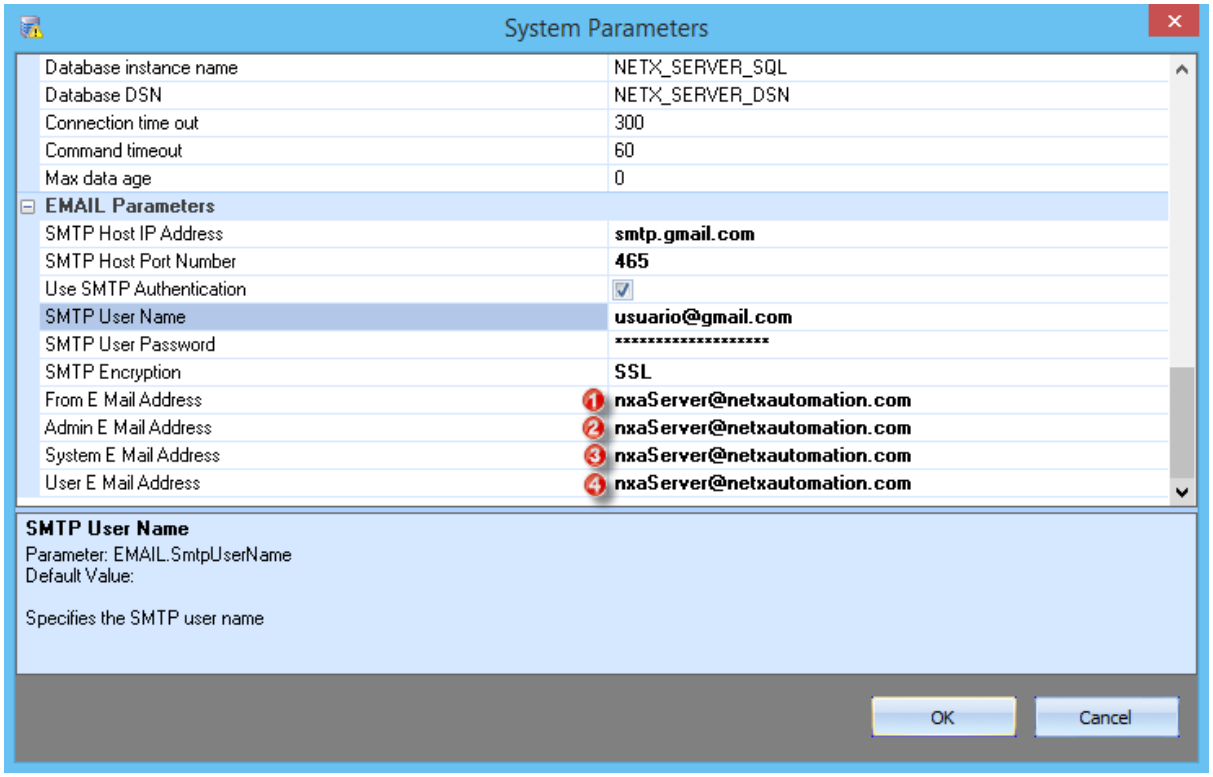

Imagen 3

- 1 Email desde el que queremos enviar la notificación.
- 2 Email de destino al administrador del programa.
- 3 Email de destino al instalador del sistema.
- 4 Email de destino al usuario del sistema.

Una vez que hemos configurado el servidor de correo con el que queremos trabajar, ya estamos preparados para enviar notificaciones por email a los distintos interesados de la instalación.

Antes de poder enviar notificaciones, el SMTP de Gmail nos puede pedir que habilitemos el acceso a aplicaciones menos seguras. Al intentar enviar un correo desde el NETx, recibiremos un email, en la dirección que hemos configurado, alertándonos de que una aplicación “no segura” intenta conectarse. Si no es así, encontraremos la información en el siguiente enlace:

https://www.google.com/settings/security/lesssecureapps

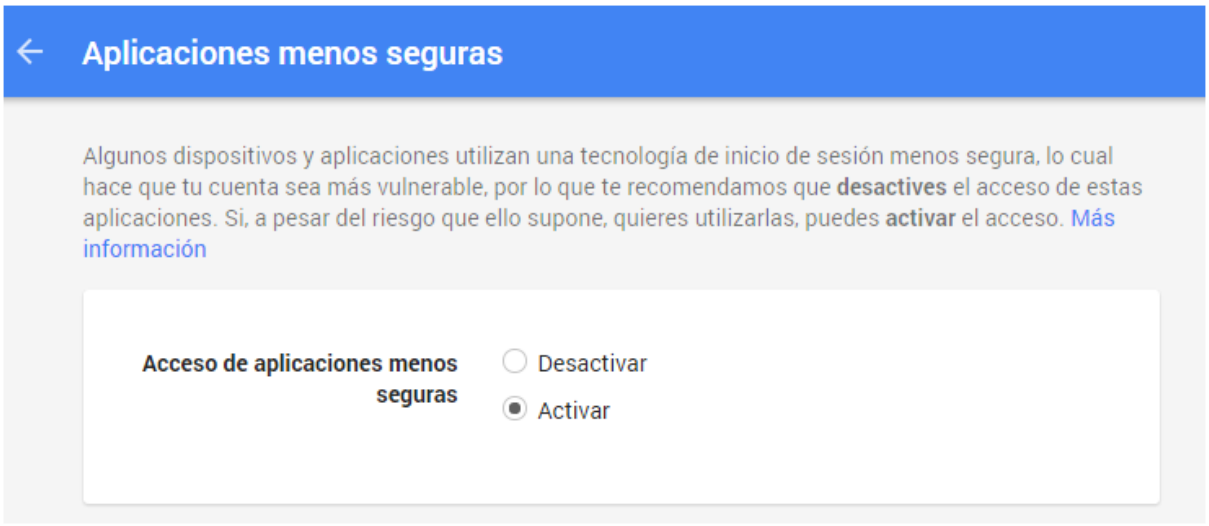

Imagen 4

Cambiando la configuración para permitir el acceso del programa al servidor, tendríamos todo listo para empezar a enviar las notificaciones.

El siguiente paso es escribir en los campos de EMAIL el mensaje que queremos enviar y quién es su destinatario. Por defecto nos encontramos la configuración que aparece en la *Imagen 5.*

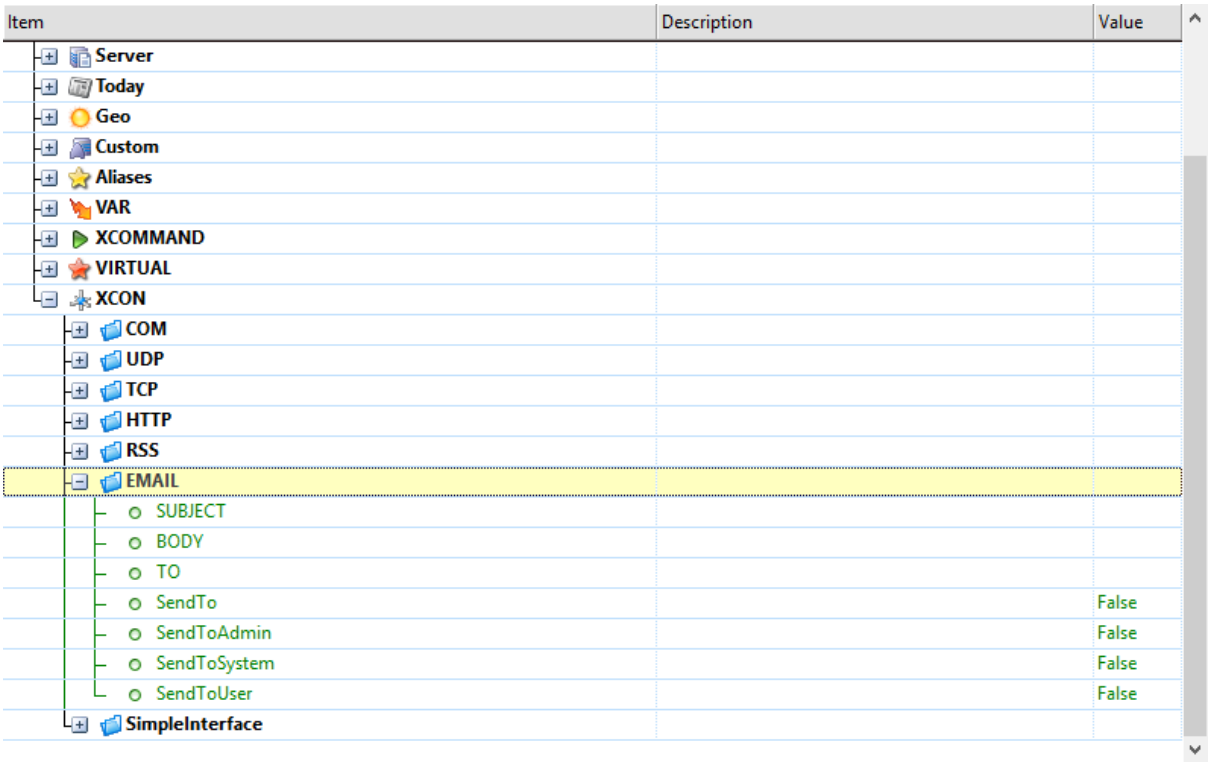

Imagen 5

- El campo SUBJECT es una cadena de caracteres donde se escribe el asunto del mensaje.

- El campo BODY es una cadena de caracteres donde se escribe el contenido del mensaje.
- El campo TO es una cadena de caracteres donde podemos escribir distintos emails para enviar el mensaje a varios destinatarios.
- El campo SendTo es un dato de tipo booleano, un 0 indica que no se hace nada mientras que un 1 manda la orden de enviar un email a los usuario definidos en TO.
- El campo SendToAdmin es un dato de tipo booleano, un 0 indica que no se hace nada mientras que un 1 manda la orden de enviar un email al usuario definido en la configuración como Admin (*Imagen 3*).
- El campo SendToAdmin es un dato de tipo booleano, un 0 indica que no se hace nada mientras que un 1 manda la orden de enviar un email al usuario definido en la configuración como System (*Imagen 3*).
- El campo SendToAdmin es un dato de tipo booleano, un 0 indica que no se hace nada mientras que un 1 manda la orden de enviar un email al usuario definido en la configuración como User (*Imagen 3*).

Para editar manualmente cada uno de los campos, bastará con hacer doble click con el ratón sobre el campo que queramos modificar. Finalmente para enviar el email debemos rellenar los campos SUBJECT y BODY. Después escribiremos un 1 en el campo correspondiente a la dirección de correo que queremos enviárselo. Si queremos enviarlo a varios destinatarios también es necesario rellenar el campo TO.

EJEMPLO DE ENVÍO DE EMAILS A TRAVÉS DE UN SCRIPT

Para entender el funcionamiento real del envío de mensajes vía email se ha escrito un script, en lenguaje de programación LUA, con un ejemplo de aplicación real. El programa describe de manera sencilla el funcionamiento, en caso de fallo, de un sistema con puerta de garaje automática.

Descripción del programa

El sistema de funcionamiento de una puerta de garaje automática se simula de manera sencilla con dos variables: una para el sensor de apertura-cierre y otra para el motor de funcionamiento de la puerta. Para poder utilizar el máximo de funciones del interfaz EMAIL se ha añadido un indicador que muestra al administrador si el sistema está funcionando correctamente o no (apertura indebida de caja del motor).

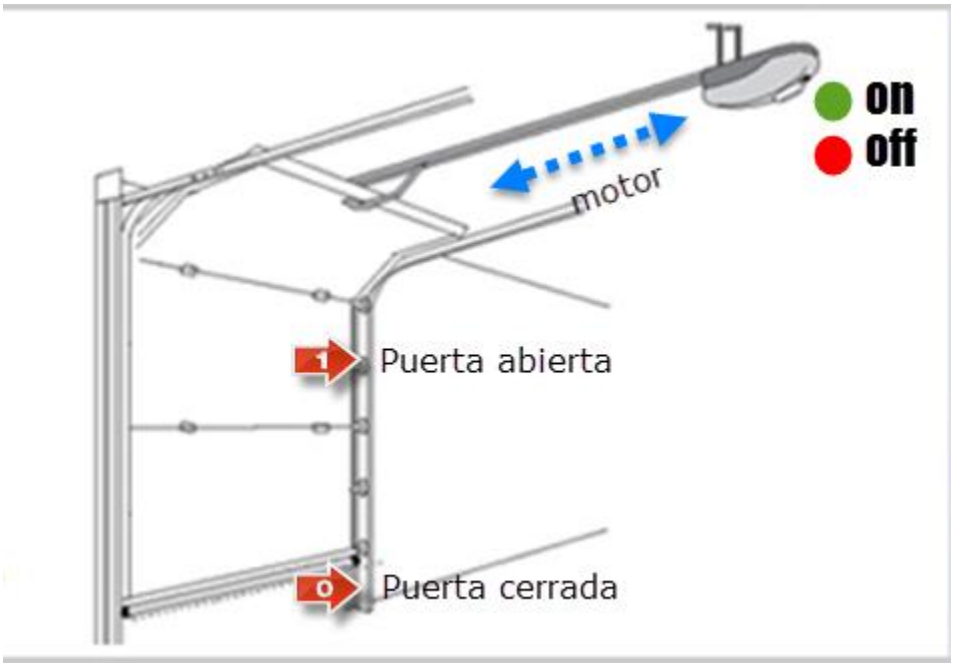

Imagen 6

Como se ve en la *Imagen 6*, cuando la puerta está cerrada, el sensor de la puerta vale 0. Para abrir la puerta se acciona el motor con sentido hacia la derecha, recogiendo la puerta. Así, una vez que el sensor de posición infrarrojo deja de estar activo, se convierte en un 1 lo que indica que la puerta está abierta. Al llegar al final del recorrido del motor, la puerta permanecerá abierta hasta que el vehículo termine de pasar o pasen 15 segundos. Entonces comenzará a cerrarse, sentido hacia la izquierda, hasta llegar a la posición final de puerta cerrada (sensor infrarrojo de nuevo en 0).

Si el sistema funciona bien no es necesario el envío de ningún email. Sin embargo, se han planteado tres tipos de problemas distintos por los que podría estar fallando:

1. Indicador de funcionamiento on-off, comunica al administrador del sistema si la caja del motor ha sido manipulada sin consentimiento.

2. Funcionamiento incorrecto del motor, el sentido de giro y el estado del sensor no son coherentes.
3. La puerta del garaje permanece abierta durante un largo periodo de tiempo.

En función del tipo de error, el sistema envía un email de notificación a un usuario distinto. 1-administrador, 2-técnico del sistema y 3-usuario o cliente.

Configuración de las variables en NETx BMS Studio

Para simular el funcionamiento del sistema hay que configurar algunos registros del NETx BMS Studio:

- Un Holding Register para simular el funcionamiento de un motor bidireccional (-500, 0, 500).
- Un Coil para simular el funcionamiento del sensor de puerta abierta-cerrada.
- Un Discrete Input para simular el funcionamiento del sensor de caja de motor abierta sin autorización.

Las tareas se definen en la pestaña que aparece en la *Imagen 7.*

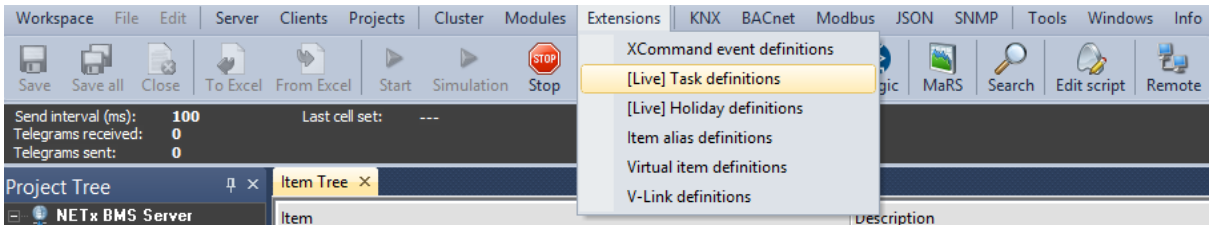

Imagen 7

El cambio de valor en las variables ModBus *NETx\XIO\Modbus\MB1\Coils\0 ó en NETx\XIO\Modbus\MB1\Holding Registers\1* puede afectar en dos funciones distintas (*mandoCorreoSystem()* y *mandoCorreoSystem()*), mientras que la variable *NETx\XIO\Modbus\MB1\Discrete Inputs\0* solo afecta a la función *mandoCorreoAdmin()*. Por claridad, se repiten las dos primeras variables en la definición de tareas, aunque también se podía haber incluido la llamada a ambas funciones en una misma fila.

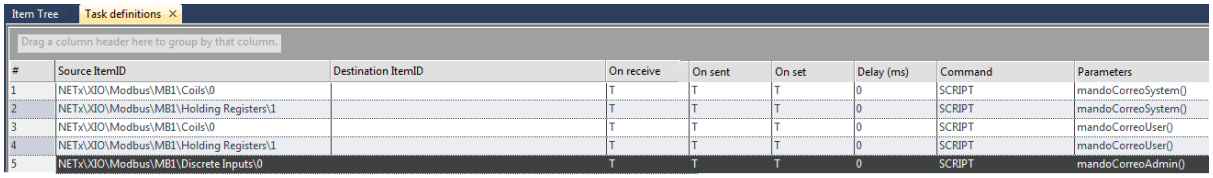

Imagen 8

Los valores de las variables que indican un funcionamiento correcto del sistema se muestran en la *Imagen 9.*

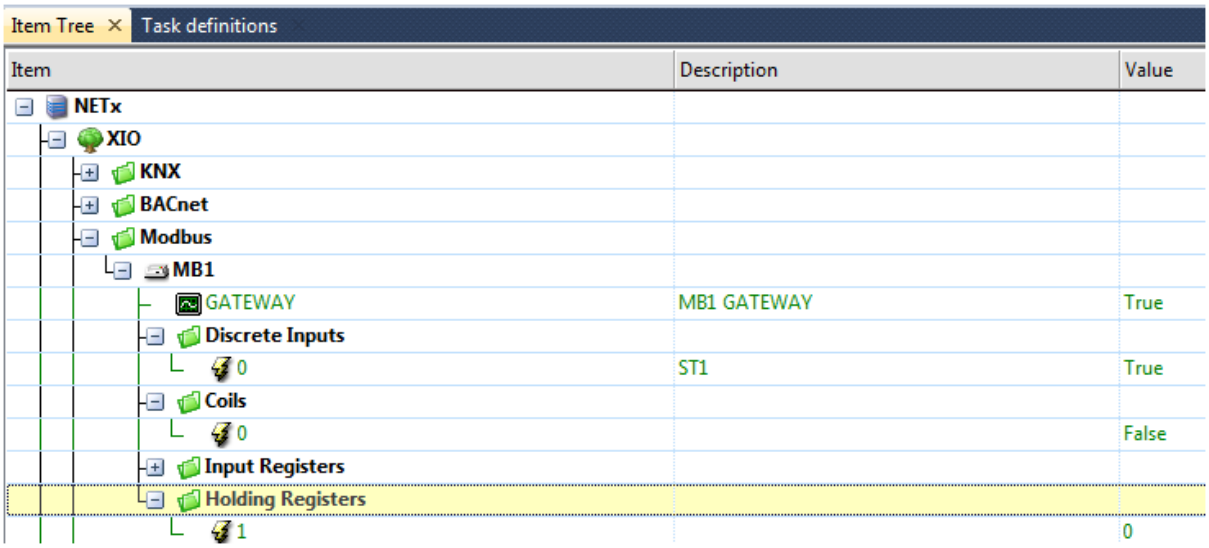

Imagen 9

ANEXO I - Código ejemplo Script LUA para envío de emails

A continuación se puede observar el código realizado en un fichero .lua que es leído y utilizado por el proyecto NETx BMS realizado:

```
--[[============================================================================

NETxAutomation Software GabH - all rights reserved
nxaScriptEngine is using LUA $ 1 scripting language (http://www.lua.org/)

--! @nxaEmailConf.lua
--! @brief Este script contiene funciones para el envío de correos a los diversos
                        tipos de usuarios del sistema NETx. Este sistema controla el uso de
                        una puerta de garaje configurado con tres puntos de control.

                1) La seguridad de la caja sellada del motor se realiza mediante el
                        punto NETx\XIO\Modbus\MB1\Discrete Inputs\0 que comunicará al Administrador
                        que ha sido manipulada indebidamente.

                2) Cuando el motor de la puerta de garaje funcione incorrectamente, esto
                es cuando:
                        - El motor siga intentando cerrar la puerta del garaje y está ya esté
                        completamente cerrada.
                        - El motor siga intentando abrir la puerta del garaje y está esté
                        completamente abierta.
                Esto se llevará a cabo con el uso de dos puntos:
                        NETx\XIO\Modbus\MB1\Coils\0 controla que puerta esté cerrada (false)
                                                                        ó abierta (true).
                        NETx\XIO\Modbus\MB1\Holding Registers\1 controla la velocidad del motor.
                                                                        > 0 está abriendo la puerta.
                                                                        < 0 está cerrando la puerta.
                Al suceder alguno de estos casos, se le comunicará al instalador del sistema
                para que pueda subsanar el fallo.

                3) En este caso, se le indica al usuario del sistema que la puerta del garaje
                se ha quedado abierta.
                Esto se controla con los dos puntos indicados en el caso anterior para cuando
                esté abierta y el motor no se esté moviendo (velocidad == 0).

--============================================================================]]

--[[============================================================================
--! @brief Funcion que manda un correo al Administrador cuando se dan las condiciones
                        del caso 1
--============================================================================]]
function mandoCorreoAdmin()
        nxa.SetValue("NETx.XCON.EMAIL.SUBJECT", "Notificación Centro de Control NETx")
        if(nxa.GetValue("NETx\\XIO\\Modbus\\MB1\\Discrete Inputs\\0") == false) then
                nxa.LogInfo('Mando correo Admin');
                nxa.SetValue("NETx.XCON.EMAIL.BODY", "Error en el direccionamiento del Motor de puerta garaje.
".."Apertura de caja de control de motor no autorizada")

xcon.SendAdminEmail(nxa.GetValue("NETx.XCON.EMAIL.SUBJECT"),nxa.GetValue("NETx.XCON.EMAIL.BODY"));
        end

end
```

```
--[[=========================================================================
--! @brief Funcion que manda un correo al Instalador cuando se dan las condiciones
                        del caso 2
--=========================================================================]]
function mandoCorreoSystem()
        nxa.SetValue("NETx.XCON.EMAIL.SUBJECT", "Notificación Centro de Control NETx")

        if(nxa.GetValue("NETx\\XIO\\Modbus\\MB1\\Coils\\0") == false
                and
                nxa.GetValue("NETx\\XIO\\Modbus\\MB1\\Holding Registers\\1") < 0) then
                        nxa.LogInfo('Mando correo System1');
                        nxa.SetValue("NETx.XCON.EMAIL.BODY", "Fallo de funcionamiento en puerta de garaje. Motor
activado con puerta cerrada")

        xcon.SendSystemEmail(nxa.GetValue("NETx.XCON.EMAIL.SUBJECT"),nxa.GetValue("NETx.XCON.EMAIL.BODY"));
        end
        if(nxa.GetValue("NETx\\XIO\\Modbus\\MB1\\Coils\\0") == true
                and
                nxa.GetValue("NETx\\XIO\\Modbus\\MB1\\Holding Registers\\1") > 0) then
                        nxa.LogInfo('Mando correo System2');
                        nxa.SetValue("NETx.XCON.EMAIL.BODY", "Fallo de funcionamiento en puerta de garaje. Motor
activado con puerta abierta")

        xcon.SendSystemEmail(nxa.GetValue("NETx.XCON.EMAIL.SUBJECT"),nxa.GetValue("NETx.XCON.EMAIL.BODY"));
        end


end

--[[=========================================================================
--! @brief Funcion que manda un correo al Usuario cuando se dan las condiciones
                        del caso 3
--=========================================================================]]
function mandoCorreoUser()
        nxa.SetValue("NETx.XCON.EMAIL.SUBJECT", "Notificación Centro de Control NETx")

        if(nxa.GetValue("NETx\\XIO\\Modbus\\MB1\\Coils\\0") == true
                and
                nxa.GetValue("NETx\\XIO\\Modbus\\MB1\\Holding Registers\\1") == 0) then
                        nxa.LogInfo('Mando correo User');
                        nxa.SetValue("NETx.XCON.EMAIL.BODY", "Puerta del garaje ABIERTA")

        xcon.SendUserEmail(nxa.GetValue("NETx.XCON.EMAIL.SUBJECT"),nxa.GetValue("NETx.XCON.EMAIL.BODY"));
        end


end
```