

Router Titan

Firmware V6

Actuando sobre registros Modbus mediante
comandos SMS

• Detalles del escenario

Los router Titan disponen de todas las funcionalidades típicas de un router 4G/3G/2G pero además cuentan con una serie de prestaciones adicionales que lo convierten en uno de los routers con más prestaciones del mercado.

Una de las prestaciones adicionales es la capacidad de poder ejecutar Scripts personalizados dentro del propio router. En esta nota de aplicación vamos a programar un pequeño script que permita la gestión de dispositivos modbus mediante simples comandos SMS. Para esta aplicación se va a utilizar el dispositivo Titan: Webdyn-Easy-Tunnel.

Escenario ejemplo: se dispone de un pequeño PLC que permite comunicaciones Modbus RTU a través de su puerto RS485. Ese PLC tiene varios sensores conectados, entre ellos una sonda de temperatura. El PLC también dispone de varias salidas de relé. Se pretende poder crear una aplicación de control a través de simples comandos SMS. Se necesita:

- 1.- Poder disponer de un número de teléfono especial de administrador del sistema
- 2.- Poder disponer de hasta 9 números de teléfono autorizados para poder enviar comandos SMS.
- 3.- El número de teléfono de administrador debe ser capaz de añadir o eliminar esos 9 números de teléfonos autorizados. Para poder añadir un número de teléfono autorizado, el teléfono de administrador deberá enviar un comando SMS con el texto "AUTH,x,yyyyyyyyyy" donde "x" es un número de 2 a 10 y "yyyyyyyyyy" es el número de teléfono a añadir a la lista de teléfonos autorizados. Un ejemplo de comando sería: "AUTH,2,+34666123456".
- 4.- El número de teléfono de administrador debe ser capaz de poder reiniciar el dispositivo Titan (esto es necesario, por ejemplo, para que una vez añadidos o modificados los números de teléfonos autorizados por SMS, el dispositivo Titan utilice la nueva configuración. El comando por SMS para reiniciar el dispositivo Titan será "REBOOT".
- 5.- Todos los números de teléfono deben poder enviar el comando por sms "TEMP". El dispositivo Titan, al recibir el comando "TEMP", debe leer del PLC (que tendrá dirección modbus @1 y cuyo puerto RS485 está configurado a 9600,8,N,1) el registro modbus @30005, que contendrá la temperatura. Dicho registro deberá ser dividido por 10 para obtener la temperatura en grados con 1 decimal. La temperatura deberá ser enviada con un mensaje SMS al número de teléfono que envió el mensaje "TEMP".
- 6.- Todos los números de teléfono deben poder enviar el comando por sms "GET,xxxx". Esto lo que permite es poder leer cualquier registro modbus del PLC, donde xxxx es la dirección del registro modbus a leer. Una vez leído el registro modbus, el dispositivo Titan enviará como respuesta un SMS con dicho registro al número de teléfono que envió el comando "GET,xxxx". Un comando de ejemplo podría ser "GET,30010" para leer el registro 30010.
- 7.- Todos los números de teléfono deben poder enviar el comando por sms "SET,xxxx,yyyy". Esto lo que permite es poder escribir en cualquier registro modbus del PLC, donde "xxxx" es la dirección de registro e "yyyy" el valor a escribir. El dispositivo Titan enviará como respuesta un SMS con el resultado de la escritura del registro al número de teléfono que envió el comando SMS. También enviará un mensaje SMS adicional al número de teléfono del administrador con el comando ejecutado así como con el número de teléfono que realizó la acción de cambio.



1. Configuración de la Mobile WAN.

En esta aplicación no será necesaria las comunicaciones IP, pues todo se va a realizar mediante comandos SMS. Por lo que la configuración de esta sección debe ser la de defecto de los dispositivos Titan, en modo Disabled (only SMS and CSD). Configuraremos el menú “Mobile > Basic Settings” de la siguiente manera, configurando también el APN, user y password si es necesario, pues las redes 4G lo precisan.

The screenshot shows the 'Mobile > Basic Settings' configuration page. The 'Mobile WAN' section is highlighted with a red box. The settings are as follows:

- Mobile WAN: Disabled (only SMS and CSC)
- Sim Mode: SIM1
- SIM1 APN: movistar.es
- SIM1 Username: MOVISTAR
- SIM1 Password: *****
- SIM1 Pin: (empty)
- SIM1 Auth: None

2. Configuración de las comunicaciones Serie RS485.

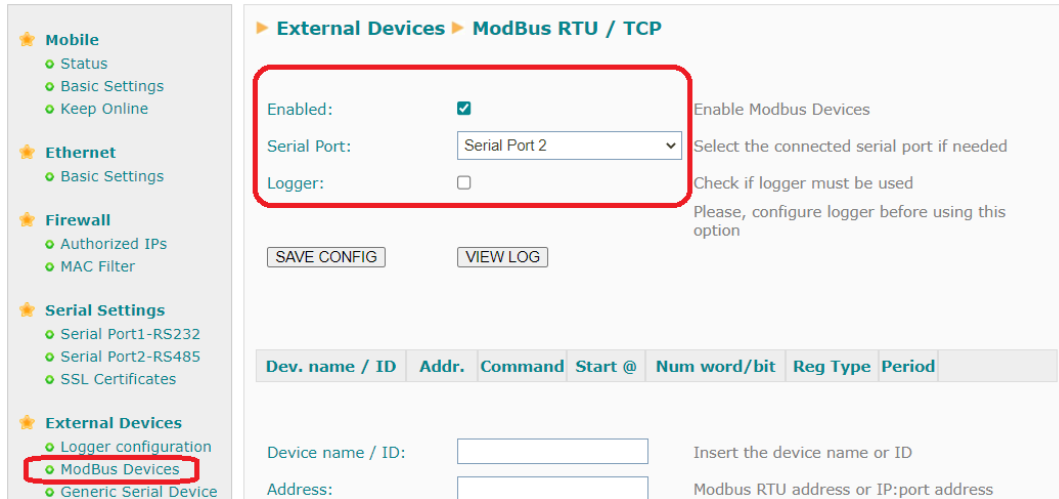
Tal y como se indicó en el inicio de esta nota de aplicación, el PLC dispone de comunicaciones serie RS485 a 9600,8,N,1, por lo que debemos configurar el puerto serie RS485 del dispositivo Titan con la misma configuración desde el menú “Serial Settings >Serial Port2-RS485”

The screenshot shows the 'Serial Gateway > Com2 Settings' configuration page. The 'Serial Port2-RS485' option in the left sidebar is highlighted with a red box. The main settings are as follows:

- Baudrate: 9600
- Data bits: 8
- Parity: none
- Stop bits: 1
- Timeout ms: 50
- Function: **Nothing or used by External Device or Script**

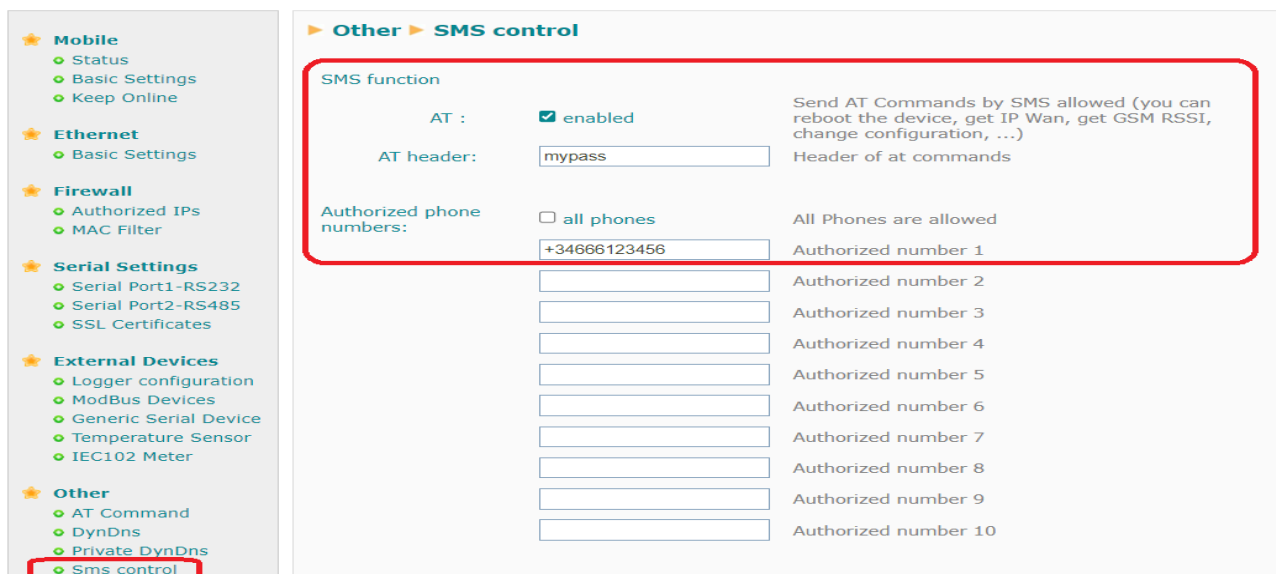
3. Configuración de las comunicaciones Modbus.

Las comunicaciones con el PLC van a ser vía Modbus a través del puerto RS485 del dispositivo Titan, por lo que debe asociarse el servicio Modbus con el puerto RS485 del Titan. Ello debe hacerse desde el menú “External Devices > Modbus Devices”:



4. Configuración SMS y del número de teléfono de administrador.

Debemos configurar el número de teléfono de administrador. Este teléfono será el que permita añadir después números de teléfonos autorizados a la lista. Es muy importante también establecer el “AT header” (podemos considerarlo como un password) y mantenerlo en secreto sólo para el administrador. De esa manera sólo el administrador podrá enviar comandos AT especiales al dispositivo Titan. En este ejemplo permitiremos la ejecución de comandos AT (con password) sólo a los números sea autorizados (sólo queremos que sean los indicados en la configuración). Es decir, la configuración de la sección “Other > SMS control” tendrá el siguiente aspecto, donde por ejemplo “+34666123456” es el número de teléfono de administrador y “mypass” el password para poder ejecutar comandos AT especiales sobre el dispositivo Titan.



5. Programación del Script.

Por último debemos programar el script que permita realizar la aplicación descrita. Esta programación se debe realizar desde el menú “Other > Titan Scripts”. Una vez programado el script debe seleccionarse la opción “Once after Power Up” para que el script se ejecute cada vez que se inicie el dispositivo Titan (por ejemplo, tras un reboot o un reinicio de alimentación)

The screenshot shows the configuration page for Titan Scripts. On the left, a sidebar lists various system settings, with 'Titan Scripts' selected. The main content area shows configuration options for a script. The 'Autostart' checkbox is checked, and the 'Period' dropdown menu is set to 'Once after Power Up'. Below these options is a 'SAVE CONFIG' button. The breadcrumb navigation at the top of the main area reads 'Other > Titan Scripts v2 > Options'.

El script lo programaremos en la siguiente sección:

The screenshot shows the script editor for a Titan Script. The left sidebar has 'Titan Scripts' selected. The main area displays a code editor with the following JavaScript code:

```
3 mtx.println("Running...");
4
5 //***** FUNCTION: returns 'true' if is admin telephone number
6 function isAdminNumber(number)
7 {
8     if (number==authNumbers[0])
9         return true;
10    else
11        return false;
12 }
13
14
15 //***** FUNCTION: init list with authorized telephone numbers
16 function initAuthNumbers()
17 {
18     //***** READ THE 10 AUTHORIZED PHONE NUMBERS FROM TITAN CONFIGURATION (from MENU 'Other > SM
19     for (var i=0;i<=9;i++)
20     {
21         var res=mtx.configParamGet("SMS_VALIDPHONE" + (i+1));
22         if (res!=null)
23         {
24             authNumbers[i]=res;
25             mtx.println("Authorized phone number " + (i+1) + ": " + authNumbers[i]);
26         }
27     }
28 }
29
30
31 //***** INIT CONFIGURATION
32 initAuthNumbers();
33
34 //***** INFINITE LOOP
35 while (true)
36 {
37
38
```

Below the code editor are buttons for 'Run Script', 'Stop Script', 'Save Script', 'Delete Script', and 'Encrypt Script'. The breadcrumb navigation at the bottom of the main area reads 'Other > Titan Scripts v2 > Console'.

El código de ejemplo del script podría ser el siguiente (úselo y modifíquelo bajo su propia responsabilidad):

```
var authNumbers=[];
mtx.println("Running...");

//***** FUNCTION: init list with authorized telephone numbers
function initAuthNumbers()
{
//***** READ THE 10 AUTHORIZED PHONE NUMBERS FROM TITAN CONFIGURATION (from MENU
'Other > SMS control')
    for (var i=0;i<=9;i++)
    {
        var res=mtx.configParamGet("SMS_VALIDPHONE" + (i+1));
        if (res!=null)
        {
            authNumbers[i]=res;
            mtx.println("Authorized phone number " + (i+1) + ": " + authNumbers[i]);
        }
    }
}

//***** FUNCTION: returns 'true' if is admin telephone number
function isAdminNumber(number)
{
    if (number==authNumbers[0])
        return true;
    else
        return false;
}
```

```

//***** INIT CONFIGURATION
initAuthNumbers();

//***** INFINITE LOOP
while (true)
{
    var smsMessage=mtx.smsRead()

    //***** IF THERE IS A NEW SMS ...
    if (smsMessage!=null)
    {
        mtx.println("Received SMS. Number: " + smsMessage[0] + ". Message : " + smsMessage[1]);

        var messageToken=smsMessage[1].split(",");

        //***** REBOOT COMMAND
        if (messageToken[0].toUpperCase()=="REBOOT")
        {
            //***** ONLY ADMIN CAN EXECUTE THIS COMMAND
            if (isAdminNumber(smsMessage[0]))
                mtx.atSend("AT^MTXTUNNEL=REBOOT", 2);
        }

        //***** EXAMPLE, "AUTH,2,+34666223344" ADD/CHANGE THE AUTH NUMBER 2 TO
        //***** +34666223344
        else if (messageToken[0].toUpperCase()=="AUTH")
        {
            //***** ONLY ADMIN CAN EXECUTE THIS COMMAND
            if (isAdminNumber(smsMessage[0]))
                var res=mtx.configParamSet("SMS_VALIDPHONE" + messageToken[1], messageToken[2]);
        }
    }
}

```

```

//***** READ TEMPERATURE COMMAND
else if (messageToken[0].toUpperCase()=="TEMP")
{
//***** READING REGISTER 5 (TEMPERATURE)
    var res=mtx.modbusRTUGetWords(1,3,30005,1);
    //***** IF THE MODBUS READING WAS CORRECT ...
    if (res!==null)
    {
        //***** SHOW THE MODBUS VALUE BY CONSOLE
        mtx.println("Temperature: " + res[0]/10);
var smsOUT=mtx.smsSend(smsMessage[0],"TEMPERATURE: " + res[0]/10 );
    }
}

//***** READ MODBUS REGISTER COMMAND. EXAMPLE COMMAND: "GET,30000"
else if (messageToken[0].toUpperCase()=="GET")
{
    var res=mtx.modbusRTUGetWords(1,3,messageToken[1],1);
    //***** IF THE MODBUS READING WAS CORRECT ...
    if (res!==null)
    {
        mtx.println("The value of register is:" + res[0]);
//***** SENDING A SMS WITH THE READ VALUE
        var smsOUT=mtx.smsSend(smsMessage[0],"Value of register " + messageToken[1] + ":"
+ res[0]);
    }
}

//***** SET MODBUS COMMAND. EXAMPLE COMMAND: "SET,30005,12"
else if (messageToken[0].toUpperCase()=="SET")
{

```



```

modbusRTUSetWords(1,6,parseInt(messageToken[1]),[parseInt(messageToken[2])]);

                                                                    var      res=mtx.

//***** IF THE MODBUS WRITE WAS CORRECT ...
if (res)
{
mtx.println("The value of register " + parseInt(messageToken[1]) + " changed to value: " +
parseInt(messageToken[2]));

//***** SENDING A SMS TO USER WITH THE RESULT

var smsOUT=mtx.smsSend(smsMessage[0],"Register " + messageToken[1] + " changed to " +
parseInt(messageToken[2]) + " --> OK");

//***** SENDING A SMS TO ADMIN WITH THE ACTION AND RESULT

var smsOUTAdmin=mtx.smsSend(authNumbers[0],"Register " + messageToken[1] + " changed to " +
parseInt(messageToken[2]) + " by " + smsMessage[0] + " --> OK");

}
}

}

//***** 1 SECOND PAUSE
mtx.pause(1000);
}

```

Ahora describiremos más algunas partes del script:

Esta función `initAuthNumbers()` lo que hace leer y almacenar en el array `authNumbers[]` la lista de teléfonos autorizados. Estos teléfonos autorizados son los que se encuentran en la sección "Other >SMS control" del dispositivo Titan. Para ello se leen los parámetros de configuración `SMS_VALIDPHONE1 ... SMS_VALIDPHONE10`. Consideraremos que el teléfono del parámetro `SMS_VALIDPHONE1` es el teléfono de administrador.

```

1  var authNumbers=[];
2
3  mtx.println("Running...");
4
5  /****** FUNCTION: init list with authorized telephone numbers
6  function initAuthNumbers()
7  {
8      /****** READ THE 10 AUTHORIZED PHONE NUMBERS FROM TITAN CONFIGURATION (from MENU 'Other > SM
9      for (var i=0;i<=9;i++)
10     {
11         var res=mtx.configParamGet("SMS_VALIDPHONE" + (i+1));
12         if (res!=null)
13         {
14             authNumbers[i]=res;
15             mtx.println("Authorized phone number " + (i+1) + ": " + authNumbers[i]);
16         }
17     }
18 }
19

```

La función isAdminNumber básicamente devuelve “true” si el número de teléfono pasado como parámetro es el del administrador. Esta función se utilizará para permitir ejecutar aquellos comandos por SMS que sólo debe poder ejecutar el administrador.

```

20 /****** FUNCTION: returns 'true' if is admin telephone number
21 function isAdminNumber(number)
22 {
23     if (number==authNumbers[0])
24         return true;
25     else
26         return false;
27 }
28
29
30 /****** INIT CONFIGURATION
31 initAuthNumbers();
32

```

Esta sección muestra el bucle principal del script. Continuamente se está comprobando si ha llegado un nuevo mensaje SMS. En caso de haber llegado un SMS nuevo en smsMessage[0] tendremos el número de teléfono que envió el SMS y en smsMessage[1] tendremos el texto del mensaje SMS. Nótese que el mensaje sms (smsMessage[1]) se divide en tokens con el método split(). Por ejemplo, si llega el mensaje SMS: SET,30005,12 en el array messageToken, tras ejecutar el comando var messageToken=smsMessage[1].split(","); tendremos en messageToken[0] el valor “SET” (es decir, el comando), en messageToken[1] el valor 30005 y en messageToken[2] el valor 12.

En el fragmento mostrado, puede verse que si se recibe el comando “REBOOT” y el número que ha enviado el SMS es el número de administrador, se ejecuta internamente el comando AT^MTXTUNNEL=REBOOT que reiniciará el dispositivo Titan.

```

33
34 /****** INFINITE LOOP
35 while (true)
36 {
37     var smsMessage=mtx.smsRead()
38
39     /****** IF THERE IS A NEW SMS ...
40     if (smsMessage!=null)
41     {
42         mtx.println("Received SMS. Number: " + smsMessage[0] + ". Message : " + smsMessage[1]);
43
44         var messageToken=smsMessage[1].split(",");
45
46         /****** REBOOT COMMAND
47         if (messageToken[0].toUpperCase()=="REBOOT")
48         {
49             /****** ONLY ADMIN CAN EXECUTE THIS COMMAND
50             if (isAdminNumber(smsMessage[0]))
51                 mtx.atSend("AT^MTXTUNNEL=REBOOT", 2);
52         }
53     }
54 }

```

En el siguiente fragmento, puede verse que si se recibe el comando SMS “AUTH” , y este mensaje SMS ha sido enviado por el número de teléfono del administrador, se añadirá/modificará un nuevo teléfono autorizado. Por ejemplo, si se recibe el comando SMS: AUTH,2,+34666223344 se modificará el parámetro de configuración del Titan SMS_VALIDPHONE2 con el valor +34666223344 Si tras el envío del SMS consulta el menú “Other > SMS Control” verá el teléfono añadido. Recuerde que deberá enviar un comando SMS “REBOOT” para que el dispositivo Titan tome la nueva configuración.

```

//***** EXAMPLE, "AUTH,2,+34666223344" ADD/CHANGE THE AUTH NUMBER 2 TO +34666223
else if (messageToken[0].toUpperCase()=="AUTH")
{
    //***** ONLY ADMIN CAN EXECUTE THIS COMMAND
    if (isAdminNumber(smsMessage[0]))
    {
        var res=mtx.configParamSet("SMS_VALIDPHONE" + messageToken[1], messageTok
    }
}

```

El siguiente fragmento de código permite ejecutar el comando “TEMP” para leer la temperatura:

Al recibirse el comando TEMP, se ejecuta el comando para leer del PLC con dirección @1 y usando el comando modbus 3, el registro modbus 30005. Si la lectura modbus es correcta, se devuelve un mensaje al número de teléfono que envió el mensaje (que está almacenado en la variable smsMessage[0]) el valor del registro leído (almacenado en res[0]) dividido por 10 para tener el valor de temperatura en decimal.

```

//***** READ TEMPERATURE COMMAND
else if (messageToken[0].toUpperCase()=="TEMP")
{
    //***** READING REGISTER 5 (TEMPERATURE)
    var res=mtx.modbusRTUGetWords(1,3,30005,1);
    //***** IF THE MODBUS READING WAS CORRECT ...
    if (res!==null)
    {
        //***** SHOW THE MODBUS VALUE BY CONSOLE
        mtx.println("Temperature: " + res[0]/10);
        var smsOUT=mtx.smsSend(smsMessage[0],"TEMPERATURE: " + res[0]/10 );
    }
}

```

El siguiente comando es “GET”. Por ejemplo, si se recibe el comando sms GET,30000 , como vimos anteriormente al explicar el método .split(), en messageToken[1] tendremos dicho valor 30000. Por lo que se procede a leer dicho registro y a enviarlo posteriormente por SMS al número de teléfono que envió el comando.

```

//***** READ MODBUS REGISTER COMMAND. EXAMPLE COMMAND: "GET,30000"
else if (messageToken[0].toUpperCase()=="GET")
{
    var res=mtx.modbusRTUGetWords(1,3,messageToken[1],1);
    //***** IF THE MODBUS READING WAS CORRECT ...
    if (res!=null)
    {
        mtx.println("The value of register is:" + res[0]);
        //***** SENDING A SMS WITH THE READ VALUE
        var smsOUT=mtx.smsSend(smsMessage[0],"Value of register " + messageToken[1] + " is " + res[0]);
    }
}

```

Por último, el script gestiona el comando "SET", que permite cambiar el valor de un registro modbus del PLC, en este caso utilizando el comando modbus 6.

Si la acción de escritura modbus es correcta, pues se envía el resultado de la acción al teléfono que envió el comando SMS y también al número de teléfono del administrador.

```

88 //***** SET MODBUS COMMAND. EXAMPLE COMMAND: "SET,30005,12"
89 else if (messageToken[0].toUpperCase()=="SET")
90 {
91     var res=mtx.modbusRTUSetWords(1,6,parseInt(messageToken[1]),[parseInt(messageToken[2]),parseInt(messageToken[3]),parseInt(messageToken[4]),parseInt(messageToken[5]),parseInt(messageToken[6]),parseInt(messageToken[7])]);
92
93     //***** IF THE MODBUS WRITE WAS CORRECT ...
94     if (res)
95     {
96         mtx.println("The value of register " + parseInt(messageToken[1]) + " changed to " + res);
97
98         //***** SENDING A SMS TO USER WITH THE RESULT
99         var smsOUT=mtx.smsSend(smsMessage[0],"Register " + messageToken[1] + " changed to " + res);
100
101         //***** SENDING A SMS TO ADMIN WITH THE ACTION AND RESULT
102         var smsOUTAdmin=mtx.smsSend(authNumbers[0],"Register " + messageToken[1] + " changed to " + res);
103     }
104 }
105

```

¿Más dudas?

Escríbanos sus consultas a iotsupport@mtxm2m.com