

## Estudios previos

Antes de abordar la construcción final de la aplicación, y teniendo en cuenta los requisitos funcionales que se habían marcado para el producto final, se realizaron en paralelo una serie de trabajos previos en investigación, diseño, creación de herramientas software, etc.

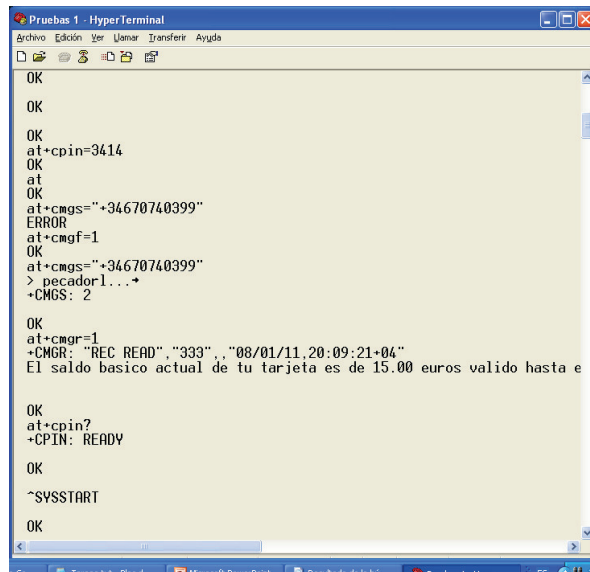
### Control del modem desde el hiperterminal

El primer paso para el desarrollo de la aplicación para el PLC consistió en conocer a fondo el protocolo de intercambio de información con el modem, mediante los comandos AT. Este estudio se realizó prescindiendo del PLC, y conectando el modem directamente a la aplicación hiperterminal del PC, que permite el acceso directo al puerto serie. De esta forma el puerto serie del modem queda directamente enlazado a la pantalla y teclado del ordenador.

El objetivo de este estudio fue primero conocer la lista de comandos para configurar el modem exactamente según las necesidades y para procesar el envío y recepción de SMSs de una forma asumible por el PLC. Uno de los objetivos principales en esta fase fue alcanzar una configuración del modem en el que todas las comunicaciones pudieran ser siempre iniciativa de la parte del PLC (por ejemplo ante la llegada de un nuevo SMS al modem, su configuración por defecto es enviar una trama de aviso, pero se buscó la opción de que las consultas al buffer de entrada se hicieran siempre a petición del controlador).

Un segundo objetivo fue conocer la familia de respuestas a los comandos utilizados. La más común de todas, especialmente a los comandos de configuración, es "OK".

Un tercer objetivo fue conocer el comportamiento del modem ante fallos, incidencias y errores, para generar en el PLC un driver de comunicaciones robusto y capaz de reponerse de forma autónoma a respuestas de error y a cualquier eventualidad.



```
OK
OK
OK
at+cpin=3414
OK
at
OK
at+cmgs="+34670740399"
ERROR
at+cmgf=1
OK
at+cmgs="+34670740399"
> pecador1...+
+CMGS: 2

OK
at+cmgr=1
+CMGR: "REC READ", "333", "08/01/11,20:09:21+04"
El saldo basico actual de tu tarjeta es de 15.00 euros valido hasta e

OK
at+cpin?
+CPIN: READY

OK
^SYSSTART
OK
```

Por último se estudió el tiempo de respuesta a los comandos del proceso, para establecer de forma correcta la gestión de tiempos del driver de comunicaciones.

## Desarrollo del driver de comunicaciones

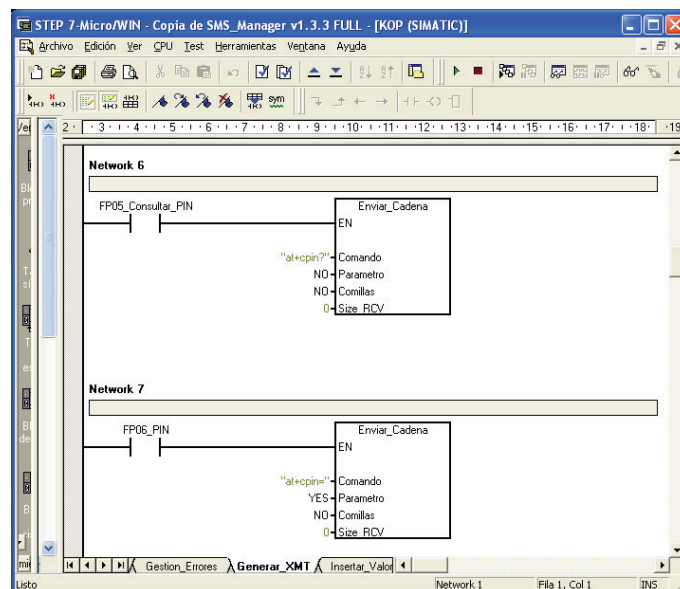
El driver de comunicaciones es una pieza del código del PLC que se desarrolló para facilitar de forma completamente transparente al resto del proyecto una funcionalidad de diálogo a través del puerto del PLC, en protocolo ASCII. Este driver es transparente al usuario final de la aplicación, es decir, está concebido para facilitar la programación a nivel interno para esta y para otras librerías de comunicación entre un PLC y un módem.

El driver se contempla como una herramienta de comando y respuesta para gestión dentro de una máquina de estados, de forma que en cada estado se lanza de salida una cadena de texto como comando (o consulta), a continuación se pasa a modo de escucha, y en función de la respuesta recibida la máquina cambia a otro estado, donde si procede se puede iniciar de nuevo el proceso de comunicación.

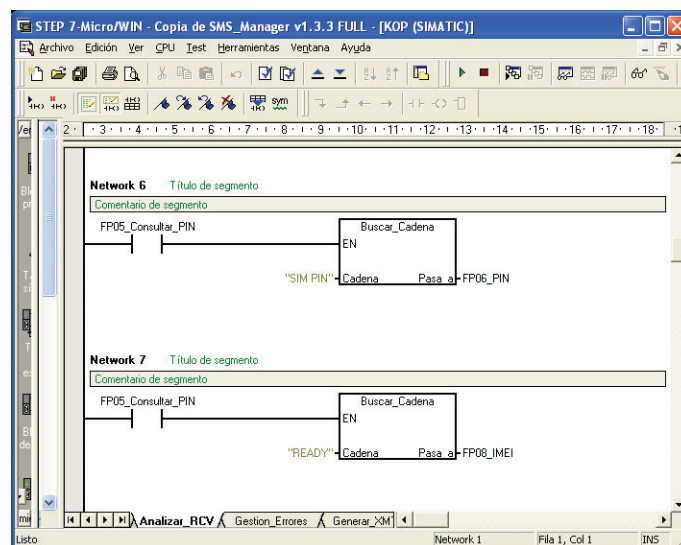
El driver gestiona eventos asíncronos de la comunicación (fin de mensaje saliente, nuevo mensaje entrante, etc.) pero ofrece a la capa superior una interfaz síncrona que facilita su uso en la librería. En cada ciclo de comunicación internamente el driver pasa por los siguientes estados: reposo, mensaje de salida en marcha (síncrono por evento de programación), mensaje de salida finalizado y transición a modo de escucha (asíncrono por evento del puerto de comunicaciones), mensaje recibido (asíncrono por evento del puerto de comunicaciones), cuerpo del mensaje recibido listo para análisis (síncrono e interno al driver), fin del análisis (síncrono por evento de programación), y de nuevo a reposo.

Este driver ofrece a la capa superior funcional un conjunto de dos funciones, una para realizar un envío de una cadena y otra para detectar qué respuesta se ha recibido. Esto permite que desarrollar muy fácilmente cualquier nuevo protocolo de relación con el módem, enviando comandos y analizando respuestas.

La función de envío de cadena se llama “Enviar\_Cadena” y tiene como parámetros el texto a enviar, una variable booleana para indicar si hay o no que concatenar un parámetro en el comando (debe ser colocado con anterioridad en una posición predeterminada de la librería), otra variable para indicar si el parámetro debe concatenarse entrecomillado, y finalmente un parámetro para expresar el tiempo máximo que requiere la respuesta en recibirse completa una vez que comienza a recibirse (por ejemplo la cadena con el grado de cobertura requiere más de los 500 ms previstos por defecto).



La función de análisis de la respuesta se llama “Buscar\_Cadena” y tiene como parámetro de entrada una palabra que hay que encontrar en la respuesta y como parámetro de salida el bit que activa el nuevo estado al que se pasa si la búsqueda ha resultado positiva. Dentro de un estado se puede llamar varias veces a la función “Buscar\_cadena”, en caso de que se puedan esperar varias respuestas.



El driver también indica a la capa superior funcional si hay un error por timeout en la respuesta es decir, no ha habido respuesta al último comando) o si la respuesta recibida no coincide con ninguna de las que se han previsto. Como lo que prima es la robustez del sistema, y dadas también las limitaciones de un sistema PLC, en general ante cualquier imprevisto en las comunicaciones la opción es reiniciar el modem y reconfigurarlo todo de nuevo antes de retomar la aplicación.

## Diseño de la máquina de estados

La máquina de estados es la arquitectura ideal para gestionar una aplicación de este tipo. Permite trabajar de forma muy ordenada sobre un proceso complejo y lleno de eventos asíncronos. Los tipos de procesos que se han aplicado a estados particulares son:

- **Realizar un ciclo completo de comando y respuesta con el modem.** En función de la respuesta se hace la nueva transición a un estado u otro.
- **Realizar un periodo de espera.** Por ejemplo, después de cargar el PIN se pasa a un estado que consiste en esperar 10 segundos antes de continuar con la configuración del modem.
- **Realizar un procesado de la respuesta del modem.** Por ejemplo, al solicitar el nivel de cobertura la cadena recibida es compleja y contiene mucha información adicional. Por tanto tras este ciclo de comunicación se pasa a un estado en el que lo que se hace es analizar la cadena recibida (que permanece en el buffer de recepción) para extraer el dato de cobertura.
- **Realizar un procesado para generar un comando de salida al modem.** Por ejemplo, cuando se está enviando un SMS de salida, se realizan dos ciclos de comunicación con el modem, uno para indicar el teléfono al que se envía el SMS, y un segundo con el texto. Entre ambas fases de comunicación (2 estados) existe un tercer estado en el que se construye la cadena de texto.
- **Reposo.** En el estado de reposo no se realizan acciones concretas, sino que se monitorizan de manera continua los eventos que lanzan procesos, como por ejemplo, enviar un nuevo SMS, consultar los SMSs entrantes, etc.

Utilizando estas 4 acciones, más la gestión del estado de reposo, la máquina de estados permite construir la aplicación como una secuencia controlada. Desde el punto inicial se realiza una larga cadena secuencial de pasos que

tiene como objeto inicializar y configurar el modem, y que una vez que termina da paso al estado de reposo. Desde el estado de reposo se pueden lanzar, de forma ordenada, determinados procesos. Todos ellos vuelven al estado de reposo una vez finalizados.

Si se encuentra una nueva orden de salida en el buffer de mensajes salientes, se arranca la secuencia de estados que envían un SMS.

Si vence el temporizador de lectura entrante, se arranca la secuencia de estados que consultan si hay nuevos mensajes en el buffer de SMS entrantes del modem, y si la respuesta es positiva se lee el mensaje y se analiza.

Si se ha solicitado un paso a modo teleservicio, se arranca la secuencia de estados que modifica la configuración del puerto para pasarlo de modo freeport a modo PPI, y a continuación se detiene la aplicación de gestión de SMSs.