LIBSTARGATE:
Programación de clientes para acceso a los Stargates

Juan González Gómez <juan@iearobotics.com>

20/oct/2003

1. Introducción

En este documento se describe brevemente la librería **libstargate**, que permite el acceso a los servicios básicos (PING e IDENTIFICACIÓN) implementados en todos los StarGates convencionales.

También se encuentran implementados los servicios LOAD y STORE del servidor GENERICO.

Mediante una extensión, la **libstargateh**, se pueden crear clientes con diferentes hilos que tengan acceso al Stargate.

La versión documentada aquí es la 1.0.

2. Arquitectura

La arquitectura de la librería se puede ver en la figura 1. Está constituida por dos módulos:

- *sg-serial*: Módulo para comunicaciones serie
- *sg-tramas*: Módulo para construir las tramas para la comunicación con los servidores NULO y GENERICO. Los servicios de estos servidores son accesibles mediante las funciones de interfaz.

3. Módulo sg-serial

3.1. Descripción

Módulo de comunicaciones serie. Se configura el puerto según lo establecido en el proyecto Stargate: 9600 baudios, 8 bits de datos, 1 bit de stop y sin paridad (8N1).

3.2. Interfaz

- ***int sg_serial_open(char *disp):*** Abrir el puerto serie y configurarlo adecuadamente para comunicarse con un Stargate.
  - *disp*: Cadena con el dispositivo serie. Por ejemplo "/dev/ttyS0" para el COM1 ó "/dev/ttyS1" para el COM2.
  - *Devuelve*: El descriptor del puerto serie o -1 si ha ocurrido algún error
- ***void sg_serial_enviar(int serial_fd, char *cadena, int tam)***: Enviar un bloque de caracteres por el puerto serie.
  - *serial_fd*: Descriptor del puerto serie
  - *cadena*: Puntero al bloque de caracteres a enviar
  - *tam*: Tamaño del bloque a enviar

Para recibir datos desde el puerto serie se usa la llamada al sistema estándar: *read()*.

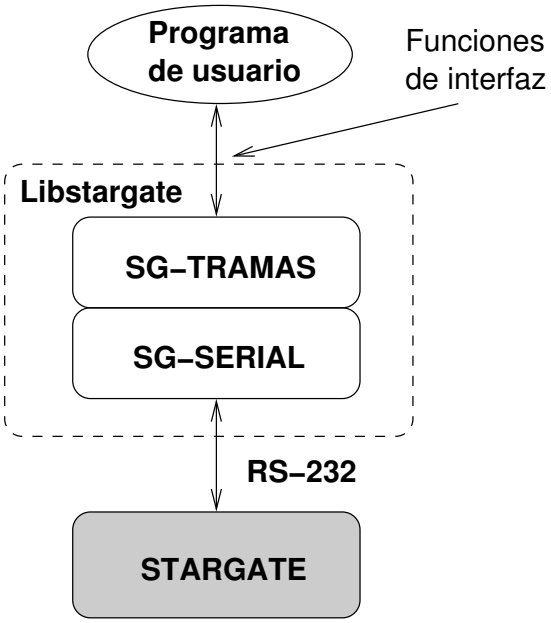

Figura 1: Arquitectura de la librería libstargate

4. Módulo sg-tramas

4.1. Descripción

Este es el módulo principal, que crea las diferentes tramas y las envía al Stargate a través del puerto serie. Además incluye algunas definiciones y funciones de apoyo para hacer más sencilla la implementación de los clientes.

4.2. Interfaz

- ***void sg_tramas_init(int fd);*** Inicialización del módulo. Se le pasa como parámetro el descriptor del puerto serie a emplear. Este descriptor, normalmente, será el que se obtiene al abrir el puerto serie con la función *sg_serial_open()*.
- ***int sg_ping(void);*** Solicitar el servicio ping. Todas las funciones de acceso a los servicios devuelven los siguientes valores:
  - -1: Ha ocurrido algún error
  - 0: Error de TIMEOUT. El StarGate NO responde
  - 1: OK. Se ha accedido correctamente al servicio
- ***int sg_id(byte *is, byte *im, byte *ipv);*** Solicitar servicio de Identificación. El valor devuelto por la función indica lo mismo que la del servicio PING. Los parámetros que devuelve son:
  - *is*: Identificador del servidor
  - *im*: Identificador del microcontrolador
  - *ipv*: Identificación de la placa y la versión del servidor
- ***int sg_store(int dir, int valor);*** Servicio STORE del Stargate Genérico. Se almacena el valor en la dirección indicada. La función devuelve los mismos valores que en el caso del servicio PING.

- *dir*: Dirección (0x0000-0xFFFF) a la que acceder
- *valor*: Dato que almacenar en la dirección indicada

- ***int sg_load(int dir, int *valor);*** Servicio LOAD del Stargate Genérico. Se lee el valor que se encuentra almacenado en la dirección dir. Se devuelvel los mismos valores que en el caso del servicio PING. Los parámetros son:

  - *dir*: Dirección (0x0000-0xFFFF) de la que leer
  - *valor*: Dato que se encuentra en esa posición

- ***void sg_tramas_is_to_string(byte is, char *cad);*** Convertir la identifictación del servidor (is) en una cadena. Por ejemplo, en el caso del haberse comunicado con el servidor genérico, el valor is se obteiene mediante el servicio de identificación y despues, mediante la llamada a esta función se obtendría la cadena "GENERIC".

- ***void sg_tramas_im_to_string(byte im, char *cad);*** Convertir la identificación del microcontrolador (im) en una cadena. El el caso de que el StarGate esté implementada en un 68hc11e2, se obtendría la cadena "6811E2"

- ***void sg_tramas_ipv_to_string(byte im, byte ipv, char *cad);*** Convertir el par de valores *im* e *ipv* en una cadena. Estos valores determinan el tipo de placa que se está empleando.

5. Libstargateh: manejo de hilos

5.1. Descripción

En muchas aplicaciones con utilizar la librería *libstargate* es suficiente. En otras puede ser interesante utilizar hilos (threads). Por ejemplo, se puede tener un hilo haciendo PINGs al Stargate, para detectar si está operativo. Otros hilos pueden necesitar acceder a otros servicios.

Es necesaria una manera de poder coordinar todas las peticiones al Stargate. Para ello se tiene que crear un hilo monitor que gestione los acceso de los hilos clientes. Este hilo se denomina motor.

En la figura 2 se muestra esta idea gráficamente. Todas las peticiones de los clientes son recogidas por el hilo motor, garantizando que sólo un hilo tenga acceso en cada momento al puerto serie.

5.2. Arquitectura

Con el propósito de manejar hilos, se ha creado la librería *libstargateh*, que es una extensión de la *libstargate* que incluye el motor, como se muestra en la figura 3.

5.3. Interfaz

En esta librería, además de tener acceso a las funciones de la librería libstargate, los hilos tienen acceso a los servicios de Stargate usando las siguientes funciones:

- ***int sgm_id(byte *is, byte *im, byte *ipv);*** Acceso al servicio de identificación

- i***nt sgm_ping(void);*** Acceso al servicio de PING

Son similares a las funciones de acceso a los servicios de la libstargate, pero con el **prefijo sgm** en vez de **sg**.

Además es necesario inicializar el motor. Esto se hace con la función:

- ***void sgm_motor_init(int fd);***

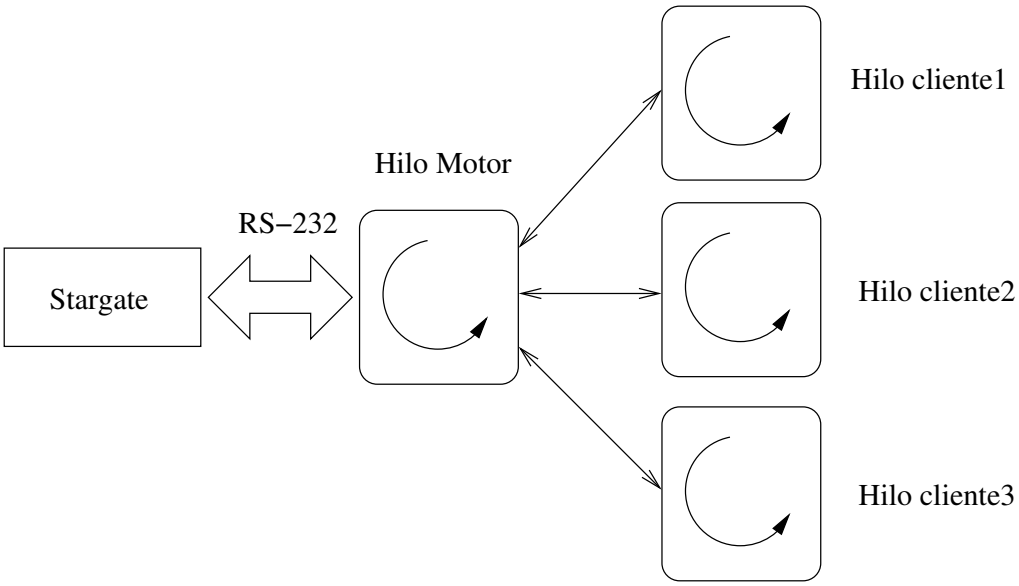

Figura 2: Arquitectura para el acceso de difetentes hilos al Stargate

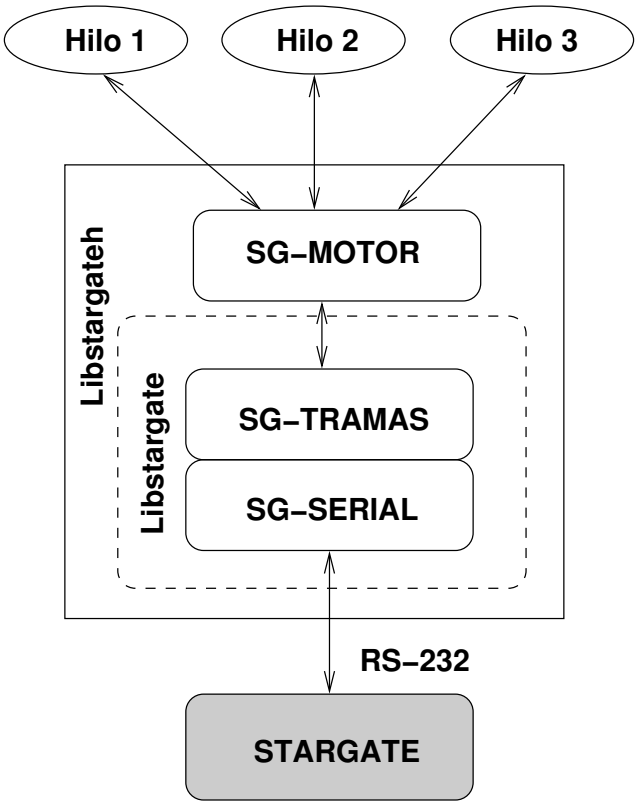

Figura 3: Esquema de la librería libstargateh

6. Ejemplos

6.1. sgc-null-test.c [libstargate]

6.1.1. Funcionamiento

Programa de prueba para la librería libstargate. Sólo se prueban los servicios de PING e identificación. Cuando se ejecuta el programa, y si el Stargate está conectado, aparecerá en la consola algo parecido a lo siguiente:

```
$ ./sgc-null-test

Cliente de prueba para servidor Nulo
Conectando...SG-GENERIC-PIC16F876-USER-0
Conexion establecida

Menu de Opciones
----------------
1.-   PING
2.-   IDENTIFICACION
ESC.- Terminar
```

En este caso se está probando con un StarGate Genérico implementado en un PIC16F876, pero sería válido con cualquier otro StarGate convencional.

Al arrancar el programa se establece la conexión con el StarGate y se detecta de qué tipo es. A continuación aparece un menú de opciones que nos permite acceder a los servicios de PING e IDENTIFICACIÓN.

Si pulsamos la opción '1' y el Stargate está conectado obtendremos los siguiente:

```
PING OK
```

Si el Stargate está desconectado se verá:

```
[TIMEOUT]
```

Con la opción 2 accedemos al servicio de identificación. Para este Stargate se obtiene:

```
SG-GENERIC-PIC16F876-USER-0
```

Y si no se encuentra conectada aparecerá el mismo mensaje de [TIMEOUT] que antes.

6.1.2. Inicialización

Para poder acceder a los servicios de la librería libstargate hay que hacer lo siguiente:

- Inicializar el puerto serie:

```
int serial_fd;  /*-- Descriptor puerto serie */
//-- Abrir el puerto serie
serial_fd=sg_serial_open("/dev/ttyS0");
```

- Inicializar el módulo sg-stramas:

```
sg_tramas_init(serial_fd);
```

Esto se puede ver en la función *main()* del programa.

6.1.3. Acceso a los servicio

El acceso a los servicios es trivial. Como ejemplo se muestra la manera de hacer un PING al stargate:

```
  int status;
  status=sg_ping(); //-- Hacer un ping
//-- Comprobar lo que ha pasado
  switch(status) {
  case 1:
    printf ("PING OK\n");
    break;
  case 0:
    printf ("[TIMEOUT]\n");
    break;
  default:
    printf ("[ERROR]\n");
  }
```

6.1.4. Cierre del puerto serie

Antes de que la aplicación termine sólo hay que cerrar el puerto serie:

```
  //-- Cerrar puerto serie
  close(serial_fd);
```

6.2. sgc-null-test-hilos [libstargateh]

6.2.1. Funcionamiento

Se trata del mismo ejemplo que el programa sgc-null-test-hilos pero que utiliza dos hilos (además del hilo motor incluido en la libstargateh). Un hilo se encarga de gestionar el menu y lanzar las peticiones que solicita el usuario. Otro hilo está constantemente haciendo PINGs al Stargate. Si el PING es correcto se mueve una barrita en la consola.

```
$ ./sgc-null-test-hilos

Cliente de prueba para servidor Nulo
Conectando...SG-GENERIC-PIC16F876-USER-0
Conexion establecida

Menu de Opciones
----------------
1.-   PING
2.-   IDENTIFICACION
ESC.- Terminar

/
```

El usuario puede elegir cualquier de las dos opciones, que se comportan exactamente igual que en el caso del programa ejemplo sgc-null-test, sin embargo en la parte inferior hay una “barrita”, controlada por otro hilo, que gira por cada PING correcto realizado. Si se para es debido a que no hay conexión con el Stargate.

6.2.2. Inicialización

La inicialización es similar a como se usa la libstargate. Primero se habre el puerto serie y luego se inicializa el módulo sgm_motor(), pasándo como parámetro el descriptor del puerto serie.

```
int serial_fd; /*-- Descriptor puerto serie */
//-- Abrir el puerto serie
serial_fd=sg_serial_open("/dev/ttyS0");
//-- Inicializar el motor
sgm_motor_init(serial_fd);
```

6.2.3. Acceso a los servicios

Se hace de la misma manera que en el ejemplo sgc-null-test. Sólo cambian los prefijos. El servicio de PING se invoca de la siguiente manera:

```
int status;
status=sgm_ping();
switch(status) {
case 1:
  printf ("PING OK\n");
  break;
case 0:
  printf ("[TIMEOUT]\n");
  break;
default:
  printf ("[ERROR]\n");
}
```

6.2.4. Finalización

Sólo hay que terminar con los threads lanzados y cerrar el puerto serie, de la forma usual.