

3. Dispositivos estudiados

3.1 Transceptor de infrarrojos Fyre-Fly

Se trata de un transceptor (transmisor y receptor) de infrarrojos que se conecta a cualquier PC a través de su puerto serie. Es un dispositivo “Plug & Play” con dos diodos: uno emisor y otro receptor.



Figura 3.1 Imágenes frontal y trasera del transceptor Fyre-Fly

Sus características principales son:

1. Se une directamente al puerto serie del ordenador personal o de cualquier otro dispositivo con capacidades asincrónicas RS232.
2. Soporta tasas de datos de hasta 38.400 bps.
3. Ideal para infrarrojos de alta velocidad:
 - Arranque inalámbrico para tarjetas de desarrollo integradas.
 - Comunicaciones inalámbricas con microcontroladores 8051, PIC y BASIC Stamp.
 - Aplicaciones robóticas inalámbricas.
 - Control inalámbrico de servocontroladores serie y LCD's serie.
 - Aplicaciones de control remoto inalámbrico.
 - Transferencia de archivos de PC a PC con tasas de datos de hasta 38.400 bps.
 - Sistemas de registro de datos industriales inalámbricos.
4. El rango de operación en un recinto cerrado es del orden de 20 metros.

5. Dos LED's infrarrojos con 44° y 22° de ángulos de aceptación respectivamente.
6. Puede trabajar con adaptadores de USB a puerto serie.

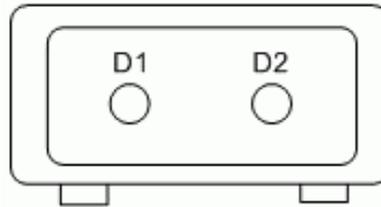


Figura 3.2 Diagrama frontal del transceptor Fyre-Fly

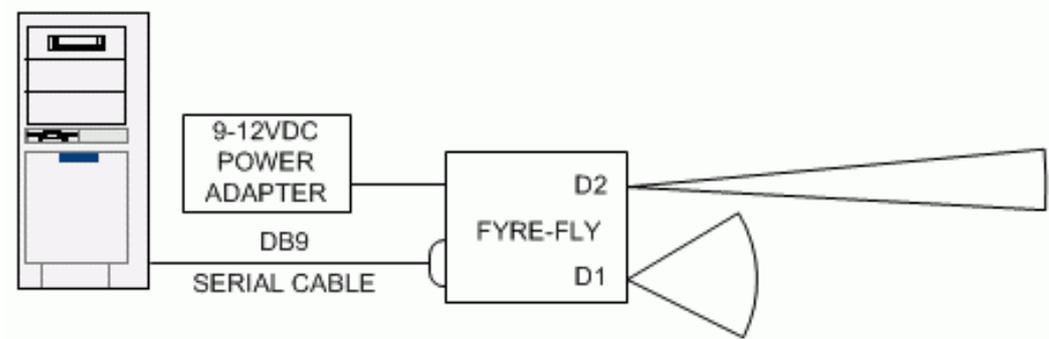


Figura 3.3 Funcionamiento del transceptor Fyre-Fly

Especificaciones:

1. Frecuencia de portadora: 455 kHz +/- 0,1%
2. Duty Cycle de portadora: \approx 45%
3. Tasa de datos máxima: 38.400 bps
4. Longitud de onda del emisor: 870 nm
5. Corriente del LED D1: \approx 158 mA pico; 71,1 mA Avg a 12 VDC
6. Corriente del LED D2: \approx 326 mA pico; 146,7 mA Avg a 12 VDC
7. Corriente reactiva: \approx 11,3 mA a 12 VDC
8. Corriente recibida: \approx 14,8 mA a 12 VDC
9. Interfaz serie: Conector RS232 DB-9 de 9 pines hembra
10. Requisitos eléctricos: De 7 a 12 VDC 500 mA

3.2 Receptor de infrarrojos GC-IRL IR Learner

3.2.1 Introducción

El Global Caché GC-IRL IR Learner se conecta al puerto RS232 de cualquier ordenador personal y se usa para aprender el espectro completo, de 30 kHz a 500 kHz, de los códigos IR que se usan para controlar equipos electrónicos mediante mandos a distancia. Usado junto al software GC-IRL Utility (suministrado gratuitamente por el fabricante), los códigos IR recibidos se muestran por pantalla, se convierten a otros formatos, y se copian automáticamente en el portapapeles para poder crear una base de datos de códigos IR. Los códigos almacenados pueden ser usados para controlar dispositivos desde cualquier ordenador de la red. La alimentación del GC-IRL es suministrada por la línea RTS del puerto serie y no requiere ninguna fuente de energía externa.

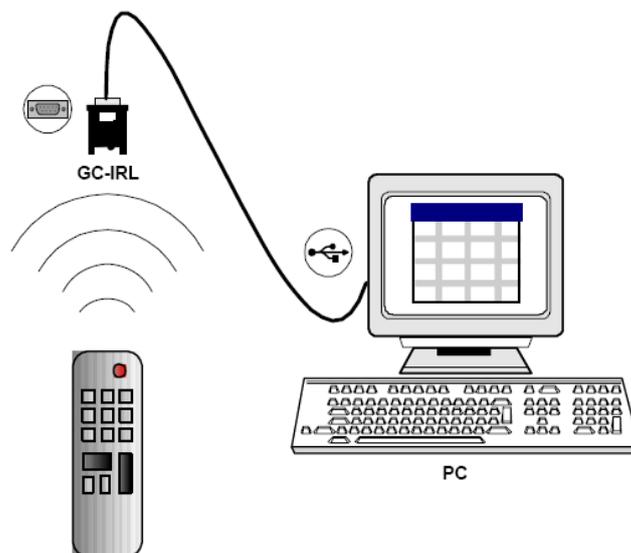


Figura 3.4 Funcionamiento del receptor GC-IRL

3.2.2 Codificación de la señal IR

El GC-IRL transmite señales IR en tiempo real como cadenas de caracteres ASCII delimitados por coma y terminadas por el carácter retorno de carro. La estructura de ASCII es similar a las órdenes de salida IR del GC-100 para simplificar la memorización y repetición.

Una señal IR es una secuencia de estados “on” y “off” modulados con una frecuencia portadora durante el estado *on* (ver figura 3.5). La mayor parte de dispositivos de control remoto IR funcionan en frecuencias cercanas a los 40 kHz. Los estados *on* y *off* se miden en períodos de la frecuencia portadora ($\tau = 1 / f$). Por ejemplo, un estado *on* de 24 representa 600 μ s para una frecuencia portadora de 40 kHz ($600 \mu\text{s} = 24 / 40.000\text{Hz}$).

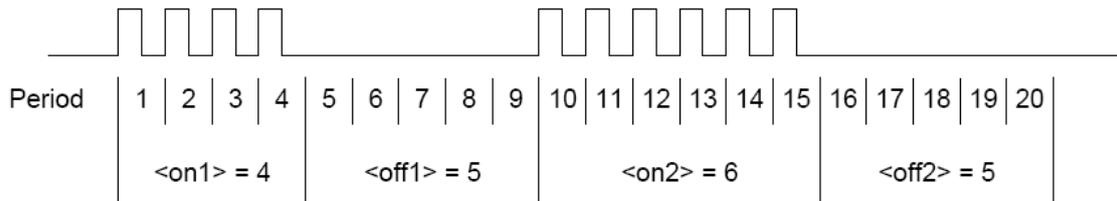


Figura 3.5 Secuencia de estados de una señal IR

La secuencia IR comienza con “GC-IRL” para identificar los datos. Desde el GC-IRL se recibe lo siguiente:

GC-IRL, <frecuencia>, <on1>, <off1>, <on2>, <off2>, ..., <onN>, <offN> €

Donde:

<frecuencia> es /32000/33000/.../500000/ (en Hz)

<on1> es /4/5/.../65635/ (períodos de la portadora)

<off1> es /4/5/.../65635/ (períodos de la portadora)

Lo siguiente es una cadena de caracteres ASCII que representa una señal típica de mando a distancia IR:

GC-IRL,55000,22,340,24,156,23,92,23,157,23,157,23,92,23,156,23,156,23,157,23,157,23,156,23,157,23,157,23,156,24,156,23,157,23,157,23,1375 €

Esta cadena de caracteres es interpretada como una señal con una frecuencia portadora de 55 kHz. El primer estado *on* es de 22 períodos de la frecuencia portadora o 400 μ s. El siguiente estado *off* es de 340 períodos o 6,182 ms de duración. El estado *off* final de 1375 en realidad puede ser más largo, pero es cortado por el GC-IRL tras

sobrepasar el período “IR end timeout”. Este período tiene que ser bastante largo para distinguirse entre dos señales separadas. La mayoría de mandos a distancia envían repetidamente órdenes IR cuando el botón se mantiene pulsado. Por lo general se inserta una pausa entre cada secuencia de instrucción IR. Algunas veces, las instrucciones repetidas difieren de la primera enviada. El GC-IRL codifica las órdenes repetidas como órdenes separadas si la pausa entre las órdenes es más larga que el “IR end timeout”. De lo contrario las órdenes repetidas se envían como una orden larga.

3.2.3 Compresión de la señal IR

Algunos controles remotos IR envían señales continuas sin hacer una pausa entre órdenes, por ejemplo cuando mantenemos pulsado el control de volumen. Esto puede crear una situación donde el GC-IRL es incapaz de transmitir datos serie lo bastante rápido para impedir que las órdenes IR desborden el buffer interno. Para evitar esto, el GC-IRL emplea un modo de compresión para reducir los datos serie a un tercio de su tamaño original. Si se produce un desbordamiento, se añade una X ASCII al final de la cadena de texto IR para indicar el error.

En este formato comprimido, las parejas de estados *on* y *off* repetidas se representan en el flujo de datos serie como letras mayúsculas (A, B, C, o D). Hasta cuatro combinaciones diferentes son condensadas de varios caracteres de ASCII a sólo uno. El proceso de compresión asigna la letra A a la primera pareja *on/off* para representarla en una nueva señal IR, la B a la siguiente pareja diferente, y así sucesivamente. Usando la compresión, la primera aparición de una pareja *on/off* se transmite como antes, todas las repeticiones subsecuentes de esa pareja se representan por la letra asignada. Para facilitar la compresión, las parejas *on/off* se consideran equivalentes si se diferencian en menos de 3 el uno del otro, reduciendo así los efectos indeseables de ruido y error de redondeo. En el siguiente ejemplo vemos en negrita las parejas alternativas para facilitar su lectura.

```
GC-IRL,55000,22,340,24,156,23,92,23,157,23,157,23,92,23,156,23,156,23,157,23,157,23,156,  
      A   B   C  
23,310,23,311,23,156,24,156,23,157,23,157,23,1375,␣  
D
```

Donde las letras de la A a la D son asignadas así:

A = 22,340 B = 24,156 C = 23,92 D = 23,310

Siguiendo con este ejemplo, la instrucción IR quedará comprimida de la siguiente manera:

GC-IRL,55000,22,340,24,156,23,92BBCBBBBB,23,310DBBBB,23,1375€

La primera pareja (22,340) nunca se repite, por tanto la letra A no se usa. Además, si el último par (23,1375) se hubiese repetido, sería enviado sin comprimir porque las cuatro letras posibles ya han sido asignadas a otras parejas. Finalmente, las letras se envían sin usar el carácter ‘,’ para separarlas, reduciendo la señal original de 138 a 58 caracteres.

3.2.4 Filtrado y medida de ruido

El GC-IRL contiene algoritmos de filtración para mejorar el funcionamiento eliminando señales IR no deseadas que provienen de la luz solar directa o la iluminación fluorescente. Otra fuente de señales no deseadas viene provocada por controles remotos usados desde una gran distancia, parcialmente bloqueada por una obstrucción, apuntando mal, o en con las pilas muy descargadas. La filtración del GC-IRL no sirve para corregir una señal corrupta, pero impide que sea transmitida como datos serie. Cuando una señal IR se corrompe durante la transmisión, el GC-IRL finaliza inmediatamente la salida IR de datos con la letra “Z”.

Se emplean algunas técnicas para identificar señales IR incorrectas, incluyendo un valor mínimo para *on* y *off* de 12. Si en algún momento un valor de *on* o de *off* es menor que el mínimo admitido, la transmisión no se realiza, o si está en curso, se detiene finalizando el flujo de datos con la letra “Z”.

3.2.5 Alimentación e interfaz serie

El GC-IRL utiliza un conector DB9 hembra para conectarse directamente al GC-100 y a la mayor parte de ordenadores personales. El GC-IRL viene preparado para trabajar a 9600 baudios, 8 bits, 1 bit de stop, sin paridad, y sin control de flujo.

La interfaz serie es también la fuente de alimentación para el GC-IRL, que emplea para ello el pin 7. La tensión RTS debe estar entre 8 y 25 voltios y ser capaz de dar 5mA. Los drivers RS232 cumplen estas exigencias. También, la conformidad de conexión hardware (o el control de flujo) debe ser deshabilitada para evitar la interrupción de la tensión RTS.

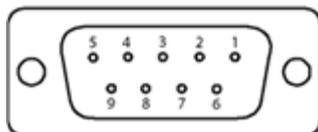


Figura 3.6 Conector DB-9

Pines: 2. Recepción, 3. Transmisión, 5. Tierra, 7. Alimentación

3.2.6 Especificaciones

- Conector: DB9 hembra
- Tasa de símbolos: 9600 baudios
- Otros: sin paridad, un bit de stop
- Control de flujo: no
- Alimentación: suministrada por RTS (pin 7), de 8 a 25 voltios (5mA)
- Ancho de banda: de 32 a 500 kHz

3.3 Adaptador de Red GC-100

El Adaptador de Red GC-100 es un dispositivo fabricado por Global Caché que aporta una serie de funcionalidades a una red IP.

Hay tres versiones disponibles del GC-100, a saber, el GC-100-06, el GC-100-12 y el GC-100-18. La versión 06 tienes menos dispositivos de entrada/salida como veremos más adelante. Las versiones 12 y 18 únicamente varían en el tamaño.



Figura 3.7 Adaptadores de red GC-100-06, GC-100-12 y GC-100-18 respectivamente

Estas funcionalidades son las siguientes:

1. Seis conectores (tres en la versión 06) de audio de 3,5 mm que pueden ser configurados como salida de infrarrojos o sensor de entrada.
2. Tres cierres de relé para conectar o desconectar una determinada conexión eléctrica (no disponibles en la versión 06).
3. Dos puertos serie para comunicación RS-232 (sólo uno en la versión 06).

Para nuestro proyecto dispondremos de la versión GC-100-12. Nos centraremos en el servicio que nos ofrece como emisor de infrarrojos para dotar a la red de la capacidad de gobernar equipos electrónicos caseros.

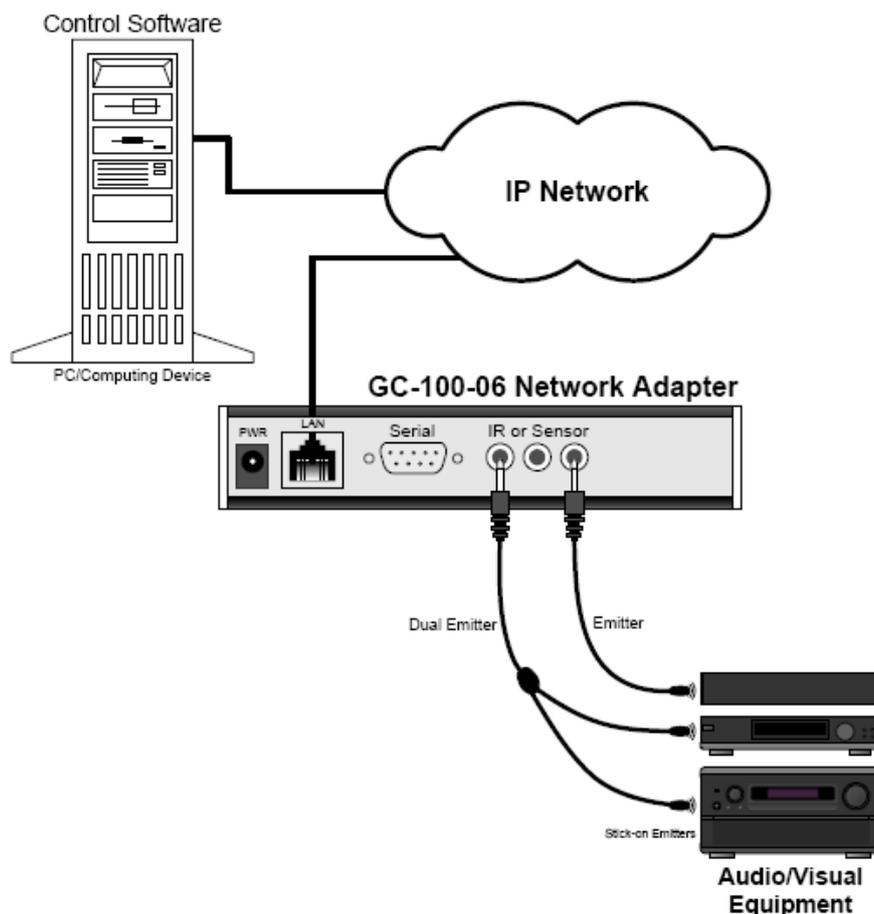


Figura 3.8 GC-100-06 como puente para gobernar equipos electrónicos

Los emisores de infrarrojos que se conectan al GC-100 los suministra el fabricante junto con el Adaptador de Red. Tienen una longitud de 1,5 metros y un alcance de 5 metros en visión directa.

3.3.1 Concepto modular del GC-100

El concepto de diseño modular del Adaptador de Red GC-100 proporciona una variedad de capacidades que pueden ser combinadas. Cada módulo proporciona una función particular, como infrarrojo (IR), entrada digital, o cierres de relé. Un módulo puede tener uno o varios conectores del mismo tipo. Por ejemplo, un módulo IR tiene tres salidas independientes IR; un módulo serie tiene sólo un conector DB9 para datos serie. Esto es porque el número de conectores que un módulo puede tener es dictado por su anchura física de 1.5 pulgadas.

Es importante entender que la dirección de un módulo es determinada únicamente por su posición física dentro del GC-100. El concepto es que cada módulo ocupa 1.5 pulgadas de espacio de panel delantero, incluso si éste forma parte de una tarjeta de circuitos impresos más grande que contiene otros tipos de módulo. Una vez encendido, las direcciones de módulo son asignadas comenzando con 0 para el módulo más a la izquierda y aumentando secuencialmente hacia la derecha hasta que todas las direcciones de módulo sean asignadas. Esto presenta un interfaz de programa coherente de modo que los módulos adicionales puedan ser añadidos en las posiciones vacías.

La dirección de un conector es su posición dentro de un módulo, que comienza en 1 a la izquierda y aumenta hacia la derecha. Una dirección de conector completa incluye la dirección del módulo y la posición del conector dentro del módulo separado por dos puntos (:). La dirección de un conector no tiene por qué estar de acuerdo con la etiqueta del panel delantero del GC-100.

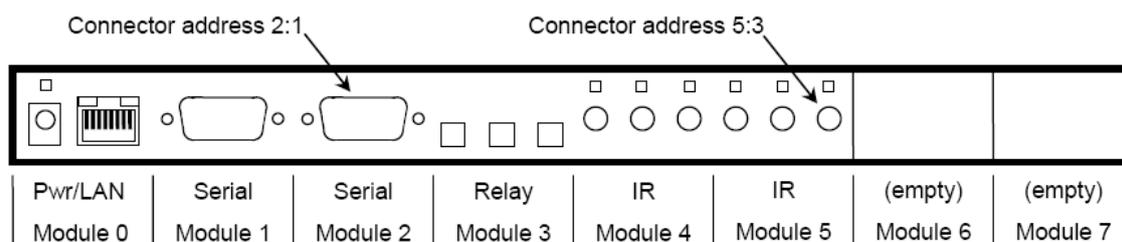


Figura 3.9 Concepto modular del Adaptador de Red GC-100

3.3.2 Cómo configurar la dirección IP

El GC-100 está configurado por defecto para usar DHCP y, de este modo, obtener automáticamente una dirección IP. Para determinar la dirección IP de un GC-100 que usa DHCP, usamos la “Utilidad de Descubrimiento y Configuración” que proporciona Global Caché. En los tres primeros segundos tras el encendido, el GC-100 anuncia su dirección IP y la muestra en la lista. El GC-100 seguirá anunciando su dirección IP en intervalos de entre 10 y 60 segundos mientras esté encendido. Si el GC-100 se conecta directamente a un ordenador personal por medio de un cable cruzado, o si no hay disponible un servicio DHCP, el GC-100 usará por defecto la dirección IP 192.168.1.70. La dirección IP y la configuración para usar DHCP pueden ser cambiadas a través de las páginas web internas del GC-100 a las que se accede escribiendo su IP en la barra de dirección de un navegador web. Entrando en la opción “Network” podemos

seleccionar DHCP o IP estática y, eligiendo IP estática, podemos escribir una nueva dirección IP. El GC-100 comenzará de nuevo con los nuevos ajustes de dirección IP activos.

3.3.3 Baliza de descubrimiento del GC-100

El Adaptador de Red GC-100 cuenta con un mensaje de baliza que puede ayudar en la localización de los GC-100 en la red. La baliza es un paquete UDP enviado a la dirección IP de multidifusión 239.255.250.250 sobre el puerto UDP número 9131. Cualquier sistema que esté escuchando en esta dirección y puerto recibirá el mensaje de baliza periódicamente. El mensaje es enviado 3 segundos después del encendido del GC-100 y, a partir de ese momento, en intervalos aleatorios de 10 a 60.

El mensaje de baliza tiene el siguiente formato:

```
AMXB<-UUID=GC100_000C1E0AF4E1_GlobalCache><-  
SDKClass=Utility><-Make=GlobalCache><-Model=GC-100-12><-  
Revision=1.0.0><Config-Name=GC-100><Config-URL=http://192.168.1.70>
```

El valor UUID contiene la única dirección MAC del GC-100 y es también el nombre registrado con el servidor DHCP. El valor Model puede ser GC-100-12 o GC-100-06. Un GC-100-18 dará como valor en Model el de GC-100-12.

3.3.4 Instrucciones y estructura de datos

La comunicación con el GC-100 se lleva a cabo abriendo un socket TCP sobre el puerto 4998. Socket designa un concepto abstracto por el cual dos programas (posiblemente situados en sistemas distintos) pueden intercambiarse cualquier flujo de datos, generalmente de manera fiable y ordenada. Un socket queda definido por una dirección IP, un protocolo y un número de puerto. Todas las órdenes y datos, a excepción de los que van por el puerto serie (RS-232), son comunicados por el puerto 4998. El puerto 4998 es usado para cosas como el estado del GC-100, los datos IR, la activación de relés y la lectura de estados de entradas digitales. Toda la información, a excepción de la que va por puerto serie, es comunicada por cadenas de caracteres ASCII delimitadas por comas y terminadas por un retorno de carro.

En los puertos serie la comunicación se lleva a cabo sobre los puertos 4999 y sucesivos. Las conexiones serie (RS-232) con el número de módulo más bajo se realizan sobre el Puerto 4999; las conexiones serie con el siguiente número de módulo se llevan a cabo sobre el Puerto 5000; y así sucesivamente.

Sólo una instrucción IR puede ser ejecutada a la vez, por tanto deberá de tenerse una consideración especial a la hora de enviar órdenes IR “back-to-back”. Además, las instrucciones IR deben estar preparados para repetir su patrón múltiples veces, por ejemplo, para subir el volumen de un televisor o de un equipo de sonido. Esta característica puede ser aprovechada para obtener resultados muy deseables. Ya lo veremos más adelante.

3.3.5 Instrucciones del GC-100

Las órdenes empiezan siempre por una cadena de caracteres ASCII que representa el tipo de instrucción. Normalmente la dirección y los datos vendrán a continuación. Las estructuras de las órdenes del GC-100 serán descritas más adelante. El texto incluido entre paréntesis (<texto>) debe ser sustituido por su definición en ASCII. Las diferentes opciones de ASCII son divididas por el separador (!).

Por ejemplo, el estado de un relé es puesto en “ON” por la siguiente instrucción:

```
setstate,<dirección del conector>,<estado> !
```

Donde:

<dirección del conector> es 3:2 (tercer módulo, segundo relé en el módulo)

<estado> es 1 (cierra contactos de un relé abierto)

Para este ejemplo, la cadena de caracteres ASCII de la instrucción sería:

```
setstate,3:2,1 !
```

3.3.5.1 Instrucciones generales

1) *getdevices*

Esta orden del GC-100 es usada para determinar los módulos instalados y sus capacidades. Cada módulo responde con su dirección y tipo. El proceso se completa tras recibir la respuesta *endlistdevices*.

Enviaríamos:

getdevices↵

2) *device*

Es enviado desde el GC-100 como respuesta al *getdevices*:

dispositivo,<dirección del modulo>, <tipo de módulo>↵ □ (una cadena de caracteres como ésta enviada por cada módulo)

Donde:

<dirección del modulo> es |1|2|3|4|...|n|

<tipo de módulo> es |3 RELAY|3 IR|1 SERIAL|

3) *endlistdevices*

endlistdevices↵

Como ejemplo veamos la respuesta de un Adaptador de Red GC-100-12 a la instrucción *getdevices*:

device,1,1 SERIAL↵*device,2,1 SERIAL*↵*device,3,3 RELAY*↵*device,4,3 IR*↵

device,5,3 IR↵*endlistdevices*↵

4) *getversion,<dirección de módulo>*

El número de versión del módulo puede ser obtenido de cualquiera de los módulos de un GC-100. Los módulos combinados sobre la misma tarjeta de circuitos impresos tendrán el mismo número de versión.

Se le enviaría al GC-100:

getversion, <dirección de módulo> ↵

Donde:

<dirección de módulo> es |1|2|3|4|...|n|

5) *version*

Enviado desde el GC-100 como respuesta a *getversion*

version, <dirección de módulo>, <texto> ↵

Donde:

<dirección de módulo> es |1|2|3|4|...|n|

<texto> puede ser cualquier cadena de caracteres ASCII

6) *blink*

La instrucción *blink* (parpadear) sirve para hacer parpadear el LED de encendido del GC-100. Esto es provechoso en una situación de montaje de rack o bastidor donde el instalador necesite identificar un GC-100 en particular.

Enviaríamos al GC-100 lo siguiente:

blink, <onoff> ↵

Donde:

<onoff> es |0|1|. Un valor de '1' hará que el LED comience a parpadear, '0' lo detendrá.

7) *set_NET*

La orden *set_NET* permite al desarrollador configurar los ajustes de red de un GC-100 por medio de una conexión TCP sin necesidad de acceder a las páginas de configuración web.

Enviaríamos al GC-100 lo siguiente:

```
Set_NET,0:1,<configlock>,<opciones IP> ↵
```

Donde:

<configlock> es |LOCKED|UNLOCKED|

<opciones IP> es |DHCP|STATIC,dirección IP,subnet,gateway|

Ejemplo:

```
set_NET,0:1,DHCP ↵
```

Esto hará que la dirección IP se asigne por DHCP.

```
set_NET,0:1,STATIC,192.168.1.70,255.255.255.0,192.168.1.1 ↵
```

Así seleccionaremos IP estática y asignaremos los valores mostrados.

8) *get_NET*

Esta instrucción solicita los ajustes de red actuales y devuelve una cadena de caracteres delimitada por coma con dichos ajustes.

```
get_NET,0:1 ↵
```

9) *set_IR*

Esta instrucción permite al desarrollador configurar cada uno de los puertos individuales IR como salida IR o como entrada de sensor. Los modos posibles son: salida IR, entrada de sensor, entrada de sensor con notificación automática, y salida IR sin portadora.

```
set_IR,<dirección de conector>,<modo> ↵
```

Donde:

<dirección de conector> ya lo hemos definido.

<modo> es |IR|SENSOR|SENSOR_NOTIFY|IR_NOCARRIER|

Ejemplo:

set_IR,5:2,IR ↵

De este modo configuramos el puerto IR 5 como salida IR.

10) *get_IR*

Esta instrucción recupera del GC-100 el actual modo de trabajo de un puerto.

get_IR,<dirección de conector> ↵

Donde:

<dirección de conector> ya lo hemos definido.

11) *set_SERIAL*

Con esta orden podemos configurar los puertos serie de un GC-100.

set_SERIAL,<dirección de conector>,<tasa de baudios>,<control de flujo>,<paridad> ↵

Donde:

<dirección de conector> ya lo hemos definido.

<tasa de baudios> es |57600|38400|19200|9600|4800|2400|1200|

<control de flujo> es |FLOW_HARDWARE|FLOW_NONE|

<paridad> es |PARITY_NO|PARITY_ODD|PARITY_EVEN|

Ejemplo:

set_SERIAL,1:1,38400,FLOW_HARDWARE,PARITY_NO ↵

Esto configuraría el puerto serie número 1 del GC-100 para operar a 38400 baudios, usar control de flujo hardware, y sin paridad.

12) *get_SERIAL*

Con esta orden recuperamos la configuración actual de un puerto serie determinado.

get_SERIAL,<dirección de conector> ↵

Donde:

<dirección de conector> ya lo hemos definido.

13) *unknowncommand*

Una respuesta *unknowncommand* será enviada por el GC-100 cuando no entienda una instrucción o bien ésta no proceda. Esto puede pasar si, por ejemplo, un conector es configurado como una entrada digital y la instrucción solicitada es *sendir*.

El GC-100 contestaría así:

unknowncommand, [código de error]

Más adelante veremos los códigos de error.

3.3.5.2 Instrucciones IR

Estructura IR

Una transmisión IR, o infrarrojo, se crea enviando un modelo IR adecuado al GC-100. Este modelo es una colección de estados <on> y <off> modulados con una frecuencia portadora (*f*) que está presente durante el estado <on>. La frecuencia de portadora está normalmente entre 35 y 45 KHz aunque algunos fabricantes usan 200 KHz y más. El tamaño de tiempo de un estado <on> y <off> se calcula en unidades de

periodo de la portadora. Por ejemplo, un valor <on> de 24 modulado con una frecuencia portadora de 40 KHz produce un estado <on> de 600 μ S.

$$600\mu S = \langle on \rangle * P = 24 / f = 24 / 40,000 \quad \text{donde } P = 1 / f$$

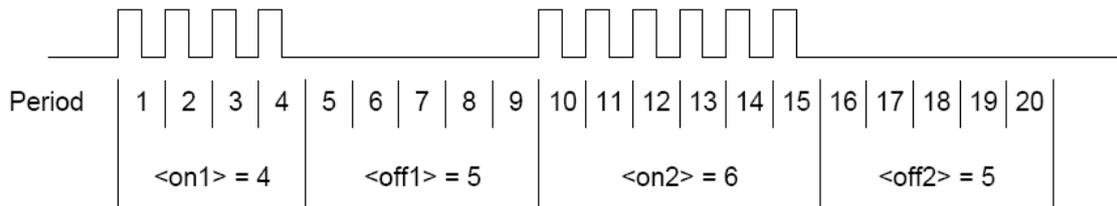


Figura 3.10 Secuencia de estados de una señal IR

Enviar IR

1) *sendir*

El control de dispositivos IR se lleva a cabo mediante la instrucción *sendir*. Puesto que las instrucciones IR necesitan varios cientos de milisegundos para completarse, el GC-100 proporciona un reconocimiento (acknowledgement) para indicar cuando está listo para aceptar la siguiente instrucción.

sendir, <dirección de conector>, <ID>, <frecuencia>, <contador>, <offset>, <on1>, <off1>, <on2>, <off2>, ..., <onN>, <offN> (donde N es menor que 256)

Donde:

<dirección de conector> ya lo hemos definido.

<ID> es |0|1|2|...|65535|⁽¹⁾ (para la instrucción *completeir*)

<frecuencia> es |20000|20001|...|250000| (en hercios)

<contador> es |0|1|2|...|31|⁽²⁾ (la instrucción IR es enviada un número de veces igual a <contador>)

<offset> es |1|3|5|...|511|⁽³⁾ (usado si <contador> es mayor que 1)

<on1> es |1|2|...|65635|⁽⁴⁾ (medido en periodos de frecuencia de la portadora)

$\langle off1 \rangle$ es $|1|2|\dots|65635|^{(4)}$ (medido en periodos de frecuencia de la portadora)

- (1) El $\langle ID \rangle$ es un número ASCII generado por el emisor de la instrucción *sendir*, éste se incluye en la instrucción *completeir* para indicar la terminación de la transmisión respectiva.
- (2) El $\langle contador \rangle$ es el número de veces que una transmisión IR es enviada, si no es antes detenida por un *stopir* u otra instrucción IR. En cualquier caso, el preámbulo sólo se envía una vez (ver $\langle offset \rangle$ a continuación). Si $\langle contador \rangle$ es igual a “0” la cadena de caracteres será repetida continuamente hasta que sea detenida. Si no es detenida, la transmisión parará tras 65535 repeticiones.
- (3) El $\langle offset \rangle$ se aplica cuando $\langle contador \rangle$ es mayor que uno. Para las instrucciones IR que tienen preámbulos, el $\langle offset \rangle$ se emplea para evitar repetir el preámbulo en todas las repeticiones de la cadena de caracteres. El valor de $\langle offset \rangle$ indica la posición dentro del patrón de tiempos donde empezar a repetir la instrucción IR como vemos indicado debajo. El $\langle offset \rangle$ siempre será un valor impar ya que el patrón de tiempo siempre empieza con un estado $\langle on \rangle$.

| $\langle offset \rangle$ impar; | Inicio repetición; | $\langle offset \rangle$ par; | Inicio repetición |
|---------------------------------|---------------------------------|-------------------------------|----------------------------|
| 1 | $\langle on1 \rangle$ | 2 | $\langle off1 \rangle$ |
| 3 | $\langle on2 \rangle$ | 4 | $\langle off2 \rangle$ |
| ... | ... | ... | ... |
| n-1 | $\langle on((n/2) - 1) \rangle$ | n | $\langle off(n/2) \rangle$ |

Donde n es un número par.

- (4) Ya que una transmisión IR termina siempre con un estado $\langle off \rangle$, debe haber un número igual de estados $\langle on \rangle$ y $\langle off \rangle$. Además, los estados $\langle on \rangle$ y $\langle off \rangle$ deben durar como mínimo 80 μ S para que el GC-100 pueda trabajar correctamente. Por ejemplo, con una frecuencia de portadora de 48 KHz el valor mínimo para los estados se calcula como sigue.

$$\langle off \rangle_{\min} = \langle on \rangle_{\min} \geq 80\mu\text{S} * f = 80\mu\text{S} * 48\text{KHz} = 3.84$$

Para un funcionamiento correcto del GC-100, todos los valores $\langle on \rangle$ y $\langle off \rangle$ en el patrón de tiempo deben ser de 4 o superior. Cuando una variable se pierde o está fuera del rango aceptado, el GC-100 enviará un *unknowncommand*. Por ejemplo, las

siguientes órdenes *sendir* serán respondidas con un *unknowncommand* por un GC-100-12.

sendir,2:3,3456,23400,1,1,24,48,24,960 dirección invalida para un GC-100-12, el módulo 2 es un puerto serie.

sendir,5:2,23333,40000,2,3,24,48,24,48,960 no hay el mismo número de valores *<on>* y *<off>*.

sendir,5:3,0,40000,2,2,24,48,24,960 el *<offset>* es par.

2) *completeir*

Todas las órdenes *sendir* son respondidas con un *completeir* del GC-100 tras completarse la transmisión IR. La respuesta *completeir* se asocia con la instrucción *sendir* por medio del *<ID>*. El *<ID>* proporciona un identificador único para determinar que la transmisión IR se ha completado.

completeir,<dirección de conector>,<ID>

Donde:

<dirección de conector> ya lo hemos definido.

<ID> es |0|1|2|...|65535|

Veamos un ejemplo. La instrucción siguiente enviará la secuencia de tiempo IR que dibujamos antes a un GC-100-12.

sendir,5:2,2445,40000,1,1,4,5,6,5

La respuesta del GC-100 sería:

completeir,5:2,2445

Instrucciones “Back-to-Back”

Las órdenes IR son ejecutadas una a una en el GC-100, como mucho, un número de veces igual a $\langle \text{contador} \rangle$. Puede llevarse varios segundos para completar la transmisión. Si el GC-100 recibe una nueva instrucción IR durante la ejecución de otra instrucción previa, ésta última terminará; ninguna otra repetición del patrón IR (debido a un valor $\langle \text{contador} \rangle$ restante) será transmitida. Por lo tanto, las instrucciones IR con un valor de $\langle \text{contador} \rangle$ de 1 siempre terminarán antes de que la siguiente orden IR comience. Sólo la parte restante de una instrucción IR que debe aparecer al repetir el patrón es desechada.

Por ejemplo, la siguiente instrucción IR:

sendir,5:2,45234,34500,4,3,34,48,24,12,24,960↵

generará el siguiente patrón IR si se le permite completarse:

34,48,24,12,24,960,24,12,24,960,24,12,24,960,24,12,24,960

Si la siguiente instrucción IR se recibe durante el envío del tercer estado $\langle \text{on} \rangle$ igual a 24, la repetición del patrón de tiempos *24,12,24,960* se detendrá tras completarse el envío de la repetición actual, con lo que nos quedaría lo siguiente:

34,48,24,12,24,960,24,12,24,960

Esta característica puede ser explotada para determinadas funciones, como el aumento del volumen de audio o el avance rápido en un lector de DVD, o cualquier control que requiera continuas transmisiones IR. Usando un valor de $\langle \text{contador} \rangle$ apropiado la instrucción IR se repite hasta alcanzar el volumen o la escena del DVD deseados, tras lo cual es detenido enviando el siguiente *sendir* o la orden *stopir*. En cualquier caso, cuando se usa un *sendir* o un *stopir* para detener una orden IR el GC-100 no envía el mensaje de reconocimiento *completeir*.

Las instrucciones “no IR” (generales) no se ven afectadas por transmisiones IR y son ejecutadas cuando se reciben. Sin embargo, si una orden *sendir* es enviada antes de que una transmisión IR anterior haya terminado, la nueva instrucción IR permanecerá en la cola de entrada del GC-100 con todas las órdenes no IR subsiguientes hasta que la

transmisión IR actual haya terminado, como hemos explicado antes. Esto nos puede llevar varios cientos de milisegundos.

stopir

La orden *stopir* se usa para detener la repetición de transmisiones IR. Tras la recepción del *stopir* la transmisión IR actual se parará al final del patrón de tiempo que se esté retransmitiendo en ese momento. Las restantes retransmisiones indicadas por el valor de *<contador>* son desechadas.

Sería así:

stopir,<dirección de conector> ↵

Donde:

<dirección de conector> ya lo hemos definido.

3.3.5.3 Entrada y salida discreta

Entradas digitales

El GC-100 envía notificaciones de los cambios de estado de las entradas digitales, además permite que estas entradas sean sondeadas en cualquier momento para conocer su estado actual.

Los conectores de entrada digitales son los mismos conectores usados para la salida IR. La configuración de los conectores del GC-100 se hace individualmente para cada conector. Para que las órdenes siguientes funcionen correctamente deben ir dirigidas a un conector configurado como entrada digital. Si una instrucción solicita información de un conector incorrectamente configurado, el GC-100 dará como respuesta un *unknowncommand*.

1) *getstate*

Enviamos al GC-100 lo siguiente:

getstate,<dirección de conector> ↵

Donde:

<dirección de conector> ya lo hemos definido.

2) *state*

Enviado por el GC-100 en respuesta a *getstate*.

state, <dirección de conector>, <estado de entrada> ↵

Donde:

<dirección de conector> ya lo hemos definido.

<estado de entrada> es |0|1|

Un “1” representa un nivel alto de voltaje digital o ausencia de entrada (ninguna conexión), y un “0” es un nivel bajo de entrada.

3) *statechange*

Si el puerto sensor está configurado como “entrada de sensor con notificación automática”, el GC-100 envía automáticamente un mensaje de notificación cuando se produce un cambio de estado de un conector de entrada digital.

statechange, <dirección de conector>, <estado de entrada> ↵

Donde:

<dirección de conector> ya lo hemos definido.

<estado de entrada> es |0|1|

Cierres de relé

Los cierres de relé del GC-100 se activan poniendo sus estados a “1” y se desactivan con un valor de “0”. La activación de un contacto normalmente abierto cerrará (o conectará) los pines de salida del relé, mientras que un contacto cerrado abrirá

(o desconectará) los pines de salida del relé. Los estados de relé no se conservan al apagar el GC-100, por tanto, todos los relés volverán al estado no activo.

setstate

Los estados de relé se configuran enviando lo siguiente:

setstate,<dirección de conector>,<estado de salida> ¶

Donde:

<dirección de conector> ya lo hemos definido.

<estado de salida> es |0|1| (donde “0” es no activo, y “1” es activo)

La respuesta del GC-100 es:

state,<dirección de conector>,<0|1> ¶

3.3.5.4 Comunicación serie

La comunicación serie en el GC-100 es bidireccional, protocolo RS-232. Toda la comunicación va en 8 bits de datos y un bit de parada. La velocidad de transmisión en baudios se configura a través de la pagina web interna del GC-100, o mediante la instrucción de configuración correspondiente, hasta un máximo de 57.6 Kbaud. La paridad y el control de flujo hardware también pueden ser modificados. El buffer de entrada para la comunicación serie es de 255 octetos sin el control de flujo. Todos los datos serie pasan sin ser procesados hacia un puerto asignado. Cada conector serie tiene asignado un número de puerto único. Al conector serie con el número de módulo más bajo se le asigna el puerto 4999. Al conector serie con el siguiente número de módulo se le asigna el puerto 5000, y así sucesivamente.

Si un buffer se desborda, los datos se perderán. El desbordamiento no ocurrirá a no ser que la conexión de red sea bloqueada y el GC-100 sea incapaz de comunicarse.

3.3.6 Códigos de error

Vamos a ver una lista de los mensajes de error que puede devolver el GC-100 a través del puerto 4998 y la explicación de cada mensaje.

unknowncommand, 1

Se agotó el tiempo de espera porque el retorno de carro (↵) no se recibió. La petición no fue procesada.

unknowncommand, 2

Dirección de módulo inválida (el módulo no existe) recibida cuando intentaba averiguar el número de versión (*getversion*).

unknowncommand, 3

Dirección de módulo inválida (el módulo no existe).

unknowncommand, 4

Dirección de conector inválida.

unknowncommand, 5

La dirección de conector 1 está configurada como “entrada de sensor” e intentó enviar una instrucción IR.

unknowncommand, 6

La dirección de conector 2 está configurada como “entrada de sensor” e intentó enviar una instrucción IR.

unknowncommand, 7

La dirección de conector 3 está configurada como “entrada de sensor” e intentó enviar una instrucción IR.

unknowncommand, 8

El *<offset>* es un número par y debería ser impar.

unknowncommand, 9

Excedido el número máximo de transiciones (256 transiciones on/off permitidas).

unknowncommand, 10

El número de transiciones en la instrucción IR no es par (deber haber el mismo número de estados on y off).

unknowncommand, 11

Enviado un comando de cierre de relé a un módulo que no es un relé.

unknowncommand, 12

Retorno de carro perdido. Todas las órdenes deben terminarse con un retorno de carro.

unknowncommand, 13

Solicitado el estado de una dirección de conector inválida, o bien el conector está programado como salida IR y no como sensor de entrada.

unknowncommand, 14

La instrucción enviada a la unidad no es soportada por el GC-100.

unknowncommand, 15

Excedido el número máximo de transiciones IR.

unknowncommand, 16

Número inválido de transiciones IR (debe ser un número par).

unknowncommand, 21

Intentó enviar un comando IR a un módulo no IR.

unknowncommand, 23

La instrucción enviada no es soportada por este tipo de módulo.