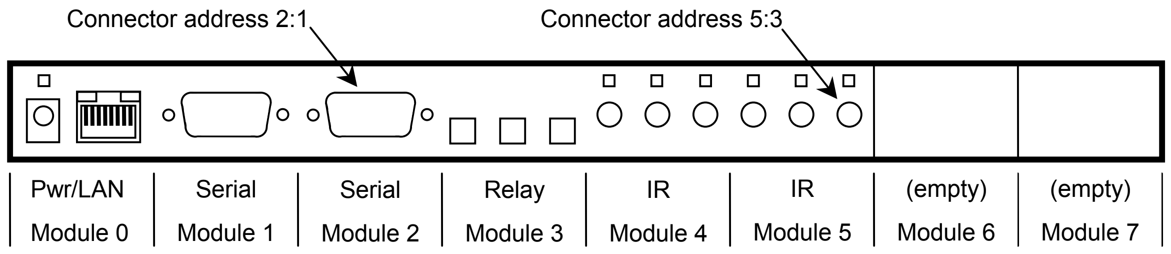
GC-100 API Specification
Version 1.0

1. The GC-100 Modular Concept

The GC-100 Home Network Adapter is designed on a modular concept where a variety of capabilities can be combined within a single enclosure. Each module provides a particular function, such as infrared (IR), digital input, or relay closures. A module may support one to many connectors of the same type. For example, an IR module has three independent IR outputs; where as, a serial module has only one DB9 connector for serial data. This is because the number of connectors a module can support is dictated by its 1.5 inch width.

It is important to understand that a module's address is determined solely by its physical position within the GC-100 enclosure. The concept is that each module occupies 1.5 inches of front panel space, even if it's part of a larger printed circuit board containing other module types. At power on, module addresses are assigned starting with 0 for the left-most modules and increasing sequentially to the right until all module addresses are assigned (see figure 1a). This presents a consistent programming interface as additional modules are added in the empty locations. If module positions are determined to have changed at power on, a devicechange command will by sent out by the GC-100 (refer to section 4.1 below).

A connector's address is its position within a module, starting at 1 on the left and increasing to the right. A complete connector address includes the module address and the connector location within the module separated by a colon. See figure 1a for examples of connector addresses. Note: a connector's address does not necessarily have to agree with the front panel label. Below, the IR connector at address 5:3 is labeled as 6 on the front panel of the GC-100-12.



GC-100-12

figure 1a

2. How to Configure the IP address

The GC-100 default static IP address is 192.168.1.70 (units shipped before May 1, 2003 have the default static IP address of 192.168.1.101). DHCP is not currently supported. The IP address is changed through the GC-100 internal web pages at http://192.168.1.70. Follow the link to "Network Setup" and enter the new IP address and select "Apply." The GC-100 will restart with the new IP address active.

3. Command and Data Structure

Communication with the GC-100 is accomplished by opening a TCP socket on Port 4998. All commands and data, with the exception of serial (RS232) data, are communicated through Port 4998. Port 4998 is used for such things as GC-100 status, IR data, and reading digital input states. All information, with the exception of serial data, is communicated by comma delimited ACSII text strings terminated by a carriage return (↵).

Serial data is communicated over Ports 4999 and higher. Serial connections with the lowest module number will communicate over Port 4999; serial connection with the next higher module number will communicate over Port 5000; and so on.

Only one IR command can be executed at a time, so special consideration must be given when sending back-to-back IR commands. Also, IR commands may be set up to repeat their IR timing pattern multiple times, for increasing volume or fast forward a tape drive. This characteristic can be taken advantage of to create some very desirable results. See "Back-to-Back IR Commands" in section 4.2.3.

4. Command Set

Commands are always initiated by short ASCII string representing the command type. Typically, address and data information will follow. The structures of GC-100 commands are described in the following sections. Text enclosed in brackets (<text>) must be substituted by its ASCII definition. Multiple ASCII choices are divided by separator (|) characters. Note: commands are case sensitive and ASCII spaces (20h) are ignored.

For example, a relay state is set to ON by the following command:

```
setstate,<connectoraddress>,<state>↵
```

where;

<connectoraddress> is 3:2	(3rd module, 2nd relay in module)
<state> is 1	(close contacts on a "normally open" relay)

For this example the command ASCII string is,

```
setstate,3:2,1↵
```

and the hex representation is,

```
73 65 74 73 74 61 74 65 2C 33 3A 32 2C 31 0D
```

4.1 General Commands

getdevices

The GC-100 command is used to determine installed modules and capabilities. Each module responds with its address and type. This process is completed after receiving an endlistdevices command.

Sent to GC-100:

```
getdevices↵                         (query for modules and capabilities)
```

device

Sent from GC-100 in response to getdevices:

```
device,<moduleaddress>,<moduletype>↵   (one string sent for each module)

    where;
    <moduleaddress> is |1|2|3|4|….|n|
    <moduletype> is  |3 RELAY|3 IR|SERIAL|
```

endlistdevices

```
endlistdevices↵                     (end of query response)
```

The following is the GC-100-12 (figure 1a) response to a getdevices command:

```
device,1,SERIAL↵device,2,SERIAL↵device,3,3 RELAY↵device,4,3 IR↵
    device,5,3 IR↵endlistdevices↵
```

devicechange

At power on the GC-100 will transmit a devicechange command if modules where moved or added to the GC-100 enclosure. A getdevices command is required to determine the new module configuration.

Sent from GC-100:

```
devicechange↵
```

getversion,<moduleaddress>

The module version number may be obtained from any or all modules in a GC-100. Modules combined on the same printed circuit board will have the same version number.

Sent to GC-100:

```
getversion,<moduleaddress>↵
    where;
    <moduleaddress> is |1|2|3|4|….|n|
```

version

Sent from GC-100 in response to getversion:

version,<moduleaddress>,<textversionstring>↵

where;
<moduleaddress> is |1|2|3|4|….|n|
<textversionstring> can be any ACSII string

unknowncommand

An unknowncommand will be sent by the GC-100 if a command is not understood. This can happen if a connector is set up as a digital input and the command requested is sendir.

Sent from GC-100 is response to unknown commands:

unknowncommand

4.2 IR Commands

4.2.1 IR Structure

An IR, or infrared, transmission is created by sending an IR timing pattern to the GC-100. This pattern is a collection of <on> and <off> states modulated with a carry frequency (f) which is present during the <on> state. A carry frequency is typically between 35 to 45KHz with some equipment manufacturers using 200 KHz and above. The length of time for an <on> or <off> state is calculated in units of the carry frequency period. For example, an <on> value of 24 modulated with a 40KHz carry frequency produces an <on> state of 600µS, as calculated below.

$$600\mu S = \text{<on>} * P = 24 / f = 24 / 40{,}000 \qquad \text{where } P = 1 / f$$

Figure 4.2a illustrates an IR timing pattern that would be created for the value shown below. IR timing patterns typically have a long finally <off> values to ensure the next IR command is not interpreted as part of the current IR transmission.

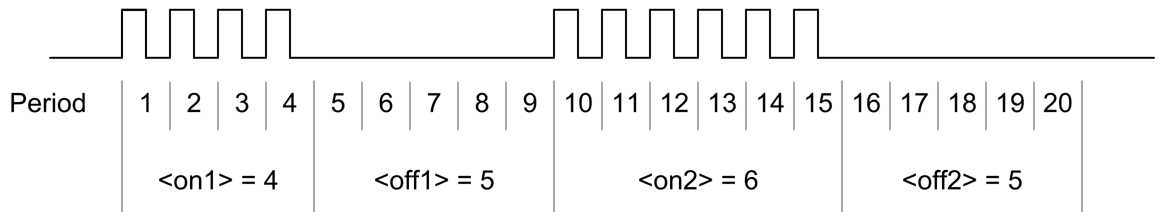


figure 4.2a

11894 Upper Applegate Road
Jacksonville, Oregon 97530
Phone: 541-899-4800
Fax: 541-899-4808
www.globalcache.com

4.2.2 Sending IR

sendir

Control of IR devices is accomplished through the sendir command. Since IR commands may take several 100mS to complete, the GC-100 provides an acknowledgement to indicate when it is ready to accept the next command.

Sent to GC-100:

```
sendir,<connectoraddress>,<ID>,<frequency>,<count>,<offset>,<on1>,
     <off1>,<on2>,<off2>,….,onN,offN↵     (where N is less than 256)
```

where;
<connectoraddress> is as defined in section 1.

<ID> is	|0|1|2|…|65535| [1]	(for the completeir command, see below)
<frequency> is	|20000|20001|….|250000|	(in hertz)
<count**>** is	|0|1|2|….|31| [2]	(the IR command is sent <count> times)
<offset> is	|1|3|5|….|511| [3]	(used if <count> is greater than 1, see below)
<on1> is	|1|2|…|65635| [4]	(measured in periods of the carry frequency)
<off1> is	|1|2|…|65635| [4]	(measured in periods of the carry frequency)

[1] The <ID> is an ASCII number generated by the sender of the sendir command, which is included later in the completeir command to indicate completion of the respective sendir transmission.

[2] The <count> is the number of times an IR transmission is sent, if it is not halted early via a stopir or another IR command (see section 4.2.3). In all cases, the preamble is only sent once, see <offset> below. A <count> of "0" is a special case where the IR timing pattern is continually repeated until halted. However, IR transmission will halt on its own after the IR timing pattern is repeats 65535 times.

[3] An <offset> applies when the <count> is greater than one. For IR commands that have preambles, an <offset> is employed to avoid repeating the preamble during repeated IR timing patterns. The <offset> value indicates the location within the timing pattern to start repeating the IR command as indicated below. The <offset> will always be an odd value since a timing pattern begins with an <on> state.

<offset> odd value	repeat start location	<offset> even value	should not point here	
1	<on1>	2	<off1>	no preamble
3	<on2>	4	<off2>	
….	….	….	….	
n-1	<on((n/2) -1)>	n	<off(n/2)>	where n is an even number

[4] Since an IR transmission ends in an <off> condition, there must be an equal number of <on> and <off> states. Also, every <on> and <off> state must meet an 80µS minimum time requirement for the GC-100 to work properly. For example, with a carry frequency of 48KHz the minimum value for <on> and <off> states is calculated below.

$$<\text{off}>_{\text{min}} = <\text{on}>_{\text{min}} \geq 80\mu S * f = 80\mu S * 48KHz = 3.84$$

For proper GC-100 operation, all <on> and <off> values in the timing pattern must be 4 or higher.

All of the conditions above must be met for valid sendir commands. When a variable is missing or outside the accepted range an unknowncommand will be sent by the GC-100. As an exercise, the sendir commands below will trigger a GC-100-12 unknowncommand response.

sendir,2:3,3456,23400,1,1,24,48,24,960↵	invalid GC-100-12 address, module 2 is serial
sendir,5:2,23333,40000,2,3,24,48,24,48,960↵	not an equal number of <on> and <off>
sendir,5:3,0,40000,2,2,24,48,24,960↵	<offset> is an even number

completeir

All sendir commands are acknowledged with a completeir command from the GC-100 after completion of the IR transmission. The completeir command is associated with the sendir command through an <ID>. When utilized, the <ID>s can provide a unique identifier to determine which IR transmission has completed.

Sent from GC-100 in response to sendir:

```
completeir,<connectoraddress>,<ID>↵
```

where;
<connectoraddress> is as defined in section 1.
<ID> is |0|1|2|…|65535|

A few example IR commands are shown below:

The following will send the IR timing sequence illustrated in figure 4.2a to the 5th IR connector on the GC-100-12 shown in figure 1a.

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

completeir,5:2,2445↵                    (GC-100 response after completion)
```

In the next example, the following two IR commands will send the same IR timing pattern. Note: the carry frequency is 34.5KHz and <ID>s are different so as to provide unique completeir acknowledgements. The following is a simple IR timing pattern of 24,12,24,960 which is sent four times with a preamble of 34,48:

```
sendir,5:2,4444,34500,1,1,34,48,24,12,24,960,24,12,24,960,24,12,24,960,24,12,24,960↵

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

Acknowledgements for above IR commands are,

```
completeir,5:2,4444↵

completeir,5:2,45234↵
```

The second IR command structure is the recommended method, avoiding long commands and allowing repeats of the command to be halted if requested. See "Back-to-Back IR Commands" section below.

4.2.3 Back-to-Back IR Commands

A general discussion is necessary to better understand how IR commands are executed in the GC-100. IR commands are executed one at a time, which with large <count> values, may take several seconds to complete transmission. If a new IR command is received during execution of an earlier IR command, the IR command in progress will terminate at the end of its current timing pattern; no further repeat timing patterns, due to a remaining <count> value, are transmitted. Therefore, IR commands with a <count> of 1 will always finish before the next IR command is started. Only the remaining portion of an IR command that may arise from restarting a repeating timing pattern is discarded.

For example, the above IR command

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

will generate the following IR pattern if allowed to complete:

```
34,48,24,12,24,960,24,12,24,960,24,12,24,960,24,12,24,960
                   ⇑
                   assume the next IR command is sent during the 24 state.
```

However, if the next IR command is received at the location shown above, the repeating timing pattern 24,12,24,960 is halted after completion of the current <count>; creating the timing pattern below.

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

This characteristic may be exploited to create desirable effects, such as increasing audio volume, fast-forwarding a DVD player, or any control requiring continuous IR transmissions. By using an appropriately high <count>, an IR command repeats until the desired volume or DVD scene is reached, where upon it is then halted by sending the next sendir or stopir command. In either case, when a sendir or stopir is used to halt a previous IR command a completeir acknowledgement is not sent from the GC-100.

Other non-IR commands are not affected by IR transmissions and execute when received. However, if a sendir command is sent before an earlier IR transmission is finished, the new IR command will remain in the GC-100 input queue along with all other non-IR subsequent commands until the present IR transmission has halted, as explained above. This may take several 100mS.

stopir

A stopir command is used to halt repeating IR transmissions. After receipt of stopir the present IR transmission will halt at the end of the current timing pattern. Any remaining <count> will be discarded.

Sent to GC-100:

```
stopir,<connectoraddress>↵
```

where;
<connectoraddress> is as defined in section 1.

4.3 Discrete Input and Output

4.3.1 Digital Inputs

The GC-100 sends out notifications for digital input state changes as well as allowing the inputs to be polled for their current state at any time. These change notifications can not be disabled.

Digital input connectors are the same connectors used for IR output. The connector configuration is determined by the GC-100 internal web pages on an individual connector basis. For the following commands to operate correctly the connector being addressed must be configured for digital input. If a command requests information from an improperly configured connector an unknowncommand will be sent from the GC-100.

getstate

Sent to GC-100:

```
getstate,<connectoraddress>↵
        where;
         <connectoraddress> is define in section 1.
```

state

Sent from GC-100 in response to getstate:

```
state,<connectoraddress>,<inputstate>↵

        where;
        <connectoraddress> is defined in section 1.
        <inputstate> is |0|1|
```

Note: A "1" represents a high digital voltage level input and a "0" is a low input or absence of an input (no connection).

statechange

GC-100 sends a notification upon a state change of the any digital input connector as follows:

```
statechange,<connectoraddress>,<inputstate>↵

        where;
        <connectoraddress> is defined in section 1.
        <inputstate> is |0|1|
```

4.3.2 Relay Closures

GC-100 relays are activated by sending a "1" state and deactivated with a "0." Activation of a normally open contact will close (or connect) the relay output pins, while a normally closed contact will open (or disconnect) the relay output pins. Note: relay states are not preserved through a power cycle and all relays will go back to their non-active state until a 1 state is re-sent.

setstate

Relay state is set as follows:

```
setstate,<connectoraddress>,<outputstate>↵

        where;
        <connectoraddress> is defined in section 1.
        <outputstate> is |0|1|          (where 0 is non-active, 1 is active)
```

4.4 Serial Communication

GC-100 serial is bi-directional, RS232 communication. All communication is 8 data bits with no parity and one stop bit. Baud rate is set through a GC-100 internal web page up to 38.4KBaud. Each serial input buffer is 128bytes with no flow control. All serial data is passed through without interpretation via an assigned IP port. Each serial connector is assigned a unique port number. The serial connector with the lowest module number is assigned to IP Port 4999. The serial connector with next highest module number is assigned to IP Port 5000, and so on.

If a serial buffer overflows, data will be lost. The following command is sent by the GC-100 if this occurs. Overflow will not occur unless the network connection is blocked where the GC-100 is unable to communicate.

Sent from GC-100:

serialoverflow

```
serialoverflow,<moduleaddress>↵

        where;
        <moduleaddress> is defined in section 1.
```