



# HCA Tech Note

This is a supplement to the User Guide IR Appendix for Global Cache GC-100 and iTach

(updated 13-Aug-2016)

The GC-100 Global Cache IR interface contains several IR ports that can be configured as inputs or outputs. Some units also include relay outputs and serial ports. The iTach units come in several models that contain IR ports, or serial ports, or relays. They are like the GC-100 split into separate boxes.

## Step 1

The Global Cache interfaces communicate over a TCP/IP connection. After connecting the Global Cache unit to your network you have to locate its IP address.

### Method 1

From the Global Cache web site download the iHelp utility. This will locate any GC-100 or iTach unit on your network subnet.

<http://www.globalcache.com/downloads/>

### Method 2

From your router web interface – most have one - look in its device table for the Global Cache unit. How you do this depends upon the make and model of your router. Here is an example of part of the device table from my router.

	SonosZP	192.168.0.118	00:0e:58:79:d3:7e		Unavailable
				Ethernet	
	GC100_000C1E0 1AA9E_GlobalC ache	192.168.0.180	00:0c:1e:01:aa:9e		Unavailable
				Ethernet	
	android-b76da e5f1b00ebce	192.168.0.95	7c:f9:0e:76:74:1f		Unavailable
				SSID 1	

In this example, a GC-100 is at 192.168.0.180

## Step 2

Use the interface web page to configure the IR ports, and if available, relays and serial ports as you want them. From your favorite browser connect to the IP address of the IR interface.



# HCA Tech Note



### Step 3

Use the IR learner from Global Cache to capture the IR sequences you want to send using HCA. Use the iLearn software. If you are using a GC-100 interface, the IR Learner is separate hardware. The iTach IR interface comes with a built-in IR Learner. The iLearn software can be downloaded from here:

<http://www.globalcache.com/downloads/>

When started, the IR Learner software appears as:







# HCA Tech Note

## Step 5

At this point it is necessary to create one or more keypads that use the IR button definitions in the file created by the iLearn software. This is described in the IR appendix of the user guide and will not be repeated here.

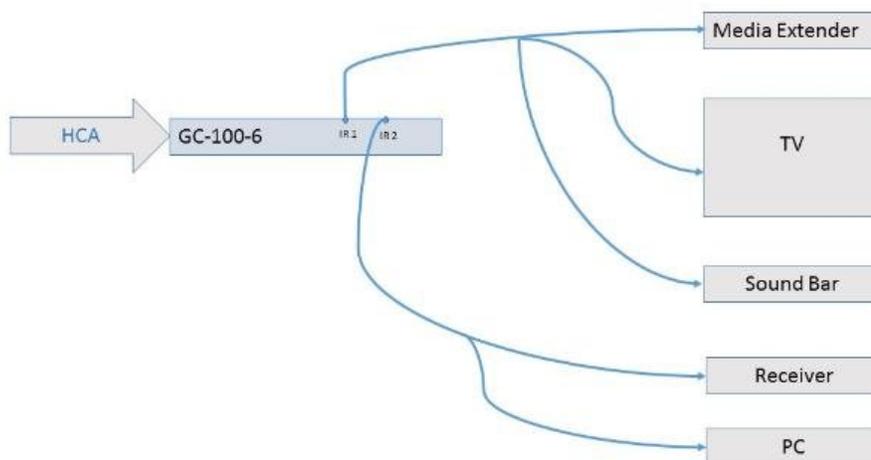
The IR Appendix: [http://www.hcatech.com/download/V13/Doc/a05\\_IR.pdf](http://www.hcatech.com/download/V13/Doc/a05_IR.pdf)

After you have identified the IR interface to HCA and have created the keypads you want, next create IR devices. These are created through the normal “New Device” mechanism and you can create as many IR devices as you need. Associated with each is the name of a keypad you created – how this is done is in the User Guide Appendix referenced above. That keypad displays in HCA when opened from the UI, or when creating schedules and programs that send IR commands. On the keypad you press a button to have the IR interface send the command for that button.

When creating an IR device, in addition to the keypad, you also specify the IR port used when controlling the device. That is, what IR port on the back of the Global Cache interface do you plug the cable with the IR emitter that you put in front of the device’s IR receiver. This is the only location where the IR port is specified. Any place in HCA where a IR device is selected – in a schedule, program, or from the UI, IR commands sent to the device are sent using the port specified in the device properties. The port is associated with the device not with a keypad.

If your Global Cache unit has relays you can create devices for them and control from the UI, programs, and schedules using ON and OFF.

There are several possibilities in how you organize your IR definition files, keypads, and devices. Here is an example.



In this example, there are 5 devices being controlled by IR using 2 GC-100 ports. Here are some possible organizations:

### Choice One

You can use the IR learner to record a separate definition file containing the buttons used to control each device – MediaExtender.txt, TV.txt, etc.



# HCA Tech Note

---

Then in HCA you would create 5 separate keypads, each connected to the definition file for that IR device – Media Extender keypad, TV Keypad, etc.

Finally, you would create in HCA 5 IR devices. One for each item to be controlled with its own keypad.

## **Choice Two**

If you have one physical keypad you use for many different devices – a common thing – you can use the IR Learner to create a single definition file for it.

Then in HCA you can create one or more keypads associated with that definition file. You can have more than one keypad that uses the same definition file created by the IR Learner software. This way you can make keypads with just those buttons needed for that IR device. You would still create 5 IR devices in HCA to represent the 5 real world devices to be controlled.

## **Choice Three**

Again, you can use the IR Learner to create a single definition file for all the buttons on the IR Remote. Then in HCA create a single keypad for that definition file. You would still create 5 IR devices in HCA associating each with the same keypad.

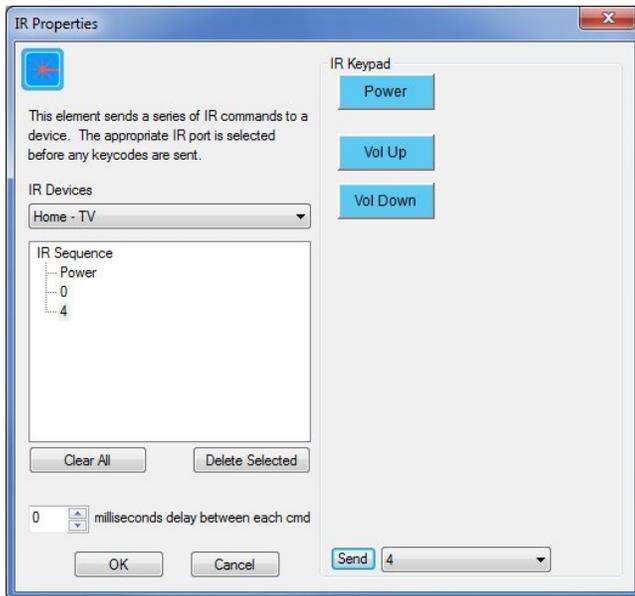
## **Important note about the IR definition files**

When in client-server mode, the IR definition file must be on the server computer and also on the client computer if you want to make changes to work with IR. When operating in client-server mode the suggestion is either to put the IR definition file on a network share where both the server and client computers can see it, or create a folder on both the client and server computers whose path is the same (“c:\IR Key Files” or some such) and place the file in both places and keep them in sync if changed.



# HCA Tech Note

## Using IR sequences in programs



When creating programs that send IR sequences, you select an IR device and then see the keypad associated with that device. You can then add any of the IR buttons to the sequence by pressing keypad buttons. You can also add any IR command from the definition file even if it is not on the keypad by selecting the key name in the dropdown at the bottom of the dialog, and then press the *Send* button and it is added to the sequence.

## Using the Serial ports on the Global Cache Interface

If your interface has serial ports you can send to and receive from them in HCA. The first step is to configure the port using the interface web page – described above – to set the serial parameters of baud rate, number bits, etc. This is an example when configuring a GC-100.

Main Network Serial 1 Serial 2 IR/Sensor 1 IR/Sensor 2

### Serial 1

Module:

Protocol: RS232

Baud Rate:

Flow Control:

Data Bits: 8

Stop Bits: 1

Parity:

Parity Errors: 0

Serial Overflows: 0



**Tech Tip**

If connecting a GC-IRL or GC-IRE to the serial port, select 9600 baud, no parity, and no flow control.



# HCA Tech Note

The way that the serial ports work in the Global Cache interfaces is by sending and receiving over a TCP/IP connection. The first serial port is at IP port 4999, the second is at port 5000, etc.

To communicate with the port, In HCA on the *interfaces* ribbon category add a “Generic IP Interface” as the type and press the configure button:

Configuration for: GC100 Serial 1 (192.168.0.180:4999)

A name is needed when referencing this generic IP interface in programs that use it.

Name: GC100 Serial 1

Connection

IP Port: 4999

Device address for example: dev1.myhome.homedns.org

Device IP address 192 . 168 . 0 . 180

During connect allow 10 seconds before giving up.

On connect use Telnet protocol with User Name: Password:

Auto reconnect if the connection becomes closed

Disconnect when idle and reconnect on use. Idle if not used in 1 minutes.

Logging

Log receptions from this interface

Log sends to this interface

Log entries associated with this interface go to: Main Log

Data

Data is binary

Data is text

What character(s) delimit each message SENT TO this interface? CR

What character(s) delimit each message RECEIVED FROM this interface? CR

OK Cancel

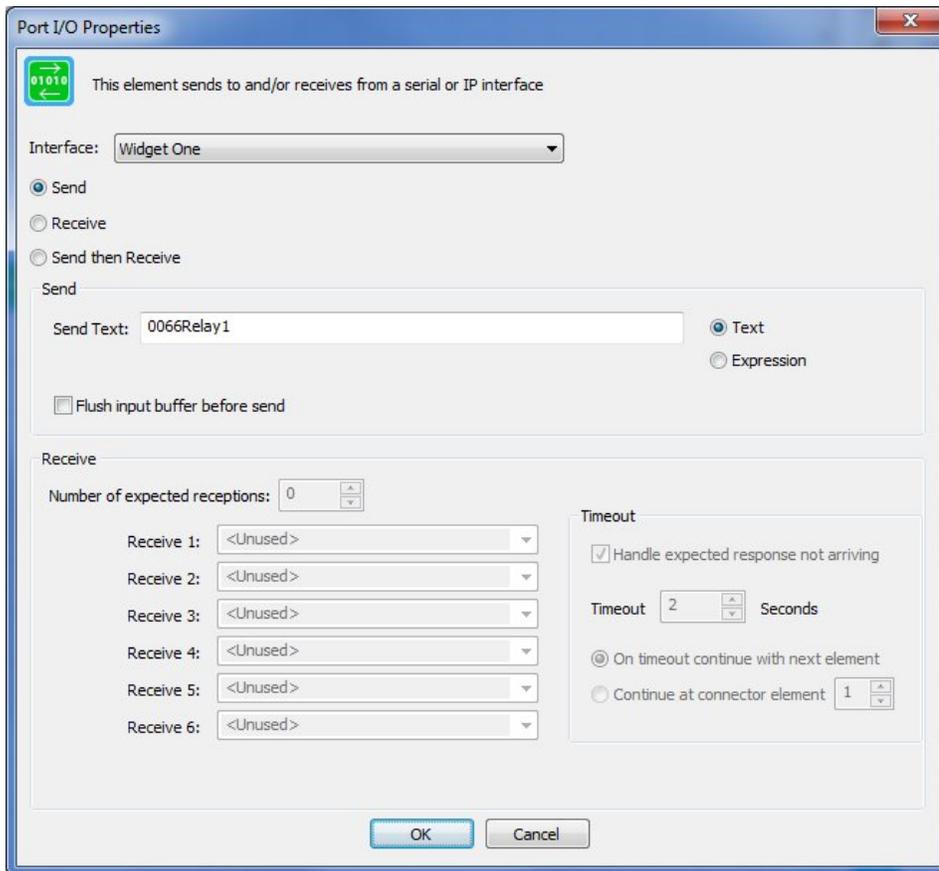
All generic interfaces are given a name and that name is used in programs to refer to it. The IP address and port is specified on this dialog. The IP address can be entered as the usual 4-part set of numbers or as text that is resolved using DNS.

HCA can both receive from and send messages to the serial port but it has to know how to delimit messages being sent and how to parse a single message out from the data stream when receiving. There are selections for this and the choices depend upon if the message is text or binary. How messages are formatted should be part of whatever information you have on the commands you will be sending to the device.

Once you have made all the choices then close the configuration dialog. The next step is to create programs you will run to send and receive from the Global Cache serial port. The Port/IO Visual Programmer element can send to, receive from, or send and receive messages from a generic IP interface.



# HCA Tech Note



The configuration of this element determines if it sends or receives. For receptions, the format of the message as specified in the interface configuration lets HCA parse the data received from the serial port into separate messages. Each message is saved in one or more variables.

Set the *Send Text* to the command you want to pass to the other end of the serial cable. If the message is binary, then the data is entered in hex as text. It is converted to bytes before being sent. For binary messages there is an option in the interface configuration to “frame” the messages being sent using the same format as specified for message parsing.

In this User Guide Appendix linked below is more information about Generic Interfaces. Refer to that appendix for information about how text and binary messages are handled as well as using the Visual Programmer Port/IO element to read and write from a generic IP port.

[http://www.hcatech.com/download/V13/Doc/a12\\_Generic%20Interfaces.pdf](http://www.hcatech.com/download/V13/Doc/a12_Generic%20Interfaces.pdf)

You can also create triggers that process messages received from the Generic IP port. This is also described in the Generic Interfaces appendix.

##end##