

GPI Commander 2 User Manual

For GPI Commander 2 Box
With Software version 1.1.3 minimum



Index

- Chapre 1** **GPI Commander 2 - Hardware Overview**
- Chapre 2** **Introduction to GPI Commander Software**
- Chapre 3** **Triggering AppleScripts on GPI input state change**
- Chapre 4** **Triggering GPI outputs from an Application**

GPI Commander 2 Hardware overview

1

GPI Commander 2 is a USB to GPI and GPI to USB converter. It connects to your Mac or PC using a USB cable. GPI Commander has 8 outputs and 24 inputs. It can start AppleScripts when one of the 24 input is triggered, and each of the 8 outputs can be triggered by AppleScript.

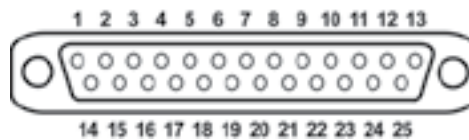
GPI Commander is the ideal companion of all the Softron Applications but also to other applications provided they can send and/or receive AppleScripts.

Power

GPI Commander 2 is powered by the USB cable. Some USB connectors (on keyboard, hubs, etc...) may not have sufficient power for GPI Commander.

GPI Inputs

The 24 GPI inputs come in a DB25 connector. The 24 inputs are on the first 24 pins and the Ground is on the 25th pin.



Do not connect a Serial, nor SCSI cable/connector to the GPI Commander 2 box.





GPI Outputs

The 8 Outputs go out using a phoenix connector with screw contacts.



Rack mounting

GPI Commander has 3 holes on the back so it can be mounted in a Rack.



Introduction to GPI Commander Software

2

The Application GPI Commander is a free application that can be downloaded from our website www.softron.tv. GPI Commander can control one or more GPI Commander 2 boxes via USB. These boxes have 8 outputs and 24 inputs.

When you start the application GPI Commander and that there's a GPI Commander 2 box connected to the Mac, a new window will open. If you connect 2 GPI Commander boxes on the same Mac, you will have 2 windows enabling you to control the 2 boxes from the same Mac.



Each GPI Commander Box has a unique serial number that is shown in the "Serial Number" area on the top of the User Interface.

State of the Inputs

Each of the 24 inputs state is seen in the User interface. If the contact is closed on any of these 24 inputs, the relative checkbox will be checked. When one of the input state changes from closed to opened or from opened to closed, an AppleScript can be started. These AppleScripts will enable you to control almost any application on your Mac. You can for example start a OnTheAir Video playlist when you receive a pulse on the input 1, and/or even send an email! Working with AppleScripts on the input is explained in Chapter 3.

Manually controlling the outputs

Each of the 8 outputs can be controlled manually. In front of each Input you have a checkbox to put the state to on or off. You can also send a pulse to any of the 8 output.

But the interest of GPI Commander is that it can be controlled by AppleScript which means that this process can be automated and the output can be triggered by any application that is able to send an AppleScript. With Softron products, it means for example that you can trigger any of the 8 output at a particular moment of a clip in a playlist OR you can trigger any of the 8 output when you start or stop a recording in MovieRecorder. Working with AppleScripts in the output is explained in Chapter 4.

Triggering AppleScripts on GPI input state change

3

With GPI Commander, you can trigger an AppleScript when the state of any of the 24 input changes. you can trigger an AppleScript when the state changes from closed to opened or from opened to closed. These AppleScripts can control any application on the Mac that is AppleScriptable.

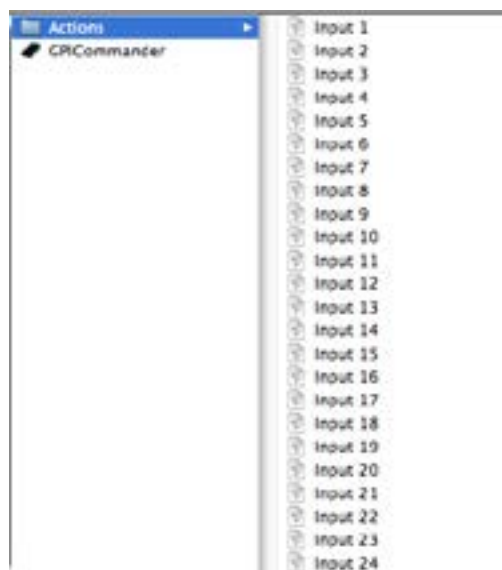
You can easily see if an Application is AppleScriptable simply by drag&dropping it on the Application "Script Editor". If a dictionary opens, it means it is AppleScriptable. If it doesn't it will then tell you that this Application cannot be controlled by AppleScripts. For more information and AppleScripts samples, check the AppleScript folder located in your Applications Folder

Most of Softron Applications can be controlled by AppleScript. It means that a GPI trigger input can start a MovieRecorder recording, start a OnTheAir Video playlist, start a OnTheAir CG project, switch a Matrix Video Switcher with OnTheAir Switch, etc...

Check on our website, on each product download page, you will find sample AppleScripts to control our applications.

Configuring GPI Commander

When you download GPI Commander from the website, you need to unzip the file and then place the entire content of the folder in the Applications Folder. In the GPI Commander folder you will see that there is the application GPI Commander and a folder named "Actions". In this Actions folder, you can see that there are 24 AppleScripts named Input 1, Input 2, Input3,...



The AppleScript named "Input 1" will be triggered when the state of the Input 1 will change. The AppleScript named "Input 2" will be triggered when the state of the Inputs 2 will change, etc...

If you double click on one of the AppleScript, it will open in Script Editor. All the AppleScripts are pre-formatted with almost the same AppleScript sample:

```
on action(parameter)
  if parameter then
    display dialog "Input 1 changed to TRUE = CONTACT CLOSE = ON"
  else
    display dialog "Input 1 changed to FALSE = CONTACT OPEN = OFF"
  end if
end action
```

With this AppleScript, if on your GPI Commander box you have made all the connections and that you close the relay of Input 1 (checkbox checked), it will show you the message "Input 1 has changed to TRUE". If you open the relay of Input 1 (checkbox unchecked), it will show you the message "Input 1 changed to FALSE".

You can then replace these two parts of the AppleScript with anything you want any Application to do.

Below are 2 examples. The first one is to control MovieRecorder and it will start recording when the relays close, and stop recording when the relays open.

```
on action(parameter)
  if parameter then
    tell application "MovieRecorder"
      record
    end tell
  else
    tell application "MovieRecorder"
      stop
    end tell
  end if
end action
```

The second example will start playing the selected clip of the selected playlist in OnTheAir Video when the connectors close, and it will stop playing when the connectors open.

```
on action(parameter)
  if parameter then
    tell application "OnTheAir Video"
      tell front playlist
        push play
      end tell
    end tell
  else
    tell application "OnTheAir Video"
      tell front playlist
        stop
      end tell
    end tell
  end if
end action
```

Don't forget that you can send multiple commands to multiple applications, so for example when you receive a pulse on input 1, you can at the same time start a OnTheAir Video playlist, start a recording in MovieRecorder, switch a Matrix Video Switcher with OnTheAir Switch, and even send an email. Remember, any AppleScriptable application can be controlled by GPI Commander.

Triggering GPI outputs from an application

4

The general idea with AppleScripts to control GPI Commander is to define:

1. To which application we want to talk (GPI Commander)
2. To which Box you want to talk (using a serial number or absolute number)
3. To which output you want to do which action

1. "tell Application"

Each AppleScript talking to GPI Commander must start with "tell Application GPI Commander" and end with "end tell".

2. "tell Device"

Then as you can connect multiple GPI Commander boxes to one computer, you need to specify which box you want to send command to. If you have only one GPI Commander box connected to your computer, you can simply use "tell first device". But if you connect multiple box to one computer, it is difficult to know which box is the first one. You can then use the serial number to define it and use "tell first device whose serial is "00002951"" for example.

3. "Set, Get, Pulse"

Then you can tell what you want to do on this GPI Commander box. The main actions you can perform on any of the 8 outputs are :

- set state of output X to TRUE
- set state of output X to FALSE
- set state of every output to TRUE
- set state of every output to FALSE
- pulse output X
- pulse every output

You can also get the state of every input

- get state of input X

(replace X by the number of the output to control, a number between 1 and 8)

Remember that :

- TRUE = relay is closed (checkbox is checked = ON)
- FALSE = relay is opened (checkbox is unchecked = OFF)
- a pulse will close the relay during a 1/60 of second.
(has no effect if the relay is already closed)

All these commands can be made in one AppleScript. It means that in one AppleScript, you can pulse on output 5, set output 1 to FALSE, and set output 3 to TRUE, and all possible combinations. Refer to the samples given on the next page.

Get the serial Number of a GPI Commander box

There are three ways to know the serial number of a GPI Commander box.

1) It is written physically on the bottom of the GPI Commander box.

2) It is written in on the top of the User Interface.

3) You can write an AppleScript to get it:

```
tell application "GPICommander"  
    get serial of first device  
end tell
```

or :

```
tell application "GPICommander"  
    get serial of second device  
end tell
```



Get the state of each input

You can know the state of every input simply by looking at the user interface, but if you intend to control GPI Commander from another application, you may need to know what is the state of each input before performing an action. For that you can use this AppleScript and replace by the input number you need:

```
tell application "GPICommander"  
    get state of input 1  
end tell
```

Examples

When using multiple boxes on one Mac, specifying the Serial Number of the GPI Commander box

```
tell application "GPICommander"  
    tell first device whose serial is "00002951"  
        set state of output 3 to TRUE  
    end tell  
end tell
```

When using only one box on one mac, you can just write "tell first device":

```
tell application "GPICommander"  
    tell first device  
        set state of output 3 to TRUE  
    end tell  
end tell
```

You can also change the state of all outputs at the same time:

```
tell application "GPICommander"  
    tell first device  
        set state of every output to TRUE  
    end tell  
end tell
```

Instead of changing the state to TRUE or FALSE, you can also pulse on any of the output. This will change the state of the selected output to TRUE for 1/60 second.

```
tell application "GPICommander"  
    tell first device  
        pulse output 3  
    end tell  
end tell
```

You can pulse on every output (this will be made in sequence, one after the other and not all at the same time).

```
tell application "GPICommander"  
    tell first device  
        pulse every output  
    end tell  
end tell
```

Actions can be combined to do multiple actions in the same AppleScript

```
tell application "GPICommander"  
    tell first device  
        set state of output 3 to TRUE  
        set state of output 5 to FALSE  
        pulse output 6  
    end tell  
end tell
```

Other Information

The explanation we provide here on AppleScripts are pretty basic. It is a very easy and powerful tool that you can use in many ways. If you want to see the complete dictionary of GPICommander; don't forget that you can drag&drop the GPICommander application to the ScriptEditor application. It will open a Dictionary specifying almost all commands that can be send to the Application.

Also a very good source of information is this website:

<http://www.macosxautomation.com/applescript/>

Have fun!