Go-Timer

Plugin Guide

v 1.0.6

Giaffo Designs

Contents

Plugin Overview

The Go-Timer plugin is a GrandMA2 plugin that reads a pre-recorded timecode track in a cue stacked show and runs a countdown timer that shows how much time is left until the next [GO] on a sequence, allowing the user to know approximately when to trigger the next cue in a cue stack.

This plugin can be used to learn a show quickly, help a new board op run a cue stacked show on short notice, or simply help an operator know how long he or she can nap during a long and boring show.

System Requirements

It is recommended that the user run this plugin on GrandMA2 version 3.3.4.3 or later.

This plugin requires that the user first have a show file with a sequence of cues that have been recorded into a timecode track at the proper moment in a show. The Go-Timer plugin must be imported into the Plugins pool in the show file. For assistance with importing plugins or creating a timecode recording of your show, please refer to the GrandMA2 User Manual or visit the Tutorials page of GiaffoDesigns.com.

DISCLAIMER:

Plugin Use Tutorial

Step 1: Edit User Variables

Edit the Go-Timer plugin in the Plugins pool and define the three user variables in the **User Config Variables** section of the script.

userconfig.NAME should be given a label describing what the plugin will be timing (for example, **[[main]]** if it is timing your main cue stack, or **[[spots]]** if it is timing a sequence of followspot calls).

userconfig.TIMER should be given a number identifying an unused timer in the clocks pool will be used to show the countdown. There does not need to be an existing timer already. The plugin will generate and overwrite the timer in that slot.

userconfig.PROGRESS_BARS can be given a value of either **true** or **false**. A value of **true** will display a load bar on screen when the plugin first runs, while a value of **false** will hide the load bar.

Finally, click [Save] to save the plugin to the pool, and [Reload] to refresh the script after editing the user variables. *Note: Any time cue numbers or their timecode stamps in the timecode track are change, the plugin engine **must** be reloaded.*
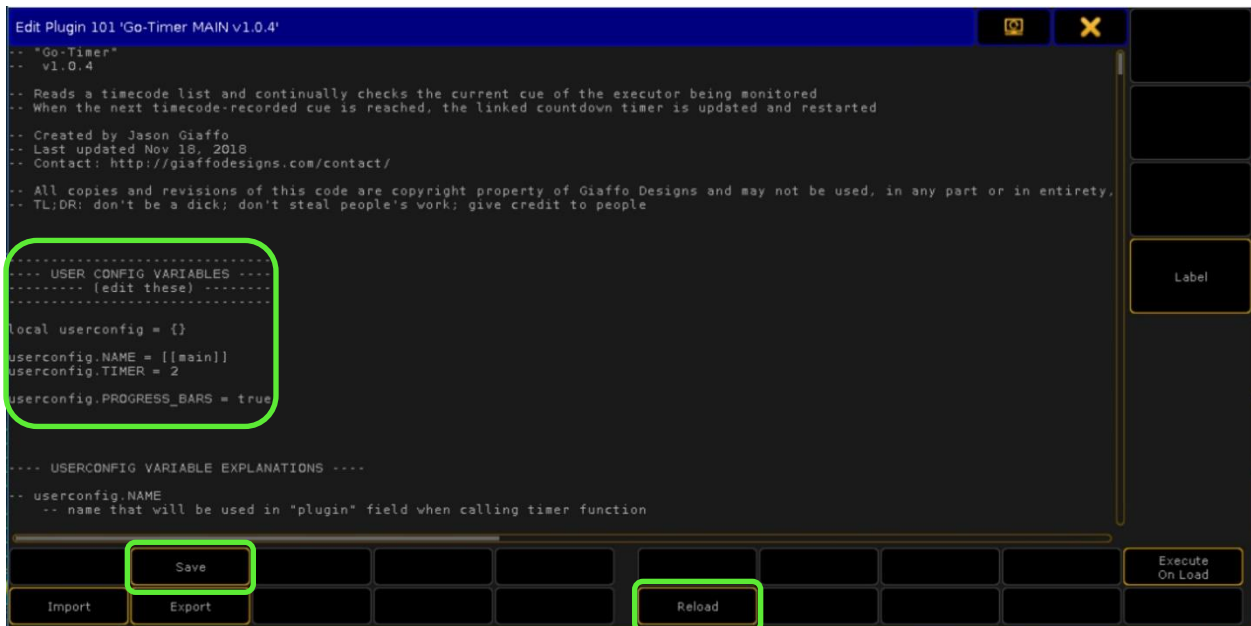

Fig 1: userconfig section, save button, and Reload button for plugin engine

Step 2: Build Trigger Macro

This plugin is triggered via a macro with a few variables that must be properly defined. This can be incorporated as one line in a song load macro of a show file or can exist as its own separate macro. The macro should look like the following:
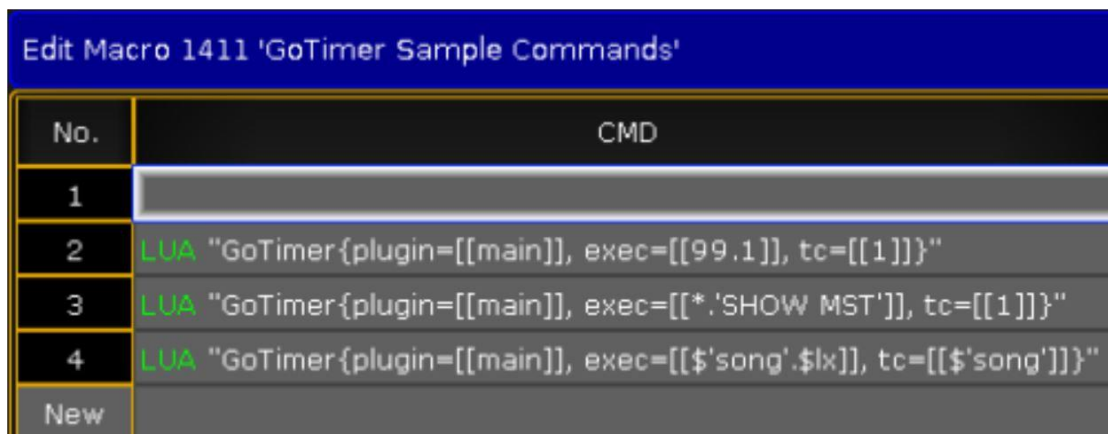
LUA "GoTimer{plugin=[[*userconfig.NAME*]], exec=[[*executor*]], tc=[[*timecode*]]}"

userconfig.NAME defines which plugin is being used for the timer (for instances when more than one timer plugin is running at once). This should be replaced with the label used in the **User Config Variables** section of the plugin script, for example [[**main**]].

executor defines which executor is being referenced for the plugin to calculate timing, and can be replaced with a few options depending on how the user wishes to use the plugin:
- To reference the executor by number, replace [[*executor*]] with the number of the sequence being timed, for example [[**99.1**]] if the cuestack is on Executor 1 of FaderPage 99.
- Text references to the executor's name may be used as well, just as in the command line. For example, if Executor 99.1 is on a page labelled "MAIN" and the executor is labelled "SHOW MST", replace [[*executor*]] with [[***.'SHOW MST'**]], [[**99.'SHOW MST'**]], [[**'MAIN'.1**]], or [[**'MAIN'.'SHOW MST'**]]. Text references must use single quotations rather than double quotations, as the LUA command is already contained within double quotations.
- Variable references may also be used as they would in the command line, replacing double quotes with singles, and *only using global variables.* UserVars will not be read by the plugin. To reference the executor with global variables, replace [[*executor*]] with [[**$var**]] or [[**$'var'**]]. Use [[**$var**]] for numeric values and [[**$'var'**]] for text values.

timecode defines which timecode track is being used to reference timing between cues. This parameter has the same options for replacing as *executor* does – referencing by number, name, or variable.

Fig. 2: sample commands for values for Macro call

Step 3: Create Timer

Open the Timer Pool and right click on the pool item that you defined in the **User Config Variables** section of the plugin script to create a new timer. Open a Clock window (located in the "Other" tab of the Create Basic Window view) and set the clock to view the desired Timer.

Step 4: Run Plugin With Show

Before triggering your first [GO] cue, run the macro that loads the plugin. After each [GO] of the defined cuelist, the timer should now run a countdown to the next [GO] recorded in the timecode sequence.

Fig. 3: plugin running with show

Advanced Use

Running More Than One Timer At Once

It is possible to run more than one timer at once. For example, it's possible to have a countdown for the main cuelist and a second timer that counts down an arbitrary cuelist for followspot calls. To do this, simply copy/paste the plugin into a second pool item and follow the normal instructions for timer setup a second time. *Note: Be sure to reload the plugin engine after updating the User Config Variables section of each plugin! See Fig. 1 for reference.*

Additional Option: *loadonly*

The *loadonly* option can be used in the plugin macro to load cue timings into the plugin memory before starting the plugin, to avoid delay when the timer starts. To utilize this function, create a second plugin macro with *loadonly=true* added to the end of the plugin macro line. It should look like this:

    LUA "GoTimer{plugin=[[userconfig.NAME]], exec=[[executor]], tc=[[timecode]], ***loadonly=true***}"

Run this macro to load cue timings into the plugin memory, then run the original macro when ready to start the timer.

Additional Option: *forceload*

Once the plugin is run in a show file, the MA2 Plugin Engine saves a copy of the cue numbers and their timecode stamps into a cache. When changes are made to cue numbers or to the timecode track being used by the plugin, it is necessary to update the information that was stored to memory. This can be accomplished by reloading the MA2 plugin engine. Alternatively, adding *forceload=true* to the end of the macro line will automatically reload this information every time the macro is run. It should look like this:

    LUA "GoTimer{plugin=[[userconfig.NAME]], exec=[[executor]], tc=[[timecode]], ***forceload=true***}"

FAQ

Do I need to send/receive timecode for this plugin to work?
Timecode signals are not needed for the Go-Timer plugin to operate. The plugin only uses timecode tracks as a reference to count down the time between each cue.

The timing between some of my cues isn't accurate!
Due to small changes in timing during a performance, including small tempo fluctuations in a song or longer talking breaks, some cues might be executed at different times from one night to the next.

If I accidentally [GO] a cue at entirely the wrong time, will it affect the countdown for the rest of the cues in my show?
No. The Go-Timer plugin counts down each individual [GO], rather than all cues as a whole, so future cue timing will remain the same.

For additional help and resources, please visit GiaffoDesigns.com.

Release Notes

To report any potential plugin bugs, please visit GiaffoDesigns.com/contact.