

Timer

Overview

Timer is a program for timing Combat Robot matches. Being highly customizable, it may also be useful for timing many other kinds of events. It can also control and be controlled by devices connected to the parallel port.

Disclaimer

This program is provided by Scott Ferguson and Forest Moon Productions as freeware as a public service, free of charge. If you use it, you assume responsibility for any and all outcomes of that use. You may not sell the package or include it in any other package without my written consent.

Comments, questions, bug reports and suggestions are encouraged.

Contact me at <http://www.ForestMoonProductions.com/Software>.

System Requirements and Installation

This program should install and operate properly on any Windows operating system capable of supporting the latest version of the Microsoft .NET Framework. See the [system requirements for the .NET Framework 2.0](#).

Installation can be accomplished at <http://www.ForestMoonProductions.com/Software>. All required components, including the .NET Framework, are installed as required.

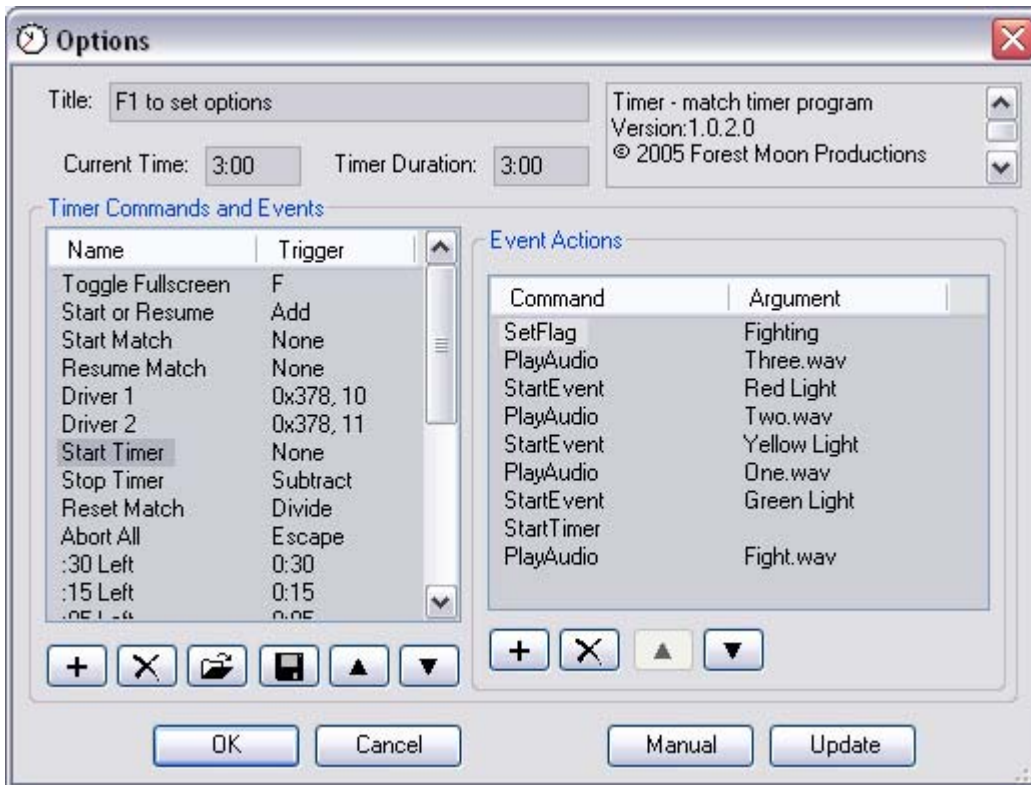
Timer installs and uses **Input32.dll** which is provided as freeware from **Logix4u.net** to access the parallel port on protected operating systems. For tutorial information about interfacing to the PC parallel port see <http://www.logix4u.net/parallelport1.htm>.

Getting Started

Clicking anywhere in the main window brings up a context menu listing all the current **Timer** commands and events showing the keystrokes or time values that trigger them. This list, except for the 'Set Options' command (F1), is entirely customizable.

When you first start the **Timer** it presents a default set of commands and events loaded from the "default.TOP" file. There is also a more sophisticated configuration, "Fancy.TOP", demonstrating (and requiring) the usage of the parallel port for input switches and to control lights.

The Options Dialog (F1)



The **Options** dialog (F1) is used to edit events. **Events** are the fundamental units for making things happen in the **Timer**. Events consist of an arbitrary **Name**, a **Trigger** and a list of **Actions**. Triggers cause events to be executed by associating them with a keyboard key, a time value or a printer port pin. An event with no trigger can only be executed using a **StartEvent** command. An action consists of a **Command** and a possible **Argument** for the command.

The left side of the dialog lists the existing events. The right side of the dialog lists the actions for the selected event. Select an event by left-clicking its name. Select an action by left-clicking its Command. Right-click any cell to change it. (Left-click also works to change triggers and arguments.) Buttons across the bottom of each section allow you to add and remove events and actions as well as move them up and down in the lists. All options can also be saved to or loaded from Timer options (.TOP) files. Clicking OK commits all options and also saves them to the system registry. (All options are loaded from the registry when the Timer program starts up.) Cancel abandons all changes made.

The Options Dialog is also used to set the title displayed, to change the current time (when a manual intervention is required) and to change the timer duration. The timer is set to this timer duration value by the [ResetTimer](#) action command. The timer duration value is also changed by the [SetDuration](#) action command.

If you create an Event with the special name **Startup** it will be executed when the timer program launches and when the Options dialog closes.

The **Manual** button displays this User Manual PDF file.

The **Update** button looks at ForestMoon.com for newer versions of this program.

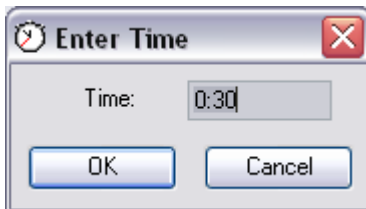
Triggers

Key Triggers



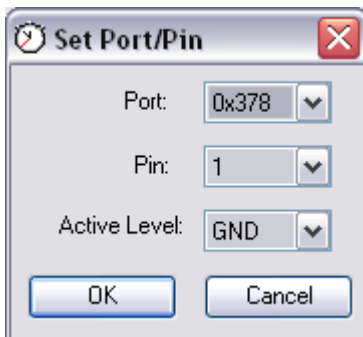
Selecting a key trigger for an event brings up this dialog. A single keystroke then maps that key to the event.

Time Triggers



Selecting a time trigger for an event brings up this dialog. Enter a value in "MM:SS" format. The associated event will be executed when the timer counts down to that value.

Input Triggers



Selecting an input trigger for an event brings up this dialog. Select a standard parallel (printer) port address from the list. The pins listed are the actual pin numbers on a standard DB25 parallel port connector. Select the active level for the signal as "GND" or "OPEN". The event will execute when the specified input port pin reaches the specified level. A simple "normally-open" switch can be connected between the input pin and one of the ground pins 18 thru 25. The GND level is achieved when the button is pushed. More sophisticated electronics can also be connected to the inputs, if you know what you're doing.

Actions

Actions are executed in the order as listed. With the exception of the [PlayAudio](#) command, actions execute until the end of the list before other events are allowed to execute. The event's action list resumes execution when the audio completes. Meanwhile, other events may be triggered and begin execution. Actions consist of one of a list of built-in commands and a possible argument.

StartTimer

Starts the timer.

StopTimer

Stops the timer.

ResetTimer

Stops the timer and sets the current time to the duration specified in the Options dialog.

SetDuration <time>

Sets the default timer duration. Follow with a [ResetTimer](#) to actually change the timer's current display.

SetColor <color>

Sets the background color, which can provide visual cues regarding the time remaining or if the timer is stopped or about to resume counting, etc. The background color is not used when a background image is set.

SetTimeColor <color>

Sets the text color used to display the time.

SetTitleColor <color>

Sets the text color used to display the title.

SetBackground <file>

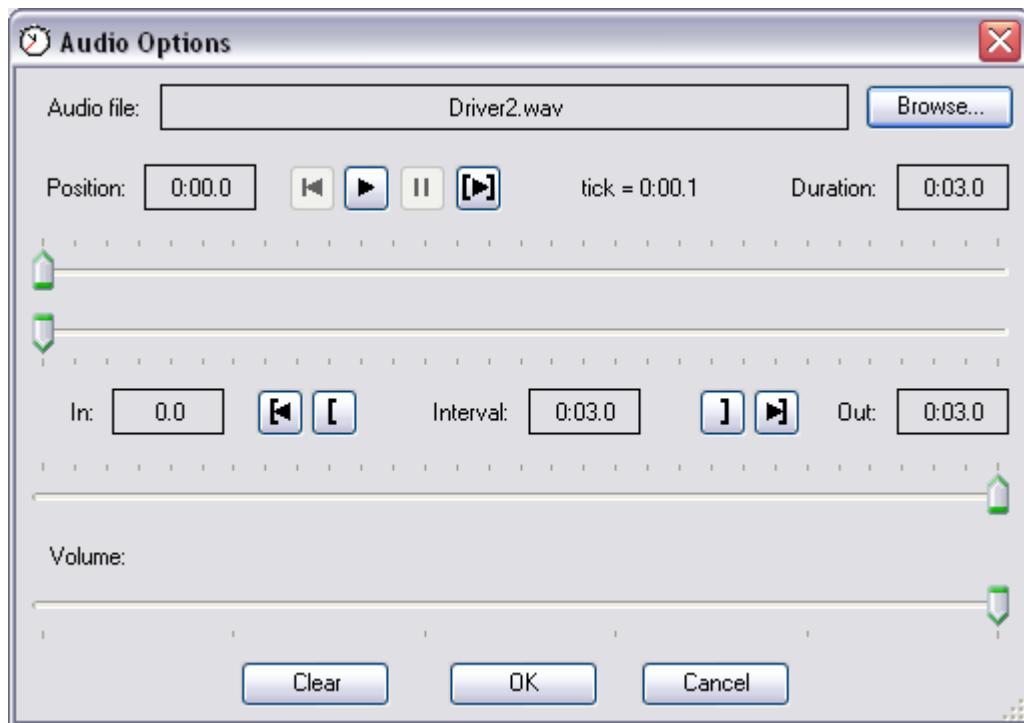
Applies the image file as a background for the timer by stretching it to fill the window. The background color is not used when a background image is set.

ClearBackground

Clears the background image.

PlayAudio <file>

Plays a WAV or MP3 audio file. This command can be used to create a playlist of background music. Just create an event assigned to a key trigger and make the playlist as a series of PlayAudio actions to play sound files. To stop a playlist from playing, you can create another event assigned to another key trigger to stop the event that contains your playlist.



The Audio Options dialog lets you select an audio file and set its play parameters (in point, out point and volume).

- Use the **position** slider to set the current time position.
- Use the **in** and **out** sliders to set the in and out points directly.
- Use buttons to set the **in** [and **out**] points from the current play position.
- Use buttons to set the current time position from the **in** ⏮ and **out** ⏭ points.
- Use the play clip button ▶ to play the audio from the in point to the out point.
- Use the **volume** slider to set the volume directly.

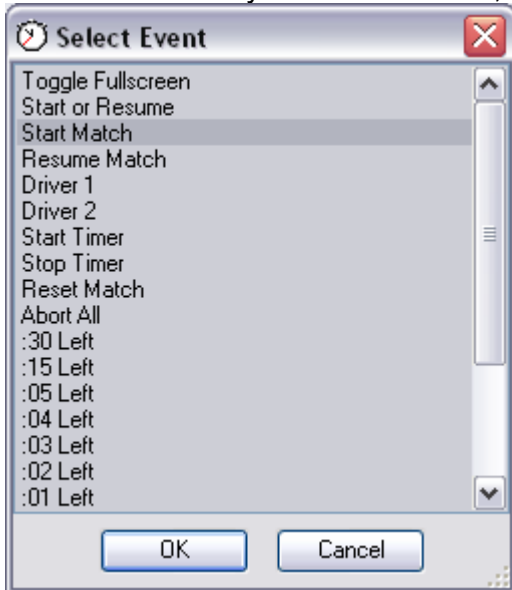
Only the clip from the in point to the out point will play when the command runs in an event.

ToggleFullscreen

Toggles fullscreen versus windowed mode. When not in fullscreen mode you can size, move and close the window using its title bar. You can always close the window using Alt+F4.

StartEvent <event name>

Starts an event which runs immediately. When the event completes or encounters a PlayAudio command, the event starting it continues running.



The Select Event dialog lets you select the name of an event to execute.

StopEvent <event name>

Stops an event. (Truly effective only if the event has audio playing.)

StopAllEvents

Performs a Stop Event on all running events. (Including the event containing this command!)

ExitIfTimerRunning

Exits the event if the timer is running. Can be used to disallow commands you don't want to run while the timer is running.

ExitIfTimerNotRunning

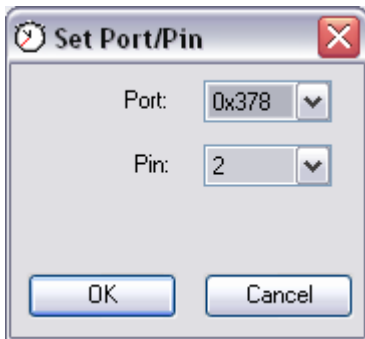
Exits the event if the timer is not running. Can be used to disallow commands you don't want to run while the timer is not running.

ExitIfTimerZero

Exits the event if the timer is zero. Used to determine if the timer duration has elapsed.

SetOutput <port/pin>

Sets a parallel (printer) port output pin to a high voltage level.



The Set Port/Pin dialog is used to select the standard parallel port address and DB25 pin number. These lines might be used to drive LEDs directly, but to do anything useful, like controlling incandescent lights, requires more sophisticated electronics.

ClearOutput <port/pin>

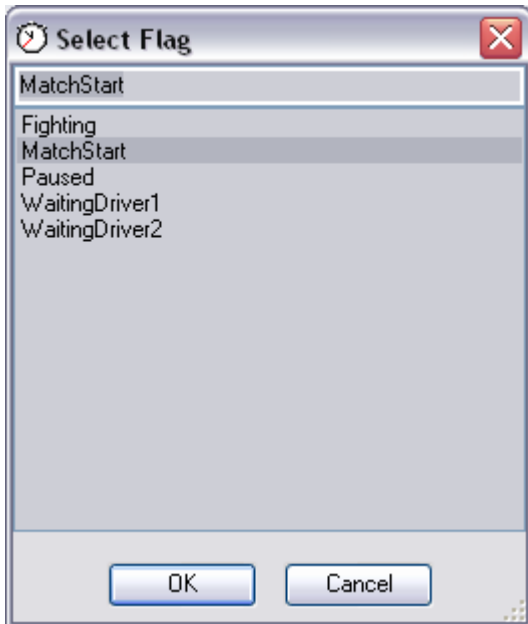
Clears a parallel (printer) port output pin to a low voltage level.

ToggleOutput <port/pin>

Toggles a parallel (printer) port output pin level.

SetFlag <flag name>

Sets a flag. A flag is given an arbitrary name and can be used to control execution of an event's action list using the [ExitIfFlagSet](#) and [ExitIfFlagClear](#) commands.



The Select Flag dialog allows you to select an existing flag name from the list or type in a new flag name.

ClearFlag <flag name>

Clears a flag.

ToggleFlag <flag name>

Toggles a flag.

ClearAllFlags

Clears all flags.

ExitIfFlagSet

Exits the event if the flag is set.

ExitIfFlagClear

Exits the event if the flag is clear.

Sample Configuration

This section describes the configuration contained in the 'Fancy.TOP' Timer options file. This configuration demonstrates the usage of the parallel port for driver input buttons and to control countdown lights. It uses flags to define a set of states for the system and events to cause transitions between those states.

States

MatchStart

The **Reset Match** event sets this state flag so the system is waiting for the **Add** (numeric pad '+') key to execute the **Start Match** event.

WaitingDriver1

The **Start Match** event sets this flag after announcing Driver #1 so the system is waiting for the driver to push his button (connected to parallel port pin 10) to execute the **Driver 1** event.

WaitingDriver2

The **Driver 1** event sets this flag after announcing Driver #2 so the system is waiting for the driver to push his button (connected to parallel port pin 11) to execute the **Driver 2** event.

Fighting

The **Start Timer** event sets this flag when it starts the timer, indicating we're in fighting mode.

Paused

Pressing the **Subtract** (numeric keypad '-') key executes the **Stop Timer** event, setting this flag to indicate that the match is paused.

Events

Toggle Fullscreen

This event simply toggles from windowed to fullscreen mode.

Start or Resume

This event is invoked with the **Add** key and starts a match or resumes a paused match. All the events with triggers must determine if they're invoked in an appropriate state before taking action. So this event will exit if the timer is already running or has elapsed. It then starts **both** the **Start Match** and **Resume Match** events. Since these events also evaluate their state, at most only one of them will take action.

Start Match

This event exits and takes no action if the **MatchStart** state flag is not set. It then clears that state and sets the **WaitingDriver1** state flag and announces Driver #1 by playing an audio file.

Resume Match

This event exits and takes no action if the **Paused** state flag is not set. It then clears that state and starts the **Start Timer** event.

Driver 1

This event is invoked by Driver #1 pressing his button connected to parallel port pin 10. It exits and takes no action if the **WaitingDriver1** state flag is not set. It stops the **Start Match** event since it may still be playing the audio announcing the driver when the button is pushed. It then clears the **WaitingDriver1** state flag and sets the **WaitingDriver2** state flag and announces Driver #2 by playing an audio file.

Driver 2

This event is invoked by Driver #2 pressing his button connected to parallel port pin 11. It exits and takes no action if the **WaitingDriver2** state flag is not set. It stops the **Driver 1** event since it may still be playing the audio announcing the driver when the button is pushed. It then clears that state and starts the **Start Timer** event.

Start Timer

This event performs the countdown and starts the timer. It uses audio files to announce the countdown stages and relies on the time duration of those audio clips to set the timing for actions in the event. It uses other events, such as **Red Light**, to set the visual cues for the countdown.

Stop Timer

This event is invoked with the **Subtract** (numeric keypad '-') key and allows a match in progress to be manually stopped. It exits and takes no action if the **Fighting** state flag is not set. It then clears that state, sets the **Paused** state flag, uses the **Red Light** event to set the visual cues and plays an audio announcement.

Reset Match

This event is invoked with the **Divide** (numeric keypad '/') key and resets the system for the next match. It exits and takes no action if the **Fighting** state flag is set, so that it will not accidentally stop a match in progress. It resets the timer, clears all state flags, stops all events that might be playing audio, turns on the red light and sets the **MatchStart** state flag. **You need to invoke this event before you can start the first match.**

Abort All

This event is invoked with the **Esc** key. It clears all flags and stops all events. This is mostly useful while developing events.

:xx Left

A series of events with time triggers has also been created here to give warning cues as the match timer hits 30 and 15 seconds remaining and a countdown from 5 seconds to 1 second remaining.

Match Over

This event is invoked when the match timer reaches zero. It sets the **MatchStart** state flag, sets the red lighting cues and plays an audio announcement.

3:00 match & 5:00 match

These events are invoked with the '3' key and '5' key respectively to set the timer duration for a 3 or 5 minute match. It exits and takes no action if the **MatchStart** state flag is not set, so it will not work while a match is in progress.

Red / Yellow / Green Light

These events set the visual clues for the match state. They use pins 2, 3 and 4 of the parallel port to drive electronics to control red, yellow and green lights. The macros also set the background to the same color. They have no triggers; they are only invoked from other events.

Revision History

Version 1.0.4 **5/09/2007**

- Added new Actions:
 - SetBackground
 - ClearBackground
 - SetTimeColor
 - SetTitleColor
- Added invocation of Startup event.

Version 1.0.3 **5/08/2007**

- Added standard MSI install option.

Version 1.0.2 **12/04/2005**

- Changed to “ClickOnce” deployment installation.
- Added “Update” button to Options dialog to check for application updates.
- Changed audio playback support from DirectX AudioVideoPlayback to Windows Media Player, eliminating dependency on DirectX.
- Update to .NET Framework 2.0.
- Changed to XP style buttons

Version 1.0.1 **11/29/2005**

- Added support for parallel port I/O.
 - Set/Clear/ToggleOutput commands
 - Input triggers
- Added support for flags
 - Set/Clear/ToggleFlag and ClearAllFlags commands
 - ExitIfFlagSet/Clear commands