

Timer

COLLABORATORS

	<i>TITLE :</i> Timer		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 26, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Timer	1
1.1	Timer	1
1.2	allocatetimer	1
1.3	freetimer	2
1.4	inittimer	2
1.5	starttimer	2
1.6	stoptimer	2

Chapter 1

Timer

1.1 Timer

PureBasic - Timer

This timer library is a good way to get very short but accurate elapsed time between two part of your program. If you need to know how many time take a subroutine to be performed, this library is the only way. The calculated time can never be superior to one frame (1/50 of secs). This function is not dependent of the processor so the results should be the same on slow and fast Amigas.

Commands summary:

```
AllocateTimer  
FreeTimer  
InitTimer  
StartTimer  
StopTimer
```

Timer Demo

1.2 allocatetimer

SYNTAX

```
Result.b = AllocateTimer()
```

FUNCTION

This function try to allocate one CIA timer.

Don't call it more than once without a call to FreeTimer() in between.

Result

TRUE if a timer is allocated else FALSE.

1.3 freetimer

SYNTAX

```
FreeTimer()
```

STATEMENT

To free the CIA timer which is allocated with `AllocateTimer()`, then call this statement.

1.4 inittimer

SYNTAX

```
InitTimer()
```

STATEMENT

Always call this statement before any other routines in `Timer Lib`.

1.5 starttimer

SYNTAX

```
StartTimer()
```

STATEMENT

This statement starts the timer.

1.6 stoptimer

SYNTAX

```
Result.w = StopTimer()
```

FUNCTION

This function stops the timer.

Result

If this is -1 then the time taken is greater or equal to the time of one frame. That means it is out of range for the timing purpose done by `Timer Lib`. If it instead is a positive value it should be multiplied with 1.39 to convert it to microseconds.
