

**pTimerD**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> pTimerD		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		March 29, 2025	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>pTimerD</b>	<b>1</b>
1.1	pTimerD.doc . . . . .	1
1.2	ptimer.device/TimerAllgemeines() . . . . .	1
1.3	ptimer.device/pOS_AddTime() . . . . .	1
1.4	ptimer.device/pOS_GetSysTime() . . . . .	2
1.5	ptimer.device/pOS_SubTime() . . . . .	2
1.6	ptimer.device/pOS_CmpTime() . . . . .	3

# Chapter 1

## pTimerD

### 1.1 pTimerD.doc

ptimer.device

pOS\_AddTime()      pOS\_CmpTime()    pOS\_GetSysTime()    pOS\_SubTime()  
TimerAllgemeines()

### 1.2 ptimer.device/TimerAllgemeines()

STRUKTUREN

```
struct pOS_TimerDevice;  
struct pOS_TimeVal;  
struct pOS_TimerUnit;  
struct pOS_TimerIO;
```

INCLUDES

```
Device/Timer.h  
proto/pTimer.h
```

BESCHREIBUNG

### 1.3 ptimer.device/pOS\_AddTime()

PROTOTYP

```
VOID pOS_AddTime  
(  
    pOS_TimerDevice *timerdevice,  
    pOS_TimeVal *timeval1,  
    const pOS_TimeVal *timeval2  
);
```

## FUNKTION

Addieren zweier Zeitwerte,  
timevall = timevall + timeval2

## PARAMETER

timerdevice (\_R\_LB)  
Zeiger auf pOS\_TimerDevice  
timevall (\_R\_A0)  
Zeiger auf timeval-Struktur, die auch das Ergebnis enthält  
timeval2 (\_R\_A1)  
Zeiger auf zweite timeval-Struktur, die zu timevall addiert wird

## ERGEBNIS

timevall enthält das Ergebnis timevall + timeval2.

## SIEHE AUCH

pOS\_SubTime()

## AMIGA FUNKTION

void AddTime(struct timeval \*,struct timeval \*);

## 1.4 ptimer.device/pOS\_GetSysTime()

## PROTOTYP

```
VOID pOS_GetSysTime  
(  
    pOS_TimerDevice *timerdevice,  
    pOS_TimeVal *timeval  
);
```

## FUNKTION

Ermitteln der aktuellen Systemzeit.

## PARAMETER

timerdevice (\_R\_LB)  
Zeiger auf pOS\_TimerDevice  
timeval (\_R\_A0)  
Zeiger auf TimeVal-Struktur, die gefüllt werden soll.  
Die Zeitrechnung beginnt beim 1.1.1992 mit 0.

## AMIGA FUNKTION

void GetSysTime(struct timeval \*);

## 1.5 ptimer.device/pOS\_SubTime()

## PROTOTYP

```
VOID pOS_SubTime  
(  
    pOS_TimerDevice *timerdevice,  
    pOS_TimeVal *timevall,
```

```
        const pOS_TimeVal *timeval2
    );

FUNKTION
    Subtrahieren zweier Zeitwerte,
    timevall = timevall - timeval2

PARAMETER
    timerdevice (_R_LB)
        Zeiger auf pOS_TimerDevice
    timevall (_R_A0)
        Zeiger auf timeval-Struktur, die auch das Ergebnis enthält
    timeval2 (_R_A1)
        Zeiger auf zweite timeval-Struktur, die von timevall subtrahiert wird

ERGEBNIS
    timevall enthält das Ergebnis timevall - timeval2.

SIEHE AUCH
    pOS_AddTime()

AMIGA FUNKTION
    void SubTime(struct timeval *,struct timeval *);
```

## 1.6 ptimer.device/pOS\_CmpTime()

```
PROTOTYP
    SLONG result = pOS_CmpTime
    (
        pOS_TimerDevice *timerdevice,
        const pOS_TimeVal *timevall,
        const pOS_TimeVal *timeval2
    );

FUNKTION
    Vergleichen zweier Zeitwerte.

PARAMETER
    timerdevice (_R_LB)
        Zeiger auf pOS_TimerDevice
    timevall (_R_A0)
        Zeiger auf timeval-Struktur, die auch das Ergebnis enthält
    timeval2 (_R_A1)
        Zeiger auf zweite timeval-Struktur, die von timevall subtrahiert wird

ERGEBNIS
    result (_R_D0)
        -1, wenn timevall später als timeval2 ist
        +1, wenn timevall früher als timeval2 ist
        0, falls beide Zeitwerte gleich sind.

AMIGA FUNKTION
    LONG CmpTime(struct timeval *,struct timeval *);
```

---

END