

pmbsonline

COLLABORATORS

	<i>TITLE :</i> pmbsonline		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 22, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	pmbsonline	1
1.1	pmbsonline.doc	1
1.2	pmbsonline.library/pt_OpenPrometheus()	1
1.3	pmbsonline.library/pt_ClosePrometheus()	2
1.4	pmbsonline.library/pt_Send()	2
1.5	pmbsonline.library/pt_Read()	3
1.6	pmbsonline.library/pt_BlueRead()	3
1.7	pmbsonline.library/pt_Quit()	4
1.8	pmbsonline.library/pt_AnsiStatus()	5
1.9	pmbsonline.library/pt_AskUser()	5
1.10	pmbsonline.library/pt_AskLog()	6
1.11	pmbsonline.library/pt_AskBoxname()	6
1.12	pmbsonline.library/pt_ReceiveFile()	7
1.13	pmbsonline.library/pt_Sendfile()	7
1.14	pmbsonline.library/pt_CheckCarrier()	8
1.15	pmbsonline.library/pt_Typefile()	9
1.16	pmbsonline.library/pt_GetChar()	9
1.17	pmbsonline.library/pt_Command()	10
1.18	pmbsonline.library/pt_Arexx()	10

Chapter 1

pmbsonline

1.1 pmbsonline.doc

pmbsonline.library

pt_OpenPrometheus	Öffnet den Prometheusport
pt_ClosePrometheus	Schliesst die Prometheuschnittstelle
pt_Send	Sendet einen String ueber das Modem und Bildschirm
pt_Read	Liest beliebig viele Zeichen von der Tastatur ein.
pt_BlueRead	Liest beliebig viele Zeichen von der Tastatur ein.
pt_Quit	Signalisiert PMBS Schliessung des Programmes.
pt_AnsiStatus	Prueft ob der User Ansi anhat
pt_AskUser	Liefert den Namen eines eingeloggten Users.
pt_AskLog	Fragt nach Consolen oder Modemlogin.
pt_AskBoxname	Fragt nach dem Namen der Mailbox.
pt_ReceiveFile	Empfaengt ein File ueber das Modem.
pt_Sendfile	Sendet ein File ueber das Modem.
pt_CheckCarrier	Prueft ob ein Carrier besteht.
pt_Typefile	Sendet ein TextFile ueber das Modem.
pt_GetChar	Liefert sofort eine gedruckte Taste zurueck.
pt_Command	Fuehrt einen Prometheus Befehl aus.
pt_Arexx	Fuehrt einen Promehteus-ArexxBefehl aus.

1.2 pmbsonline.library/pt_OpenPrometheus()

NAME

pt_OpenPrometheus - Öffnet den Prometheusport

SYNOPSIS

```
PMBSPort = pt_OpenPrometheus (Portnummer)
                        d0
```

```
LONG pt_OpenPrometheus (LONG);
```

FUNCTION

Diese Routine öffnet den Port zu Prometheus.
Dies muss immer gemacht werden, wenn Portbefehle genutzt werden.

INPUTS

Portnummer -- Eine Nummer von 0-9 (0 = Mainport)

RESULTS

PMBSPort -- Adresse des Prometheusportes.

EXCEPTIONS

SEE ALSO

```
pt_ClosePrometheus();
```

BUGS

Diese Funktion gibt immer eine Adresse zurueckt, auch bei einem FAIL

1.3 pmbsonline.library/pt_ClosePrometheus()

NAME

pt_ClosePrometheus - Schliesst die Prometheusschnittstelle

SYNOPSIS

```
pt_ClosePrometheus (PMBSPort)
                        d0
```

```
void pt_ClosePrometheus (LONG);
```

FUNCTION

Diese Routine schliesst den mit `pt_OpenPrometheus()` geöffneten `PrometheusPort`. Dies muss immer am ende gemacht werden.

INPUTS

PMBSPort -- Adresse des Prometheusportes.

RESULTS

EXCEPTIONS

SEE ALSO

```
pt_OpenPrometheus();
```

BUGS

1.4 pmbsonline.library/pt_Send()

NAME _____

pt_Send - Sendet einen String ueber das Modem und Bildschirm

SYNOPSIS

```
pt_Send(PMBSPort, String)
           d0          a0
```

```
void pt_Send(LONG, char *);
```

FUNCTION

Diese Routine senden den Text mit Umlautwandlung auf den Bildschirm und ueber das Modem.

INPUTS

PMBSPort -- Adresse des Prometheusportes.
String -- Pointer auf einen mit 0 abgeschlossenen String.

RESULTS

EXCEPTIONS

SEE ALSO

BUGS

1.5 pmbsonline.library/pt_Read()

NAME

pt_Read - Liest beliebig viele Zeichen von der Tastatur ein.

SYNOPSIS

```
pt_Read(PMBSPort, Zeichen, Buffer)
          d0          a1          a0
```

```
void pt_Read(LONG, LONG, char *);
```

FUNCTION

Mit dieser Funktion kann man beliebig viele Zeichen eingeben lassen. Das Ergebnis im Buffer ist IMMER mit einem 0-Byte abgeschlossen.

INPUTS

PMBSPort -- Adresse des Prometheusportes.
Zeichen -- Anzahl der einzugebenen Zeichen.
Buffer -- Pointer auf einen Bereich, in dem der Text mit 0 abgeschlossen geschrieben wird

RESULTS

Siehe Buffer

EXCEPTIONS

SEE ALSO

pt_BlueRead(), pt_GetChar();

BUGS

1.6 pmbsonline.library/pt_BlueRead()

NAME

pt_BlueRead - Liest beliebig viele Zeichen von der Tastatur ein.

SYNOPSIS

```
pt_BlueRead(PMBSPort, Zeichen, Buffer)
           d0          a1          a0
```

```
void pt_BlueRead(LONG, LONG, char *);
```

FUNCTION

Mit dieser Funktion kann man beliebig viele Zeichen eingeben lassen.
Das Ergebnis im Buffer ist IMMER mit einem 0-Byte abgeschlossen.
Der Bereich, in dem der Text eingegeben wird ist Blau unterlegt.

INPUTS

PMBSPort -- Adresse des Prometheusportes.
Zeichen -- Anzahl der einzugebenen Zeichen.
Buffer -- Pointer auf einen Bereich, in dem der Text mit 0
abgeschlossen geschrieben wird

RESULTS

Siehe Buffer

EXCEPTIONS

SEE ALSO

pt_Read(), pt_GetChar();

BUGS

1.7 pmbsonline.library/pt_Quit()

NAME

pt_Quit - Signalisiert PMBS Schliessung des Programmes.

SYNOPSIS

```
pt_Quit(PMBSPort)
           d0
```

```
void pt_Quit(LONG);
```

FUNCTION

Diese Funktion signalisiert Prometheus, dass das externe Programm
beendet ist. Sie muss vor pt_ClosePrometheus() aufgerufen werden.

INPUTS

PMBSPort -- Adresse des Prometheusportes.

RESULTS

EXCEPTIONS

SEE ALSO
pt_ClosePrometheus();

BUGS

1.8 pmbsonline.library/pt_AnsiStatus()

NAME

pt_AnsiStatus - Prueft ob der User Ansi anhat

SYNOPSIS

```
Emulation = pt_AnsiStatus(PMBSPort)
                d0
```

```
LONG pt_AnsiStatus(LONG);
```

FUNCTION

Hiermit kann man pruefen, ob der eingeloggte User die Ansi-Emulation angeschaltet hat.

INPUTS

PMBSPort -- Adresse des Prometheusportes.

RESULTS

Emulation -- 0 == Ansi aus, 1 == Ansi an.

EXCEPTIONS

SEE ALSO

BUGS

1.9 pmbsonline.library/pt_AskUser()

NAME

pt_AskUser - Liefert den Namen eines eingeloggten Users.

SYNOPSIS

```
pt_AskUser(PMBSPort, Buffer)
                d0          a0
```

```
void pt_AskUser(LONG, char *);
```

FUNCTION

Diese Funktion liefert den Usernamen desjenigen, der das Programm aufgerufen hat.

INPUTS

PMBSPort -- Adresse des Prometheusportes.

Buffer -- Pointer auf einen Bereich, in dem der Text mit 0

abgeschlossen geschrieben wird

RESULTS

Siehe Buffer

EXCEPTIONS

SEE ALSO

BUGS

1.10 pmbsonline.library/pt_AskLog()

NAME

pt_AskLog - Fragt nach Consolen oder Modemlogin.

SYNOPSIS

```
Logtyp = pt_AskLog(PMBSPort)
          d0
```

```
LONG pt_AskLog(LONG);
```

FUNCTION

Mit dieser Funktion kann man erfragen ob der User sich von der Console eingeloggt hat, oder mit dem Modem.

INPUTS

PMBSPort -- Adresse des Prometheusportes.

RESULTS

Liefert 0 wenn Consolenlogin
Liefert 1 wenn Modemlogin

EXCEPTIONS

SEE ALSO

BUGS

1.11 pmbsonline.library/pt_AskBoxname()

NAME

pt_AskBoxname - Fragt nach dem Namen der Mailbox.

SYNOPSIS

```
pt_AskLog(PMBSPort, Buffer)
          d0
```

```
void pt_AskLog(LONG, char *);
```

FUNCTION

Diese Funktion liefert in Buffer den Namen der Mailbox.
Buffer ist mit einem 0-Byte abgeschlossen.

INPUTS

PMBSPort -- Adresse des Prometheusportes.

RESULTS

Buffer -- Name der Mailbox mit einem 0-Byte abgeschlossen.

EXCEPTIONS

SEE ALSO

BUGS

1.12 pmbsonline.library/pt_ReceiveFile()

NAME

pt_ReceiveFile - Empfaengt ein File ueber das Modem.

SYNOPSIS

```
pt_ReceiveFile(PMBSPort)
                d0
```

```
void pt_ReceiveFile(LONG);
```

FUNCTION

Diese Funktion empfaengt ein oder mehrere Files ueber das Modem in das aktuelle Verzeichniss.
Die/den Filenamen muss der Programmierer selber herausbekommen.
Es empfiehlt sich vorher mit dem Prometheus-Arexxbefehl SETCD in ein leeres Verzeichniss zu wechseln.

INPUTS

PMBSPort -- Adresse des Prometheusportes.

RESULTS

EXCEPTIONS

SEE ALSO

pt_Sendfile()

BUGS

1.13 pmbsonline.library/pt_Sendfile()

NAME

pt_Sendfile - Sendet ein File ueber das Modem.

SYNOPSIS

```
pt_Sendfile(PMBSPort, Filename)
           d0          a0
```

```
void pt_Sendfile(LONG, char *);
```

FUNCTION

Diese Funktion sendet ein File an den User.

INPUTS

```
PMBSPort  --  Adresse des Prometheusportes.
Filename  --  Ein Pointer auf einen mit einem 0-Byte abgeschlossenen
               Filenamem.
```

RESULTS

EXCEPTIONS

SEE ALSO

```
pt_ReceiveFile()
```

BUGS

1.14 pmbsonline.library/pt_CheckCarrier()

NAME

pt_CheckCarrier - Prueft ob ein Carrier besteht.

SYNOPSIS

```
Carrier pt_CheckCarrier(PMBSPort)
           d0
```

```
LONG pt_CheckCarrier(LONG);
```

FUNCTION

Mit dieser Funktion kann man ueberpruefen ob noch ein Carrier besteht.

INPUTS

```
PMBSPort  --  Adresse des Prometheusportes.
```

RESULTS

```
Carrier --  entaehlt eine 0, wenn kein Carrier besteht.
           entaehlt eine 1, wenn ein Carrier besteht.
```

EXCEPTIONS

In Carrier steht immer eine 0, wenn ein Consolenlogin besteht.

SEE ALSO

```
pt_AskLog()
```

BUGS

1.15 pmbsonline.library/pt_Typefile()

NAME

pt_Typefile - Sendet ein TextFile ueber das Modem.

SYNOPSIS

```
pt_Typefile(PMBSPort, Filename)
             d0          a1
```

```
void pt_Typefile(LONG, char *);
```

FUNCTION

TypeFile gibt den Inhalt eines TextFiles ueber die Serielle aus.
Beachtet wird hierbei unter anderem auch das Paging.

INPUTS

PMBSPort -- Adresse des Prometheusportes.
Filename -- Pointer auf einen mit einem 0-Byte abgeschlossenen
 Filennamen.

RESULTS

EXCEPTIONS

SEE ALSO

BUGS

1.16 pmbsonline.library/pt_GetChar()

NAME

pt_GetChar - Liefert sofort eine gedruckte Taste zurueck.

SYNOPSIS

```
pt_GetChar(PMBSPort, Buffer)
             d0          a0
```

```
void pt_GetChar(LONG, char *);
```

FUNCTION

Diese Funktion springt immer SOFORT zurueck! Wenn der User also nichts gedrueckt hat, wird auch nur ein 0 Byte zurueck gegeben! Diese Funktion ist also nur in einer Schleife sinnvoll! Sie liefert wenn ueberhaupt in den Buffer einen Buchstaben + 0 Byte zurueck. Wenn nichts gedrueckt wurde halt nur ein 0 Byte

INPUTS

PMBSPort -- Adresse des Prometheusportes.

RESULTS

Buffer -- Pointer auf einen Buffer, in dem Prometheus den eingegebenen Buchstaben schreibt. Buffer wird IMMER

mit einem 0-Byte abgeschlossen.

EXCEPTIONS

SEE ALSO

pt_Read(), pt_BlueRead()

BUGS

1.17 pmbsonline.library/pt_Command()

NAME

pt_Command - Fuehrt einen Prometheus Befehl aus.

SYNOPSIS

```
pt_Command(PMBSPort, String)
           d0      a0

void pt_Command(LONG, char *);
```

FUNCTION

Mit dieser Funktion ist es moeglich einen Prometheus eigenen Befehl auszufuehren (I *, B *).

INPUTS

PMBSPort -- Adresse des Prometheusportes.

String -- Pointer auf einen mit 0 abgeschlossenen String Dieser String muss ein Prometheus Befehl wie B * oder so etwas sein!

RESULTS

EXCEPTIONS

SEE ALSO

pt_Arexx() :

BUGS

1.18 pmbsonline.library/pt_Arexx()

NAME

pt_Arexx - Fuehrt einen Promehteus-ArexxBefehl aus.

SYNOPSIS

```
pt_Arexx(PMBSPort, String, Buffer)
           d0      a0      a1

void pt_Arexx(LONG, char *, char *);
```

FUNCTION

Mit dieser Funktion ist es moeglich die kompletten Prometheus eigenen Arexx-Befehle zu nutzen.

INPUTS

PMBSPort -- Adresse des Prometheusportes.
String -- Pointer auf einen mit einem 0-Byte abgeschlossenen Bereich, der als ArexxBefehl an Prometheus gesendet wird.

RESULTS

Buffer -- Pointer auf einen Bereich, nach dem Prometheus den Result des ArexxBefehles schreibt.

EXCEPTIONS

SEE ALSO

pt_Command() :

BUGS