Application Programmer's Guide

Solaris Drivers for TPRO/TSAT-PCI

Edition 1.4

KSI

a division of DSPCon, Inc.

For technical support, contact KSI Corporation (866) KSI-KSI3

Solaris Driver APPLICATION PROGRAMMERS GUIDE
Edition 1.4 —September 10, 2001

A Publication of
KSI

KSI has worked to verify the accuracy of the information contained in this document as of its publication date; however, such information is subject to change without notice and KSI is not responsible for any errors that may occur in this document.

REVISION HISTORY

Revision Date	Revision No.	Revised Section(s)	Comments/Notes
June 1, 2001	1	——	First Edition
July 11, 2001	1.1	Caveats General	Added 32-bit driver uninstall instructions to the Caveats section Removed indications of this manual being applicable to only the 32-bit version Removed the note in the Caveats section about interrupts not being supported
July 12, 2001	1.2	Caveats	Chanced uninstall command to pkgrm tpro
August 7, 2001	1.3	Revision	Indicated that this was for both 32-bit and 64-bit Solaris Drivers
September 10, 2001	1.4	Revision	Fixed example bugs and modified the installation instructions

Table of Contents

Chapter One

Overview

This guide provides comprehensive information on the Solaris Driver for the KSI TPRO/TSAT-PCI board.

Introduction

The Driver for the KSI TPRO/TSAT-PCI Board provides functionality to the board's interfaces and devices.

There are two varieties of the KSI Solaris driver for the TPRO/TSAT-PCI boards.

The 32-bit variety supports all variations of Solaris.
The 64-bit variety supports only Solaris versions 2.7 (also known as Solaris 7) and higher.

Product Description

The TPRO/TSAT-PCI performs timing and synchronization functions referenced to an input timecode signal. The board synchronizes its on-board clock to the incoming timecode. The on-board clock's time is also provided as an IRIG-B output. The board includes a time-tag TTL input, a programmable "heartbeat" pulse or squarewave output (with interrupt capability), and a programmable "match" start/stop time output (with interrupt capability)

The TPRO/TSAT-PCI continues to increment time ("freewheel") in the absence of an input timecode. Thus, the board can be used as an IRIG-B timecode generator by setting the initial time via the PCI bus.

The input timecode format (IRIG-B, IRIG-A, or NASA36) is detected automatically. Synchronization to the input timecode is also automatic and can be enabled/disabled via the PCI bus. A propagation delay offset may be specified to compensate for cable delays.

The timecode input is an amplitude-modulated sine wave. An automatic gain control (AGC) circuit permits a wide range of input amplitudes. The timecode input is differential; the board does not reference this signal to ground. A single-ended input (referenced to ground) is also acceptable.

The board can be ordered with option "-M" to synchronize to a one-pulse-per-second (1PPS) input instead of an incoming timecode. In this case, the initial time is programmed via the PCI bus, and the board begins counting on the next 1 PPS pulse.

Caveats

Caution

- *If you have previously installed the 32-bit Solaris driver and are now attempting to install the 64-bit Solaris driver, you must first UNINSTALL the 32-bit version.*

  Type > pkgrm tpro in your Solaris shell to uninstall the 32-bit version

This page intentionally left blank.

Chapter Two

Installation and Application

Installing the Driver

In your shell, type pkgadd-d <path to CD-ROM drive> tpro.pkg

Uninstalling the Driver

In your shell, type pkgrm tpro

Application Example

```
#include "tpro.h"

main()
{
  TPRO_BoardObj *hnd;
  int rc;
  TPRO_TimeObj timep;

  rc = TPRO_open(&hnd, "/dev/tpro0");
  if (rc) {
    printf("Could Not open board!!\n");
    exit(1);
  }

  rc = TPRO_getTime(hnd, &timep);
  if (rc) {
    printf("Could Not retrieve time!!\n");
    exit(1);
  }

  printf("day %d, hours %d, minutes %d, seconds %f\n",
       timep.days, timep.hours, timep.minutes, timep.secsDouble);

  TPRO_close(hnd);
}
```

Chapter Three

The TPRO/TSAT-PCI Solaris Driver

Header File

The following is the “TPRO.H” Driver Interface Header File.

```
/***************************************************************************
  DSPCon TPRO/tSAT - Interface Header

               DSPCon, Inc.
               380 FootHill Road
               Bridgewater, NJ 08807

               e-mail: info@dspcon.com

  (C) Copyright 2001 DSPCon, Inc.  All rights reserved.
  Use of copyright notice is precautionary and does not imply publication
***************************************************************************/

/***************************************************************************
  TPRO.H
***************************************************************************/

#ifndef _defined_TPRO_
#define _defined_TPRO_

#ifdef __cplusplus
extern "C" {
#endif


#define DLL_EXPORT /* */


/*==========================================================================
                       SUPPORT CONSTANTS
==========================================================================*/


/**
***  Heartbeat constants
**/

#define SIG_PULSE      (0xE5)         // heartbeat is a pulse
#define SIG_SQUARE     (0xE7)         // heartbeat is a squarewave

#define SIG_NO_JAM     (0)            // start next cycle
#define SIG_JAM        (1)            // start immediately


/**
***  Match constants
**/

#define MATCH_TIME_START      (0)            // start time
#define MATCH_TIME_STOP       (1)            // stop time


/*==========================================================================
                       TPRO BOARD OBJECT
==========================================================================*/

typedef struct TPRO_BoardObj
{ /*----------------------------------------------------------------------*/
  int file_descriptor;
  unsigned short devid;
} /*----------------------------------------------------------------------*/
TPRO_BoardObj;
```

```
/*==========================================================================
                       TPRO ALTITUDE OBJECT
==========================================================================*/

typedef struct TPRO_AltObj
{ /*----------------------------------------------------------------------*/
  float         meters;                      /*-- meters --------------*/
} /*----------------------------------------------------------------------*/
TPRO_AltObj;


/*==========================================================================
                         TPRO DATE OBJECT
==========================================================================*/

typedef struct TPRO_DateObj
{ /*----------------------------------------------------------------------*/
  unsigned short year;                       /*-- year ----------------*/
  unsigned char  month;                      /*-- month ---------------*/
  unsigned char  day;                        /*-- day -----------------*/
} /*----------------------------------------------------------------------*/
TPRO_DateObj;


/*==========================================================================
                 TPRO LONGITUDE/LATTITUDE OBJECT
==========================================================================*/

typedef struct TPRO_LongLat
{ /*----------------------------------------------------------------------*/
  unsigned short degrees;                    /*-- degrees -------------*/
  float         minutes;                     /*-- minutes -------------*/
} /*----------------------------------------------------------------------*/
TPRO_LongObj, TPRO_LatObj;


/*==========================================================================
                        TPRO MATCH OBJECT
==========================================================================*/

typedef struct TPRO_MatchObj
{ /*----------------------------------------------------------------------*/
  unsigned char   matchType;                 /*-- start/stop time -----*/

  double         seconds;                    /*-- seconds -------------*/
  unsigned char   minutes;                   /*-- minutes -------------*/
  unsigned char   hours;                     /*-- hours ---------------*/
  unsigned short  days;                      /*-- days ----------------*/
} /*----------------------------------------------------------------------*/
TPRO_MatchObj;


/*==========================================================================
                       TPRO SATINFO OBJECT
==========================================================================*/

typedef struct TPRO_SatObj
{ /*----------------------------------------------------------------------*/
  unsigned char satsTracked;                 /*-- num sats tracked ----*/
  unsigned char satsView;                    /*-- num sats in view ----*/
} /*----------------------------------------------------------------------*/
TPRO_SatObj;


/*==========================================================================
                      TPRO HEARTBEAT OBJECT
==========================================================================*/

typedef struct TPRO_HeartObj
{ /*----------------------------------------------------------------------*/
  unsigned char signalType;                  /*-- square or pulse -----*/
  unsigned char outputType;                  /*-- jamming option ------*/

  double  frequency;                         /*-- heartbeat freq ------*/
} /*----------------------------------------------------------------------*/
TPRO_HeartObj;


/*==========================================================================
                         TPRO TIME OBJECT
==========================================================================*/

typedef struct TPRO_TimeObj
{ /*----------------------------------------------------------------------*/
  double         secsDouble;                 /*-- seconds floating pt -*/
  unsigned char  seconds;                    /*-- seconds whole num ---*/
  unsigned char  minutes;                    /*-- minutes -------------*/
  unsigned char  hours;                      /*-- hours ---------------*/
  unsigned short days;                       /*-- days ----------------*/
} /*----------------------------------------------------------------------*/
TPRO_TimeObj;


/*==========================================================================
                         TPRO WAIT OBJECT
==========================================================================*/

typedef struct TPRO_WaitObj
{ /*----------------------------------------------------------------------*/
  int ticks;                                 /*-- # ticks to wait -----*/
  unsigned char status;                      /*-- event status --------*/
} /*----------------------------------------------------------------------*/
TPRO_WaitObj;

/*==========================================================================
                  TPRO MEM OBJECT FOR PEEK/POKE
==========================================================================*/

typedef struct TPRO_MemObj
{ /*----------------------------------------------------------------------*/
  unsigned short offset;
  unsigned short value;
} /*----------------------------------------------------------------------*/
TPRO_MemObj;


/*==========================================================================
                    PUBLIC ROUTINE PROTOTYPES
==========================================================================*/

DLL_EXPORT
unsigned char TPRO_open             (TPRO_BoardObj **hnd, char *deviceName);

DLL_EXPORT
unsigned char TPRO_close          (TPRO_BoardObj *hnd);

DLL_EXPORT
unsigned char TPRO_getAltitude      (TPRO_BoardObj *hnd, TPRO_AltObj *Altp);

DLL_EXPORT
unsigned char TPRO_getDate          (TPRO_BoardObj *hnd, TPRO_DateObj *Datep);

DLL_EXPORT
unsigned char TPRO_getDriverVersion (char *version);

DLL_EXPORT
unsigned char TPRO_getFirmware      (TPRO_BoardObj *hnd, unsigned char *firmware);

DLL_EXPORT
unsigned char TPRO_getLatitude      (TPRO_BoardObj *hnd, TPRO_LatObj *Latp);

DLL_EXPORT
unsigned char TPRO_getLongitude     (TPRO_BoardObj *hnd, TPRO_LongObj *Longp);
```

```
DLL_EXPORT
unsigned char TPRO_getSatInfo       (TPRO_BoardObj *hnd, TPRO_SatObj *Satp);

DLL_EXPORT
unsigned char TPRO_getTime        (TPRO_BoardObj *hnd, TPRO_TimeObj *Timep);

DLL_EXPORT
unsigned char TPRO_resetFirmware    (TPRO_BoardObj *hnd);

DLL_EXPORT
unsigned char TPRO_setHeartbeat     (TPRO_BoardObj *hnd, TPRO_HeartObj *Heartp);

DLL_EXPORT
unsigned char TPRO_setMatchTime     (TPRO_BoardObj *hnd, TPRO_MatchObj *Matchp);

DLL_EXPORT
unsigned char TPRO_setPropDelayCorr (TPRO_BoardObj *hnd, int *us);

DLL_EXPORT
unsigned char TPRO_setTime          (TPRO_BoardObj *hnd, TPRO_TimeObj *Timep);

DLL_EXPORT
unsigned char TPRO_setYear          (TPRO_BoardObj *hnd, unsigned short *yr);

DLL_EXPORT
unsigned char TPRO_simEvent       (TPRO_BoardObj *hnd);

DLL_EXPORT
unsigned char TPRO_synchControl     (TPRO_BoardObj *hnd, unsigned char *enbp);

DLL_EXPORT
unsigned char TPRO_synchStatus      (TPRO_BoardObj *hnd, unsigned char *status);

DLL_EXPORT
unsigned char TPRO_waitEvent        (TPRO_BoardObj *hnd, TPRO_WaitObj *waitp);

DLL_EXPORT
unsigned char TPRO_waitMatch        (TPRO_BoardObj *hnd, TPRO_WaitObj *waitp);

DLL_EXPORT
unsigned char TPRO_peek                   (TPRO_BoardObj *hnd, TPRO_MemObj *Mem);

DLL_EXPORT
unsigned char TPRO_poke                   (TPRO_BoardObj *hnd, TPRO_MemObj *Mem);


#ifdef __cplusplus
}
#endif

#endif  // _defined_TPRO_
```

Support Routine Descriptions

TPRO_open

```
unsigned char TPRO_open  (TPRO_BoardObj **hnd, char *deviceName);
```

This routine opens the driver interface.

```
Arguments:  Pointer to TPRO_BoardObj handle
            Device name

Returns:    (0) Success
            (1) Error
```

TPRO_close

```
unsigned char TPRO_close (TPRO_BoardObj *hnd);
```

This routine closes the driver interface.

```
Arguments:  Pointer to TPRO_BoardObj

Returns:    (0)
```

TPRO_getAltitude

```
unsigned char TPRO_getAltitude (TPRO_BoardObj *hnd, TPRO_AltObj *Altp);
```

This routine retrieves the altitude information from the tSAT board. Altitude distance is in meters.

```
Arguments:  Pointer to TPRO_BoardObj
            Pointer to TPRO_AltObj

Returns:    (0) Success
            (1) Error
```

TPRO_getDate

unsigned char TPRO_getDate (TPRO_BoardObj *hnd, TPRO_DateObj *Datep);

This routine retrieves the current date from the TPRO/tSAT board. The date is in Gregorian Format.

```
Arguments:  Pointer to TPRO_BoardObj
            Pointer to TPRO_DateObj

Returns:    (0) Success
            (1) Error
```

TPRO_getFirmware

unsigned char TPRO_getFirmware(TPRO_BoardObj *hnd, unsigned char *firmware);

This routine retrieves the firmware revision programmed in the TPRO/tSAT device. It should be used for troubleshooting purposes only.

```
Arguments:  Pointer to TPRO_BoardObj
            pointer to unsigned char [version string]

Returns:    (0) Success
            (1) Error
```

TPRO_getLatitude

unsigned char TPRO_getLattitude(TPRO_BoardObj *hnd, TPRO_LatObj *Latp);

This routine retrieves the latitude information from the tSAT device.

```
Arguments:  Pointer to TPRO_BoardObj
            Pointer to TPRO_LatObj

Returns:    (0) Success
            (1) Error
```

TPRO_getLongitude

unsigned char TPRO_getLongitude(TPRO_BoardObj *hnd, TPRO_LongObj *Longp);

This routine retrieves the longitude information from the /tSAT device.

```
Arguments:  Pointer to the TPRO_BoardObj
            Pointer to the TPRO_LongObj

Returns:    (0) Success
            (1) Error
```

TPRO_getSatInfo

```
unsigned char TPRO_getSatInfo(TPRO_BoardObj *hnd, TPRO_SatObj *Satp);
```

This routine retrieves the number of satellites tracked from the tSAT device.

```
Arguments:   Pointer to the TPRO_BoardObj
             Pointer to the TPRO_SatObj

Returns:     (0) Success
             (1) Error
```

TPRO_getTime

```
unsigned char TPRO_getTime(TPRO_BoardObj *hnd, TPRO_TimeObj *Timep);
```

This routine retrieves the current time from the TPRO/tSAT device. The seconds value is received as type double.

```
Arguments:   Pointer to the TPRO_BoardObj
             Pointer to the TPRO_TimeObj

Returns:     (0) Success
             (1) Error
```

TPRO_resetFirmware

```
unsigned char TPRO_resetFirmware(TPRO_BoardObj *hnd);
```

This routine resets the firmware programmed on the TPRO/tSAT device. This function is for troubleshooting purposes only and should not be used in the main application.

```
Arguments:   Pointer to the TPRO_BoardObj

Returns:     (0) Success
             (1) Error
```

TPRO_setHeartbeat

```
unsigned char TPRO_setHeartbeat(TPRO_BoardObj *hnd, TPRO_HeartObj *Heartp);
```

This routine controls the hearbeat output. Heartbeat output can be either a square wave or pulse at various frequencies.

```
Arguments:   Pointer to the TPRO_BoardObj
             Pointer to the TPRO_HeartObj

Returns:     (0) Success
             (1) Error
```

TPRO_setMatchTime

```
unsigned char TPRO_setMatchTime(TPRO_BoardObj *hnd, TPRO_MatchObj *Matchp);
```

This routine is the conditional time to either start or stop on-board clock.

```
Arguments:  Pointer to the TPRO_BoardObj
            Pointer to the TPRO_MatchObj

Returns:    (0) Success
            (1) Error
```

TPRO_setPropDelayCorr

```
unsigned char TPRO_setPropDelayCorr (TPRO_BoardObj *hnd, int *us);
```

This routine sets the propagation delay correction factor.

```
Arguments:  Pointer to the TPRO_BoardObj
            Pointer to correction factor in microseconds

Returns:    (0) Success
            (1) Error
```

TPRO_setTime

```
unsigned char TPRO_setTime(TPRO_BoardObj *hnd, TPRO_TimeObj *Timep);
```

This routine sets the time on the on-board clock of the TPRO/tSAT device.

```
Arguments:  Pointer to the TPRO_BoardObj
            Pointer to the TPRO_TimeObj

Returns:    (0) Success
            (1) Error
```

TPRO_setYear

```
unsigned char TPRO_setYear(TPRO_BoardObj *hnd, unsigned short *yr);
```

This routine programs the TPRO/tSAT device with the desired year.

```
Arguments:  Pointer to the TPRO_BoardObj
            Pointer to the desired year

Returns:    (0) Success
            (1) Error
```

TPRO_simEvent

unsigned char TPRO_simEvent(TPRO_BoardObj *hnd);

This routine simulates an external time tag event.

```
Arguments:   Pointer to the TPRO_BoardObj

Returns:     (0) Success
             (1) Error
```

TPRO_synchControl

unsigned char TPRO_synchControl(TPRO_BoardObj *hnd, unsigned char *enbp);

This routine commands the TPRO/tSAT device to synchronize to input or freewheel. This distinction is made using the enable argument. If the enable arguments is (0) the clock will freewheel, otherwise it will synchronize to input.

```
Arguments:   Pointer to the TPRO_BoardObj
             Pointer to the synch enable

Returns:     (0) Success
             (1) Error
```

TPRO_synchStatus

unsigned char TPRO_synchStatus(TPRO_BoardObj *hnd, unsigned char *status);

This routine reports the synchronization status of the TPRO/tSAT device. When status is equal to zero, the device is freewheeling. Otherwise the device is synchronized to its input.

```
Arguments:   Pointer to the TPRO_BoardObj
             Pointer to the synch status variable

Returns:     (0) Success
             (1) Error
```

TPRO_waitEvent

unsigned char TPRO_waitEvent(TPRO_BoardObj *hnd, TPRO_WaitObj*waitp);

This routine waits for an external time tag event.

```
Arguments:   Pointer to the TPRO_BoardObj
             Pointer to wait time

Returns:     (0) Success
             (1) Error
```

TPRO_waitMatch

`unsigned char TPRO_waitMatch(TPRO_BoardObj *hnd, TPRO_WaitObj*waitp);`

This routine waits for a match time to occur

```
Arguments:  Pointer to the TPRO_BoardObj
            Pointer to wait time

Returns:    (0) Success
            (1) Error
```

This page intentionally left blank.

Warranty

KSI warrants that all products manufactured by KSI conform to published DSP*Con* specifications and are free from defects in materials and workmanship for a period of one (1) year from the date of delivery when used under normal conditions and within the service conditions for which they were furnished.

The obligation upon KSI arising from a warranty claim shall be limited to repairing, or, at its option, replacing without charge, any product which, in KSI's sole opinion, proves to be defective within the scope of this warranty.

KSI must be notified in writing of the defect or nonconformity within the warranty period, and the affected product must be returned to KSI within thirty (30) days after discovery of such defect or nonconformity.

The buyer shall prepay shipping charges, taxes, duties and insurance for products returned to KSI for warranty service. KSI shall pay for the return of products to buyer except for products returned to another country or from outside the forty-eight contiguous United States.

KSI shall have no responsibility for any defect or damage caused by improper installation, unauthorized modification, misuse, neglect, inadequate maintenance, accident or for any product that has been repaired or altered by anyone other than KSI or its authorized representatives.

The warranty described above is the buyer's sole and exclusive remedy and no other warranty, whether oral or written, is expressed or implied KSI specifically disclaims fitness for a particular purpose. Under no circumstances shall KSI be liable for any direct, indirect, special, incidental or consequential damages, expenses, losses or delays (including loss of profits) based on contract, tort, or any legal theory.

This warranty is valid for a period of one (1) year from the date of shipment. Upon warranty expiration, all services on KSI hardware and software are subject to a current hourly rate, plus travel expenses where applicable. As an option, KSI offers extended warranties when purchased within ten (10) days of receipt of product.

Extended Warranties

Hardware

KSI offers an extended warranty on its hardware at a cost of 15% of the current list price per year of coverage. This coverage is available for KSI hardware only.

Software

KSI offers an extended warranty on its software at a cost of 15% of the current list price per year of coverage. This coverage is available for KSI software only.

Service and Repair

Before returning a product for service and repair, please contact KSI at (866) KSI-KSI3 to obtain a Return Material Authorization (RMA) number. Have available the Model and Serial numbers of the item, a description of the problem, your company name, and the name and address of the person returning the product.

KSI

a division of DSP*Con*, Inc.

11209 Armour Drive
El Paso, TX 79935

Toll Free Phone: **866-KSI-KSI3**
Toll Free Fax: **866-593-2080**
Website: **www.ksi-corporation.com**

For software technical support, please contact:

For technical support, contact:
KSI Corporation
866-KSI-KSI3
pxd@ksi-corporation.com

A Publication of

KSI

© MMI

KSI has worked to verify the accuracy of the information contained in this document as of its publication date; however, such information is subject to change without notice and KSI is not responsible for any errors that may occur in this document.