TPRO-PMC /TSAT-PMC
SYNCHRONIZABLE TIMECODE GENERATOR with PMC BUS INTERFACE

VxWorks Driver
Application Programmer's Guide

95 Methodist Hill Drive
Rochester, NY 14623
Phone: US +1.585.321.5800
Fax: US +1.585.321.5219

www.spectracomcorp.com
Part Number 1153-5002-0050
Manual Revision A
28 April 2008

Specifications subject to change or improvement without notice.

SPECTRACOM LIMITED WARRANTY

LIMITED WARRANTY

Spectracom warrants each new product manufactured and sold by it to be free from defects in software, material, workmanship, and construction, except for batteries, fuses, or other material normally consumed in operation that may be contained therein AND AS NOTED BELOW, for five years after shipment to the original purchaser (which period is referred to as the "warranty period"). This warranty shall not apply if the product is used contrary to the instructions in its manual or is otherwise subjected to misuse, abnormal operations, accident, lightning or transient surge, repairs or modifications not performed by Spectracom.

The GPS receiver is warranted for one year from date of shipment and subject to the exceptions listed above. The power adaptor, if supplied, is warranted for one year from date of shipment and subject to the exceptions listed above.

THE ANALOG CLOCKS ARE WARRANTED FOR ONE YEAR FROM DATE OF SHIPMENT AND SUBJECT TO THE EXCEPTIONS LISTED ABOVE.

THE TIMECODE READER/GENERATORS ARE WARRANTED FOR ONE YEAR FROM DATE OF SHIPMENT AND SUBJECT TO THE EXCEPTIONS LISTED ABOVE.

The Rubidium oscillator, if supplied, is warranted for two years from date of shipment and subject to the exceptions listed above.

All other items and pieces of equipment not specified above, including the antenna unit, antenna surge suppressor and antenna pre-amplifier are warranted for 5 years, subject to the exceptions listed above.

WARRANTY CLAIMS

Spectracom's obligation under this warranty is limited to in-factory service and repair, at Spectracom's option, of the product or the component thereof, which is found to be defective. If in Spectracom's judgment the defective condition in a Spectracom product is for a cause listed above for which Spectracom is not responsible, Spectracom will make the repairs or replacement of components and charge its then current price, which buyer agrees to pay.

Spectracom shall not have any warranty obligations if the procedure for warranty claims is not followed. Users must notify Spectracom of the claim with full information as to the claimed defect. Spectracom products shall not be returned unless a return authorization number is issued by Spectracom.

Spectracom products must be returned with the description of the claimed defect and identification of the individual to be contacted if additional information is needed. Spectracom products must be returned properly packed with transportation charges prepaid.

Shipping expense: Expenses incurred for shipping Spectracom products to and from Spectracom (including international customs fees) shall be paid for by the customer, with the following exception. For customers located within the United States, any product repaired by Spectracom under a "warranty repair" will be shipped back to the customer at Spectracom's expense unless special/faster delivery is requested by customer.

Spectracom highly recommends that prior to returning equipment for service work, our technical support department be contacted to provide trouble shooting assistance while the equipment is still installed. If equipment is returned without first contacting the support department and "no problems are found" during the repair work, an evaluation fee may be charged.

EXCEPT FOR THE LIMITED WARRANTY STATED ABOVE, SPECTRACOM DISCLAIMS ALL WARRANTIES OF ANY KIND WITH REGARD TO SPECTRACOM PRODUCTS OR OTHER MATERIALS PROVIDED BY SPECTRACOM, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTY OR MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Spectracom shall have no liability or responsibility to the original customer or any other party with respect to any liability, loss, or damage caused directly or indirectly by any Spectracom product, material, or software sold or provided by Spectracom, replacement parts or units, or services provided, including but not limited to any interruption of service, excess charges resulting from malfunctions of hardware or software, loss of business or anticipatory profits resulting from the use or operation of the Spectracom product or software, whatsoever or howsoever caused. In no event shall Spectracom be liable for any direct, indirect, special or consequential damages whether the claims are grounded in contract, tort (including negligence), or strict liability.

EXTENDED WARRANTY COVERAGE

Extended warranties can be purchased for additional periods beyond the standard five-year warranty. Contact Spectracom no later than the last year of the standard five-year warranty for extended coverage.

SPECTRACOM 95 Methodist Hill Drive Rochester, NY 14623

+1.585.321.5800 FAX: +1.585.321.5218 www.spectracomcorp.com sales@spectracomcorp.com

Table of Contents

1 Overview

The VxWorks Driver for the Spectracom TPRO/TSAT PCI boards provides the interface for multiple users to access the board using the API documented in Chapter Three.

The TPRO-PMC and TSAT-PMC provide high-accuracy timing functions on a plug-in board for the PMC computer bus. The board has an on-board clock, which is kept in sync to either an external timecode input (TPRO-PMC) or to time provided by the GPS satellites (TSAT-PMC). Several timing functions are derived from the on-board clock, including a programmable periodic pulse rate output ("Heartbeat"), a programmable start/stop output ("Match"), a selectable frequency output ("Oscillator Out", 1 kHz, 1, 5, or 10 MHz), and a time-stamping input ("Time-Tag").

The TSAT-PMC includes an externally mounted GPS antenna and a 100-foot cable to connect the antenna to the board. The GPS satellites provide continuous time and position information, available anywhere in the world. It automatically syncs its on-board clock to the time transmitted by the GPS satellites. The board outputs a timecode signal, in IRIG-B format, which conveys the day, hour, minute, and second, and also has a 1 kHz carrier referenced to the on-board oscillator.

The TPRO-PMC is similar to the TSAT-PMC, except it obtains time from an input timecode. The timecode can be in IRIG-A, IRIG-B or NASA36 format; the board automatically detects which format is being used. Timing accuracy is the same regardless of which format is being used. The timecode conveys the day, hour, minute, and second. The on-board 10 MHz oscillator is disciplined to maintain a 1 µs accuracy. An IRIG-B timecode output is provided, which is in-sync with the incoming timecode.

Either board may be used as a stand-alone timecode generator. The computer programs the day, hour, minute, and second. The board then continues to count from that time, using the on-board oscillator as the timebase reference. This is called "freewheeling."
The host computer communicates to either board through a set of memory-mapped registers. When the computer boots up, the board identifies itself to the PMC bus by specifying the unique Subsystem

Vendor ID and Subsystem Device ID. The host computer can then read the instantaneous time, and command the board to set time, and/or to provide an interrupt at a periodic rate, at a specified time, and/or when a time-tag event occurs.

Front panel indicator lights indicate when the board is in the process of synchronizing ("acquiring") the GPS or timecode input signal, and when the board has established valid synchronization. The host computer can also interrogate the status register to determine these and other conditions.

2 Software Development Kit

The TSAT/TPRO-PMC VxWorks C Language SDK consists of an interface header file, an interface library, a test application, and hardware access example source code. Hardware access example software is dependant on the board support package (BSP) supplied with the host processor board. These files must be modified with the PCI bus interface defined in the host BSP.

tpro.h
TSAT/TPRO-PMC interface header file provides interface routine prototypes, interface structures, and interface constant definitions.

tpro.o
TSAT/TPRO-PMC interface library file provides interface routine definitions.
testapp.c
Test application used to test functionality of hardware.

hwaccess.h
Describes the hardware access interface.

hwaccess.c
PCI bus access library which must be tailored for host board support package.

3 Interface to the VxWorks Driver API

3.1 Header File

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

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

              DSPCon, Inc.
              380 FootHill Road
              Bridgewater, NJ 08807

              e-mail: info@dspcon.com

  (C) Copyright 2004 DSPCon, Inc.  All rights reserved.
  Use of copyright notice is precautionary and does not imply publication
*****************************************************************************/

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

#ifndef _defined_TPRO_
#define _defined_TPRO_

#pragma pack(8)


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


/**
***  Heartbeat constants
**/

#define HEART_NORMAL   (0)            /** signal type for CPCI card **/
#define HEART_INVERT          (1)            /** signal type for CPCI card **/

#define HEART_DISABLE  (0)            /** output type for CPCI card **/
#define HEART_ENABLE          (1)            /** output type for CPCI card **/

#define HEART_CLK_10MHZ       (0)            /** clock frequency for CPCI board **/
#define HEART_CLK_3MHZ (1)            /** clock frequency for CPCI board **/
#define HEART_CLK_1MHZ (2)            /** clock frequency for CPCI board **/
#define HEART_CLK_1KHZ (3)            /** clock frequency for CPCI board **/


/**
***  Match constants
**/

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


/**
***  Oscillator frequencies - for Compact PCI Card Only
**/

#define OSC_OUT_OFF                   (0)
#define OSC_OUT_1KHZ                  (1)
#define OSC_OUT_1MHZ                  (2)
```

```
#define OSC_OUT_5MHZ                (3)
#define OSC_OUT_10MHZ         (4)


/*************************************************************************
                            OBJECTS
*************************************************************************/

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

typedef struct TPRO_BoardObj
{ /*-----------------------------------------------------------------*/
  unsigned baseAddr;                        /*-- base address ---------------*/
  unsigned short options;                   /*-- device options -------------*/

  unsigned intLevel;                        /*-- VxWorks interrupt level -----*/
  unsigned intVector;               /*-- VxWorks interrupt vector ----*/
} /*-----------------------------------------------------------------*/
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 clockSel;                                /*-- clock select --------*/
  unsigned divider;                                 /*-- heartbeat divider ---*/
  unsigned invert;                                  /*-- 0-normal, 1-invert --*/
  unsigned enable;                                  /*-- 0-disable, 1-enable -*/
} /*-------------------------------------------------------------------------*/
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 ----------------*/

  unsigned short year;                          /*-- year for CPCI board -*/
} /*-------------------------------------------------------------------------*/
TPRO_TimeObj;


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

typedef struct TPRO_WaitObj
{ /*-------------------------------------------------------------------------*/
  int ticks;                                        /*-- # ticks to wait -----*/

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

  unsigned short year;                      /*-- year for cPCI card --*/
  unsigned char  month;                         /*-- month for cPCI card -*/
  unsigned char  day;                       /*-- day for cPCI card ---*/
} /*-------------------------------------------------------------------------*/
TPRO_WaitObj;

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

typedef struct TPRO_MemObj
```

```
{ /*---------------------------------------------------------------------*/
  unsigned short offset;
  unsigned short value;

  unsigned long l_value;
} /*---------------------------------------------------------------------*/
TPRO_MemObj;


/*************************************************************************
                          ERROR CODES
*************************************************************************/

#define TPRO_SUCCESS                        (0)     /** success **/
#define TPRO_HANDLE_ERR                     (1)     /** error creating handle to device
**/
#define TPRO_OBJECT_ERR                     (2)     /** error creating device object **/
#define TPRO_CLOSE_HANDLE_ERR       (3)     /** error closing device handle **/
#define TPRO_DEVICE_NOT_OPEN_ERR            (4)     /** tpro device was not opened **/
#define TPRO_INVALID_BOARD_TYPE_ERR (5)     /** function is not available for board type
**/
#define TPRO_FREQ_ERR               (6)     /** invalid frequency **/
#define TPRO_YEAR_PARM_ERR                  (7)     /** invalid year parameter **/
#define TPRO_DAY_PARM_ERR                   (8)     /** invalid day parameter **/
#define TPRO_HOUR_PARM_ERR          (9)             /** invalid hour parameter **/
#define TPRO_MIN_PARM_ERR           (10)            /** invalid minutes parameter **/
#define TPRO_SEC_PARM_ERR           (11)            /** invalid seconds parameter **/
#define TPRO_DELAY_PARM_ERR         (12)            /** invalid delay factor **/
#define TPRO_TIMEOUT_ERR            (13)            /** device timed out **/
#define TPRO_COMM_ERR        (14)           /** error communicating with driver **/


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

unsigned char TPRO_open                     (TPRO_BoardObj *hnd);
unsigned char TPRO_close                    (TPRO_BoardObj *hnd);
unsigned char TPRO_flushFIFO        (TPRO_BoardObj *hnd);
unsigned char TPRO_getAltitude              (TPRO_BoardObj *hnd, TPRO_AltObj *Altp);
unsigned char TPRO_getDate                  (TPRO_BoardObj *hnd, TPRO_DateObj *Datep);
unsigned char TPRO_getDriver        (TPRO_BoardObj *hnd, char *driver);
unsigned char TPRO_getFirmware              (TPRO_BoardObj *hnd, char *firmware);
unsigned char TPRO_getFPGA                  (TPRO_BoardObj *hnd, char *fpga);
unsigned char TPRO_getLatitude              (TPRO_BoardObj *hnd, TPRO_LatObj *Latp);
unsigned char TPRO_getLongitude             (TPRO_BoardObj *hnd, TPRO_LongObj *Longp);
unsigned char TPRO_getSatInfo       (TPRO_BoardObj *hnd, TPRO_SatObj *Satp);
unsigned char TPRO_getTime                  (TPRO_BoardObj *hnd, TPRO_TimeObj *Timep);
unsigned char TPRO_resetFirmware            (TPRO_BoardObj *hnd);
unsigned char TPRO_setHeartbeat             (TPRO_BoardObj *hnd, TPRO_HeartObj *Heartp);
unsigned char TPRO_setMatchTime             (TPRO_BoardObj *hnd, TPRO_MatchObj *Matchp);
unsigned char TPRO_setOscillator            (TPRO_BoardObj *hnd, unsigned char *freq);
unsigned char TPRO_setPropDelayCorr (TPRO_BoardObj *hnd, int *us);
unsigned char TPRO_setTime                  (TPRO_BoardObj *hnd, TPRO_TimeObj *Timep);
unsigned char TPRO_setYear                  (TPRO_BoardObj *hnd, unsigned short yr);
unsigned char TPRO_simEvent                 (TPRO_BoardObj *hnd);
unsigned char TPRO_synchControl             (TPRO_BoardObj *hnd, unsigned char enb);
unsigned char TPRO_synchStatus              (TPRO_BoardObj *hnd, unsigned char *status);
unsigned char TPRO_waitEvent        (TPRO_BoardObj *hnd, TPRO_WaitObj *waitp);
unsigned char TPRO_waitHeartbeat            (TPRO_BoardObj *hnd, int ticks);
unsigned char TPRO_waitMatch        (TPRO_BoardObj *hnd, int ticks);

#pragma pack()

#endif  /** _defined_TPRO_ **/
```

3.2 TPRO API — Routine Descriptions

3.2.1 TPRO_open

Accessing the TSAT/TPRO-PMC device in VxWorks is dependant on the host computer's BSP (board support package). The TPRO_open routine utilizes the functions defined in the hwaccess library to read the PCI configuration space. The device's base address location is stored in an element in the TPRO_BoardObj structure and is used to access the device throughout the library. Interrupt level and interrupt vector must be defined in the hwaccess library by the user.

Syntax
unsigned char TPRO_open (TPRO_BoardObj *hnd)

Parameters
TPRO_BoardObj *hnd – pointer to TPRO_BoardObj variable

Return Value
driver error code

Example

```
unsigned char rv;

/**
 **  we must set our interrupt level and vector for VxWorks
 **/

Device.intLevel = 0x19;
Device.intVector = 0x19;


/**
 **  call TPRO_open routine to read the pci configuration space
 **  and set the options parameter in the device object
 **/

if (rv = TPRO_open (&Device), rv != TPRO_SUCCESS){
  printf ("TPRO_open failed.  Error: %d\n", rv);
}
```

3.2.2 TPRO_close

Closes an open TSAT/TPRO-PMC device handle.

Syntax
unsigned char TPRO_close (TPRO_BoardObj *hnd)

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle

Return Value
driver error code

Example
```
unsigned char rv;

/**
 **  call TPRO_close to clear the device object base address parameter
 **/

if (rv = TPRO_close (&Device), rv != TPRO_SUCCESS) {
  printf ("TPRO_close failed.  Error: %d\n", rv);
}
```

3.2.3 TPRO_getAltitude

Retrieves altitude information, in meters, from TSAT-PMC devices. This routine is not valid for TPRO-PMC devices, as TPRO boards can only utilize IRIG inputs.

Syntax
unsigned char TPRO_getAltitude (TPRO_BoardObj *hnd, TPRO_AltObj *Altp)

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle
TPRO_AltObj *Altp - pointer to altitude object

Return Value
driver error code

Example
```
TPRO_AltObj Altitude;
unsigned char rv;

/**
 **  call TPRO_getAltitude routine to retrieve altitude information
 **/

if (rv = TPRO_getAltitude (&Device, &Altitude), rv != TPRO_SUCCESS) {
  printf ("TPRO_getAltitude failed.  Error: %d\n", rv);
  return;
}
printf ("Altitude: %f\n", Altitude.meters);
```

3.2.4 TPRO_getDate

Retrieves the Gregorian date from the TSAT/TPRO-PMC device.

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

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle
TPRO_DateObj *Datep - pointer to date object

Return Value
driver error code

Example

```
TPRO_DateObj Date;
unsigned char rv;

/**
 **  call TPRO_getDate routine to retrieve date information
 **/

if (rv = TPRO_getDate (&Device, &Date), rv != TPRO_SUCCESS) {
  printf ("TPRO_getDate failed.  Error: %d\n", rv);
  return;
}

printf   ("Date:   %02d-%02d-%04d\n\n",   Date.month,   Date.day,
Date.year);
```

3.2.5 TPRO_getDriver

Retrieves the VxWorks driver version string for the TSAT/TPRO-PMC device.

Syntax
unsigned char TPRO_getDriver (TPRO_BoardObj *hnd, char *driver)

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle
char *driver - driver string to hold version

Return Value
driver error code

Example

```
char driver [10];
unsigned char rv;

/**
 **  call TPRO_getDriver routine to retrieve driver version
 **/

if (rv = TPRO_getDriver (&Device, driver), rv != TPRO_SUCCESS) {
  printf ("TPRO_getDriver failed.  Error: %d\n", rv);
  return;
}

printf ("Driver version: %s\n", driver);
```

3.2.6 TPRO_getFirmware

Retrieves the on-board ROM firmware version for the TSAT/TPRO-PMC device.

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

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle
char *firmware - firmware array to hold version

Return Value
driver error code

Example

```
char firmware [3];
unsigned char rv;

/**
 **  call TPRO_getFirmware routine to retrieve firmware version
 **/

if   (rv   =   TPRO_getFirmware   (&Device,   firmware),   rv   !=
TPRO_SUCCESS) {
  printf ("TPRO_getFirmware failed.  Error: %d\n", rv);
  return;
}

printf ("Firmware version: %02X-%02X-%02X\n", firmware [0],
       firmware [1], firmware [2]);
```

3.2.7 TPRO_getFPGA

Retrieves the on-board FPGA ROM version for the TSAT/TPRO-PMC device.

Syntax
unsigned char TPRO_getFPGA (TPRO_BoardObj *hnd, char *fpga)

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle
char *fpga - fpga array to hold version

Return Value
driver error code

Example

```
unsigned char fpga [3];
unsigned char rv;

/**
 **  call TPRO_getFPGA routine to retrieve the FPGA version
 **/

if (rv = TPRO_getFPGA (&Device, fpga), rv != TPRO_SUCCESS) {
  printf ("TPRO_getFPGA failed.  Error: %d\n", rv);
  return;
}

printf ("FPGA version: %02X-%02X-%02X\n", fpga [0],
       fpga [1], fpga [2]);
```

3.2.8 TPRO_getLatitude

Retrieves latitude information, in degrees and minutes, from TSAT-PMC devices. This routine is not valid for TPRO-PMC devices, as TPRO boards can only utilize IRIG inputs.

Syntax
unsigned char TPRO_getLatitude (TPRO_BoardObj *hnd, TPRO_LatObj *Latp)

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle
TPRO_LatObj *Latp - pointer to latitude object

Return Value
driver error code

Example

```
TPRO_LatObj Latitude;
unsigned char rv;

/**
 **  call TPRO_getLatitude routine to retrieve the latitude
 **/

if (rv = TPRO_getLatitude (&Device, &Latitude), rv != TPRO_SUCCESS) {
  printf ("TPRO_getLatitude failed.  Error: %d\n", rv);
  return;
}

printf ("Latitude: %d degrees, %f minutes\n", Latitude.degrees, Latitude.minutes);
```

3.2.9 TPRO_getLongitude

Retrieves longitude information, in degrees and minutes, from TSAT-PMC devices. This routine is not valid for TPRO-PMC devices, as TPRO boards can only utilize IRIG inputs.

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

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle
TPRO_LongObj *Longp - pointer to longitude object

Return Value
driver error code

Example

```
TPRO_LongObj Longitude;
unsigned char rv;

/**
 **  call TPRO_getLongitude routine to retrieve the longitude
 **/

if (rv = TPRO_getLongitude (&Device, &Longitude), rv != TPRO_SUCCESS) {
  printf ("TPRO_getLongitude failed.  Error: %d\n", rv);
  return;
}

printf ("Longitude: %d degrees, %f minutes\n", Longitude.degrees, Longitude.minutes);
```

3.2.10 TPRO_getSatInfo

Retrieves satellite information, number of satellites tracked, from TSAT-PMC devices. This routine is not valid for TPRO-PMC devices, as TPRO boards can only utilize IRIG inputs.

Syntax
unsigned char TPRO_getSatInfo (TPRO_BoardObj *hnd, TPRO_SatObj *Satp)

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle
TPRO_SatObj *Satp - pointer to satellite infomation object

Return Value
driver error code

Example

```
TPRO_SatObj SatInfo;
unsigned char rv;

/**
 **     call   TPRO_getSatInfo   routine   to   retrieve   satellite
informatio
 **/

if (rv = TPRO_getSatInfo (&Device, &SatInfo), rv != TPRO_SUCCESS)
{
  printf ("TPRO_getSatInfo failed.  Error: %d\n", rv);
  return;
}

printf     ("Number     of     satellites     tracked:     %d\n",
SatInfo.satsTracked);
```

3.2.11 TPRO_getTime

Retrieves the time from the TSAT/TPRO-PMC device and stores it into the TPRO_TimeObj structure. This time has microsecond resolution and therefore uses the secsDouble element of the structure to represent seconds. The seconds element of the structure is used for setting the time on the device.

Syntax
unsigned char TPRO_getTime (TPRO_BoardObj *hnd, TPRO_TimeObj *Timep)

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle
TPRO_TimeObj *Timep - pointer to tpro time object

Return Value
driver error code

Example

```
TPRO_TimeObj TproTime;
unsigned char rv;

/**
 **  call TPRO_getTime routine to retrieve device time
 **/

if (rv = TPRO_getTime (&Device, &TproTime), rv != TPRO_SUCCESS) {
  printf ("TPRO_getTime failed.  Error: %d\n", rv);
  return;
}

/**
 **  print device time
 **/

printf ("Device Time:\n");
printf ("  %03d:%02d:%02d:%f\n\n", TproTime.days, TproTime.hours,
     TproTime.minutes, TproTime.secsDouble);
```

3.2.12 TPRO_resetFirmware

Resets the on-board firmware on the TSAT/TPRO-PMC device. The routine waits for the hardware to be completely in the reset state.

Syntax
unsigned char TPRO_resetFirmware (TPRO_BoardObj *hnd)

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle

Return Value
driver error code

Example

```
unsigned char rv;

/**
 **  call TPRO_resetFirmware routine to reset on-card firmware
 **/

if (rv = TPRO_resetFirmware (&Device), rv != TPRO_SUCCESS) {
  printf ("TPRO_resetFirmware failed.  Error: %d\n", rv);
  return;
}

printf ("Firmware has been successfully reset.\n");
```

3.2.13 TPRO_setHeartbeat

Configures the characteristics of the heartbeat output on the TSAT/TPRO-PMC device. The TPRO_setHeartbeat routine utilizes the TPRO_HeartObj structure to program the heartbeat registers on the device. The structure consists of a clockSel element, which determines the clock source for the heartbeat output. The divider element programs the divide number for the heartbeat output (see equation in hardware manual). The invert element determines the polarity of the heartbeat pulse and the enable element enables/disables the output.

Syntax
unsigned char TPRO_setHeartbeat (TPRO_BoardObj *hnd, TPRO_HeartObj *Heartp)

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle
TPRO_HeartObj *Heartp - pointer to tpro heartbeat object

Return Value
driver error code

Example

```
TPRO_HeartObj Heartbeat;
unsigned char rv;

/**
 **  store parameters in heartbeat object
 **/

Heartbeat.clockSel = HEART_CLK_1MHZ;
Heartbeat.divider = div;
Heartbeat.invert = HEART_NORMAL;
Heartbeat.enable = HEART_ENABLE;


/**
 **  call TPRO_setHeartbeat routine to set hearbeat rate
 **/

if (rv = TPRO_setHeartbeat (&Device, &Heartbeat), rv !=
TPRO_SUCCESS) {
  printf ("TPRO_setHeartbeat failed.  Error: %d\n", rv);
  return;
}

printf ("Heartbeat rate has been successfully set.\n");
```

3.2.14 TPRO_setMatchTime

Configures the match start/stop time on the TSAT/TPRO-PMC device. The TPRO_SetMatchTime routine utilizes the TPRO_MatchObj structure to program the match registers on the device. The structure contains a matchType element used to determine a start or stop match condition (MATCH_TIME_START, MATCH_TIME_STOP).

Syntax
unsigned char TPRO_setMatchTime (TPRO_BoardObj *hnd, TPRO_MatchObj *Matchp)

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle
TPRO_MatchObj *Matchp - pointer to tpro match time object

Return Value
driver error code

Example

```
TPRO_TimeObj TproTime;
TPRO_MatchObj Match;

unsigned char rv;

/**
 **  call TPRO_getTime routine to retrieve device time
 **/

if (rv = TPRO_getTime (&Device, &TproTime), rv != TPRO_SUCCESS) {
  printf ("TPRO_getTime failed.  Error: %d\n", rv);
  return;
}

printf ("Current Time:\n");
printf ("  %03d:%02d:%02d:%f\n\n", TproTime.days, TproTime.hours,
     TproTime.minutes, TproTime.secsDouble);

/**
 **   set match time for 10 seconds later - no support for
rollover
 **/

Match.matchType = MATCH_TIME_START;
Match.seconds = TproTime.secsDouble + 10.0;
Match.minutes = TproTime.minutes;
Match.hours = TproTime.hours;
Match.days = TproTime.days;


/**
 **  call TPRO_setMatchTime routine to set match time
```

```
 **/

if (rv = TPRO_setMatchTime (&Device, &Match), rv != TPRO_SUCCESS)
{
  printf ("TPRO_setMatchTime failed.  Error: %d\n", rv);
  return;
}

printf ("Match time:\n");
printf ("  %03d:%02d:%02d:%f\n\n", Match.days, Match.hours,
     Match.minutes, Match.seconds);
```

3.2.15 TPRO_setOscillator

Configures the oscillator output rate on the TSAT/TPRO-PMC device.

Syntax
unsigned char TPRO_setOscillator (TPRO_BoardObj *hnd, unsigned char *freq)

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle
unsigned char *freq - valid oscillator rate (OSC_OUT_OFF, OSC_OUT_1KHZ, OSC_OUT_1MHZ, OSC_OUT_5MHZ, OSC_OUT_10MHZ)

Return Value
driver error code

Example

```
unsigned char rate = OSC_OUT_1MHZ;
unsigned char rv;

/**
 **  call TPRO_setOscillator routine to set oscillator frequency
 **/

if (rv = TPRO_setOscillator (&Device, (unsigned char *) &rate),
rv != TPRO_SUCCESS) {
  printf ("TPRO_setOscillator failed.  Error: %d\n", rv);
  return;
}

printf ("Oscillator rate successfully set.\n");
```

3.2.16 TPRO_setPropDelayCorr

Configures the propagation delay on the TSAT/TPRO-PMC device.

Syntax
unsigned char TPRO_setPropDelayCorr (TPRO_BoardObj *hnd, int *us)

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle
int *us - propagation delay correction in microseconds

Return Value
driver error code

Example

```
unsigned char rv;
int delay = 5;

/**
 **  call TPRO_setPropDelayCorr routine to set propagation delay
factor
 **/

if  (rv  =  TPRO_setPropDelayCorr  (&Device,  &delay),  rv  !=
TPRO_SUCCESS) {
  printf ("TPRO_setPropDelayCorr failed.  Error: %d\n", rv);
  return;
}

printf ("Propagation delay successfully set.\n");
```

3.2.17 TPRO_setTime

Sets the time on the TSAT/TPRO-PMC device using a TPRO_TimeObj structure. The time can only be set with a whole number of seconds and therefore uses the seconds element of the structure to represent seconds. The secsDouble element of the structure is used for retrieving the time from the device.

Syntax
unsigned char TPRO_setTime (TPRO_BoardObj *hnd, TPRO_TimeObj *Timep)

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle
TPRO_TimeObj *Timep - pointer to tpro time object

Return Value
driver error code

Example

```
TPRO_TimeObj TproTime;
unsigned char rv;

TproTime.seconds = 0;
TproTime.minutes = 0;
TproTime.hours   = 0;
TproTime.days    = 0;
TproTime.year    = 2000;

/**
 **  call TPRO_setTime routine to set the device time
 **/

if (rv = TPRO_setTime (&Device, &TproTime), rv != TPRO_SUCCESS) {
  printf ("TPRO_setTime failed.  Error: %d\n", rv);
  return;
}
printf ("Time has been successfully set.\n");
```

3.2.18 TPRO_setYear

sets the year on the TSAT/TPRO-PMC device.

Syntax
unsigned char TPRO_setYear (TPRO_BoardObj *hnd, unsigned short *yr)

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle
unsigned short *yr - pointer to year variable

Return Value
driver error code

Example

```
unsigned short year = 2004;
unsigned char rv;

/**
 **  call TPRO_setYear routine to set the device year
 **/

if (rv = TPRO_setYear (&Device, (unsigned short *) &year), rv !=
TPRO_SUCCESS) {
  printf ("TPRO_setYear failed.  Error: %d\n", rv);
  return;
}

printf ("Year has been successfully set.\n");
```

3.2.19 TPRO_simEvent

Simultes an external time-tagged event on the TSAT/TPRO-PMC device.

Syntax
unsigned char TPRO_simEvent (TPRO_BoardObj *hnd)

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle

Return Value
driver error code

Example

```
unsigned char rv;

/**
 **  call TPRO_simEvent routine to simulate external event
 **/

if (rv = TPRO_simEvent (&Device), rv != TPRO_SUCCESS) {
  printf ("TPRO_simEvent failed.  Error: %d\n", rv);
  return;
}

printf ("Successfully simulated external event.\n");
```

3.2.20 TPRO_synchControl

Enables or disables synchronization to IRIG or GPS inputs on the TSAT/TPRO-PMC device. The routine accepts an enable variable for synchronization control (0:freewheel, 1:synchronize)

Syntax
unsigned char TPRO_synchControl (TPRO_BoardObj *hnd, unsigned char enb)

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle
unsigned char enb - synchronization enable

Return Value
driver error code

Example

```
unsigned char ctrl = 1;
unsigned char rv;

/**
 **      call   TPRO_synchControl   routine   to   enable/disable
synchronization
 **/

if (rv = TPRO_synchControl (&Device, ctrl), rv != TPRO_SUCCESS) {
  printf ("TPRO_synchControl failed.  Error: %d\n", rv);
  return;
}

printf ("Successfully controlled synchronization.\n");
```

3.2.21 TPRO_synchStatus

Retrieves the synchronization status of the TSAT/TPRO-PMC device. The routine reports not synchronized, synchronized to IRIG-A, synchronized to IRIG-B, synchronized to NASA36, or synchronized to GPS antenna.

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

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle
unsigned char *status - synchronization status (0-not synchronzied, 1-IRIGA, 2-IRIGB, 3-NASA36, 4-GPS)

Return Value
driver error code

Example

```
unsigned char status;
unsigned char rv;

/**
 **   call TPRO_synchStatus routine to retrieve synchronization
status
 **/

if (rv = TPRO_synchStatus (&Device, &status), rv != TPRO_SUCCESS)
{
  printf ("TPRO_synchStatus failed.  Error: %d\n", rv);
  return (rv);
}

printf ("Synchronization status: %d\n", status);
return (TPRO_SUCCESS);
```

3.2.22 TPRO_waitEvent

Waits on an external time-tagged event captured by the TSAT/TPRO-PMC device. The ticks element in the TPRO_WaitObj structure is a timeout value used if no event occurs. The routine returns the time of the event in the TPRO_WaitObj structure.

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

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle
TPRO_WaitObj *waitp - pointer to TPRO_WaitObj structure

Return Value
driver error code

Example

```
TPRO_WaitObj Wait;
unsigned char rv;

/** set timeout **/
Wait.ticks = ticks;

/**
 ** call TPRO_waitEvent routine to retrieve time-tagged event
 **/

if (rv = TPRO_waitEvent (&Device, &Wait), rv != TPRO_SUCCESS) {

  /** timeout **/
  if (rv == TPRO_TIMEOUT_ERR) {
    printf ("TIMEOUT WAITING FOR EVENT!\n");
    return (rv);
  }
  printf ("TPRO_waitEvent failed.  Error: %d\n", rv);
  return (rv);
}


/**
 **  print time-tagged event information
 **/

printf ("Event:\n");
printf ("  %02d-%02d-%04d\n", Wait.month, Wait.day, Wait.year);
printf ("  %03d:%02d:%02d:%f\n\n", Wait.days, Wait.hours,
       Wait.minutes, Wait.seconds);
return (TPRO_SUCCESS);
```

3.2.23 TPRO_waitHeartbeat

Waits on a heartbeat interrupt caused by the TSAT/TPRO-PMC device. A timeout value is passed in as an argument if no heartbeat interrupt occurs.

Syntax
unsigned char TPRO_waitHeartbeat (TPRO_BoardObj *hnd, unsigned int ticks)

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle
unsigned int *ticks - timeout value in ticks

Return Value
driver error code

Example

```
unsigned char rv;

/**
 **  call TPRO_waitHeartbeat routine to get heartbeat interrupt
 **/

if (rv = TPRO_waitHeartbeat (&Device, ticks), rv != TPRO_SUCCESS)
{

  /** timeout **/
  if (rv == TPRO_TIMEOUT_ERR) {
    printf ("TIMEOUT WAITING FOR HEARTBEAT!\n");
    return (rv);
  }
  printf ("TPRO_waitHeartbeat failed.  Error: %d\n", rv);
  return (rv);
}


/**
 **  received heartbeat interrupt
 **/

printf ("Received Heartbeat Interrupt!\n");
return (TPRO_SUCCESS);
```

3.2.24 TPRO_waitMatch

Waits on a match time interrupt caused by the TSAT/TPRO-PMC device. A timeout value is passed in as an argument if no match time interrupt occurs.

Syntax
unsigned char TPRO_waitMatch (TPRO_BoardObj *hnd, unsigned int ticks)

Parameters
TPRO_BoardObj *hnd - valid TPRO device handle
unsigned int ticks - timeout value in ticks

Return Value
driver error code

Example

```
unsigned char rv;

/**
 **  call TPRO_waitMatch routine to get match time interrupt
 **/

if (rv = TPRO_waitMatch (&Device, ticks), rv != TPRO_SUCCESS) {

  /** timeout **/
  if (rv == TPRO_TIMEOUT_ERR) {
    printf ("TIMEOUT WAITING FOR MATCH TIME!\n");
    return (rv);
  }
  printf ("TPRO_waitMatch failed.  Error: %d\n", rv);
  return (rv);
}


/**
 **  received match time interrupt
 **/

printf ("Received Match Time Interrupt!\n");
return (TPRO_SUCCESS);
```

3.3 Error Codes

```
/*************************************************************************
                  ERROR CODES
*************************************************************************/

#define TPRO_SUCCESS                    (0)    /** success **/
#define TPRO_HANDLE_ERR                 (1)    /** error creating handle to device **/
#define TPRO_OBJECT_ERR                 (2)    /** error creating device object **/
#define TPRO_CLOSE_HANDLE_ERR           (3)    /** error closing device handle **/
#define TPRO_DEVICE_NOT_OPEN_ERR        (4)    /** tpro device was not opened **/
#define TPRO_INVALID_BOARD_TYPE_ERR     (5)    /** function is not available for board type **/
#define TPRO_FREQ_ERR                   (6)    /** invalid frequency **/
#define TPRO_YEAR_PARM_ERR              (7)    /** invalid year parameter **/
#define TPRO_DAY_PARM_ERR               (8)    /** invalid day parameter **/
#define TPRO_HOUR_PARM_ERR              (9)    /** invalid hour parameter **/
#define TPRO_MIN_PARM_ERR               (10)   /** invalid minutes parameter **/
#define TPRO_SEC_PARM_ERR               (11)   /** invalid seconds parameter **/
#define TPRO_DELAY_PARM_ERR             (12)   /** invalid delay factor **/
#define TPRO_TIMEOUT_ERR                (13)   /** device timed out **/
#define TPRO_COMM_ERR                   (14)   /** error communicating with driver **/
```

3.4 Library Structures

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

typedef struct TPRO_BoardObj
{ /*-----------------------------------------------------------------------*/
  unsigned baseAddr;                    /*-- base address ----------------*/
  unsigned short options;               /*-- device options --------------*/

  unsigned intLevel;                    /*-- VxWorks interrupt level -----*/
  unsigned intVector;                   /*-- VxWorks interrupt vector ----*/
} /*-----------------------------------------------------------------------*/
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 clockSel;                         /*-- clock select --------*/
  unsigned divider;                          /*-- heartbeat divider ---*/
  unsigned invert;                           /*-- 0-normal, 1-invert --*/
  unsigned enable;                           /*-- 0-disable, 1-enable -*/
} /*-----------------------------------------------------------------------*/
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 ----------------*/

  unsigned short year;                        /*-- year for CPCI board -*/
} /*-------------------------------------------------------------------*/
TPRO_TimeObj;


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

typedef struct TPRO_WaitObj
{ /*-------------------------------------------------------------------*/
  int ticks;                                  /*-- # ticks to wait -----*/

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

  unsigned short year;                        /*-- year for cPCI card --*/
  unsigned char  month;                       /*-- month for cPCI card -*/
  unsigned char  day;                         /*-- day for cPCI card ---*/
} /*-------------------------------------------------------------------*/
TPRO_WaitObj;

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

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

  unsigned long l_value;
} /*-------------------------------------------------------------------*/
TPRO_MemObj;
```

REVISION HISTORY

Revision Level	*ECN Number*	*Description*
A	**2187**	First draft of Spectracom documentation for this product.

Spectracom Corporation
95 Methodist Hill Drive
Rochester, NY 14623
www.spectracomcorp.com
Phone: US +1.585.321.5800
Fax: US +1.585.321.5219