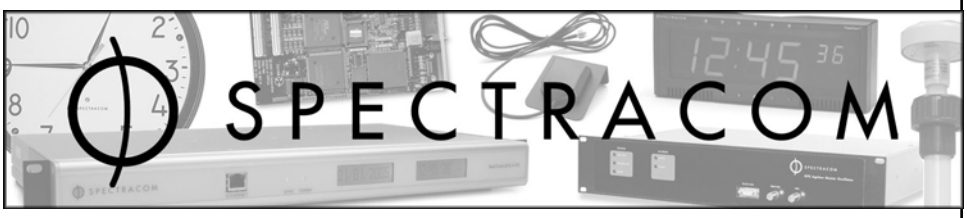
TPRO-VME /TSAT-VME
SYNCHRONIZABLE TIMECODE GENERATOR with VME BUS INTERFACE

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 1156-5002-0050
Manual Revision A
28 April 2008

Copyright © 2008 Spectracom Corporation. The contents of this publication may not be reproduced in any form without the written permission of Spectracom Corporation. Printed in USA.

Specifications subject to change or improvement without notice.

Spectracom, NetClock, Ageless, TimeGuard, TimeBurst, TimeTap, LineTap, MultiTap, VersaTap, and Legally Traceable Time are Spectracom registered trademarks. All other products are identified by trademarks of their respective companies or organizations. All rights reserved.

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-VME performs timing and synchronization functions referenced to an input timecode signal. The board synchronizes its on-board clock to the incoming timecode. The clock is also provided as an IRIG-B output. Other features include a time-tag TTL input, a 1 Mhz TTL output, and two user-configurable TTL pulse rate outputs.

The board continues to increment time (“freewheel”) in the absence of an input timecode. Thus, the host computer can set the on-board clock over the VME bus, and the board can be used as an IRIG-B timecode generator.

The input timecode format (IRIG-B, IRIG-A, NASA36, SR3, or 2137) is automatically detected. Synchronization to the input timecode is also automatic and a propagation delay offset may be specified to compensate for cable delays.

Front panel indicators include presence of input timecode and successful synchronization. An optional seven segment LED display shows day and time in DDD:HH:MM:SS format.

The timecode input is an amplitude modulated sine wave. The peak amplitude can be between 0.5 Vp-p and 8.0 Vp-p. The timecode input is differential; the board does not reference this signal to ground. A single-ended input (referenced to ground) is also acceptable.

2 Application Example

```
/*************************************************************************
  TPRO/TSAT VME Test Program

               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
*************************************************************************/

/*************************************************************************
  appmain.c
*************************************************************************/

#include <vxWorks.h>

#include <semLib.h>
#include <taskLib.h>

#include <stdlib.h>           /* ansi run-time support */
#include <stdarg.h>           /* ansi run-time support */
#include <string.h>           /* ansi run-time support */

#include "tpro.h"


TPRO_BoardObj Board;


/*************************************************************************
  open
*************************************************************************/

void open (void)
{
  /**
  ***  open device
  **/

  Board.intLevel = 5;
  Board.intVector = 0xC0;
  Board.a16baseAddr = 0x0700;
  Board.boardOptions = TPRO_M_OPTION;

  if (TPRO_open (&Board) != TPRO_SUCCESS) {
    printf ("Open Failed!!\n");
    return;
  }
}


/*************************************************************************
  close
*************************************************************************/

void close (void)
{
  /**
  ***  close device
  **/

  if (TPRO_close (&Board) != TPRO_SUCCESS) {
    printf ("Could not close device!!\n");
    return;
  }
  printf ("\n");
}


/*************************************************************************
  get_time
*************************************************************************/
```

```
void get_time (void)
{
  STATUS rv;
  TPRO_TimeObj TproTime;


  /**
  ***  get time
  **/

  printf ("Retrieving Time....\n");
  if (TPRO_getTime (&Board, &TproTime) != TPRO_SUCCESS) {
    printf ("Get Time Failed!!\n");
    return;
  }

  printf ("  Days: %d\n", TproTime.days);
  printf ("  Hours: %d\n", TproTime.hours);
  printf ("  Minutes: %d\n", TproTime.minutes);
  printf ("  Seconds: %f\n", TproTime.seconds);
  printf ("\n");
}


/*************************************************************************
  clear_time
*************************************************************************/

void clear_time (void)
{
  STATUS rv;
  TPRO_TimeObj TproTime;


  /**
  ***  clear time
  **/

  TproTime.days = 0;
  TproTime.hours = 0;
  TproTime.minutes = 0;
  TproTime.seconds = 0.0;

  printf ("Setting Time....\n");
  if (TPRO_setTime (&Board, &TproTime) != TPRO_SUCCESS) {
    printf ("Set Time Failed!!\n");
    return;
  }
  printf ("\n");
}


/*************************************************************************
  disable_synch
*************************************************************************/

void disable_synch (void)
{
  STATUS rv;
  unsigned char cmd = 0;


  /**
  ***  disable synchronization
  **/

  printf ("Disabling synchronization....\n");
  if (TPRO_synchControl (&Board, &cmd) != TPRO_SUCCESS) {
    printf ("Unable to disable synchronization!!\n");
    return;
  }
  printf ("\n");
}


/*************************************************************************
  enable_synch
*************************************************************************/

void enable_synch (void)
```

```
{
  STATUS rv;
  unsigned char cmd = 1;


  /**
  ***  enable synchronization
  **/

  printf ("Enabling synchronization....\n");
  if (TPRO_synchControl (&Board, &cmd) != TPRO_SUCCESS) {
    printf ("Unable to enable synchronization!!\n");
    return;
  }
  printf ("\n");
}


/*****************************************************************************
  synch_status
*****************************************************************************/

void synch_status (void)
{
  STATUS rv;
  unsigned char stat;


  /**
  ***  check synchronization status
  **/

  printf ("Checking synchronization....");
  if (TPRO_synchStatus (&Board, &stat) != TPRO_SUCCESS) {
    printf ("Synchronization status failed!!\n");
    return;
  }
  printf ("status => %d\n", stat);
  printf ("\n");
}


/*****************************************************************************
  wait_event
*****************************************************************************/

void wait_event (void)
{
  STATUS rv;
  TPRO_WaitObj waitObj;


  /**
  ***  wait for event
  **/

  printf ("\nWaiting for event....\n");
  waitObj.ticks = 1000;
  if (TPRO_waitEvent (&Board, &waitObj) == TPRO_TIMEOUT_ERR) {
    printf ("TIMEOUT ERROR!!!\n\n");
    return;
  }

  printf ("  Days: %d\n", waitObj.days);
  printf ("  Hours: %d\n", waitObj.hours);
  printf ("  Minutes: %d\n", waitObj.minutes);
  printf ("  Seconds: %f\n", waitObj.seconds);

  printf ("\n");
}


/*****************************************************************************
  sim_event
*****************************************************************************/

void sim_event (void)
{
  STATUS rv;
```

```
  /**
  ***  simulate event
  **/

  if (TPRO_simEvent (&Board) != TPRO_SUCCESS) {
    printf ("Unable to simulate event!!\n\n");
    return;
  }
  printf ("Simulated Time-tag event!!\n");
  printf ("\n");
}


/*************************************************************************
  get_altitude
*************************************************************************/

void get_altitude (void)
{
  STATUS rv;
  TPRO_AltObj altitude;


  /**
  ***  get altitude
  **/

  if (TPRO_getAltitude (&Board, &altitude) != TPRO_SUCCESS) {
    printf ("Unable to retrieve altitude!!\n\n");
    return;
  }
  printf ("Altitude: %f\n", altitude.meters);
  printf ("\n");
}


/*************************************************************************
  get_latitude
*************************************************************************/

void get_latitude (void)
{
  STATUS rv;
  TPRO_LatObj latitude;


  /**
  ***  get latitude
  **/

  if (TPRO_getLatitude (&Board, &latitude) != TPRO_SUCCESS) {
    printf ("Unable to retrieve latitude!!\n\n");
    return;
  }
  printf ("Latitude:\n");
  printf (" Degrees: %d\n", latitude.degrees);
  printf (" Minutes: %f\n", latitude.minutes);
  printf ("\n");
}


/*************************************************************************
  get_longitude
*************************************************************************/

void get_longitude (void)
{
  STATUS rv;
  TPRO_LatObj longitude;


  /**
  ***  get longitude
  **/

  if (TPRO_getLongitude (&Board, &longitude) != TPRO_SUCCESS) {
    printf ("Unable to retrieve longitude!!\n\n");
    return;
  }
  printf ("Longitude:\n");
  printf (" Degrees: %d\n", longitude.degrees);
```

```
  printf (" Minutes: %f\n", longitude.minutes);
  printf ("\n");
}
```

3 Interface to the VxWorks Driver API

NOTE: The distribution media contains three directories containing the appropriate files:

- "Documentation" – directory containing the TPRO-VME_Application Prog Guide_Ed 1.pdf
- "H" – directory containing the tpro.h header file
- "Lib" – directory containing the tprovme.o import library

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 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_


/*===========================================================================
                      HARDWARE OPTION DEFINES
===========================================================================*/

#define TPRO_NO_OPTIONS       (0)           /*-- no option -----------*/
#define TPRO_M_OPTION         (1)           /*-- -M option -----------*/
#define TPRO_INT32_OPTION     (2)           /*-- INT32 option --------*/


/*===========================================================================
                    TPRO SYNCHRONIZATION DEFINES
===========================================================================*/

#define TPRO_FREEWHEEL        (0)
#define TPRO_SYNCHRONIZE      (1)


/*===========================================================================
                          TPRO ERROR CODES
===========================================================================*/

#define TPRO_SUCCESS                 (0)    /** success **/
#define TPRO_HANDLE_ERR              (1)    /** error creating handle to device **/
#define TPRO_DEVICE_NOT_OPEN_ERR     (4)    /** tpro device was not opened **/
#define TPRO_FREQ_ERR                (6)    /** invalid frequency **/
#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 **/


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

typedef struct TPRO_BoardObj
{ /*-----------------------------------------------------------------------*/
  void *hnd;                                /*-- handle to our device */

  unsigned short a16baseAddr;               /*-- A16 base address ----*/
  unsigned short boardOptions;              /*-- tpro board options --*/

  unsigned char  intLevel;                  /*-- VME interrupt level -*/
  unsigned char  ttagVector;                /*-- time tag int vector -*/
  unsigned char  ppsVector;                 /*-- pps vector -INT32 ---*/

  unsigned char driver[7];                  /*-- driver version ------*/
} /*-----------------------------------------------------------------------*/
TPRO_BoardObj;


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

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


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

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


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

typedef struct TPRO_TimeObj
{ /*-----------------------------------------------------------------------*/
  double        seconds;                    /*-- seconds -------------*/
  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 -----*/
```

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


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

unsigned char TPRO_open             (TPRO_BoardObj *hBoard);
unsigned char TPRO_close            (TPRO_BoardObj *hBoard);

unsigned char TPRO_getAltitude      (TPRO_BoardObj *hBoard, TPRO_AltObj *Altp);
unsigned char TPRO_getLatitude      (TPRO_BoardObj *hBoard, TPRO_LatObj *Latp);
unsigned char TPRO_getLongitude     (TPRO_BoardObj *hBoard, TPRO_LongObj *Longp);
unsigned char TPRO_getTime          (TPRO_BoardObj *hBoard, TPRO_TimeObj *Timep);
unsigned char TPRO_setPropDelayCorr (TPRO_BoardObj *hBoard, int *us);
unsigned char TPRO_setTime          (TPRO_BoardObj *hBoard, TPRO_TimeObj *Timep);
unsigned char TPRO_simEvent         (TPRO_BoardObj *hBoard);
unsigned char TPRO_synchControl     (TPRO_BoardObj *hBoard, unsigned char *enbp);
unsigned char TPRO_synchStatus      (TPRO_BoardObj *hBoard, unsigned char *status);
unsigned char TPRO_waitEvent        (TPRO_BoardObj *hBoard, TPRO_WaitObj *waitp);

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

3.2 TPRO API — Routine Descriptions

3.2.1.1.1 TPRO_open

unsigned char TPRO_open (TPRO_BoardObj *hBoard);

This routine sets the handle to the timing board and initializes the interrupt parameters (level and vector). The A16 base address, all interrupt elements, and the board options element must be set before calling this routine.

Arguments: Pointer to TPRO_BoardObj object

Returns: TPRO_OK on success
TPRO Error code otherwise

3.2.1.1.2 TPRO_close

unsigned char TPRO_close (TPRO_BoardObj *hBoard);

This routine frees the bus handle.

Arguments: Pointer to TPRO_BoardObj object

Returns: TPRO_OK on success
TPRO Error code otherwise

3.2.1.1.3 TPRO_getAltitude

TPRO_getAltitude
unsigned char TPRO_getAltitude (TPRO_BoardObj *hBoard, TPRO_AltObj *Altp);

This routine retrieves altitude information from the timing board. Only TSAT boards can retrieve satellite information.

Arguments: Pointer to TPRO_BoardObj object
Pointer to TPRO_AltObj object

Returns: TPRO_OK on success
TPRO Error code otherwise

3.2.1.1.4 TPRO_getLatitude

unsigned char TPRO_getLatitude (TPRO_BoardObj *hBoard, TPRO_LatObj *Latp);

This routine retrieves latitude information from the timing board. Only TSAT boards can retrieve satellite information.

Note: If the reported value is greater than or equal to 180, subtract 180 degrees and change the sign to negative (representing a latitude south of the equator).

Arguments: Pointer to TPRO_BoardObj object
Pointer to TPRO_LatObj object

Returns: TPRO_OK on success
TPRO Error code otherwise

3.2.1.1.5 TPRO_getLongitude

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

This routine retrieves longitude information from the timing board. Only TSAT boards can retrieve satellite information.

Note: If the reported value is greater than or equal to 180, subtract 180 degrees and change the sign to negative (representing a west longitude).

Arguments: Pointer to TPRO_BoardObj object
Pointer to TPRO_LongObj object

Returns: TPRO_OK on success
TPRO Error code otherwise

3.2.1.1.6 TPRO_getTime

unsigned char TPRO_getTime (TPRO_BoardObj *hBoard, TPRO_TimeObj *Timep);

This routine retrieves the time information from the timing board.

Arguments: Pointer to TPRO_BoardObj object
Pointer to TPRO_TimeObj object

Returns: TPRO_OK on success
TPRO Error code otherwise

3.2.1.1.7 TPRO_setPropDelayCorr

unsigned char TPRO_setPropDelayCorr (TPRO_BoardObj *hBoard, int *us);

This routine sets the propagation delay correction delay factor.

Arguments: Pointer to TPRO_BoardObj object
Pointer to integer containing correction factor in microseconds

Returns: TPRO_OK on success
TPRO Error code otherwise

3.2.1.1.8 TPRO_setTime

unsigned char TPRO_setTime (TPRO_BoardObj *hBoard, TPRO_TimeObj *Timep);

This routine is used to set the current time to the timing board. This routine only works when the board is NOT synchronized to an input.

Arguments: Pointer to TPRO_BoardObj object
Pointer to TPRO_TimeObj object

Returns: TPRO_OK on success
TPRO Error code otherwise

3.2.1.1.9 TPRO_simEvent

unsigned char TPRO_simEvent (TPRO_BoardObj *hBoard);

This routine is used to simulate external time tag events.

Arguments: Pointer to TPRO_BoardObj object

Returns: TPRO_OK on success
TPRO Error code otherwise

3.2.1.1.10 TPRO_synchControl

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

This routine is used to control whether the board synchronizes to external source or "freewheels".

Arguments: Pointer to TPRO_BoardObj object
Pointer to unsigned char containing the values
[TPRO_FREEWHEEL, TPRO_SYNCHRONIZE]

Returns: TPRO_OK on success
TPRO Error code otherwise

3.2.1.1.11 TPRO_synchStatus

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

This routine is used to decipher the timing boards synchronization state (synchronized or freewheeling).

Arguments: Pointer to TPRO_BoardObj object
Pointer to unsigned char containing synchronization status
[TPRO_FREEWHEEL, TPRO_SYNCHRONIZE]

Returns: TPRO_OK on success
TPRO Error code otherwise

3.2.1.1.12 TPRO_waitEvent

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

This routine is used to retrieve the latched time of an external event. The wait object contains a timeout period along with elements of time representing the event time.

Arguments: Pointer to TPRO_BoardObj object
Pointer to TPRO_WaitObj object

Returns: TPRO_OK on success
TPRO Error code otherwise

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