



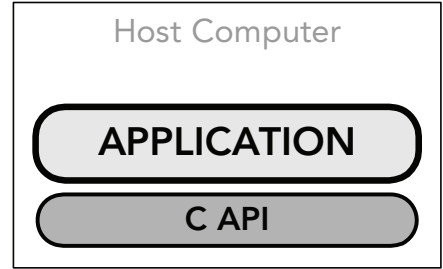
PTU C Language Programmers Interface

Custom programs that directly control pan-tilts

Portable C Language Interface

The PTU C Language Interface (PTU-CPI) allows you to write custom programs that directly control Directed Perception pan-tilts. Some feature highlights of the C Language Interface include:

- C Language Interface (ANSI and K&R) provides easy portability across machines and operating systems
- Compile and link with your host computer program: source code is provided
- Host computer serial port interface code provided for popular operating systems
- Employs binary communications with PTU for dynamic applications such as tracking



```

*****
***** PTU
***** PTU BINARY OPCODES INCLUDE FILE *****
***** (c)1992..2006 Directed Perception, Inc. *****
***** All Rights Reserved. *****
***** Licensed users may freely distribute compiled code including *****
***** this code and data. Source data and code may NOT be *****
***** distributed without the prior written consent from *****
***** Directed Perception, Inc. *****
***** Directed Perception, Inc. reserves the right to make *****
***** changes without further notice to any content herein to *****
***** improve reliability, function or design. Directed Perception *****
***** shall not assume any liability arising from the application *****
***** or use of this code, data or function. *****
*****
*****
CHANGE HISTORY:
2/16/06 v2.12.12r5 Added ISM support and OEM TTL controls
9/ 3/04 v2.12.10 Check for pending asynchronous event in get_current and
get_desired
5/28/04: v2.12.09 Added set_PTU_motion and get_PTU_motion
12/10/99: v1.09.12 added a write and wait after open in open_host_port
for networked operation
11/10/99: v1.09.11. Added asynchronous event status handling functions.
8/10/98: v1.08.09. In firmware_version_OK, removed addressing to string
constant.
7/15/97: v1.08.00. Compiles with MSVC v1.52. Unified with win16/32
PTU interface calls.
6/20/97: #define of ALL misdefined. Fixed to equal PAN+TILT
11/2/95: v1.07.07d. Firmware version check bug fixed.
7/11/95: v1.07.06d. Updated opcode structure and added new support.
2/19/95: v1.07.04d. Generalized for Windows, DOS, & UNIX.
Added networking.
10/12/94: v1.07.03d. Pre-release working version.
XON/XOFF removed from PTU firmware to allow for binary
mode.
*****/
/**/
/**/ Conditionally include the required serial interface declarations. /**/
/**/ If your compiler doesn't have the macro symbol defined, you can /**/
/**/ manually select the right include file yourself. /**/
#if defined(_WIN32)
#include "..\include\w32SERIA.H"
#elif defined(_WIN16)
#include "..\include\w16SERIA.H"

```

Features

Provides full access to all PTU commands for:

- Position Control and Queries
- Initialization/Setup
- Position Limits
- Speed & Acceleration Control and Queries
- Power Control Commands

Allows control bandwidths to 60Hz

Benefits

- Reduce development time
- Reduce development risk via proven software
- Eliminate complexity from application
- Eliminate need to program device command parsers
- Binary communications provide high bandwidth control

General Features

```

/*****
****
****          SAMPLE FUNCTION LISTING (Partial)
****/
/* open_host_port(<portname>, <addr of status variable>) ==> <portstream> */
extern portstream_fd open_host_port(char *);

/* close_host_port(<portstream>) ==> <status> */
extern char close_host_port(portstream_fd);

/* reset_PTU_parser(<timeout_in_msec>) ==> [PTU_OK|PTU_NOT_RESPONDING] */
extern char reset_PTU_parser(long);

/* set_desired( [PAN|TILT], [POSITION|SPEED|ACCELERATION|BASE|UPPER|LOWER],
               [<position>|<speed>|<acceleration>],
               [RELATIVE|ABSOLUTE]) ==> <status>
   set_desired([PAN|TILT], HOLD_POWER_LEVEL, <power mode>, NULL) ==> <status>
   set_desired( [PAN|TILT], [HOLD_POWER_LEVEL,MOVE_POWER_LEVEL], [PTU_REG_POWER|PTU_LOW_POWER|PTU_OFF_POWER],
               NULL) ==> <status>
extern char set_desired(char, char, PTU_PARM_PTR *, char);

/* get_current( [PAN|TILT], [POSITION|SPEED|ACCELERATION|BASE|UPPER|LOWER]
               HOLD_POWER_LEVEL|MOVE_POWER_LEVEL|RESOLUTION)
               ==> <value> */
extern long get_current(char, char);

/* get_desired( [PAN|TILT], [POSITION|SPEED|ACCELERATION|BASE|UPPER|LOWER]
               HOLD_POWER_LEVEL|MOVE_POWER_LEVEL|RESOLUTION)
               ==> <value> */
extern long get_desired(char, char);

/* set_mode( COMMAND_EXECUTION_MODE, [EXECUTE_IMMEDIATELY|EXECUTE_UPON_IMMEDIATE_OR_AWAIT])
               ==> <status>
   set_mode( ASCII_VERBOSE_MODE, [VERBOSE|TERSE|QUERY_MODE]) ==> <status>
   set_mode( ASCII_ECHO_MODE, [ON_MODE|OFF_MODE|QUERY_MODE]) ==> <status>
   set_mode( POSITION_LIMITS_MODE, [ON_MODE|OFF_MODE|QUERY_MODE]) ==> <status>
   set_mode( DEFAULTS, [SAVE_CURRENT_SETTINGS|RESTORE_SAVED_SETTINGS|
                       RESTORE_FACTORY_SETTINGS]) ==> <status> */
extern char set_mode(char,char);
/* halt([ALL|PAN|TILT]) ==> <status> */
extern char halt(char);
/* await_completion() ==> <status> */
extern char await_completion(void);
/* reset_PTU() ==> <status> */
extern char reset_ptu (void);

```



Directed Perception

Advanced and Innovative products for the control and positioning of sensors and precision instruments

890C Cowan Road Burlingame, California 94010

Office 650-692-3900 | Fax 650-692-3930

sales@DPerception.com | www.DPerception.com