

ADVAS-PT1 通信制御 SDK 利用ガイド

文書番号	ADX40015
発行日	2010 年 1 月 7 日
版数	1.0



アドバス株式会社 ADVAS CO., LTD.

変更履歴

版数	日付	変更内容
1.0	2010/01/07	初版

目次

1. はじめに.....	4
2. 特長.....	4
3. SDK 付属のファイル.....	4
3. 1 ヘッダファイル.....	4
3. 2 LIB ファイル.....	4
3. 3 DLL ファイル.....	4
4. API 関数.....	5
4. 1 シリアルポートオープン.....	5
4. 2 接続クローズ.....	5
4. 3 LAN 接続.....	5
4. 4 システム情報読出し.....	5
4. 5 システム設定情報読出し.....	6
4. 6 赤外線リモコン受光設定.....	6
4. 7 工場出荷状態設定.....	6
4. 8 モータ位置リセット.....	7
4. 9 モータのパルス数移動.....	7
4. 10 モータ現在位置取得.....	7
4. 11 モータ連続動作.....	8
4. 12 モータ停止.....	8
4. 13 モータ速度取得.....	8
4. 14 モータ速度設定.....	9
4. 15 ユーザ定義ホームポジション設定.....	9
4. 16 ユーザ定義ホームポジション移動.....	9
4. 17 プリセットポジション登録.....	10
4. 18 プリセットポジション移動.....	10
4. 19 オートリターン時間取得.....	10
4. 20 オートリターン時間設定.....	10
4. 21 オートパイロット実行.....	11
4. 22 PT 状態取得.....	11
4. 23 PT 指定位置移動.....	12
4. 24 PT 位置取得.....	12
4. 25 PT 連携モード取得.....	12
4. 26 PT 連携モード設定.....	13
5. API 関数使用の流れ.....	14

1. はじめに

本ドキュメントは ADVAS-PT1（以下 PT1）と WindowsPC を接続するアプリケーションを開発するための SDK について説明したものです。

2. 特長

PT1 は RS-232C、もしくは LAN（TCP/IP）を通して外部の機器（パソコンなど）から制御することができます。PT1 と外部制御の通信には弊社独自のプロトコルを使って通信します。

本 SDK は WindowsPC を使って PT1 の制御プログラムを開発するに際して、通信プロトコルの細部を意識することなく、開発を進めることができます。

3. SDK 付属のファイル

3. 1 ヘッダファイル

ファイル名 : ptzApi.h

SDK が提供する API 関数についてのプロトタイプを記述したファイルです。C/C++で開発する場合にこのヘッダファイルをインクルードして下さい。

3. 2 LIB ファイル

ファイル名 : ptzApi.lib

C/C++でアプリケーションをビルドする際にリンクするファイルです。

3. 3 DLL ファイル

ファイル名 : ptzApi.dll

作成したアプリケーションを実行する際に必要となるファイルです。

4. API 関数

以下に SDK が提供している API 関数を記述します。

4. 1 シリアルポートオープン

■関数名 PTZ_DevOpen

機能 シリアルポートをオープンする。

```
int PTZ_DevOpen( int port )
```

パラメータ

port

オープンするポート番号。

戻り値

成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

4. 2 接続クローズ

■関数名 PTZ_DevClose

機能 接続をクローズする。

```
int PTZ_DevClose()
```

パラメータ

無し

戻り値

成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

4. 3 LAN 接続

■関数名 PTZ_LanOpen

機能 LAN の接続を行う。

```
int PTZ_LanOpen(char* strIp, int port)
```

パラメータ

strIp IP アドレス指定した文字列。(例"192.168.1.100")

port ポート番号。(2000 固定)

戻り値

成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

4. 4 システム情報読出し

■関数名 PTZ_GetSystemInfo

機能 システム情報を読出す。

```
int PTZ_GetSystemInfo(PTZ_SYSTEM_INFO *systeminfo )
```

パラメータ

systeminfo

PTZ_SYSTEM_INFO 構造体へのポインタ。

戻り値

成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

PTZ_SYSTEM_INFO 構造体

```
typedef struct _PTZ_SYSTEM_INFO
{
    char    main_ver[16];    //メインファームウェアのバージョン
    char    pan_ver[16];    //パンファームウェアのバージョン
    char    tilt_ver[16];    //チルトファームウェアのバージョン
    char    dummy[16];       //未使用
} PTZ_SYSTEM_INFO;
```

4. 5 システム設定情報読出し

■関数名 PTZ_GetSystemSettingInfo

機能 システム設定情報を読出す。

```
int PTZ_GetSystemSettingInfo (PTZ_SYSTEM_SETTING_INFO *settinginfo )
```

パラメータ

settinginfo
PTZ_SYSTEM_SETTING_INFO 構造体へのポインタ。

戻り値

成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

PTZ_SYSTEM_SETTING_INFO 構造体

```
typedef struct _PTZ_SYSTEM_SETTING_INFO
{
    int mode;                //未使用
    int irControl;           //赤外線リモコン受光...0 : OFF、1 : ON
    int sleep;               //未使用
    int led;                 //未使用
} PTZ_SYSTEM_SETTING_INFO;
```

4. 6 赤外線リモコン受光設定

■関数名 PTZ_SetIrControl

機能 赤外線リモコン受光状態を設定する。

```
int PTZ_SetIrControl (int irControl)
```

パラメータ

irControl
0 : リモコン受光 OFF 1 : リモコン受光 ON。

戻り値

成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

4. 7 工場出荷状態設定

■関数名 PTZ_ExecInitialize

機能 工場出荷状態に初期化する。

```
int PTZ_ExecInitialize ()
```

パラメータ
無し

戻り値
成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

4. 8 モータ位置リセット

■関数名 PTZ_ExecMotorHP

機能 モータのホームポジション移動を行う。

```
int PTZ_ExecMotorHP (int motorId)
```

パラメータ
motorId
モータ ID
0:TILT
4: PAN

戻り値
成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

4. 9 モータのパルス数移動

■関数名 PTZ_SetMotorPos

機能 モータのパルス数指定移動を行う。

```
int PTZ_SetMotorPos (int motorId, int direction, int pulse)
```

パラメータ
motorId
モータ ID
direction
方向
PAN モータの場合
0: 左 (反時計回り) 1: 右 (時計回り)
TILT モータ場合
0: 下 1: 上
pulse
移動パルス数。
パルス数は PAN モータの場合 360° が 204000 パルスとなり、TILT モータの場合 32400 パルスとなる。

戻り値
成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

4. 10 モータ現在位置取得

■関数名 PTZ_GetMotorPos

機能 モータの現在位置を取得する。

```
int PTZ_GetMotorPos (int motorId, int *pulse)
```

パラメータ
motorId
モータ ID

pulse
パルス位置。

戻り値
成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

4. 1 1 モータ連続動作

■関数名 PTZ_ActMotor

機能 モータの連続動作を行う。

```
int PTZ_ActMotor (int motorId, int direction)
```

パラメータ

motorId
モータ ID
0:TILT
4:PAN
7:PAN、TILT 同時
direction
方向
PAN モータの場合
0: 左 (反時計回り) 1: 右 (時計回り)
TILT モータの場合
0: 下 1: 上
PAN、TILT モータ同時の場合
0x00: 左下 0x01: 右上 0x10: 左上 0x11: 右上

戻り値
成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

4. 1 2 モータ停止

■関数名 PTZ_StopMotor

機能 モータの停止を行う。

```
int PTZ_StopMotor (int motorId)
```

パラメータ

motorId
モータ ID

戻り値
成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

4. 1 3 モータ速度取得

■関数名 PTZ_GetMotorSpeed

機能 モータの速度を読み出す。

```
int PTZ_GetMotorSpeed (int motorId, int *speed)
```

パラメータ

motorId
モータ ID

speed
 PAN モータの場合
 1～4
 TILT モータ場合
 1～4

戻り値

成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

4. 1 4 モータ速度設定

■関数名 PTZ_SetMotorSpeed

機能 モータの速度を設定する。

```
int PTZ_SetMotorSpeed (int motorId, int speed)
```

パラメータ

motorId
 モータ ID
 speed
 PAN モータの場合
 1～4
 TILT モータ場合
 1～4

戻り値

成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

4. 1 5 ユーザ定義ホームポジション設定

■関数名 PTZ_SetUserHP

機能 ユーザ設定ホームポジションを登録、削除する。

```
int PTZ_SetUserHP (int mode)
```

パラメータ

mode
 0:削除 1:登録

戻り値

成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

4. 1 6 ユーザ定義ホームポジション移動

■関数名 PTZ_MoveUserHP

機能 ユーザ設定ホームポジションに移動する。

```
int PTZ_MoveUserHP ()
```

パラメータ

mode
 無し

戻り値

成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

4. 17 プリセットポジション登録

■関数名 PTZ_SetPresetPos

機能 プリセットポジションを登録、削除する。

```
int PTZ_SetPresetPos (int presetNo, int mode)
```

パラメータ

```
    presetNo
        プリセット番号。
    mode
        0:削除 1:登録
```

戻り値

成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

4. 18 プリセットポジション移動

■関数名 PTZ_MovePresetPos

機能 プリセットポジションに移動する。

```
int PTZ_MovePresetPos (int presetNo)
```

パラメータ

```
    presetNo
        プリセット番号。
```

戻り値

成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

4. 19 オートリターン時間取得

■関数名 PTZ_GetAutoReturn

機能 オートリターン時間を読み出す。

```
int PTZ_GetAutoReturn (int *autoReturn)
```

パラメータ

```
    autoReturn
        オートリターン時間
        0:OFF
        1:30 秒
        2:1 分
        3:2 分
        4:5 分
        5:10 分
```

戻り値

成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

4. 20 オートリターン時間設定

■関数名 PTZ_SetAutoReturn

機能 オートリターン時間を設定する。

```
int PTZ_SetAutoReturn (int autoReturn)
```

パラメータ

autoReturn
オートリターン時間

戻り値

成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

4. 2 1 オートパイロット実行

■関数名 PTZ_ExecAutoPan

機能 オートパイロットを実行する。

```
int PTZ_ExecAutoPan (int mode, int sAngle, int eAngle)
```

パラメータ

mode
モード
0:OFF
1:未使用
2:オートパイロット
sAngle
未使用
eAngle
未使用

戻り値

成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

4. 2 2 PT 状態取得

■関数名 PTZ_GetPTZStatus

機能 PAN、TILT の動作状態を読み出す。

```
int PTZ_GetPTZStatus (PTZ_PTZ_STATUS *ptzStatus)
```

パラメータ

ptzStatus
PTZ_PTZ_STATUS 構造体へのポインタ。

戻り値

成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

PTZ_PTZ_STATUS 構造体

```
typedef struct _PTZ_PTZ_STATUS
{
    int    panStatus;      //0:PAN停止中 1:PAN動作中
    int    tiltStatus;     //0:TILT停止中 1:TILT動作中
    int    dummy1;         //未使用
    int    dummy2;         //未使用
} PTZ_PTZ_STATUS;
```

4. 2 3 PT 指定位置移動

■関数名 PTZ_SetPTPos

機能 PAN、TILT の指定位置（角度）に移動する。

```
int PTZ_SetPTPos (int mode, int panPos, int tiltPos)
```

パラメータ

mode

0：位置指定が移動先角度 1：位置指定が移動パルス数

panPos

mode が 0 の場合

移動先の PAN の角度（×100）。正面を 0° として反時計回り方向が+（プラス）。
360×100 より大きい値を指定すると、無効データとして PAN 動作を行わない。

mode が 1 の場合

移動する PAN のパルス数。反時計回り方向が+（プラス）。

tiltPos

mode が 0 の場合

移動先の TILT の角度（×100）。水平位置を 0° として上方向が+（プラス）。
360×100 より大きい値を指定すると、無効データとして TILT 動作を行わない。

mode が 1 の場合

移動する TILT のパルス数。上方向が+（プラス）。

戻り値

成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

4. 2 4 PT 位置取得

■関数名 PTZ_GetPTPos

機能 PAN、TILT の現在位置（角度）を取得する。

```
int PTZ_GetPTPos (int *panPos, int *tiltPos)
```

パラメータ

panPos

現在の PAN の角度（×100）。正面を 0° として反時計回り方向が+（プラス）。

tiltPos

現在の TILT の角度（×100）。水平位置を 0° として上方向が+（プラス）。

戻り値

成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

4. 2 5 PT 連携モード取得

■関数名 PTZ_GetPanTiltSyncMode

機能 パン、チルト動作連携モードを取得する。

```
int PTZ_GetPanTiltSyncMode ( int *mode )
```

パラメータ

mode

モード

0:OFF

1:ON

戻り値

成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

4. 2 6 PT 連携モード設定

■関数名 PTZ_SetPanTiltSyncMode

機能 パン、チルト動作連携モードを設定する。

```
int PTZ_SetPanTiltSyncMode ( int mode )
```

パラメータ

mode

モード

0:OFF

1:ON

戻り値

成功した場合 PTZ_SUCCESS。失敗した場合 PTZ_ERROR。

5. API 関数使用の流れ

以下に本 SDK の API 関数を使った一般的な処理の流れを表します。

