

VERSION

1.0

KNPICC30

UDP Library Manual .NET Framework 4

Kiko-Net Co., Ltd.

ご注意

本文書の著作権は(有)キコ・ネットが保有します。

本文書の内容を無断で転載することは一切禁止します。

本文書に記載されているサンプルプログラムの著作権は(有)キコ・ネットが保有します。

本文書の内容およびサンプルプログラムに基づき、アプリケーションを運用した結果、万一損害が発生しても、弊社では一切責任を負いませんのでご了承下さい。

本文書の内容については、万全を期して作成いたしましたが、万一ご不審な点、誤りなどお気付きの点がありましたら弊社までご連絡下さい。

本文書の内容は、将来予告なしに変更されることがあります。

商標について

PIC® microcontroller, MPLAB® IDEは、米国およびその他の国々におけるMicrochip Technology Incの登録商標です。

Windows®の正式名称はMicrosoft®Windows®Operating System です。

Microsoft、Windows、Windows NT は、Microsoft Corporation.の米国およびその他の国における商標または登録商標です。

Windows®7、Windows®Vista、Windows®XP、Windows®2000 Professional、

Windows®Millennium Edition、Windows®98 は、Microsoft Corporation の商品名称です。

その他、記載されている会社名、製品名、アプリケーションは各社の登録商標もしくは商標です。

本文書は®、©、™ を記述していない場合があります。

目次

概要.....	5
ファームウェアの配備	6
ライブラリの配備	6
ネットワーク ライブラリ	7
KNPIC30 ファームウェアの構造	9
通信コマンドのサポート	10
ハートビート通信機能	10
通信品質の確認機能.....	10
デバイス内蔵モジュールの診断機能	10
アナウンス通信	11
KnUdpAnnounce クラス	11
コンストラクター	11
プロパティ	11
メソッド	11
イベント	11
Start メソッド	12
Stop メソッド.....	12
AnnounceReceived イベント.....	12
パラメーター:	12
ディスカバリ通信	13
KnUdpDiscovery クラス.....	13
コンストラクター	13
プロパティ	13
ディスカバリ.....	14
リセット.....	14
SendDiscovery メソッド.....	15
SendDiscoveryCompleted イベント	15
BroadcastReset メソッド.....	16
コントロール通信	17
KnUdpControl クラス	17
コンストラクター	17

プロパティ	18
DeviceMode プロパティ	18
共通イベント	18
StatisticsCompleted イベント	19
リセット/ブート	20
Reset メソッド	20
Boot メソッド	20
プログラム書き込み/読み込み	21
ReadProgram メソッド	22
ReadProgramCompleted イベント	22
WriteProgram メソッド	23
WriteProgramCompleted イベント	23
UDP 通信設定情報書き込み/読み込み	24
ReadUdpDescriptor メソッド	25
ReadUdpDescriptorCompleted イベント	25
WriteUdpDescriptor イベント	26
WriteUdpDescriptorCompleted イベント	26
LED コントロール	27
WriteLED メソッド	28
WriteLEDCompleted イベント	28
ReadLED メソッド	29
ReadLEDCompleted イベント	29
SW コントロール	30
ReadSW メソッド	31
ReadSWCompleted イベント	31
IO コントロール	32
WriteIOConfig メソッド	33
WriteIOConfigCompleted イベント	33
ReadIO メソッド	34
ReadIOCompleted イベント	34
WriteIO メソッド	35
WriteIOCompleted イベント	35
Serial コントロール	36
WriteSerialConfig メソッド	38
WriteSerialConfigCompleted イベント	38
OpenSerial メソッド	39
OpenSerialCompleted イベント	39
CloseSerial メソッド	40
CloseSerialCompleted イベント	40

ReadSerial メソッド.....	41
ReadSerialCompleted イベント	41
WriteSerial メソッド	42
WriteSerialCompleted イベント	42
ADC コントロール	43
WriteADCConfig メソッド	44
WriteADCConfigCompleted イベント	44
ReadADC メソッド	45
ReadADCCompleted イベント	45
ハートビート通信 コントロール	46
WriteHeartBeat メソッド	46
WriteHeartBeatCompleted イベント	47
デバイス設定情報一括取得	48
ReadDeviceConfig メソッド	48
ReadDeviceConfigCompleted イベント	49
デバイス統計情報取得	50
ReadDeviceStatistics メソッド	51
ReadDeviceStatisticsCompleted イベント	51
ClearDeviceStatistics メソッド	52
ClearDeviceStatisticsCompleted イベント	52
イベント通信	53
KnUdpEvent クラス	53
コンストラクター	53
プロパティ	53
メソッド	54
共通イベント	54
デバイスイベント	54
StatisticsCompleted イベント	55
EventIOReceived イベント	55
EventSerialReceived イベント	56
EventADCReceived イベント	56
EventHeartBeatReceived イベント	57
共通クラス	58
UdpDescriptor クラス	58
コンストラクター	58
プロパティ	58
IntelForm クラス	59
コンストラクター	59
プロパティ	59

IOConfigArg クラス.....	60
コンストラクター	60
プロパティ	60
SerialPortArg クラス	61
コンストラクター	61
プロパティ	61
ADCConfigArg クラス.....	62
コンストラクター	62
プロパティ	62
DeviceConfig クラス.....	65
コンストラクター	65
プロパティ	65
ComnStatistics クラス	66
コンストラクター	66
プロパティ	66
メソッド.....	67
Clear メソッド.....	67
Add メソッド	67
DeviceStatistics クラス	68
コンストラクター	68
プロパティ	68
IOStatistics クラス	68
コンストラクター	68
プロパティ	68
SerialStatistics クラス.....	69
コンストラクター	69
プロパティ	69
ADCStatistics クラス	69
コンストラクター	69
プロパティ	69
エラー コード表.....	70
Bit-Map とコネクタの対応	71
IO ポート コネクタ マップ	71
Rs232C コネクタ マップ	72
ADC コネクタ マップ	72

概要

本書は、弊社通信端末CPUボードKNPIC30用UDPソフトウェアライブラリについて解説しています。KNPIC30のハードウェア仕様は、『KNPIC30 Hardware Manual』を参照してください。HIDソフトウェアライブラリの仕様は、『KNPIC30 HID Library Manual』を参照してください。WinUSBソフトウェアライブラリの仕様は、『KNPIC30 WinUSB Library Manual』を参照してください。

KNPIC30は、マイクロチップ社製ハーバードアーキテクチャ 16ビット PIC24F CPU を搭載した通信端末用CPUボードです。10/100BASE-TX対応Ethernetコントローラと、USB コントローラ(Full-Speed対応)を搭載しています。RS232Cライン レシーバ/トランシーバを各2チャンネル搭載し、最大2チャンネルのRS232C通信が可能です。デジタルI/Oポートを16本、AD変換器入力を6チャンネル搭載しています。

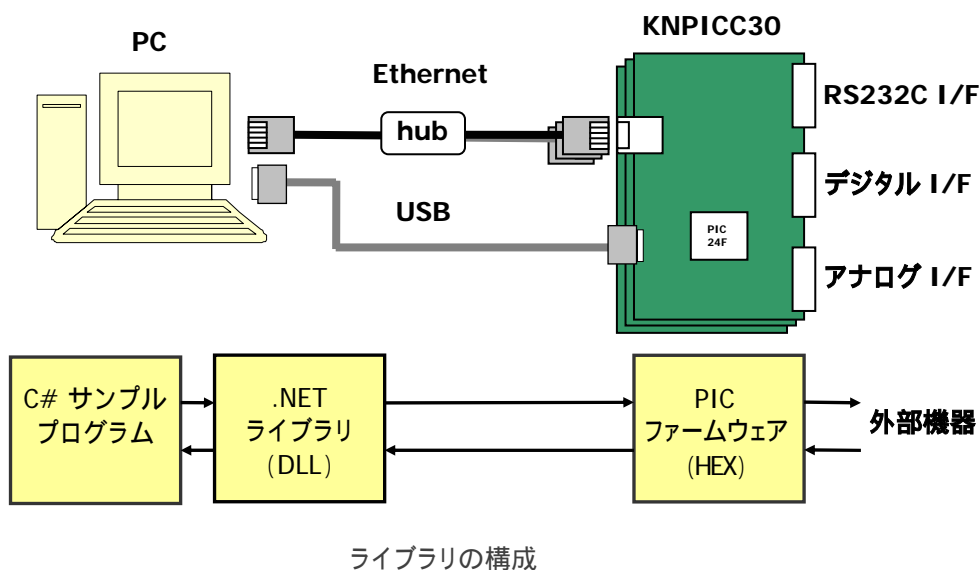
本ソフトウェアライブラリは、KNPIC30用のPICファームウェアを新たに開発することなく、パソコンからKNPIC30の機能 (LED, SW, RS232C, デジタルI/Oポート, AD変換器)を使用することを目的としています。

本ソフトウェアライブラリは、マイクロソフト社.NET Framework 4 ライブラリ(dll) とマイクロチップ社PIC24Fのファームウェアで構成されています。本ソフトウェアライブラリを利用することで、NET Frameworkアプリケーションを作成するだけで、KNPIC30の機能を利用できるようになります。

本ソフトウェアには、次が含まれます。

- I. NET Framework 4 UDP KNPIC30 アクセス クラスライブラリ (PC)
- II. に対応した UDP 通信制御ファームウェア (KNPIC30)
- III. に対応した C# NET Framework UDP 通信 サンプルソースコード (PC)
- IV. C# NET Framework HID プログラマー サンプルソースコード (PC)
- V. に対応した HID ブートローダー ファームウェア (KNPIC30)
- VI. C# NET Framework UDP プログラマー サンプルソースコード (PC)
- VII. に対応した PIC UDP ブートローダー ファームウェア (KNPIC30)

PIC 通信制御ファームウェアは、マイクロチップ社が提供する TCP/IP Stack、USB Framework ファームウェアを使用しています。



ファームウェアの配備

KNPICC30には、工場出荷時に、次のプログラムが事前にインストールされています。ファームウェアの再インストール方法/確認方法については、『KNPICC30 Firmware Manual』を参照してください。

- I. **HID ブートローダー**
UDP ブートローダーのインストール/通信パラメータの設定 (USB)
LED の設定, SW の取得, I/O ポートへのアクセス
- II. **UDP ブートローダー**
通信制御ファームウェアのインストール/通信パラメータの設定 (ネットワーク)
- III. **UDP 通信制御プログラム (ユーザプログラム)**
LED, SW, RS232C, デジタル I/O ポート, AD変換器の制御 (ネットワーク)

ライブラリの配備

KNPICC30を制御するNET Frameworkアプリケーションを作成するには、KNPICC30アクセス クラスライブラリ (**KnApiC30.dll**) をプロジェクトの参照に追加する必要があります。

KNPICC30をHID経由で制御するNET Frameworkアプリケーションを作成するには、HIDアクセス ライブラリ (**KnHidUsb.dll**) を実行ファイルと同じディレクトリに配置する必要があります。

KNPICC30をWinUsb経由で制御するNET Frameworkアプリケーションを作成するには、WinUsbドライバをインストールし、WinUsbアクセス ライブラリ (**KnWinUsb.dll**) を実行ファイルと同じディレクトリに配置する必要があります。

ご注意

KNPICC30 クラスライブラリ、および サンプル プログラム内で、NET Framework 4 以前のNET Framework ライブラリを使用している場合があります。

サンプルを実行するためには、Visual C# 2010 Express(無償)をインストールしてください。実行に必要なNET Frameworkライブラリがすべてインストールされます。

ネットワークライブラリ

ネットワークライブラリは、UDP通信を使用して、ホスト(パソコン)とデバイス(KNPICC30)間の通信を行います。UDP通信は、軽量で高速なプロトコルですが、UDP自身は通信の信頼性(送達確認、到着順序の保障、再送処理)を担保していません。TCP/IPと同等の信頼性を確保するために、ライブラリ内で、送達確認、到着順序の保障、再送処理を行っています。これにより、TCP/IPと同等の信頼性で、高速かつ透過性の高い(アプリケーションから通信の状況を把握しやすい) **コネクションレス**通信プロトコルを実現しています。一般のプロトコルスタックでは、アプリケーション側からプロトコルスタック内部の再送回数やエラーの詳細な原因を知ることはできませんが、本ライブラリのプロトコルスタックは、これらの情報をアプリケーションに提供します。したがって、アプリケーション側で、高度な通信品質の管理を行うことができます。

(注) 本書では、KNPIC30を制御する装置(主にPC)をホストと記述し、KNPIC30をデバイスと記述しています。

ネットワークライブラリには、4つのUDP通信があり、4つのUDPポートを使用します。

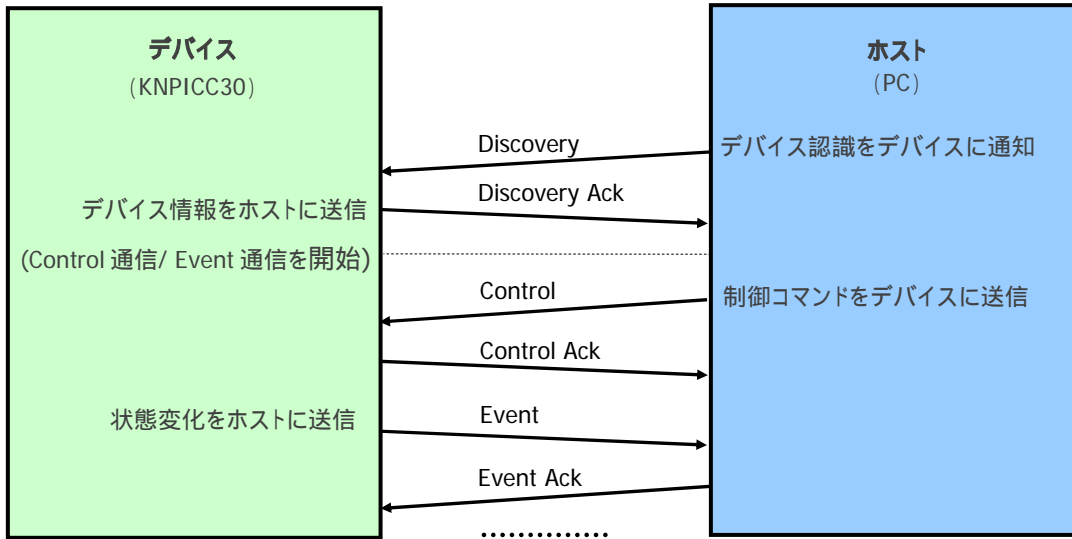
通信名	送信元	送信先	キャスト	機能
アナウンス通信	デバイス	ホスト (片道通信)	ブロード キャスト	デバイスがホストに対して自身の存在を通知
ディスカバリ通信	ホスト	デバイス (往復通信)	ブロード/ユニ キャスト	ホストがデバイスを認識したことをデバイスに通知
コントロール通信	ホスト	デバイス (往復通信)	ユニ キャスト	ホストがデバイスを制御
イベント通信	デバイス	ホスト (往復通信)	ユニ キャスト	デバイスの状態変化をホストに通知

ネットワーク内にDHCPサーバが存在する場合(アナウンス通信有効):

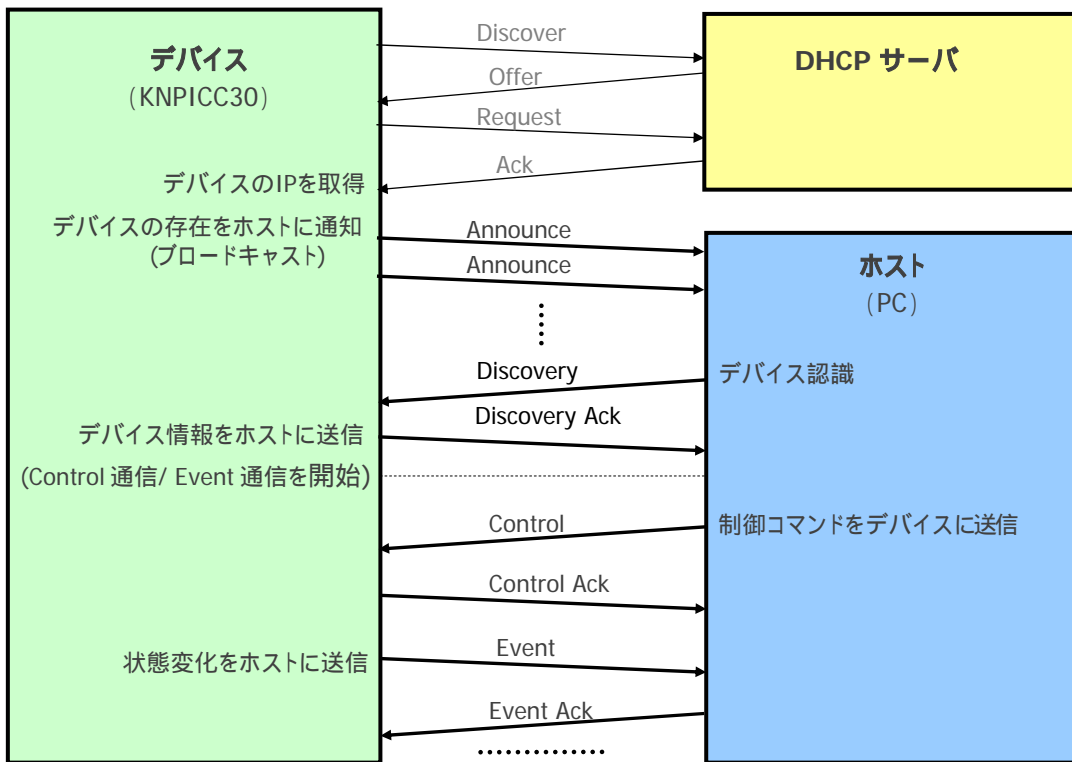
デバイスは、DHCPサーバからIPアドレスを割り当てられると、未知のホストに対して、Announceパケットをブロードキャストで送信します。Announceパケットには、デバイスのMacアドレスが含まれます。デバイスはホストから自分宛のDiscoveryパケットを受信するまで、アナウンス通信を一定周期で繰り返します。Announceパケットを受信したホストは、デバイスを認識したことを通知するために、Discoveryパケットをデバイスに送信します。Discoveryパケットには、アナウンス通信で受信したデバイスのMacアドレスが含まれます。デバイスは、Discoveryパケット内のMacアドレスと自身のMacアドレスを比較し、一致すれば、ホストに対してDiscoveryパケットの応答を返します。この時点で、デバイスは、コントロール通信/イベント通信を開始し、アナウンス通信を停止します。ホストは、Discoveryパケットの応答を受信した時点で、コントロール通信/イベント通信を開始できます。

Macアドレスが既知のデバイスに対して通信を開始する場合(アナウンス通信無効):

Macアドレスを指定したDiscoveryパケットをブロードキャストで送信します。自身のMacアドレスと一致したデバイスは、Discoveryパケットに対して応答を返します。この時点で、デバイスは、コントロール通信/イベント通信を開始します。ホストは、Discoveryパケットの応答を受信した時点で、コントロール通信/イベント通信を開始できます。



Macアドレスが既知のデバイス (アナウンス通信無効)



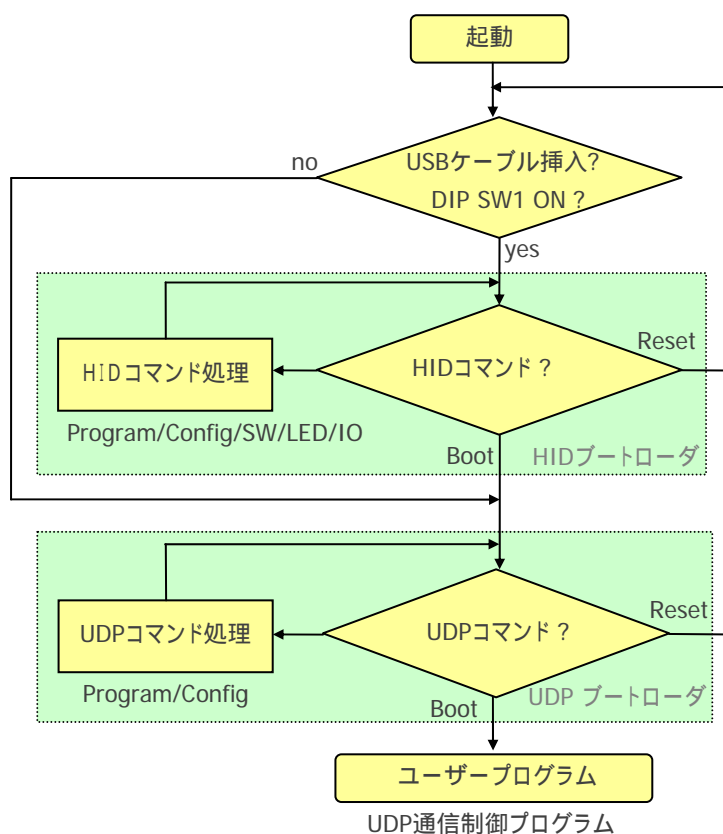
DHCP環境 (アナウンス通信有効)

KNPIC30 ファームウェアの構造

PICファームウェアは、HIDブートローダとUDPブートローダの2重のブートストラップで構成されます。起動時に、USBケーブルが挿入されているか、DIP SW1 が ONであることを検知すると、HIDブートローダが起動し、HIDコマンド待ち状態になります。USBケーブルが挿入されておらず、DIP SW1 が OFFの場合、UDPブートローダが起動し、UDPコマンド待ち状態になります。HIDブートローダの状態では、UDPブートローダプログラムを書き換えることができます。HIDブートローダの状態では、Bootコマンドが入力されると、UDPブートローダが起動します。UDPブートローダの状態では、PIC通信制御プログラムを書き換えることができます。Bootコマンドが入力されると、ユーザープログラム (UDP通信制御プログラム)が起動します。

従って、UDP通信制御プログラム本体を起動させるためには、UDPブートローダの状態では、Bootコマンドを**KNPIC30**に送信する必要があります。また、UDP通信制御プログラムが実行している状態で、UDP通信制御プログラムを書き換える場合、UDP通信制御プログラムに対して、Resetコマンドを送信するか、電源を再投入し、UDPブートローダの状態にしてから、書き換える必要があります。

ファームウェアの詳細については、『**KNPIC30 Firmware Manual**』を参照してください。



起動フロー

通信コマンドのサポート

UDPブートローダは、アナウンス通信、ディスカバリ通信、およびコントロール通信のReset / Boot / Program / UdpDescriptor メソッドをサポートしています。UDP通信制御プログラムは、アナウンス通信、ディスカバリ通信、イベント通信のすべてのイベント、およびコントロール通信のProgram / UdpDescriptor / Boot・メソッド以外のすべてのメソッドをサポートしています。

	アナウンス通信	ディスカバリ通信	コントロール通信	イベント通信
HIDブートローダー	-	-	-	-
UDP ブートローダー			Reset Boot ReadProgram WriteProgram ReadUdpDescriptor WriteUdpDescriptor	×
UDP通信制御プログラム (ユーザープログラム)			Boot ReadProgram WriteProgram ReadUdpDescriptor WriteUdpDescriptor 以外のすべてのコマンド	

ハートビート通信機能

ハートビート通信機能は、ネットワーク上のデバイス(KNPICC30)が自身が正常に稼働していることをホスト(PC)に通知する機能です。デバイス(KNPICC30)から長時間通信がないときに、用がないから黙っているのか、ダウンしてしまったのか確認するために使用します。ハートビート通信機能は、定期的(秒単位)にイベント通信を発生させます。これにより、ホスト(PC)は、デバイス(KNPICC30)が正常に稼働していることを確認できます。ハートビート通信機能は、「ハートビート通信 コントロール」で設定します。

通信品質の確認機能

単に、通信メソッドの実行が成功したかどうかだけでなく、プロトコル スタック内部の詳細な通信情報(試行回数、成功回数、失敗回数、リトライ回数、通信エラーの種類)を取得することができます。ホスト側の通信履歴は、「StatisticsCompleted」、「EventStatisticsCompleted」イベントから取得します。デバイス側の通信履歴は、「ReadDeviceStatistics」メソッドで取得します。

デバイス内蔵モジュールの診断機能

デバイス内蔵モジュール(RS232C, デジタルI/Oポート, AD変換器)の動作異常の情報は、デバイス内に蓄積されています。この情報を読み込むことで、RS232Cパリティエラー、RS232Cフレーミングエラー、バッファオーバーフロー等のエラーの発生をホスト側から検知できます。デバイス内部のエラー情報は、「ReadDeviceStatistics」メソッドで取得します。

アナウンス通信

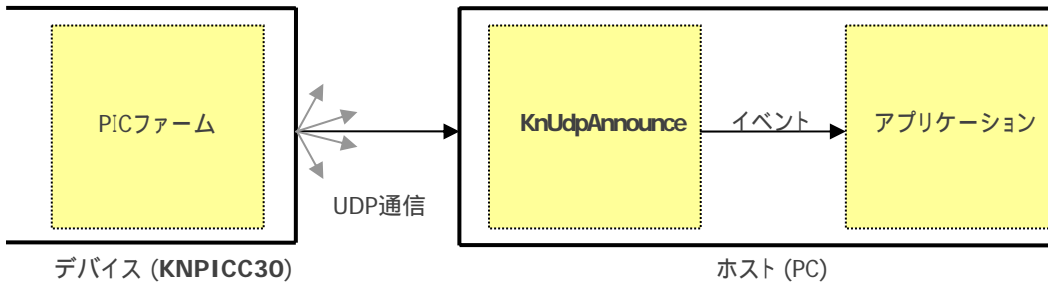
ネットワーク内にDHCPサーバが存在する場合に使用します。デバイスは、DHCPサーバからIPアドレスを割り当てられた後、Announceパケットを送信します。アナウンス通信は、未知のホストに対して、自デバイスの存在を通知するためのブロードキャスト、片道通信です。デバイスがホストからのDiscoveryパケットを受信するまで、この通信は一定周期で繰り返されます。デバイスがホストからのDiscoveryパケットを受信すると、Discoveryパケットに対する応答パケットをホストに送信した後、デバイスは、Announce通信を停止します。

KnUdpAnnounce クラス

.NET Framework 4

名前空間: KnApis.Udp

説明: KnUdpAnnounce クラスは、デバイスからのAnnounce通信を受信し、イベントとしてアプリケーションに通知します。アプリケーションがAnnounce通信を取得するためには、対応するイベントのイベントハンドラを登録する必要があります。



コンストラクター

名前	説明
KnUdpAnnounce	KnUdpAnnounce クラスの新しいインスタンスを初期化します。

プロパティ

名前	型	説明
Port	int	Announce通信を受け付けるポート番号を取得または設定します。

メソッド

名前	説明
Start	Announce通信の受信を開始します。
Stop	Announce通信の受信を停止します。

イベント

名前	説明
AnnounceReceived	Announce通信の到着を通知します。

Start メソッド

解説: Announce通信の受信を開始します。

構文: int Start()

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

Stop メソッド

解説: Announce通信の受信を停止します。

構文: int Stop()

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

AnnounceReceived イベント

解説: Announce通信の受信結果を取得します。

定義: event KnUdpAnnounceReceivedEventHandler AnnounceReceived

デリゲート宣言: delegate void KnUdpAnnounceReceivedEventHandler(int result, IPAddress ip, byte[] mac, int devicemode)

パラメーター:

名前	型	説明
result	int	0: Announce通信受信 負の値: エラー コード表を参照
ip	IPAddress	デバイスのIPアドレス
mac	byte[6]	デバイスのMacアドレス
devicemode	int	デバイスモード: 1: UDP ブートローダ 2: ユーザー プログラム

ディスカバリ通信

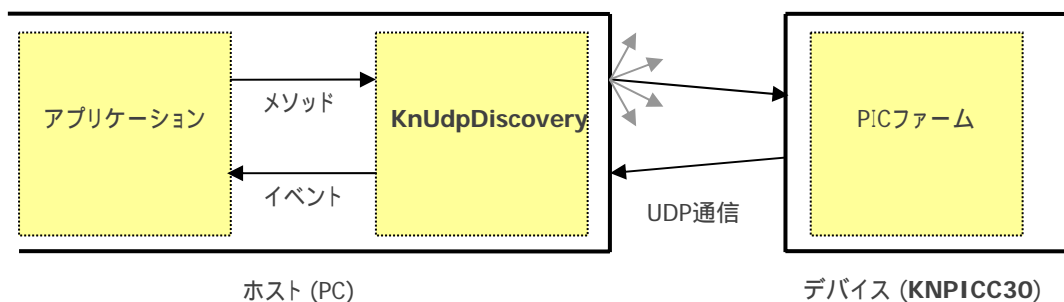
ディスカバリ通信は、ホストがデバイスを認識したことをデバイスに通知し、その応答を受け取るユニキャスト/ブロードキャスト、往復通信です。デバイスは、応答パケットを送信後、コントロール通信とイベント通信を開始します。ディスカバリ通信では、ホストはデバイスに対してMacアドレスを送信します。デバイスは、受信したMacアドレスが自Macアドレスと一致した場合のみ、応答パケットを返信します。ホストがデバイスにDiscoveryパケットを送信後、タイムアウト時間内に応答パケットを受信しなかった場合、または 応答パケットが不正だった場合、指定回数リトライします。指定回数リトライ後、なおも正常な応答パケットを受信しない場合、ディスカバリ通信失敗と判定します。

KnUdpDiscovery クラス

.NET Framework 4

名前空間: KnApis.Udp

説明: KnUdpDiscovery クラスは、ディスカバリ通信を行うための手段を、メソッドとイベントのペアで提供しています。メソッドで、Discoveryパケットをデバイスに送信し、イベントで、デバイスからの応答結果をアプリケーションに通知します。ディスカバリ通信メソッドを実行する前に、対応するイベントのイベントハンドラを登録する必要があります。



コンストラクター

名前	説明
KnUdpDiscovery	KnUdpDiscovery クラスの新しいインスタンスを初期化します。

プロパティ

名前	型	説明
Port	int	ディスカバリ通信に使用するポート番号を取得または設定します。
Timeout	int	ディスカバリ通信のタイムアウト時間を取得または設定します。
Retry	int	通信失敗した場合のリトライ回数を取得または設定します。 (Max 255)
IsBusy	bool	ディスカバリ通信スレッドの状態を取得します。 true: 通信中 false: 通信可能 (通信停止状態)

ディスカバリ

機能: Discoveryパケットを送信します。

メソッド

名前	説明
SendDiscovery	Discoveryパケットを送信します。

イベント

名前	説明
SendDiscoveryCompleted	SendDiscoveryの操作の完了時、キャンセル時に発生します。

リセット

機能: ネットワーク上の複数のデバイスをリセットします。

メソッド

名前	説明
BroadcastReset	RESETコマンドを送信します。

SendDiscovery メソッド

解説: ip で指定されたデバイスに対して、Discoveryパケットを送信します。通信結果は、ConfigIOCompletedイベントで取得します。

構文: int SendDiscovery(IPAddress ip, byte[] mac, int devicemode)

パラメーター:

名前	型	説明
ip	IPAddress	デバイスのIPアドレス (ブロードキャスト可)
mac	byte[6]	デバイスのMacアドレス
devicemode	int	デバイスモード: 1: UDP ブートローダ 2: ユーザー プログラム

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

SendDiscoveryCompleted イベント

解説: Discoveryパケットの送信結果を取得します。SendDiscoveryメソッドの実行後に発生します。

定義: event SendDiscoveryCompletedEventHandler SendDiscoveryCompleted

デリゲート宣言: delegate void SendDiscoveryCompletedEventHandler(int result, DeviceDescriptor dsc)

パラメーター:

名前	型	説明
result	int	0: Discovery送信成功 負の値: エラー コード表を参照
dsc	DeviceDescriptor	デバイス(KNPICC30)情報

BroadcastReset メソッド

解説: ip で指定されたデバイスに対して、Discoveryパケットを送信します。この通信に、応答イベントはありません。

構文: int BroadcastReset (IPAddress ip)

パラメーター:

名前	型	説明
ip	IPAddress	デバイスのIPアドレス (ブロードキャスト可)

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

コントロール通信

コントロール通信は、ホストがデバイスを制御するためにControlパケットを送信し、その応答を受け取るユニキャスト、往復通信です。ホストがデバイスにControlパケットを送信後、タイムアウト時間内に応答パケットを受信しなかった場合、または 応答パケットが不正だった場合、指定回数リトライします。指定回数リトライ後、なおも正常な応答パケットを受信しない場合は、コントロール通信失敗と判定します。

KnUdpControl クラス

.NET Framework 4

名前空間: KnApis.Udp

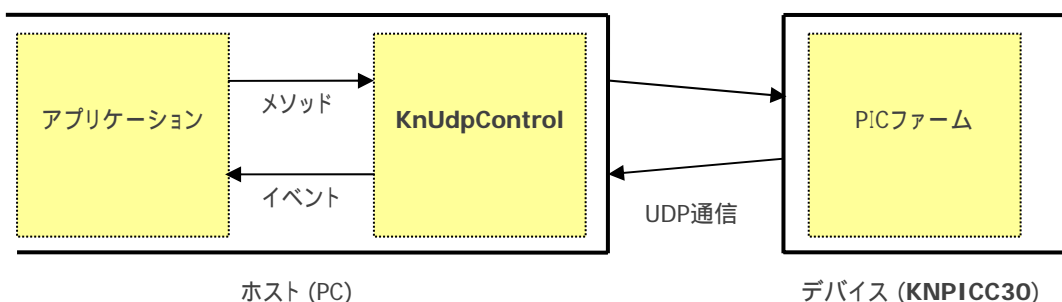
説明: KnUdpControl クラスは、コントロール通信を行うための手段を、メソッドとイベントのペアで提供しています。メソッドで、Controlパケットをデバイスに送信し、イベントで、デバイスからの応答結果をアプリケーションに通知します。コントロール通信メソッドを実行する前に、対応するイベントのイベントハンドラを登録する必要があります。

ご注意

KnUdpControl クラスのメソッドには、同じ名前前で引数の最後に object ID が付くものと、付かないものの2種類のメソッドがあります(オーバーロード)。

例: int ReadSW(IPAddress ip) メソッド(1)
 ReadSW(IPAddress ip, object ID) メソッド(2)
 ReadSWCompleted (int result, int sw1, int sw2, object ID) イベント

メソッド(1)を使用した場合、イベントの ID は nullになります。メソッド(2)を使用した場合、イベントのID はメソッド(2)で指定したIDがそのまま返されます。IDは、プログラム上にメソッドを発行する場所が複数ある場合に、イベント内で、このイベントがどちらのメソッドによるものかを識別するために使用できます。



コンストラクター

名前	説明
KnUdpControl	KnUdpControl クラスの新しいインスタンスを初期化します。

プロパティ

名前	型	説明
Port	int	コントロール通信に使用するポート番号を取得または設定します。
Timeout	int	コントロール通信のタイムアウト時間を取得または設定します。(mS)
Retry	int	通信失敗した場合のリトライ回数を取得または設定します。
DeviceMode	int	コントロール通信のデバイス モードをを取得または設定します。 1: UDP ブートローダ 2: ユーザー プログラム
IsBusy	bool	現在の通信状態を取得します。 true: 通信中 false: 通信可能 (通信停止状態)

DeviceMode プロパティ

解説: コントロール通信のデバイス モードをを取得または設定します。

型	説明
int	コントロール通信のデバイス モードをを取得または設定します。 1: UDP ブートローダ 2: ユーザー プログラム

各デバイス モードで使用できるコマンドは、次のとおりです。

UDP ブートローダー	Reset / Boot / ReadProgram / WriteProgram / ReadConfig / WriteConfig
ユーザー プログラム	Boot / ReadProgram / WriteProgram / ReadConfig / WriteConfig 以外のすべてのコマンド

共通イベント

名前	説明
StatisticsCompleted	すべてのコントロール通信の通信終了時に発生します。コントロール通信情報を返します。

StatisticsCompleted イベント

解説: 各コントロール通信情報を返します。このイベントは、すべてのコントロール通信の通信終了時に発生します。

定義: event KnUdpStatisticsCompletedEventHandler StatisticsCompleted

デリゲート宣言: delegate void KnUdpStatisticsCompletedEventHandler(IPAddress ip, ComnStatistics statistics)

パラメーター:

名前	型	説明
statistics	ComnStatistics	通信情報

通信完了イベントの通信結果引数 **result** は、最後の通信の結果を返しますが、statisticsは、各リトライ毎の通信結果を返します。

例:

通信リトライが3回(Retry = 3)に設定されていて、最初の2回の通信がタイムアウトになり、最後の通信が成功した場合、statistics.Timeout = 2、statistics.Success = 1 となり、イベント引数 **result** は 0 となります。

リセット/ブート

機能: デバイスをリセットします。

メソッド:

名前	説明
Reset	デバイスをリセットします。

機能: ユーザー プログラムを起動します。

メソッド:

名前	説明
Boot	ユーザー プログラムを起動します。 (UDP ブートローダモード時のみ有効)

Reset メソッド

解説: ip で指定したデバイスをリセットします。この通信に、応答イベントはありません。

構文: int Reset(IPAddress ip)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

*ネットワーク上のすべてのデバイスをリセットせる場合、BroadcastReset メソッドを使用します。

Boot メソッド

解説: ip で指定したデバイスをBOOTコマンドを送信します(UDP ブートローダモードのみ)。この通信に、応答イベントはありません。

構文: int Boot(IPAddress ip)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

プログラム書き込み/読み込み

機能: デバイスからユーザー プログラムを読み込みます。

メソッド:

名前	説明
ReadProgram	デバイスからユーザー プログラムを読み込みます。 (UDP ブートローダモード時のみ有効)

イベント:

名前	説明
ReadProgramCompleted	ReadProgramの操作の完了時、キャンセル時に発生します。

機能: デバイスにユーザー プログラムを書き込みます。

メソッド:

名前	説明
WriteProgram	WRITE_PROGRAM設定コマンドを送信し、応答を待ちます。 (UDP ブートローダモード時のみ有効)

イベント:

名前	説明
WriteProgramCompleted	WriteProgramの操作の完了時、キャンセル時に発生します。

ReadProgram メソッド

解説: ip で指定したデバイスからユーザー プログラムを読み込みます。結果は、ReadProgramCompletedイベントで取得します。(UDP ブートローダモードのみ)

構文: int ReadProgram(IPAddress ip, int saddr, int eaddr, bool blankterminate)

int ReadProgram(IPAddress ip, int saddr, int eaddr, bool blankterminate, object ID)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)
saddr	int	読み込み開始アドレス
eaddr	int	読み込み終了アドレス
blankterminate	bool	true: ブランクが128バイト連続した場合読み込みを終了する false: ブランクが128バイト連続しても読み込みを終了しない
ID	object	メソッドの識別子

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

ReadProgramCompleted イベント

解説: ユーザー プログラムの読み込み結果を取得します。ReadProgramメソッドの実行後に発生します。

定義: event KnUdpReadProgramCompletedEventHandler ReadProgramCompleted

デリゲート宣言: delegate void KnUdpReadProgramCompletedEventHandler(int result, List<IntelForm> program, object ID)

パラメーター:

名前	型	説明
result	int	0: 読み込み成功 負の値: エラー コード表を参照
pgm	List<IntelForm>	Intel Hex形式(binary)の読み込みプログラム
ID	object	ReadProgramで指定されたID

WriteProgram メソッド

解説: ip で指定したデバイスにユーザー プログラムを書き込みます。結果は、WriteProgramCompletedイベントで取得します。(UDP ブートロードモードのみ)

構文: int WriteProgram(IPAddress ip, List<IntelForm> program, int saddr, int eaddr)

int WriteProgram(IPAddress ip, List<IntelForm> program, int saddr, int eaddr, object ID)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)
pgm	List<IntelForm>	Intel Hex形式(binary)の書き込みプログラム
saddr	int	消去開始アドレス
eaddr	int	消去終了アドレス
ID	object	メソッドの識別子

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

WriteProgramCompleted イベント

解説: ユーザー プログラムを書き込み結果を取得します。WriteProgramメソッドの実行後に発生します。

定義: KnUdpWriteProgramCompletedEventHandler WriteProgramCompleted

デリゲート宣言: delegate void KnUdpWriteProgramCompletedEventHandler(int result, object ID)

パラメーター:

名前	型	説明
result	int	0: 読み込み成功 負の値: エラー コード表を参照
ID	object	WriteProgramで指定されたID

ご注意

0h - 004000hの領域に ユーザー プログラムを書き込むことはできません。

UDP 通信設定情報書き込み/読み込み

機能: デバイスからUDP通信設定情報を読み込みます。

メソッド:

名前	説明
ReadUdpDescriptor	UDP通信設定情報読み込みコマンドを送信し、応答を待ちます。 (UDP ブートローダモード時のみ有効)

イベント:

名前	説明
ReadUdpDescriptorCompleted	ReadUdpDescriptorの操作の完了時、キャンセル時に発生します。

機能: デバイスにUDP通信設定情報を書き込みます。

メソッド:

名前	説明
WriteUdpDescriptor	UDP通信設定情報書き込みコマンドを送信し、応答を待ちます。 (UDP ブートローダモード時のみ有効)

イベント:

名前	説明
WriteUdpDescriptorCompleted	WriteUdpDescriptorの操作の完了時、キャンセル時に発生します。

ReadUdpDescriptor メソッド

解説: ip で指定したデバイスからUDP通信設定情報を読み込みます。結果は、ReadUdpDescriptorCompletedイベントで取得します。(UDP ブートローダモードのみ)

構文: int ReadUdpDescriptor(IPAddress ip)

int ReadUdpDescriptor(IPAddress ip, object ID)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)
ID	object	メソッドの識別子

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

ReadUdpDescriptorCompleted イベント

解説: UDP設定情報の読み込み結果を取得します。ReadUdpDescriptorメソッドの実行後に発生します。

定義: KnUdpReadUdpDescriptorCompletedEventHandler ReadUdpDescriptorCompleted

デリゲート宣言: delegate void KnUdpReadUdpDescriptorCompletedEventHandler(int result, UdpDescriptor dsc, object ID)

パラメーター:

名前	型	説明
result	int	0: 読み込み成功 負の整数: エラー コード表を参照
dsc	UdpDescriptor	UDP設定情報クラス ReadOnly のパラメータは 常に 0 で意味をもちません
ID	object	ReadUdpDescriptorで指定されたID

WriteUdpDescriptor イベント

解説: ip で指定したデバイスにUDP設定情報を書き込みます。結果は、WriteUdpDescriptorCompletedイベントで取得します。(UDP ブートローダモードのみ)

構文: int WriteUdpDescriptor(IPAddress ip, DeviceDescriptor dsc)

int WriteUdpDescriptor (IPAddress ip, DeviceDescriptor dsc, object ID)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)
dsc	UdpDescriptor	設定情報クラス ReadOnly のパラメータは書き込みません
ID	object	メソッドの識別子

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

WriteUdpDescriptorCompleted イベント

解説: UDP設定情報の書き込み結果を取得します。WriteUdpDescriptorメソッドの実行後に発生します。

定義: event KnUdpWriteComnConfigCompletedEventHandler WriteComnConfigCompleted

デリゲート宣言: delegate void KnUdpWriteComnConfigCompletedEventHandler(int result, object ID)

パラメーター:

名前	型	説明
result	int	0: 読み込み成功 負の値: エラー コード表を参照
ID	object	WriteUdpDescriptorで指定されたID

LED コントロール

機能: デバイスのLEDを設定します。

メソッド:

名前	説明
WriteLED	LED設定コマンドを送信し、応答を待ちます。

イベント:

名前	説明
WriteLEDCompleted	WriteLEDの操作の完了時、キャンセル時に発生します。

機能: デバイスのLEDの状態を取得します。

メソッド:

名前	説明
ReadLED	LED状態取得コマンドを送信し、応答を待ちます。

イベント:

名前	説明
ReadLEDCompleted	ReadLEDの操作の完了時、キャンセル時に発生します。

WriteLED メソッド

解説: ip で指定したデバイスのLEDを設定します。結果は、WriteLEDCompletedイベントで取得します。

構文: int WriteLED(IPAddress ip, int led1, int led2)

int WriteLED(IPAddress ip, int led1, int led2, object ID)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)
led1	int	LEDの状態: 0: 消灯 1: 点灯
led2	int	LEDの状態: 0: 消灯 1: 点灯
ID	object	メソッドの識別子

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

WriteLEDCompleted イベント

解説: LED設定コマンドの通信結果を取得します。WriteLEDメソッドの実行後に発生します。

定義: event KnUdpWriteLEDCompletedEventHandler WriteLEDCompleted

デリゲート宣言: delegate void KnUdpWriteLEDCompletedEventHandler(int result, object ID)

パラメーター:

名前	型	説明
result	int	0: 書き込み成功 負の値: エラー コード表を参照
ID	object	WriteLEDで指定されたID

ReadLED メソッド

解説: ip で指定したデバイスのLEDの状態を取得します。結果は、ReadLEDCompletedイベントで取得します。

構文: int ReadLED(IPAddress ip)

int ReadLED(IPAddress ip, object ID)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)
ID	object	メソッドの識別子

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

ReadLEDCompleted イベント

解説: LED状態取得コマンドの通信結果を取得します。ReadLEDメソッドの実行後に発生します。

定義: event KnUdpReadLEDCompletedEventHandler ReadLEDCompleted

デリゲート宣言: delegate void KnUdpReadLEDCompletedEventHandler(int result, int led1, int led2, object ID)

パラメーター:

名前	型	説明
result	int	0: 取得成功 負の値: エラー コード表を参照
led1	int	LED1の状態: 0: 消灯 1: 点灯
led2	int	LED2の状態: 0: 消灯 1: 点灯
ID	object	ReadLEDで指定されたID

SW コントロール

機能: デバイスのSWの状態を取得します。

メソッド:

名前	説明
ReadSW	SWの状態取得コマンドを送信し、応答を待ちます。

イベント:

名前	説明
ReadSWCompleted	ReadSWの操作の完了時、キャンセル時に発生します。

ReadSW メソッド

解説: ip で指定したデバイスのSWの状態を取得します。結果は、ReadSWCompletedイベントで取得します。

構文: int ReadSW(IPAddress ip)

int ReadSW(IPAddress ip, object ID)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)
ID	object	メソッドの識別子

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

ReadSWCompleted イベント

解説: SW状態取得コマンドの通信結果を取得します。ReadSWメソッドの実行後に発生します。

定義: event KnUdpReadSWCompletedEventHandler ReadSWCompleted

デリゲート宣言: delegate void KnUdpReadSWCompletedEventHandler(int result, int sw1, int sw2, object ID)

パラメーター:

名前	型	説明
result	int	0: 読み込み成功 負の値: エラー コード表を参照
sw1	int	SW1の状態: 0: OFF 1: ON
sw2	int	SW2の状態: 0: OFF 1: ON
ID	object	ReadSWで指定されたID

IO コントロール

機能: デバイスのIOポートの構成を設定します。

メソッド:

名前	説明
WriteIOConfig	IOポートの構成設定コマンドを送信し、応答を待ちます。

イベント:

名前	説明
WriteIOConfigCompleted	WriteIOConfigの操作の完了時、キャンセル時に発生します。

機能: デバイスのIOポートからデータを読み取ります。

メソッド:

名前	説明
ReadIO	IOポートの読み込みコマンドを送信し、応答を待ちます。

イベント:

名前	説明
ReadIOCompleted	ReadIOの操作の完了時、キャンセル時に発生します。

機能: デバイスのIOポートにデータを書き込みます。

メソッド:

名前	説明
WriteIO	IOポートの書き込みコマンドを送信し、応答を待ちます。

イベント:

名前	説明
WriteIOCompleted	WriteIOの操作の完了時、キャンセル時に発生します。

WriteIOConfig メソッド

解説: ip で指定されたデバイスに対して、IOポートの構成設定コマンドを送信します。通信結果は、WriteIOConfigCompletedイベントで取得します。

構文: int WriteIOConfig(IPAddress ip, IOConfigArg param)

int WriteIOConfig(IPAddress ip, IOConfigArg param, object ID)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)
param	IOConfigArg	構成設定データ
ID	object	メソッドの識別子

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

WriteIOConfigCompleted イベント

解説: IOポート設定コマンドの通信結果を取得します。WriteIOConfigメソッドの実行後に発生します。

定義: event KnUdpWriteIOConfigCompletedEventHandler WriteIOConfigCompleted

デリゲート宣言: delegate void KnUdpWriteIOConfigCompletedEventHandler(int result, object ID)

パラメーター:

名前	型	説明
result	int	0: 設定成功 負の値: エラー コード表を参照
ID	object	WriteIOConfigで指定されたID

ReadIO メソッド

解説: ip で指定されたデバイスに対して、IOポート読み込みコマンドを送信します。結果は、ReadIOCompletedイベントで取得します。

構文: int ReadIO(IPAddress ip)

int ReadIO(IPAddress ip, object ID)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)
ID	object	メソッドの識別子

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

ReadIOCompleted イベント

解説: IOポート読み込みコマンドの通信結果を取得します。ReadIOメソッドの実行後に発生します。

定義: event KnUdpReadIOCompletedEventHandler ReadIOCompleted

デリゲート宣言: delegate void KnUdpReadIOCompletedEventHandler(int result, int data, object ID)

パラメーター:

名前	型	説明
result	int	0: 読み込み成功 負の値: エラー コード表を参照
data	int	読み込みデータ
ID	object	ReadIOで指定されたID

WriteIO メソッド

解説: ip で指定したデバイスに対して、IOポートへの書き込みコマンドを送信します。結果は、WriteIOCompletedイベントで取得します。

構文: int WriteIO(IPAddress ip, int data)

int WriteIO(IPAddress ip, int data, object ID)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)
data	int	書き込みデータ
ID	object	メソッドの識別子

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラーコード表を参照

WriteIOCompleted イベント

解説: IOポートの書き込みコマンドの通信結果を取得します。WriteIOメソッドの実行後に発生します。

定義: event KnUdpWriteIOCompletedEventHandler WriteIOCompleted

デリゲート宣言: delegate void KnUdpWriteIOCompletedEventHandler(int result, object ID)

パラメーター:

名前	型	説明
result	int	0: 書き込み成功 負の値: エラーコード表を参照
ID	object	WriteIOで指定されたID

Serial コントロール

機能: デバイスのシリアル ポートの構成を設定します。

メソッド:

名前	説明
WriteSerialConfig	シリアル ポートの構成設定コマンドを送信し、応答を待ちます。

イベント:

名前	説明
WriteSerialConfigCompleted	WriteSerialConfigの操作の完了時、キャンセル時に発生します。

機能: デバイスのシリアル ポートをオープンします。

メソッド:

名前	説明
OpenSerial	シリアル ポートのオープン コマンドを送信し、応答を待ちます。

イベント:

名前	説明
OpenSerialCompleted	OpenSerialの操作の完了時、キャンセル時に発生します。

機能: デバイスのシリアル ポートをクローズします。

メソッド:

名前	説明
CloseSerial	シリアル ポートのクローズ コマンドを送信し、応答を待ちます。

イベント:

名前	説明
CloseSerialCompleted	CloseSerialの操作の完了時、キャンセル時に発生します。

機能: デバイスのシリアル ポートからデータを送信します。

メソッド:

名前	説明
WriteSerial	シリアル ポート送信コマンドを送信し、応答を待ちます。

イベント:

名前	説明
WriteSerialCompleted	WriteSerialの操作の完了時、キャンセル時に発生します。

機能: デバイスのシリアル ポートで受信したデータを取得します。

メソッド:

名前	説明
ReadSerial	シリアル ポート データ取得コマンドを送信し、応答を待ちます。

イベント:

名前	説明
ReadSerialCompleted	ReadSerialの操作の完了時、キャンセル時に発生します。

WriteSerialConfig メソッド

解説: ip で指定されたデバイスに対して、シリアル ポートの構成設定コマンドを送信します。通信結果は、WriteSerialConfigCompleted イベントで取得します。

構文: int WriteSerialConfig(IPAddress ip, int channel)

int WriteSerialConfig(IPAddress ip, int channel, object ID)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)
channel	int	構成設定 2: Rs232C 2チャンネル 1: Rs232C 1チャンネル (Hwフロー制御付)
ID	object	メソッドの識別子

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

WriteSerialConfigCompleted イベント

解説: シリアル ポートの構成設定コマンドの通信結果を取得します。WriteSerialConfigメソッドの実行後に発生します。

定義: event KnUdpWriteSerialConfigCompletedEventHandler WriteSerialConfigCompleted

デリゲート宣言: delegate void KnUdpWriteSerialConfigCompletedEventHandler(int result, object ID)

パラメーター:

名前	型	説明
result	int	0: 設定成功 負の値: エラー コード表を参照
ID	object	WriteSerialConfigで指定されたID

OpenSerial メソッド

解説: ip で指定されたデバイスのシリアル ポートをオープンします。通信結果は、OpenSerialCompleted イベントで取得します。

構文: int OpenSerial(IPAddress ip, int port, SerialPortArg param)

int OpenSerial(IPAddress ip, int port, SerialPortArg param, object ID)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)
port	int	ポート番号 (1 または 2)
param	SerialPortArg	ポート構成データ
ID	object	メソッドの識別子

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

OpenSerialCompleted イベント

解説: シリアル ポートのオープン コマンドの通信結果を取得します。OpenSerialメソッドの実行後に発生します。

定義: event KnUdpOpenSerialCompletedEventHandler OpenSerialCompleted

デリゲート宣言: delegate void KnUdpOpenSerialCompletedEventHandler(int result, int port, object ID)

パラメーター:

名前	型	説明
result	int	0: オープ成功 負の値: エラー コード表を参照
port	int	ポート番号 (1 または 2)
ID	object	OpenSerialで指定されたID

CloseSerial メソッド

解説: ip で指定されたデバイスのシリアル ポートをクローズします。通信結果は、CloseSerialCompletedイベントで取得します。

構文: int CloseSerial(IPAddress ip, int port)

int CloseSerial(IPAddress ip, int port, object ID)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)
port	int	ポート番号 (1 または 2)
ID	object	メソッドの識別子

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

CloseSerialCompleted イベント

解説: シリアル ポートのクローズ コマンドの通信結果を取得します。CloseSerialメソッドの実行後に発生します。

定義: event KnUdpCloseSerialCompletedEventHandler CloseSerialCompleted

デリゲート宣言: delegate void KnUdpCloseSerialCompletedEventHandler(int result, int port, object ID)

パラメーター:

名前	型	説明
result	int	0: クローズ成功 負の値: エラー コード表を参照
port	int	ポート番号
ID	object	CloseSerialで指定されたID

ReadSerial メソッド

解説: ip で指定されたデバイスのシリアル ポートで受信したデータを取得します。通信結果は、ReadSerialCompletedイベントで取得します。

構文: int ReadSerial(IPAddress ip, int port, int len, int timeout)

int ReadSerial(IPAddress ip, int port, int len, int timeout, object ID)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)
port	int	ポート番号 (1 または 2)
len	int	取得するバイト数 (Max 512バイト)
timeout	int	取得するバイト数に対して、データが不足した場合に、シリアルデータの受信を待つ時間 (mS: Max 65534mS)
ID	object	メソッドの識別子

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

ReadSerialCompleted イベント

解説: シリアル ポート読み込み コマンドの通信結果を取得します。ReadSerialメソッドの実行後に発生します。

定義: event KnUdpReadSerialCompletedEventHandler ReadSerialCompleted

デリゲート宣言: delegate void KnUdpReadSerialCompletedEventHandler(int result, int port, int len, byte[] data, object ID)

パラメーター:

名前	型	説明
result	int	0: 受信成功 負の値: エラー コード表を参照
port	int	ポート番号
len	int	取得したバイト数
data	byte[]	取得したデータ (Max 512バイト)
ID	object	ReadSerialで指定されたID

WriteSerial メソッド

解説: ip で指定されたデバイスのシリアル ポートからデータを送信します。通信結果は、WriteSerialCompletedイベントで取得します。

構文: int WriteSerial(IPAddress ip, int port, int len, byte[] data)

int WriteSerial(IPAddress ip, int port, int len, byte[] data, object ID)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)
port	int	ポート番号 (1 または 2)
len	int	送信するバイト数 (Max 512バイト)
data	byte[]	送信データ
ID	object	メソッドの識別子

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

WriteSerialCompleted イベント

解説: シリアル ポートへの送信 コマンドの通信結果を取得します。WriteSerialメソッドの実行後に発生します。

定義: event KnUdpWriteSerialCompletedEventHandler WriteSerialCompleted

デリゲート宣言: delegate void KnUdpWriteSerialCompletedEventHandler(int result, int port, int len, object ID)

パラメーター:

名前	型	説明
result	int	0: 送信成功 負の値: エラー コード表を参照
port	int	ポート番号
len	int	送信したバイト数
ID	object	WriteSerialで指定されたID

ご注意

このメソッドは、デバイスがデータを送信完了してからイベントを返す。

ADC コントロール

機能: デバイスのAD変換器の構成を設定します。

メソッド:

名前	説明
WriteADCConfig	AD変換器の構成設定コマンドを送信し、応答を待ちます。

イベント:

名前	説明
WriteADCConfigCompleted	WriteADCConfigの操作の完了時、キャンセル時に発生します。

機能: デバイスのAD変換データをワンショットで取得します。

メソッド:

名前	説明
ReadADC	ADCデータ取得コマンドを送信し、応答を待ちます。

イベント:

名前	説明
ReadADCCompleted	ReadADCの操作の完了時、キャンセル時に発生します。

WriteADCConfig メソッド

解説: ip で指定されたデバイスに対して、AD変換器の構成設定コマンドを送信します。通信結果は、WriteADCConfigCompletedイベントで取得します。

構文: int WriteADCConfig(IPAddress ip, ADCConfigArg param)

int WriteADCConfig(IPAddress ip, ADCConfigArg param, object ID)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)
param	ADCConfigArg	AD変換器構成設定データ
ID	object	メソッドの識別子

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

WriteADCConfigCompleted イベント

解説: AD変換器の構成設定コマンドの通信結果を取得します。WriteADCConfigメソッドの実行後に発生します。

定義: event KnUdpWriteADCConfigCompletedEventHandler WriteADCConfigCompleted

デリゲート宣言: delegate void KnUdpWriteADCConfigCompletedEventHandler(int result, object ID)

パラメーター:

名前	型	説明
result	int	0: 設定成功 負の値: エラー コード表を参照
ID	object	WriteADCConfigで指定されたID

ReadADC メソッド

解説: ip で指定されたデバイスのAD変換データをワンショットで取得します。通信結果は、ReadADCCompletedイベントで取得します。

構文: int ReadADC(IPAddress ip, int pchannel, int npchannel)

int ReadADC(IPAddress ip, int pchannel, int nchannel, object ID)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)
pchannel	int	正入力チャンネル指定 (0 – 5: AN0 – AN5)
nchannel	int	負入力チャンネル指定 (0: VR- 1: AN1)
ID	object	メソッドの識別子

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

ReadADCCompleted イベント

解説: AD変換読み込み コマンドの通信結果を取得します。ReadADCメソッドの実行後に発生します。

定義: event KnUdpReadADCCompletedEventHandler ReadADCCompleted

デリゲート宣言: delegate void KnUdpReadADCCompletedEventHandler(int result, int data, object ID)

パラメーター:

名前	型	説明
result	int	0: 取得成功 負の値: エラー コード表を参照
data	int	取得したデータ
ID	object	ReadADCCompletedで指定されたID

ハートビート通信 コントロール

機能: デバイスのハートビート通信機能を設定します。

メソッド:

名前	説明
WriteHeartBeat	ハートビート設定コマンドを送信し、応答を待ちます。

イベント:

名前	説明
WriteHeartBeatCompleted	WriteHeartBeatの操作の完了時、キャンセル時に発生します。

WriteHeartBeat メソッド

解説: ip で指定されたデバイスにハートビート通信設定コマンドを送信します。通信結果は、WriteHeartBeatCompletedイベントで取得します。

構文: int WriteHeartBeat(IPAddress ip, int period)

int WriteHeartBeat(IPAddress ip, int period, object ID)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)
period	int	ハートビート通信周期 (0 : 停止、1 – 65535 S)
ID	object	メソッドの識別子

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

ご注意

実際の通信周期は次のとおりです。

ハートビート通信周期 = 設定値 (S) Rand (mS) Rand: 0 から 512までの乱数

WriteHeartBeatCompleted イベント

解説: ハートビート通信設定 コマンドの通信結果を取得します。WriteHeartBeatメソッドの実行後に発生します。

定義: event KnUdpWriteHeartBeatCompletedEventHandler WriteHeartBeatCompleted

デリゲート宣言: delegate void KnUdpWriteHeartBeatCompletedEventHandler(int result, object ID)

パラメーター:

名前	型	説明
result	int	0: 取得成功 負の値: エラー コード表を参照
ID	object	WriteHeartBeatで指定されたID

デバイス設定情報一括取得

機能: デバイスの設定情報 (LED, SW, RS232C, デジタルI/Oポート, AD変換器の設定状態) を一括取得します。

メソッド:

名前	説明
ReadDeviceConfig	設定情報一括取得コマンドを送信し、応答を待ちます。

イベント:

名前	説明
ReadDeviceConfigCompleted	ReadDeviceConfigの操作の完了時、キャンセル時に発生します。

ReadDeviceConfig メソッド

解説: ip で指定されたデバイスの設定情報一括取得コマンドを送信します。通信結果は、ReadDeviceConfigCompletedイベントで取得します。

構文: int ReadDeviceConfig(IPAddress ip)

int ReadDeviceConfig(IPAddress ip, object ID)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)
ID	object	メソッドの識別子

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

ReadDeviceConfigCompleted イベント

解説: 設定情報一括取得コマンドの通信結果を取得します。ReadDeviceConfigメソッドの実行後に発生します。

定義: event KnUdpReadDeviceConfigCompletedEventHandler ReadDeviceConfigCompleted

デリゲート宣言: delegate void KnUdpReadDeviceConfigCompletedEventHandler(int result, DeviceConfigArg param, object ID)

パラメーター:

名前	型	説明
result	int	0: 取得成功 負の値: エラー コード表を参照
param	DeviceConfigArg	設定情報
ID	object	ReadDeviceConfigで指定されたID

デバイス統計情報取得

機能: デバイスの統計情報を取得します。

メソッド:

名前	説明
ReadDeviceStatistics	統計情報取得コマンドを送信し、応答を待ちます。

イベント:

名前	説明
ReadDeviceStatisticsCompleted	ReadDeviceStatisticsの操作の完了時、キャンセル時に発生します。

機能: デバイスの統計情報をクリアします。

メソッド:

名前	説明
ClearDeviceStatistics	統計情報クリアコマンドを送信し、応答を待ちます。

イベント:

名前	説明
ClearDeviceStatisticsCompleted	ClearDeviceStatisticsの操作の完了時、キャンセル時に発生します。

ReadDeviceStatistics メソッド

解説: ip で指定されたデバイスの統計情報取得コマンドを送信します。通信結果は、ReadDeviceStatisticsCompletedイベントで取得します。

構文: int ReadDeviceStatistics(IPAddress ip)

int ReadDeviceStatistics(IPAddress ip, object ID)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)
ID	object	メソッドの識別子

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

ReadDeviceStatisticsCompleted イベント

解説: 統計情報取得コマンドの通信結果を取得します。ReadDeviceStatisticsメソッドの実行後に発生します。

定義: event KnUdpReadDeviceStatisticsCompletedEventHandler ReadDeviceStatisticsCompleted

デリゲート宣言: delegate void KnUdpReadDeviceStatisticsCompletedEventHandler(int result, DeviceStatisticsArg param, object ID)

パラメーター:

名前	型	説明
result	int	0: 取得成功 負の値: エラー コード表を参照
param	DeviceStatistics	デバイス統計情報
ID	object	ReadDeviceConfigで指定されたID

ClearDeviceStatistics メソッド

解説: ip で指定されたデバイスに統計情報クリアコマンドを送信します。通信結果は、ClearDeviceStatisticsCompletedイベントで取得します。

構文: int ClearDeviceStatistics(IPAddress ip)

int ClearDeviceStatistics(IPAddress ip, object ID)

パラメーター:

名前	型	説明
ip	IPAddress	対象となるデバイスのIPアドレス (ブロードキャスト不可)
ID	object	メソッドの識別子

戻り値:

型	説明
int	0: メソッド受付成功 負の値: エラー コード表を参照

ClearDeviceStatisticsCompleted イベント

解説: 統計情報クリアコマンドの通信結果を取得します。ClearDeviceStatisticsメソッドの実行後に発生します。

定義: event KnUdpClearDeviceStatisticsCompletedEventHandler ClearDeviceStatisticsCompleted

デリゲート宣言: delegate void KnUdpClearDeviceStatisticsCompletedEventHandler(int result, object ID)

パラメーター:

名前	型	説明
result	int	0: 取得成功 負の値: エラー コード表を参照
ID	object	ClearDeviceStatisticsで指定されたID

イベント通信

イベント通信は、デバイスがホストに対して、デバイスの状態変化を通知するために、Event パケットを送信し、その応答を受け取るユニキャスト、往復通信です。デバイスがホストに対して、Event パケットを送信後、タイムアウト時間内に応答パケットを受信しなかった場合、または 応答パケットが不正だった場合、指定回数リトライします。指定回数リトライ後、なおも正常な応答パケットを受信しない場合、イベント通信失敗情報をデバイス内の統計情報エリアに登録します。

重要

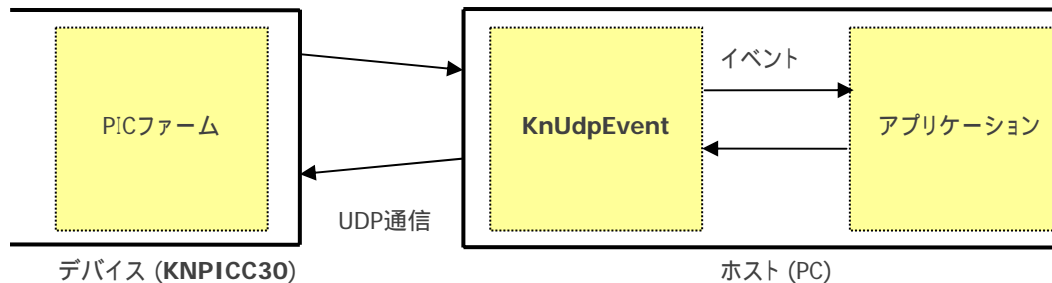
ネットワーク ライブラリは、アプリケーションのイベント処理 (コールバック) が終了してから、デバイスにイベント通信の応答パケットを送信します。アプリケーションのイベント処理 に時間がかかると、通信のスループットが低下します。アプリケーションのイベント コールバック内では、時間のかかる処理 (表示、ファイルへの保存等) を避けてください。

KnUdpEvent クラス

.NET Framework 4

名前空間: KnApis.Udp

説明: KnUdpEvent クラスは、デバイスからのイベント通信を受信し、イベントとしてアプリケーションにその情報を通知します。アプリケーションのイベント処理が終了したら、デバイスに対してEvent応答パケットを送信します。



コンストラクター

名前	説明
KnUdpEvent	KnUdpEvent クラスの新しいインスタンスを初期化します。

プロパティ

名前	型	説明
Port	int	イベント通信に使用するポート番号を取得または設定します。

メソッド

名前	説明
Start	イベント通信を起動します。
Stop	イベント通信を停止します。

共通イベント

名前	説明
StatisticsCompleted	すべてのイベント通信の通信終了時に発生します (通信エラーでも発生します)。イベント通信情報を返します。

デバイスイベント

名前	説明
EventIOReceived	IOポートの状態変化イベントを通知します。
EventSerialReceived	シリアルポートのデータ受信イベントを通知します。
EventADCReceived	AD変換データの取得イベントを通知します。
EventHeartBeatReceived	ハートビート通信受信イベントを通知します。

StatisticsCompleted イベント

解説: 各イベント通信の通信情報を返します。このイベントは、すべてのイベント通信で発生します。通信エラーの場合にも発生し、通信エラーの解析に利用できます。

定義: event KnUdpStatisticsCompletedEventHandler StatisticsCompleted

デリゲート宣言: delegate void KnUdpStatisticsCompletedEventHandler(IPAddress ip, ComnStatistics statistics)

パラメーター:

名前	型	説明
statistics	ComnStatistics	通信情報

EventIOReceived イベント

解説: IOポートの状態変化イベントを通知します。状態変化のトリガーは、IOConfigArg内のTriggerプロパティで指定します。このイベントは、IOConfigArg内の Eventプロパティで有効化/無効化できます (WriteIOConfigメソッドで設定)。

定義: KnUdpEventIOReceivedHandler EventIOReceived

デリゲート宣言: delegate void KnUdpEventIOReceivedHandler(IPAddress ip, int data, int previous)

パラメーター:

名前	型	説明
IPAddress	ip	デバイスのIPアドレス
data	int	現在のIOポートの状態
previous	int	状態変化前のIOポートの状態

*IOポートのbitとコネクタの対応は、『Bit-Mapとコネクタの対応』の「IOポート コネクタ マップ」を参照してください。

EventSerialReceived イベント

解説: シリアル ポートのデータ受信イベントを通知します。イベントの発生条件は、SerialPortArg内のThresholdプロパティとTimeoutプロパティで指定します。このイベントは、SerialPortArg内のEventプロパティで有効化/無効化できます (OpenSerialメソッドで設定)。

定義: event KnUdpEventSerialReceivedHandler EventSerialReceived

デリゲート宣言: delegate void KnUdpEventSerialReceivedHandler(IPAddress ip, int port, int len, byte[] data)

パラメーター:

名前	型	説明
IPAddress	ip	デバイスのIPアドレス
port	int	ポート番号
len	int	受信データ数
data	byte[]	受信データ

EventADCReceived イベント

解説: ADCデータ受信イベントを通知します。イベントの発生条件は、ADCCConfigArg内のThresholdプロパティで指定します。このイベントは、ADCCConfigArg内のIEC0AD1IEプロパティで有効化/無効化できます (WriteADCCConfigメソッドで設定)。

定義: event KnUdpEventADCReceivedHandler EventADCReceived

デリゲート宣言: delegate void KnUdpEventADCReceivedHandler(IPAddress ip, int len, ushort[] data)

パラメーター:

名前	型	説明
IPAddress	ip	デバイスのIPアドレス
len	int	受信データ数
data	ushort[]	受信データ

ADCデータ形式

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
チャンネルID				AD変換データ (000 – 3FF)											

EventHeartBeatReceived イベント

解説: ハートビート通信受信イベントを通知します。イベントの周期は、WriteHeartBeatメソッドで指定します。(0 を指定するとハートビート通信は停止します)。

定義: event KnUdpEventHeartBeatReceivedHandler EventHeartBeatReceived

デリゲート宣言: delegate void KnUdpEventHeartBeatReceivedHandler(IPAddress ip, int id)

パラメーター:

名前	型	説明
IPAddress	ip	デバイスのIPアドレス
id	int	0から始まる連続した値。 ハートビート通信発生毎にインクリメント。

共通クラス

この項には、各通信で使用される汎用的なクラスが含まれます。

UdpDescriptor クラス

.NET Framework 4

名前空間: KnApis.Common

説明: UdpDescriptor クラスは、デバイス(KNPICC30)のUDP通信設定情報を表現するクラスです。

コンストラクター

名前	説明
UdpDescriptor	UdpDescriptor クラスの新しいインスタンスを初期化します。

プロパティ

名前	型	説明
DeviceIP	IPAddress	現在のIPアドレス (ReadOnly)
DeviceSubnet	IPAddress	現在のサブネットマスク (ReadOnly)
DeviceMac	byte[6]	Macアドレス (ReadOnly)
DefaultIP	IPAddress	IPアドレスの初期値
DefaultSubnet	IPAddress	サブネットマスクの初期値
DefaultGateway	IPAddress	デフォルト ゲートウェイ
PrimaryDNS	IPAddress	プライマリDNSサーバ
SecondaryDNS	IPAddress	セカンダリDNSサーバ
DHCPEnable	bool	True: DHCP有効 false: DHCP無効
NetBIOSName	byte[16]	NetBIOS名
HostIP	IPAddress	ホストのIPアドレス (ReadOnly)
HostMac	byte[6]	ホストのMacアドレス (ReadOnly)
AnnouncePort	int	Announce通信用ポート
DiscoveryPort	int	ディスカバリ通信用ポート
ControlPort	int	コントロール通信用ポート
EventPort	int	イベント通信用ポート
AnnounceneEnable	bool	True: Announce通信有効 false: 無効
AnnounceInterval	int	Announce通信インターバル (mS Max 65536)

名前	型	説明
EventTimeOut	int	イベント通信タイムアウト時間 (mS Max 65536)
EventRetry	int	イベント通信リトライ回数 (Max 255)
DeviceMode	int	1: UDP ブートローダ 2: ユーザー プログラム (ReadOnly)
StackVersion	byte[16]	TCP/IPプロトコル スタックのバージョン (ReadOnly)
FirmMajor	int	ファームウェア バージョン (Major) (ReadOnly)
FirmMinor	int	ファームウェア バージョン (Minor) (ReadOnly)

IntelForm クラス

.NET Framework 4

名前空間: KnApis.Common

説明: IntelForm クラスは、IntelHex形式の1レコード(128byte)を表現するクラスです(Binary形式)。デバイスにプログラムを読み込み、または 書き込みをおこなう場合に使用します。

コンストラクター

名前	説明
IntelForm	IntelForm クラスの新しいインスタンスを初期化します。

プロパティ

名前	型	説明
Type	int	レコードのタイプ (通常: 0)。
Length	int	データ長
Offset	int	データの開始アドレス
bData	byte[]	データ (128byte)

IOConfigArg クラス

.NET Framework 4

名前空間: KnApis.Common

説明: IOConfigArgクラスは、デバイス(KNPIC30)のIOポート構成を指定するクラスです。

コンストラクター

名前	説明
IOConfigArg	IOConfigArgクラスの新しいインスタンスを初期化します。

プロパティ

名前	型	説明
Direction	int	IOポート入出力をbit単位で指定します。 0: 出力 1: 入力
OpenDrain	int	IOポート出力形式をbit単位で指定します。 0: トーテン ポール 1: オープン ドレイン (Directionで、出力を指定した場合の意味をもちます)
Trigger	int	イベント通信のトリガーをbit単位で指定します。 0: 状態変化した場合にイベント通信を発生しない 1: 状態変化した場合にイベント通信を発生する
Interval	int	イベント通信 を発生させるためのIOポートのポーリング周期を指定します。(mS: 1 < Interval < 65534)
Event	int	イベント通信の有効化/無効化。 0: イベント通信無効 1: イベント通信有効
Readed	int	IOポートの読み込み値 (ReadOnly)

*IOポートのbitとコネクタの対応は、『Bit-Mapとコネクタの対応』の「IOポート コネクタ マップ」を参照してください。

SerialPortArg クラス

.NET Framework 4

名前空間: KnApis.Common

説明: SerialPortArg クラスは、シリアル ポートの各ポート構成を指定するクラスです。

コンストラクター

名前	説明
SerialPortArg	SerialPortArg クラスの新しいインスタンスを初期化します。

プロパティ

名前	型	説明
BaudRate	int	ボーレートを指定します。 1: 2.4k 2: 9.6k 3: 19.2k 4: 38.4k 5: 57.6k 6: 115.2k
Parity	int	パリティ/データ長を指定します。 1: 8-bit データ, パリティなし 2: 8-bit データ, パリティ EVEN 3: 8-bit データ, パリティ ODD 4: 9-bit データ, パリティなし
StopBit	int	ストップ ビットを指定します。 1: 1ストップ ビット 2: 2ストップ ビット
Event	int	イベント通信の有効化/無効化。 0: イベント通信無効 1: イベント通信有効
Threshold	int	イベント通信 のトリガーとなる受信バイト数の閾値を指定します(0 < Threshold <= 1024)。デバイスの各シリアル ポートで指定したバイト数以上のデータを受信した場合、EventSerialReceived イベントが発生します。(ここで指定したバイト数以上のデータがイベント通信で送信されます。)
Timeout	int	デバイスのシリアル ポートの受信バッファ内に、Thresholdで指定したバイト数未満のデータが存在している場合に、ここで指定したタイムアウト時間内に新たなデータを受信しない場合、EventSerialReceived イベントが発生します。(mS: Max 65534mS)
IsOpen	int	現在のポートの状態 (ReadOnly) 0: クローズ 1: オープン
Count	int	受信バッファに存在するデータ数 (ReadOnly)

ADCConfigArg クラス

.NET Framework 4

名前空間: KnApis.Common

説明: ADCConfigArgクラスは、デバイス(KNPICC30)のAD変換器の構成を指定するクラスです。

ご注意

AD変換の設定を柔軟にするために、ADCConfigArgクラスは、PIC24Fの内蔵ADCのレジスタを直接設定する構成になっています。ADCConfigArgクラスを設定するためには、PIC24Fの内蔵ADCの機能に関する知識が必要です。内蔵ADCの機能詳細は **PIC24FJ256GB110 Family Data Sheet**を参照してください。

コンストラクター

名前	説明
ADCConfigArg	ConfigADCConfigArg クラスの新しいインスタンスを初期化します。

プロパティ

名前	型	説明
Threshold	int	各イベント通信のトリガーとなる閾値を指定します(0 < Threshold <= 512)。指定した数のAD変換が完了した場合、イベント通信が発生します。 ここで指定したデータ数のデータがイベント通信で送信されます。(1データ = 2バイト)
AD1PCFG	int	AN0 – AN5をAD変換器入力として使用するか、IOポートとして使用するかをBit単位で選択します(bit6以上は意味がありません)。 0 : AD変換器入力 1 : IOポート 例: 0x00 : すべてAD変換器入力 0x01 : AN0のみIOポート、その他はAD変換器入力 0x3F : すべてIOポート 0x3E : AN0のみAD変換器入力、その他はIOポート
IEC0AD1IE	int	AD変換割り込み許可 0 : 割り込み禁止 1 : 割り込み許可 割り込み許可するとイベント通信が発生します。

AD1CSSL	int	AD変換をスキャンする入力 (AN0 – AN5)をBit単位で選択します(bit6以上は意味がありません)。 0 : スキャンしない 1 : スキャンする 例: 0x3F : AN0 – AN5すべてスキャンする 0x01 : AN0のみスキャンする 0x03 : AN0、AN1をスキャンする
AD1CON1SSRC	int	AD変換トリガ要因選択 0x03 : Timer3 0x07 : 内蔵カウンタ
AD1CON1ASAM	int	自動サンプリング選択 0x00 : 手動サンプリング 0x01 : 自動サンプリング
AD1CON2VCFG	int	基準電圧選択 0x00 : Vref+ = AVDD Vref- = AVSS 0x01 : Vref+ = 外部 Vref- = AVSS 0x02 : Vref+ = AVDD Vref- = 外部 0x03 : Vref+ =外部 Vref- = 外部
AD1CON2CSCNA	int	スキャン選択 0x00 : スキャンしない 0x01 : スキャンする
AD1CON2SMPI	int	割り込み周期選択 0x00 : 1 AD変換毎 0x01 : 2 AD変換毎 ... 0x0F : 16 AD変換毎
AD1CON3ADCS	int	AD変換クロック選択 0x00 : 0.5TCY 0x01 : 1.0TCY ... 0xFF : 128TCY TCY = 16MHz
AD1CON3SAMC	int	サンプリング時間選択 0x00 : 0TAD 0x01 : 1TAD ... 0x0F : 31TAD TAD = AD変換クロック

PR3	int	Timer3周期選択 0x0000 – 0xFFFF
T3CONTCKPS	int	Timer3プレスケーラ選択 0x00 : 1/1 0x01 : 1/8 0x02 : 1/64 0x03 : 1/256 Timer3クロック = 16MHz
T3CONTON	int	Timer3動作 0x00 : 停止 0x01 : 動作
Count	int	AD変換バッファに存在するデータ数 (ReadOnly)

DeviceConfig クラス

.NET Framework 4

名前空間: KnApis.Common

説明: DeviceConfigクラスは、デバイス(KNPIC30)の設定情報を一括表示するためのクラスです。すべてプロパティは、読み込み専用です。

コンストラクター

名前	説明
DeviceConfig	DeviceConfigクラスの新しいインスタンスを初期化します。

プロパティ

名前	型	説明
LED1	int	LED1の設定状態
LED2	int	LED2の設定状態
SW1	int	SW1の設定状態
SW2	int	SW2の設定状態
IOConfig	IOConfigArg	IOポートの設定状態
SerialConfig	int	シリアルポート構成の設定状態
Serial1Port	SerialPortArg	シリアルポート1の設定状態
Serial2Port	SerialPortArg	シリアルポート2の設定状態
ADCCConfig	ADCCConfigArg	AD変換器の設定状態
HeartBeat	int	ハートビート通信の設定状態

ComnStatistics クラス

.NET Framework 4

名前空間: KnApis.Common

説明: ComnStatisticsクラスのプロパティは、「エラー コード表」に対応しています。ComnStatistics クラスは、通信エラー情報を返します。

コンストラクター

名前	説明
ComnStatistics	ComnStatisticsクラスの新しいインスタンスを初期化します。

プロパティ

名前	型	説明
Try	int	試行回数
Success	int	成功回数
Failure	int	失敗回数
Timeout	int	タイムアウト
Cancel	int	キャンセル
Exception	int	例外発生
Busy	int	スレッドがビジー
NotBusy	int	スレッドがビジーでない
NoCallback	int	コールバックが未登録
Parameter	int	パラメータ エラー
Unknown	int	未知のエラー
DeviceNotSupport	int	デバイスがサポートしていない
DeviceDeny	int	デバイスが拒否
DeviceInternal	int	デバイス内部エラー
DeviceErase	int	フラッシュ イレース エラー
DeviceSend	int	送信エラー

Command	int	コマンド エラー
Length	int	レングス エラー
SeqNumber	int	シーケンス番号 エラー
DeviceMode	int	モード エラー
CtrlCode	int	コントロールコード エラー
ErrorCode	int	エラーコード エラー
HeaderSize	int	ヘッダーサイズ エラー
DataSize	int	データサイズ エラー
AccessTime	int	通信が成功した場合の通信時間

メソッド

名前	説明
Clear	すべてのプロパティを 0 にします。
Add	ComnStatistics クラスの各プロパティに数値を加算します。

Clear メソッド

解説: すべてのプロパティを 0 にします。

構文: void Clear()

パラメーター: なし

戻り値: なし

Add メソッド

解説: プロパティに数値を加算します。

構文: void Add(ComnStatistics statistics)

パラメーター:

名前	型	説明
statistics	ComnStatistics	加算するComnStatistics クラス

戻り値: なし

DeviceStatistics クラス

.NET Framework 4

名前空間: KnApis.Common

説明: DeviceStatisticsクラスは、デバイスのエラー情報を返します。

コンストラクター

名前	説明
DeviceStatistics	DeviceStatisticsクラスの新しいインスタンスを初期化します。

プロパティ

名前	型	説明
Control	ComnStatistics	コントロール通信のエラー情報
Event	ComnStatistics	イベント通信のエラー情報
IO	IOStatistics	IOポートのエラー情報
Serial1	SerialStatistics	シリアルポート1のエラー情報
Serial2	SerialStatistics	シリアルポート2のエラー情報
ADC	ADCStatistics	AD変換器のエラー情報

IOStatistics クラス

.NET Framework 4

名前空間: KnApis.Common

説明: IOStatistics クラスは、IOポートのエラー情報を返します。

コンストラクター

名前	説明
IOStatistics	IOStatisticsクラスの新しいインスタンスを初期化します。

プロパティ

名前	型	説明
Update	int	ポート状態変化回数 (イベント通信)
Stall	int	イベント通信FIFOオーバーフロー

SerialStatistics クラス

.NET Framework 4

名前空間: KnApis.Common

説明: SerialStatistics クラスは、シリアルポートのエラー情報を返します。

コンストラクター

名前	説明
SerialStatistics	SerialStatisticsクラスの新しいインスタンスを初期化します。

プロパティ

名前	型	説明
Sent	int	送信バイト数
Received	int	受信バイト数
Parity	int	パリティエラー回数
Framing	int	フレーミングエラー回数
OverFlow	int	バッファオーバーフロー回数
OverRun	int	バッファオーバーラン回数

ADCStatistics クラス

.NET Framework 4

名前空間: KnApis.Common

説明: ADCStatisticsクラスは、AD変換器のエラー情報を返します。

コンストラクター

名前	説明
ADCStatistics	ADCStatisticsクラスの新しいインスタンスを初期化します。

プロパティ

名前	型	説明
Convert	int	変換回数
OverRun	int	バッファオーバーラン回数

エラーコード表

名前空間: KnApis.Common

列挙子宣言 enum ERRORTYPE

列挙子	数値	説明
SUCCESS	0	成功
TIMEOUT	-1	タイムアウト
CANCEL	-2	キャンセル
EXCEPTION	-3	例外発生
BUSY	-4	スレッドがビジー
NOT_BUSY	-5	スレッドがビジーでない
NOCALLBACK	-6	コールバックが未登録
PARAMETER	-7	パラメータ エラー
UNKNOWN	-8	未知のエラー
DEVICE_NOT_SUPPORT	-20	デバイスがサポートしていない
DEVICE_DENY	-21	デバイスが拒否
DEVICE_INTERNAL	-22	デバイス内部エラー
DEVICE_ERASE	-23	フラッシュ イレース エラー
DEVICE_SEND	-24	送信エラー
COMMAND	-30	コマンド エラー
LENGTH	-31	レングス エラー
SEQUENCENUM	-32	シーケンス番号 エラー
DEVICEMODE	-33	モード エラー
CTRLCODE	-34	コントロールコード エラー
ERRORCODE	-35	エラーコード エラー
HEADERSIZE	-36	ヘッダーサイズ エラー
DATASIZE	-37	データサイズ エラー

Bit-Map とコネクタの対応

IO ポート コネクタ マップ

Bit-Map	コネクタ	ピン番号	CPU ピン番号	CPUピン機能
bit23	CN7	1	24	VREF+/PMA6/CN42/RA10
bit22	CN7	2	23	VREF-/PMA7/CN41/RA9
bit21	CN7	5	20	PGED1/AN0/RP0/CN2/RB0
bit20	CN7	6	19	PGEC1/AN1/RP1/CN3/RB1
bit19	CN7	9	18	AN2/C2INB/VMIO/ RP13 /CN4/RB2
bit18	CN7	10	17	AN3/C2INA/VPIO/CN5/RB3
bit17	CN7	13	16	PGED3/AN4/C1INB/USBOEN/RP28/CN6/RB4
bit16	CN7	14	15	PGEC3/AN5/C1INA/VBUSON/RP18/CN7/RB5
bit15	CN5	1	38	RP5/CN21/RD15
bit14	CN5	2	37	RPI43/CN20/RD14
bit13	CN5	3	65	CN19/RD13
bit12	CN5	4	64	RPI42/CN57/RD12
bit11	CN5	5	56	SCL1/RP3/PMCS2/CN55/RD10
bit10	CN5	6	55	SDA1/DPLN/RP4/CN54/RD9
bit9	CN5	7	54	DMLN/RTCC/RP2/CN53/RD8
bit8	CN5	8	63	RP22/PMBE/CN52/RD3
bit7	CN5	9	62	DPH/RP23/CN51/RD2
bit6	CN5	10	61	VCPCON/RP24/CN50/RD1
bit5	CN5	11	58	DMH/RP11/INT0/CN49/RD0
bit4	CN5	12	53	SDA2/RPI35/CN44/RA15
bit3	CN5	13	52	SCL2/RPI36/CN43/RA14
bit2	CN5	14	73	VCMPST2/CN69/RF1
bit1	CN5	15	74	CN78/RG1
bit0	CN5	16	75	CN77/RG0

*bit23 – bit16 は、ADCを無効化した場合に使用できます。

Rs232C コネクタ マップ

Mode 2 (Rs232Cを2チャンネル使用)

Rs232Cポート	Rs232C機能	コネクタ	ピン番号
1	TXD	CN1	1
1	RXD	CN1	2
2	TXD	CN1	5
2	RXD	CN1	6

Mode 1 (ハードウェア フロー制御Rs232cを2チャンネル使用)

Rs232Cポート	Rs232C機能	コネクタ	ピン番号
1	TXD	CN1	1
1	RXD	CN1	2
1	RTS	CN1	5
1	CTS	CN1	6

ADC コネクタ マップ

ADCチャンネル	ADC機能	コネクタ	ピン番号
0	AN0入力	CN7	5
1	AN1入力	CN7	6
2	AN2入力	CN7	9
3	AN3入力	CN7	10
4	AN4入力	CN7	13
5	AN5入力	CN7	14
-	ADC基準電圧	CN7	1
-	ADC基準電圧	CN7	2