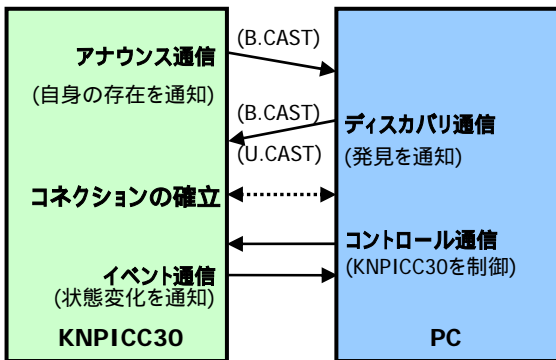


KNPICC30 UDPネットワーク ライブラリの特長 .NET Framework 4

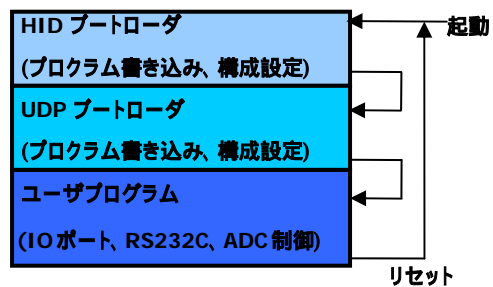
- ◆ **TCP/IPと同等の信頼性:** KNPICC30ネットワーク ライブラリは、UDP通信を使用して、ホスト(PC)とKNPICC30間の通信を行います。UDP通信は、軽量で高速なプロトコルですが、UDP自身は通信の信頼性(送達確認、到着順序の保障、再送処理)を担保していません。TCP/IPと同等の信頼性を確保するために、ライブラリ内で、送達確認、到着順序の保障、再送処理を行っています。これにより、TCP/IPと同等の信頼性で、高速かつ透過性の高い(アプリケーションから通信の状況を把握しやすい) **コネクションレス**の通信プロトコルを実現しています。一般のプロトコル スタックでは、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等を知ることはできません。KNPICC30の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することができ、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。
- ◆ **コネクションの確立:** UDP通信は、コネクションレスの通信プロトコルですが、KNPICC30ネットワーク ライブラリは、KNPICC30がネットワーク上に存在することをホスト(PC)に自発的に通知することができます (KNPICC30起動時、または、DHCPでIP取得時) (**アナウンス通信**)。また、そのホスト(PC)は、そのKNPICC30を認識したことをKNPICC30に通知することができます (**ディスカバリ通信**)。すなわち、両者が互いの存在を確認し合うことができます。
- ◆ **双方向通信:** ホスト(PC)側は、KNPICC30の機能 (LED, SW, RS232C, I/Oポート, AD変換器)を制御することができます (**コントロール通信**)。また、KNPICC30側からも、KNPICC30の状態変化イベント (RS232C受信, I/Oポートの変化, AD変換完了)を自発的にホスト(PC)に通知することができます (**イベント通信**)。
- ◆ **ネットワーク経由でプログラムダウンロード:** ファームウェア側には、HID(USB)、およびUDPブートローダが内蔵されており、USB経由で、または ネットワーク経由で、ファームウェア プログラムの更新ができます。

KNPICC30ネットワークライブラリは、.NET Framework 4 ライブラリ(dll) 形式で提供されており、対となるPICFファームウェアは、バイナリ形式 (Intel Hex)で提供されています。また、複数のKNPICC30を同時に管理するサンプルプログラム(C#)のソース コードが提供されています。KNPICC30用のPICファームウェアを新たに開発することなく、ホスト(PC)とKNPICC30の通信プロトコルを新たに開発することなく、サンプル アプリケーションのソース コード(C#)をカスタマイズするだけで、貴社独自のKNPICC30制御アプリケーションを作成することができます。

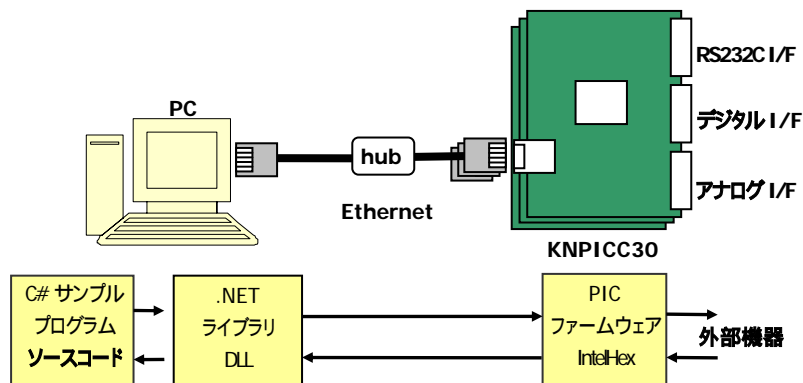
コネクションの確立



ブートストラップ

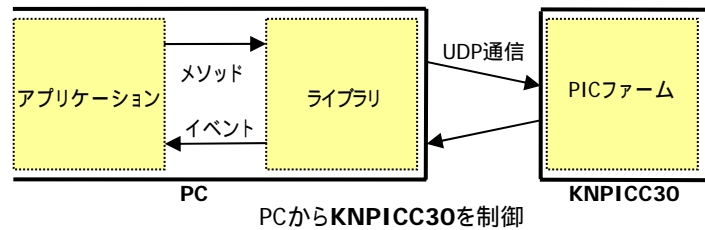


ライブラリ ソフトウェアの構成



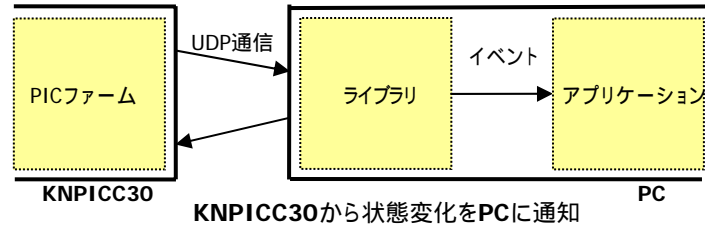
PCからKNPIC30を制御:

制御メソッドを実行し、制御パケットをKNPIC30に送信します。KNPIC30から応答パケットが返されると通信完了イベントが発生し、アプリケーションに通知します(非同期通信)。



KNPIC30から状態変化をPCに通知:

KNPIC30からの状態変化通知パケットを受信すると、状態変化イベントが発生し、アプリケーションに通知します。アプリケーションのコールバック処理が完了したら、応答パケットをKNPIC30に送信します。



制御メソッド (コントロール通信)

Reset	リセットします。
Boot	ユーザー プログラムを起動します。
ReadProgram	ユーザー プログラムを読み込みます。
WriteProgram	ユーザー プログラムを書き込みます。
ReadUdpDescriptor	UDP通信設定情報を読み込みます。
WriteUdpDescriptor	UDP通信設定情報を書き込みます。
WriteLED	LEDを設定します。
ReadLED	LEDの状態を取得します。
ReadSW	SWの状態を取得します。
WriteIOConfig	IOポートの構成を設定します。
ReadIO	IOポートからデータを読み取ります。
WriteIO	IOポートにデータを書き込みます
WriteSerialConfig	シリアル ポートの構成を設定します。
OpenSerial	シリアル ポートをオープンします。
CloseSerial	シリアル ポートをクローズします。
WriteSerial	シリアル ポートからデータを送信します。
ReadSerial	シリアル ポートで受信したデータを取得します。
WriteADCConfig	AD変換器の構成を設定します。
ReadADC	ADCデータを読み取ります。
WriteHeartBeat	ハートビート通信機能を設定します。
ReadDeviceConfig	設定情報 (LED, SW, RS232C, IO, ADCの設定状態) を一括取得します。
ReadDeviceStatistics	統計情報を取得します。
ClearDeviceStatistics	統計情報をクリアします。

状態変化通知イベント (イベント通信)

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。

ブロードキャスト系メソッド (ディスカバリ通信)

SendDiscovery	Discoveryパケットを送信します (接続の確立、KNPIC30の検索)。
BroadcastReset	ネットワーク上の複数のデバイスをリセットします。