

パソコン用とマイコン用の ソケットAPI比較



与曾井 陽一

マイコンで使われるTCP/IP用APIは、表1に示すように、パソコンで使われるソケットAPIを意識した構成になっているので、理解が比較的簡単です。そこで、よく使われるマイコン用OS ITRON^{注1}のTCP/IPのAPI仕様についてパソコン用ソケットAPIと比較しながら解説します。

ITRON用TCP/IPプロトコル・スタックは各社・団体から有償・無償で配布されていますが、使い方がバラバラだとソフトウェアの互換性がなくなるうえ、学習が困難です。そこでITRON向けには表2のような統一されたTCP/IPのAPI仕様が

注1：かつては32ビットCPU向けをITRON、8ビット/16ビットCPU向けをμITRONと呼んでいたが、現在はμITRONで統一されている。

表1 ソケットとITRONのネットワークAPIの比較

ソケット API	μITRON TCP/IP (TCPのみ) API	機能
socket	tcp_cre_cep	通信端点(ソケット)の生成
connect	tcp_con_cep	ソケットの接続
bind	tcp_cre_rep	受付口の生成
listen	—	—
accept	tcp_acp_cep	接続の受け入れ
closesocket	tcp_cls_cep	ソケットの切断
shutdown	tcp_sht_cep	データ送信の終了
setsockopt	tcp_set_opt	通信端点の設定
—	tcp_get_opt	通信端点の参照
—	tcp_can_cep	処理のキャンセル
send	tcp_snd_dat	データの送信
recv	tcp_rcv_dat	データの受信
select	—	—

表2 ITRON TCP/IPのAPI一覧

API名称	機能	API名称	機能
tcp_cre_rep	TCP受付口の生成	tcp_rel_buf	受信用バッファの解放
tcp_del_rep	TCP受付口の削除	tcp_snd_oob	緊急データの送信
tcp_cre_cep	TCP通信端点の生成	tcp_rcv_oob	緊急データの受信
tcp_del_cep	TCP通信端点の削除	tcp_can_cep	ペンディングしている処理のキャンセル
tcp_acp_cep	接続要求待ち(受動オープン)	tcp_set_opt	TCP通信端点オプションの設定
tcp_con_cep	接続要求(能動オープン)	tcp_get_opt	TCP通信端点オプションの読み出し
tcp_sht_cep	データ送信の終了	udp_cre_cep	UDP通信端点の生成
tcp_cls_cep	通信端点のクローズ	udp_del_cep	UDP通信端点の削除
tcp_snd_dat	データの送信	udp_snd_dat	パケットの送信
tcp_rcv_dat	データの受信	udp_rcv_dat	パケットの受信
tcp_get_buf	送信バッファの取得	udp_can_cep	ペンディングしている処理のキャンセル
tcp_snd_buf	バッファ内のデータの送信	udp_set_opt	UDP通信端点オプションの設定
tcp_rcv_buf	受信したデータの入ったバッファの取得	udp_get_opt	UDP通信端点オプションの読み出し

策定されており、理解しておくとも自由にTCP/IPが使えます。
(編集部)

ITRON TCP/IPの特徴

● マイコン用TCP/IPに求められること

ITRON TCP/IP仕様は組み込み機器のためのプロトコル・スタックに求められる以下のような性質を考慮して策定されました。

- (1) バッファのためのメモリ領域や、データのコピーの回数が最小限であること
- (2) できる限りプロトコル・スタック内部で動的なメモリ管理を行う必要がないこと
- (3) 非同期インターフェースないしはノンブロッキング・コール(後述)をサポートすること
- (4) APIごとのエラーの詳細が分かることが望ましい
これらについて以下に見ていきます。

● その1：よく使う用途だけに特化したAPI

ソケットAPIはTCPやUDP以外のさまざまなプロトコルに対応するため、APIが抽象化されています。そのためにconnect, acceptといった名称のAPIが用意されています。しかし現在ではほとんどのケースでTCPもしくはUDPが使用されるため、ITRON TCP/IP仕様ではこの二つのプロトコルに特化されています。表2を見ると、API名称もtcp_con_cepやudp_snd_datのように頭にTCPとUDPが付いています。

● その2：ソケットに似た概念を導入している

ITRON TCP/IPでは通信端点(Connection Point)と呼ばれる