

# μITRONにおけるデバイス・ドライバとミドルウェアの作成事例

山岡 賢一

組み込みシステムに多く搭載例のある、μITRON準拠OS上にデバイス・ドライバとミドルウェアをどう実現するかを解説する。ハードウェアは本誌2007年5月号付属V850 CPU基板とCQ-BB100 A ベースボード、OSはTOPPERS/JSPを使う。μITRON準拠OSではデバイス・ドライバとアプリケーションの明確な区別はない。今回は、NICデバイス・ドライバと通信ミドルウェアを作成する。  
(編集部)

## はじめに

本章では、μITRONにおけるデバイス・ドライバとミドルウェアの作成例を紹介します。対象ハードウェアは、本誌2007年5月号付属V850 CPU基板と、CQ-BB100Aベースボードです(写真1)。μITRONは、TOPPERS/JSPを使用します。V850 CPU基板向けのTOPPERS/JSPは参考文献(3)、参考文献(4)に紹介されているPizzaFactory3向けのものを使用します。

## 1. デバイス・ドライバの概要

μITRON系リアルタイムOSが提供する機能は、メモリ管理とタスク管理、タスク間同期、割り込みハンドラ設定機能、周期ハンドラ設定機能です。OSの機能としてデバイス・ドライバは提供されていません。また、アプリケーションとデバイス・ドライバに明確な区別はありません。簡単な例として、V850 CPU基板上のLEDを点滅させる例を示します(リスト1)。

LEDデバイスを駆動しているのでデバイス・ドライバかもしれませんが、タスク関数で休眠しながらIOポートを制御しているだけです。この例のように、アプリケーションとデバイス・ドライバの区別はありません。ただ、後からアプリケーションを追加するときのことを考えると、デバイスをアクセスする部分は分離しておいた方が得策です。μITRONでは、このデバイス・アクセス部分をデバイス・ドライバと呼びます。

## 2. 作成ソフトウェアの構成

まず最初に、V850 CPU基板+ CQ-BB100A ベースボード+ TOPPERS/JSP上で動作するプログラムを作成しま

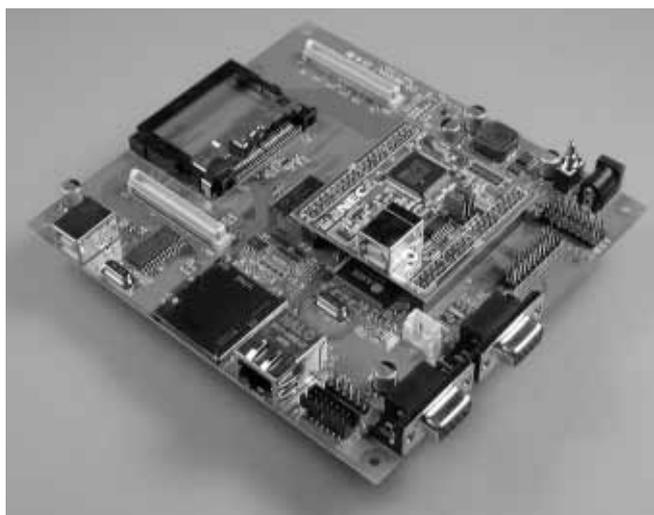


写真1 V850基板とCQ-BB100Aベースボード

リスト1  
V850基板上  
のLEDを点  
滅させる

```
#include <t_services.h>
#define PCT (*(volatile UB*)0xFFFFF00A)
void led_task(VP_INT exinf)
{
    while(1) {
        PCT |= 0x40; /* LED OFF */
        tslp_tsk(500); /* 500ms 休眠 */
        PCT &= ~0x40; /* LED ON */
        tslp_tsk(500); /* 500ms 休眠 */
    }
}
```

す。CQ-BB100AにはNIC(Network Interface Controller)が搭載されており、これを利用してUDP/IP、PING(ICMP)通信プログラムを作成します。繰り返しになりますが、μITRONではアプリケーションとデバイス・ドライバの明確な区別はありません。しかし特集のテーマに合わせて次

## リスト2 デバイス・ドライバ

```

/* NIC デバイス初期化 */
void lan_init(unsigned char* mac)
{
    IO_Adr = 0;
    NIC_Init(mac);
    NIC_Interrupt_Init();
    NIC_SendEnableCtrl(2);
    NIC_RecvEnableCtrl(1);
    AX88796_START();
    act_tsk(LAN_RX_TASK);
    *((volatile unsigned char *)0xFFFFF112)=0x07;
}
/* 受信処理タスク */
void lan_rcv_task(VP_INT exinf)
{
    static rx_pkt_t pkt;
    while(1) {
        /* 受信パケット発生待ち */
        while (NIC_CheckRecvPacket()==0) {
            tslp_tsk(100); /* 休眠(割り込みハンドラが解除する) */
        }
        /* NICから受信パケットを読み出す */
        pkt.len = NIC_GetRecvPacket(pkt.data);
        /* 通信ミドルウェアに受信パケットを渡す */
        mac_input(&pkt);
    }
}
/* 送信処理関数 */
void lan_send(const void *dat1, int len1, const void *dat2,
int len2)
{
    static unsigned char tmp[1500];
    while(g_lan_tx_tid) { /* 先に待ち送信待ちタスクがある */
        tslp_tsk(10);
    }
    get_tid(&g_lan_tx_tid);
    while(1) {
        if (dat2==NULL) {
            if (NIC_SetSendPacket((UBYTE*)dat1, len1)==0)
                break; /* 送信成功 */
        } else {
            memcpy(tmp, dat1, len1);
            memcpy(tmp+len1, dat2, len2);
            if (NIC_SetSendPacket(tmp, (len1+len2))==0)
                break; /* 送信成功 */
        }
        /* 送信可能状態でないので休眠(割り込みハンドラが解除する) */
        tslp_tsk(100);
    }
    g_lan_tx_tid = 0;
}
/* 割り込みハンドラ */
void lan_interrupt(unsigned int arg)
{
    NIC_Interrupt();
    if (NIC_CheckSendBuffer()==0) {
        if (g_lan_tx_tid) {
            /* 送信可能待ちタスクがあるので休眠解除 */
            iwup_tsk(g_lan_tx_tid);
        }
    }
    if (NIC_CheckRecvPacket()) {
        /* 受信パケットが発生したので受信処理タスクの休眠解除 */
        iwup_tsk(LAN_RX_TASK);
    }
}
}

```

の二つの部分から構成します。

### ●通信ミドルウェア

アプリケーション向けに、PING送受信とUDP送受信APIを提供する。

### ●NICデバイス・ドライバ

通信ミドルウェア向けにパケット送受信インターフェースを提供する。

図1にソフトウェアの構成を示します。アプリケーションは独立したタスクとして動作します。通信ミドルウェアAPIを呼び出してPINGやUDP通信を行います。NICドライバは、NICデバイスからパケットの受け渡しを担当します。パケット受信は、NICドライバの受信処理タスクが通信ミドルウェア関数に渡して処理します。送信パケットは、アプリケーションが送信API関数を呼び出すとミドルウェアがパケットの組み立てを行い、NICデバイス・ドライバに渡します。

## 3. NICデバイス・ドライバ

CQ-BB100A ベースボードには、NICとしてASIX AX88796が搭載されています。このデバイスの詳細は参考文献(1)に解説されており、付属CD-ROMにはAX88796デバイス制御プログラムのソース・コードが収録されています。本章で作成するドライバは、デバイス制御部分はすべて参考文献(1)付属CD-ROMに収録したソース・コードを流用しています。

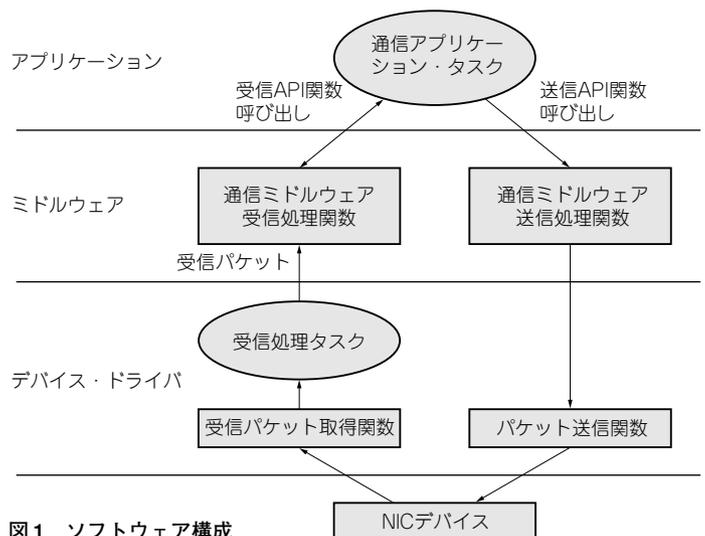


図1 ソフトウェア構成