

意外と簡単に ネットワーク接続はできる

編集部

ネット接続できたらうれしいこと

組み込み機器を**ネットワーク**に対応させるとどんな**利点**があるんだろう？

今までは表示装置や入力装置、大容量のフラッシュROMが必要だったけど、ネットワーク経由にすれば不要になる！
装置を超シンプルにできるよね

今まで！



ネット接続対応すれば作るのは**これだけ！**

これから！

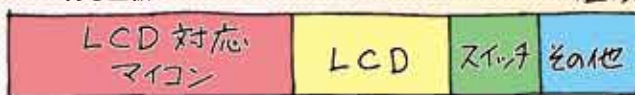
Ethernet対応マイコン

超シンプル

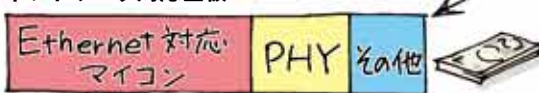


データは**サーバ**に保存

LCD付き基板



ネットワーク対応基板



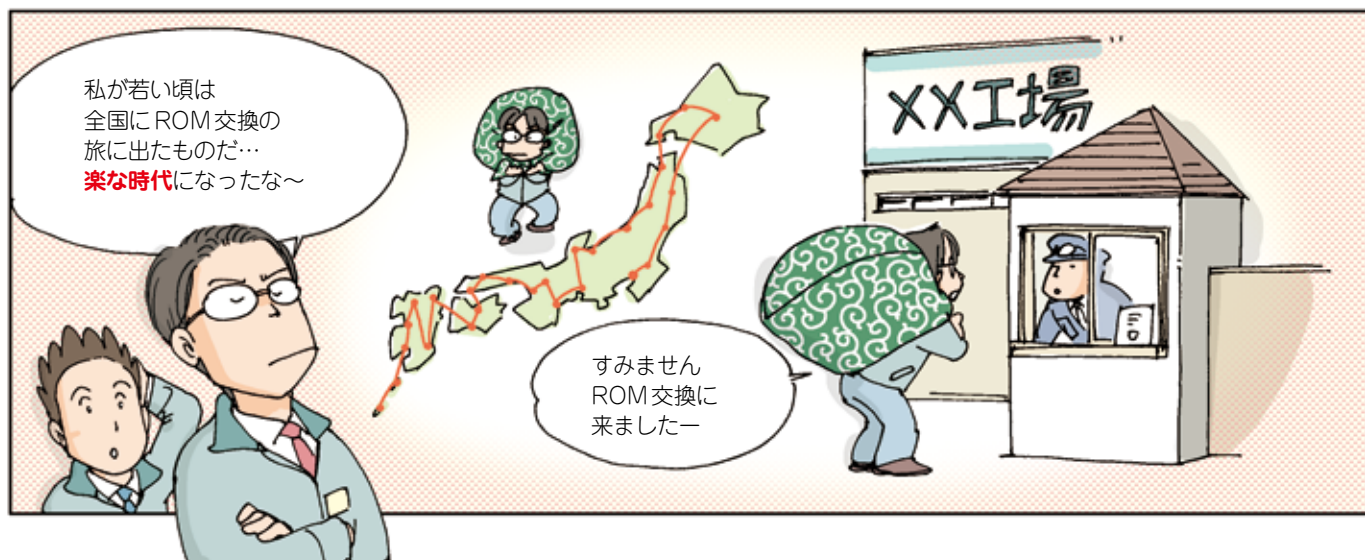
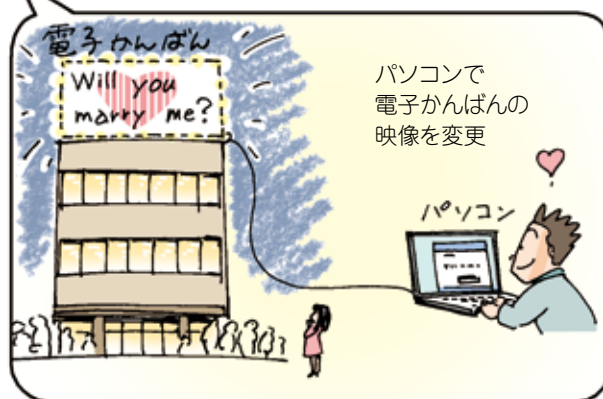
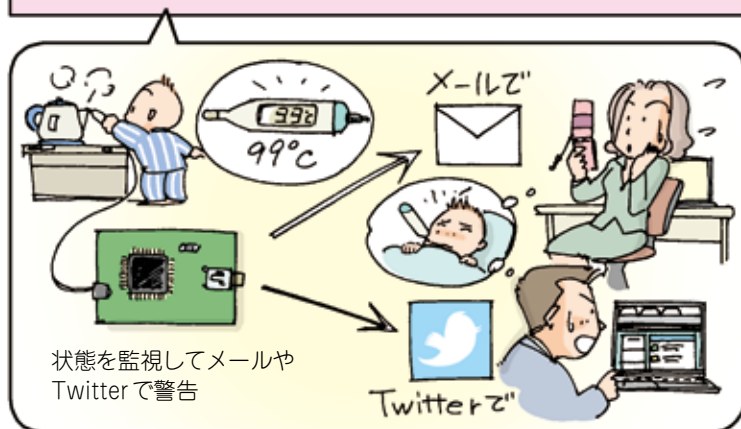
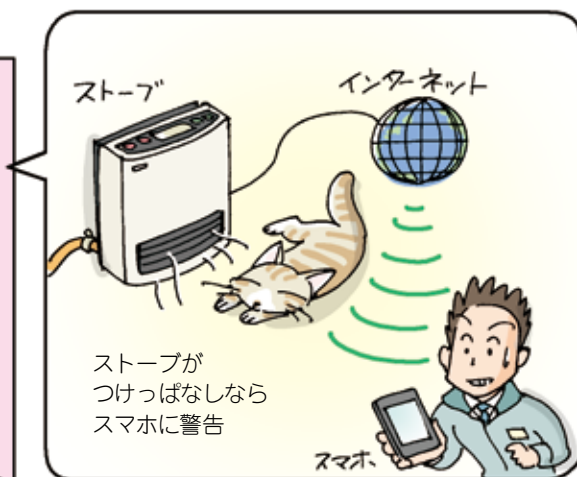
コストダウン

ディスプレイ用のLCDが不要で、大容量フラッシュROMがいらないなら、**ずいぶんコストダウン**になりますね

さらにこういうこともできる！



- ① 機器の**状態**を **Twitter** でつぶやく / 画像を **自動で投稿**
- ② **携帯電話**や**スマホ**から組み込み機器の状態を監視できる
- ③ 緊急事態は**メール**でお知らせ！
- ④ 無線LANルータを経由すれば、**無線化**することも可能
- ⑤ 電子かんぱんの映像データを**設置場所に行かずに更新**



Intro

Pre

1

2

3

4

App

5

6

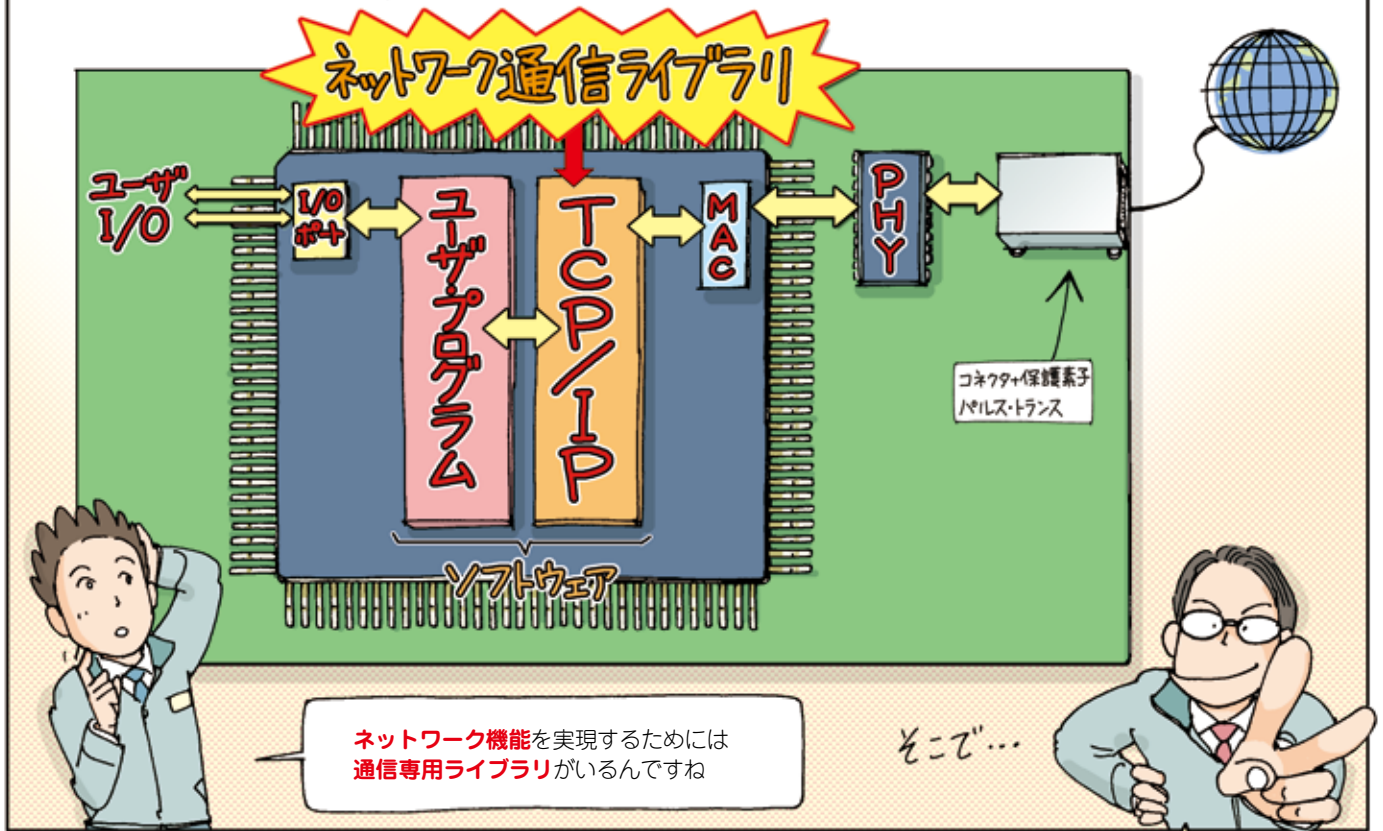
7

8

9

ネット接続を実現するには？

～ネットワーク対応機器の構成～



ネットワーク通信ライブラリを用意する二つの方法

移植済みを買ってくる派

最近**便利**だね

移植済み!

安価なネットワーク対応マイコン基板も

¥3,000～

ネットワーク・ライブラリ移植済みの基板を買ってくる

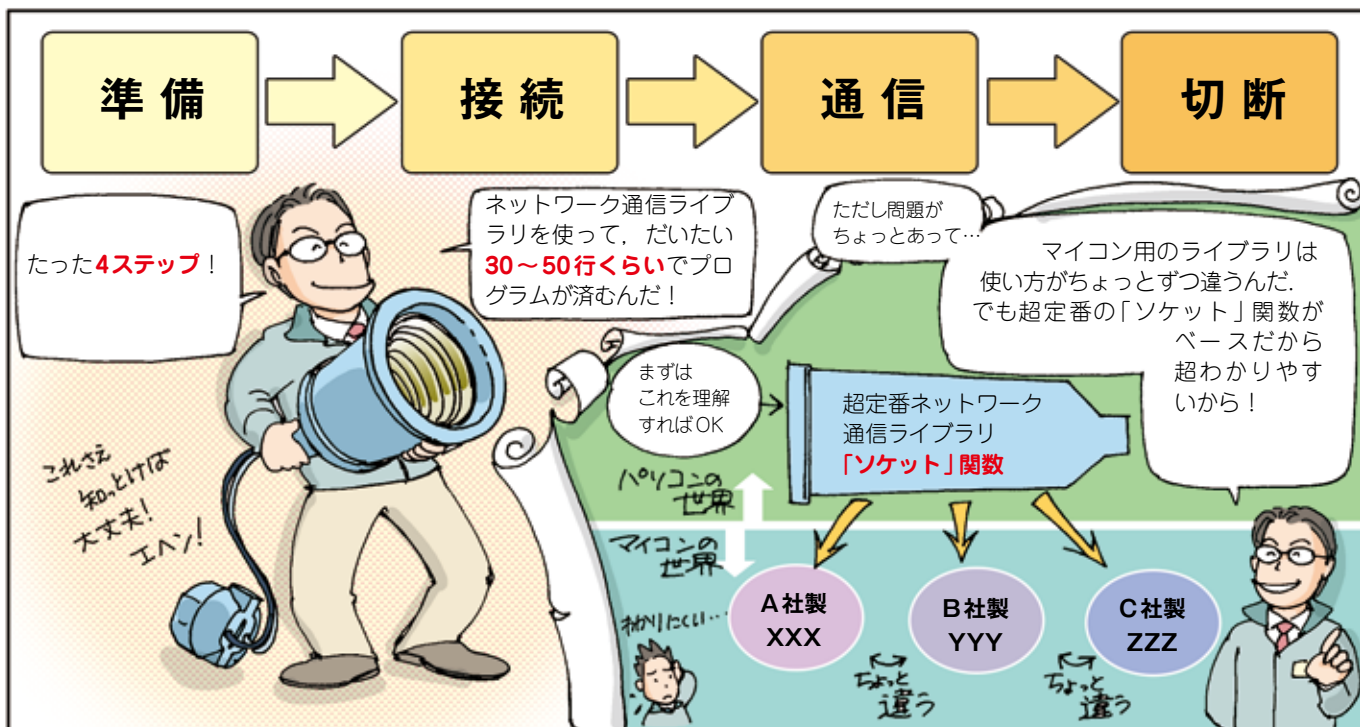
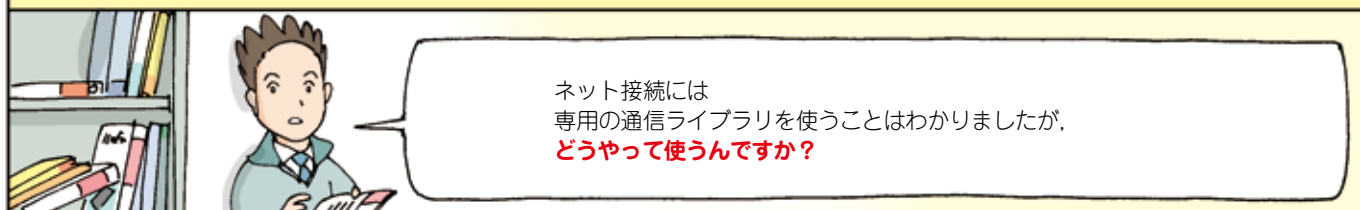
⇔

自分で移植する派

無料のネットワーク通信ライブラリも

ネットワークのライブラリはすでにあるものを使い、自分のマイコンで動くように移植する

ネットワーク通信ライブラリを使うと超楽チン！



ソケットを知っていればよくわかる！

```
#define BSIZE (10*1024)
main()
{
    int s, c, n, nR, nS;
    char *p;
    char buf[BSIZE];
    struct sockaddr_in svr;
    struct sockaddr_in cli;

    s = socket(AF_INET, SOCK_STREAM, 0);
    memset(&svr, 0, sizeof(svr));
    svr.sin_family = AF_INET;
    svr.sin_addr.s_addr = htonl(INADDR_ANY);
    svr.sin_port = htons(49152);

    bind(s, (struct sockaddr *)&svr, sizeof(svr));
    listen(s, 1);

    printf("svr: Waiting for connection ...Yn");
    n = sizeof(cli);
    c = accept(s, (struct sockaddr *)&cli, &n);
    printf("svr: connected from %sYn", inet_ntoa(cli.sin_addr));

    while (nR = recv(c, buf, BSIZF, 0)) {
        printf("svr: received %dYn", nR);
        printf("svr: echo backYn");
        if ((nS = send(c, buf, nR, 0)) < nR)
            printf("svr: can'tYn");
    }
    close(c);
    close(s);
    printf("svr: connection closedYn");
    return (0);
}
```

たった50行くらいでできる

ソケット関数が分かっていれば理解が簡単

