

付属ColdFireマイコン基板の活用

ワンチップネットワーク時代を先取りしよう

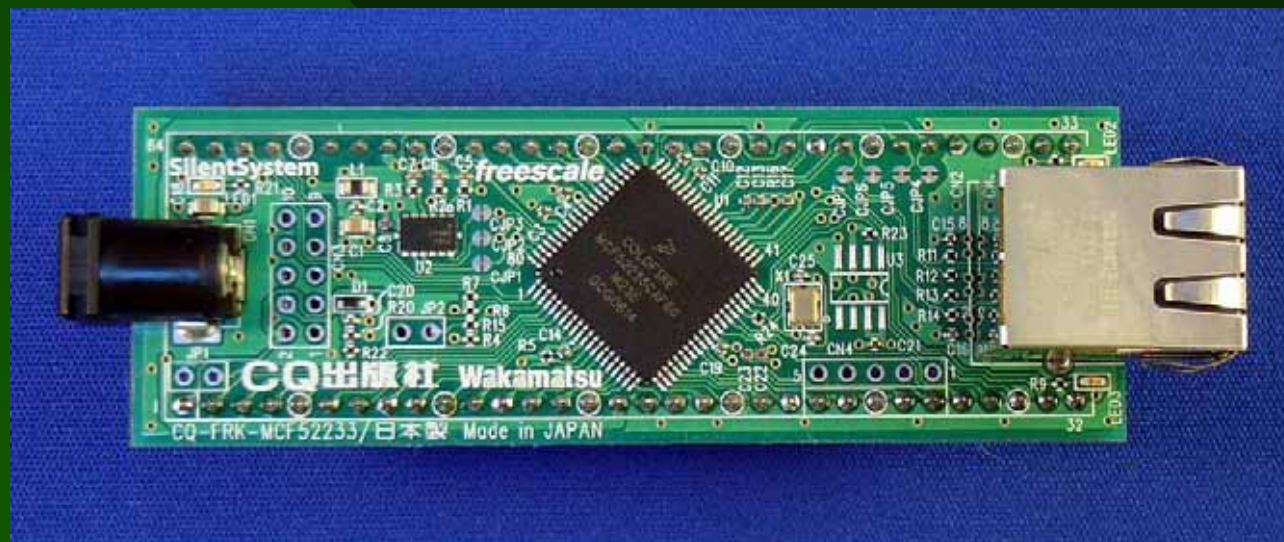
有限会社サイレントシステム エクゼクティブエンジニア

中本 伸一

SilentSystem

付属ColdFireマイコン基板とは？

- Freescale社のMCF52233を使用
- すぐにイーサネットに接続可能
- 開発環境としてのSilentCインタープリター
- gdbスタブによるオープンソース対応



付属ColdFireマイコン基板の応用例

- イーサネット経由によるI/Oポートのコントロール
- イーサネット経由でのA/Dデータ転送
- シリアルとイーサネットの相互変換
- インターネットラジオサーバーとレシーバー
- 広域振動センサーネットワーク
- JPEGカメラ画像サーバー
- メール送受信モジュール
- サーバー監視モジュール
- ネットワークプログラミング学習教材として

ColdFireに搭載されているデバイス

- シリアル 2ch独立に利用可能
- SPI SDカードの読み書き
- IIC センサーやI/Oポートを接続
- A/D 電圧を測定 音声入力
- PWM D/A変換 音声出力
- RTC 周期的なイベント発生
- すべてSilentCからコントロール可能

C言語インタープリター SilentC

- 会話型インタープリタ言語
- ダイレクト実行可能
- ソケットによるTCP/IPプログラムを開発可能
- ユーザードライバでデバイスをコントロール
- HTTPサーバーとCGIが利用可能
- tftpによるファイル転送
- セルフアップデートが可能

SilentCによるTCP/IPプログラミング例 (サーバー側)

```
main() // SilentCはmainから実行を開始します
{ // 簡単なサーバーを書いて見ましょう
char *recv_buf; char listen,soc; // 変数の宣言
    recv_buf = MemoryAlloc(256); // データを格納するバッファを確保
    listen = CreateSocket(1); // TCP/IP待ち受けソケットを作成
    Bind(listen, 30049, 1); // ポート番号30049を監視するように指定
    GetNetLine(0,0,0,0); // TCP/IPの1行入力ライブラリ初期化
    soc = Accept(listen, -1); // クライアントからの接続を待ち受ける
    GetNetLine(soc, recv_buf, 255, -1); // クライアントデータを1行入力する
    CloseSocket(soc); // ソケットを閉じる
    GetNetLine(0,0,0,1); // TCP/IPの1行入力ライブラリの終了処理
    PrStr(recv_buf); // クライアントから送られた1行を表示
    MemoryFree(recv_buf); // 入力バッファを開放して終了
}
```

SilentCによるTCP/IPプログラミング例 (クライアント側)

```
main()                                // SilentCはmainから実行を開始します
{                                       // 簡単なクライアントを書いて見ましょう
char soc;long ip;                       // 変数の宣言
    soc = CreateSocket(1);              // TCP/IP送信ソケットを作成
    ip = GetIP("192.168.1.10");         // サーバーのIPアドレスを作成
    Connect(soc, ip, 30049);            // サーバーのポート30049に接続要求
    Write(soc, "1234¥r", 5);            // データを1行送信する
    WaitWriteComplete(soc);             // サーバーがデータを受け取るまで待つ
    CloseSocket(soc);                   // ソケットを閉じる
}
```

SilentCによるTCP/IPプログラミング例 (実行結果)

```
main(){char *recv_buf; char listen,soc;recv_buf = MemoryAlloc(256);  
listen = CreateSocket(1);Bind(listen, 30049, 1);GetNetLine(0,0,0,0);  
soc = Accept(listen, -1);GetNetLine(soc, recv_buf, 255, -1);  
CloseSocket(soc);GetNetLine(0,0,0,1);PrStr(recv_buf);MemoryFree(recv_buf);}
```

```
main(){char soc;long ip;soc=CreateSocket(1);ip=GetIP("192.168.1.10");  
Connect(soc,ip,30049);Write(soc,"1234¥r",5);WaitWriteComplete(soc);  
CloseSocket(soc);}
```

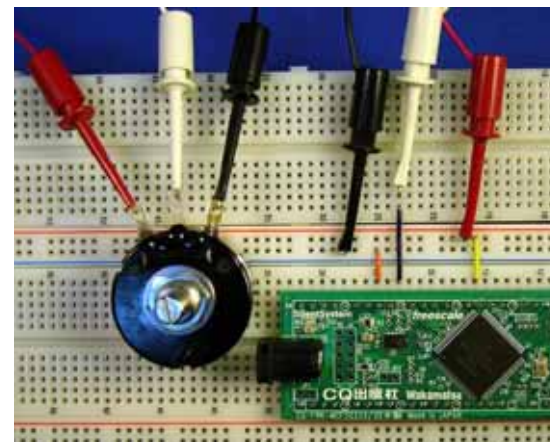
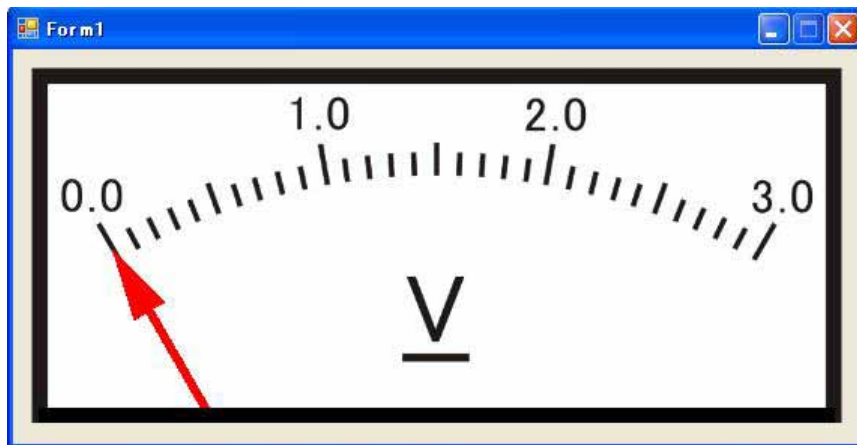


```
Tera Term - 192.168.1.10 VT  
File Edit Setup Control Window Help  
SilentC (Oct 31 2008)  
OK  
run  
1234OK
```


UDPによるリモートA/D変換の例 (イーサネット電圧計)

```
main(){char soc,*buf;long ip=GetIP("192.168.1.2");//変数宣言と送信先のIPを得る
buf=MemoryAlloc(10);soc=CreateSocket(0);InitAd(0x70);//UDPソケット作成と初期化
for(;;){
    GetDigit(GetAd(4),buf);
    SendTo(soc,ip,30049,buf,StrLen(buf));
    if(Getc(0)=='q')break;}
CloseSocket(soc);MemoryFree(buf);}
```

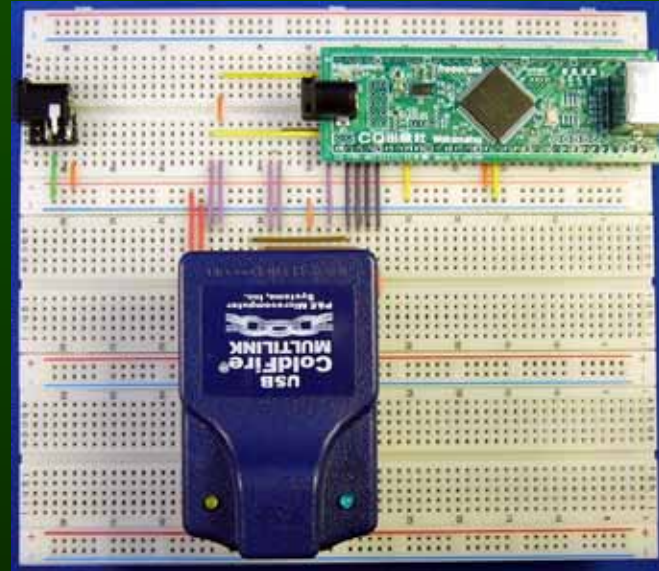
// A/D変換した値を文字列に変換
//UDPでパソコンに送信
//qキーが押されたらループを抜ける
//ソケットを閉じメモリを開放して終了



本格的なソフト開発について

- SilentCは手軽に開発できるが欠点もある
 - インタープリターなので実行速度が遅い
 - 割り込みを利用できない
- 無料でダウンロードできるCodeWarriorを利用すればネイティブなCプログラムが開発できる
- デバッグ用のモジュール(BDM)でフラッシュ書き込みとハードウェア・ブレークが利用可能

本格的なソフト開発について



- OS無しではネットワークを利用したアプリケーションの開発は難しい
- TCP/IPは自分の責任で用意しなければならない

ユーザードライバについて

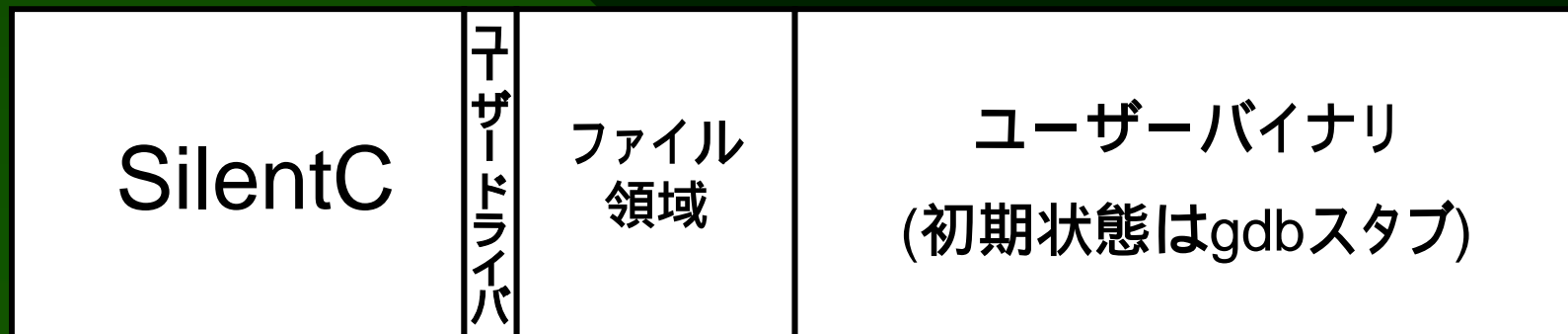
- SilentCは手軽に開発できるが欠点もある
 - インタープリターなので実行速度は遅い
 - 割り込みを利用できない
- ユーザードライバを利用すれば一挙に解決できる
- CodeWarriorで開発したプログラムをユーザードライバとして実行すると便利
- SilentCのファイル転送機能を利用することでデバッグ用のハードウェア(BDM)が不要に

CodeWarriorとSilentCの共存

- ネットワーク部分はSilentCを利用
 - ネットワーク部分のリアルタイム性は低い
- デバイス割り込みやタイムクリティカルな部分はユーザードライバを利用する
- SilentCはローダー、デバッグ出力、ユーザーインターフェースとして利用
- CodeWarriorのコードサイズ制限は128Kバイト
- メモリの前半にSilentC,後半はユーザードライバ

ユーザードライバのインストール方法

- SilentCのtftpサーバーを利用してファイルを転送
- 予約されたファイル名によりフラッシュに書き込み
 - UserDriver.bin ユーザードライバ(4Kバイト)
 - FILEDATA.BIN ファイル領域(48Kバイト)
 - COLDFIRE.BIN ユーザーバイナリ(128Kバイト)
 - SILENTC.BIN SilentCのアップデート用



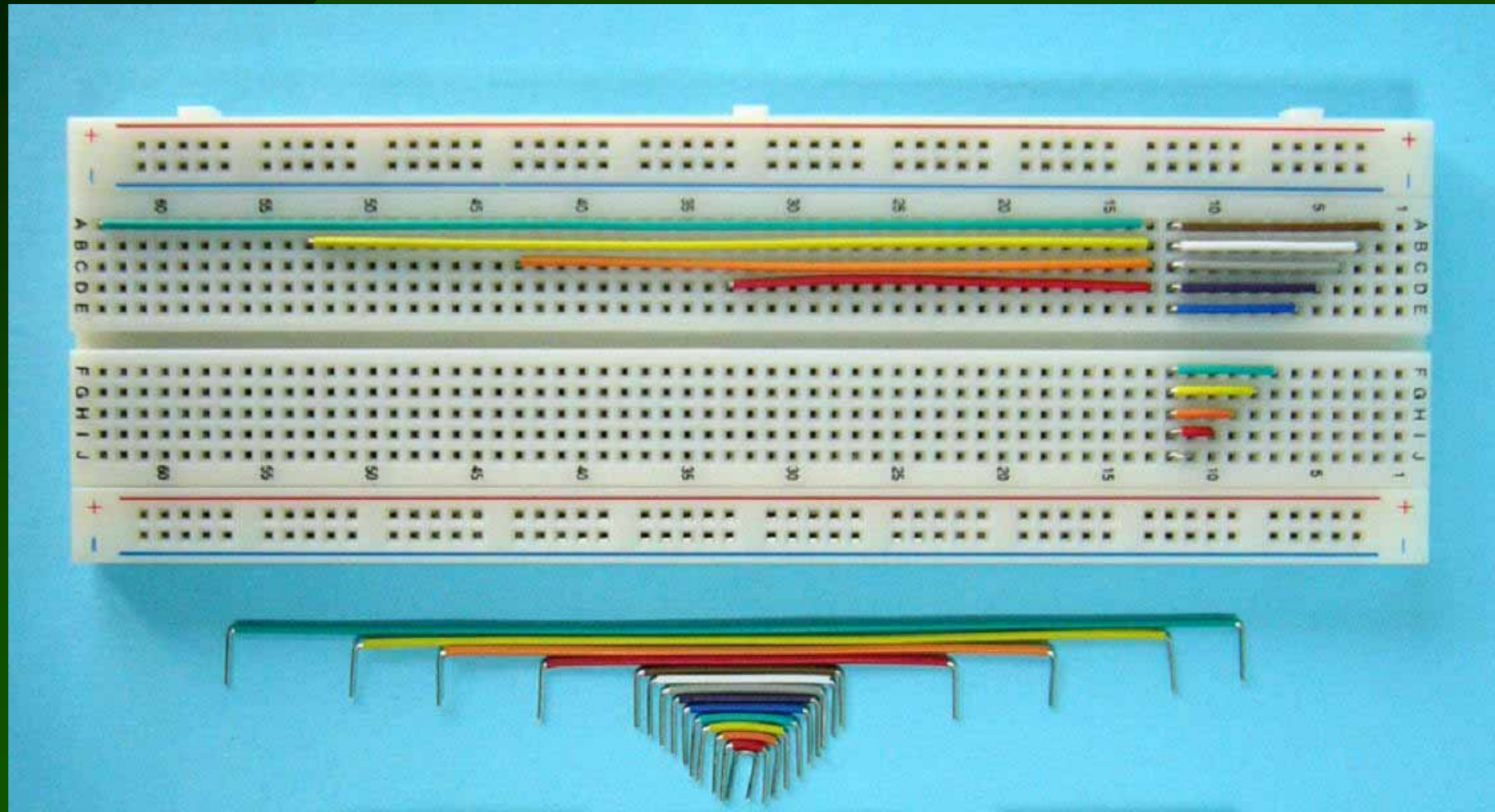
00000番地

14000番地

20000番地

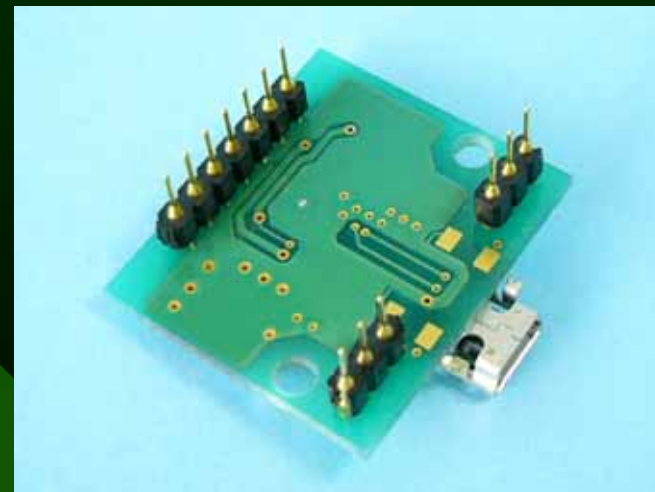
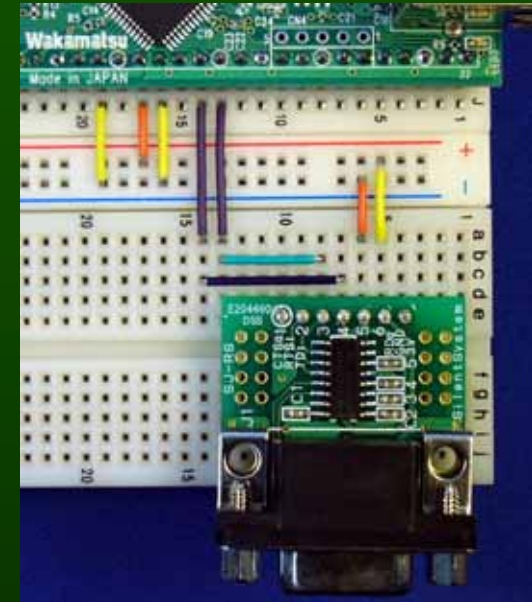
40000番地

ブレッドボードについて



ブレッドボードについて

- 試作に最適
- 回路変更も簡単
- デジカメで回路図を保存可能
- ブレッドボード専用モジュールを活用すると便利

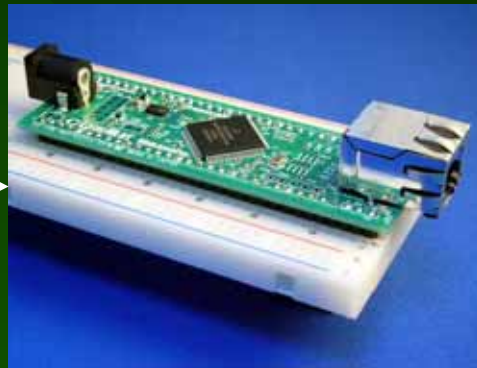


ネットワークシリアルケーブルについて

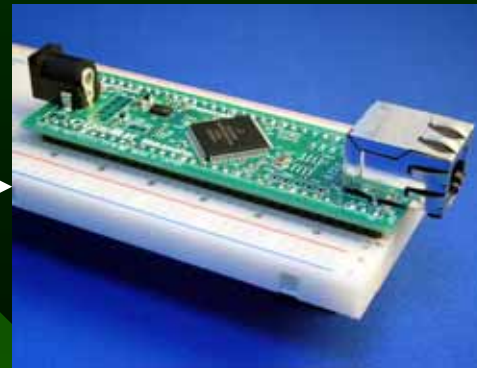
- イーサネットでシリアルケーブルエミュレーション
- ケーブルとして動作するのでソフト開発不要
- 2009年3月号で詳しく解説します



シリアルコネクタ
(オス)



イーサネットで接続(インターネット経由)



シリアルコネクタ
(メス)

質疑・応答

ありがとうございました

有限会社サイレントシステム エクゼクティブエンジニア

中本 伸一

SilentSystem