電子工作
Hi-Tech
シリーズ

見本

おまかせ表示！ 動画から天気予報までなんでもウェブから

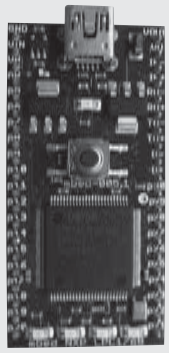
ハイパー・マイコン mbedで インターネット 電子工作

飯田忠夫[著]
Tadao Iida

実験や
研究開発に
ピッタリ！

分厚い
マニュアルは
見なくて
OK！

CQ出版社



mbedを使って組み込みマイコンのプログラム開発をマスタしよう!

本章では，最初に mbed について紹介し，続いてプログラム開発を行うための準備として，mbed の Web ページにログインするためのアカウント作成やデバッグのための環境構築を行います．最後に実際にプログラムを作成し，実行してみます．

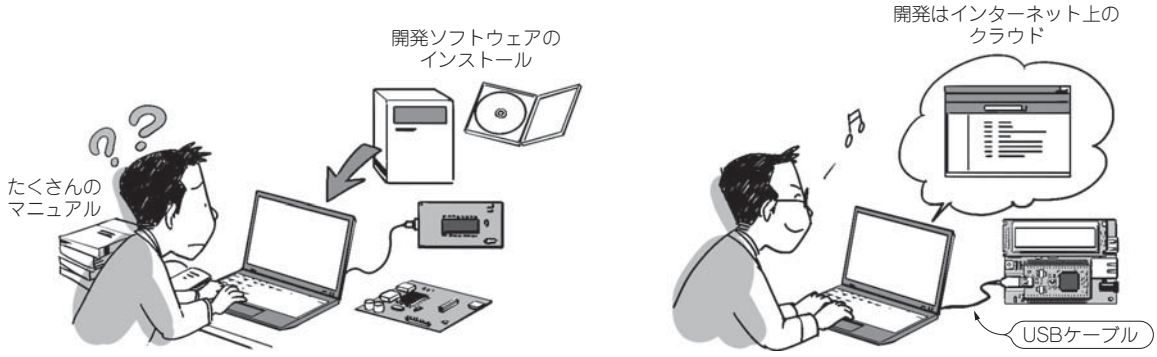
本章を読み終えると，開発手順をひととおり体験でき，mbed を使ったプログラム開発の流れをつかむことができます．

1-1 mbed はただのマイコンではない

mbed は ARM 社の CPU を搭載したマイコンです．しかし，これまでのマイコンとは少し違い，マイコン開発の初心者でも早くしかも簡単に「動くもの」が作れるように開発されました．そのため，これまでのマイコンとは違ったいくつかの特徴があります．

一つ目は，mbed の開発を行うための開発環境がクラウド（後述）で提供されているため，パソコンに開発用ソフトウェアをインストールするなどの面倒な作業が必要がありません．しかも，開発環境は Web ブラウザから利用でき，インターネットが利用できる環境さえ整っていれば，mbed での開発をすぐに始めることができます（**図 1-1**）．

二つ目は，コンパイラが出力するバイナリ・ファイル（実行ファイル）をマイコンに書き込む際にも，USB ストレージ(*1)として認識される mbed に実行（バイナリ）ファイルを「保存するだけ」という手軽さ



（a）一般的な開発　　（b）mbedでの開発

図 1-1　従来の複雑なマイコン開発と単純な mbed を使った開発

従来のマイコン開発では，開発用ソフトウェアをインストールしマイコンを開発するための環境を構築する必要がある．また，システムによっては書き込み用のハードウェアも別途必要になる．このため，初心者の場合プログラムを作成する前の段階でかなりの時間を費やすことも多く，マイコンでプログラムを作成するための敷居が高い．一方 mbed は，クラウドにある開発環境を使うため，開発環境の構築に要する時間はほとんどかからない（アカウント作成だけ）．しかも，マイコンへのプログラムの書き込みは，ファイルを mbed に保存するだけという手軽さである．

です(**図 1-2**)．したがって，マイコンの開発では必須だった書き込み器や書き込むためのソフトウェアなどは一切必要ありません．パソコンと mbed，そしてそれらをつなぐ USB ケーブルさえあれば，すぐに開発したプログラムを実行できます．

三つ目が，世界中にいる何万という mbed ユーザが開発した，センサや各種デバイスを使用するためのライブラリやプログラムが自由に利用できることです．これらは，多くの有用な情報とともに ARM 社が運営している mbed の公式 Web ページ(`http://mbed.org`)に公開されていて，しかも，その多くがオープン・ソースとして公開されています．これにより，ライブラリを使用したり，公開されているプログラムをベースにして開発することで，開発者は自分で作成するプログラムのソース・コードの量を減らすことができ，それにともない，デバッグにかかるコストも大幅に削減することができます．

このように，mbed はこれらの機能により驚くほど容易にしかも素早く「動くもの」を完成させることができるのです．

◆ Cortex M シリーズのマイコンを搭載

mbed で使われている CPU は，Cortex M3 もしくは Cortex M0 という ARM 社の 32 ビット組み込み用プロセッサです．**図 1-3** は最初に発売された mbed で，NXP 社の LPC1768(Cortex M3)が搭載された，汎用性の高い高性能マイコンです．ARM プロセッサは特に消費電力が少ないという特徴があり，モバイル機器や組み込み機器の用途に世界中で使われていて，mbed に搭載されている Cortex M3 の上位シリー

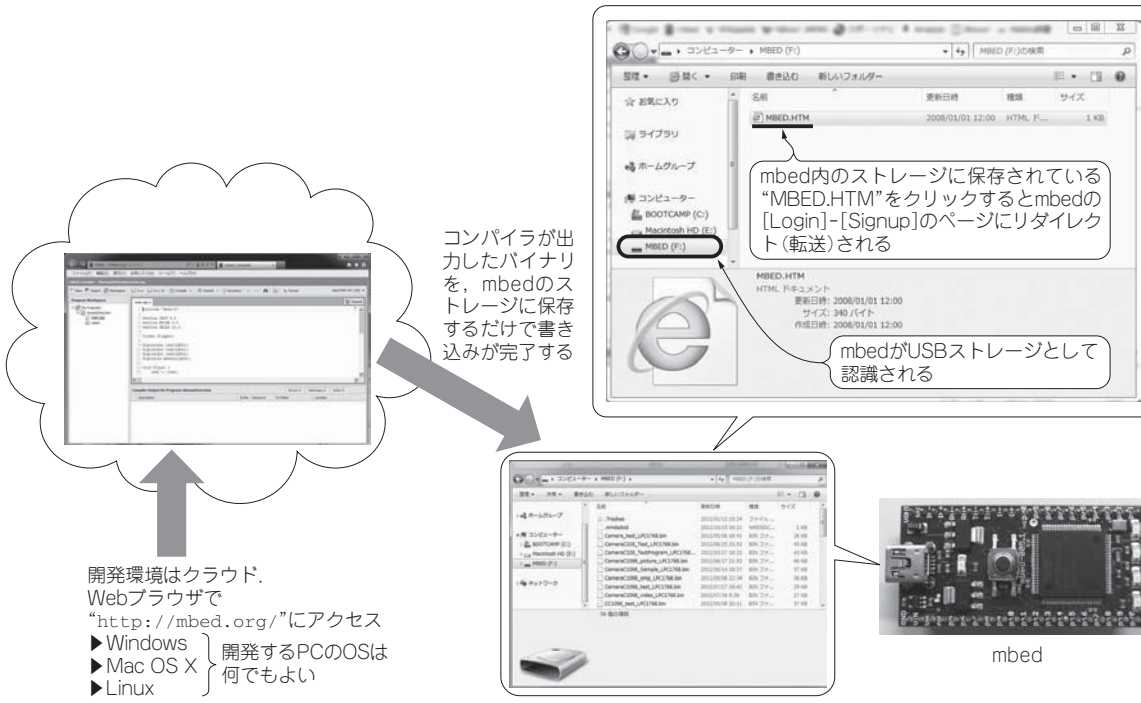


図 1-2　開発したプログラムは実行ファイルを mbed にコピーするだけという手軽さ!
mbed での開発はクラウドにアクセスして行うため，開発に使用するコンピュータの OS には依存しない．Web ブラウザで http://mbed.org/ にアクセスするだけ．作成したプログラムをコンパイルしてできたバイナリ・ファイルを mbed のストレージに保存するだけでプログラムを実行できる．

(＊1) USB ストレージ：ストレージとは補助記憶装置のことで，USB を介して補助記憶装置を使用する機器を USB ストレージという．普段使っている，USB メモリや USB の外付ハードディスクも USB ストレージ．mbed はパソコンに接続すると，2M バイトの USB ストレージとして認識される．

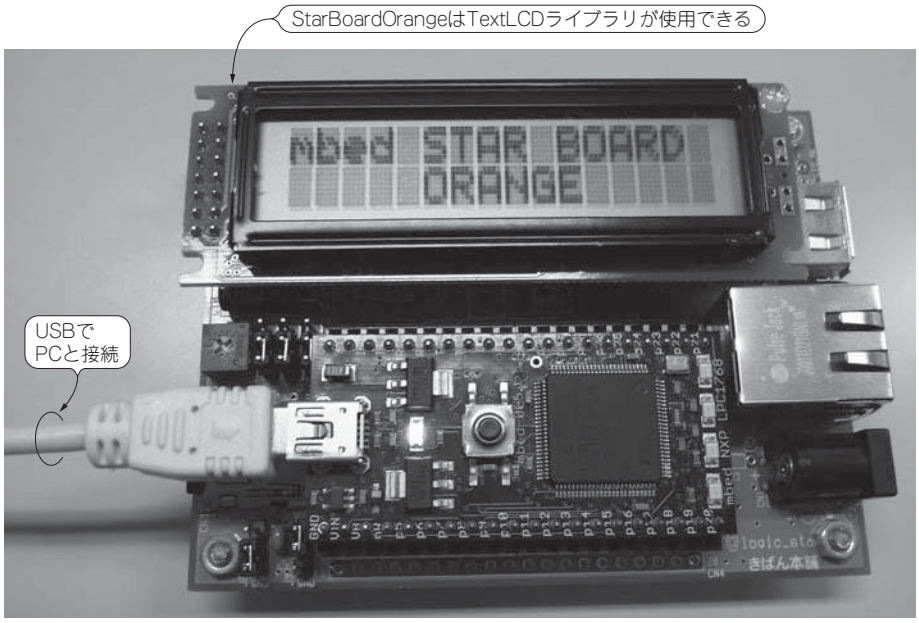


図 2-7　キャラクタ LCD 表示プログラムの実行結果
TextLCD ライブラリをプログラムにインポートし，実際に**リスト 2-1** のプログラムを実行した結果．プログラムでは上段に「mbed STAR BOARD」，下段の中央に「ORANGE」と表示している．

うか．確かに少し前までは，初心者が手を出すにはかなりハードルが高かったのですが，mbed であればライブラリを活用することで簡単にネットワークを利用したプログラムを作成できます．

ここでは，第 1 章で作成した「人検知システム」をさらに発展させ，人を検知した情報をメールで送信する「帰宅お知らせシステム」を作成します．

● 帰宅お知らせシステムの作成

回路は**図 2-8** のように第 1 章で作成した「人検知システム」の回路(**図 1-13**)をそのまま流用していますが，新しく mbed を単体で動作させるための電源としてニッケル水素電池の eneloop 4 本と，ネットワークを利用するための有線LANを追加しています．**図2-9**は「帰宅お知らせシステム」の全体イメージでメールの流れを説明しました．

例えば，玄関や廊下にこのシステムを設置し，センサが人を検知するとその日付と時刻を記録し一定時刻ごとにその情報をメールで送信します．プログラムでは，メール・サーバに Yahoo の SMTP(*3)サーバ(`smtp.mail.yahoo.co.jp`)を利用しましたが，通常は読者の方が加入しているプロバイダのメール・サーバを利用するのがよいでしょう．メール・システムの連携のイメージを**図 2-10** に示します．私たちが普段何気なく使っているメールはたくさんのシステムが連携して動作しています．これにより，Gmail や Hotmail，スマートホンやプロバイダのメールはそれぞれ違うシステムで動作していますが，互いにメールの送受信を行うことができます．

製作したシステムを玄関や廊下に設置する場合は，有線 LAN が利用できないことも多いと思います．

(*3) SMTP：Simple Mail Transfer Protocol．インターネットのメール・サーバが使うプロトコル．基本，テキスト・ベースでやり取りする．

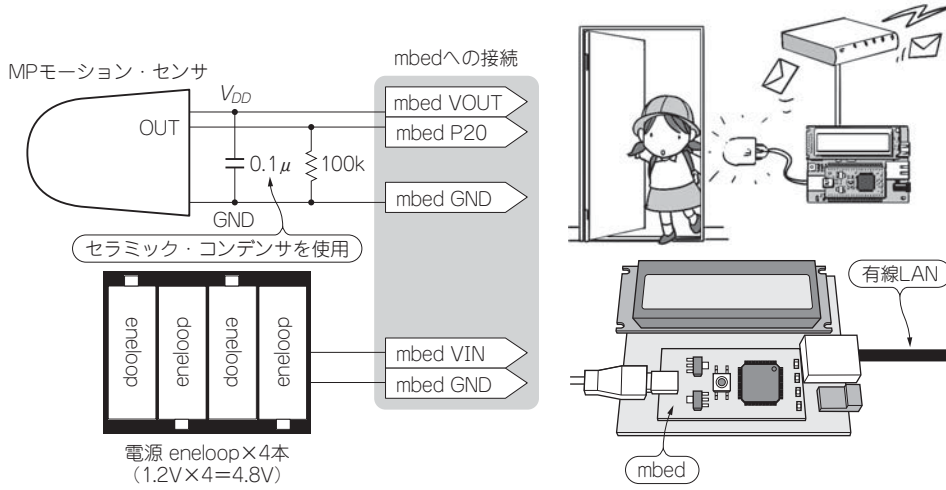


図 2-8　帰宅お知らせシステムの回路図

人検知システムで使用した回路をそのまま流用している．帰宅お知らせシステムは，StarBoardOrange 単体で動作するため電源をパソコンの USB 給電から eneloop 4 本に変更したところが異なる．また，検知した結果を Email を使って知らせるので，StarBoardOrange に有線 LAN を接続した．
eneloop 4 本だと約 8 時間動作するが，それ以上安定して動作させたい場合は，StarBoardOrange に AC－DC アダプタを利用するとよい．MP モーション・センサの V_{DD} と GND を間違えると壊れてしまうので，配線はしっかり確認する．

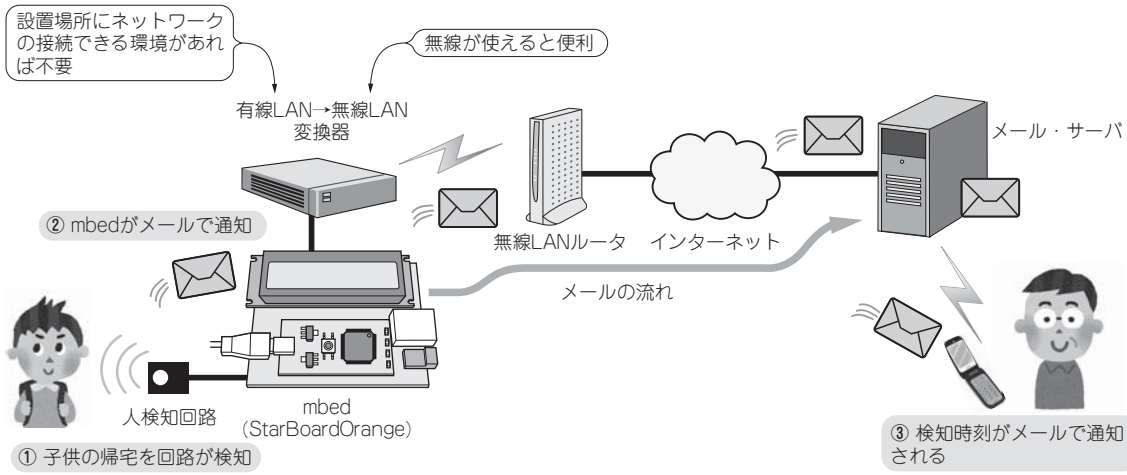


図 2-9　帰宅お知らせシステムのメールの流れ

図では人検知回路が男の子を検知し，mbed がメール・サーバに検知した結果を送信する．そして，男性の携帯電話に帰宅お知らせシステムからの通知メールが送付されている．帰宅お知らせシステムを設置する玄関や廊下には有線 LAN のポートがないと思われる．そこで，有線 LAN と無線 LAN の変換器を利用して自宅の無線 LAN ルータに接続している．

そんなときは，無線 LAN ⇆有線 LAN のコンバータ（例えば WLAE-AG300N Buffalo 製など）を使って mbed をネットワークに接続することができます．その機器の本来の用途は，液晶テレビやビデオ・レコーダに付いている有線 LAN を無線化するためのものです．**図 2-11** は，筆者の自宅の廊下に「帰宅お知らせシステム」を設置して動作させたものです．

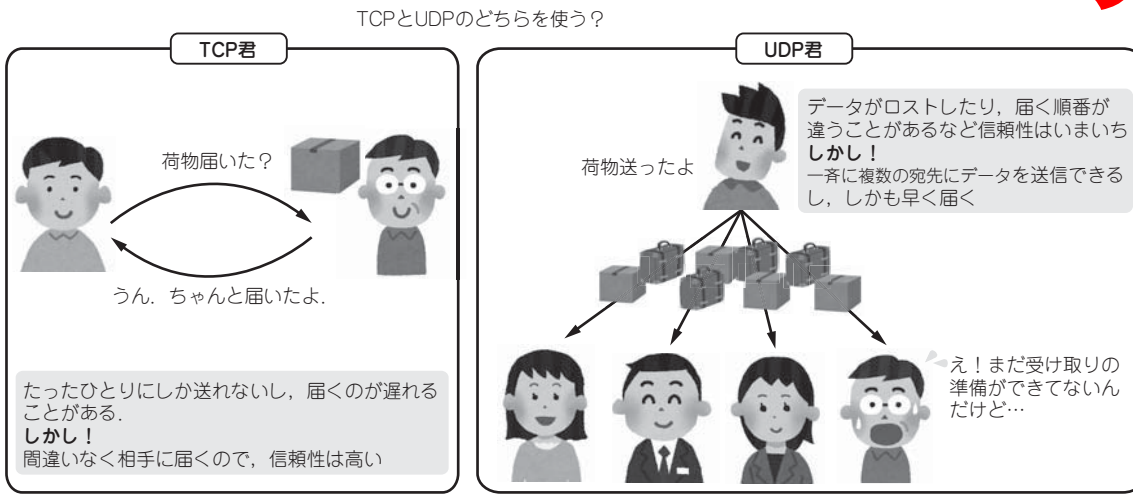


図 3-4　TCP と UDP の特徴
作成するプログラムによって TCP と UDP を使い分けることで，パフォーマンスのよい通信が実現できる．

が送信元に対してデータの再送を要求することで，信頼性の高い通信をサポートしています．ネットワークを利用したプログラムのほとんどが TCP を使っていますが，UDP に比べるとヘッダ・サイズが大きく通信制御も複雑なためデータを送信する効率は UDP に比べると劣ります．また，これらの理由からデータの到着に遅延が生じる場合があります．

◆ UDP の特徴

TCP とは逆で，途中でデータが紛失したりデータの届く順番が違っても UDP は何もしないため，プログラム側で対応する必要があります．ただし，TCP に比べヘッダ・サイズが小さく，データを一方的に相手側に送信するだけなので，TCP に比べると効率的にデータを送受信することができます．また，フロー制御や再送要求などの複雑な制御をしないため，信頼性は高くありませんが，その代わりデータの遅延は発生しにくいという特徴があり，音声データのようにデータの遅延や多少データをロストしても影響が小さいアプリケーションに利用されます．

ほかにも，TCP は 1 対 1 の通信（ユニキャスト）ですが，UDP は 1 対多の通信にも対応しており，ネットワーク内のすべての機器に一斉にデータを送付するブロードキャストや，ネットワーク内の一部の機器にだけデータを送付するマルチキャスト通信などにも利用されます．

3-2　TCPSocket を使ったデータ受信プログラムの作成

最初に TCP を使ったプログラムを作成します．このプログラムは**図 3-5** のように，パソコンで動作するプログラムから文字列を入力し送信すると，データがネットワーク経由で mbed に送信され，キャラクタ LCD にその文字列が表示されます．

通常 TCP を使った通信プログラムはどれも似たような処理の流れになり，これから作成するプログラムも**図 3-6** に示す流れで処理が行われます．

それでは，プログラムについて説明します．

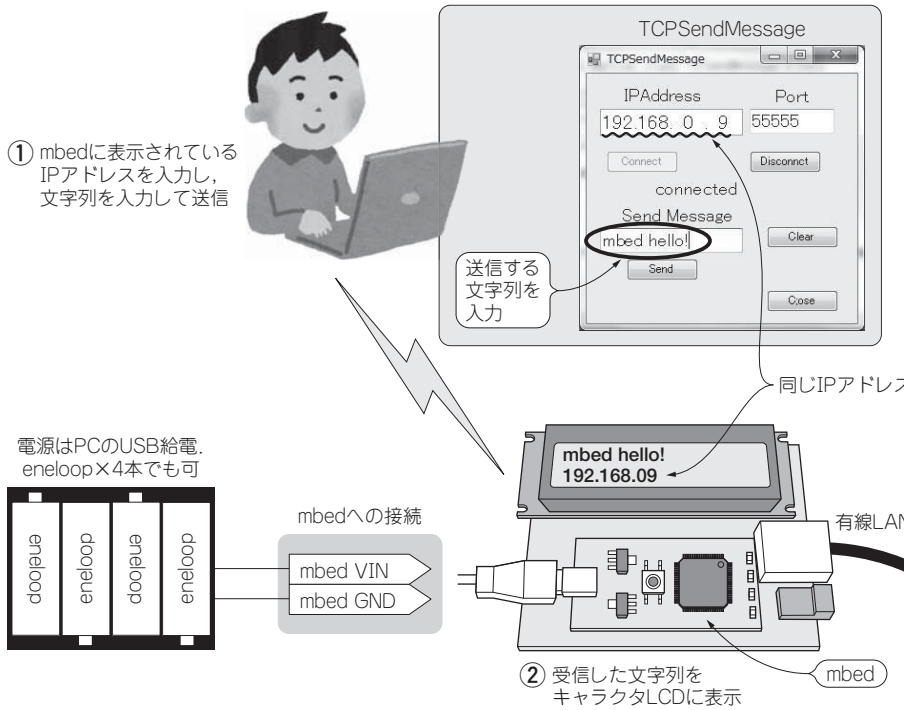


図 3-5　メッセージ送信プログラムのイメージ

TCPSocket を使って作成するメッセージを送信する．パソコンで動作するプログラム TCPSendMessage に mbed の IP アドレスとポート番号(55555)を設定し，[Connect]ボタンを押す．接続が完了したら，テキスト・ボックスに文字列を入力して，[Send]ボタンを押すと，mbed のキャラクタ LCD に送信した文字列が表示される．

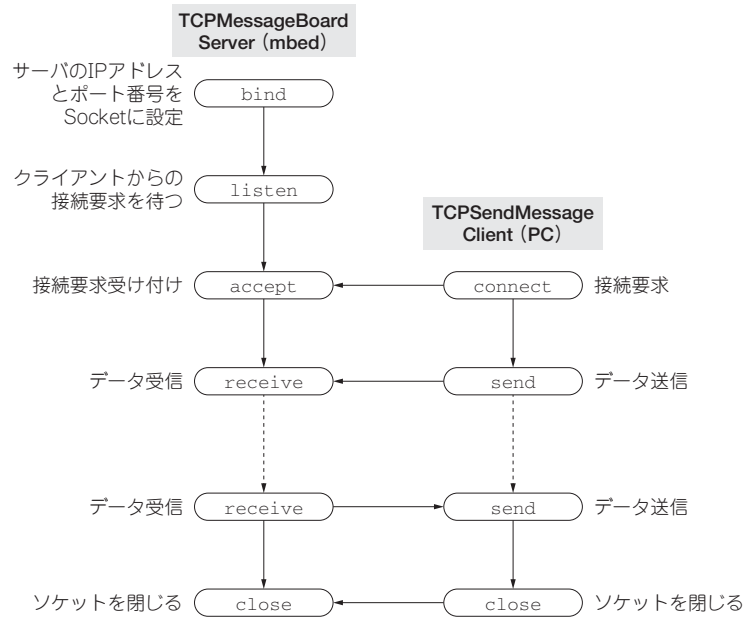


図 3-6
TCP プログラムの処理の流れ

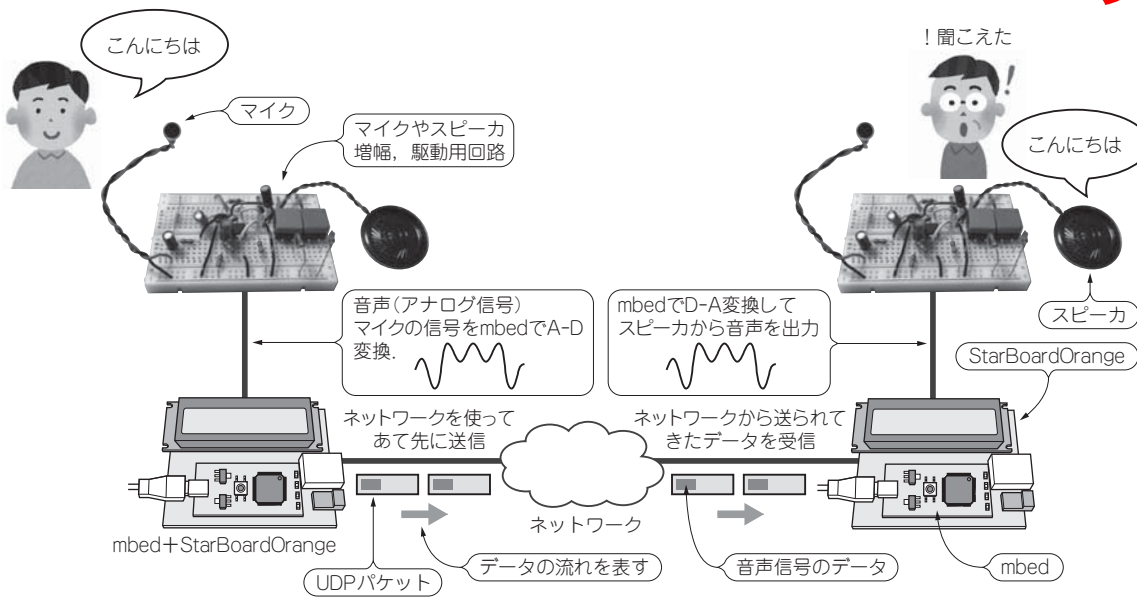


図 4-1　IP トランシーバの概要

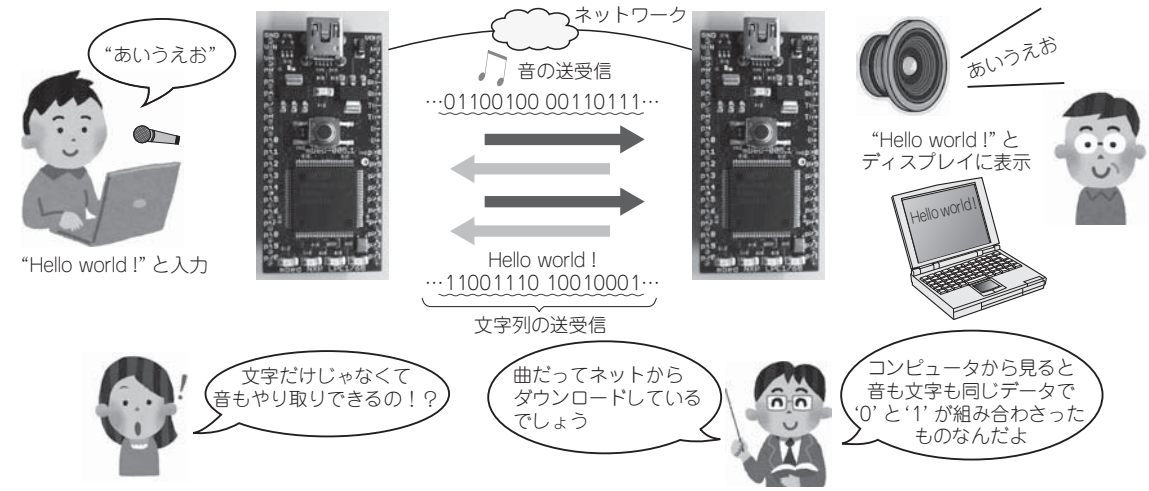


図 4-2　文字列も音もコンピュータから見ると ‘1’ と ‘0’ の組み合わせ

これまでは，主に文字列をネットワークで送受信していた．今回は音声をネットワークでやりとりする．「音をネットワークでやり取りする」と聞くとやけに難しく感じるが，コンピュータには文字も音のどちらも ‘1’ と ‘0’ の組み合わせにしか見えないので，大きな違いはない．

4-2　IP トランシーバ回路の製作

◆ 音声信号を mbed に入力する

音声を mbed に入力するにはどうしたらよいでしょうか？ mbed の Interface には Digital/Analog のほかに，UART や Ethernet など多くの I/O（Input/Output）が利用できますが，これらはすべて電圧を入力したり出力します．

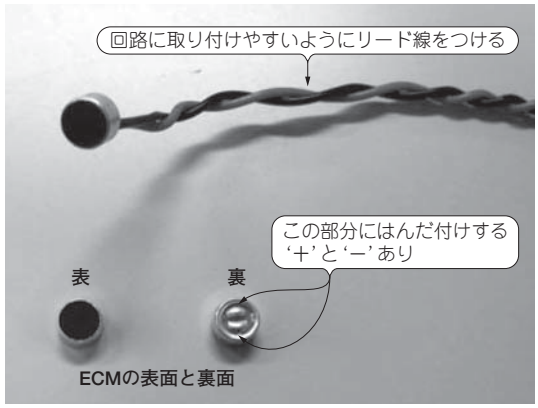


図 4-3　コンデンサ・マイク（ECM）の外観
使用したコンデンサ・マイクには端子が付いていないので，単線をはんだ付けして増幅回路に接続できるようにした．

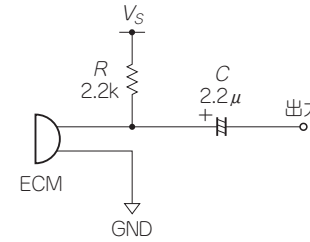


図 4-4　ECM 動作回路
コンデンサ・マイクを動作させるための回路．コンデンサ・マイクはそれ単体では動作しない．図のように電源を供給し直流成分をコンデンサで除去することで出力信号が得られる．

そこで，音声を電圧に変換するセンサを使うと mbed に入力することができそうです．具体的には皆さんも一度は使ったことがあるマイクロホン（以下マイク）を使います．マイクにはいくつか種類がありますが，電子工作では手軽に利用でき，かつ安価であることから，ECM（Electret Condenser Microphone）がよく利用されます．今回使用する ECM には回路に配線するためのリード線が付いていないので，**図 4-3** のように取り付けました．

ECM をマイクとして動作させるためには，**図 4-4** のように電源と抵抗，コンデンサを使った簡単なマイクを動作させるための回路が必要になります．この動作回路を製作し ECM から「あー」という音声を入力したときの出力をオシロスコープで観測したものが**図 4-5** です．マイクからの出力信号が小さいため，「あ」の音声の波形がはっきりわかるようにマイクに近づいて大きな声を入力しました．そのため，本来であればマイクの出力は十数 $mV_{p\text{-}p}$ 程度ともっと小さな信号ですが，$140mV_{p\text{-}p}$ と少し出力波形が大きくなっています．ちなみに mV の単位の横に付いている p-p とは「ピーク to ピーク」のことで，信号の最大値（一番大きな値）からもう一方の最大値（一番小さな値）までの幅を表しています．

mbed の AnalogIn（AI）の入力は 0 ～ 3.3V の範囲で電圧を読み取るため，入力電圧が十数 mV では少し小さすぎます．そこで，ECM の出力信号を増幅回路で 1 ～ $2V_{p\text{-}p}$ 程度に増幅した後，mbed に入力します．

● ECM の信号を増幅する

信号を増幅するには一般的にトランジスタや OP アンプを使いますが，ここでは簡単に増幅回路が設計・製作できる OP アンプを使用します．OP アンプは用途によって多くの種類がありますが，ここでは，以下のような理由により LM358 を使用しました．

① mbed の VOUT 端子から OP アンプの電源が供給できる［単電源・低電圧（単電源 3.0V から）］
② 小型（DIP 8 ピン）で OP アンプが 2 個内蔵されている（回路をコンパクトに製作できる）
③ 安価で汎用的な用途に利用できる（お財布にやさしく再利用しやすい）

一つの LM358 のパッケージには OP アンプが 2 個内蔵されています．そこで，一つは ECM からの入

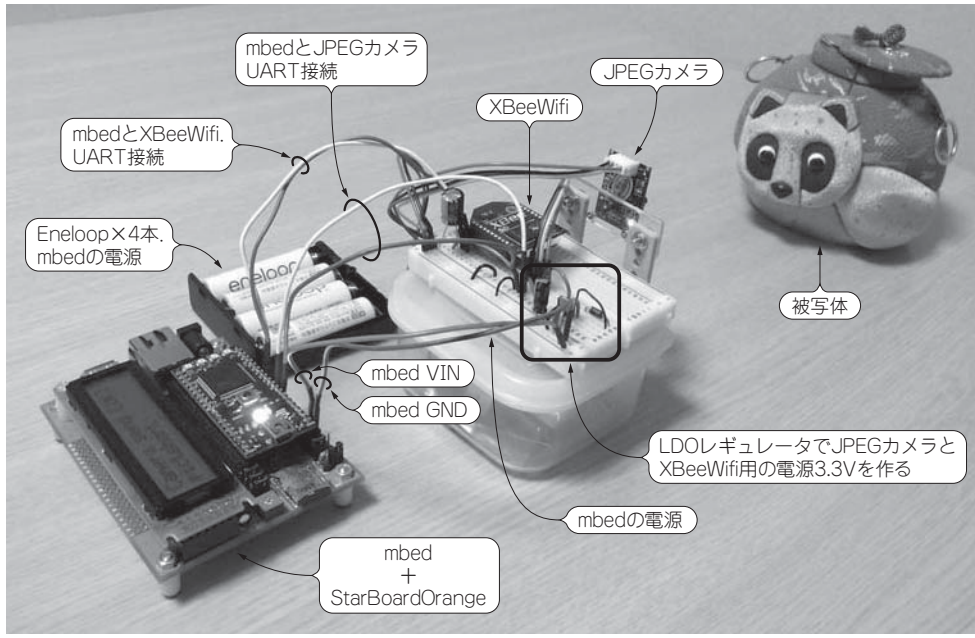


図 6-18　画像表示システムの接続例

図 6-17 は製作する回路図で，Eneloop×4 本から LDO レギュレータを使って 3.3V の電圧を作っていますが，これは XBeeWifi と JPEG カメラの電源になります．また，mbed の電源は直接 Eneloop から供給しています．

図 6-18 は実際に製作した回路です．

6-3　mbed のプログラム Camera-XBeeWifi の作成

mbed のプログラムを作成していきます．プログラム名は `Camera-XBeeWifi`（**リスト 6-1**）になります．まず，以下のライブラリを import してください．

```
CameraC1098
TextLCD
```

カメラのライブラリは以下の URL から import できます．

```
http://mbed.org/users/sunifu/code/CameraC1098/
```

プログラムの `main` 関数内の以下の部分で，JPEG カメラ，XBeeWifi の初期化処理を行っています．

カメラが取得する画像や通信速度は `init` 関数で，**表 6-3** のパラメータを与えることで変更できます．

カメラの画像サイズを大きくすると転送に時間がかかるため，画像を取得してから表示までにかなりの遅れが生じます．そこで，通信速度を速くすると当然画像は速く送信できますが，転送時にエラーが発生

リスト 6-1　画像表示システム Camera-XBeeWifi のプログラム

```
/**
 * CameraXBeeWifi program.
 *
 * CameraC328Library
 * Copyright (C) 2010 Shinichiro Nakamura (CuBeatSystems)
 * http://shinta.main.jp/
 *
 * CameraC1098-SS Library
 * Copyright (C) 2012 Tadao Iida
 */


#include "mbed.h"
#include "CameraC1098.h"
#include "TextLCD.h"

TextLCD lcd(p24, p26, p27, p28, p29, p30);

// JPEG カメラ・オブジェクト宣言
CameraC1098 camera(p9, p10);

// XBeeWifi 通信用オブジェクト宣言
Serial xbeewifi(p13,p14);


/**
 * A callback function for jpeg images.
 * You can block this function until saving the image datas.
 *
 * @param buf A pointer to the image buffer.
 * @param siz A size of the image buffer.
 */
// Callback 関数 CameraC1098 ライブラリの getJpegSnapshotPicture 関数から
// 呼び出される
void jpeg_callback(char *buf, size_t siz) {
    for (int i = 0; i < (int)siz; i++) {
            // XBeeWifi へ画像データを送信
        xbeewifi.putc(buf[i]) ;
    }
}

// JPEG カメラとの同期処理
void sync(void) {
    CameraC1098::ErrorNumber err = CameraC1098::NoError;

    err = camera.sync();
    lcd.locate(0,0);
    if (CameraC1098::NoError == err) {
        printf("[ OK ] : CameraC1098::sync¥r¥n");
        lcd.printf("Camera Sync [OK]");
    } else {
        printf("[FAIL] : CameraC1098::sync (Error=%02X)¥r¥n", (int)err);
        lcd.printf("Camera init [NG]");
    }
}

// JPEG カメラ画像取得関数
void test_jpeg_snapshot_picture(void) {
    CameraC1098::ErrorNumber err = CameraC1098::NoError;

    err = camera.getJpegSnapshotPicture(jpeg_callback);

    lcd.locate(0,0);
    if (CameraC1098::NoError == err) {
        printf("[ OK ] : CameraC1098::getJpegSnapshotPicture¥r¥n");
        lcd.printf("Camera send [OK]");
    } else {
        printf("[FAIL] : CameraC1098::getJpegSnapshotPicture (Error=%02X)¥r¥n", (int)err);
        lcd.printf("Camera send [NG]");
    }
}

int main() {

    wait(2.0) ;

    printf("¥r¥n");
```

ISBN978-4-7898-1890-2
C3055 ¥2400E
CQ出版社
定価：本体2,400円（税別）
9784789818902
1923055024007

ハイパー・マイコン mbedでインターネット電子工作