

DS80C400によるネットワーク型マルチメディアアプリケーションの構築

...DS80C400は、外付けの映像圧縮ハードウェアなしで、1秒あたり4フレームの白黒映像(240x180)を送信することができます。

ネットワーク型の低コストマイクロコントローラを使うことによって、拡声(PA)装置やネットワークドア、MP3プレーヤ、セキュリティカメラなどのマルチメディアアプリケーションを構築できます。このアーティクルでは、音声と映像を処理するシステムを例に、DS80C400ネットワーク型マイクロコントローラの使い方を解説します。

ネットワーク型PAシステムの構築

「ただいまより火災報知器のテストを行います」や「薬品倉庫責任者は、薬品倉庫までお越しくください」といったアナウンスに使用するPAシステム(ホールや屋外で使用する拡声装置)を想像してください。普通は、独立した配線とインフラを用い、トランジスタが登場する以前からある技術を使って作られているはずですが、このシステムをネットワークに移行したらどうなるでしょうか。オーディオケーブルが不要になるだけでなく、システム自体をインテリジェントにすることさえ可能になります。たとえば、呼び出しシステムをビルの入出場管理システムやネットワークサーバに接続すれば、呼び出す従業員がいる可能性の高い場所を知ることが可能になります。PAシステムをコンピュータ制御にすれば、オペレータの手を煩わせることなくアナウンスを繰り返すことができます。また、社内電子メールシステムと接続し、電子メールから音声アナウンスを行うシステムや、ウェブサイトから呼び出し依頼を入力するだけで自動的にアナウンスが行われるシステムも作ることができます。

ネットワーク型PAシステムは、どのようにしたら構築できるでしょうか。最低限、ウェブインタフェースと電子メールゲートウェイが走り、マイクなどが接続されたサーバが1台必要です。このサーバを「マスタコントロール」と呼びます。次に、複数のスピーカモジュールが必要です。スピーカモジュールは、ネットワーク型ユニットで、スピーカを駆動するデジタル・アナログ変換器(DAC)を持ちます。スピーカユニットは、低コストで設置が容易である必要があります。

図1に示す例は、2つのビルに7つのスピーカユニットを配したネットワークを1台のマスタコントロールサーバで制御するものです。ビル内ネットワーク間の接続には、ブリッジではなくルータを用います(後述しますが、この結果、ソフトウェアに大きな影響が出ます)。このアーティクルでは、DS80C400ネットワーク型マイクロコントローラによってスピーカユニットを駆動します。

マイクロコントローラの処理能力も搭載メモリ量も、最新PCには遠く及びませんが、PAシステムは帯域が狭ければ複雑な処理も不要だからです。22.05kHz x 8ビットでサンプリングされたモノラルオーディオ信号は、十分な音声品質を持ちますが、圧縮なしで180kbps以内に収まります。圧縮信号伸長用ハードウェアによるコストアップもありません。

図2に、ネットワーク型音声システムで必要とされる帯域を示します。実効帯域が5Mbpsしかない旧式の半二重10Mbネットワークであっても、音声アプリケーションが消費する帯域幅は4%以下にすぎません。現在使用されているイーサネットネットワークは、100Mbps以上がほとんどです。

スピーカハードウェア

スピーカモジュールでは、DS80C400以外に、メモリ(SRAM 512kBで十分)やネットワークPHY、

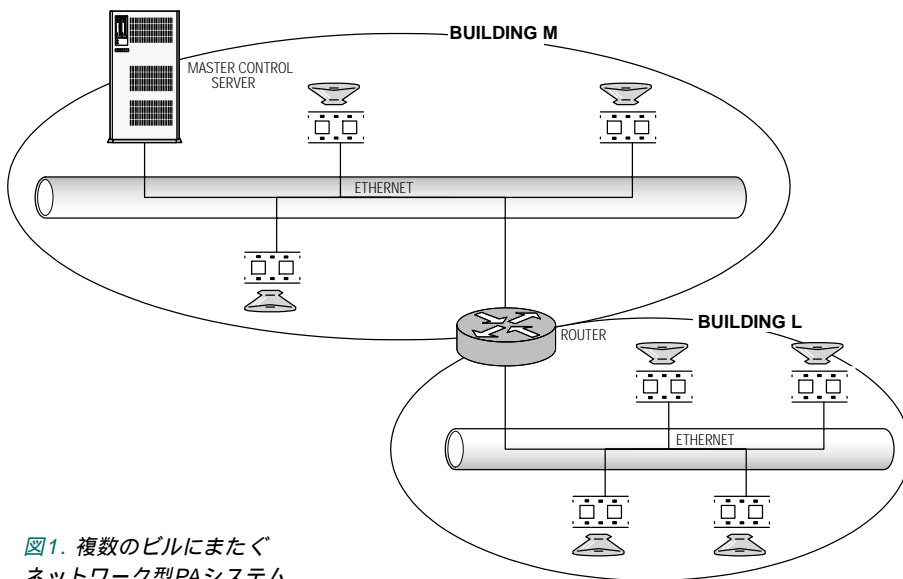


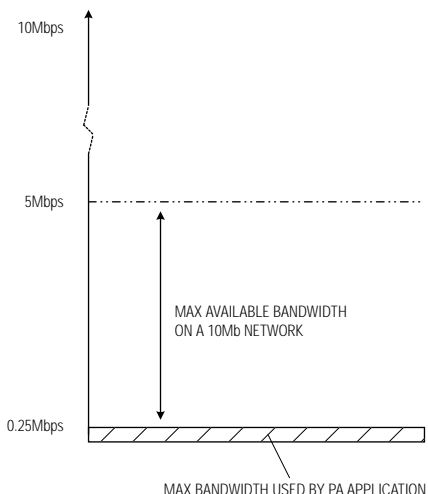
図1. 複数のビルにまたぐネットワーク型PAシステムも可能です。

DAC、アンプ、スピーカが必要です。詳しくは、アプリケーションノート609: *Internet Speaker with the DS80C400 Silicon Software* (英語版。www.maxim-ic.com/ja/appnoteindex)をご覧ください。

DS80C400は、以下の機能を持ち、設置やプログラミングが容易です。

- DS80C400 NetBoot(*DS80C400のNetworked Application Update*をご覧ください)。
- IP構成不要なDHCP機能により、設置コストと構成コストが削減できます。
- パワーオーバーサネット機能により(「パワーオーバーサネット」の項をご覧ください)、配線と材料費が削減できます。

電源が投入されると、DS80C400は、ROMに記録されたプログラムによりDHCP経由でIPアドレスを取得し、ネットワークにクエリを発行して最新のアプリケーションを探します。その後、見つけたアプリケーションが実行されると、音声データの受信準備が完了したことになります。スピーカモジュールの追加は、使われていないネットワークポートにケーブルを接続するだけで行えます。



DS80C400のNetworkedApplicationUpdate

DS80C400は、ネットワーク経由でブートすることが可能で、新しい装置へのプログラムロードが簡単に行えます。「NetBoot」機能は、シリアルローダのNコマンドで起動します。サポートしているメモリは、SRAMとフラッシュメモリです。

ネットワークブートは、新しい装置に対するプログラムのマニュアルロードを自動的に行ってくれるだけではありません。フラッシュメモリなどの不揮発性メモリを持たない設計では、電源が落ちた後、復旧するために必要なアプリケーションのインストールを最も簡単に行えるのがNetBootです。また、自動的にネットワークをチェックし、新しいプログラムが見つかったらプログラムのアップデートも行います(ネットワーク接続ができないときには、「古い」プログラムを実行します)。プログラムデータは、DHCPやTFTPといったネットワークプロトコルでIPアドレスを取得し、ロードします。また、ネットワーク構成を1-Wireデバイスに静的に設定・保存することもできます。IPv6方式のアドレスもサポートしています。

TINI400評価モジュール/ソケットで自動ネットワークコードアップデートを使うためには、以下のステップを実行します(サポートされている1-Wireデバイスなどの詳細情報は、www.maxim-ic.com/ja/microcontrollersにある "*High-Speed Microcontroller User's Guide: DS80C400 Supplement*" (英語版)をご覧ください)。

- 1) ネットワーク上にDHCPサーバがあることを確認します。DHCPサーバがない場合には、IPアドレスを静的に割り当て、そのアドレスを1-Wireデバイスに記録します。
- 2) TFTPサーバ(tftpd)をインストールし、そのIPアドレスをDHCP経由で公開するか、1-Wireデバイスに記録します。
- 3) *tbin2*フォーマットのコードを、TINI400というファイル名でTFTPサーバにアップロードします。
- 4) TINI400のNetBootジャンパをセットします。
- 5) システムがリセットされるたびに、自動的に、コードがTFTPサーバからリロードされるかアップデートされます。

図2. イーサネットネットワークで利用できる帯域幅よりも大幅に狭い帯域しか、ネットワーク型音声システムは必要としません。

スピーカソフトウェア

ハードウェアの設置を容易にした分、ソフトウェアに負担がかかります。ビル間にルータがあるため(図1)、ブロードキャストメッセージは、こちらのビルから向こうのビルまで到達できません。つまり、シンプルなブロードキャストメッセージは使えないのです。新しく設置されたスピーカシステムは、マスタコントロールがスピーカの位置とパラメータを確認するまで、マルチキャストメッセージを送出しなければなりません(「マルチキャスト」の項をご覧ください)。しかし、新しく設置されたスピーカシステムはマスタコントロールの位置を事前に知ることができないので、マスタコントロールに位置を知らせよう依頼するマルチキャストメッセージを送る必要があります。セキュリティを確保しなければならない場合には、このメッセージ

交換にデジタル署名を添付し、悪意のあるシステムがサーバとして応えられないようにします。ユニキャストメッセージを交換して構成が完了すれば、スピーカはマルチキャストグループの一部となり、オーディオパケットを待ちます。オーディオパケットは、マスタコントロールからマルチキャストされます。C言語で書かれたネットワーク型PAシステムのプログラム例があります(*DS80C400用ソフトウェアの開発をご覧ください*)。以下のプログラムは、ネットワークからマルチキャストソケットで音声信号を受け取るためのものです。

```
#define MULTICAST_IP_MSB 239
#define MULTICAST_IP_2 192
#define MULTICAST_IP_3 0
#define MULTICAST_IP_LSB 22
#define MULTICAST_PORT 6789

int s; /* socket handle */
struct sockaddr address; /* IP address */
unsigned char xdata buffer[1600];

/* Step 1: Create the socket */
s = socket(PF_INET, SOCK_DGRAM, 0);
/* Step 2: Join the multicast group */
memset(address, '\0', sizeof(address));
address.sin_addr[12] = MULTICAST_IP_MSB;
address.sin_addr[13] = MULTICAST_IP_2;
address.sin_addr[14] = MULTICAST_IP_3;
address.sin_addr[15] = MULTICAST_IP_LSB;
address.sin_port = MULTICAST_PORT;
join(s, &address, sizeof(struct sockaddr));

/* Step 3: Listen for incoming packets */
memset(address, '\0', sizeof(address));
address.sin_port = MULTICAST_PORT;
bind(socket_handle, &address, sizeof(struct sockaddr));

/* Step 4: Now we're ready to receive data */
recvfrom(s, buffer, 1600, 0, &address, sizeof(struct sockaddr));
/* We actually received data! We could play it. */
printf("Data received:\r\n");
...
```

TCP/IPネットワーク用プログラムを書いたことがあれば、join()関数以外は見覚えがあるはず。関数の呼び出し方は、PosixやWinsockとよく似ています(注：xdataとは、データを割り当てる場所をコンパイラに指示するKeil Cキーワードです)。DS80C400では、join()によりマルチキャストグループに参加し、leave()により離脱します。この例及び後述の例で使用しているbind()やrecvfrom()などのライブラリの呼び出しは、すべてステータスコードを返します。ここに示す簡略例では何もしていませんが、実際のプログラムでは、戻り値を検査し、エラーなどの場合は必要な処理を行うようにすべきです。

パワーオーバーサネット

デバイスをネットワークに接続することには、ネットワーク用のケーブルが必要になるというマイナス面もあります。幸い、イーサネットケーブルの余分なワイヤに電源を統合してしまうことが可能です。さまざまなソリューションがありますが、最もよく利用されるのはIEEE802.3af規格で、8ピンイーサネットコネクタの7番、8番(+)と4番、5番(GND)で48Vを供給します。この電圧は電話アプリケーションで使われているため、ケーブルキャビネットの多くは48V電源を供給することができます。

マイクロコントローラでは、適切な電圧まで下げる必要があります。www.maxim-ic.com/ja/appnoteindexのアプリケーションノートの中に、MAX5910とMAX5014を使って効率の良いIEEE802.3af準拠回路を構築する方法が説明されています。

ネットワーク型PAシステムをビルの入出場管理システムやネットワークサーバに接続すれば、呼び出す従業員がいる可能性の高い場所を知ることが可能になります。また、アナウンスを自動的にくり返したり、ウェブサイトから呼び出し依頼を入力するだけで自動的にアナウンスが行われたりするシステムにすることが可能です。

テキスト音声変換

PAシステムには、マイクで音声を拾ったり、録音し、ネットワークサーバ上に保存された音声を再生したりという使い方があります。これ以外に、電子メールやウェブページから受け取ったテキストや携帯電話のショートメッセージで受け取ったテキストを音声に変換して放送するという使い方があります。

音声合成の追加は簡単です。マスタコントロールサーバにテキスト音声変換エンジンを搭載し、入力されたテキストから音声波形を生成すればいいのです。生成された音声波形は、他の音声信号と同じようにスピーカに送信することが可能です。スピーカモジュールの変更は不要です。

テキスト音声変換エンジンは広く普及しており、Mac OS Xのように変換エンジンを組み込んだOSもあるほどです。freetts.sourceforge.net/には、フリーのJavaスピーチエンジンもあります。商用ソリューションの方がより自然な音声を生成しますがwww.nuance.comでは、その1つであるVocalizerのデモとしてアメリカ英語、イギリス英語、オーストラリア英語が体験できます。

エンターテインメント品質のオーディオ

CDのオーディオ信号を無圧縮で送信するのは問題があります。44.1kHz x 16ビットのステレオサンプリングでは、1.4Mbpsものネットワーク帯域を消費し(これは10Mbネットワークの30%近くに相当します)、ほとんどのネットワークで帯域が不足します。

MP3などの圧縮アルゴリズムを利用すれば、データレートが低下しネットワーク負荷が1/10に下がるので、実現可能なシステムになります。これは、ハードウェア伸長チップ(DS80C400)を使えば簡単に行えます。36MHzほどのクロック速度で、192kbのMP3をシームレスに再生することが可能です。www.mp3elf.netにあるTINI MP3プロジェクトは、ダラスセミコンダクタのマイクロコントローラを使ったネットワークMP3です。このマルチメディアアプリケーションについては、概略図と詳細が入手可能です。

マルチキャスト

DS80C400のROMとTINIランタイムは、マルチキャストをサポートしています。あるソースから単一デスティネーションに送られるユニキャストパケットと異なり、マルチキャストでは、1つのデータを複数のデスティネーションホストに送ることが可能で、回線トラフィックの重複を防ぎ帯域の浪費を防ぐことができます。マルチキャストは、ブロードキャストとも異なります。マルチキャストでは、ホストは、IGMPプロトコルによって1つ、あるいは複数のマルチキャストグループにサブスクライブするため、マルチキャストトラフィックは、受信者が少なくとも1つは存在する部分にのみルーティングされます。複数のネットワークがあるとき、ブロードキャストメッセージはブリッジを必要としますが、マルチキャストメッセージならルータが転送してくれます。

ネットワーク上を流れるマルチキャストパケットは、デスティネーションIPアドレスを示す特殊なクラス(マルチキャストグループとも呼ばれる)を持ちます。アプリケーション側から見ると、マルチキャストのサポートはごく簡単です。join()関数を呼びだけでホストをマルチキャストグループに追加できますし、マルチキャストパケットの受信は他のUDPトラフィックと変わる点がありません。パケット受信が不要になりグループから脱退するときはleave()関数を使います。なお、マルチキャストグループに参加したホストは定期的にメンバシップリポートを送出します。グループに参加したままホストがクラッシュしても、ある時間が経つとメンバシップは失効します。マルチキャストグループを採用する場合には、RFC 2365のガイドラインに従い、複数グループを割り当てることがないように気をつける必要があります。

映像

DS80C400にはビデオソースも接続できます。このようなシステムは、安価なカメラで1秒ごとに取得した画像をネットワーク経由で受信し、表示あるいは保存するセキュリティカメラとしての使い方ができます。サーバサイドのポストプロセスで動きを検出し、警備員に対して警報を発するようにすることも可能です。

カメラとしては、携帯電話に搭載されているものの方がいいでしょう。小型で安価、しかも容易に手に入るからです。製品の多くはシリアルプロトコルを採用していますが、メーカーごとに微妙な違いがあります。必要な技術情報をすべて確認してからカメラのメーカーや機種を決めることをお勧めします。

ネットワーク型

カメラシステムの例:

ネットワーク型カメラシステムの例として、模式図やプログラム例が

ftp://ftp.dalsemi.com/pub/tini/ds80c400/reference_designs/netcam/にあります。

実験の結果、DS80C400は、外付けの映像圧縮ハードウェアなしで、1秒あたり4フレームの白黒映像(240x180)を送信することができること、また、通常品質の音声に十分な余裕があることが確認されています。以下に示すプログラム例は、ネットワークサーバに対するTCP接続を開始し、画像を送信するものです(例では接続を終了してはいますが、実際のアプリケーションでは、オーバーヘッドを削減するため接続したままに保ちます)。

```
int s; /* socket handle */
struct sockaddr address; /* IP address */
unsigned char xdata buffer[1600];

/* Step 1: Create the socket */
s = socket(PF_INET, SOCK_STREAM, 0);

/* Step 2: Fill in the target address 192.168.0.11 */
memset(address, '\0', sizeof(address));
address.sin_addr[12] = 192;
address.sin_addr[13] = 168;
address.sin_addr[14] = 0;
address.sin_addr[15] = 11;
address.sin_port = 8080; /* Target port */

/* Step 3: Connect to the server */
connect(s, &address, sizeof(address));

/* Step 4: Send the data in buffer */
send(s, buffer, sizeof(buffer), 0);

/* Step 5: Close the connection */
closesocket(s);
```

closesocket()関数は、Unixネットワークプログラムにおけるclose()関数に相当します。DS80C400では、close()関数はファイルシステムで使用されています。Windowsシステムと同じようにDS80C400でも、ソケットハンドルとファイルハンドルに互換性がなく、ソケットにはソケット専用の関数を使う必要があるのです。

カメラシステムでは、DS80C400を限界の75MHzに近い73.7MHzで駆動します。この73.7MHzは、まず基本モード水晶発振器で18.432MHzを生成し、それをDS80C400内蔵PLLで4倍に周波数を逡倍して得ます。こうすることにより相対的なシステムコストを削減しつつ、マイクロコントローラの最大周波数近くで動作させることが可能になるのです。また、18.432MHz x 4という周波数は、非同期シリアル通信用ボーレートとしても適切です。

ネットワークドア

セキュリティカメラに双方向性の音声とボタンスイッチ、ブザーを組み合わせてみます。これで、ネットワークドアが構築可能です(図3)。これは無限に繰り返されるアプリケーションで、特にアクセス管理とセキュリティログと組み合わせる場合にはそうなります。

DS80C400では、ボタンスイッチとブザーを周辺機器としてI/Oポートに接続可能です。Keil Cでは、sfrとsbitによってI/Oポートを定義します。

```
/* Define port 1 */
sfr p1 = 0x90;
/* Define P1.7 (port 1 is bit addressable) */
sbit p1_7 = p1^7;

/* Toggle P1.7 */
p1_7 = !p1_7;
```

DS80C400に内蔵されたiButtons及び1-Wire用マスタインタフェースを使えば、ネットワークドアシステムに認証機能を追加できます(このインタフェースはプログラミングが難しいので、ライブラリを用意してあります)。自動ドアの鍵受座の接続方法が、www.ibutton.com/TINI/applications/lock/にあります。

もう一つ、ご紹介しておくことがあります。ネットワークドアなどのシステムでは、マルチプロセス(マルチタスク)とすることが多いでしょう。DS80C400のROMには、タスクスケジューラが搭載されています。その使い方(C言語の例)は、以下のとおりです。この場合も、工業用アプリケーションでは戻り値をチェックすることをお勧めします。

```
unsigned char pri, task;

/* Get the current task */
task = task_getcurrent();

/* The current task's priority */
pri = task_getpriority(0);

/* Decrease the priority */
task_setpriority(0, pri-1);

/* Sleep */
task_sleep(0, 0, 500);
```

便利な関数として、現在のタスクをコピーして新しいタスクを作成するtask_fork()関数なども用意されています。また、task_kill()によりタスクを終了したり、task_suspend()によりタスクを保留にしたりすることもできます。詳細は、ウェブサイトにある"High-Speed Microcontroller User's Guide: DS80C400 Supplement"(英語版)をご覧ください。

まとめ

小型で安価なネットワーク型マイクロコントローラを中心として、便利なマルチメディアアプリケーションをいろいろと構築することが可能です。この記事を参考に、DS80C400を使ってみてください。なお、www.maxim-ic.comでサンプルの無償提供をしています。

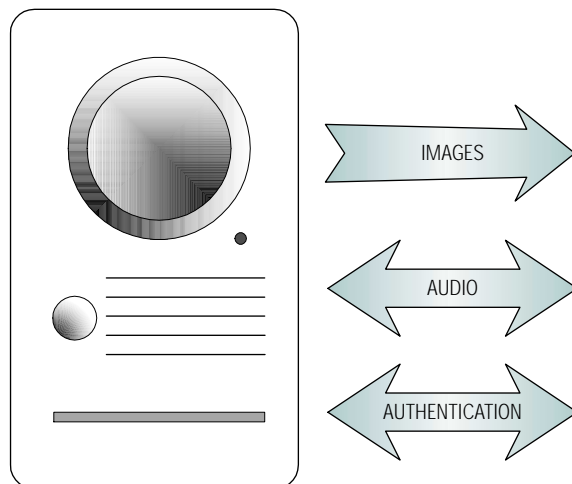


図3. 映像と音声、入出場管理を組み合わせたネットワークドアインタフェースの例

DS80C400用ソフトウェアの開発

DS80C400用ソフトウェアは、さまざまな環境で開発可能です。短期間でプロトタイプを作成したい場合には、JavaとTINIランタイム環境が適当でしょう。あらゆるサイクルが重要なアプリケーションでは、アセンブリ言語を使って人間が最適化するのがベストです。

今回のアークティクルでは、C言語を例に取り上げました。Keil Cコンパイラ(www.keil.com)は、DS80C400の24ビット連続アドレッシングモードをサポートしており、プログラミング/データスペースとして16MBが利用できます(アプリケーションノート606、"Configuring Keil PK51 Tools to Support 24-Bit Contiguous Addressing Mode"(英語版)をご覧ください)。このモードを使うためには、eXtendedバージョンのコンパイラとリンカ(CX51、LX51)が必要です。これらのツールは、Professional Developer's Kit (PK51)に含まれています。

ダラスセミコンダクタは、DS80C400内蔵ネットワークスタックとのインタフェース用Cライブラリを提供しています。ダラスセミコンダクタのftpサイトでは、このライブラリだけでなく、Keil開発環境でDS80C400用プログラムを開発する詳細な方法の図解が入手できます(<ftp://ftp.dalsemi.com/pub/tini/ds80c400/>)。このライブラリを使用すれば、ネットワークプログラミングが容易になります。たとえば、TCP接続の確立が、socket()とconnect()というよく知られたシーケンスで可能になります。lists.dalsemi.comのTINIメーリングリストで技術サポート(英語によるサービス)を受けることも可能です。