

つながるワイヤレス通信機器の開発手法

ファームウェアを設計する(その2)

第14回 — エラー制御と高周波制御を実現する方法

太田博之

前回(本誌2004年2月号, pp.140-145)は、ワイヤレス通信機器用のファームウェア、特に、通信系ファームウェアの低レベル通信制御、エラー制御、高周波回路の制御の概要と、低レベル通信制御の詳細を説明した。今回は、通信系ファームウェアの機能についてさらに少し突っ込んで説明する。また、ファームウェア開発全般について述べる。(筆者)

通信系ファームウェアには主に次の三つの機能がある。

1) 低レベル通信制御

(制御メッセージの授受、状態遷移制御)

2) エラー制御

3) 高周波回路の制御

(周波数設定、送信出力設定、送信タイミング設定)

1)については前回(本誌2004年2月号, pp.140-145)詳しく説明した。今回は2)と3)を中心に話を進める。

1. 「単方向」と「双方向」、二つのエラー制御方法

2)のエラー制御を大きく分けると、図1のようにFEC(forward error correction)とARQ(automatic request)の2種類がある。

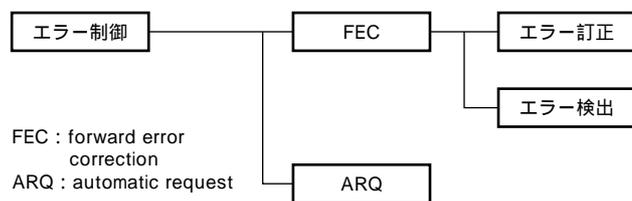


図1 エラー制御

エラー制御は、大きくFEC(forward error correction)とARQ(automatic request)に分類できる。

FECは、送信側で符号化したエラー制御用のデータを送信し、受信側で復号する過程において通信路で起きたエラーを訂正または検出する方法である。FECは、エラーの訂正や検出を受信側のみで行う。

● 双方向のエラー制御を行うARQ

一方、ARQは受信側でエラーを検出した場合、送信側に対して同じデータの再送を要求し、複数回のやり取りで正確なデータを受信する方法である(図2)。FECと異なり、ARQは双方向で通信できなければ実現できない。

身近な例として、テレビ放送のエラー制御を考えてみよう。一般に、テレビ局からは送信するだけ、各家庭では受信するだけの片方向通信であるため、FECを使用することは可能だが、ARQを使うことはできない(ただし、最近はデータ放送などによる双方向サービスも行われている)。

ARQを実現するためには、通信路で生じたエラーを受信側で検出する機能と、エラーの有無を送信側に伝えられる双方向の通信路を備えることが必要である。これら二つ

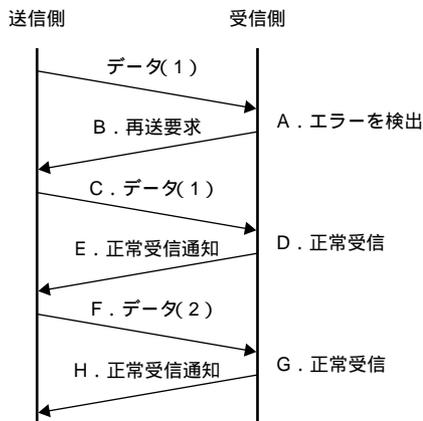


図2
ARQ

ARQでは、受信側でエラーを検出したとき、送信側に対して同じデータを再送することを要求する。複数回のやり取りで正確なデータを受信できる。

の条件がそろっているものとして、図2の動作を説明する。

送信側は最初のデータであるデータ(1)を送信する。受信側ではCRC(cycle redundancy code)などのエラー検出処理によってエラーの有無を確認する。図2のAにおいてエラーが検出されると、受信側から再送要求を出す(図2のB)。再送要求を受けた送信側ではもう一度最初と同じデータ(1)を送信する(図2のC)。Cで送信されたデータ(1)が、今度はDにおいて正常に受信されたとする。すると、受信側では正常に受信されたことを正常受信通知として送信側に知らせる(図2のE)。この通知を受けた送信側は次のデータであるデータ(2)を送信する。Gにおいて正常受信されると、データ(2)に対する正常受信通知を送信側に送る(図2のH)。以降、上記の動作を繰り返す。

この動作の中で使われている再送要求はNAK(non-acknowledge)、正常受信通知はACK(acknowledge)と呼ばれる。

● エラー検出とメッセージの組み立てはハードで実現

図3に、ARQの機能をハードウェアとファームウェアの協調動作で実現しているようすを示す。図3の場合、携帯電話の基地局が図2の送信側、携帯電話が図2の受信側に

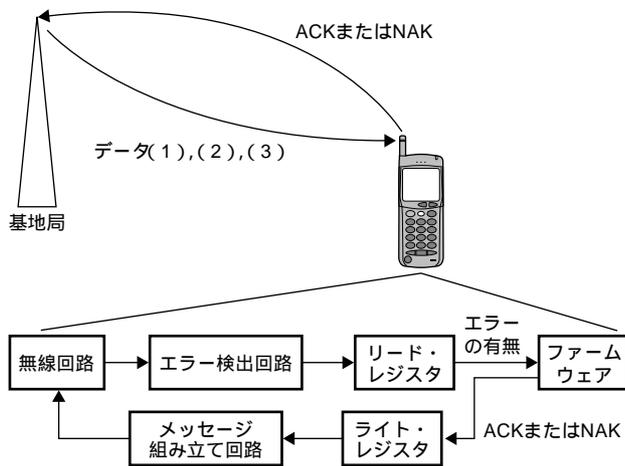


図3 ARQの実現方法

エラーの検出、通知メッセージの組み立てはハードウェアで行い、どのような通知(NAKまたはACK)を返すかの決定はファームウェアで行う。

● ファームウェア開発者は処理時間の伸縮を考慮するべし

相当する。

基地局から送られてきた信号は無線回路を通り、エラー検出回路でエラーが検出される。ファームウェアはリードレジスタを介してエラーの有無を知る。ファームウェアでエラーなしを告げる正常受信通知(ACK)を返すか、エラーありを通知する再送要求(NAK)を返すかを決定する。ライトレジスタでは、正常受信通知または再送要求のいずれかを設定する。ライトレジスタに書かれた情報を見て、ハードウェアであるメッセージ組み立て回路が正常受信通知または再送要求のメッセージを組み立て、無線回路を通して携帯電話の基地局に送信する。

図3の機能ブロックの代わりに、時間軸をもとにARQの動作を示したものが図4である。時間軸における動作の流れも、受信 エラー検出 ACKまたはNAKの指示 メッセージの組み立て 送信の順序である。ほとんどのシステムでは受信して次に送信するまでの時間が決まっている(ただし、これは通信システムによる)。

なお、図4はBluetoothのシステムの例である。Bluetoothの規定では、受信フレームの先頭から正常受信通知(または再送要求)の送信フレームの先頭までが625μsである。そのため、これらの時間を満足するようにハードウェアとファームウェアの処理時間を決める必要がある。ハードウェアで行っているエラー検出とメッセージ組み立ての部分の処理時間は、通常、数μs程度である。これに対してファームウェアの処理は非常に時間がかかり、数十μs~数百μsかかることもよくある。これらの処理時間を考慮してハードウェアとファームウェアの処理の切り分けを行うことがたいせつである。

● ファームウェア開発者は処理時間の伸縮を考慮するべし

ハードウェアの処理時間はほぼ一定であるが、ファームウェアの処理時間はそのときの条件によって延びたり縮んだりする。ファームウェアの処理時間の伸縮を一般的な形で示すと図5のようになる。ファームウェアは、複数の処理(図5の場合は処理1, 2, 3)を一つのマイコンで順番に

図4 ARQの動作シーケンス

Bluetoothのシステムについて、時間軸をもとにARQの動作を示した。Bluetoothの仕様では受信フレームの先頭から送信フレームの先頭までの時間が625μsと規定されている。この時間を満足するようにハードウェアとソフトウェアの切り分けを行う必要がある。

