AVR336：ADPCM復号器

要点

- 実時間でADPCM信号を復号するAVR応用
- 16,24,32,40kbpsのﾋﾞｯﾄ速度を支援
- ATmega128で1分以上の再生時間(16kbpsに於いて)
- PWM動作でのﾀｲﾏ/ｶｳﾝﾀを用いて再生される復号済み信号

1. 序説

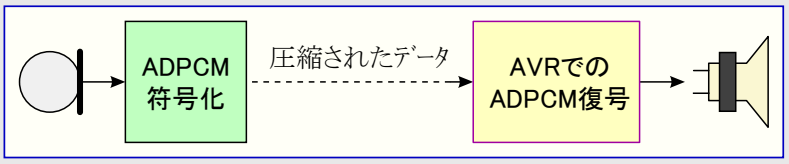
適応的差分ﾊﾟﾙｽ符号変調またはADPCM(Adaptive Differential Pulse Code Modulation)は主に電気通信での通話圧縮に使われるﾃﾞｼﾞﾀﾙ圧縮技術です。ADPCMは音楽や音響効果のように、話以外の他の信号にも使うことができる波形符号化/復号(ｺｰﾃﾞｯｸ)です。ADPCMは高度な低ﾋﾞｯﾄ速度音声符号化技術よりも単純で、重い計算を必要とせず、それは符号化と復号が相対的に短い時間で行えることを意味します。

ADPCMは64kbpsの本来の流れの速度で8ﾋﾞｯﾄ、8kHzの圧縮に通常使われます。ADPCM信号の符号化に2ﾋﾞｯﾄだけが使われる最高圧縮比で符号化されると、流れの速度は16kbps、換言すると元の25%に減らされます。4ﾋﾞｯﾄ符号化の使用では流れの速度が32kbps、元の50%で、信号の品質は殆どの応用に関して良好です。

この応用記述はADPCM信号の復号に集中します。ﾊﾟﾙｽ幅変調(PWM)された出力信号を生成するのにﾁｯﾌﾟ上のﾀｲﾏ/ｶｳﾝﾀを使い、これはその後に簡単な外部の濾波器を通して渡されます。濾波器は只少しの外部部品から成り、ﾃﾞｼﾞﾀﾙ信号をｽﾋﾟｰｶへの接続に向くｱﾅﾛｸﾞに戻します。

ﾀｲﾏ/ｶｳﾝﾀでのD/A変換についての更なる情報に関しては「**AVR335:AVR®とDataFlash®とでのﾃﾞｼﾞﾀﾙ録音再生器**」応用記述を参照してください。

図1-1. AVRﾏｲｸﾛ ｺﾝﾄﾛｰﾗでのADPCM復号



2. 動作の理屈

ADPCMの原理は直前の値から現在の信号値を予想して実際と予想した値間の差だけを送出することです。素のﾊﾟﾙｽ符号変調(PCM:Pulse Code Modulation)では現実または実際の信号値が送出されるでしょう。ADPCMでは予測信号値と実際の信号値間の差が通常かなり小さく、これは対応するPCM値よりも少ないﾋﾞｯﾄを用いて表すことができます。

望む品質と圧縮比に依存して、差信号は4、8、16または32ﾚﾍﾞﾙを用いて量子化されます。ADPCM符号化器の構成図が図2-1.で示されます。符号化部がどう動くかのより多くの情報については「参考文献」の2.をご覧ください。

8ﾋﾞｯﾄ AVR® マイクロ コントローラ

応用記述

本書は一般の方々の便宜のため有志により作成されたもので、Atmel社とは無関係であることを御承知ください。しおりのはじめにでの内容にご注意ください。

Rev. 2572A-11/04, 2572AJ2-01/21

図2-1. ADPCM符号化部の構成図

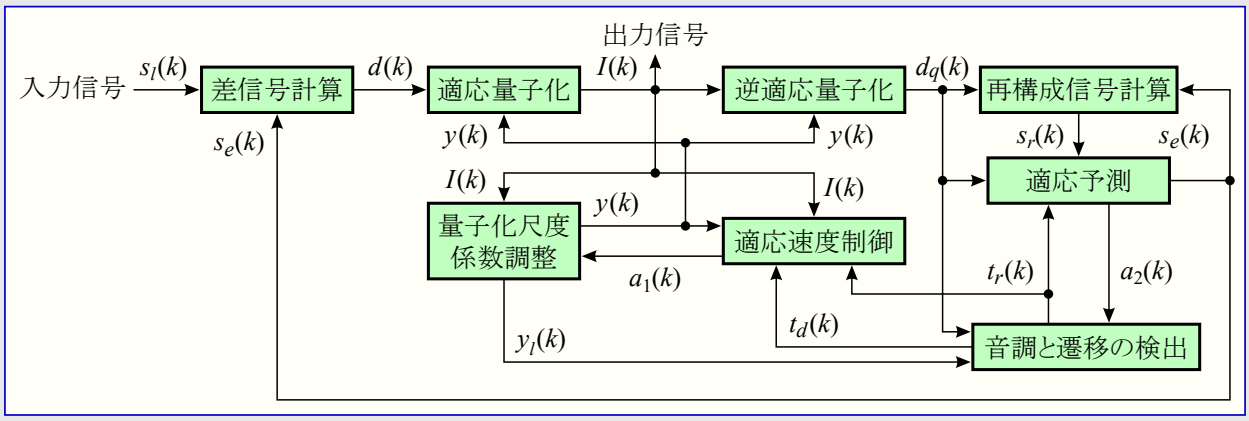
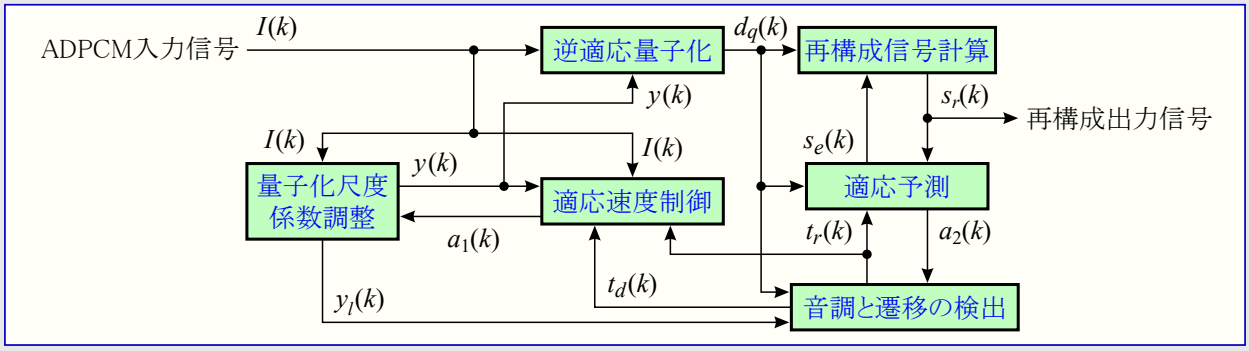


図2-2.で示される復号部は量子化された値を取って、逆量子化を実行し、そして復号された信号を得るために予測された信号から結果を引きます。

復号部は以下でもっと詳細に記述されます。項目については図2-2.での構成名を参照してください。正確な式については「追補B:法式」をご覧ください。

図2-2. ADPCM復号部の構成図



2.1. 逆適応量子化

$$d_q(k) = 2^{d_q \ln(k)+y(k)}$$

この部分は対数量子化差信号 $d_q \ln(k)$ と適応係数 $y(k)$ から直線量子化差信号 $d_q(k)$ を計算します。対数量子化差信号 $d_q \ln(k)$ は指標としてADPCM符号語 $I(k)$ を用いて静的参照表から得られます。直線領域差信号は $2^{d_q \ln(k)+y(k)}$ を解くことによって計算されます。

2.2. 量子化尺度係数調整

逆適応量子化で使われる尺度係数 $y(k)$ はここで計算されます。尺度係数 $y(k)$ は他の2つの係数、高速(非固定)尺度係数 $y_u(k)$ と低速(固定)尺度係数 $y_l(k)$ から成ります。2つの尺度係数は異なる形式の信号を扱うのに必要で、高速尺度係数は大きな変動を持つ信号(例えば、会話)への適応を許し、一方低速尺度係数は信号が緩やかに変化する時(例えば、楽音信号)に必要とされます。尺度係数はこれら2つの直線結合です。比率は次項で記述される速度制御パラメータ a_1 によって決められます。

2.3. 適応速度制御

適応速度はパラメータ a_1 によって制御されます。それは会話信号での1と音響帯域データ信号での0で近似します。a_1 を得るためにADPCM符号語 $I(k)$ の2つの平均的な大きさの程度(長期間と短期間)が計算されます。これら2つの差は $I(k)$ の平均的な大きさがどう変化するかを示します。速度制御パラメータ a_p によって示されるように、差が小さい場合は $I(k)$ の大きさが一定で、差が大きい場合は $I(k)$ の大きさが変化しています。このパラメータは遷移が検出された場所や信号がアイドルの時の特別な場合も考慮します。最後に、制限された速度制御パラメータ a_1 は(0～2に及ぶ) a_p からそれを0と1の間の範囲に制限することによって得られます。これは高速から低速への適応形態遷移を遅らせるように非対称で行われます。

2.4. 音調と遷移の検出

会話の代わりにデータ信号を扱う時の符号化/復号(コーデック)応答を改善するために音調と遷移の検出が含まれます。

信号が狭い周波数帯域(例えば、単調信号)だけを使う場合、量子化は高速適応形態に設定されます。

遷移が検出された場合、量子化は高速適応形態(t_r=1)に設定され、適応予測の係数が0に設定されます。

2.5. 適応予測

$$s_e(k) = s_{ep} + s_{ez}$$

適応予測は量子化された差信号 $d_q(k)$ から信号予測 $s_e(k)$ を計算します。これは極を形成する2次構造体と零点を形成する6次の構造体の2つの構造体を使います。両構造体の係数は簡単化した勾配法で更新されます。安定を保証するために極模式構造体での係数は制限されなければなりません。零点模式構造体での係数更新用の式は各種ﾋﾞｯﾄ率(5ﾋﾞｯﾄ対2,3,4ﾋﾞｯﾄ)に対して僅かに違います。

遷移が検出されたなら、両構造体の全ての予測係数が0に設定されます(直前項をご覧ください。)

2.6. 再構成信号計算

$$s_r(k) = s_e(k) + d_q(k)$$

再構成信号 $s_r(k)$ は信号予測 $s_e(k)$ を差信号 $d_q(k)$ へ単純に加算することによって計算されます。

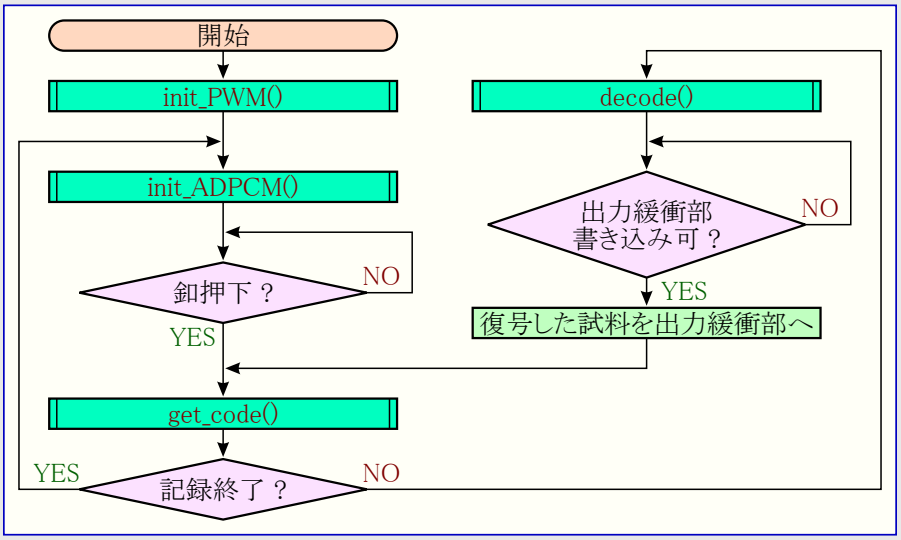
3. 実装

本章は復号器のｿﾌﾄｳｪｱ実装を記述します。

3.1. ｿﾌﾄｳｪｱ

この復号器はｱｾﾝﾌﾞﾘ言語で書かれたいくつかの部分と共にIAR SystemsのCｺﾝﾊﾟｲﾗを用いて実装されます。図3-1.で示される復号器の主構造は簡単で、ﾌﾟﾛｸﾞﾗﾑは使用者の釦押下を持ち、そして予め符号化された試料をﾌﾗｯｼｭ ﾒﾓﾘから復号します。最初にﾌﾗｯｼｭ ﾒﾓﾘからﾃﾞｰﾀを読むget_code()を呼び、それからADPCM符号語を抽出します(1ﾊﾞｲﾄが多数の符号語を含みます)。

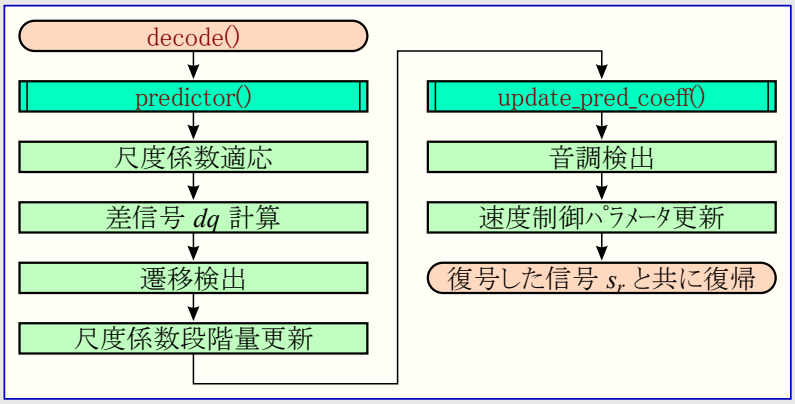
図3-1. 主ﾌﾟﾛｸﾞﾗﾑの流れ図



次に、図3-2.で示されるようなdecode()関数が呼ばれます。この関数は信号予測(predictor()をご覧ください)、量子化尺度係数 y、差信号 dq を計算します。起こり得る遷移信号が検出され、尺度係数段階量ﾊﾟﾗﾒｰﾀと適応予測係数が更新されます。最後に、復号部は起こり得る音調信号を検出して適応速度制御ﾊﾟﾗﾒｰﾀを更新します。

再構成された信号 s_r は(ﾎﾟｰﾄBのPB5に出力する)PWM動作形態のﾀｲﾏ/ｶｳﾝﾀ1と外部濾波器から成る、簡単なD/A変換器を通して出力されます。

図3-2. 復号部関数の流れ図



3.1.1. ITU-T G.726との適合性

ソフトウェアは前述されたものにかなり近く従いますが、より効率的な計算のためにいくつかの変更が行われています。ADPCM規格のITU-T G.726は浮動小数点演算で行うための再構成信号 $s_r(k)$ に関係するいくつかの計算を定義します。この実装ではこれらの計算が省かれ、代わりに固定小数点演算を用いて実現されます。これはプロセッサ クロック周期を節約するために行われています。浮動小数点演算の代わりに固定小数点を使うことは再構成信号の品質に於いて小さな変形を引き起こします。

2つの部分、即ちμ-lawとA-lawの変換と同期符号化調整が完全に省かれています。μ-lawとA-lawの変換は試料毎のビット数を減らすのに用いられます。出力が直接D/A段を駆動するため、μ-lawとA-lawの変換の必要は全くありません。

G.726規格で定義される同期符号化調整もこの実装では省かれています。同期符号化調整の目的は、この応用で使われない、同期縦走符号化(例えば、ADPCM⇒PCM⇒ADPCM)での歪を防ぐことです。

可能な最良の品質を達成するため、符号化と復号の処理は同じ方法が使われるべきで、即ち符号化も固定小数点演算で行われるべきです。

3.2. ハードウェア

ソフトウェアを走らせるのにATmega128プロセッサが使われます。最良の品質を得るため、PWM基本周波数を濾波するのにPWM出力と拡声器(スピーカ)/ヘッドフォン/増幅器の間に能動低域通過濾波器の使用が高く推奨されます。PWM出力と提案される濾波器回路はAVR335応用記述で記述されています。より簡単な回路図、片側低電圧演算増幅器での2次能動濾波器実装が「追補C」で示されます。

STK501拡張部とのSTK500開発基板も推奨されます。8kHz採取速度での信号の復号時、AVRは16MHzで走行しなければならず、このような場合は外部発振器が必要とされます。

4. 実装例

本章はSTK500/501評価基板とATmega128で圧縮されたADPCM試料を復号する方法の段階的な指示を与えます。8kHz採取速度でのADPCM信号を復号するために、システム クロック周波数は16MHzが必要です。

4.1. 即時開始

ADPCM復号器を使うには以下のとおりです。

- 含まれている符号化された音試料の1つをソース フォルダに複写してそれを"data.c"に改名してください。
- IAR Embedded Workbenchを開始し、作業空間ファイルを開いて作業空間を構成設定し、"adpcm.h"のBITSとINPUTSIZEのパラメータを選択した音試料ファイル("data.c")に従って編集し、プロジェクトを再構築してください。
- コンパイルしてリンクしたプログラムを目的対象のAVRにプログラミングして(書き込んで)ください。
- PWM出力(ATmega128のPB5)と拡声器(スピーカ)間に出力濾波器を組み立てて接続してください。

信号は今や復号されて再生される準備が整っています。

4.2. ADPCM音ファイル

この応用記述と共にいくつかの音試料が含まれています。ファイルの1つをソース フォルダに複写してそれを"data.c"に改名してください。代替として、新しいADPCM音ファイルを作成するのに第三者のソフトウェアを使ってください。独自のADPCMデータを使う場合、含まれる試料ファイルで示されるように、2進データが配列内に複写されなければならないことに注意してください。ADPCMファイルの符号化の方法は「追補E」をご覧ください。

4.3. コンパイルとリンク

IAR Embedded Workbenchでのソース コードのコンパイルとリンクは次のとおりです。

- 作業空間"AVR336.eww"を開いてください。
- 出力形式をUBROF-8に設定してください(Project⇒Options⇒XLINK⇒Output⇒UBROF-8)。IAR Embedded Workbench 3.10は既定でUBROF 9を生成しますが、AVR Studio 4.10は現在UBROF 8とそれ以前を支援します。これは新しい公開と送り出しのために変わるかもしれません。
- プロジェクト ウィンドウに於いて、それを開くために"adpcm.h"ファイル上でダブルクリックしてください。ADPCM音ファイルの形式に合うようにそのファイル内のADPCM設定を更新してください。INPUTSIZEは音記録の長さを定義し、BITSは符号化に使われたビット数を定義します。
- プロジェクト ウィンドウに於いて、"data.c"ファイル上を左クリックしてポップアップ メニューからコンパイラを選んでください。これはコンパイラが古い音ファイルの予めコンパイルされた版を使わないことを確実にするためです。
- プロジェクトをコンパイルしてリンクしてください(Project⇒Make)。

4.4. ハードウェア構成設定

STK501をSTK500に装着して上乗せ部にATmega128を配置してください。フラット ケーブルを用いて押し釦をポートDに接続してください(SWITCHESとPORTD間にケーブルを接続してください)。望む周波数で走行するように目的AVRを構成設定してください(既定は16MHz外部クリスタル使用です)。直接のケーブル接続または書き込み器/デバッガの選択を使ってAVRとSTK500をPCに接続してください。

4.5. ファームウェア構成設定

AVR Studioを開始してIARの出力フォルダ("Debug¥Exe"または"Release¥Exe")から"adpcm.dbg"ファイルを開いてください。デバッグ基盤と使うAVRに関してAVR Studioを構成設定してください。プログラムは今や走行の準備が整っています。

4.6. 濾波器回路

出力濾波器回路は必須ではありませんが、音の品質をかなり改善します。1つの有り得る濾波器回路がAVR335応用記述で記載されています。AVR335で記述される回路を使う場合、回路の出力部だけが必要とされ、マイクロフォン増幅器関しては全く必要とされないことに注意してください。代替として、より簡単な濾波器回路が「追補C」で示されます。

5. 移設

プログラムはかなり容易に他のAVRデバイスに移せます。変更が必要かもしれないものには(デバイスに)封入されたデータ メモリとPWM出力ピンに関する参照型を含みます。例えば、ATmega32で復号器を使うには、"main.c"、"adpcm.c"、"data.c"のファイルに於いて全ての"_hugeflash"を"_flash"で置き換えてください。ATmega32のPWM出力OC1AはPB5の代わりにPD5ピンに配置されています。故に、"adpcm.c"ファイル内の"initialize_pwm()"関数はそれによって変更されなければなりません。

統合された乗算器付きのAVRを使うことが推奨されます。

6. 性能

音記録に必要とされるフラッシュ メモリの量は採取速度、ADPCMビット率、そして勿論記録の長さに依存します。下表は各種のビット率と採取速度に於ける1秒の音に対するメモリ必要条件を示します。

例えば、8kHzの採取速度と4ビットで符号化された10秒の記録は320kビット=40Kバイトのメモリが必要です。これは例えば、ADPCM復号器(約3Kバイト)を走行するATmega128は最大31秒よりも多く(=(128-3)Kバイト÷(32÷8))、8kHz採取、4ビットADPCM符号化データを再生することができます。

表6-1. 1秒の音データに対して必要とされるフラッシュ メモリ

ビット割合 (ビット数/採取)	採取速度 (Hz)		
	4000	6000	8000
2	1.00Kバイト	1.50Kバイト	2.00Kバイト
3	1.50Kバイト	2.25Kバイト	3.00Kバイト
4	2.00Kバイト	3.00Kバイト	4.00Kバイト
5	2.50Kバイト	3.75Kバイト	5.00Kバイト

6.1. クロック周期数

下表は1つの試料に対するクロック周期数必要条件を示します。ビット率はクロック周期数必要条件に於いて最小への影響だけを持ちます(表の平均値は4ビット版からです)。必要とされるAVRのクロック周波数を探すには総クロック周期数に採取速度を乗算してください。

例えば、音記録が7812Hz採取で、平均総クロック周期数必要条件が1930の場合、AVRは1930×7812Hz=15MHzでの動作が必要で、これは上余裕で16MHzへの丸めです。最悪の場合のクロック周期数必要条件は16MHzを越え、故に長くかかりすぎる計算によって引き起こされる可聴歪を防ぐために小さな緩衝部が必要とされます。緩衝部の長さは調整することができます(ソース コードをご覧ください)。長い緩衝部は緩衝不足の危険を減らしますが、より多くのメモリ(要素毎に2バイト)が必要です。

表6-2. 平均と最悪の場合のクロック周期数必要条件

関数	平均クロック周期数/試料	最悪の場合のクロック周期数/試料
get_code()	130	140
decode()	1600	1860
T/C3ISR	60	70
その他	140	150
復号器周回合計	**1930**	**2220**

6.2. メモリ

メモリ必要条件は下表で一覧にされます。ソース コードは全てのビット レベル(2～5ビット)を使う可能性を含みますが、代表的な応用は1つのビット レベルだけが必要とされます。他のビット レベルの支援を省くことがプログラム メモリの必要条件を減らします。

表6-3. (音データを除く)メモリ必要条件

コード メモリ (バイト)	データ メモリ (バイト)
<3000 (+音データ)	<40 (+出力緩衝部)

7. 追補A：表

この追補は4ﾋﾞｯﾄADPCM符号化で必要とされる表を示します。他のﾋﾞｯﾄ ﾚﾍﾞﾙ(2,3と5)に関する対応値はｿｰｽ ｺｰﾄﾞ(″adpcm.c″)から直接得られます。ｿｰｽ ｺｰﾄﾞではいくつかの表の値が固定小数点10進数を用いて表されています。10進部に対して予約された多数のﾋﾞｯﾄはQn表記によって示されます。例えば、Q4表記は数値4.75が2進で″0000 0000 0100.1100″として書かれます。

表7-1. 4ﾋﾞｯﾄ符号化用のW()とF()の関数の値

|I(k)|	W|I(k)| (Q4表記)	F|I(k)| (Q9表記)
0	-0.75	0
1	1.13	0
2	2.56	0
3	4.00	1
4	7.00	1
5	12.38	1
6	22.19	3
7	70.13	7

表7-2. 4ﾋﾞｯﾄ符号化用逆量子化表

|I(k)|	|DQLN(k)| (Q7表記)
0	-2048
1	4
2	135
3	213
4	273
5	323
6	373
7	425

8. 追補B：法式

この追補はADPCM信号の復号で使われる演算式を示します。項目は図2-2.の構成部名を参照してください。

8.1. 逆適応量子化

対数領域から直線へ変換するには次式が用いられます。

$$d_q(k) = 2^{d_q \ln(k) + y(k)}$$

8.2. 量子化尺度係数調整

固定化と非固定化の尺度係数は次式で結合されます。

$$y(k) = a_l(k)\, y_u(k-1) + (1-a_l(k))\, y_l(k-1)$$

ここで、

$$0 \leqq a_l(k) \leqq 1$$

非固定化(高速)尺度係数 $y_u(k)$ は次のとおりです(異なるﾋﾞｯﾄ率に対する関数W値の定義については「追補A」の表をご覧ください)。

$$y_u(k) = (1-2^{-5})\, y(k) + 2^{-5} W|I(k)|$$

固定化(低速)尺度係数 $y_l(k)$ は次のとおりです。

$$y_l(k) = (1-2^{-6})\, y_l(k-1) + 2^{-6} y_u(k)$$

$y_u(k)$ は1.06と10.00の間に制限されます。

8.3. 適応速度制御

適応速度はﾊﾟﾗﾒｰﾀ a_l で制御されます。それは会話信号での1と音響帯域ﾃﾞｰﾀ信号での0で近似します。このﾊﾟﾗﾒｰﾀは正確に1またはそれ未満に速度制御ﾊﾟﾗﾒｰﾀ $a_p(k)$ を制限することによって計算されます。

$$a_l(k) = \begin{cases} a_p(k-1) > 1 \text{なら、} 1 \\ a_p(k-1) \leqq 1 \text{なら、} a_p(k-1) \end{cases}$$

制限されない速度制御ﾊﾟﾗﾒｰﾀ $a_p(k)$ は次のように定義されます。

$$a_l(k) = \begin{cases} |d_{ms}(k) - d_{ml}(k)| \geqq 2^{-3} d_{ml}(k) \text{なら、} (1-2^{-4})\, a_p(k-1) + 2^{-3} \\ y(k) < 3 \text{なら、} (1-2^{-4})\, a_p(k-1) + 2^{-3} \\ t_d(k) = 1 \text{なら、} (1-2^{-4})\, a_p(k-1) + 2^{-3} \\ t_r(k) = 1 \text{ならば1、さもなくば} (1-2^{-4})\, a_p(k-1) \end{cases}$$

以下の変数($F|I(k)|$の短区間と長区間の平均)が a_p の計算で必要とされます。

$$d_{ms}(k) = (1-2^{-5})\, d_{ms}(k-1) + 2^{-5} F|I(k)|$$

$$d_{ml}(k) = (1-2^{-7})\, d_{ml}(k-1) + 2^{-7} F|I(k)|$$

$F|I(k)|$に関する値は静的な表から取得されます(「追補A」をご覧ください。

8.4. 音調と遷移の検出

音調検出は次のとおりです。

$$t_d = \begin{cases} a_2(k) < -0.71875 \text{なら、} 1 \\ \text{そうでなければ、} 0 \end{cases}$$

遷移検出は次のとおりです。

$$t_r = \begin{cases} a_2(k-1) < -0.71875 \text{且つ} |d_q(k)| > 24 \times 2^{y_l(k-1)} \text{なら、} 1 \\ \text{そうでなければ、} 0 \end{cases}$$

8.5. 適応予測

信号予測は次のとおりです。

$$s_e(k) = s_{ep} + s_{ez}$$

ここで、

$$s_{ep} = \sum_{i=1}^{2} a_i(k-1)\, s_r(k-i) \quad \text{それと} \quad s_{ez} = \sum_{j=1}^{6} b_j(k-1)\, d_q(k-j)$$

A-係数は次のように更新します。

$$a_1(k) = (1-2^{-8})\, a_1(k-1) + 3 \times 2^{-8} \times \mathrm{sgn}(p(k))\, \mathrm{sgn}(p(k-1))$$

$$a_2(k) = (1-2^{-7})\, a_2(k-1) + 2^{-7} \times \{\mathrm{sgn}(p(k))\, \mathrm{sgn}(p(k-2)) - f(a_1(k-1))\, \mathrm{sgn}(p(k))\, \mathrm{sgn}(p(k-1))\}$$

ここで、

$$p(k) = d_q(k) + s_{ez}(k)$$

$$f(a_1) = \begin{cases} |a_1| \leqq 0.5 \text{なら、} 4\, a_1 \\ |a_1| > 0.5 \text{なら、} 2\mathrm{sgn}(a_1) \end{cases}$$

制限は次のとおりです。

$$|a_2(k)| \leqq 0.75 \quad \text{且つ} \quad |a_1(k)| \leqq 1-2^{-4} - a_2(k)$$

ADPCM符号が2,3,4ﾋﾞｯﾄの時にB-係数は次のように更新します。

$$b_i(k) = (1-2^{-8})\, b_i(k-1) + 2^{-7} \times \mathrm{sgn}[d_q(k)]\, \mathrm{sgn}[d_q(k-i)]$$

ADPCM符号が5ﾋﾞｯﾄの時にB-係数は次のように更新します。

$$b_i(k) = (1-2^{-9})\, b_i(k-1) + 2^{-7} \times \mathrm{sgn}[d_q(k)]\, \mathrm{sgn}[d_q(k-i)]$$

遷移が検出された場合は以下のとおりです。

$$t_r = 0 \Rightarrow \begin{cases} a_i(k) = 0,\ i = 1,2 \\ b_i(k) = 0,\ i = 1 \sim 6 \end{cases}$$

8.6. 再構成信号計算

再構成信号 $s_r(k)$ は次のように信号予測を単に差信号に加算することによって計算されます。

$$s_r(k) = s_e(k) + d_q(k)$$

9. 追補C：濾波器回路

下図で図解される濾波器はセイレンキー形で2次バターワース低域通過濾波器です。遮断周波数は3000Hzです。31.25kHzのPWM基本周波数は約40dB減衰されます。この濾波器は片電源低電圧演算増幅器のTS921を用いて実装され、これは2.7～12Vの範囲の動作電圧で使うことができます。

負荷(RL)に依存して利得と動作電圧(VCC)は調整を必要とするかもしれません。この構成設定用の適切な動作電圧は+5Vです。

図9-1. 濾波器回路図

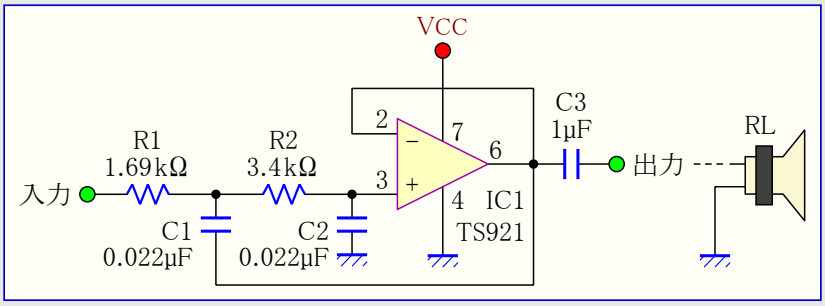


表9-1. 濾波器部品値

部品	値	説明
IC1	TS921	演算増幅器
R1	1.69kΩ	バターワース濾波器抵抗
R2	4.3kΩ	バターワース濾波器抵抗
C1	0.022μF	バターワース濾波器コンデンサ
C2	0.022μF	バターワース濾波器コンデンサ
C3	1μF	交流結合コンデンサ
RL	≧32Ω（注）	極小スピーカ、ヘッドフォンなど

注: IC1依存、製造業者のデータシートをご覧ください。

10. 追補D：内包ファイル

本章は応用記述と共に配給されるファイルを一覧にします。

10.1. 復号部

"data.c"ファイルは符号化された音記録を含みます。違う録音で実験するにはdata.cを削除して"Samples"フォルダからのファイルの1つをソースフォルダに複写し、それを"data.c"に改名してそれによって"adpcm.h"ファイルのBITSとINPUTSIZEのパラメータを変更してください。そしてプログラムをコンパイルしてリンクしてください。

復号部ファイルはSource¥Decoderフォルダに配置されています。

表10-1. 復号部ファイル

ファイル名	説明
adpcm.c	ADPCM算法
adpcm.h	ADPCMヘッダファイル
AVR336.eww	IAR作業空間/プロジェクトファイル
data.c	符号化された音データファイル(情報については先頭部をご覧ください。)
decoder.ewp	IAR作業空間/プロジェクトファイル
main.c	主プログラム
update_yl.asm	尺度係数更新のためのASMルーチン

10.2. 符号化部

符号化部ファイルはSource¥Toolsフォルダに配置されています。

ソフトウェアの一部はSun Micrsystems®, Inc.によって公の領域に公開されたコードに基きます。

表10-2. 符号化部ファイル

ファイル名	説明
decode.c	復号部用ソースコード
decode.exe	コンパイル済み復号部
encode.c	符号化部用ソースコード
encode.exe	コンパイル済み符号化部
g7*.*	符号化と復号を支援するソースコードファイル
c-izer.cpp	文字ファイル用ソースコードのCコード変換器
c-izer.exe	コンパイル済み文字ファイルのCコード変換器

10.3. 音試料

音試料はSource¥Samplesフォルダに配置されています。

表10-3. 音試料

ファイル名	技術的な詳細
Example_5bit.c	ADPCM、5ビット、7.8kHz
Example_4bit.c	ADPCM、4ビット、7.8kHz
Example_3bit.c	ADPCM、3ビット、7.8kHz
Example_2bit.c	ADPCM、2ビット、7.8kHz

11. 追補E：ADPCMﾌｧｲﾙ生成

ADPCM符号化部は、この応用記述で記述される復号部用の詰め込みﾌｧｲﾙを生成するのに必要とされます。復号部はﾃﾞｰﾀを復号し、これは何れかの標準適合ADPCM符号化で詰め込まれていますが、可能な最高品質を得るため、ADPCM音ﾌｧｲﾙを生成するのに独自化された非標準の符号化の使用が推奨されます。これは復号部がITU-T規格に厳密に従っていないためです(「ITU-T G.726との適合性」節をご覧ください)。

符号化部の動作原理は1頁で簡単に説明されます。

11.1. 符号化部の使用

この応用記述と共に含まれるのは予めｺﾝﾊﾟｲﾙされた符号化部で、これはWindows®ｵﾍﾟﾚｰﾃｨﾝｸﾞ ｼｽﾃﾑとのPCで走行することができます。符号化部は以下のﾊﾟﾗﾒｰﾀと共にｺﾏﾝﾄﾞ行から動きます。

```
encode [-2|3|4|5] [-a|u|l] -i 入力ﾌｧｲﾙ -o|c 出力ﾌｧｲﾙ
```

```
-2 G.726 16kbps (2ﾋﾞｯﾄ)ﾃﾞｰﾀ生成
-3 G.726 24kbps (3ﾋﾞｯﾄ)ﾃﾞｰﾀ生成
-4 G.726 32kbps (4ﾋﾞｯﾄ)ﾃﾞｰﾀ生成[既定]
-5 G.726 40kbps (5ﾋﾞｯﾄ)ﾃﾞｰﾀ生成

-a 8ﾋﾞｯﾄ A-law入力ﾃﾞｰﾀ処理
-u 8ﾋﾞｯﾄμ-law入力ﾃﾞｰﾀ処理
-l 16ﾋﾞｯﾄ直線PCM入力ﾃﾞｰﾀ処理[既定]

-i 入力ﾌｧｲﾙ名(2進数)

-o 出力ﾌｧｲﾙ名(2進数)
-c 出力ﾌｧｲﾙ名(C配列に貼り付けられるべく準備された形式の文字ﾌｧｲﾙ)
```

例えば、以下の命令行は"sound1.bin"と呼ばれる16ﾋﾞｯﾄ直線PCMﾌｧｲﾙを"encoded1.txt"と名付けられた文字ﾌｧｲﾙに符号化するのに32kbpsADPCMを使うことを符号化部に指示します。

```
encode -4 -l -i sound1.bin -c encoded1.txt
```

出力ﾌｧｲﾙは次のような16進一覧を含みます(16進ﾃﾞｰﾀは出力がこのように見えるかもしれないものの例に過ぎません)。

```
0xef, 0xc1, 0xff, 0xde, 0x7b, 0xff, 0xff, 0xff,
0xfe, 0xfb, 0xff, 0xff, 0xf7, 0xfe, 0x7f, 0xff,
                        ～
```

この文字ﾌｧｲﾙはその後に有効な*.cﾌｧｲﾙ内に形式化されなければならず、これはADPCM復号部ﾌﾟﾛｼﾞｪｸﾄをｺﾝﾊﾟｲﾙす時に使うことができます。この目的のために、下の太字で示される行を追加してください(16進ﾃﾞｰﾀは未だ只の例です)。

```
#pragma memory = __hugeflash
char packed_ [] = {

0xef, 0xc1, 0xff, 0xde, 0x7b, 0xff, 0xff, 0xff,
0xfe, 0xfb, 0xff, 0xff, 0xf7, 0xfe, 0x7f, 0xff,
                        ～
0x10, 0xf1, 0x45, 0x21, 0x44, 0x8c, 0xff, 0xff,
0xff
};

#pragma memory = default
char __hugeflash *packed = packed_;
```

ｺﾏﾝﾄﾞ行符号化部が"，"で終わる16進出力ﾌｧｲﾙを生成することに注意してください。

data.cとしてﾌｧｲﾙを保存してそれをADPCM復号部のｿｰｽ ﾌｫﾙﾀﾞに複写してください。最後に、配列内の要素数を数えてadpcm.hﾌｧｲﾙのINPUTSIZEﾊﾟﾗﾒｰﾀを設定するのにこの数を使ってください(「実装例」章をご覧ください)。全行の各々の要素は8個のﾃﾞｰﾀを含みます。ｶｰｿﾙがどの行かを表示する平文ｴﾃﾞｨﾀを用いて総行数を素早く数えるには、編集時にｶｰｿﾙを先頭から最後に移動してください。

C-IZERと名付けられた小さな応用も含まれます。これはENCODEによって生成された文字ﾌｧｲﾙから有効なCﾌｧｲﾙを作成するために上で記述される必要な全ての操作を実行します。応用はINPUTSIZEﾊﾟﾗﾒｰﾀ用の値も返します。

11.2. WAVﾌｧｲﾙの使用

WAVはMicrsoft®によって開発された音形式でMicrsoftのWindows®で広範囲に使われます。他の殆どのｵﾍﾟﾚｰﾃｨﾝｸﾞ ｼｽﾃﾑにWAVﾌｧｲﾙの再生を許すための変換ﾂｰﾙが利用可能です。この応用記述に含まれる符号化部はWAVﾌｧｲﾙをADPCM記録へ変換するのに使うことができ、これはその後に目的対象のAVRで再生することができます。

ADPCM符号化部でWAVﾌｧｲﾙを使うために、ﾌｧｲﾙはﾓﾉﾗﾙで16ﾋﾞｯﾄPCMとして保存されるべきです。復号部の既定設定は7.8 kHzの採取周波数を期待します。

WAVﾌｧｲﾙは(ADPCM符号化部が音ﾃﾞｰﾀと区別しない)先頭部を含みます。これは符号化された音ﾌｧｲﾙが、音試料として処理された先頭部ﾃﾞｰﾀを現実的に表現する少数のﾊﾞｲﾄで始まるかもしれないことを意味します。代表的に、これは聞き取れず、通常WAVﾌｧｲﾙ先頭部は単純に無視することができます。

11.3. 符号化部のｺﾝﾊﾟｲﾙ

符号化部はWindows®以外の他の基盤のPCで使う場合に再ｺﾝﾊﾟｲﾙされなければなりません。ｿｰｽ ｺｰﾄﾞは応用記述と共に含まれています。

12. 参考文献

1. Atmelの「AVR335:AVR®とDataFlash®とでのﾃﾞｼﾞﾀﾙ録音再生器」応用記述
   http://www.atmel.com
2. ITU-T推奨G.726: 40, 32, 24, 16 kﾋﾞｯﾄ/s 適応的差分ﾊﾟﾙｽ符号変調(ADPCM:Adaptive Differential Pulse Code Modulation)
   http://www.itu.int
3. Sun Microsystems, Inc.によるCCITT G.711, G.721とG.723音声圧縮のANSI-C実装
   ftp://ftp.cwi.nl/pub/audio/ccitt-adpcm.tar.gz

本社

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131, USA
TEL 1(408) 441-0311
FAX 1(408) 487-2600

国外営業拠点

Atmel Asia
Unit 1-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
Hong Kong
TEL (852) 2245-6100
FAX (852) 2722-1369

Atmel Europe
Le Krebs
8, Rue Jean-Pierre Timbaud
BP 309
78054 Saint-Quentin-en-Yvelines
Cedex
France
TEL (33) 1-30-60-70-00
FAX (33) 1-30-60-71-11

Atmel Japan
104-0033 東京都中央区
新川1-24-8
東熱新川ﾋﾞﾙ 9F
ｱﾄﾒﾙ ｼﾞｬﾊﾟﾝ株式会社
TEL (81) 03-3523-3551
FAX (81) 03-3523-7581

製造拠点

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
TEL 1(408) 441-0311
FAX 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
TEL 1(408) 441-0311
FAX 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3
France
TEL (33) 2-40-18-18-18
FAX (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-42-53-60-00
FAX (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR
Scotland
TEL (44) 1355-803-000
FAX (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn
Germany
TEL (49) 71-31-67-0
FAX (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
TEL 1(719) 576-3300
FAX 1(719) 540-1759

Biometrics
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex
France
TEL (33) 4-76-58-47-50
FAX (33) 4-76-58-47-60

文献請求
www.atmel.com/literature

お断り: 本資料内の情報はAtmel製品と関連して提供されています。本資料またはAtmel製品の販売と関連して承諾される何れの知的所有権も禁反言あるいはその逆によって明示的または暗示的に承諾されるものではありません。**Atmelのｳｪﾌﾞｻｲﾄに位置する販売の条件とAtmelの定義での詳しい説明を除いて、商品性、特定目的に関する適合性、または適法性の暗黙保証に制限せず、Atmelはそれらを含むその製品に関連する暗示的、明示的または法令による如何なる保証も否認し、何ら責任がないと認識します。たとえAtmelがそのような損害賠償の可能性を進言されたとしても、本資料を使用できない、または使用以外で発生する(情報の損失、事業中断、または利益の損失に関する制限なしの損害賠償を含み)直接、間接、必然、偶然、特別、または付随して起こる如何なる損害賠償に対しても決してAtmelに責任がないでしょう。**Atmelは本資料の内容の正確さまたは完全性に関して断言または保証を行わず、予告なしでいつでも製品内容と仕様の変更を行う権利を保留します。Atmelはここに含まれた情報を更新することに対してどんな公約も行いません。特に別の方法で提供されなければ、Atmel製品は車載応用に対して適当ではなく、使用されるべきではありません。Atmel製品は延命または生命維持を意図した応用での部品としての使用に対して意図、認定、または保証されません。

© Atmel Corporation 2004. **不許複製** Atmel®、ﾛｺﾞとそれらの組み合わせ、AVR®とその他はAtmel Corporationの登録商標または商標またはその付属物です。他の用語と製品名は一般的に他の商標です。

© *HERO* 2021.

本応用記述はAtmelのAVR336応用記述(doc2572.pdf Rev.2572A-11/04)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意訳されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

必要と思われる部分には( )内に英語表記や略称などを残す形で表記しています。

青字の部分はﾘﾝｸとなっています。一般的に赤字の0,1は論理0,1を表します。その他の赤字は重要な部分を表します。