

AquesTalk pico for ESP32 マニュアル

株式会社 アクエスト
www.a-quest.com

概要

本文書は、音声合成ライブラリ AquesTalk pico for ESP32 をアプリケーションに組み込んで使用するための関数定義、プログラミングの方法、注意点を示したものです。

AquesTalk pico は、ローマ字表記の音声記号列から音声波形データを生成するライブラリです。
AquesTalk pico for ESP32 は、これを Espressif Systems 社製のマイコン ESP32 用に移植したライブラリです。

本ライブラリを使用するには、ライセンスキーの設定が必要です。このライセンスキーを設定しない場合は、評価版として動作し、以下の制限があります。

評価版の制限

「ナ行、マ行」を指定すると、すべて「ヌ」と発声します

ライセンスの種類や購入方法は、弊社サイトのライセンスのページを参照してください。
ArduinoIDE や ESP-IDF 環境での具体的な使用方法をブログでも紹介していますので、そちらも参照ください。<http://blog-yama.a-quest.com/>

仕様

ライブラリ形式	ESP32 用 static ライブラリ
動作確認環境	Arduino core for ESP32、ESP-IDF
入力データ形式	ローマ字表記音声記号列 (ASCII)
出力データ形式	WAV フォーマット (8KHz サンプリング, 16bit PCM, モノラル) データ *24KHzサンプリングにアップサンプリング機能を含む
声種	picoF4 Rev.2
関数 I/F	C 関数呼び出し
ライブラリサイズ	ROM: 約 28KByte, RAM: 約 500Byte

音声合成関数

CAqTkPicoF_Init aquestalk.h

説明	初期化
構文	uint8_t CAqTkPicoF_Init (uint32_t * <i>buf</i> , uint16_t <i>lenFrame</i> , const char * <i>key</i>)
引数	
<i>buf</i>	ワークバッファ。呼び出し側で確保。サイズは AQ_SIZE_WORKBUF
<i>lenFrame</i>	SyntheFrame()の呼出し毎に生成するサンプル数を指定(30-320)
<i>key</i>	ライセンスキー文字列を指定。NULL 指定で評価版としての動作(制限有)。
戻り値	0:正常終了 それ以外:エラーコード参照

CAqTkPicoF_SetKoe aquestalk.h

説明	音声記号列をセット。 この関数呼出し後、SyntheFrame()を連続して呼び出して音声波形を生成する
構文	uint8_t CAqTkPicoF_SetKoe (const uint8_t * <i>koe</i> , uint16_t <i>speed</i> , uint16_t <i>lenPause</i>)
引数	
<i>koe</i>	ローマ字音声記号列 (NULL 終端)。呼出し後、解放可能。
<i>speed</i>	発話速度 [%] 50-300 の間で指定 slow:50 fast:300 default:100
<i>lenPause</i>	最後のポーズ長[sample]。0xffffU を指定すると、内部の標準ポーズ長。(デフォルト)。
戻り値	0:正常終了 それ以外:エラーコード参照

説明	音声波形を生成
構文	uint8_t CAqTkPicoF_SyntheFrame (short *wav, uint16_t *pLen)
引数	
wav	音声波形出力バッファ(lenFrame×2byte 以上)。 最終フレームは *pSize<lenFrame の場合であっても残りのデータはゼロパディングされる
pLen	生成したデータのサンプル数を返す。 通常、最終フレーム以外は lenFrame で指定した値。
戻り値	0:正常終了 1:EOD それ以外:エラーコード参照

リサンプリング関数

説明	内部変数の初期化。 音声合成の単位ごとに関数呼び出す。
構文	void AqResample_Reset ()
引数	なし
戻り値	なし

説明	アップサンプリング処理(3 倍アップサンプリング) 1 サンプルずつ与えると、3 サンプルずつ返す
構文	void AqResample_Conv (short sin, short *sout)
引数	
sin	音声波形の 1 サンプルを指定
sout	音声波形の 3 サンプル返す。short[3]の配列を呼び出し側で用意。
戻り値	なし

関数呼び出しシーケンス

次コードを例に関数の呼び出しシーケンスを示します(コードはシーケンスを示すためのもので、実際に有効なソースコードではありません)。

1. CAqTkPicoF_Init()にて初期化とワークバッファを指定。
2. CAqTkPicoF_SetKoe()にて合成するメッセージを指定。
3. CAqTkPicoF_SyntheFrame()を繰り返し呼び出し、生成した 1 フレーム分の波形データを D/A に出力する。
4. 3.を、戻り値 1 (End of Data)が返るまで繰り返す。

さらに他のメッセージを合成する場合は、2.のステップから繰り返します。ただし、ワークバッファを解放した場合は 1.のステップから行います。

なお、音声合成を他の処理と並列に行う場合は、4.のループ処理を展開するようにします。

```
1  #define   FRAME_LEN 32
2  uint32_t buf[AQ_SIZE_WORKBUF];
3  const uint8_t koe[]="konnichiwa.";
4
5  uint8_t iret;
6  CAqTkPicoF_Init(buf, FRAME_LEN, "xxx-xxx-xxx");
7  iret = CAqTkPicoF_SetKoe(koe, 100, 256);
8  if(iret!=0) <<<error proc>>>
9
10 for(;;){
11     short wavFrame[FRAME_LEN];
12     uint16_t size;
13     iret = CAqTkPicoF_SyntheFrame(wavFrame, &size);
14     if(iret==1) break; // End of Data
15     if(iret!=0) <<<error proc>>>
16     <<< Copy "wavFrame" data to D/A output buffer. Length is "size" or FRAME_LEN >>>
17 }
```

ArduinoIDE や ESP-IDF 環境での具体的な使用法は、ブログで公開していますので、「AquesTalk ESP23」などのキーワードで検索して、それを参照ください。

音声記号列

AquesTalk pico は、ローマ字表記の音声記号列から音声を作成します。漢字を含んだテキスト文字列から音声を合成するときは、別途、言語処理ライブラリ AqKanji2Roman を用いて漢字仮名交じりテキストから音声記号列に変換する必要があります。

音声記号列の詳細は、当社サイトのダウンロードのページの「ローマ字音声記号列仕様書」を参照してください。

エラーコード表

関数が返すエラーコードの内容は、次の通りです。

値	内容
100	その他のエラー
105	音声記号列に未定義の読み記号が指定された
106	音声記号列のタグの指定が正しくない
107	タグの長さが制限を越えている(または[>]が見つからない)
121	CAqTkPicoF_Init()で初期化されていない
122	関数の引数の指定が正しくない
123	ワークバッファが外部から書き換えられた
200	音声記号列が長すぎる

文書履歴

日付	版	変更箇所	更新内容	更新者
2018/03/22	1.0		新規作成	N.Y