

---

言語処理ライブラリ

「AqKanji2Koe-M」

標準インターフェース仕様書

株式会社アクエスト

[www.a-quest.com](http://www.a-quest.com)

---

## 1. 概要

本文書は、弊社の言語処理ライブラリ **AqKanji2Koe-M** の関数インターフェースを示したものです。  
本文書の内容は弊社標準の基本仕様であり、カスタマイズによってはインターフェース仕様が異なることがあります。実装の際には、ライブラリ提供の際に付属されるマニュアルを参照してください。

**AqKanji2Koe-M** は、漢字仮名混じりのテキストを AquesTalk シリーズ用の音声記号列に変換するライブラリです。

## 2. 基本仕様

入力	漢字仮名交じりテキスト(UTF8/UTF16LE)
出力	AquesTalk 用音声記号列 (かな表記/ローマ字表記)
辞書サイズ	2MB/3MB/7MB(約 10/17/38 万語)から予め選択
必要メモリ	ROM: 44KB(コード+データ。Cortex-M での実測値。構成や動作環境により変動) RAM: 21KB 辞書データは、SPI フラッシュや SD カードなどの外付けメモリに別途配置
その他	C 関数呼び出し、OS 不要、ヒープ、STL、例外、浮動小数演算は不使用

## 3. 関数 API

以下に、本言語処理ライブラリに含まれる関数を示します。

AqKanji2Koe\_M.h

### CAqK2K\_Create

説明	言語処理部の初期化
構文	<code>int CAqK2K_Create(uint8_t *workbuf, int sizeWorkBuf)</code>
引数	
<i>workbuf</i>	ワークバッファ。呼び出し側で確保。 サイズは SIZE_AQK2R_MIN_WORK_BUF(20480[byte])以上 4 バイトアライメント
<i>sizeWorkBuf</i>	<i>workbuf</i> のサイズ[byte]
戻り値	0:正常終了 それ以外:エラーコード参照

AqKanji2Koe\_M.h

### CAqK2K\_Release

説明	言語処理部の解放
構文	<code>void CAqK2K_Release()</code>
引数	なし
戻り値	なし

AqKanji2Koe\_M.h

**CAqK2K\_Convert**

<b>説明</b>	漢字かな混じり文(UTF8)を、かな音声記号列に変換
<b>構文</b>	int <b>CAqK2K_Convert</b> (const char * <i>kanji_utf8</i> , char16_t * <i>wKoe</i> , int <i>nBufKoe</i> )
<b>引数</b>	
<i>kanji_utf8</i>	漢字かな混じり文テキスト(UTF8,BOM 無)
<i>wKoe</i>	かな音声記号列が返る文字列配列(UTF16LE)
<i>nBufKoe</i>	<i>roman</i> の配列のサイズ[word]
<b>戻り値</b>	0:正常終了 それ以外:エラーコード参照

AqKanji2Koe\_M.h

**CAqK2K\_Convert\_u16**

<b>説明</b>	漢字かな混じり文(UTF16LE)を、かな音声記号列に変換
<b>構文</b>	int <b>CAqK2K_Convert_u16</b> (const char16_t * <i>kanji</i> , char16_t * <i>wKoe</i> , int <i>nBufKoe</i> )
<b>引数</b>	
<i>kanji</i>	漢字かな混じり文テキスト(UTF16LE)
<i>wKoe</i>	かな音声記号列が返る文字列配列(UTF16LE)
<i>nBufKoe</i>	<i>roman</i> の配列のサイズ[word]
<b>戻り値</b>	0:正常終了 それ以外:エラーコード参照

AqKanji2Koe\_M.h

**CAqK2K\_ConvRoman**

<b>説明</b>	漢字かな混じり文(UTF8)を、ローマ字音声記号列に変換
<b>構文</b>	int <b>CAqK2K_ConvRoman</b> (const char * <i>kanji_utf8</i> , char * <i>roman</i> , int <i>nBufRoman</i> )
<b>引数</b>	
<i>kanji_utf8</i>	漢字かな混じり文テキスト(UTF8,BOM 無)
<i>roman</i>	ローマ字音声記号列が返る文字列配列
<i>nBufRoman</i>	<i>roman</i> の配列のサイズ[byte]
<b>戻り値</b>	0:正常終了 それ以外:エラーコード参照

## CAqK2K\_ConvRoman\_u16

説明	漢字かな混じり文(UTF16LE)を、ローマ字音声記号列に変換
構文	int <b>CAqK2K_ConvRoman_u16</b> (const char16_t * <i>kanji</i> , char * <i>roman</i> , int <i>nBufRoman</i> )
引数	
<i>kanji</i>	漢字かな混じり文テキスト(UTF16LE)
<i>roman</i>	ローマ字音声記号列が返る文字列配列
<i>nBufRoman</i>	<i>roman</i> の配列のサイズ[byte]
戻り値	0:正常終了 それ以外:エラーコード参照

## 4. アプリ側で用意する関数

以下の辞書データ(aqdic.m.bin)にアクセスする関数を開発者サイドで実装する必要があります。これらの関数は言語処理ライブラリから呼び出されます。

CでなくC++で実装する場合は、関数名がマングルされないように、extern "C" を指定してください。

## aqdic\_open

説明	辞書データアクセスの初期化 CAqK2K_Create()内から一度だけ呼び出される
構文	extern "C" size_t <b>aqdic_open</b> ()
引数	なし
戻り値	仮想的な辞書データの先頭アドレスを返す(0以外の任意の値。4byte アライメント) 初期化に失敗したときは0を返す

## aqdic\_close

説明	辞書データアクセスの終了 CAqK2K_Release()内から一度だけ呼び出される
構文	extern "C" void <b>aqdic_close</b> ()
引数	なし
戻り値	なし

## aqdic\_read

説明	辞書データの読み込み CAqK2K_Convert()/CAqK2K_ConvertW()から複数回呼び出される
構文	extern "C" size_t <b>aqdic_read</b> (size_t pos, size_t size, void *buf)
引数	
pos	読み込むデータの先頭アドレス[byte] = aqdic_open()で返したアドレス + 辞書データのオフセット
size	読み込むデータのサイズ[byte]
buf	読み込むデータ配列 uint8_t(size)
戻り値	読み込んだバイト数

以下に、辞書データを SD カード上のファイルとして配置し、そこから読み込む実装例を示します。

aqdic\_access.cpp

```

1  #include <SD.h>
2  const uint8_t SD_CS = 5;
3  const size_t ADDR_ORG = 0x10001000;
4  #define FILE_DIC "/aq_dic/aqdic_m.bin"
5
6  static File fp;
7
8  extern "C" size_t aqdic_open()
9  {
10     if(!SD.begin(SD_CS)) return 0; // err
11
12     fp = SD.open(FILE_DIC);
13     if(!fp) return 0; // err
14     return ADDR_ORG; // ok
15 }
16
17 extern "C" void aqdic_close()
18 {
19     if(fp) fp.close();
20 }
21
22 extern "C" size_t aqdic_read(size_t pos, size_t size, void *buf)
23 {
24     fp.seek(pos-ADDR_ORG);
25     return fp.read((uint8_t*)buf,size);
26 }

```

2 行目: SD カードの CS のピンアサイン

3 行目: 仮想的な辞書データの先頭アドレスを指定 (NULL 以外の任意。但し 4byte アライメント)

4 行目: 辞書データファイルの SD メモリカード上の配置場所 (ここでは、aq\_dic フォルダの下)

## 5. カスタマイズ項目

### 5.1. 辞書データの選択

辞書データは語彙サイズの異なる3種類が用意されています。  
本ライブラリ自体が辞書データに依存するため、予め使用する辞書データを決めておく必要があります。

### 5.2. 辞書データのアクセスが遅い場合

辞書データのアクセス速度が本ライブラリの処理速度に大きく影響します。  
一般的に、SPI や QSPI フラッシュメモリのアクセスに比べて、SD メモリカードのデータアクセスは遅くなります。  
そこで、辞書データの一部をコードメモリ上に配置することで処理を高速化するカスタマイズを承ります。コードサイズがその分増えますが、辞書データのアクセスが遅い場合にはこのカスタマイズを検討ください。

### 5.3. ユーザ辞書

AqKanji2Koe-M にはユーザ辞書がありません。語彙の追加や修正は当社で承ります。

## 6. エラーコード表

関数の戻り値とそのエラー内容を以下に示します。

100	その他のエラー
101	関数呼び出し時の引数指定エラー
102	ワークバッファ不足(入力テキストが長すぎる)
103	出力の音声記号列バッファ不足
104	CAqK2K_Create()で初期化されていない
200	辞書データのオープンエラー
210-218	辞書データが正しくない(フォーマットエラー)
220-222	辞書データが正しくない(バージョンエラー)

以上